Understanding the human diseasome and the structure of modules in the interactome

RAMTIN KERAMATI[1] AND BAZYLI KLOCKIEWICZ[1]

[1] *Institute for Computational and Mathematical Engineering (iCME), Stanford University, Stanford, CA*

December 11, 2016

In the past decade we have seen a great growth in the human molecular interaction data, causing huge attention to the emerging field of Network Medicine. In this paper, we analyze the gene-gene interaction data, often referred to as the *proteome*. First, we use the proteome to derive the human diseasome, which is a network of human diseases. We develop a method that allows to normalize the diseasome to unfold rich community structure as well as to detect associations between similar diseases. We use the same approach to study the drugsome, which is an analogous network of drugs and has not been previously studied systematically. In the second part of the paper, we apply tools from topological data analysis to study the internal structure of the subnetworks in the proteome that represent diseases, drugs and metabolic pathways. Our results show that these subnetworks do indeed have different topological structure but that the potential superiority of topological data analysis over classical approach should be investigated further.

Keywords: Diseaseome, Drugsome, Network Medicine, Module Similarity, Topological Data Analysis, Persistent Barcodes

1. INTRODUCTION

We examine the human protein-protein interaction network (the *proteome*). The nodes of the network are the human genes (the whole genome of ca. 21,500 genes)[1]. An edge between a pair of nodes exists in the network if there is any type of interaction between the genes that was confirmed by independent studies. However, it is estimated that only about 20-30 percent of the interactions have been detected so far ([2], [1]). This paper is based on a dataset of interactions that was aggregated from sources listed in [3].

The nodes belong to groups called modules. A module is a set of genes that share a common characteristic. We distinguish disease modules (i.e. genes associated with a certain disease), drug modules (genes targeted by a given drug), metabolic pathways, and functionality (genes that share a common function). A single node can belong to multiple modules.

We aim to answer two major questions. The first one concerns the structure of the diseasome, which is a meta network of disease modules. So far, most disease classifications were based on their symptoms. We study the structure of the diseaseome more systematically. We propose a method of normalizing the data to account for diseases that have very high degrees and dominate the network. Using modularity tools, we find the clusters of diseases present in the network. We also find diseases that have significantly more connections than expected based on the random null model, thus exposing potential non-intuitive associations between seemingly unrelated conditions. We also obtain principle-based visualizations of the diseasome. We use similar tools to examine the drugsome, i.e. the meta network of drug modules. This network has not been researched yet and one could expect a similar structure of both graphs.

The second major question that we address is the topological structure of different types of modules in the network (in the sense of the topological data analysis). This issue has not been examined at all and even the very existence of any difference in the structure of the modules is unknown. On the other hand, standard network science tools for analyzing the modules are limited; the modules typically do not induce one connected component in the network. This means that some notion of distance is needed to measure the relations between nodes.

Our minor goal was to obtain principle-based visualizations of the networks. Although many papers contain beautiful visualizations of the diseasome and other networks, they do not typically mention the methods that were used to obtain them; the visualizations seem much more organized than one would expect based on the "messy" results described that typically only indicate statistical significance of certain associations. In this paper, all visualizations are based directly on the data and we describe the ways in which they are obtained.

[1] Often a gene is identified with the protein that it encodes and the two terms are used interchangeably.

Graph Statistics		
	Diseasome	Null Model
# Nodes	473	473
# Edges	107282	107294
Diameter	2	2
Nodes in LWCC	473	473
Modularity	0.051	0
	DrugSome	Null Model
# Nodes	645	645
# Edges	169973	147942
Diameter	3	2
Nodes in LWCC	645	645
Modularity	0.006	0

Table 1. Statitics of networks, both diseasome and drugsome are weighted networks. *Top*: The original diseasome and the rewired null model. Both networks are very dense which makes the comparison difficult. The rewired diseasome cannot be partitioned into sets with non-zero modularity which indicates the randomness of the null model. *Bottom*: Drugsome and rewired null model, both of them are highly connected networks.

2. RELATED WORK

The problem of defining similarity between modules in the proteome was addressed in [2]. The authors calculate a minimum required network coverage to observe a disease module and conclude that the modules can be observed and analyzed in the interactome despite the incompleteness of the interactions. The authors define a certain notion of distance between modules, showing its usefulness as an indicator of their similarity; in particular they show that diseases close with respect to the defined metric tend to be related phenotypically. It is unclear, however, if this method can unveil any community structure in the diseasome.

Another approach to study the disease-disease interactions was taken in [4]. Text-mining techniques were used to identify similar diseases based on their phenotypes, which were then mapped to the proteome to confirm their similarity; this yields classification of diseases based on symptoms.

The structure of disease modules in the proteome was first studied in [5]. Graph-theoretical approach was applied to unveil certain basic common characteristics of the disease modules.

3. DISEASOME

Armed with the human proteome, we build a meta network of diseases called the *diseasome*. Nodes of this network are the disease modules whereas an edge between two diseases exists if they have a gene interaction in the interacotme, with weight equal to the number of interactions. Figure 1. shows a scheme of how the diseasome is created. The diseasome is a highly connected network; this suggests that the underlying causes of a particular disease are typically spread throughout the genome (however, this does not mean that most connections are not concentrated around some ”place” in the network); some of the basic statistics are shown in Table 1.

