Bionano Solve Theory of Operation: Hybrid Scaffold

Document Number: 30073
Document Revision: F

Table of Contents

Legal Notice

For Research Use Only. Not for use in diagnostic procedures.

Introduction

Having high-quality assemblies is essential for the study of genomes. Next-generation sequencing (NGS) short-read data provide limited long-range information, which is necessary for assembly of complex genomes. The incorporation of long-read sequencing data can help, but the resulting assemblies may still be fragmented and contain errors. Segmental duplications that span hundreds of kilobasepairs often remain unresolved.

Bionano genome mapping takes advantage of DNA molecules of hundreds of kilobasepairs to assemble long genome maps. These maps can be combined with sequencing assemblies to produce ultra-long hybrid scaffolds that represent the structure of the chromosomes (Figure 1). The genome maps can validate, order and orient sequence fragments, identify potential chimeric joins in the sequence assembly, and help estimate the gap sizes between adjacent sequences.

Genome maps are compatible with assemblies from any sequencing platform, provided that the input sequence assemblies are of sufficient quality and contiguity. Typically, input assemblies with contig/scaffold sizes of at least 100 kbp are sufficient to produce high-quality hybrid scaffolds. Scaffolding with less contiguous assemblies often produces satisfactory results; however, this is dependent on several factors, including label coverage across the genome and assembly quality.

The Hybrid Scaffold pipeline automates the comprehensive scaffolding process and is consisted of five major steps: 1) generate *in silico* maps for sequence assembly; 2) align *in silico* sequence maps against Bionano genome maps to identify and resolve potential conflicts in either data set; 3) merge the non-conflicting maps into hybrid scaffolds; 4) align sequence maps to the hybrid scaffolds; and 5) generate AGP and FASTA files for the scaffolds.

The pipeline consists of several scripts, which are streamlined by a wrapper script to allow for automated execution. The pipeline is fully integrated with Bionano Access®, which provides a convenient interface for users interested in running Hybrid Scaffold and viewing scaffolding results.

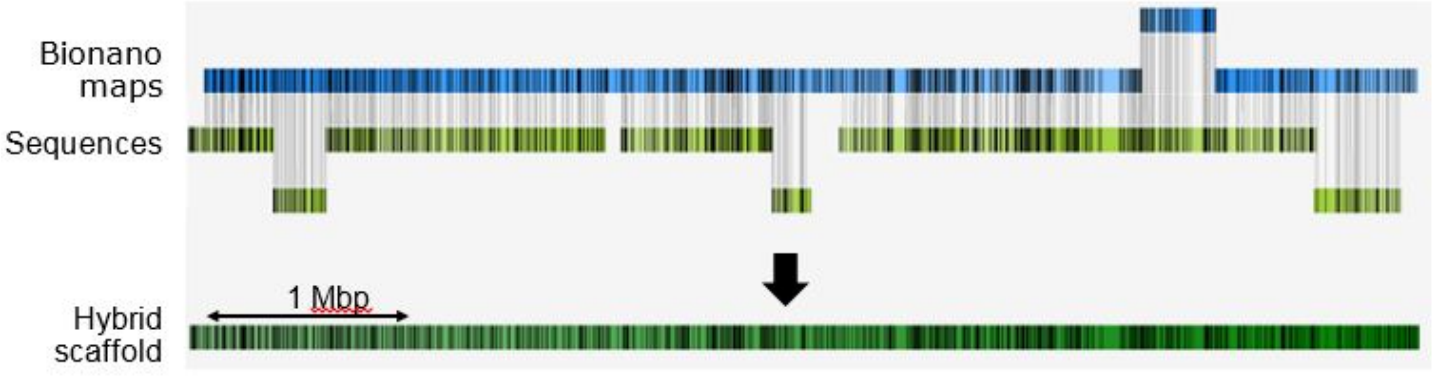


Figure 1. The concept of combining Bionano map assembly and sequence assembly to generate long-range hybrid scaffolds.

Contiguity and Completeness

Using the Hybrid Scaffold pipeline, we see significant improvements in contiguity (N50) across genomes, due to the complementary nature of Bionano and sequence data. The pipeline is compatible with both the Nick Label Repair and Stain (NLRS) Direct Label and Stain (DLS) chemistries. The DLS chemistry provides further unprecedented contiguity compared to the NLRS chemistry (Figure 2).

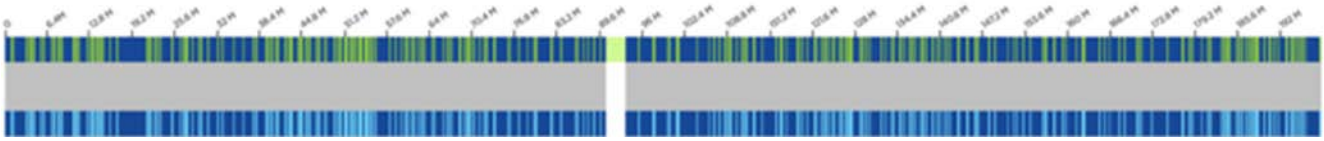


Figure 2. Complete chromosome arms of human chromosome 3 assembled with DLE-1. The reference is in the top track (green); the hybrid scaffolds are in the bottom track (blue).

