Data analysis of next generation sequencing metagenomics studies - parallel computing approaches in genome assembly algorithms

Milko Krachunov², Ivan Popov¹, Peter Petrov², Valeria Simeonova², Maria Nisheva², Elena Todorovska¹, Dimitar Vassilev¹*

¹ Faculty of Mathematics and Informatics, Sofia University “St.Kliment Ohridski”
² Bioinformatics group, AgroBioInstitute, Sofia

*Correspondence to: jim6329@gmail.com

Abstract

The new parallel sequencing technologies produce gigabases of genome information in just a few days bring with them new problems for storage and processing – necessary to be overcome. Sequencing technologies have applications in genome sequencing, metagenomics, epigenetics, discovery of non-coding RNAs and protein binding sites. The abundance of different options for managing and dealing with the data could be found confusing: but there are two major problems in NGS data processing: algorithms for alignment of sequences (for which exists a reference sequence) and algorithms for de novo assembly problems (for no reference sequence is available). Different factors define the choice of better algorithmic solution: cost, reads length, data volume, rate of data generation). As a result the particular bioinformatics solution depends on the biological application and on the type of sequencing technology used to generate the data. All the technologies have their strengths and weaknesses and limits of their performance for providing error free sequenced data.

Most sequenced microbial species are presented as data of 16S ribosomal RNA, which is highly conserved between different bacterial species. The assessment of variations in their sequences can provide information for the biological diversity of bacteria in metagenomic studies. This can lead towards a more accurate and faster alternative to the methods used for bacterial identification (such as phenotypic identification). The reliability of 16S RNA sequencing depends on distinguishing such variations from possible errors introduced by the sequencing method itself. As such errors may bias the phylogenetic analysis of the data, it is important to filter them as much as possible. On the other hand, overzealous error correction may lead to a certain loss of valuable information concerning more rare variants in the sample, again introducing bias. The data used in this work consists of two separate runs and is generated by 454 Roche sequencing platform which has an estimated error rate of 0.25%. Our purpose is to tackle the problem of error correction and estimate the amount of change we can introduce to the raw output data without losing too much biodiversity information.
Introduction

The advent of short-read sequencing machines gave rise to a new generation of IT application in molecular biology studies. The algorithms and software developed for analyzing of short-read data is focused mainly on the error/mutation detection and assembly of the already “clean” data. The constant emerging of new and improved algorithms for de novo whole-genome assembly from next-generation sequencing data explicitly emphasizes the unambiguous impact of algorithmic techniques and their software solutions for practical implementation.

Metagenomics and the de novo assembly of short sequence reads

Most microbial species can not be cultured and even within species can be observed a huge variations in genotype (and consequently in phenotype) owing to genetic plasticity. Therefore sampling signature genes or the basic data semantics such as 16S ribosomal RNA does not give much inside into metabolic activity of a microbial community. This problem has been recently addressed by emerging field of metagenomics. Metagenomics is a “brute force approach” whereby total DNA from microbial and/or viral population is sequenced and compared with previously sequenced genes. The high-troughput capability offered by Next Generation Sequencing methods makes them attractive for such approach.

Until now, we have considered applications that rely on mapping next-generation sequence data to available reference genome sequences. However, at least for smaller bacterial genomes, even the shortest reads can be used to effectively assemble genome sequences de novo, and even where complete closure of the genome is not possible, large contigs an be reliably constructed from such data provided that repeated sequences are not overly abundant. It should be noted that the continued in crease in length of reads obtained by NGS platforms suggest that in the near future, ab initio sequencing of some eukaryotic genomes with technologies such as Illumina or ABISOLiD is likely to be comearealistic prospect, while near-complete drafts of many microbial genomes can now be produced using the 454 technology[13,40,41].

The de novo assembly of sequence reads is not always necessary when comparing closely related stratin: cataloguing polymorphisms relative to a reference genome is often a satisfactory goal. The main goal of both de novo sequencing and re-sequencing projects is to generally identify SNPs (single nucleotide polymorphisms) and other type of polymorphisms such as short insertions or deletions (collectively called indels). SNP discovery is essential for genetic mapping in eukaryotic organism that have large genomes. On the other hand the detection of SNPs and rare variants in microbial genomes communities give us crucial information about the biodiversity of the prokaryote genomes and maximizes the chances of finding isolate-specific genetic differences related to important social diseases.