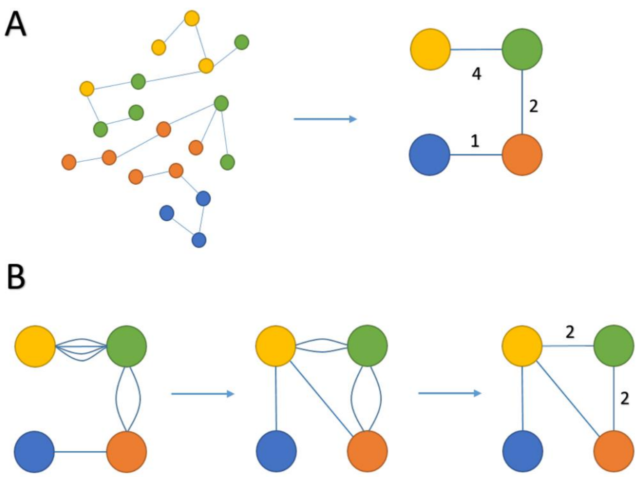

Fig. 1. *A*. Building the diseasome based on the underlying interactome: 1. Find the genes corresponding to the disease module 2. Create a super node 3. Calculate the edges and weights based on the number of shared genes. *B*. Rewiring the Diseasome: 1. Create a multigraph based on the weights of each edge 2. Rewire the multigraph 3. Retrieve the new weighted graph.

The weight of an edge connecting diseases denotes the number of genes that those diseases share, which should capture their similarity in terms of genes involved. On the other hand, finding the most central nodes, for example, can give us a sense of which diseases have the most central role. Some of the basic statistics of the diseasome can be found in Table 2.

A. Straightforward diseasome is biased

We found that the definition of the diseasome given above causes large disease modules to have huge influence over the network (typically cancer modules). A disease module with a huge number of nodes will inevitably be connected by high weight edges to many other diseases. On the other hand, the fraction of the nodes shared with the other disease may be insignificant, making the interpretation of the edge weight problematic in terms of measuring similarity between diseases. Here, we propose a different and relatively simple method to build a (normalized) diseasome, that is not biased by the size of the module and is aimed at better understanding of the relationships between the disease modules. [2]

B. Normalized Diseasome

We are interested in finding to what extent the weight of the connection between two diseases is caused by their actual (un)similarity, as opposed to just the high (low) number of outgoing edges from the diseases. To achieve that, we construct a null model in the following way.

B.1. Null Model

We treat the diseasome as an unweighted multigraph, repeating each edge w times, where w denotes its weight. We rewire the graph using the Configuration model, as depicted in Figure 1. To

[2] Yet another source of bias may be the fact that some diseases have been studied more than others, for example cancers; we do not have sufficient data to take this into account in designing the null model.

Diseasome Analysis	
	Name
Centrality	Neoplasm Invasiveness
	Stomach Neoplasms
	Cardiomegaly
	Diabetes
	Neoplasm Metastasis
	Prostatic Neoplasms
Page Rank	Stomach Neoplasms
	Diabetes
	Stomach Neoplasms
	HIV Infections
	Melanoma
	Name
Similarity	Breast Cancer and Prostatic Cancer
	Stomach Cancer and Breast Cancer
	Stomach Cancer and Prostatic Cancer

Table 2. Central and similar diseases in the diseasome. We found that the diseasome is dominated by cancers and tumores. *Top*: Nodes with the highest betweeneess centrality and the highest PageRank in the original diseasome. *Bottom*: Most similar diseases in the original diseasome, i.e. edges with the highest weight.

achieve good statistical properties, we would like the process to achieve stationarity condition. Using the benchmarks in [6] we perform $100|E|$ rewires, where $|E|$ is the number of edges in the multigraph. Both graphs are very dense which makes it difficult to confirm that the rewiring process achieved stationarity based on basic statistics (see Table 1.). We found however that the modularity maximization tools were unable to find a partition with non-zero modularity (see section C) which indicates that the rewired null model has indeed a random structure.

B.2. Normalized Diseasome

In this way, we obtain one hundred randomly generated diseasomes. Given diseases i and j, we denote by w_{ij} the weight of the edge (i, j) in the original Diseasome; analogously, for each rewired network r, we have w^r_{ij}. We denote the standard deviation of w^r_{ij} by $\sigma(w^r_{ij})$ and define the weight Z_{ij} of the edge between diseases using the Z-score as follows:

$$Z_{ij} = \frac{\overline{w^r_{ij}} - w_{ij}}{\sigma(w^r_{ij})} \qquad \textbf{(1)}$$

Z_{ij} is meant to capture how much the connection between two diseases differs from the one in a random rewired graph. Two diseases with high Z-score should be similar because the connections between them are highly non-random.

We note here that another possible way to rewire the graph would be to rewire the interactome itself, and then build the diseasome on top of the rewired graph, keeping the nodes belonging to the diseases fixed. However, in this case, disease modules with high internal edge density could potentially distort the results, as the edges connecting the nodes belonging to the same disease module could after rewiring connect also different disease modules. We note that this approach gave us different (worse) results, and we do not describe it here.