The DLS chemistry does not introduce undesirable breaks in the DNA and allows us to create contiguous maps that span complex regions in a genome. The higher label density from DLE-1 enables us to scaffold shorter sequence contigs. Shown in Figure 3, starting with NGS assemblies of the human NA12878 genome with N50s from 0.08 – 0.9 Mbp, we produced hybrid scaffolds with N50s from 50 to 80 Mbp, an improvement in contiguity of up to 1000X. Chromosome-arm length scaffolds were assembled in 20 out of 23 chromosomes (Figure 2), and alignments showed that they were consistent with the hg19 reference. The hybrid scaffolds incorporated 80-90% of total NGS sequences with over 99% scaffold accuracy (defined as the percentage of NGS contigs ordered correctly in the scaffold with respect to the hg19 reference). DLS is compatible with a vast array of organisms. In Figure 4, we show results from scaffolding PacBio sequence assemblies of one plant and two animal genomes. The final scaffolds improved the contiguity from 16-100X over the input sequence assemblies, while incorporating more than 95% of the sequence. We recommend running Hybrid Scaffold using DLE-specific parameters (for example, hybridScaffold_DLE1_config.xml and hybridScaffold_two_enzymes_DLE1.xml).

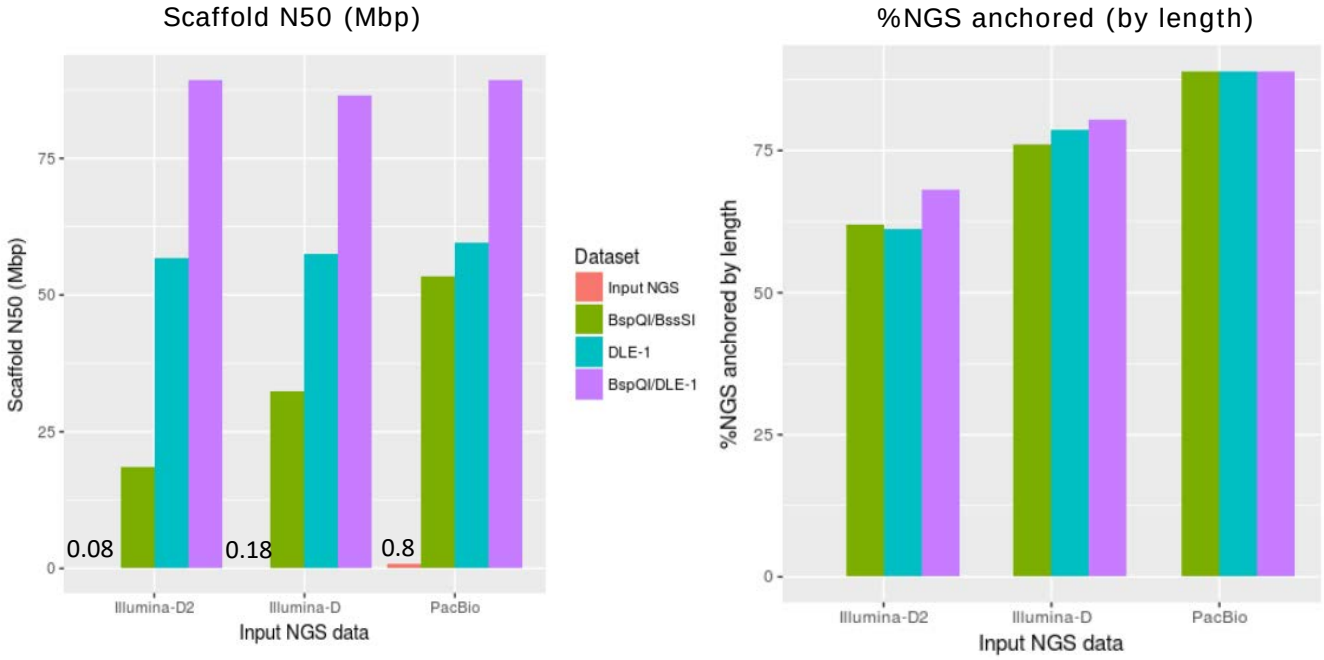


Figure 3. Hybrid scaffolds of sequence assemblies with DLE-1 Bionano maps for NA12878.

