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: Genome map; BioNano; Genome scaffolding; Genome validation; Genome finishing

Background

In this article we present software that leverages genome maps assembled from ultra-long single molecule maps to improve the contiguity of sequence assemblies. We report the results of applying these tools to a 7x Sanger draft of 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). These BNX files are referred to as molecule maps. Consensus Map (CMAP) files include the molecule length and label position for long genomic regions that are either inferred from assembly of BNX molecules (steps 7 and 8 figure 1) or *in silico* from sequence scaffolds (steps 3 and 4 figure 1). The two types of CMAPs are referred to as BioNano genome maps and *in silico* genome maps respectively. The alignment of two CMAPs is stored as an XMAP text file which includes alignment coordinates and an alignment confidence score (step 10 Figure 1).

Implementation

Assembly preparation

We have developed AssembleIrysCluster to prepare BNX files for assembly and to 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 within a flowcell (ADD BPP GRAPH TO SUPPLEMENTARY FILES). Sequence scaffolds 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* genome map 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* genome map 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* genome map. 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 $p-ValueThresholdDefault = \frac{1e-5}{GenomeLength(Mb)}$, $p-ValueThresholdStrict = \frac{p-ValueThreshold}{10}$ 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 then 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). Between three and 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 BioNano genome maps can be used to reconstruct haplotypes [1] genome assembly involves collapsing polymorphisms arbitrarily into a consensus reference genome. Therefore for an ideal BioNano genome map $Length(Mb) = GenomeLength(Mb)$. Additionally, 100% of the BioNano genome map would align non-redundantly to 100% of the *in silico* genome map. When BioNano genome maps are imperfect optimal assembly length is balanced against alignment quality when selecting the best CMAP.

Stitch: Alignment Filters

Alignments, XMAPs, of the *in silico* genome maps and the BioNano genome maps are used the predict the higher order arrangement of genome scaffolds with Stitch (Figure 3). RefAligner is designed to treat the reference genome map as an *in silico* genome map and the query as the BioNano genome map. Therefore alignment XMAPs are first inverted and sorted by BioNano genome map coordinates for efficient parsing by Stitch (section A and B Figure 3).

Before inferring super scaffolds from XMAPs, Stitch filters low quality alignments by confidence score. Alignments of *in silico* and BioNano genome maps are assigned a confidence score that is the $-log10$ of the False Positive p-Value. Misaligned labels and sizing error increase the alignment False Positive p-Value decreasing confidence scores [**?**].

Alignments are also filtered by the percent of the total possible alignment length that is aligned (section C Figure 3). Super scaffolds are built from overlapping alignments. Overlapping alignments are similar to global alignments, i.e. alignments spanning from end to end for two maps of roughly equal length, but to search for overlap alignments gaps after the ends of either map are not penalized. The RefAligner scoring scheme does not currently have a parameter to favor overlapping alignments, e.g. to initialize the dynamic programming matrix with no penalties and take the maximum score of the final row or column in the matrix. Refaligner reports local alignments between two maps and applies a fixed penalty based on the user defined likelihood of unaligned labels at the ends of an alignment. Raising or lowering this penalty selects for local or global alignments respectively but neither option favors overlapping alignments specifically. To approximate scoring that favors overlapping alignments Stitch uses thresholds for minimum percent of total possible aligned length or the percent aligned threshold (PAT).

Like a scoring structure that favors overlapping alignments the PAT filters out local alignments. However, unlike a scoring structure the PAT is applied after alignment and therefore cannot result in the aligner exploring possible extensions into an overlap in favor of a shorter local alignment with a higher cumulative score. Therefore Stitch accepts alignments with less than 100% of the potential aligned length. Default values for the PAT were determined empirically after reviewing the degree to which filtered alignments agreed with the independent genetic maps of *Tribolium* and visual inspection.

In practice it was found useful to use two sets of alignment filters and keep alignments that passed one or both sets. The first set has a lower PAT and a higher confidence score threshold. The second set has a high PAT and a lower confidence score and is intended for lower label density regions of the genome.

Stitch: Super Scaffolding

Scaffolding alignments are next selected from the the remaining high quality alignments (i.e. more than one *in silico* genome map aligns to the same BioNano genome map). From these the best alignment is selected for each *in silico* genome map with more than one passing and scaffolding alignment. The best alignment is considered to be the longest alignment in base pairs. If alignment length is identical than the highest confidence alignment is selected. If confidence scores are identical than an alignment is chosen arbitrarily.