Normalized Diseasome Analysis		
	Value	Name
1^{st} Central Node	0.00430	Brain disease
2^{nd} Central Node	0.00429	Glomerulonephritis
3^{rd} Central Node	0.00428	Breast Neoplasms
1^{st} Page Rank	0.00588	Prostatic Neoplasms
2^{nd} Page Rank	0.00586	Breast Neoplasms
3^{rd} Page Rank	0.00540	Diabetes

Table 3. *Top*: Most central diseases in the Z-score-based Diseasome. Centrality is defined as betweenness centrality and all values are normalized between 1 and 0. *Bottom*: Nodes with highest Page Rank score in the Z-score-based Diseasome.

The obtained measure differs from the original one. For example, Diabetes has the highest betweenness centrality in the original diseasome, but in the normalized diseasome is the 15^{th} with respect to that measure. Table 2. along with Table 3. show some of the differences between the two networks.

Figure 2 shows a Z-score-based network visualized using Force Atlas algorithm in which nodes with highest Page Rank score have fixed positions on the plane. In the visualization, we only show the edges with Z-score in the top 1% percentile. Table 3 describes some of the key statistics of this network.

C. Communities

We test our notion of pairwise similarity by using it as an input to the modularity-maximization-based clustering algorithm as defined in [8] and [9]. For a given parameter t (called resolution), the algorithm aims to find the partition $\mathcal{P}$ of the nodes into disjoint subsets, that maximizes:

$$R(t) = \sum_{C \in \mathcal{P}} P(C, t) - P(C, \infty) \qquad \textbf{(2)}$$

where $P(C, t)$ is the probability of a random walk that started inside the community C to remain inside it at time t, and $P(C, \infty)$ is the probability that the random walk is in C at stationarity. Thus the parameter t accounts for the desired sizes of the communities. This problem is NP-hard, but there are some approximation algorithms that work well in practice, and can be restarted with randomized input for better reliability. We use the implementation available in Gephi [10]. Of course, the assumption in this approach is that the communities do not overalap, or otherwise stated, the algorithm will find the best partition assuming that there are no overlaps between communities.

To cross-check the quality of the output of our algorithm, for each t we compute the modularity of the partition defined by :

$$Q = \frac{1}{2m} \sum_{i,j} \left[\left(A_{i,j} - \frac{d_i d_j}{2m} \right) I(i, j) \right]$$

where $I(i, j)$ equals 1 if the nodes i belong to the same community and 0 otherwise, A is the weighted adjacency matrix, and d_i

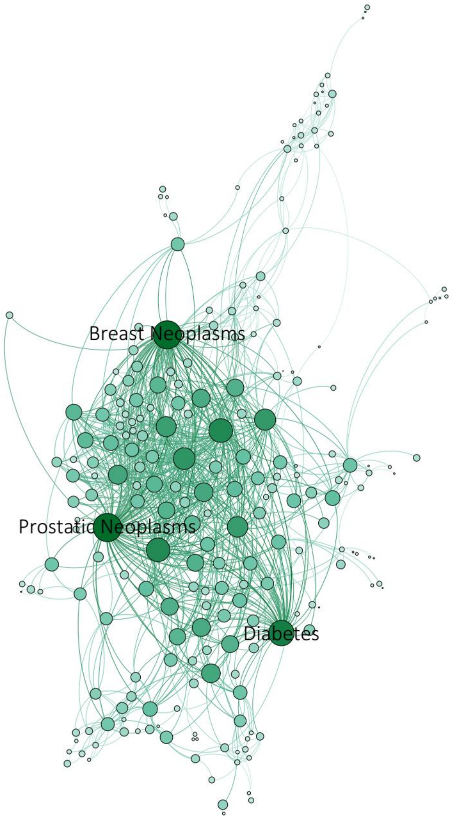

Fig. 2. Z-score based Diseasome. The visualization was performed using ForceAtlas algorithm implemented in [10]. Only top 1% Z-score edges were used. The size of the node indicates PageRank value, and the nodes with the highest PageRank have fixed positions (labeled with name).

is the sum of the weightd of edges adjacent to node i. This measure quantifies the quality of the partitioning in the sense that higher values indicate that the connections within a community are non-random. It should be maximized around $t = 1$ but its value at that point as well as its behavior versus the number of found communities for different values of t helps to understand the structure of the network and choose the most meaningful partition.

We perform the community search in two ways. First, we only use the edges whose Z-score is positive. We find three communities. One can see the plot of the number of communities captured by the algorithm for different values of resolution as well as the corresponding value of modularity obtained and the size of the smallest community in Figure 3. The modularity is maximized around $t = 1.0$ and the number of detected communities stabilizes. Also, at this point the communities have comparable sizes.