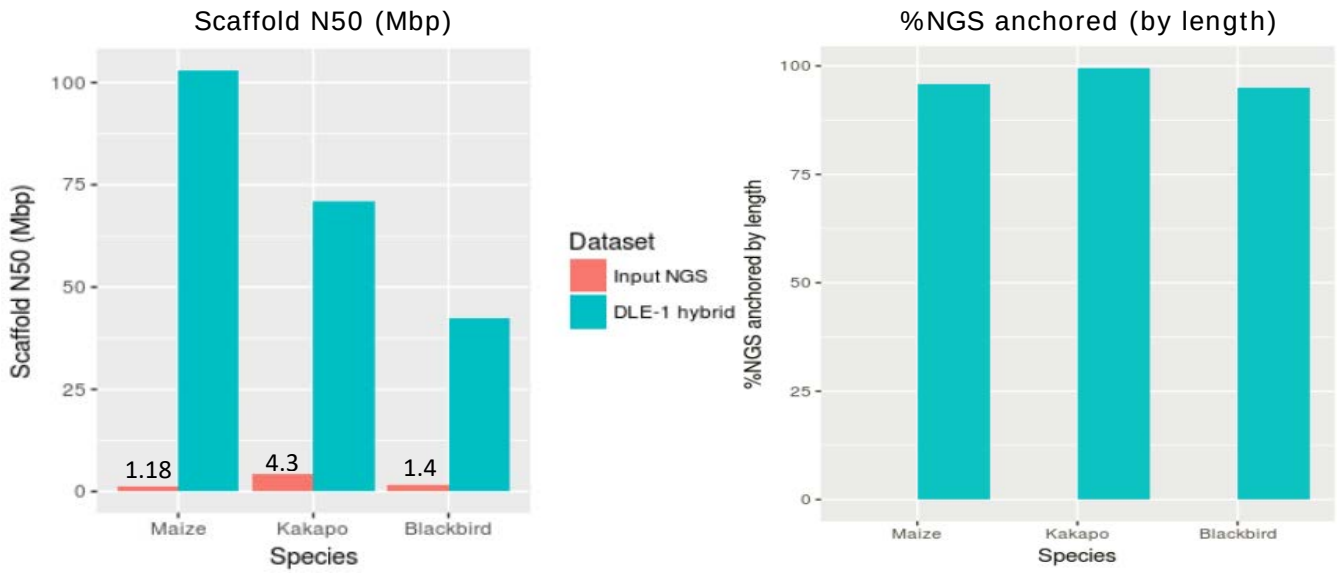


Figure 4 Hybrid scaffolds of sequence assemblies from maize, kakapo, and blackbird with DLE-1 Bionano maps.

The NLRS data are shown in Table 1. As much as 9.69-fold improvement in N50 for three large size genomes (human, goat, and maize) was observed. When a more aggressive parameter set was used in merging, a 13.33-fold improvement in N50 was observed. Compared to the default parameter set, the aggressive parameter set uses lower stringency cutoffs (merge P-value and minimum length of pairwise alignments) in aligning the

sequences to the genome maps during the merge step. See the Suggested Parameters section for a detailed explanation. The enzyme Nt.BspQI was used in these experiments.

Moreover, the majority of the available input sequence was anchored in the scaffolds; over 84% of the total length of the sequence and over 93.6% of the genome maps was incorporated (Table 2). The remaining unused sequences were too short to be anchored.

Table 1. Contiguity and genome coverage of hybrid scaffolds with the Nt.BspQI enzyme.

	Species	Sequence N50 (Mbp)	Bionano Map N50 (Mbp)	Hybrid Scaffold Contiguity		Hybrid Scaffold Coverage	
				Hybrid Scaffold N50 (Mbp)	N50 Fold Increase	Hybrid Scaffold Size (Mbp)	% of Known Reference Assembly
Default Parameters	Human NA12878	0.90	3.92	8.72	9.69	2,833.44	91.7
	Goat	4.68	1.59	17.12	3.66	2,634.83	104.3
	Maize	1.04	2.47	6.43	6.18	2,128.56	103.0
Aggressive Parameters	Human NA12878	0.90	3.92	11.94	13.33	2835.183	91.8
	Goat	4.68	1.59	23.85	5.10	2,643.93	104.7
	Maize	1.04	2.47	10.00	9.63	2,119.64	102.6

Table 2. Usage of input assemblies in scaffolding with the Nt.BspQI enzyme.

	Species	Amount of Sequence Data Utilized in Hybrid Scaffold (Mbp)	Amount of Bionano Data Utilized in Hybrid Scaffold (Mbp)
Default Parameter	Human NA12878	2,576 (84.03%)	2,804.64 (98.07%)
	Goat	2,498 (95.13%)	2,572.30 (93.60%)
	Maize	2,008 (95.42%)	2,079.08 (98.11%)

Assembly Conflicts and Resolution

Besides creating long and contiguous scaffolds, the Hybrid Scaffold pipeline also detects and resolves chimeric joins present in either input assembly. Chimeric joins may be formed when short reads, molecules, or paired-end inserts are unable to span across long DNA repeats. These errors would appear as conflicting junctions in the alignment between the two assemblies. See the top of Figure 5 for an example of a conflict. Upon the detection of a conflict, the pipeline uses Bionano's long native molecules to determine which assembly has been likely constructed incorrectly. If the genome map does not have long molecule support at the conflict junction, then the map is cut, thus removing the putative chimeric join. If it does have molecule support, the sequence fragment is

cut. Figure 5 shows that the genome map has strong molecule support, and so, the sequence is cut by the pipeline. When the sequence is aligned to a reference assembly, it aligns to two different chromosomes (Figure 5, bottom). In this case, the sequence is indeed likely misassembled. Importantly, one should note that to identify and resolve conflicts, both assemblies need to have coverage spanning both sides of a chimeric join, please see the section Step 2 Identification and Resolution of Conflicting Alignments for detail.

The accuracy of the cuts in resolving conflicts is high (Table 3). Our data on the three genomes shown in Table 1 shows that the majority of the cuts can be confirmed by comparing with the reference assembly available for the species. Note that some of the cuts could not be confirmed simply because the reference assembly used was incomplete. Moreover, the two input assemblies may capture different alleles, and that chimeric joins may have been caused by very long segmental duplications whose lengths are longer than the Bionano molecules used to evaluate the conflicts.

