

Predicting phage-host relationships using network fusion

Hielke Walinga (4373561)
hielkewalinga@gmail

June 2, 2020

1 Introduction

Here we propose a method for finding the phage-host relationships by combining different metrics using network fusion techniques.

1.1 Motivation

Bacteria are under a constant barrage of phages that act as viruses and integrate their DNA into the bacteria so that they can be multiplied. The bacterial communities are very diverse and almost all suffer from phages, making this an interesting study subject to learn more about ecological networks.

1.2 Network fusion

When creating a network from biological data, a common challenge involves being able to combine multiple sources of information. This problem is known as network fusion.

There are multiple different strategies to solve the network fusion problem. Here we propose to make use of an existing method called similarity network fusion (SNF) [2] with more details here [3]

For full details, consult the papers, but in short, create a similarity network for each source of information and then combine the networks using the following algorithm.

First generate a similarity matrix W for your information source, then normalize this matrix to create the status matrix. Take the self-similarity as $\frac{1}{2}$. Then create a kernel matrix P by normalizing using the k-nearest neighbours N_i which results in a sparse matrix P^* .

$$P(i, j) = \begin{cases} \frac{W(i, j)}{2 \sum_{k \neq i} W(i, k)} & j \neq i \\ \frac{1}{2} & j = i \end{cases} \quad (1)$$

$$P^*(i, j) = \begin{cases} \frac{W(i, j)}{\sum_{i \in N_j} W(i, j)} & i \in N_j \\ 0 & else \end{cases} \quad (2)$$

Update formulas to fuse the different status matrices P_v and P_w that represent two different information sources. Repeat the formulas till convergence.

$$\begin{aligned} P_{v,t+1} &= P_{w,t}^* \times P_{v,t} \times P_{w,t}^{*,T} \\ P_{w,t+1} &= P_{v,t}^* \times P_{w,t} \times P_{v,t}^{*,T} \end{aligned} \quad (3)$$

This can be extended to account for more than 2 information sources.

1.3 Representing the phage-host relationships in a bipartite graph

A bipartite graph is a network in which the vertices can be divided into two groups in which there are only links from one group to the other.

The phage-host relationships can be represented in a bipartite graph. Such a relationship is usually represented as a relation matrix in which an edge represents the possibility for a phage to be able to infect the bacteria.

1.4 Modularity and nestedness

...

1.5 What data is available

The proposed method tries to use information that is available for as well phage-host relationships as well as phage-phage and bacterium-bacterium relationships / similarities. Here summarized what these kinds of sources of information are.

1.5.1 Data sources for phage-host relationships

1.5.2 Data sources for phage-phage relationships

1.5.3 Data sources for bacterium-bacterium relationships

2 Related work

As far as I can find, the only computational work done on phage-host relationships involves a single metric which to measure the possibility of a phage-host relationship [1]. I could not find references to an attempt to combine this result or to combine it with phage-phage or bacterium-bacterium similarity.

3 Proposed method

3.1 Combining different information

First, the different sources of information for both the phages as the bacteria networks are combined to present a similarity graph for both the phages and the bacteria. For this the SNF method can be used.

3.2 Diffusion of the phage-host relationship

Now we have a similarity network for both the phages as the bacteria and the next step will be to ‘diffuse’ the information that exist for phage-host relationships (the connection graph) to other phage-host connection that don’t have this information available *using* the information for the similarity of the phages or the bacteria.

To do this we first pick what network to use (either the phage-phage F , or the bacterium-bacterium B). It is probably best to make use of the sparse similarity matrix here (F^* and B^*). Then diffuse using a column in the connection matrix, which corresponds to either a single phage or a single bacterium, by using the weights in the corresponding similarity graph. When working out the math it will become clear that this method can be done in a single step and has much resemblance to the previous SNF algorithm.

$$\begin{aligned} C_{t+1} &= F \times C_t \\ C_t &= (B \times C_{t-1}^T)^T \\ &= C_{t-1} \times B^T \end{aligned} \tag{4}$$

Combined :

$$C_{t+1} = F \times C_{t-1} \times B^T$$

3.2.1 Making it converging

Since we are alternating between two different similarity matrices I don't think it will converge. To make sure the algorithm converges, I propose to alter the connection matrix every iteration by gradually reducing the percentage of weight that gets moved around to zero percent.