By inspection, we find that one of the communities contains in particular almost all of the cancers and tumors. Another one contains most of the diseases connected with brain, such as autistic disorder, Parkinson's disease or Alzheimer's disease. The third one is more difficult to describe, it contains many types of diseases. In the second approach, we only consider the edges whose Z-score belongs to the top 10th percentile. This approach seems to unfold richer community structure. Inspecting the same plot as before (Figure 5) does not give as clear conclusions

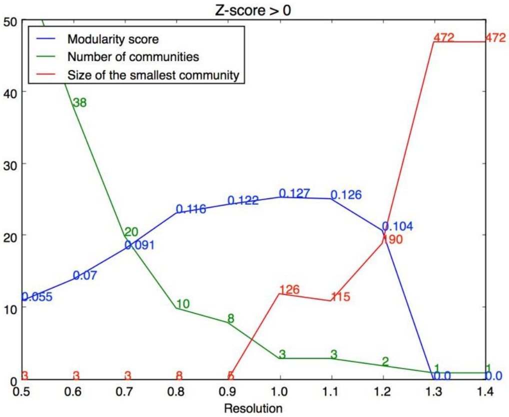

Fig. 3. The plot of the change of modularity score and the number of found communities for different values of resolution t. The analyzed network is the normalized diseasome where only edges with positive Z-scores are considered.

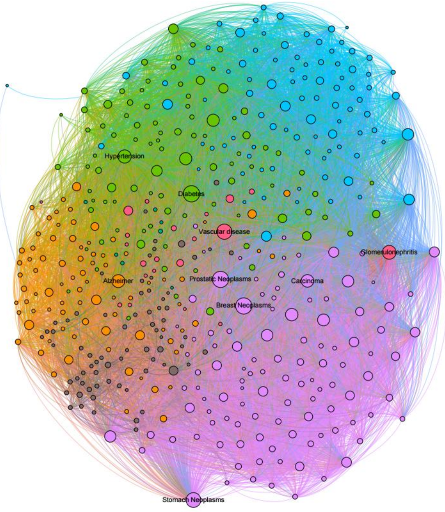

Fig. 4. Communities in the modified Diseasome graph based on the modularity defined in [8], There are 7 communities in this network as marked by the different colors. The visualization is performed using Gephi [10].

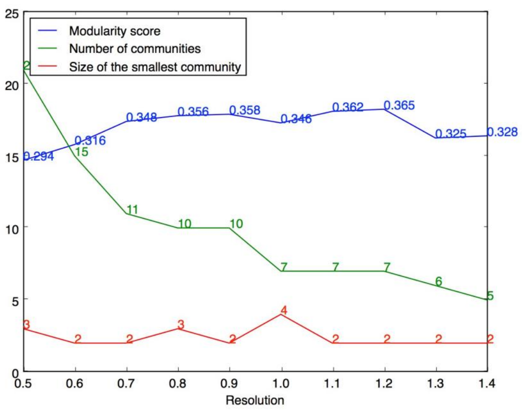

Fig. 5. The plot of the change of modularity score and the number of found communities for different values of resolution t. The analyzed network is the normalized diseasome where only edges from the 10th percentile of Z-scores are considered.

as before but the modularity is maximized at about $t = 1$ and around this point one obtains consistently seven communities.

The communities containing tumors (125 modules) and diseases connected with brain (80 modules), respectively, are now "purer". We also obtain a couple of other interesting communities. One of them (76 modules) is rich in diseases connected with infections, such as tuberculosis, pneumonia or influenza. It also contains allergies and auto-immunological diseases. Another community (109 modules) contains diseases of the organs in the abdomen, such as kidney failure, liver cirrhosis, or diabetes. Yet another one (50 modules) contains rare genetical diseases, such as Situs inversus, Seckel syndrome or Nephronophthisis. We also obtain a small community (18 modules) of diseases connected with bones and connective tissue, incuding gout, osteoporosis and rickets.

For both approaches, we cross-check our results against the output of the force-directed graph drawing algorithm ForceAtlas to confirm that the communities that we found are proximate in the network. The results with communities marked by different colors can be seen in Figure 4.

D. Pairs of similar diseases.

The Z-score normalization is designed to unfold the pairs of diseases that are really (un)similar to each other in the sense that the number of shared genes is highly non-random. Table 4. shows the most similar diseases based on the edge weights in the modified Diseasome (with Z-score above 100).

Although the statistical significance of the similarity of the pairs of diseases should be confirmed by independent study [3], by examining the relations between the diseases we find that most of them are clinically very related. For example, Hermansky-Pudlak syndrome is a direct cause of Albinism. Zellweger syndrome and Rhizomelic chondrodysplasia punctata are closely related brain disorders, whereas Obesity and Polydactyly belong to the symptoms of the Bardet-Biedl syndrome. There is one unexpected association, however – the observed similarity between Adrenoleukodystrophy and Rhizomelic chondrodysplasia punctata. All in all, it is evident that our method can capture related diseases in the network and might point to potential unexpected associations between diseases, not present directly in the symptoms.

Similar Diseases	
Oculocutaneous albinism	Hermansky-Pudlak syndrome
Zellweger syndrome	Rhizomelic chondrodysplasia punctata
Zellweger syndrome	Adrenoleukodystrophy
Cystic kidney	Nephronophthisis
Bardet-Biedl syndrome	Fundus dystrophy
Obesity	Bardet-Biedl syndrome
Adrenoleukodystrophy	Rhizomelic chondrodysplasia punctata
Bardet-Biedl syndrome	Polydactyly

Table 4. Pairs of most similar diseases based on the defined Z-score.