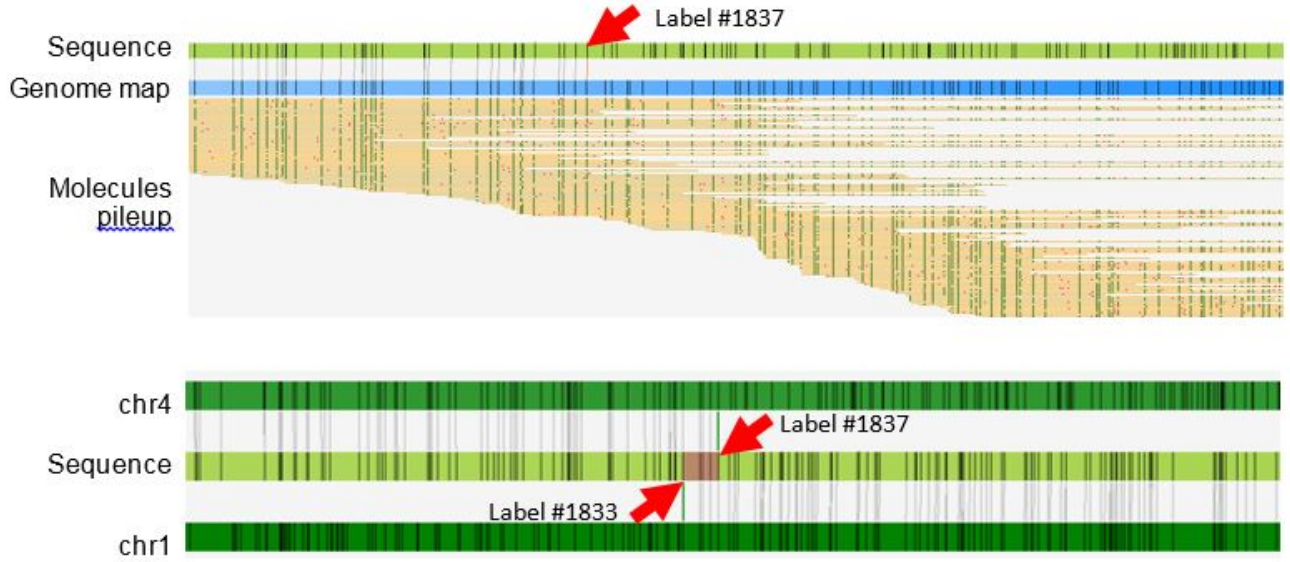


Figure 5. Example of a conflict between a sequence contig and a Bionano map in a human sample. (Top) The conflict junction as shown by the red arrow in the alignment between the sequence contig and the genome map. There is strong molecule support spanning the junction region on the genome map, so the sequence is cut at the label indicated. (Bottom) Alignment between the sequence contig and the reference assembly. The chimeric join on the sequence is confirmed as the conflict locus also displays alignments to two different chromosomes.

Table 3. The number of cuts performed by the Hybrid Scaffold pipeline to the assemblies and the number of cuts confirmed when the sequence contigs were aligned to the corresponding references of that species. Note that some of the cuts could not be confirmed simply because the reference assembly used was incomplete.

	Species	# Of Cuts On Sequence Confirmed / Total	# Of Cuts On Bionano Confirmed / Total
Default Parameters	Human NA12878	4 / 6 (67%)	1 / 1 (100%)
	Goat	66 / 79 (84%)	11 / 16 (69%)
	Maize	24 / 26 (92%)	12 / 13 (92%)

To enable inspection of the conflict-resolution results, the Hybrid Scaffold pipeline annotates the cuts in BED files, which can be displayed in Bionano Access in conjunction with the alignments between the sequence and the Bionano maps. Moreover, the pipeline delineates the IDs and the coordinates of the sequences and maps where conflicts have been detected and the corresponding resolution approach taken. Most importantly, this file can be edited by the user and re-input into the Hybrid Scaffold pipeline for a rerun based on the manual conflict resolution strategy, producing a new set of hybrid scaffolds. This manual enhancement functionality can be performed multiple times, thus enabling users to have fine control in generating high quality and complete scaffolds.

Important Note About Hi-C Data

For most input NGS data, the automatic conflict resolution is effective at identifying and resolving most large-scale NGS assembly errors. For scaffolding Hi-C data, we recommend using hybridScaffold_DLE1_HiC_config.xml and hybridScaffold_two_enzymes_HiC.xml for single- and two-enzyme hybrid scaffolding. Still, conflict resolution may fail to resolve all conflicts introduced by Hi-C data. Hi-C data creates significant order/orientation issues that the automatic conflict resolution pipeline is not optimized for. In order to reduce the number of false joins by Hi-C, we recommend users to first scaffold NGS data with Bionano's map to create high quality hybrid scaffolds, and then perform a second round of scaffolding between the hybrid scaffolds with Hi-C. In general, when combining with Hi-C data, additional manual curation of the scaffolds is recommended. These configurations may also be used when scaffolding sequences assemblies with high numbers of chimeric joins and orientation errors.

