RESEARCH

Tools and pipelines for BioNano data: molecule assembly pipeline and FASTA super scaffolding tool

Jennifer M Shelton[*], Michelle C Coleman, Nic Herndon, Nanyan Lu and Susan J Brown

[*]Correspondence:
sheltonj@ksu.edu
Department of Biology, Kansas State University, Manhattan, KS, USA
Full list of author information is available at the end of the article
[†]Equal contributor

Abstract

First part title: Text for this section.

Second part title: Text for this section.

Keywords: Genomic physical map; Bionano; Genome scaffolding; Genome validation

Background

In this article we present software that leverages physical maps assembled from ultra-long single molecule maps to improve the contiguity of the genome assemblies. We report the results of applying these tools to the *Tribolium castaneum* genome.

Data formats

The tools described make use of three file formats developed by BNG. The Irys platform images ultra-long molecules of genomic DNA that are nick-labeled at 7 (bp) motifs using one or more nicking endonucleases and fluorescently labeled nucleotides. Molecules captured in the TIFF images are converted to BNX format text files that include label position for each molecule (steps 1 and 2 Figure 1). Consensus map (CMAP) files include the length and label position for long genomic regions that are either inferred from assembly of BNX molecules (steps 7 and 8 figure 1) or from *in silico* labeling of a sequence-based assembly (steps 3 and 4 figure 1). The two types of CMAPs are referred to as BNG CMAPs and *in silico* CMAPs respectively. The alignment of two CMAPs is stored as an XMAP alignment text file which includes alignment coordinantes and an alignment confidence score (step 10 Figure 1).

Implementation

Assembly preparation

We have developed AssembleIrysCluster to prepare BNX files for assembly and write nine customized assembly scripts (sections C-G Figure 2).

Molecule Stretch

The first stage of AssembleIrysCluster is to adjust molecule stretch (section C Figure 2). Imaged molecules are presumed to have 500 bases per pixel (bpp). Stretch, or bpp, can deviate from 500 bpp and this discrepancy can vary from scan to scan for a flowcell (ADD BPP GRAPH TO SUPPLEMENTARY FILES). Sequence-based assemblies are considered to be more accurate than raw BNX molecules in terms of label position. Therefore BNX files are split by scan number, aligned to the *in silico* CMAP and a empirical bpp value is determined. The bpp indicated by this alignment is used to adjust BNX bpp to 500. Optimal average flow cell signal-to-noise ratio and a low degree of genomic divergence between the samples used for the *in silico* CMAP and BNX molecules have been associated with a characteristic pattern of empirically determined bpp when using the most recent flow cell model and chemistry (ADD BPP GRAPH TO SUPPLEMENTARY FILES HIGHLIGHT DIVERGENT AND SNR ISSUES VS SUCCESSFUL ALIGNMENT). Therefore bpp observed in alignments of scans are plotted as a QC graph (section D Figure 2). Once stretch has been evaluated and adjusted the split BNX files are merged (section E Figure 2).

Customization of Assembly Scripts

In the next stage of AssembleIrysCluster various assembly scripts are created to explore a range of parameters with the goal of selecting the optimal assembly for downstream analysis (sections F-G Figure 2). The merged adjusted BNX file is aligned to the *in silico* CMAP. The alignment error profile is used with the estimated genome length to calculate default assembly parameters and the eight other scripts include variants of these. Initially three assemblies are run with $p-ValueThreshold = \frac{1e-5}{GenomeLength(Mb)}$, $p-ValueThresholdStrict = \frac{p-ValueThreshold}{10}$ and $p-ValueThresholdRelaxed = p-ValueThreshold \times 10$ (section F Figure 2). Minimum molecule length is set to 150 kb. If these do not produce a satisfactory assembly than two minimum molecule length variants (180 kb and 100 kb) are tested with the $p-ValueThreshold$ of the current best assembly (section G Figure 2). Ultimately up to nine assemblies are run until an assembly is produced that is satisfactory.

Assembly Optimization

Ultimately the goal is to produce a CMAP that can be used to guide sequence-based haploid reference genome assembly. While BNG CMAPs can be used to reconstruct haplotypes [1] genome assembly involves collapsing polymorphisms arbitrarily into a consensus reference genome. Therefore for an ideal BNG CMAP $Length(Mb) = GenomeLength(Mb)$. Additionally, 100% of the BNG CMAP would align non-redundantly to 100% of the *in silico* CMAP. When BNG CMAPs are imperfect optimal assembly length is balanced against alignment quality when selecting the best CMAP.

