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

Background: Text for this section.

Results: In this article we present software that leverages BioNano 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.

Conclusions:

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

Background

The quality of genome assemblies varies greatly. Initial assembly drafts, whether based on lower coverage Sanger or higher coverage NGS reads, are often highly fragmented. Physical maps of BAC clones are often used to validate and scaffold sequence assemblies, but the molecular, human and computational resources required to significantly improve a draft genome are not often available to researchers working on non-model organisms. The BioNano Irys platform provides affordable, high throughput physical maps with significantly higher contiguity with which to validate draft assemblies and extend scaffolds.

Data formats

The tools described make use of three file formats developed by BioNano. 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). These 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 that 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 produce nine customized assembly scripts (sections C-G Figure 2).

Molecule Stretch

In the first stage, AssembleIrysCluster adjusts molecule stretch (section C Figure 2). BioNano software operates under the assumption that imaged molecules contain 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 (Additional file 1). Sequence scaffolds are considered to be more accurate than raw BNX molecules in terms of label position. Therefore molecule maps in a BNX files are split by scan number and after alignment to the *in silico* genome map an empirical bpp value is determined for the molecule maps in each scan. The bpp indicated by this alignment is used by RefAligner to adjust molecule map bpp to 500. Once stretch has been evaluated and adjusted the split BNX files are merged into a single file (section E Figure 2).

We observed consistent patterns of empirically determined bpp between flowcells using the same flow cell model and chemistry when signal-to-noise ratios are optimal and the degree of genomic divergence between the samples used for the *in silico* genome map and molecule maps are low (Additional file 1). To identify the low quality flowcells bpp observed in alignments of scans are plotted as a QC graph (section D Figure 2).

Customization of Assembly Scripts

In the next stage, AssembleIrysCluster creates various assembly scripts to explore a range of parameters with the goal of selecting the optimal assembly for downstream analysis (sections F-G Figure 2). Molecule maps in the adjusted merged BNX file are aligned to the *in silico* genome map. An alignment error profile generated by RefAligner is used with the estimated genome length to calculate default assembly parameters, and the eight other scripts include variants of these parameters. Initially three assemblies are run, the first with $p\text{-}ValueThresholdDefault = \frac{1e-5}{GenomeLength(Mb)}$, the second with $p\text{-}ValueThresholdStrict = \frac{p\text{-}ValueThreshold}{10}$ and the third with $p\text{-}ValueThresholdRelaxed = p\text{-}ValueThreshold \times 10$ (section F Figure 2). Minimum molecule length is set to 150 kb. If one of these runs does not produce a satisfactory assembly then two minimum molecule length variants (180 kb and 100 kb) are tested with the $p\text{-}ValueThreshold$ of the current best assembly (section G Figure 2). Between three and nine assemblies are run until a satisfactory assembly is produced.

Assembly Optimization

The ultimate goal is to produce a BioNano genome map 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 the length of an ideal BioNano genome map should equal the haploid genome length. 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 redundancy to select the best assembled BioNano genome map.

Stitch: Alignment Filters

The Stitch algorithm uses alignments, or XMAPs, of query to reference genome maps to predict higher order arrangement of genome scaffolds (Figure 3). RefAligner assumes the reference genome map has the error profile of an *in silico* genome map and the query has the error profile of a BioNano genome map. Therefore alignments run with the *in silico* genome map as the reference, are 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 $False\ Positive\ p\text{-}Value$. Misaligned labels and sizing error increase the alignment $False\ Positive\ p\text{-}Value$ and decrease confidence scores [2] .

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. Stitch filters by the percent of the total possible alignment length that is aligned (section C Figure 3). To approximate scoring that favors overlapping alignments Stitch uses thresholds for minimum percent of total possible aligned length, the percent aligned threshold (PAT).

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

In practice we used two sets of alignment filters and kept alignments that passed one or both sets. The first set had a low PAT and a high confidence score threshold. The second set had a higher PAT and a lower confidence score and was intended identify longer overlaps especially in regions of the genome where label density is low.