Further Improving Contiguity and Completeness Using Two-Enzyme Hybrid Scaffolding

Assembly contiguity can be further improved by performing hybrid scaffolding with maps using two separate enzymes. We first generated two sets of Bionano maps, each with a different enzyme, and then applied novel algorithms that use the NGS sequences as a bridge to merge the single-enzyme Bionano maps into combined maps.

Since two sets of Bionano maps were generated independently, they provide complementary evidence to detect and correct assembly errors. Bionano maps generated using nicking enzymes also tend to break at "fragile" sites where two nicking sites are very close to each other. However, maps generated with different nicking enzymes would break at different genomic locations, so the two sets of maps can compensate for one another when combined, resulting in significantly improved contiguity. The final merged map also contains motif patterns from both enzymes, effectively doubling the information density. The increased density allows us to anchor shorter sequence contigs in the final scaffolds and greatly expands the range of sequence data that can be integrated with Bionano data.

The two-enzyme approach was validated on the human NA12878 genome, a model data set for which sequence data is publicly available. Three different sequence assemblies were tested: 1) Illumina-D, 51X of 250 bp pair-end

sequence, 2) Illumina-S, 40X of 101 bp pair-end and 25X of 2.5-kbp mate-pair sequence, and 3) PacBio, 46X with mean read length of 3.6 kbp. These assemblies were relatively fragmented (with contig N50 of 0.18 Mbp, 0.50 Mbp and 0.90 Mbp, respectively). Using the two-enzyme approach, the contiguity of scaffolds improved 3-fold (up to 100-fold when compared to input NGS, Figure 6) and the amount of sequence anchored increased by up to 44.2% (Table 3) in the final scaffolds. The pipeline was also robust for animal and plant genomes as well (Figure 7 and Table 4). Overall, this approach allows us to produce highly accurate and contiguous assemblies for complex genomes.

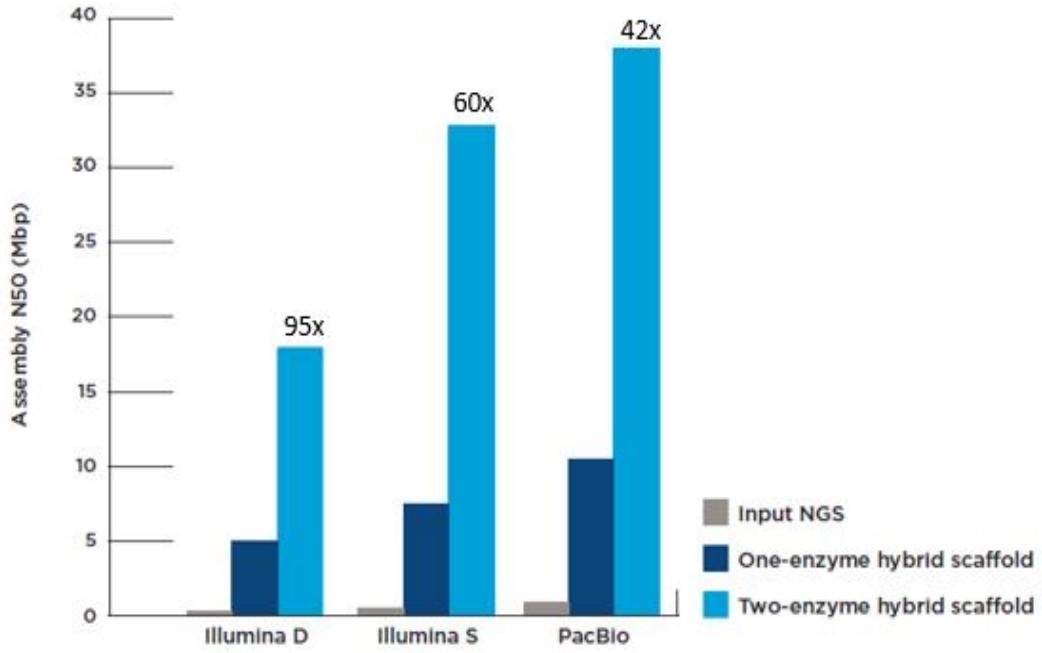


Figure 6. Improvements in NA12878 assembly contiguity after hybrid scaffold with one-enzyme and two-enzyme genome maps. The numbers at top of the bars indicate fold increase in N50 over the input NGS assemblies. Illumina-D: 51X of 250 bp pair-end sequence; Illumina-S: 40X of 101 bp pair-end and 25X of 2.5 kbp mate-pair sequence; PacBio: 46X with mean read length of 3.6 kbp.

