UNIVERSITY OF CAMBRIDGE

Solving the genome puzzle

The human genome is represented by a string of around 3 billion letters. To deal with such large numbers, genome sequencing relies on clever mathematics.

Solving the genome puzzle

Genomics is an area where size really does matter. The human genome is represented by around 3×10^9 bases — that's a sequence of A, C, G, and Ts around 3,000,000,000 letters long. With such large numbers, determining the sequence of the entire genome of a complex organism isn't just a challenge in biochemistry. It's a logistical nightmare. Historically, some of the largest non-military supercomputers were built just for the purpose of solving it.

Get the gun

One problem is that sequencing technology can only sequence DNA strands up to 1,000 bases long. Scientists use something called *shotgun sequencing* to get around this problem. Essentially this involves taking many identical copies of a DNA strand, breaking each up randomly into small pieces and then sequencing each piece separately. You end up with lots of overlapping short sequences, called *reads*. The overlaps between them should then give you enough information to assemble the sequence of the whole strand.

This sounds good in theory but there is a problem. When sequencing a new genome as complex as that of humans you might be looking at up to 50 million reads — comparing every read to every other read to see if they overlap gives you around $50{,}000{,}000^2$ comparisons. If each comparison involves around 10,000 steps and one can calculate 10^9 steps per second, it will take around 300 years to carry out the comparisons! Moreover, sequencing technology is prone to error, missing out or inserting characters, or getting them wrong.

This is where mathematicians and computer scientists come in: they can develop algorithms that are cleverer than brute force.

Bubbles and tips

One approach is to decompose each read into overlapping strings of length k (k=4 in our example below). Then construct a giant graph in which each node corresponds to a string of length k and also keeps a record of how many times that string has been seen in reads. From a given node there are arrows to all nodes that represent overlapping strings and that have been observed to follow it in one of the reads.

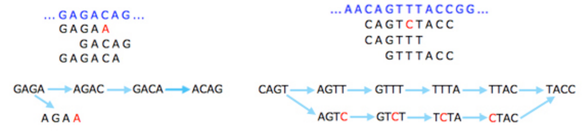
CGTACAT

CGTA → GTAC → TACA → ACAT

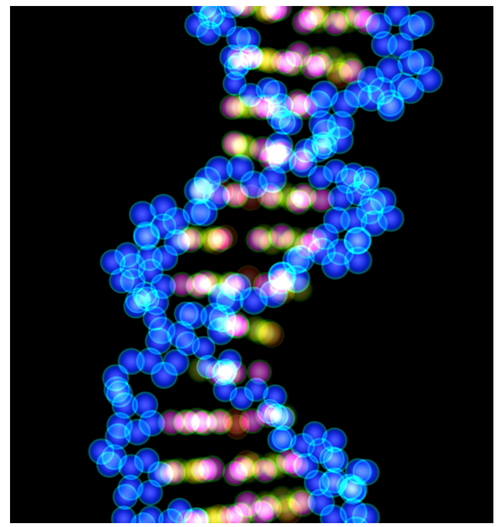
Assembling the original sequence corresponds to finding a path through your graph which traverses every arrow once. If you are lucky, there is only one such path, which then gives you the whole sequence.

For large genomes, though, errors and repeated strings of characters mean that you'll never be that lucky. But you can make clever use of the structure of the graph to weed out errors and identify repeats.

The basic idea is that a sequencing error (shown below in red) at the end of a read gives rise to a "tip" in the graph: that's a short dead end where the graph doesn't continue. This is because the exact same error will only occur in very few reads, so you'll run out of k-strings to continue the tip. Errors that occur within reads will give rise to "bubbles", alternative routes between two nodes.



When you look at your graph you can spot which tips and bubbles are likely to come from errors by keeping track of coverage: since the same error is only going to appear in very few reads, the nodes containing the error will also correspond to very few reads. So you go through your graph, pruning off tips with low coverage and flattening bubbles by removing whichever route has lowest coverage.



After this you can merge nodes where there's no ambiguity. Any loops in the graph now indicate repeated sequences. If it's not clear from the graph how to resolve these repeats, you need to make use of any extra information you might have.

Algorithms like this one are now being used for large genomes and they speed things up considerably: sequencing work that previously took days and thousands of computers working simultaneously can now be done in a day on just one computer with a sufficiently large memory for the graph (this can be 1 Terabyte of RAM or more!).

It is possible that future technologies will be able to sequence longer and longer strands of DNA. But until they can sequence a whole genome in one go, lots of computing power and clever algorithms will be needed to piece the pieces together, and mathematics will remain at the heart of genomics.

This poster is based on the article "Solving the genome puzzle", based on an interview with Dr Gos Micklem, Director of the Computational Biology Institute at the University of Cambridge. The article is published in Plus (plus.maths.org), a free online magazine opening a door to the fascinating world of mathematics. Plus is part of the University of Cambridge's Millennium Mathematics Project.