Exploring Bioinformatics: A Project-Based Approach

Key for Web Exploration Questions, Chapter 8

1. Looking at the DNA sequence traces, what conditions appear to cause the base-calling program to output N rather than designating a specific base?

   Overlapping fluorescence peaks and very low, broad peaks (both of which are conditions in which it's hard to unambiguously identify a peak) tend to lead to an N being called.

2. How many nucleotides of sequence was the base-calling program able to read for the traces you examined?

   Even though the sequence quality clearly degenerates beyond a few hundred nucleotides, the base-calling program used here was able to generate 1000-1500 nucleotides of sequence for each trace.

3. Why do you think the lowest quality sequence occurs at the beginning and the end of the sequence run?

   At the beginning of the sequence, the fragments being generated by DNA polymerase are very short—very close in size to the primer. Even a momentary mis-pairing of the primer can result in addition of one or two nucleotides here, and there could even be some primers to which a nucleotide was added by a polymerase that wasn't reading a template. Therefore, generally these very short fragments generate messy, unreliable sequence.

   Accurate reading of sequence far from the primer depends on the ability of DNA polymerase to traverse a long stretch of template and then terminate with a dideoxy nucleotide. Given a certain probability of using a dideoxy at each base, the chance of not terminating decreases as the length of the fragment increases, so there are fewer long fragments and thus less fluorescence intensity. Additionally, the DNA polymerase may simply fall off the DNA in a long stretch, leaving an unusable fragment and again reducing the number of fluorescent fragments of a given length.

4. Although each dideoxy sequencing run produces a sequence trace, in a large metagenomic or genome sequencing project, it would not be practical to examine each trace and manually assign difficult bases. How can the sequences returned by an automated base-caller be used reliably in such a project?

   Coverage takes the place of manual reading here. Given that we expect to have overlapping fragments, we can hope that a nucleotide that was difficult to call accurately in one run (maybe because it was near the beginning or end of the usable sequence) can be called correctly from other runs. When the sequences are assembled, one bad call would be eliminated if there are several calls for that nucleotide that agree.

5. How does the number of viral sequences found in the sequence runs you analyzed compare to the number of bacterial sequences? Are there fungal sequences? Protists? Do these relative numbers make sense in terms of the human gut environment and the roles of these organisms?

   The metagenomic analysis should find far more bacterial sequences than viral sequences, due to the huge numbers of bacteria that inhabit the human gut. In the analysis run to produce this key, 37 viral sequences were found, compared to some 400,000 bacterial sequences (Superkingdom = Bacteria).

   There were 3994 fungal sequences (Kingdom = Fungi) in this run, including *Saccharomyces cerevisiae* (brewer's and baker's yeast) and the common throat and vaginal pathogen *Candida albicans*. Finding protists is more difficult, because the protists are so different that they have been divided into multiple kingdoms. However, searching for specific protest groups suggests that protist sequences are indeed represented in the gut metagenome.

6. Some of the species represented among the gut sequences might seem surprising. What seemingly unlikely species were identified, and what are some possible reasons for these results?

   One might initially be surprised by some of the sequences identified, including many animal sequences ranging from cows to chimps, as well as plant sequences. Some of these are probably the result of strong conservation among the mammals and particularly the primates: a human gene sequence might well match the chimp genome so closely that the difference was not detected in the Megablast alignment. But, some of these sequences probably come from food!

7. What are the most commonly found viral sequences? Why do you think this is the case?

   Most of the viral sequences returned in this analysis came from bacteriophage: viruses whose hosts are bacteria. This makes sense because (1) there an enormous number of potential bacteriophage hosts in the human gut, and (2) viruses that can infect the cells of the human intestine would in most cases be causing some damage and thus likely subject to control by the immune system.

8. How could viruses that are normal residents of the gut community be distinguished from those that might be pathogens?

   One approach would be to repeat this metagenomic analysis for a large number of individuals. Viruses that appeared in the majority of the analyses would likely be normal residents, while viruses that appeared in only a few individuals are likely pathogens.

9. How could novel viruses be distinguished from related viruses that have already been characterized?

   Once sequences that come from viruses have been identified, these specific sequences could be selected and used in a BLAST search to see if they closely match sequenced viruses. Even those that do not can likely be placed in families by this method, and sequencing of known but unsequenced viruses from a particular group could be used

to further narrow down whether the viruses from the metagenome analysis are genuinely novel.

10. How many of the sequence reads were rejected in the sequence cleaning process? Can you determine why they were rejected?

    Only a small number of reads are rejected—two, in the analysis run to make this key. Both were very short sequences with a large number of N's.

11. Use BLAST to compare your contig sequence to known sequences in GenBank. The assembled sequence should match one known sequence with a high degree of similarity. What have we sequenced? How long is its genome?

    The sequence is clearly that of human klassevirus, with a 7988-bp genome.

12. Since next-generation sequencing produces random short reads, there is no guarantee that even 2500 reads would be sufficient to completely sequence a particular genome. Did the sequence reads you assembled cover the entire genome, or do gaps remain? In order to fill any gaps, would it make sense to simply run more sequencing reactions, or are there other approaches that should be considered?

    In this analysis, two contigs were generated by the assembler: a 7982-nt contig that covers all of the klassevirus genome except six nucleotides at the end and a 209-nt contig that is entirely contained within the larger contig but due to a number of sequencing errors (visible as mis-matches to the klassevirus genome) cannot be successfully merged with it by the algorithm used.

    To fill the gaps at the end, the simplest approach would be to design a primer that would bind the sequence perhaps 100 nt or so before the end and re-sequence with this specific primer to get those last nucleotides.

    Additional coverage might have made it possible to merge the short and long contigs.

13. Looking at the contig alignment file in the EGassembler results, you should be able to see that there are hundreds if not thousands of small sequencing errors among the sequence reads. Was the assembler able to generate a correct contig sequence (as compared to the known sequence in the database) despite these errors? Explain how the sequence errors were accurately corrected. Were all errors caught, or did some remain in the final contig sequence?

    Despite some repeated regions in the klassevirus genome, the assembler was able to generate a near-perfect contig, with only one error in 7979 nucleotides. The errors were corrected by having sufficient coverage to resolve differences. However, the second contig is spurious; enough errors here couldn't be corrected that this sequence looked to the assembler as if it were distinct from the long contig.

14. You used the default parameters for the CAP3 assembler in your EGassembler run. In a real sequencing project, however, you might want to change variables such as the overlap percent identity cutoff (the minimum percentage of nucleotides that must be identical in the overlapping region of two fragments). By default, CAP3 is quite tolerant of sequencing errors (and in fact automatically compensates for some of the common problems of high-throuput sequencing, such as low-quality sequence at the beginning and end of fragments). To see how these parameters affect the assembly,

try setting the overlap percent identity cutoff to 100%. What happens to your contig? Does the quality of your alignment change? (You can choose Step-by-Step Assembly at the top of the page to access more parameters.)

For this particular set of reads, the contigs did not change, suggesting that the degree of accuracy in the overlaps was very high. This parameter would probably have made more difference had the error rate been higher.