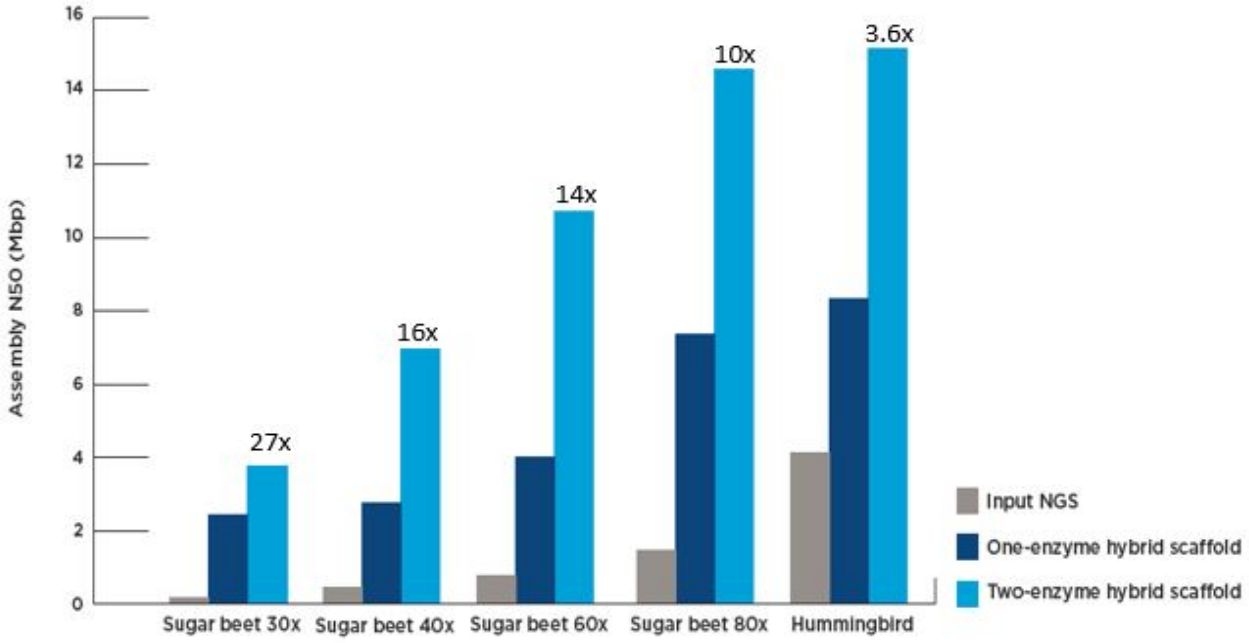


Figure 7. Improvements in sugar beet and hummingbird assembly contiguity after hybrid scaffolding with Bionano genome maps using one-enzyme and two-enzymes. For sugar beet, the fold coverage of the PacBio de novo assemblies is shown. The numbers at the top of the bars indicate fold increase in N50 over the input NGS assemblies.

Table 4. Improvement in usage of sequence contigs in one-enzyme and two-enzyme scaffolds for different genomes. *Only sequence contigs longer than 3 kbp were counted.

Genome	NGS	Contigs in input NGS*	NGS Contigs Incorporated in Scaffolds			Total Length of NGS (Mbp)	Total Length of NGS in Scaffolds (Mbp)		
			Single-Enzyme	Two-Enzyme	Percent Improvement		Single-Enzyme	Two-Enzyme	Percent Improvement
Human NA12878	Illumina-D	34349	8477	12223	44.2%	3068	2081	2340	12.4%
	Illumina-S	14045	5498	6181	12.4%	2936	2558	2619	2.4%
	PacBio	24124	3925	4387	11.8%	3065	2665	2704	1.5%
Sugar beet	PacBio 30X	3137	744	1586	113%	350	137	228	66.9%
	PacBio 40X	2096	1209	1688	39.6%	496	398	453	13.7%
	PacBio 60X	1325	904	1071	18.5%	539	491	511	4.1%
	PacBio 80X	938	606	687	13.4%	563	526	539	2.3%
Hummingbird	Illumina	2039	488	514	5.3%	1106	1023	1029	0.5%

General Information

Coverage

For Hybrid Scaffold, we recommend using as input a minimum of 80X effective molecule coverage (as presented in the MQR report) in order to build an accurate and contiguous consensus genome map assembly for each enzyme. When using nickases, using more coverage does not significantly improve map contiguity. When using a DLS enzyme such as DLE-1, effective coverage up to and beyond 100X has shown improved map contiguities for some plants and animals.

Input Bionano assembly

When running the *de novo* assembly pipeline for hybrid scaffolding applications, users are recommended to use assembly parameters for non-haplotype-aware assembly. The current Hybrid Scaffold pipeline does not explicitly handle haplotype information and assumes there is only one genome map or NGS sequence contig covering a given genomic region. If multiple haplotypes are present, the pipeline may make false positive conflict cuts and incorrectly mix haplotypes in the final scaffolds. We understand that haplotype information is important in many applications, and a fully haplotype-aware Hybrid Scaffold pipeline is in our roadmap for a future release.

Runtime

Table 5. Example runtime data of completed Hybrid Scaffold runs starting from raw molecule bnx files. *The two-enzyme workflow runtime assumes a compute server configuration where two assemblies can be run in parallel.

Genome	NGS dataset	Runtime on Saphyr and Bionano Compute Servers (hrs)		
		Assembly	Hybrid Scaffold	Total Time
Human NA12878 70X filtered (50X effective) *One-enzyme workflow*	Illumina-D	BspQI: 9.14	2.41	11.55
	Illumina-S		1.47	10.61
	PacBio		1.35	10.49
Human NA12878 70X filtered (50X effective) *Two-enzyme workflow*	Illumina-D	BspQI: 9.14 BssSI: 14.34	3.30	17.64*
	Illumina-S		2.41	16.75*
	PacBio		2.06	16.40*
Hummingbird *(50X effective)* *One-enzyme workflow*	Illumina	BspQI: 13.80	0.35	14.15
Hummingbird *(50X effective)* *Two-enzyme workflow*	Illumina	BspQI: 13.80 BssSI: 3.10	0.62	14.42*

Table 6. Example runtime data of complete Hybrid Scaffold runs with DLE-1 starting from raw molecule bnx files. *The two-enzyme workflow runtime assumes a compute server configuration where two assembly pipelines can be ran in parallel.