SNP detection and de novo assembly of microbial genomes are very sensible of detection of errors in the short reads obtained for the NGS technologies. The level of the errors can have substantial impact on subsequent SNPs determination and the assembly of the genomes of the studied microbial communities.

Next Generation Sequencing Technologies and Data Production

The development of these new massively parallel sequencing technologies has sprung from recent advances in the field of nanotechnology, from the availability of optical instruments capable of reliably detecting and differentiating millions of sources of light or fluorescence on the
surface of a small glass slide and from the ingenious application of classic molecular biology principles to the sequencing problem. Another important consideration is that, in the context of an already available genome sequence, many problems such as the identification of single nucleotide polymorphisms (SNPs) need not require the generation of ever longer sequence reads, because most possible words of length $>25$ or $30$ only occur at most once even in relatively large genomes allowing, for the most part, unambiguous assignment of even the shortest reads to a locus of origin in a reference genome. Thus, available NGS technologies produce large numbers of short sequence reads and are typically used in re-sequencing applications, implying the availability of a reference sequence identical, or highly similar, to the source of the genetic material under consideration.

Currently available next-generation sequencers rely on a variety of different chemistries to generate data and produce reads of differing lengths, but all are massively parallel in nature and present new challenges in terms of bioinformatics support required to maximize their experimental potential.

Three distinct NGS platforms have already attained wide diffusion and availability. Some characteristics of their throughput, read-lengths and costs (at the time of writing) are presented in Table 1.

Table 1. Performances and features of the major next-generation sequencing platforms (single-end reads)

<table>
<thead>
<tr>
<th>Technology</th>
<th>Roche 454</th>
<th>Illumina</th>
<th>ABI SOLiD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>GS 20</td>
<td>GA</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>FLX</td>
<td>GA II</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Ti</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Reads (M)</td>
<td>0.5</td>
<td>28</td>
<td>40</td>
</tr>
<tr>
<td>Read length</td>
<td>100</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>Run time (d)</td>
<td>0.2</td>
<td>0.4</td>
<td>6</td>
</tr>
<tr>
<td>Images (T3)</td>
<td>0.01</td>
<td>0.5</td>
<td>1.8</td>
</tr>
</tbody>
</table>

A common thread for each of these technologies over the last years has been continuous improvement in performance (increased numbers and lengths of reads and consequent reduction in costs per base sequenced), it is therefore anticipated that the figures provided will rapidly become outdated, however, they serve to illustrate that the Roche454 technology[13] already provides a realistic substitute for many applications of conventional Sanger sequencing at greatly reduced cost, while the Illumina Genome Analyser [14] and ABISOLiD [15] platforms generate an order of magnitude more reads of (relatively) reduced length, characteristics that, as we will see, render them, for now, more suitable for other applications. The aforementioned methods all rely on a template amplification phase prior to sequencing. However, the available Helicos technology [16] avoids the amplification step and provides sequence data for individual template molecules, minimizing the risk of introducing substitutions during amplification. In principle bioinformatics approaches developed for the analysis of data generated by the Illumina GA and ABISOLiD platforms should also be suitable for data generated by the Helicos method, as all three platforms provide reads of comparable lengths. Finally, other methods, based on either nanopore technology or tunneling electron microscopy have been proposed (for reviews see [1719]). Detailed information on the performance of such approaches is not yet available, although it is hoped that they could yield individual reads of lengths measured in megabases. Given that such methods remain broadly inaccessible at the present time, and that the nature of
data generated should be fundamentally different from those provided by available platforms. Potential bioinformatics developments connected to these methods are considered to be major challenge for future generations sequencing technologies. Thus to meet the unprecedented flood of data generated by the next generation of DNA sequencers, bioinformatics groups working with NGS data found it necessary to respond quickly and efficiently to the informatics and infrastructure demands. Centralized facilities newly facing this challenge need to anticipate time and design considerations of necessary components, including infrastructure upgrades, staffing, and tools for data analyses and management.

Algorithms and software solutions for de novo sequence assembly in metagenomes

Metagenomics is the sequencing of DNA in an environmental sample. Whereas whole genome sequencing (WGS) targets one genome, metagenomics usually targets several. The metagenomics assembly problem is confounded by genomic diversity and variable abundance within populations. Assembly reconstructs the most abundant sequences [20]. Simulations indicate high rates of chimera, especially in short contigs assembled from complex mixtures [21]. Studies that rely on characterization of individual reads prefer long reads [22]. The role for de novo genomic assembly from NGS metagenomics data grows as NGS read lengths and NGS paired-end options increase.

