BIOINFORMATICS APPLICATIONS NOTE Vol. 28 no. 23 2012, pages 3141–3143
doi:10.1093/bioinformatics/bts586

Genome analysis

Advance Access publication October 7, 2012

KGBassembler: a karyotype-based genome assembler for *Brassicaceae* species

Chuang Ma, Hao Chen, Mingming Xin, Ruolin Yang and Xiangfeng Wang*

School of Plant Sciences, University of Arizona, Tucson, AZ 85721, USA

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: The *Brassicaceae* family includes the most important plant model *Arabidopsis thaliana* and many cruciferous vegetable crops. A number of close relatives of *Arabidopsis* and economically important *Brassica* species are being sequenced with whole-genome shotgun sequencing technologies. However, *de novo* assembly of full chromosomes is difficult, since many non-model *Brassicaceae* species are lacking genetic and/or physical maps. As a unique feature for *Brassicaceae*, the genome of each member is composed of 24 conserved chromosomal blocks, and the arrangement of the 24 blocks can be obtained from karyotype analysis via comparative chromosome painting experiments. Taking this advantage, we developed a bioinformatic tool named KGBassembler to automatically finalize assembly of full chromosomes from contigs and/or scaffolds based on available karyotypes of *Brassicaceae* species.

Availability: KGBassembler was implemented in C++ with a graphical user interface. It is freely available to academic users at http://www.cmbb.arizona.edu/KGBassembler/.

Contact: xwang1@cals.arizona.edu

Received on April 30, 2012; revised on August 20, 2012; accepted on September 23, 2012

1 INTRODUCTION

The *Brassicaceae* family contains about 3700 species, including the most well-known model plant *Arabidopsis thaliana* (*Arabidopsis*), and many agronomically important vegetable crops (e.g. *Brassica rapa*, *B. oleracea*) and numerous species with promising use for biodiesel production (e.g. *B. napus*, *B. juncea*). Several projects have been launched to sequence the genomes of a selection of *Brassicaceae* species, for instance, the *Brassicales* Map Alignment Project (http://www.brassica.info). However, because of lacking genetic and/or physical maps for many of these non-model plants, the sequence reads from whole-genome shotgun sequencing can be only assembled to contigs or scaffolds.

An alternative strategy to achieve the assembly of chromosomes from contigs or scaffolds is utilizing the genome synteny from the completed genomes of model organisms. This is especially feasible for *Brassicaceae*, because it was postulated that the genome of species in this family is composed of 24 conserved chromosomal blocks, whose arrangement can be obtained from comparative chromosome painting (CCP) analyses (e.g. Mandakova and Lysak, 2008; Schranz *et al.*, 2006). Numerous karyotypes in different lineages of *Brassicaceae* have been profiled with CCP experiments, such as *A. thaliana* ($n = 5$), *Eutrema salsugineum* ($n = 7$), *A. lyrata* ($n = 8$), *B rapa* ($n = 10$) and so on (Lysak and Koch, 2011; Mandakova and Lysak, 2008; Schranz *et al.*, 2006). Thus, these existing and forthcoming karyotype maps of *Brassicaceae* can be used to finalize the assembly of chromosomes. The feasibility of karyotype-based assembly has been shown in the assembly of *Thellungiella (Schrenkiella) parvula* genome independent of genetic and/or physical maps (Dassanayake *et al.*, 2011).

Taking this advantage, we developed the KGBassembler, a tool to assemble full chromosomes from scaffolds if a karyotype is provided. The KGBassembler is featured with a user-friendly graphical user interface (GUI), allowing users to use automatic assembling of chromosomes based on CCP-based karyotypes and/or to manually edit the layouts of contigs according to *in silico* generated karyotypes.

