BIOINFORMATICS APPLICATIONS NOTE

Vol. 27 no. 4 2011, pages 578–579
doi:10.1093/bioinformatics/btq683

Genome analysis

Advance Access publication December 12, 2010

Scaffolding pre-assembled contigs using SSPACE

Marten Boetzer[1,2], Christiaan V. Henkel[3], Hans J. Jansen[3], Derek Butler[1] and Walter Pirovano[1,*]

[1]BaseClear B.V., Einsteinweg 5, 2333 CC Leiden, [2]Leiden Institute for Advanced Computer Science, Leiden University, Niels Bohrweg 11, 2333 CA Leiden and [3]ZF-screens B.V., Niels Bohrweg 11, 2333 CA Leiden, The Netherlands

Associate Editor: John Quackenbush

ABSTRACT

Summary: *De novo* assembly tools play a main role in reconstructing genomes from next-generation sequencing (NGS) data and usually yield a number of contigs. Using paired-read sequencing data it is possible to assess the order, distance and orientation of contigs and combine them into so-called *scaffolds*. Although the latter process is a crucial step in finishing genomes, scaffolding algorithms are often built-in functions in *de novo* assembly tools and cannot be independently controlled. We here present a new tool, called SSPACE, which is a stand-alone scaffolder of pre-assembled contigs using paired-read data. Main features are: a short runtime, multiple library input of paired-end and/or mate pair datasets and possible contig extension with unmapped sequence reads. SSPACE shows promising results on both prokaryote and eukaryote genomic testsets where the amount of initial contigs was reduced by at least 75%.

Availability: www.baseclear.com/bioinformatics-tools/.

Contact: walter.pirovano@baseclear.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on August 24, 2010; revised on November 19, 2010; accepted on December 7, 2010

1 INTRODUCTION

The determination of the complete genomic sequence of a new species, also called *de novo* sequencing, is an important next-generation sequencing (NGS) application. The task is to puzzle back millions of short sequence reads into a limited set of contiguous sequences (contigs), although in practice the total number can be rather high. This can partly be attributed to the presence of repetitive elements on the genome. Paired-read sequencing technology may help to reduce the amount of contigs as the known intermediate distance between read pairs can be used to place contigs in their likely order and orientation. The length of the resulting scaffolds (or supercontigs) also reflects the estimated distance between the initial contigs. Illumina for instance supports two types of paired-read libraries: short-insert paired-end and long-insert mate pairs. The combination of these two types of data provides an ideal input for the scaffolding process, as it can potentially resolve repetitive structures of various sizes. Nonetheless, the majority of assembly tools provide a scaffolding option only as a built-in function which cannot be independently controlled, among which SSAKE (Warren *et al.*, 2007), Abyss (Simpson *et al.,* 2009) and SOAP (Li *et al.,* 2010). To date only a few programs are able to scaffold pre-assembled contig sequences. A commonly used tool is Bambus (Pop *et al.*, 2004), although it was not designed for the current generation of sequencing technologies.

We developed the SSPACE (SSAKE-based Scaffolding of Pre-Assembled Contigs after Extension) program to scaffold pre-assemblies produced by any desired assembly tool. The input is given by a set of contig sequences and Illumina paired-read files. The user can specify whether the paired reads are used only for scaffolding or also for contig extension. We compared our method to the Abyss scaffolder on a bacterial (*Escherichia coli*) and fungal (*Grosmannia clavigera*). Both methods significantly reduce the number of contigs in a comparable manner, although SSPACE achieves overall better results. To prove the suitability of SSPACE also on large eukaryotic genomes, we tested its performance against SOAP on the giant panda genome assembly. We show that SSPACE is a valuable tool for scaffolding pre-assembled contigs.

2 METHODS

2.1 Materials

Part of the test data was taken from the NCBI Short Read Archive (SRA). For *E.coli* (strain K-12, MG1655) we used two Illumina paired-end libraries corresponding to an insert size of 200 bp (SRR001665) and 500 bp (SRR001666), respectively. For *G.clavigera* (strain kw1407; DiGuistini *et al.*, 2009), we used a combination of single-end 454 reads (SRR023307 and SRR023517 to SRR023533) and two Illumina paired-end libraries corresponding to insert sizes of 200 bp (SRR018008 to SRR018011) and 700 bp (SRR018012). For both genomes, all sequence reads were assembled as single-ends with Abyss and subsequently scaffolded into supercontigs using the Abyss and SSPACE scaffolders. The minimum number of links required for matching unambiguous contig links was set to 5. For the giant panda genome, we retrieved sequence scaffolds constructed with SOAP from http://panda.genomics.org.cn (Li *et al.*, 2010) and split these on gapped positions larger than 100 bp. The resulting 187.742 contigs (>100 bp) were scaffolded with SSPACE using all available paired-read libraries of high-quality reads (in total 35 libraries with inserts ranging from 0.1 to 12 kb). The minimum number of links required for matching unambiguous contigs was set to 3. All analyses were performed on a 32 GB Linux machine.

2.2 SSPACE algorithm

An overview of the SSPACE algorithm is given in Supplementary Figure S1. First, short-paired DNA reads are filtered by removing sequences containing non-ACTG characters. The remaining read pairs are mapped against the pre-assembled contigs using Bowtie (Langmead *et al.*, 2009). The position and orientation of each pair that could be mapped is stored in a hash. Hereafter a post-filtering step is applied to remove duplicate read-pairs. Optionally, the pre-assembled contigs can be extended using sequence reads that could not

*To whom correspondence should be addressed.