Computational Assembly

Computational assembly is the only way to efficiently assemble sequenced fragments of DNA. However, a sufficient amount of high quality sequences are required. The assembly programs should be able to handle large data sets effectively and avoid misassemblies in the presence of large repetitive or duplicated regions and redundant sequences. To accomplish this, effective algorithms to handle large input datasets with the use of minimal computer time and memory are needed.

One of the primary difficulties in computational genome assembly is to develop an algorithmic approach capable of detecting stretches of wrong repetitive DNA without causing misassemblies. Repetitive sequences complicate assembly as different pieces of sequence can share the same repeat sequence originating from different genomic locations. Since the pieces are put together by searching for matching overlapping nucleotides, repeats can be put together erroneously. Typically, for shotgun data, repetitive sequences are revealed by clusters containing more overlapping reads than would be expected by chance, illustrated in Fig.4.

In EST data sets the main difficulty is to develop an algorithmic approach that, in addition to efficient assembly, can handle highly expressed genes, paralogous genes, alternative splice forms and chimerism in the data set.

The theoretical background for genome assembly lies in computer science, and an insight into the mathematical and theoretical background can be found in (Pop, 2004) and references therein. Although pyrosequencing with a whole-genome shotgun approach has been successfully applied to bacterial genomes (Margulies et al., 2005), the construction of high-quality assemblies with high-throughput sequencing data is still a non-trivial problem even for short genomes. At present, no approach has been proposed to directly assemble large animal or plant genomes directly from short sequences obtained using HTS. As described below the Short Read Assembly Protocol (SHRAP) (Sundquist et al., 2007), however, comprises a protocol for high-throughput short read sequencing that differs in two respects from classical hierarchical sequencing approaches. This protocol however, expects read length much longer (200 nucleotides) than those produced by
SOLiD or Solexa. The assembly methodology is based on the Euler engine introduced in 2004 (Pevzner et al., 2004). The Euler-SR assembler, specifically designed to assemble short reads, uses an updated version of the Euler engine to reduce memory requirements. The results for real Solexa reads, however, were less convincing (Chaisson and Pevzner, 2008) due to the poorly understood error model and highly variable error rates across different machines and run times.

**Basic Principles of Assembly**

For the majority of traditional assembly programs the basic scheme is the same, namely the overlap-layout-consensus approach. Essentially it consists of the following steps (Green Laboratory, 1994; Huang and Madan, 1999):

- Sequence and quality data are read and there ads are cleaned.
- Overlaps are detected between reads. False overlaps, duplicate reads, chimeric reads and reads with self-matches (including repetitive sequences) are also identified and left out for further treatment.
- The reads are grouped to form a contig layout of the finished sequence.
- A multiple sequence alignment of the reads is performed, and a consensus sequence is constructed for each contig layout (often along with a computed quality value for each base).
- Possible sites of misassembly are identified by combining manual inspection with quality value validation.

**General Assembler Differences**

When different assemblers try to piece the DNA puzzle together they essentially work from the same input, but the assemblers differ in the way they utilize the sequence information, and in the way this is combined with additional information. In general the differences fall in the following categories.

- **Overlaps:** A lot of different methods are used to find potential overlaps between sequences. Some are based on BLAST (e.g. gene Distiller Gilchrist et al., 2004; while other assemblers use various other methods to find similarities between reads.

- **Additional information:** Depending on how the sequence reads are produced some additional information might be available. This information might consist of read pair information, BAC clone information, base quality information, etc. Some assemblers use this data to impose additional structure on the assembly of the sequences (e.g. GigAssembler International Human Genome Sequencing Consortium, 2001).

- **Short read assembly:** De novo assembly of the microreads generated from next generation sequencing platforms is still challenging. While assemblers have been developed and applied to assemble bacterial genomes successfully (Chaisson and Pevzner, 2008; Hernandez et al., 2008), on larger genomes the assembly is performed by mapping the microreads to reference genomes. The major next generation sequencing platforms all have built-in software to handle this task, e.g. GS Reference mapper, Gerald for Solexa. In SOLiD systems the mapping tool “mapreads” converts reference sequences into color space and perform the mapping in color space.

