The phage proteomic tree: A genome-based taxonomy for phage

Rohwer F, Edwards R (2002). Journal of Bacteriology 184:4529-4535

A tour

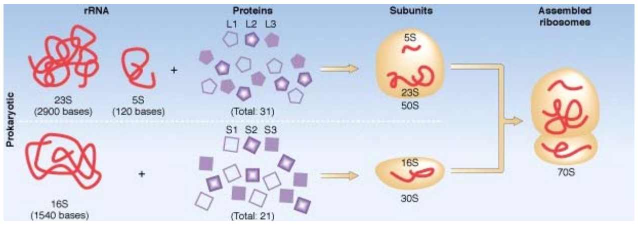
Order. It is so much easier making sense of a mass of information if we have a framework in which to place its pieces. When Theodosius Dobzhansky wrote the famously titled essay “Nothing in biology makes sense except in the light of evolution”,[*] he said nothing about how life was created (whether by chemical reaction, divine fiat, or extraterrestrial intervention) but rather focused on how the process of evolution allows us to make meaningful connections amongst the diverse forms of life that are present today. Soon you will be called upon to make sense of a wealth of bacteriophage DNA sequences. A phylogenetic framework – a family tree – will be of immense help to you.

You would think that if such a framework is so important, it would have been made by now. For cellular organisms – plants, animals, bacteria, and archaea – you’d be right. But in the case of viruses, there is no accepted framework. Indeed, that is a large part of the impetus behind the viral genome project of which your work will be a part. Perhaps traveling into the unknown is exciting, but it’s also a pain in the neck. You’re going to have to explore without a good map and to construct the map as you go.

The article by Rohwer and Edwards offers one method that may help you do so.

Introduction

I’ll leave this section to you, except to comment on the second paragraph, where Rohwer and Edwards draw an analogy between a sought after universal phage gene (which doesn’t exist) and 16S rRNA in cellular organisms. When comparing entities in a class, you want to make sure that the basis of the comparison works for all. It doesn’t help to compare all cars as to the number of their pistons if some cars don’t have pistons. Fortunately, all cellular organisms have ribosomes and the genes to encode their components. The RNA portion of ribosomes (hence the genes that produce them) are highly conserved and can serve as the basis of comparison of all organismal life forms. If there were a similar universal gene in bacteriophage, the article would be very different from what it is.



Assembly of proteins on RNA scaffolds to form ribosomes. The figure shows the components of ribosomes in eubacteria. Those of archaea and eukaryotes are similar but somewhat different. (From Bruce Roe, Oklahoma University, http://www.genome.ou.edu/5853/5853.html)

Materials and Methods

I suggest that, as usual, you skip over this section initially, except to note what sections are in it for later reference.

[*] The American Biology Teacher (1973) 35:125-129. http://people.delphiforums.com/lordorman/light.htm

Results and Discussion

Phage proteome analyses

Rohwer and Edwards hoped to find a magic protein encoded by all phage genomes. If they had, they might be able to use it to construct a universal phage tree. They said they failed to find such a protein, and it sounds like it was a lot of work! But it isn't so much work to reproduce some of what they did. They say the most popular protein they found was a putative transglycosylase (an enzyme that transfers sugars onto proteins), an example of which is YomI from the phage SPBc2. The article was written in 2002. How has the situation changed since then?

SQ1. Identify the gene encoding YomI (using the GENE-DESCRIBED-BY … IN …) function, using SPBc2 as the value of IN. Then find all similar sequences to the protein in all prokaryotic viruses. You can do this using the SEQUENCES-SIMILAR-TO function, using PROTEIN-OF gene as the query and *prokaryotic-viruses* (from the DATA / Organisms menu) as the target. Be sure to specify the PROTEIN-VS-PROTEIN option. Finally, count the number of hits you got and compare that with the total number of prokaryotic viruses. What fraction is it?

SQ2. Where does YomI fall in the graph shown in Fig. 1?

SQ3. Approximately what is the median number of significant hits for the proteins tested?

Compatibility analyses

OK, the universal gene approach doesn't work. What next? Consider the problem of categorizing all the restaurants in a city. You might think you're going to end up with Italian restaurants, Indian restaurants, Chinese restaurants, etc. But how can you make the categorization in an objective manner? Perhaps there's a universal menu item. For example, consider the bread: If it has garlic on it, the restaurant is Italian. If it has sesame seeds and is puffy, it's a hamburger place. And so forth. But Chinese and Korean restaurants don't have bread, so that approach doesn't work. An alternative is to compare pairs of menus and see how many items are in common. You'll find that the Chinese restaurants probably group together, even though there may be no item found on all the menus. That's what Rohwer and Edwards did.

How to count the number of menu items shared by two phage? Rohwer and Edwards tried two methods. The first was to blast each protein of phage 1 against the proteins of phage 2, counting how many proteins found significant hits. What "significant" means is certainly open to debate. They used three different E-values as a threshold. It will be easier if we consider just one of them, E=0.001, as that's the default value used by BioBIKE. We'll consider the second method used by Rohwer and Edwards later on.