Our results in this section show that 1) the straightforward meta-network built on top of an original network can be biased and hard to analyze and that 2) redefining the weights of the edges using the Z-score can help unfold real underlying structure, especially in terms of similarity of different modules.

E. Drugsome

In a similar way that we defined the diseasome, we form the drugsome by considering the drug modules. Table 1 shows some of the basic statistics of drugsome, and its null model obtained with our method. Figure 6 is a visualization of the Drugsome, where only top 1% Z-score edges are visible.

We found that the drugsome is a much more similar to a random network than the diseasome. In particular, the modularity maximization tools failed to obtain a partition with non-zero modularity in the Z-score-based normalized drugsome; this shows that drugsome does not have any simple community structure.

This result is consistent with [1] which shows that drugs rarely target the disease modules directly, and often have palliative effect rather than treat the cause of the disease itself. Our analysis also suggests that, perhaps expectedly, even if the drugs target proteins associated with diseases they are supposed to cure, they probably also target a number random proteins in the network; this could explain why many drugs have side effects unrelated to the symptoms that they target.

4. STRUCTURAL DIFFERENCES OF MODULES IN THE PROTEOME.

A. Topological structure of the modules.

As we have noted above, the modules in the proteome (i.e. the subnetworks corresponding to diseases, drugs or metabolic pathways) are not always connected. On the other hand, their nodes are typically located close to each other in the network. Taking the subgraph induced by the module may be oblivious to their distance in the original graph. Therefore, we compute a notion of distance (see below) for every pair of nodes in the whole

[3] We are performing a multiple hypothesis test here. Z-score can be interperted as a p-value in testing whether any two given diseases are related or not. Looking at the highest p-values to conclude that the corresponding diseases are related is not a statistically sound approach. To test whether there exists a pair with a significant p-value one can run the *Bonferroni Test*. On the other hand, one can choose a random pair and repeat our simulation to test for similarity.

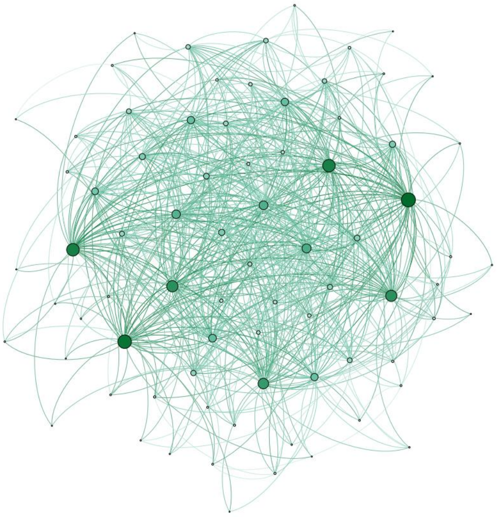

Fig. 6. Z-score based drugsome. The visualization was performed using ForceAtlas algorithm implemented in [10]. Only top 1% Z-score edges were used. The size of the node indicates the PageRank, and the nodes with the highest PageRank have fixed positions.

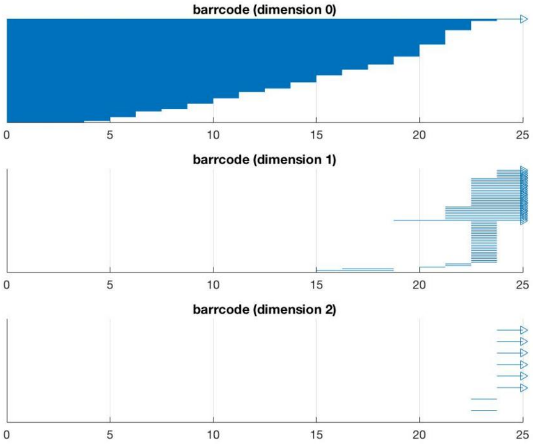

Fig. 7. An example of a barcode. Each horizontal line corresponds to a homology generator. There are three sub-barcodes, one for each dimension. The top one corresponds to dimension 0 (lines are connected components of the graph), the middle one corresponds to dimension 1 (lines are empty cycles in the graph), The bottom one corresponds to dimension 2 (lines are voids that is, empty spaces enclosed by triangles). The x-axis is the time t and the endpoints of a line denote the appearance (birth) and disappearance (death) of the homology generator. In the case above, the number of connected components decreases until the graph becomes connected at about $t = 23$, when there is only one line left in dimension 0. The first 1-dimensional hole appears when $t = 15$ and disappears at about $t = 19$. First voids are born at about $t = 22.5$.

network, and use this value as the distance between nodes in the module. This strategy should mitigate the effects of incompleteness of the interactions data in the network. We use the LTHT(β) distance between nodes in a graph as defined in [7]. This distance notion leverages between shortest paths and expected commute times between nodes. When $\beta = 0$, it becomes the stationary distribution of a random walk. When $\beta \to \infty$, it converges to the shortest path distance. The rationale for using this definition is that two nodes that are close in the full network should have on average more and shorter paths connecting them in the incomplete network.