This can be mathematically be accomplished by altering the normalization function so the the self-similarity starts at some point ($a(0) = \frac{1}{2}$) and gradually grows to one in which case all similarity is the self-similarity (i.e. the similarity matrix is an identity matrix) and no weight is shifted anymore.

$$P^*(i, j) = \begin{cases} (1 - a(t)) \times \frac{W(i, j)}{\sum_{i \in N_j} W(i, j)} & i \in N_j \\ a(t) & i == j \\ 0 & else \end{cases} \quad (5)$$

Additionally this adjustment can help to prevent weights drift to far away and can thereby help preserving some nestedness. It is probably also best that this function that is responsible for reducing the amount of weight that is shifted is convex so that most of the shifted weight happens in the first iteration. I believe this will be best to prevent disrupting the nestedness too much.

3.2.2 Performing network fusion on the connection matrix

Now we only are left with multiple connection matrices for each source of information on the phage-host relationships. We can however transform the connection matrix in a square matrix by adding the missing edges between bacteria and phages as an edge with zero. The resulting matrices can then be fused using the aforementioned fusion algorithm.

3.2.3 A note on the order and combination of steps

As can be seen there are a few different ways to combine all the information. You can for example also apply the network fusion on the connection matrix first and then diffuse it using the similarity matrices. This is also faster because then the diffusion is only applied to one connection matrix and not multiple. I think this might be a worthwhile alternative to investigate.

On the other hand, you can even apply the diffusion using the different sources of the similarity matrices. I, however, don't think that is the best way as it then will start to mix different kinds of information too much to early.

Finally, you can also create the square matrix from the connection matrix by including the similarity weights already instead of setting those edges to zero. I also think that such an approach tries to combine too much information too early. However it might still be interesting to investigate the affect of forcibly setting the similarities between bacteria and phages to zero or to let them grow as well, or even initialize them to some value (fixed or otherwise).

3.2.4 A note on computational complexity

To reduce the computational I propose to make the similarity matrices sparse by only looking at the k-nearest neighbours for each similarity neighbour. This is also done in the SNF algorithm and can also trivially be applied to the aforementioned formulas.

3.2.5 Notes on modular vs. nestedness

The approach relies on the assumption that phage-host networks are modular. I assume it would work best in situations where one wants to place unknown bacteria or phages in a spot with already fairly known relationship.

However, it should also work for relative nested networks, but only if it is true that generalized phages and general bacteria are relatively very similar to a lot of other resp. phages and the specialized phages / less targeted bacteria are more unique.

4 Summary

The discussed method can be summarized in four steps (not necessarily in this order):

* Find similarity matrices for phage-phage and bacterium-bacterium using network fusion. * Find connection matrix for phage-host. * Diffuse the phage-host matrix using the phage-phage and bacterium-bacterium matrix. * Apply network fusion to the resulting diffused phage-host relationships.

5 Further ideas

5.1 Using phage-host information for a better fusion

...

This might however, establish modular network a bit too much.

5.2 Using established phage-host relationships

There is various experimentally proven phage-host relationships available. These experimental data can be used to enhance the network fusion or to validate our method.

Still, it is important to note that those experimentally reported phage-host relationships might be biased and using it needs caution.

5.2.1 Enhancing the proposal with some semi-supervised method

It might be possible to use this as an alternative information source that can make certain relationships more important than others because of it being experimentally proven, or can be used in some semi-supervised learning way.

5.2.2 Validation

This source of information can however also be used a way to validate our method.

It is perhaps also a good idea to see if certain assumptions we make in this proposal or true for the real world. Especially the idea that phages that infect the same bacteria are more similar to each other and bacteria with the same phages as viruses are more similar to each other.

5.3 Deep learning approaches

There might be a way to represent this problem in such a way that there is a way to work on this in a deep learning approach. Because of a lack of knowledge, I find it hard to come up with something related so far, but might still be worth looking into.

References

- [1] Robert A Edwards et al. “Computational approaches to predict bacteriophage–host relationships”. In: *FEMS microbiology reviews* 40.2 (2016), pp. 258–272.
- [2] Bo Wang et al. “Similarity network fusion for aggregating data types on a genomic scale”. In: *Nature methods* 11.3 (2014), p. 333.
- [3] Bo Wang et al. “Unsupervised metric fusion by cross diffusion”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 2997–3004.