In assemblers aimed at short read assembly an approach based on mathematical graph theory is often used, namely the Eulerian fragment assembly method. The Eulerian fragment assembly avoids the costly computation of pair wise alignments between reads (Pevzner et al., 2001). The De
Bruijn graph of a genome has as its vertices all distinct $k$-tuples that occur within the sequence (where $k$ is the word length that is used). A directed edge is inserted between $s$ and $t$ if there is a $k$-tuple $(u_1, u_2, \ldots, u_k)$ in the genome such that $s = (u_1, u_2, \ldots, u_{k-1})$ and $t = (u_2, \ldots, u_{k-1}, u_k)$ if $s$ and $t$ appear shifted by single nucleotide. A sketch of a graph construction procedure is shown in Fig.2. In practice one uses the $k$-tuples appearing in the collection of the sequence reads and a value of $k$ between 6 and 9 or 10. In the error-free case, the genomic sequence can be read off directly as an Eulerian path through the De Bruijn graph (with repeats forming “tangles”). In real, error-prone data underrepresented $k$-tuples, i.e. $k$-tuples that appear less frequently than expected from the coverage rate, indicate sequencing errors and can be omitted.

**Fig 2.** Graph example. An example of how a graph is constructed. Two reads are mapped onto the different k-mer nodes ($k = 6$ in this example), and edges between the nodes are determined by the reads. The presence of a nucleotide difference (e.g. sequencing error, SNP, etc.) between the two reads cause the graph to split up, thus causing ambiguity in the sequence.

A somewhat related issue is how the sequences are cleaned of errors and contaminant sequences (i.e. vector sequences, repeat sequences, etc.). While this can essentially be considered separately and independently from the assembly itself, some assemblers incorporate cleaning in the way they process the reads (e.g. Batzoglou et al., 2002).
Fig. 3. Assembly pipeline. The typical pipeline of a sequencing project. Sequenced reads are generated, after which they are cleaned and assembled. Following the assembly annotation and analysis can be performed. The grey line show the pipeline for massively parallel sequencing where the reads are mapped to a reference genome, while the full pipeline is for de novo sequencing and assembly.

Error detection in metagenomic NGS data

Because of the nature of metagenomic data, it is neither possible to resample the data to account for the sequencing errors that inevitably occur, nor it is possible to clearly differentiate between an error and a biological variation. Small errors in the sampled data often lead to significant changes in the results of any further analyses and studies based on the data, for example during the construction of phylogenetic trees or during the evaluation of the biological diversity in the sampled environment. For improving the quality of such studies, it is essential that an approach for detecting probable errors is devised.

There are numerous published methods for error detection and correction in NGS data, but none of them are designed to work with metagenomic data, and instead focus on applications such as de novo sequencing of genomes where the appearance of biological variations that are undistinguishable from the errors are not an issue. An example of such software is SHREC which corrects errors in short-read data using a generalized suffix trie, which we use as basis for comparison.

The input data for the initial tests consists of 16S RNA short-reads. 16S RNA is very useful for metagenomic studies, because it contains highly conserved regions that can be easily isolated and compared, and at the same time it contains hypervariable regions that are greatly useful for making a distinction between different species. At the same time, this makes it more difficult to process the data. Our tests are done on sets of a few tens of thousands reads with lengths between 300 and 500 bases. For the proposed method to be applied, the read sets need to be filtered of obvious noise and then aligned to each other so that all functional parts can be easily compared to one another.
The basic idea behind error correction is that if a given a bit of data such as a single base appears too little in the dataset it is more likely for it to be an error than a biological variation, and a threshold can be established using the error rate of the sequencing equipment. This can be significantly improved if during the assessment, a discrimination on similarity is made by giving higher weights to reads that are similar to the read in question in the evaluated region. The outline of the evaluation algorithm is as follows:

1. We go over the reads evaluating each base individually.
2. For each base in question, we create a window containing the base at its centre.
3. We calculate a similarity score for the region in the window between the read in question and every other read in the dataset. The score excludes the evaluated base and the bases closest to it are assigned the biggest weights.
4. We calculate an evaluation score for the base by calculating a frequency weighted with the similarity score. The result is the ratio of the sum of the similarity scores for the reads that contain the base and the sum of the similarity scores for all the reads.
5. We compare the score of the base to a threshold that has been calculated in advance and experimentally verified. Any scores below the threshold are considered errors and the bases are replaced with the base candidate that would score most using the outlined algorithm.