Using this notion of pairwise distance, one can define persistent barcodes of the modules. For $t > 0$, we consider the Vietoris-Rips simplicial complex on the set of the nodes. In this complex, a set $\{v_1, \ldots, v_n\}$ of n nodes forms an $(n-1)$-dimensional simplex at time t iff for all $1 \leq i, j \leq n$ we have $\text{dist}(v_i, v_j) < t$, where $\text{dist}(v_i, v_j)$ is the distance that we obtain from the LTHT calculations. We only do this for $n = 1, 2, 3$, and $t < \max(dist(v_i, v_j)$ in which case it is easy to understand the Vietoris-Rips complex at time t: 1) the nodes always belong to the complex, 2) if the distance between two nodes is less than t, they are connected by an edge, 3) if three given nodes are such that their pairwise distances are less than t, they form a (filled) triangle. We use the Vietoris-Rips complex, because it is defined precisely by the pairwise distance data, and does not depend on any geometric embedding of the complex (this is not true for alpha- or Cech complexes, which are other possible methods to build simplicial complexes on top of the data).

For each specified time t, we compute the geometry of thus obtained simplicial complex (see below). By increasing t, we observe how the geometry of the module evolves over time. The geometric features that are really important and not a byproduct of the noise in the data, should have longer lifetimes. For reference on persistent homology, see [11]. The persistence algorithm [12] is implemented in package [14] and outputs the persistence barcode (see Figure 7). The barcodes show the generators of homology groups that are present at different times t in the persistence algorithm (again, we only look at zero-, one- and two-dimensional generators). For a reference on homology, see [13]. In our case, it has an intuitive meaning: each generator in dimension 0 corresponds to a connected component of the graph; a generator in dimension 1 is an empty *hole*, that is, a cycle in the graph whose interior can not be filled with triangles (Figure 8); a generator in dimension 2 is a *void*, that is, an empty space fully enclosed by triangles.

We compute the distances using the LTHT metric with $\beta = 0.01, 0.1$ and obtain barcodes for the modules (in total, we analyze 473 disease modules, 645 drug modules and 292 metabolic pathway modules). The barcode can be understood as a collection of three sequences $(b_1, d_1, \ldots, b_m, d_m)$, one sequence for each dimension, where (b_i, d_i) denotes the birth and death of the i-th homology group generator. The issue of what features of the barcodes should be used to compare them is an open area of research. The mathematical notion of similarity between barcodes, called the bottleneck distance, is not appropriate in our case, because this notion will not distinguish spaces with the same local structure if they have different sizes [15]. The other possible approach is featurizing the barcodes by certain sym-

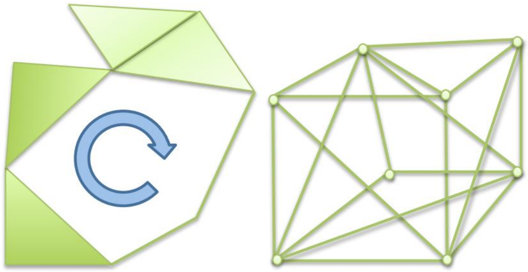

Fig. 8. *left*, An example of a *hole*. *Right*, an example of a void in the space. The hole is represented by the empty space in the middle. Every triad becomes a simplex. The triads were filled on the picture on the left but they should be thought of as filled on the right as well. Void is the empty space inside the cube.

metric polynomials [16]. We follow this approach, arbitrarily choosing simple, and easy-to-understand features of barcodes. We use four main features in each dimension. Thus for each barcode we define a vector $(x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3, w_1, w_2, w_2)$, where:

1. x_i is the minimum birth time of a generator in dimension $i \neq 0$ (or $\max t$ if no such generator exists)

2. y_i is the average lifespan of a generator in dimension $i = 1, 2, 3$

3. z_i is the average death time of a generator in dimension $i = 1, 2, 3$

4. w_i is the average birth time of a generator in dimension $i = 1, 2, 3$.

We thus obtain eleven features for each barcode. In addition to their simplicity, the empirical distributions of these features across different types of modules for $\beta = 0.1$ were significantly different as confirmed by the Kolmogorov-Smirnov tests (the largest p-value being 0.044). Since the p-values were much lower in the case of $\beta = 0.1$ than $\beta = 0.01$, we perform our tests only for this value.

As a preliminary step to see if thus defined topological information bears any meaning, we use our features as an input to the t-SNE algorithm [17] to visualize the data in the two-dimensional plane. We find that the algorithm tends to bring the modules of the same type together, as can be observed in figure 9. This is promising as it suggests that the type of the module can possibly be predicted based on the barcode.

We would like to emphasize here that this choice of features implicitly contains some information about distances between nodes because it may contain the scale information. It is the purpose of the next section to understand what type of information is contained in the barcodes as well as assess its predictive power. It should be noted that the complexity of the persistence algorithm is $O(n^3)$ so good evidence of its usefulness is needed to consider it a tool of choice in this type of analysis.