Super scaffolding

Text

Results and Discussion

Text

Conclusions

Text

Availability and requirements

Assembly scripts

Project name: AssembleIrysCluster.pl

Project home page: AssembleIrysCluster scripts are available on Github at https://github.com/i5K-KINBRE-script-share/Irys-scaffolding/tree/master/KSU_bioinfo_lab/assemble_SGE_cluster

Operating system(s): SGE Linux (Gentoo) cluster

Programming language: Perl, Rscript, Bash

License: AssembleIrysCluster.pl is available free of charge to academic and non-profit institutions.

Any restrictions to use by non-academics: Please contact authors for commercial use.

Dependencies: AssembleIrysCluster.pl requires DRMAA job submission libraries. RefAligner and Assembler are also required and can be provided by request by Bionano Genomics http://www.bionanogenomics.com/.

Super scaffolding scripts

Project name: stitch.pl

Project home page: stitch scripts are available on Github at https://github.com/i5K-KINBRE-script-share/Irys-scaffolding/tree/master/KSU_bioinfo_lab/stitch

Operating system(s): MAC and LINUX (tested on Gentoo and Ubuntu)

Programming language: Perl, Bash

License: stitch.pl is available free of charge to academic and non-profit institutions.

Any restrictions to use by non-academics: Please contact authors for commercial use.

Dependencies: stitch.pl requires BioPerl. RefAligner and Assembler are also required between iterations and can be provided by request by Bionano Genomics http://www.bionanogenomics.com/.

Map summary scripts

Project name: cmap_stats.pl and xmap_stats.pl

Project home page: all scripts are available on Github at https://github.com/i5K-KINBRE-script-share/Irys-scaffolding/tree/master/KSU_bioinfo_lab/map_editing

Operating system(s): MAC and LINUX (tested on Gentoo and Ubuntu)

Programming language: Perl

License: cmap_stats.pl and xmap_stats.pl are available free of charge to academic and non-profit institutions.

Any restrictions to use by non-academics: Please contact authors for commercial use.

Dependencies: cmap_stats.pl and xmap_stats.pl have no dependencies.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Text for this section . . .

Acknowledgements

Text for this section . . .

References

1. Lam, E.T., Hastie, A., Lin, C., Ehrlich, D., Das, S.K., Austin, M.D., Deshpande, P., Cao, H., Nagarajan, N., Xiao, M., Kwok, P.: Genome mapping on nanochannel arrays for structural variation analysis and sequence assembly. Nature Biotechnology **30**(8) (2012). doi:10.1038/nbt.2303

Figures

Figure 1 Data analysis steps. (1) Autonoise converts TIFF images of molecules to (2) BNX text files. (3) Genome FASTA is *in silico* labeled with fa2cmap_multi producing (4) a CMAP. (5) AssembleIryscluster uses *in silico* CMAPs, BNX files and estimated genome size to (6) adjust molecule stretch and set assembly parameters. (7) Assembler produces (8) a BNG CMAP. (9) RefAligner aligns the BNG CMAP to the *in silico* CMAP producing (10) an XMAP. (11) XMAP, *in silico* CMAP and BNG CMAP are used by stitch to produce a super scaffolded FASTA. (13) Until no more super scaffolds are created the super scaffold FASTA is *in silico* labeled with fa2cmap_multi producing (14) a CMAP that is aligned to (9) the BNG CMAP and steps 10-15 are iterated.

Figure 2 Assembly workflow for assemble_SGE_cluster.pl. (A) The Irys instrument produces tiff files that are converted into BNX text files. (B) Each chip produces one BNX file for each of two flowcells. (C) BNX files are split by scan and aligned to the sequence reference. Stretch (bases per pixel) is recalculated from the alignment. (D) Quality check graphs are created for each pre-adjusted flowcell BNX. (E) Adjusted flowcell BNXs are merged. (F) The first assemblies are run with a variety of p-value thresholds. (G) The best of the first assemblies (red oval) is chosen and a version of this assembly is produced with a variety of minimum molecule length filters.