Table 1. Abyss and SSPACE compared on the *E.coli* dataset

	No. contigs/ scaffolds	N50	% of shared contig-pairs	% of shared nucleotides
Original	419	21.181		
Abyss	84	119.416		
SSPACE	75	177.880	85.9	88.0
SSPACE extension	69	177.891	85.1	87.5

Results are displayed for contigs/scaffolds larger than 300 bp. The SSPACE protocols yield less scaffolds and a larger N50 value compared with Abyss. The percentage of shared contig-pairs is based on the co-occurrence of contig-pairs between Abyss and SSPACE (extension). The percentage of shared nucleotides is calculated from the number of nucleotides that are involved in contig-pairs.

be mapped. This feature is especially designed to incorporate the paired-read datasets that were not used in the pre-assembly.

The next step in the SSPACE protocol is scaffolding, which is modified and extended from the SSAKE short-read assembler (Warren *et al*., 2007). In brief, putative contig pairs (pre-scaffolding stage) are computed based on the position of the paired reads on different contigs. Contig pairs are only considered if the calculated distances between them satisfy the user-defined distance range. After pairing the contigs, scaffolds are formed by iteratively combining contigs if a minimum number of read pairs (k) support the connection (default $k=5$), starting with the largest contig. We have introduced a novel step to deal with contigs that have alternative connections (see also Supplementary Figure S2). If connections are also found between the alternatives themselves, the algorithm seeks to place all alternatives in the correct order using the estimated insertion. Otherwise a ratio is calculated between the two best alternatives. If this ratio is below a threshold (default $a=0.7$), a connection with the best scoring alternative is established. Extension of scaffolds is aborted if either a contig has no links with other contigs or the ratio for alternatives is exceeded. The scaffolding process is repeated until all contigs are incorporated into linear scaffolds. Our program has been designed to allow for multiple library input sets that are scaffolded in a hierarchical manner (starting with small insert libraries).

SSPACE provides the following output files: the final scaffolds (FASTA format) and a complementary file listing the contigs that build up each scaffold. Also a summary file containing useful statistics such as the total number of scaffolds, their (average) size and N50 statistics is provided. Optionally the connections within each scaffold can be graphically viewed.

3 RESULTS

The performance of SSPACE was compared with the built-in scaffolder of Abyss on both the *E.coli* and *G.clavigera* dataset. The contigs produced by Abyss after its single-end assembly stage served as input for SSPACE. For SSPACE, we used two protocols, one including and one excluding contig-extension. The outcomes are displayed in Tables 1 and 2. Notably, the SSPACE protocols yield a lower amount of scaffolds and a significantly higher N50 value. It can also be observed that the inclusion of contig-extension prior to scaffolding yields slightly less scaffolds. We also assessed that the consistency between Abyss and SSPACE results by evaluating the co-occurrence of contig-pairs in the two scaffold sets. The results indicate that Abyss and SSPACE produce comparable scaffolds. Additional results on the *E.coli* dataset are displayed in Supplementary Table S3.

Analyses on the giant panda dataset were performed similarly to the previous two sets. Here, we used the scaffolds produced by SOAP to construct a set of 187.742 unlinked contigs. These were re-scaffolded with SSPACE. The results (Table 3) show that SOAP and SSPACE can concatenate a major fraction of the contigs into scaffolds (>95%) that share 80.6% of the nucleotides. However, SSPACE yields less scaffolds and a fundamentally improved N50 value. Notably only a limited amount of computational resources was required to scaffold all 35 libraries.

Table 2. Abyss and SSPACE compared for *G.clavigera*

	No. contigs/ scaffolds	N50	% of shared contig-pairs	% of shared nucleotides
Original	3.475	16.023		
Abyss	1.350	145.688		
SSPACE	1.338	301.624	82.4	84.6
SSPACE extension	1.315	319.647	81.1	82.1

See capture of Table 1. Results are displayed for contigs/scaffolds larger than 300 bp.

Table 3. Scaffolding giant panda contigs with 35 different libraries.

	No. contigs/ scaffolds	N50	% of shared contig-pairs	% of shared nucleotides
Original	91.601	44.601		
SOAP	4.585	1.315.838		
SSPACE	2.041	3.718.553	93.5	94.8

See capture of Table 1. Results are displayed for contigs/scaffolds larger than 1 kb.

4 CONCLUSION

We have shown that SSPACE is a powerful and effective stand-alone scaffolder. SSPACE is able to scaffold large genomes in a reasonable amount of time even when using huge datasets. Our method is suited both for prokaryotic and eukaryotic genomes and in comparison with two built-in scaffolders SSPACE achieves competitive results. We demonstrate that applying contig extension prior to scaffolding can further enhance the outcomes. Importantly, SSPACE is designed in a user-friendly manner and requires limited computation resources (Supplementary Table S4). We think our method is a useful addition to present NGS *de novo* assembly tools in the automated reconstruction of genomic sequences.

ACKNOWLEDGEMENT

The authors would like to acknowledge Rene Warren for help with the code and discussions.

Funding: SenterNovem (grant SO201003).

Conflict of Interest: none declared.

REFERENCES

DiGuistini,S. *et al.* (2009) *De novo* genome sequence assembly of a filamentous fungus using Sanger, 454 and Illumina sequence data. *Genome Biol.*, **10**, R94.

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

Li,R. *et al.* (2010) The sequence and *de novo* assembly of the giant panda genome. *Nature*, **463**, 311–317.

Pop,M. *et al.* (2004) Hierarchical scaffolding with Bambus. *Genome Res.*, **14**, 149–159.

Simpson,J.T. *et al.* (2009) Abyss: a parallel assembler for short read sequence data. *Genome Res.*, **19**, 1117–1123.

Warren,R.L. *et al.* (2007) Assembling millions of short DNA sequences using SSAKE. *Bioinformatics*, **23,** 500–501.