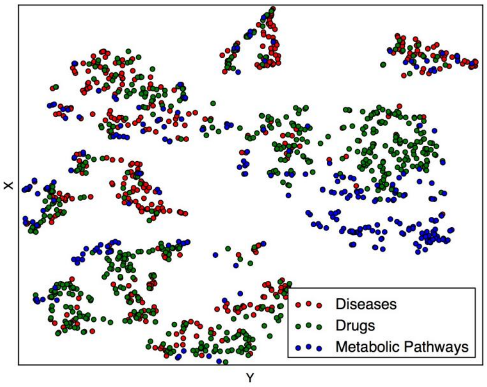

Fig. 9. The result of the t-SNE embedding algorithm applied to featurized barcodes. The algorithm brings modules of the same type together.

Further in this section, we want to assess weather the simple features of barcodes that we defined can help distinguish drugs from diseases or metabolic pathways. For comparison, we separately analyze the (typically disconnected) subgraphs induced by the modules. We compute some of their features, including those that in particular should implicitly be captured by the barcodes but can be obtained more easily from the graphs. We choose this approach also because we want to examine to what extent the fact of the modules being disconnected makes the analysis more difficult when traditional graph-based tools are used.

B. Classical and Higher Order Structures

For each induced subgraph, we compute: the number of nodes, diameter, clustering coefficient and the average degree, as well as the number of triads and some other higher order structures (see Figure 10, which shows the structures that we have used). One can see that on average the probability of a disease to have at least one of those structures is lower than pathways or drugs. For example, any type of the 4^{th} order structures are present in only 40% of diseases, whereas 83% of drugs and 91% of pathways have at least one. Our choice of these structures is also motivated by the fact that some of them can be implicitly captured by the barcodes.

C. Prediction. Classical vs. Topology vs. Higher Order

Using the different features that we extracted from our network, we can try to train a model to classify any given module into one of three groups: Diseases, Drugs, (Metabolic) Pathways. Here we use standard machine learning algorithms for classification. Splitting the data (1400 subgraphs) into two sets: the train set (75% of modules) and the test set (25% of modules), we train SVM, Logistic regression and Random Forest classifiers, and evaluate their prediction accuracy. Random prediction based on the number of different types in the data (that is, 473 Diseases, 645 Drugs and 292 Pathways) will result in 41% prediction accuracy.

We train the algorithms using 3 different types of features (so we obtain three separate experiments): *Classical*, which consists of the number of nodes, average degree, diameter, clustering coefficient and the number of triads in the induced subgraph; *Higher Order*, which consist of the Number of nodes, Average degree, and the number of 4^{th} order structures of Type I, Type II and Type III (Figure 10); *Topological*, which consists of the first 5 features of barcodes (these features are the minimum birthtimes of holes and their average lifespan in each dimension). In this way, in each experiment we perform classification based on five

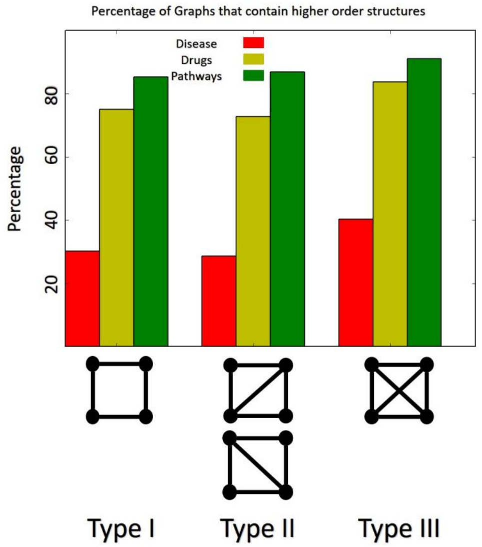

Fig. 10. Percentage of graphs that contain the structures indicated. In total, 40% of diseases have any kind of 4^{th} order structure, compared to 83% of drugs and 91% of pathways. The presence (or absence) of these structures can be tested as a potential indicator to classify the modules. On the other hand, Type I structure could appear as a short line in the barcode.

features. Table 5 shows the predicting accuracy for the SVM, Logistic Regression and Random Forrest classifiers.

As shown in Table 5, using SVM, or Logistic Regression, both Topological and Higher Order approach have significantly better accuracy than the Classical approach, and they perform better than random; However using Random Forrest classifier we obtain similar results in all approaches. Note that the Topological approach does not contain any explicit information of the graph, such as the number of nodes, average degree, etc.

We noticed that adding the number of nodes as an extra feature to the Topological approach can increase the accuracy, which suggests that using topological approach combined with Classical approach can have a strong prediction power. We found that using higher order structures have the highest prediction power given a fixed number of features, followed closely by the Topological approach (the difference being probably insignificant). Additionally, adding extra information like number of triads or clustering coefficient will not add to prediction power of this approach.

Another interesting observation is that for the Topological approach, the prediction accuracy of logistic regression and SVM is almost the same, but for Higher Order approach, logistic regression is not even much better than random. This suggests that using a mix of features, including topological, can have better predictive power. Using Average Degree, the first three Topological features and the Type I structure we obtained 72.46% prediction accuracy training a Random Forrest. [4]

Prediction accuracy; Random 41%			
Features	SVM	Logistic Reg	Rand Forrest
Topological	60.29%	58.53%	68.25%
Higher order	61.24%	45.70%	69.80%
Classical	56.12%	47.19%	69.79%
Mix	55.05%	45.01%	**72.46%**

Table 5. Prediction Accuracy of the three different approaches.

