Genome assembly validation

Introduction

A common thread in scientific computation is the fact that often the scientific answer that we seek is simply defined by the output of the program used to compute the answer. Given the fact that the results often represent previously unknown information, there is no external way to validate these results. In other words, any bugs in the software are virtually undetectable.

There are two recent examples that provide a compelling demonstration of this fact:

- A paper in the journal Science had to be retracted once the authors realized that the results reported in the paper had been incorrectly computed due to a "+1/-1" error in their code. The mistake was only detected a few years later when another research group failed to reproduce the results (in this case the scientific result was the 3D structure of a previously uncharacterized protein)
- When the records of climate researchers came under scrutiny due to an email leak, scientists and skeptics found out that the software that computes the climate models, and has predicted the effects of global warming, is full of bugs and poorly written code.

Thus, a major challenge in bioinformatics and other scientific fields is to develop approaches for ensuring that the software is correct even in the absence of external means for validation. Below I describe several ways in which validation can be achieved in the context of genome assembly.

Intrinsic measures of assembly quality

As mentioned in the assembly lecture, one of the early definitions of the goal of assembly is to determine a layout of the sequences that is consistent with the random process that generated these sequences in the first place. While this goal is difficult to achieve during the assembly process, the divergence between the output of the assembler and the characteristics of the random shotgun process can be used to assess whether the assembly is correct.

Note that, in general, we do not fully know the characteristics of the sequencing process, however useful information can still be achieved by assuming the process to be fully random (in practice many biases cause deviations from randomness).

Coverage measures

If we assume that the shotgun process is truly random, we can expect the depth of coverage (# of reads spanning a particular location in the genome) to not vary much along the genome (other than random fluctuations from the expected value). Regions where too many reads pile up can be inferred to represent collapsed repeats.

There are several ways in which overly-deep regions can be determined. A simple k-mer based measure can be determined as follows:

- for each k-mer count its number of occurrences in the set of reads. Let this value be $n_r(k)$. In unique regions this value should roughly equal the coverage c, while in repeats with R copies, the value will be approximately cR
- for each k-mer count its number of occurrences in the consensus of the contigs produced by the assembler. Let this value be $n_A(k)$. In unique regions this value will be 1, while in repeats it will be R.
- for each k-mer compute $n_r(k)/n_A(k)$ - if the assembly is correct (number of copies of a repeat in the genome matches the number of copies in the assembled contigs) the ratio will be roughly c. If, however, certain repeats are collapsed, the ratio will be correspondingly higher - allowing us to detect regions where a collapse might have occurred.

A more principled solution is to statistically evaluate whether the pile-up of reads is truly "surprising" given our expectation of the random process generating the sequences. Such an approach is implemented as the A-statistic used in Celera Assembler. Specifically, assume that the expected arrival rate (number of DNA sequences divided by length of genome) is α, then the probability that k sequences start every ρ base-pairs within a unique region is:

$$p(\rho, k) = \frac{(\rho\alpha)^k}{k!} e^{-\rho\alpha}$$

while in a two-copy collapsed repeat we have:

$$p_2(\rho, k) = \frac{(2\rho\alpha)^k}{k!} e^{-2\rho\alpha}$$

The log ratio of these two values represents the A-statistic:

$$Astat = \log_2\left(\frac{p(\rho, k)}{p_2(\rho, k)}\right) = \log_2\left(\frac{\frac{(\rho\alpha)^k}{k!} e^{-\rho\alpha}}{\frac{(2\rho\alpha)^k}{k!} e^{-2\rho\alpha}}\right) = \log_2(e)\rho\alpha - k$$

Negative values indicate that more "arrivals" occur within a region than expected, i.e. the region represents a collapsed repeat.

It is important to note that the background arrival rate α needs to be estimated from the assembly data, as generally the genome size is not known before running the assembly. This creates somewhat of a chicken and egg problem - information obtained from the assembly is used to evaluate the assembly. If errors are rare, this process should, however, allow the detection of outliers (misassemblies), as the misassemblies cannot significantly affect the statistics.

Mate-pair consistency

In addition to sequences, a typical sequencing experiment also provides mate-pair information - information specifying that pairs of sequences are separated by a given (approximate) distance,

and that they have a specific orientation (usually the two sequences are in opposite orientations - originate from different DNA strands). In a correct assembly, the constraints imposed by this information should be preserved (modulo experimental error), i.e. the paired sequences must be oriented and spaced as inferred from the parameters of the sequencing project. Any deviations from this ideal indicate the presence of a potential misassembly.

Several approaches can be used to detect disagreements between the experimental data and the information found in the assembly. First, one can easily find groups of mate-pairs that are "broken" because their corresponding endpoints are incorrectly oriented with respect to each other. These indicate the presence of rearrangements or inversions in the assembly with respect to the correct structure of the genome.