The gap lengths between *in silico* genome maps are inferred from scaffolding alignments and used to create a new super scaffolded sequence FASTA file and an AGP file. If gap lengths are estimated to be negative Stitch adds a 100 bp spacer gap to the sequence file and indicates that the gap is type "U" for unknown in the AGP.

Stitch only makes use of one alignment per *in silico* map per iteration. Stitch can be run iteratively so that each successive output FASTA file is *in silico* nicked

and the new *in silico* genome map is aligned to the original BioNano genome map. This alignment is inverted and used as input for the next iteration. Subsequent iterations of Stitch will make use of any *in silico* genome maps that join growing super scaffolds effectively using both sequence data and genome maps to stitch together the final super scaffolds.

Stitch: Flagging Potential Mis-assemblies

This algorithm is meant to be an intermediate refinement of draft genomes prior to further fine scale refinement at the sequence level. Inconsistencies between the BioNano genome maps and the *in silico* genome maps are reported in output logs to facilitate downstream sequence editing. If an alignment passes initial confidence score and PAT filtering but has a PAT less then 60% this is reported as a partial alignment. A partial alignment may occur if either the sequence scaffold or the BioNano genome map contained a chimeric assembly. Additionally, if a gap length is estimated to be negative it may indicate that the sequence scaffolds can be joined with a local assembly or that a chimeric sequence mis-assembly needs to be broken within a scaffold. Assembly errors in the BioNano genome maps or spurious alignments could also result in either of these cases. Ideally researchers could make use of the alignment of genomic sequence reads to the genome sequence assembly and the alignment of BioNano molecule maps to the BioNano assembled genome map to determine which assembly is likely to be incorrect.

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 (tested on a 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

Tables

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

	N50 (Mb)	Number	Cumulative Length (Mb)
Sequence scaffolds	1.16	2240	160.74
in silico genome map	1.20	223	152.53
BioNano genome map	1.35	216	200.47
Super scaffolded sequence scaffolds	4.46	2150	165.92

Figure 1 Data analysis steps. (1) Autonoise converts TIFF images of molecules to (2) BNX text files. (3) Sequence scaffolds are *in silico* labeled with fa2cmap_multi producing (4) a *in silico* genome map. (5) AssembleIryscluster uses *in silico* genome maps, BNX files and estimated genome size to (6) adjust molecule stretch and set assembly parameters. (7) Assembler produces (8) a BioNano genome map. (9) RefAligner aligns the BioNano genome map to the *in silico* genome map producing (10) an XMAP. (11) XMAP, *in silico* genome map and BioNano genome map (see arrows with dashed lines) are used by stitch to produce super scaffolded (stitched) sequence scaffolds. (13) Until no more super scaffolds are created the stitched sequence scaffolds are *in silico* labeled with fa2cmap_multi producing (14) a CMAP that is aligned to (9) the BioNano genome map and steps 10-15 are iterated. Arrows with dotted rather than dashed lines are used to as input durring iterations.

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. BioNano genome maps (blue) are shown aligned to *in silico* genome maps (green). Alignments are indicated with grey lines. CMAP orientation for *in silico* genome maps is indicated with a "+" or "-" for positive or negative orientation respectively. (A) The *in silico* genome map 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* genome map. Third alignment (unshaded) is filtered because the second alignment is the longest alignment for in silico genome map 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* genome maps scaffold BioNano genome maps as *in silico* genome map 2 does. Stitch is run iteratively all until super scaffolding alignments are found.

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

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 genome map of scaffold 81. Two XMAP alignments for *in silico* genome map of sequence scaffold 81 are shown. BioNano genome maps (blue with molecule coverage shown in dark blue) align to the *in silico* genome maps of scaffolds (green with contigs overlaid as translucent colored squares). Sequence scaffolds 79-83 were placed within ChLG 5 and sequence scaffolds 99-103 were placed with ChLG 7 by the *Tribolium* genetic map. (A) Half of the *in silico* genome map of sequence 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 sequence scaffold 81 with ChLG 7 disagrees with the genetic map and was manually rejected for Tcas5.2.

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* genome map and the BioNano genome map 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 genome map from FASTA	124.04	132.40	81
BioNano genome map	131.64	132.34	67

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

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

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.

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