

Bench philosophy (22): Next generation sequencing software

Next Generation Data Explosion

The Sanger method of sequencing caused a revolution in every corner of biology. With the recent emergence of Next Generation Sequencing, it is happening all over again.

It did not take long after the completion of the Human Genome Project, for sequencing to become steadily faster and for key elements in the pipeline to be handed over to automation, leading to genome after genome being sequenced. There is probably not a single biology lab in the world that hasn't had its outlook completely changed as a result.

Now, a new generation of sequencing is on the rise. It is an order of magnitude faster, cheap enough for the small lab and has deep enough coverage to look at genomes of individual animals or even tissues. These Next Generation Sequencing (NGS) approaches promise a new level of power. But there is a glitch: the potential of NGS hits a major challenge when it comes to data storage and analysis.

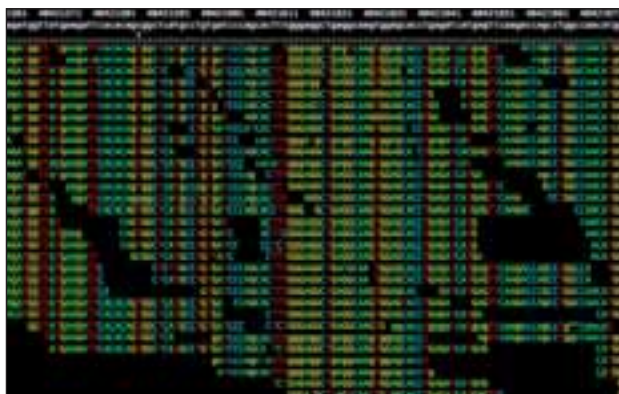
What has brought about this upshift? A major factor was the development of new sequencing methods, which departed from the established capillary-based Sanger methods. There are three major platforms that enshrine the NGS methodologies. The first to hit the scene, in 2004, was the Roche454 system. Roche454 abandoned the stop-based Sanger methods in favour of the synthesis-based pyrosequencing method, which exploits the release of phosphate upon base incorporation. Two years later, the Illumina platform, which also uses sequencing by synthesis, was released. In the meantime BioSystems, which played such a major role in the Sanger days, didn't want to be left behind. Applied Biosystems' "SOLiD", which uses sequencing by oligo ligation and detection, is one of the newest mainstream NGS platforms.

The race is on

A key element shared by all the NGS platforms that has allowed the massive up-scaling of sequencing speed has been the deployment of these new techniques in massively parallel sequencing formats. As a result, all three platforms comfortably sequence about a million DNA fragments in

parallel. And it doesn't stop there: an additional platform has recently been released – the HeliScope system (<http://www.helicobio.com>) – that, unlike the other three methods, does not even require the template to be amplified before sequencing, thereby reducing the risk of introducing errors. NGS is a fast developing world.

So, where has NGS already begun to make an impact? The most obvious application of the new technologies is to rush ahead and sequence yet more genomes. This is the reason why so many bacterial genomes are being announced. NGS has



It takes a lot of software power to assemble a whole sequence from tens of millions of short sequencing fragments.

major technical challenges when it comes to *de novo* sequencing of eukaryote genomes but even these have attracted the attention of our brightest bioinformaticians and there are signs that this barrier will soon be overcome. But it's not just all about doing the same old thing quicker and stuffing more and more genomes into the databases. NGS is not only changing how quickly we do things, it is also changing the way we do things and the kinds of questions we can ask.

The two outstanding features of NGS, which are speed and depth of coverage, have already spawned a number of quite diverse biological applications. For example, NGS was soon adapted to replace or complement existing methodologies such as microarray analysis. NGS has higher resolution than microarray analysis and does

not require any prior genomic knowledge for the design of the chip. On top of that, NGS is much more powerful than microarrays in detecting splice variants, editing and allele-specific variations. The list of possible applications, in which NGS could provide a more accurate, quicker and possibly even cheaper solution, would fill pages.

It is not only the rate of sequencing that is so exciting. Indeed, the very way in which NGS does its sequencing opens the door to a host of scientific and medical applications. The main point here must not be missed: first generation methods used simultaneous sequencing of many templates amplified from an original clone, obscuring individual differences between transcripts. But because NGS methods amplify individual DNA sequences, these methods reveal individual sequence variations, such as the Single Nucleotide Variations (SNVs), not just a consensus sequence. Hence, new technologies, lots more data and endless opportunities limited only by our imagination. Good news for all but there are problems. Not the least of which is, how to store and analyse the data. An NGS sequencing lab can easily generate a terabyte of data a day.

We have become used to cheap terabyte drives at consumable-budget cost, so, in theory, you probably could try just stacking external terabyte drives on shelves. And after all, as the cost of sequencing continues to fall, there comes a point when the cost of storing sequences becomes greater than the cost of extracting the sequence in the first place. This has led more than one person to comment that we could store the sequence in the DNA!

But for the major database repositories, such as NCBI and EBI, this flippant solution will not do, and they are looking closely at how they should store readouts from NGS projects and provide the scientific community with the software it needs if this information is to be put to the best use (http://nar.oxfordjournals.org/cgi/content/full/38/suppl_1/D17). Clearly, if the first genomics

revolution was a storage challenge, the NGS revolution will be even more so.