Figure 3 Steps of the stitch.pl algorithm. BNG CMAPs (blue) are shown aligned to *in silico* CMAPs (green). Alignments are indicated with grey lines. CMAP orientation for *in silico* CMAPs is indicated with a ”+” or ”-” for positive or negative orientation respectively. (A) The *in silico* CMAP is aligned as the reference. (B) The alignment is inverted and used as input for stitch.pl. (C) The alignments are filtered based on alignment length (purple) relative to total possible alignment length (black) and confidence. Here assuming all alignments have a high confidence score and the minimum percent aligned is 30% two alignments fail for aligning over less than 30% of the potential alignment length for that alignment. (D) Filtering produces an XMAP of high quality alignments with short (local) alignments removed. (E) High quality scaffolding alignments are filtered for longest and highest confidence alignment for each *in silico* CMAP. Third alignment (unshaded) is filtered because the second alignment is the longest alignment for in silico CMAP 2. (F) Passing alignments are used to super scaffold (captured gaps indicated in dark green). (G) Stitch is iterated and additional super scaffolding alignments are found using second best scaffolding alignments. (H) Iteration takes advantage of cases where *in silico* CMAPs scaffold BNG CMAPs as *in silico* CMAP 2 does. Stitch is run iteratively until super scaffolding alignments are found.

Figure 4 Putative haplotypes assembled as BNG CMAPs. (A) Two BNG CMAPs (blue with molecule coverage shown in dark blue) align to the *in silico* CMAP of scaffold 131 (green with contigs overlaid as translucent colored squares). (B and C) Both BNG CMAPs are shown (blue) with molecule pileups (yellow). Both BNG CMAPs have similar label patterns except within the lower coverage region indicated with a black square.

Tables

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Figure 5 Histogram of gap lengths in Tcas5.1. Positive and negative gaps lengths for Tcas5.1 added to the automated output of stitch.pl based on filtered scaffolding alignments. The majority of gap lengths added by stitch.pl, 66, were positive (red). The remaining 26 gaps had negative lengths (purple).

Figure 6 Extremely small negative gap length for in silico CMAP of scaffold 81. Two XMAP alignments for *in silico* CMAP of scaffold 81 are shown. BNG CMAPs (blue with molecule coverage shown in dark blue) align to the *in silico* CMAPs of scaffolds (green with contigs overlaid as translucent colored squares). Scaffolds 79-83 were placed within ChLG 5 and scaffolds 99-103 were placed with ChLG 7 by the *Tribolium* genetic map. (A) Half of the *in silico* CMAP of scaffold 81 aligns with its assigned ChLG (black arrow). (B) The other half aligns with ChLG 7 (red arrow) producing a negative gap length smaller than -20 kb. The alignment that places scaffold 81 with ChLG 7 disagrees with the genetic map and was manually rejected for Tcas5.2.

Table 1 Assembly Results. Assembly metrics for Tcas5.0 (the starting scaffolded FASTA), the *in silico* CMAP, the BNG CMAP of assembled molecules and the final super scaffolded FASTA (Tcas5.2) produced using stitch.pl for the *Tribolium* genome.

	N50 (Mb)	Number	Cumulative Length (Mb)
Genome FASTA	1.16	2240	160.74
in silico CMAP	1.20	223	152.53
BNG CMAP	1.35	216	200.47
Super scaffold FASTA	4.46	2150	165.92

Table 2 Alignment of BNG assembly to reference genome. Breadth of alignment coverage (non-redundant alignment), length of total alignment (including redundant alignments) and percent of CMAP covered (non-redundantly) were calculated for the *in silico* CMAP and the BNG CMAP of the *Tribolium* genome the using xmap_stats.pl.

	Breadth of alignment coverage (Mb)	Length of total alignment (Mb)	Percent of CMAP aligned
in silico CMAP from FASTA	124.04	132.40	81
BNG CMAP	131.64	132.34	67

Table 3 Chromosome linkage groups before and after super scaffolding. The number of scaffolds and contigs in the Tcas5.0 ChLG bins and the number of super scaffolds, scaffolds and contigs in the Tcas5.2 ChLG bins. The number of scaffolds or contigs that were unplaced in Tcas5.0 and placed with a ChLG in Tcas5.2 is also listed.

Chromosome linkage group (ChLG)	Tcas5.0 scaffolds	Tcas5.2 super scaffolds	Unplaced scaffolds added in Tcas5.2
X	13	2	2
2	18	10	1
3	29	20	4
4	6	2	2
5	17	4	1
6	12	6	6
7	15	6	0
8	14	8	1
9	21	9	0
10	12	11	2
Total	157	78	19

Additional file 2 — Sample additional file title
Additional file descriptions text.