Genome	NGS dataset	Runtime on Saphyr and Bionano Compute Servers (hrs)		
		Assembly	Hybrid Scaffold	Total Time
Human NA12878 70X filtered (50X effective) *One-enzyme workflow*	Illumina-D	DLE-1:16.21	3.85	20.06
Human NA12878 70X filtered (50X effective) *Two-enzyme workflow*	Illumina-D	DLE-1:16.21 BspQI:9.14	6.35	22.56*

Workflow and Program Files

Section I. Single-enzyme workflow

The Hybrid Scaffold pipeline consists of a package and auxiliary modules implemented in Perl. It also requires RefAligner, which performs a variety of functions such as alignments, refinement and merging of maps. The descriptions of each step of the Hybrid Scaffold pipeline and the corresponding script files are provided in this document. See Figure 8 for an illustration of the workflow.

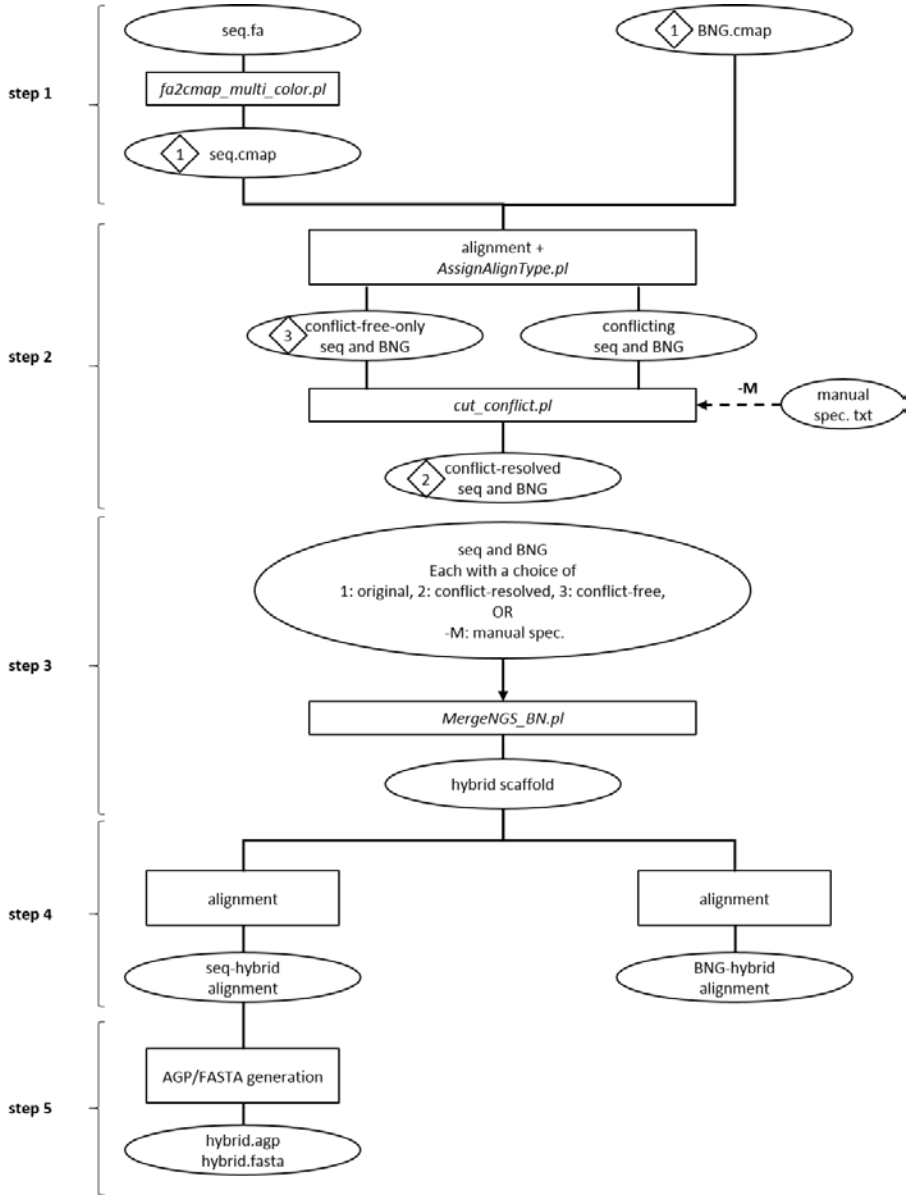


Figure 8. Workflow of the Hybrid Scaffold pipeline. Oval boxes represent entities, such as assembly. Rectangular boxes represent scripts or processes. Diamond boxes represent specific assemblies that are used as input to merge in MergeNGS_BN.pl. Step 1 performs conversion of the sequence assembly from FASTA format to CMAP format. Step 2 identifies and resolves conflicts. Step 3 merges the two assemblies into hybrid scaffolds. Step 4 aligns the sequences to the scaffolds, and it also aligns the genome maps to the scaffolds. Step 5 generates AGP and FASTA files representing the hybrid scaffolds.