5. CONCLUSIONS

In the first part of the paper we showed that the straightforward analysis of a meta-network of disease modules is biased and difficult to analyze. We proposed a simple method of normalizing the meta-network using the configuration model and the Z-score that allowed for unfolding reach community structure consistent with the underlying nature of the modules. We also obtained evidence that the method can be used to find associations between modules in the meta-networks. We also found that the analogous network of drugs is much more complicated, and probably more random than the disease network. This finding is consistent with the current knowledge but adds to the understanding of drug design and potential sources of sides-effects in drugs.

In the second part of the paper, we obtained evidence of structural differences between different types of modules in the proteome. In particular, we found that the simple topological features of barcodes do indeed carry information about the modules. Our results suggest that topological features of networks can be used to classify them and perform much better than random, but that the chosen naive way of featurizing the barcodes was not superior to other simple approaches. Further work is needed to assess its real usefulness in practice. In particular, a more principled way of featurizing barcodes with symmetric polynomials (for example using metric learning), would be desirable and should be tested next. Also, as the barcodes are essentially functions of distance matrices, other machine learning methods that exploit the structure of the distance metric itself, could be tested as well.

The visualizations used in the paper were all principle-based. We believe that this is not of no importance, as one would expect visualizations in research papers in network science to not only serve the purpose of understanding a given notion, but – whenever possible – actually contain genuine information of the represented data.

ACKNOWLEDGEMENTS

We are very grateful to Marinka Zitnik who made available the data to us and suggested to us the topological data analysis part of the paper, including some of the approaches. We performed the LTHT and persistence homology computations on *madmax* clusters of the SNAP group.

REFERENCES

1. Barabási, Albert-László, Natali Gulbahce, and Joseph Loscalzo. *Network medicine: a network-based approach to human disease.* Nature Reviews Genetics 12, no. 1 (2011): 56-68
2. Menche, Jörg, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. *Uncovering disease-disease relationships through the incomplete interactome.* Science 347, no. 6224 (2015): 1257601.

[4] To avoid over-fitting on the test data, we did not test different sets of features to find the optimal one, we just selected those features and tested the results; the data is not large enough for a validation set.

3. Guney, Emre, Jörg Menche, Marc Vidal, and Albert-László Barábasi. *Network-based in silico drug efficacy screening.* Nature communications 7 (2016).
4. Hoehndorf, Robert and Schofield, Paul N and Gkoutos, Georgios V, *Analysis of the human diseasome using phenotype similarity between common, genetic, and infectious diseases*, Scientific reports, vol. 5, (2015), Nature Publishing Group
5. Goh, Kwang-Il, et al., *The human disease network.*, Proceedings of the National Academy of Sciences 104.21 (2007): 8685-8690.
6. Milo, Ron, et al., *On the uniform generation of random graphs with prescribed degree sequences.* arXiv preprint cond-mat/0312028 (2003).
7. Hashimoto, Tatsunori and Sun, Yi and Jaakkola, Tommi. *From random walks to distances on unweighted graphs.* Advances in Neural Information Processing Systems 28, p.3429–3437, 2015, Curran Associates, Inc.
8. Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre *Fast unfolding of communities in large networks (2008)*
9. R. Lambiotte, J.-C. Delvenne, M. Barahona, *Laplacian Dynamics and Multiscale Modular Structure in Networks*, arXiv preprint arXiv:0812.1770 (2008)
10. Bastian M., Heymann S., Jacomy M. *Gephi: an open source software for exploring and manipulating networks*, International AAAI Conference on Weblogs and Social Media, (2009)
11. Carlsson, Gunnar. *Topology and data.* Bulletin of the American Mathematical Society 46.2 (2009): 255-308.
12. Edelsbrunner, Herbert, David Letscher, and Afra Zomorodian. *Topological persistence and simplification.* Discrete and Computational Geometry 28.4 (2002): 511-533.
13. Hatcher, Allen. *Algebraic topology.* 2002. Cambridge UP, Cambridge 606.9.
14. Tausz, Andrew and Vejdemo-Johansson, Mikael and Adams, Henry, *JavaPlex: A research software package for persistent (co)homology*, Proceedings of ICMS 2014, Lecture Notes in Computer Science 8592, (2014): 129-136.
15. Cohen-Steiner, David, Herbert Edelsbrunner, and John Harer. *"Stability of persistence diagrams.* Discrete and Computational Geometry 37.1 (2007): 103-120.
16. Carlsson, Erik, Gunnar Carlsson, and Vin De Silva. *An algebraic topological method for feature identification.* International Journal of Computational Geometry and Applications 16.04 (2006): 291-314.
17. Maaten, Laurens van der, and Geoffrey Hinton. *Visualizing data using t-SNE.* Journal of Machine Learning Research 9.Nov (2008): 2579-2605.