The biggest challenge in the implementation of this approach apart from the evaluation of the parameters is the preprocessing of the data, in particular the sequence alignment. For a sensible result, the reads have to be properly aligned, which is an extremely resource intensive task even for the used task datasets. Trading accuracy for speed is not desirable as alignment errors affect both the evaluation and any further studies.

For the parameter evaluation we are also experimenting with an automatic approach which would use simulated data that matches the manufacturer specifications on error rates and is constructed in awareness of the mutation rates in nature, and the different thresholds would be tested against the simulated data set.

### Parallel computing opportunities for error detection, for metagenomic studies and other sequencing tasks

The processing of a great number of large samples of metagenomic data creates a computational challenge. While it is interesting to discuss parallel execution of the error detection algorithm itself, the preprocessing step is both more expensive and a more challenging problem to solve. Due to an inherent similarity between the preprocessing and some of the further studies, such as phylogenetic trees, the usefulness of parallel computation would double.

The popular algorithms for multiple sequence alignment are based on the construction of a rough similarity score matrix for the reads which is then used to construct a tree in which similar reads are neighbours using the neighbour joining method or some derivative, then neighbours are aligned against each other. While it is trivial to compute the the similarity matrix in parallel, the same is difficult for the construction of the tree.

The direct approach requires the use of shared memory which is neither popular in computer clusters, nor does it scale well. To overcome the need for shared memory, there are published methods that rely on heuristics to split the data into groups by similarity, then every group can be aligned on its own and all the groups can be aligned in parallel. This is similar to the approach
that we're currently using in our error detection pipeline, although the final choice for a suitable heuristic has not been made yet.

The error evaluation is almost independent for each region, so it won't be a challenge to run multiple regions in parallel, and the slight window overlap would waste an insignificant amount of memory. However, it is one of the less intensive task in the pipeline. If the number of reads is n, and the maximum read length is m, then the matrix calculation and the error evaluation are of time complexity $O(mn^2)$, while the tree construction is of time complexity $O(n^3)$, and thus it is the main focus, although some of the heuristics for splitting the dataset can reduce the time complexity or otherwise significantly reduce the CPU time required for the tree construction, so a parallel implementation for all the steps is preferable.

Parallel processing would be crucial for the actual studies as well. Essentially, the construction of a phylogenetic tree is almost identical to the algorithm for multiple sequence alignment. The only difference is that the similarity matrix is computed over sequences that have been already aligned and a different metric for the distance is used, so solving either would also solve the other one, and both steps have essentially the same computational requirements. Any differences created by the changing of the choice of distance or the sequence gaps are negligible.

The data sets for a given scientific experiment are not only big, but they also tend to be numerous. In fact they tend to grow more in number than in size, which makes them an excellent target for parallel processing regardless of the algorithms used during the data processing, and regardless of the nature of the task that is to be solved.

**Discussion and conclusions**

New high-throughput sequencing technologies have emerged. However, the sequencing methods as well as the computational tools have to be further improved, to allow a complete de novo assembly for large genomes with these technologies. However, today only little data on the error models of different massively parallel sequencing technologies is available. These error models are crucial to interpret and analyze the sequence data correctly (Dohm et al., 2008). When it comes to de novo assembly, the short read lengths of some of the major technologies seem to be a momentous disadvantage and the high number of reads produced might not be able to compensate for this handicap. However, all manufacturers aim to increase the read lengths. Currently, areas on able approach to the assembly of such short sequences could include data from low coverage Sanger sequencing aside with data from the NGS platforms. Although hybrid data set approaches are cumbersome (Chaisson and Pevzner, 2008), they have already been shown to produce useful assemblies (Goldberg et al., 2006).

The choice of sequencing strategy should also be influenced by the goal of the project. In some organisms it might be desirable to quickly generate a few contigs covering key points in the genome, while in others a broader strategy might apply. Still other projects combine whole genome with hierarchical shot gun in a hybrid approach trying to utilize the strengths of each (Havlak et al., 2004).

Common to most assembler solutions, a core set of features is apparent:
- Error detection and correction based on sequence composition of the reads.
- Graph construction to represent reads and their shared sequence.
- Reduction of simple non-intersecting paths to single nodes in the graph.
- Removal of error-induced paths. These are recognized as spurs or bubbles.
- Collapse of polymorphism – induced complexity. This is recognized as bubbles.
- Simplification of tangles using information outside the graph.
- Individual reads or paired-end reads act as constraints on path distance and outcome.
- Conversion of reduced paths to contigs and scaffolds.
- Reduction of alignments to a consensus sequence.