It doesn't do any good to predict the relationships amongst phage if you have no way of testing whether the predictions are accurate.

SQ4. How did Rohwer and Edwards test their predictions? What makes them think that the test is valid?

We'll test two of the predictions: "*ssDNA phage Pf3 is only distantly related to M13, f1, and fd*" and "*Fs-2 is similar to f1, fd,M13, Ike, and Pf3*". All of these phages are unusual in that they package single-stranded DNA into their phage heads. Virobike doesn't have phage fd integrated into the system, so we'll consider only the other five.

SQ5. Define a family (call it M13-family) consisting of the five ssDNA phages listed above (but not fd). You can readily make the list by enclosing the names of the phages in parenthesis: (M13 iF1 Pf3 Ike Fs2). Note that ViroBIKE calls the phage f1 iF1 (i standing for Inovirus, the phage's family) and the last phage Fs2, not Fs-2.

SQ6. Find the sequences similar to each of the proteins of Pf3 in M13. Do this in a way whose utility you will appreciate in a moment: Define phage1 as Pf3 and phage2 as M13. Construct the function SEQUENCES-SIMILAR-TO PROTEINS-OF phage1 IN phage2 PROTEIN-VS-PROTEIN. First execute the PROTEINS-OF box. How many proteins are there in phage1? (note that they are numbered sequentially). Then execute the entire function. How do you interpret the results?

The function takes each protein of phage1 in turn and asks if there are any similar proteins in phage2. You should get an answer for each of the phage1 proteins. Most of the proteins have no matches, so NIL is returned. You can readily see how many proteins have matches, just by ignoring the NILs and counting the rest, but this process is readily automated.

SQ7. Remove the NILs from the result by the SUBTRACT-SET function, which you'll find in the LIST-TABLES / LIST-PRODUCTION menu. Drag the SEQUENCE-SIMILAR-TO function you created into the *set* box of SUBTRACT-SET, it is the set produced by that function that you want to modify. From the Options menu, select BY and enter NIL. Execute this function to subtract all NILs from the result.

The one result that remains is plain to see, but again, it will be helpful to automate the counting.

SQ8. Count the number of matches by bringing down the COUNT-OF function. Drag the now large SEQUENCE-SIMILAR-TO function into the *query* box. Execute this function to count the number of matches you got from the protein comparison.

Now repeat this for all possible pairs of phage1 and phage2.

SQ9. How many pairs are there? Do you want to site there defining all of these combinations (DEFINE phage1 = Pf3, then = M13, then …, and do the same with phage2…!!!)?

No, you do not! You want to let the computer do repetitive tasks while you kick back and think thoughts only humans can think.

SQ10. How can you convince the computer to APPLY the function five times, once for each member of the M13-family in the phage2 slot? (leave phage 1 as Pf3 for now). Execute the function.

Now you have a list of numbers, representing the number of matches between phage1 (currently Pf3) and each of the other members of the M13 family.

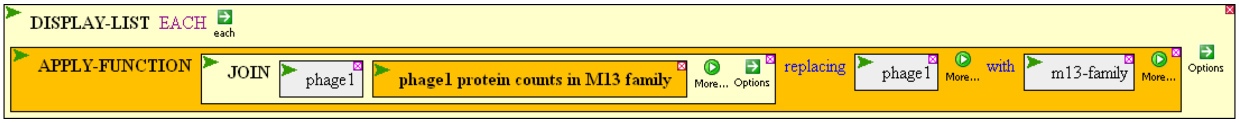
SQ11. Which number is the largest? Why is it so much larger than the rest?

Finally, you can automate the replacement of `phage1` with members of the M13 family. Before you do this, to create more room (both on the screen and in your head), collapse the APPLY-FUNCTION by mousing over the action icon and clicking Collapse. Then mouse over the action icon again, click Name Me, and give it a descriptive name (something like: phage1 protein counts in M13 family).

SQ12. How can you APPLY this complicated APPLY-FUNCTION so that phage1 is replaced by each member of the M13 family? Execute the function.

SQ13. Display the table (for that is what you have created) by surrounding the outer APPLY-FUNCTION by DISPLAY-LIST EACH. Execute the function.

That is a nice table, but it is difficult to interpret because there are no labels. To add labels, you just need to add the name of phage 1 to the beginning of each row. It's complicated to explain… here's how it should look like in the end:



DISPLAY-LIST EACH each
APPLY-FUNCTION JOIN phage1 **phage1 protein counts in M13 family** More... Options replacing phage1 More... with m13-family More... Options

SQ14. What are the relationships amongst the five phage of this family? Who is most closely related to whom?

SQ15. Use the information to draw a phylogenetic tree of these phage, where the length of the lines between each pair of phages represents the degree of difference between them.

SQ16. Now that we've done what seems reasonable, let's look at the instructions. What did Rohwer and Edwards actually do? (You'll finally want to pay a visit to the Methods section)

(More to come!)