Stitch: Super Scaffolding

Scaffolding alignments are selected from the remaining high quality alignments (i.e. more than one *in silico* genome map aligns to the same BioNano genome map (section D Figure 3)). For each *in silico* genome map with more than one high quality, scaffolding alignment the longest alignment in base pairs is selected (section D and E Figure 3). If alignment length is identical then the highest confidence alignment is selected. If confidence scores are identical then an alignment is chosen arbitrarily.

Gap lengths between *in silico* genome maps are inferred from scaffolding alignments and used to create new super scaffolds (section F Figure 3) in a new genome FASTA file and associated 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 (steps 10-15 Figure 1) such that each successive output FASTA file is converted into an *in silico* genome map and 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 (section G-H Figure 3).

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 is 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 genome map to determine which assembly is likely to be incorrect.

Results and Discussion

Datasets

High molecule weight DNA was isolated from young *Tribolium castaneum* pupae from the GA2 that was inbred 20 generations.

Using the Knickers, an *in silico* label density calculator, we estimated that the *Tribolium castaneum* genome had 8.66 nt.BspQI labels per 100 kb and 5.51nt.BbvCI labels per 100 kb. The ideal number of labels per 100 kb is between 10 and 15 therefore we dual nicked. DNA was nick, label repaired with nt.BspQI and nt.BbvCI from New England BioLabs and run in 94 flowcells on the Irys genome mapping platform.

We generated ∼250x coverage of the *Tribolium* genome for molecule maps >150 kb, the default minimum molecule map length. The 239,558 molecule maps with lengths >150kb had an N50 of 202.63 kb and a cumulative length of 50,579.12 Mb (Table 1).

Molecule map quality metrics were calculated using bnx_stats (version 1.0). Histograms of per-molecule quality metrics for maps after applying a minimum length filter of either 100 kb, 150 kb and 180kb are reported in Additional file 2.

Sequence-based assembly scaffolds are from version 5.0 of the *Tribolium castaneum* genome (Tcas5.0). Between version 3.0 [3] and version 5.0 (Table 2) of the *Tribolium* genome 1.03 Mb of sequence was added for a length of 160.74 Mb in 2,240 scaffolds after linking with Atlas-Link (version 0.01) and gap filling with GapFiller (version 1.10). Tcas5.0 was converted into an *in silico* genome map with 223 contigs, a cumulative length of 152.53 Mb (Table 2).

Assembly: selecting the optimal assembly

Molecule maps were assembled into five distinct BioNano genome maps for *Tribolium*. First, molecule maps were prepped for assembly and noise parameters were estimated using AssembleIrysCluster (version 1.0). Second, three genome maps were assembled using the relaxed, default and strict "-T" parameters (5e-08, 5e-09 and 5e-10) as well as the stretch-adjusted molecule maps and customized noise parameters. Assemblies with these p-value thresholds are named $Relaxed\text{-}T$, $Default\text{-}T$ and $Strict\text{-}T$ respectively.

The $Strict\text{-}T$ assembly was selected as the best of these three assemblies because it has a cumulative length close to 200 Mb, the estimated size of the *Tribolium* genome, and a small difference between non-redundant aligned length, breath of alignment, and total aligned length (Figure 4 and Additional file 3). The best "-T" parameter was used for two further assemblies that had relaxed minimum molecule length ($Relaxed\text{-}Minlen$) of 100 kb rather than the 150 kb default or a strict minimum molecule length ($Strict\text{-}Minlen$) of 180 kb. The $Strict\text{-}TAndStrict\text{-}Minlen$ assembly improves alignment redundancy slightly. However the cumulative length of the assembly is 21.45 Mb smaller than the estimated genome size. The $Strict\text{-}TAndRelaxed\text{-}Minlen$ assembly has a worse cumulative length and alignment redundancy than the $Strict\text{-}T$ assembly. Because neither of the assemblies using 100 or 180 kb as the minimum molecule length improved both assembly metrics when compared to the $Strict\text{-}T$ assembly, $Strict\text{-}T$ was selected as the final *Tribolium* genome map.

Assembly metrics were calculated using the BNGCompare script (version 1.0). More detailed assembly metrics for all five assembled genome maps are available in Additional file 2.