The problem of data storage, however, pales before the even greater challenge of data analysis. The nature of NGS data mean there is an urgent need for the development of software to turn raw data into understanding. A main part of the problem is due to NGS methods differing from first generation sequencing, whereby they amplify individual DNAs, many of the algorithms used for first generation sequencing cannot simply be used “out of the box” for NGS.

Urgent need for software

Consequently, new software is needed at all levels of the NGS pipeline. Let's start at the very beginning: all NGS projects begin with the generation of a large number of short sequences, which are then either assembled into a new genome or matched against a reference genome to establish their identity. Given the size of the reference genome (or genomes if you are tackling a comparative genomics problem) and the huge number of template sequences, this is a major computational challenge.

This is largely being addressed by adapting established bioinformatic algorithms using a variety of “hacks”, or heuristics, that make searches faster and less memory consuming. The first problem is to pack the genome and the sequences into a limited amount of computer memory. The indexed tables are compressed, so that the latest analysis programmes such as SOAP2 (<http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btp336>) and Bowtie (<http://genomebiology.com/2009/10/3/R25>), both freely available, manage to fit the human genome into about 1.3GB of memory.

But compression is a comparatively trivial problem. Other issues are a bit trickier. Take, for instance, the important first step of identifying your sequences. NGS produces short sequences which contain the genuine variations of biological interest but these are interspersed with reading errors. The combination of short length of sequence (“short reads”) and the presence of errors and variations is why established alignment methods, such as BLAST, are just not suitable. This is one of the major challenges to NGS data analysis and is a hot topic in the bioinformatics community.

So just how are NGS data analysis algorithms going about solving this problem? The most common method being used by most short-read mapping tools is to assem-

ble a hash table of all possible k-mers of short sequences of specified length. Each sequence is divided into a small number of substrings. We assume that there is a small number of mismatches. If that is so, then at least one of these substrings will be error- or SNV-free, in which case a perfect match for that substring can be found. Any such match is then extended outwards to the full length of the sequence. This is the approach taken by programmes such as BLAT (<http://www.ncbi.nlm.nih.gov/pubmed/11932250?dopt=Abstract>), which is sufficiently memory and CPU-efficient to allow the operation to be performed on ordinary desktop computers.

Another particularly successful algorithm to make short sequence mapping efficient is the “spaced seed” approach, originally developed for the “PatternHunter” alignment algorithm, in which the genome is searched using a template of 1's (look for matches here) and 0's (don't care about mismatch here). Zoom (<http://www.bioinformaticsolutions.com/products/zoom/index.php>) is one such commercial package that deploys this algorithm. However, the bioinformatics world is also full of exciting new programmes that are free to use. For instance, PerM (<http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btp486>) uses periodically spaced seeds combined with parallel processing to speed mapping, while its method of representing the genome index allows storage of a whole genome in computer memory.

The wide range of possible hacks to improve the performance of data analysis algorithms is the reason for so many new algorithms being published every year. They are wrapped up in the many commercial and free software packages available, all of which differ in the strategies they use to index the genome or sequence data in order to maximise the speed of matching and the memory use, as well as the way in which they deal with mismatches.

Once you have got your mapping, visualising the datasets generated by NGS is another major challenge. Any data visualisation software, fit for working with massive datasets, needs to be able not only to manipulate those datasets but also provide the user with the ability to have a broad view, whilst being able to focus on areas of interest. Again, while viewers for Sanger-type data are available, they do not work effectively with NGS data. Happily, a range of free software is available. EagleView (<http://genome.cshlp.org/content/18/9/1538.full>), available for non-

profit use, offers views of individual traces (the readouts of different platforms, each different from the other) along with genome annotations and other views of the data. It is suitable both for SNP validation and de novo genome assembly.

But if you find that matching a large number of short sequences, whose quality is not perfect, to a reference genome is not straightforward, the problem for assembling a genome de novo from the short sequences generated by NGS is even more acute. We have already mentioned how these difficulties have so far prevented the application of NGS to de novo sequencing of eukaryotes. However, here again, new developments in data analysis algorithms are likely to bridge the gap. For instance, one group recently showed that up to 65% of the genome of one human individual could be assembled using the ABYSS (assembly by short sequences) algorithm (<http://www.ncbi.nlm.nih.gov/pubmed/19251739?dopt=Abstract&holding=np>).

Separate pipelines

Differences in the methodologies used between NGS platforms result in differences in the way data is represented. This is not just a formatting problem – it is ingrained in the way sequences are measured. The Illumina method is sufficiently close to capillary-based sequencers, allowing conversion of its raw readout (“trace”) to be converted easily into a sequence of bases. But because the Roche454 inserts each type of nucleotide in turn, there is inherent uncertainty about sequences of repeated bases. The situation with the SOLiD system is even more complicated: the redundant way in which it represents nucleotides means that the trace can be translated into a base sequence if some of the sequence is known but information distinguishing sequence error from sequence variation gets lost. Although these differences between platforms will not often present a difficulty to an individual project, it does mean that separate pipelines have to be built for each platform and it is a hurdle to developing software that can cope with, and combine, data from different platforms.

STEVEN D. BUCKINGHAM

Fancy composing an installment of “Bench Philosophy”?

Contact Lab Times
E-mail: editors@lab-times.org