In order to identify regions in the assembly where the distance between mates is different from the experimentally derived values, we can phrase the problem as a statistical hypothesis test:

- First compute the "correct" mate-pair spacing by averaging the lengths of mate-pairs found within the assembly. As described under coverage statistics, this approach may be confused by misassemblies but should generally work if there are few misassemblies. Also, it is usually necessary to "second-guess" the biologists that generated the data given that the methods for measuring mate-pair sizes are highly inaccurate. The result of this step are the values L_{true}, and SD_{true}, the mean length and standard deviation of the "correct" mate-pairs.
- Second, use a sliding window through the assembly to identify the mate-pairs spanning each particular location in the assembly. Compute $L_{assembly}$ and $SD_{assembly}$ from the local mate-pair neighborhood
- Compare L_{true} and $L_{assembly}$ using a statistical test, e.g. the t-test: $\frac{|L_{assembly} - L_{true}|}{SD_{true}}$

The t-test provides an estimate of the statistical confidence in the disagreement between the two length estimates. The sign of the difference in length indicates whether the assembled region has been collapsed or expanded with respect to the correct assembly of the genome. This simple test is also called the C/E statistic (Zimin et al.).

Detecting "missing" data

It is not uncommon for an assembler to only use part of the data provided as input, i.e. not all the sequence reads end up being used in the final assembly. Valuable validation information can be obtained by mapping the unassembled sequences onto the assembly, in order to explain why they were not used by the assembler. Several cases are possible:

1. Unassembled reads to not match the assembly - this indicates the reads are likely contaminant sequences.
2. Unassembled reads match the assembly perfectly - possibly bug in the assembler but that does not affect the quality of the reconstruction (though it can affect coverage estimates)

3. Unassembled reads match the assembly imperfectly, as shown in the figure below - this indicates that a tandem (2-copy) repeat has been collapsed into a single copy. Sequences spanning the joint between the two copies cannot fit in the resulting assembly and get tossed out.

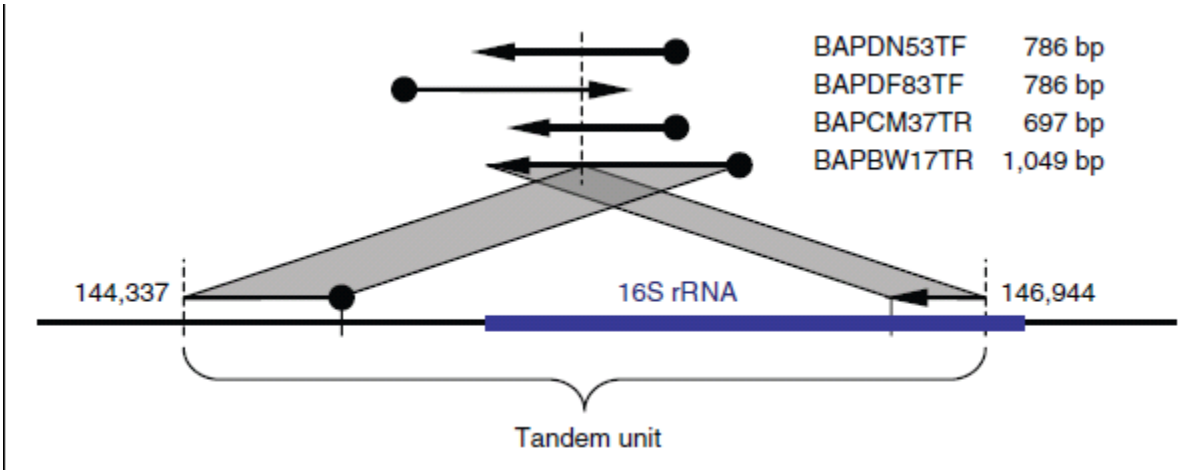

Illustration 1: Alignment of reads to the assembly in a region where a tandem repeat has been collapsed.

"Validation" as a tool for detecting genomic variation

The tools for assembly validation described above are not only valuable for assessing the quality of a genome assembly. Rather, there is increasing interest in understanding how the genome structure varies within a population. E.g. in the human population there is increasing evidence that the genomes of different people, and even within a same person, differ by more than just simple mutations: entire segments of DNA are duplicated, deleted, or rearranged between the genomes. Also, certain phenotypic traits (e.g. hair/skin color) might be caused by differing copy numbers for certain genes.

In the context of genomic variation, we want to map sequences/mate-pairs generated from a person's DNA, to a reference sequence. Discrepancies between the two datasets indicate places where there are structural differences between the two genomes.