This Whole Genome Shotgun project has been deposited at DDBJ/EMBL/GenBank under the accession AAJJ00000000. The version described in this paper is version AAJJ02000000. Two scaffolds were removed from the genome assembly because they were identified as contaminants after they blasted to the *Bos frontalis* genome.

Assembly: putative haplotypes

Although overall alignment redundancy was low for the $Strict\text{-}T$ assembly, where redundant alignments of BioNano genome map contigs occurred often two contigs

aligned to the same *in silico* genome map contig. Redundancy in alignment to the *in silico* genome map contig could indicate genomic duplications in the pupae used to generate molecule maps, assembly of alternative haplotypes in the BioNano genome map contigs, mis-assembly producing redundant contigs in the BioNano genome map or collapsed repeats in the sequence-based assembly. In Figure 5, a typical redundant alignment is shown. A small region of one BioNano genome map contig aligns while the other aligns across most of the *in silico* genome map contig(section A Figure 5). Molecule map coverage dips for both BioNano map contigs in the region where only one BioNano genome map contig aligns (section A,B and C Figure 5). Taken together the lower coverage in the partially aligned region and the number of contigs aligning may indicate the assembly of two haplotypes.

Stitch: automated and manually edited assemblies

Tcas5.1 is the output of Stitch (version 1.4.4) run for five iterations with a first minimum confidence of 13 and PAT of 30 and a second minimum confidence of 8 and PAT of 90. This resulted in a greater than three fold increase in N50 (Table 2). Tcas5.1 had an additional 66 gaps estimated to have positive lengths and 26 gaps where gap length is estimated to be negative (Figure 6). All extremely small negative gap lengths were flagged for further evaluation at a sequence level. In some cases alignments suggest that a chimeric sequence scaffold may need to be broken at the sequence level and its fragments incorporated into different ChLGs. For example, half of scaffold 81 from Tcas5.0 aligns between the scaffolds 80 and 82 on ChLG5 while the other half aligns between scaffolds 102 and 103 from ChLG7 (Figure 7). In cases like this for version 5.2 the arrangement supported by the genetic map was manually selected. Joins were also manually accepted if they agree with the genetic map but the alignment quality was low. Ultimately, Tcas5.2 had a higher N50 than the automated Stitch output, 4.46 Mb (Table 2), with 66 gaps with positive estimated lengths and 24 negative length gaps.

Nearly every ChLG was less fragmented in the Tcas5.2 assembly. The number of scaffolds was reduced for each ChLG (Table 4) except for the 26 highly fragmented and unlocalized scaffolds from ChLGY. For example, ChLGX was reduced from 13 scaffolds to 2 and 5 scaffolds were reoriented in the final super scaffold (Figure 8). Several scaffolds were reordered based on alignment to the BioNano genome maps. For example, scaffold 2 from ChLGX aligned between scaffold 36 and 37 of ChLG 3 and was therefore moved in Tcas5.2 (Additional file 4). Also, 19 previously unplaced scaffolds were placed within a ChLG (Table 4).

Alignments of Tcas5.0 and Tcas5.2 *in silico* genome maps to BioNano genome maps are shown for all ChLGs in Additional file 4.

Stitch: Comparison to other software

Conclusions

Regions where BioNano genome map contigs disagree with sequence assemblies (e.g. negative gap lengths or partial alignments) have been flagged by Stitch for investigation at a sequence level. Bioinformatics tools that could automate assembly editing based on such discrepancies are needed.

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, Rscript, 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: BNGCompare.pl, bnx_stats.pl, 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_tools and https://github.com/i5K-KINBRE-script-share/BNGCompare

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

Programming language: Perl, Rscript, Bash

License: bnx_stats.pl, 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: bnx_stats.pl, 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
2. Anantharaman, T., Mishra, B.: A probabilistic analysis of false positives in optical map alignment and validation. (2001). https://cs.nyu.edu/mishra/PUBLICATIONS/01.falsepos.ps
3. Richards, S., Gibbs, R.A., Weinstock, G.M., Brown, S.J., Denell, R., Beeman, R.W., Gibbs, R., Bucher, G., Friedrich, M., Grimmelikhuijzen, C.J.e.a.: The genome of the model beetle and pest tribolium castaneum. Nature **452**, 949–955 (2008). http://dx.doi.org/10.1038/nature06784

Figures

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 during 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) One BNX file is produced for each flowcell on a chip. (C) BNX files are split by scan and aligned to the sequence reference. Stretch (bases per pixel) is recalculated for each scan 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 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.