2 IMPLEMENTATION

2.1 Architecture

KGBassembler was implemented in C++ with a GUI built with the open-source QT toolbox. The information of 24 blocks, including letter-labels (A–X), color-codes and the orthologous genes in *Arabidopsis*, was integrated into KGBassembler (Schranz *et al.*, 2006). KGBassembler requires three input files: the contig (or scaffold) sequences from *de novo* assembly in FASTA format; the alignment result of *Arabidopsis* genes (protein sequences) against the contigs by BLAT (Kent, 2002) and the karyotype file in plain text. Four outputs will be generated, including the assembly of chromosomes, an *in silico* karyotype, chromosome-scale synteny maps between assembled species and *Arabidopsis*, and a report containing the predicted layouts of contigs on chromosomes.

2.2 Algorithms and workflow

KGBassembler runs in three steps (Phases I–III). Phase I (*Contigs2Blocks*) is to identify the 24 blocks in contigs via analyzing BLAT alignments using two algorithms: local and global search (Fig. 1a). Specifically, according to the BLAT alignments, KGBassembler first assigns the strings of block-labels (A–X) associated with *Arabidopsis* genes to the mapped positions on contigs. Then, the local search is performed to identify the conserved genomic blocks in contigs using a sliding window with size of L adjacent block labels and step of one block label. In each window, KGBassembler finds the block label

*To whom correspondence should be addressed.

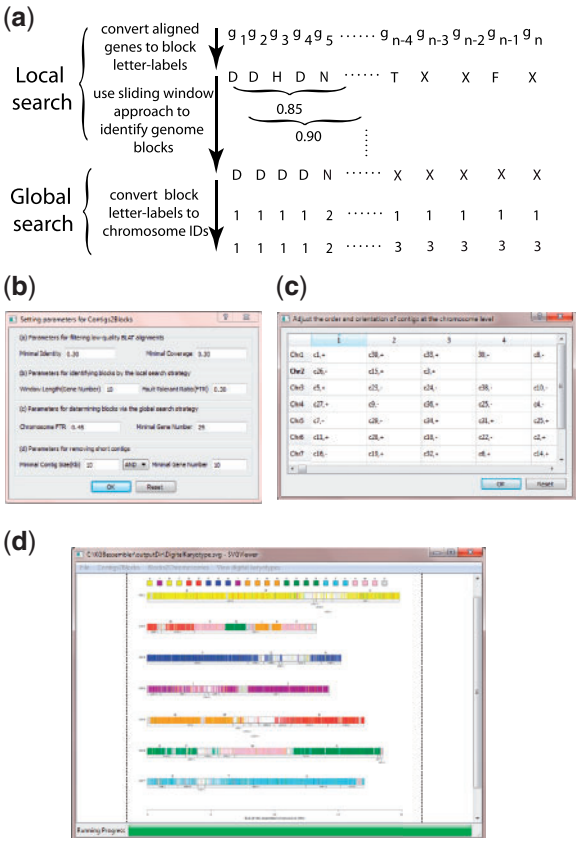

Fig. 1. Algorithms of inferring syntenic blocks in contigs and screenshots of KGBassembler. (**a**) The algorithms of local and global search to infer the 24 syntenic blocks in contigs based on BLAT alignments. The numbers represent the highest proportion of all probable blocks in the analyzed windows. (**b**) Control panel to set up parameters for running Phase I analysis. (**c**) A report panel displaying the layout of contigs on assembled chromosomes and allowing users to perform manual adjustment of layouts at Phase II. (**d**) Visualization of the assembled *in silico* karyotype

($i, i \in [A, B, \cdots, X]$) with the highest number (n) and predicts the corresponding region of this window belonging to blocks i if $n/L \geq$ (1-FTR). Thus, a lower fault-tolerant ratio (FTR) indicates that a strict rule is applied to determine the genome blocks in contigs and yields the assembled chromosomes with higher quality. A less-stringent FTR (e.g. 0.30) is recommended for the first-time run so that more contigs with block information can be retained to estimate the quality and coverage of the genome assembly. Third, the global search is initiated to merge regions with discontinuous strings of block labels with the following steps: (i) convert the block labels of predicted regions to chromosome IDs with the block order information obtained from karyotype file; (ii) find the chromosome ID with the highest proportion (pr) and predict the contig belonging to this chromosome if $pr \geq$ 1-'Chromosome FTR'. Similar to FTR, 'Chromosome FTR' is also a user-adjustable parameter that can be used to control the quality of genome assembly; (iii) mask the blocks that are inconsistent with the majority of predicted chromosome IDs and (iv) merge the adjacent regions as whole chromosomal blocks with the same block labels.