Tables
Additional Files
Additional file 1 — Molecule stretch per scan in recent flowcells.
Bases per pixel (bpp) is ploted for scans $1..n$ for each flowcell of mouse lemur molecules (purple). The first scan of each flowcell is indicated with a grey dashed line. The pre-adjusted molecule map stretch was determined by

Figure 4 Comparison of the BioNano genome maps assembled from the Tribolium data using five sets of parameters. Relaxed, default and strict "-T" parameters were set to 5e-08, 5e-09 and 5e-10. Relaxed, default and strict minimum molecule length were set to 100, 150 and 180 kb.

Figure 5 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 6 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 7 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 1 BioNano molecule maps from *Tribolium* filtered by minimum length. Molecule map N50, cumulative length and number of maps are listed for all three molecule length filters for the *Tribolium* genome data.

Minimum molecule map length (kb)	Molecule map N50 (kb)	Cumulative length (Mb)	Number of molecule maps
100	165.35	82,738.71	503,414
150	202.64	50,579.12	239,558
180	232.57	34287.15	139,949

Table 2 *Tribolium* assembly Results. Assembly metrics for Tcas5.0 (the starting sequence scaffolds), the *in silico* genome map, the BioNano genome map of assembled molecules, the automated output of Stitch (Tcas5.1) and the manually currated sequence assembly (Tcas5.2) for the *Tribolium* genome.

	N50 (Mb)	Number	Cumulative Length (Mb)
Tcas5.0 sequence scaffolds	1.16	2240	160.74
in silico genome map	1.20	223	152.53
BioNano genome map	1.35	216	200.47
Tcas5.1 sequence scaffolds	3.85	2148	165.72
Tcas5.2 sequence scaffolds	4.46	2150	165.92

Table 3 Alignment of BioNano assembly to the *in silico* genome map of Tcas5.0. 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	124.04	132.40	81
BioNano genome map	131.64	132.34	67

aligning molecules to the *in silico* genome map. Data made available by P.A. Larsen, J. Rogers, A.D. Yoder and the Duke Lemur Center.

Additional file 2 — Molecule map metrics and histograms from *Tribolium* DNA

Detailed metrics for molecule maps including map N50, cumulative length and number of maps. Figures show histograms of per molecule map quality metrics including length, molecule map SNR and intensity, label count, label SNR and label intensity. Molecule maps are filter for minimum molecule lengths of 100, 150 or 180 kb.

Table 4 *Tribolium* chromosome linkage groups before and after super scaffolding. The number of sequence scaffolds in the ordered 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 scaffolds	Unplaced scaffolds added in Tcas5.2	Tcas5.2 super scaffolds
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	10
Total	157	19	77

Additional file 3 — Assembly of *Tribolium* genome maps with range of parameters
Detailed assembly metrics for assembled genome maps using strict, default and relaxed "-T" parameter, p-value threshold are named Relaxed-T, Default-T and Strict-T respectively. The best "-T" parameter was used for two additional assemblies with either relaxed minimum molecule length (relaxed-minlen) of 100 kb, rather than the 150 kb default, or a strict minimum molecule length (strict-minlen) of 180 kb.

Additional file 4 — ChLGs before and after super scaffolding
Alignments of Tcas5.0 and Tcas5.2 *in silico* genome maps to BioNano genome maps for all ChLGs. BioNano genome maps (blue with molecule coverage shown in dark blue) aligned to the *in silico* genome maps (green with contigs overlaid as translucent colored squares). Alignment to both Tcas5.2 super scaffolds (top alignment) and Tcas5.0 scaffolds (bottom alignment) are shown.