To ensure the accuracy of assigning blocks to contigs, it should be better for users to eliminate chimeric contigs before BLAT mapping. KGBassembler allows users to remove low-quality BLAT alignments by a minimal identity and coverage cutoff and to ignore short contigs (e.g. ≤10 kb and/or ≤5 genes), if the information is not sufficient to assign the blocks to contigs. KGBassembler provides a control panel for users to set up the parameters to optimize the results generated from Phase I (Fig. 1b).

Phase II (*Blocks2Chromosomes*) allows users to manually adjust the layout of contigs to form chromosomes based on the comparison of *in silico* and experimental karyotypes. The predicted assignments of blocks to contigs information from Phase I are organized in a table for manual inspection. The order and orientation of a contig can be further edited to make necessary adjustments to build a high-quality genome assembly (Fig. 1c).

Phase III (*AssemblyFinishing*) connects the contig sequences to generate chromosome sequences based on the manually inspected and edited layouts (i.e. order and orientation) of the contigs assigned to each chromosome. KGBassembler also produces a pseudo-chromosome named 'ChrUd', with contigs whose chromosomes are not clearly determined in Phases I and II. Because of the difficulty of estimating the gap size, KGBassembler directly links the sequences of adjacent contigs together to form the assembled chromosomes. The sequence of contigs with '−' orientation were reverse complemented before the generation of chromosome sequences. Simultaneously, an *in silico* karyotype of the assembled genome is generated in the scalable vector graphics format (Fig. 1d). At last, a series of chromosome-scale synteny maps between assembled species and *Arabidopsis* are produced.

3 RESULTS

We ran KGBassembler to assemble the seven chromosomes from the 1496 gap-free contigs (~137 Mb) of *T. parvula*, sequenced by 454 and Illumina technology (Dassanayake *et al.*, 2011). Based on the *Thellungiella (Eutrema)* karyotype (Mandakova and Lysak, 2008), 40 large contigs were unambiguously connected as seven chromosomes, covering 114.39 Mb (83.44%) of the total sequences. The order of these contigs in the assembled genome was consistent with that in the reported assembly (Dassanayake *et al.*, 2011) (Fig. 1d). KGBassembler can be run on Windows and Linux. Except the time spent on the manual inspection of the contig layouts, KGBassembler can finish assembling the *T. parvula* chromosomes with the input of 1496 contigs and ~400 000 BLAT alignments on a laptop in several minutes. Because the KGBassembler assembles the chromosomes primarily based on gene synteny information, the short contigs, repeat-rich contigs and contigs without sufficient amount of genes were usually ignored. In addition, the performance of KGBassembler depends on the quality of contigs and/or scaffolds preassembled by *de novo* assemblers to achieve the high-quality chromosome assemblies.

Conflict of Interest: none declared.

REFERENCES

Dassanayake,M. *et al.* (2011) The genome of the extremophile crucifer *Thellungiella parvula*. *Nat. Genet.*, **43**, 913–918.

Kent,W.J. (2002) BLAT—the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.

Lysak,M.A. and Koch,M.A. (2011) Phylogeny, genome, and karyotype evolution of crucifers (*Brassicaceae*). In Schmidt,R. and Bancroft,I. (eds.) *Genetics and Genomics of the Brassicaceae*. Springer, New York, pp. 1–31.

Mandakova,T. and Lysak,M.A. (2008) Chromosomal phylogeny and karyotype evolution in $x=7$ crucifer species (*Brassicaceae*). *Plant Cell*, **20**, 2559–2570.

Schranz,M.E. *et al.* (2006) The ABC's of comparative genomics in the *Brassicaceae*: building blocks of crucifer genomes. *Trends Plant Sci.*, **11**, 535–542.