Controller

The file hybridScaffold.pl is a wrapper script that streamlines the hybrid scaffolding process. It accepts input files, prints runtime messages, generates output files, and displays summary statistics of results. The wrapper script takes as input a) sequence assembly in FASTA format, b) Bionano genome map assembly in CMAP format, c) a configuration XML file containing the run parameters, and d) the RefAligner program. It generates output files to a specified output directory.

STEP 1: CONVERSION OF FASTA FILES TO CMAP FILES

The Perl script fa2cmap_multi_color.pl generates an *in silico* map of the sequence data. The input is a FASTA sequence assembly file. The script identifies motif sites in the sequence and outputs their coordinates for sequence contigs that are of a minimal length and a minimal number of labels, both of which are specified in the XML configuration file (see below). An enzyme name or recognition sequence is required (note that this script also enables users to specify multiple enzymes for *in silico* digest, for example BspQI and BbvCI, in the XML configuration file.). This script generates a key file that denotes the translation of FASTA sequence identification to CMAP identification. The output files – the CMAP file and the key file – are written to a sub-directory called fa2cmap/. This Perl script also outputs summary statistics of the sequence data to a log file.

STEP 2: IDENTIFICATION AND RESOLUTION OF CONFLICTING ALIGNMENTS

The second step of the Hybrid Scaffold pipeline identifies and resolves conflicts between the sequence and Bionano data in order to prevent them from propagating. Conflicts may be due to genuine allelic differences or assembly errors. They feature an excessive number of unaligned labels in the alignment (Figure 9).

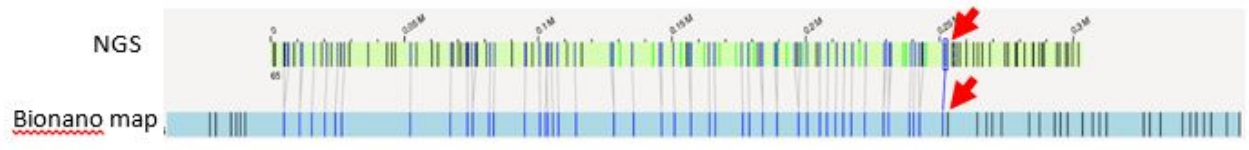


Figure 9. Example conflict between sequence and Bionano map. A significant number of unaligned labels outside the aligned region (left of the red arrows) indicate the presence of conflict between the two assemblies. The number of unaligned labels can be specified in the XML configuration file (see main text).

The Hybrid Scaffold pipeline first uses RefAligner to align *in silico* digested sequence maps with Bionano genome maps, and then calls the AssignAlignType.pl script to identify conflict junctions. The inputs to this script include XMAP and CMAP files generated from the alignment, the original sequences, and original Bionano genome maps. The script counts the number unaligned labels in each alignment, and the maximum number of unaligned labels tolerable can be specified in the assignAlignType.max_overhang field in the XML configuration file. Finally, the script outputs a list of alignments where conflicts have been found (assignAlignType.xmap). It also outputs two CMAP files that contain the remaining non-problematic sequences (assignAlignType_r.cmap) and genome maps (assignAlignType_q.cmap). Importantly, the Hybrid Scaffold pipeline outputs the file conflicts.txt that

delineates the conflict junctions. The detailed format of the conflict file is described in Bionano document 30166.

Note that the output of the alignment of sequence with Bionano maps resides in the align0/ and align1/ sub-directories, and the output of conflict-detection resides in the assignAlignType/ sub-directory. The file conflicts.txt is an important file that shows the coordinates of where the sequence-to-map alignments stop prematurely. In the Figure 9 example, suppose that the IDs of the sequence and genome map are 18616 and 1210, respectively, the conflicts.txt file would look like the following table:

Table 7. An example conflict file entry showing the locations of the conflicts. See Figure 8 for visual guidance. Each conflicting alignment is represented by a row, but to fit the page width here, it is shown in two rows. Sequence is treated as the reference, while the genome map, the query. Here, a conflict is detected to the left of the alignment, so the left breakpoint columns show the last aligned position in both the sequence and the genome map. There is no conflict to the right of the alignment, so the right breakpoint columns are -1.

xMapId	refQry	refId	leftRefBkpt	rightRefBkpt	alignmentOrientation
4273	ref	18616	505984.0	-1	+

refQry	qryId	leftQryBkpt	rightQryBkpt	alignmentOrientation
qry	1210	268750	-1	+

After identifying discrepancies, the Hybrid Scaffold pipeline attempts to resolve the discrepancies using the script cut_conflicts.pl, which examines the conflicting loci as indicated by conflicts.txt. Specifically, it examines the molecule coverage and the chimeric quality scores surrounding the conflicting label on the genome map (e.g., the bottom red arrow in Figure 9) for any evidence of misassembly. The chimeric quality score of a label represents the percentage of Bionano molecules that can align to the genome map fully to the left and to the right of that label. A label on a genome map with a high score would indicate that its vicinity was assembled correctly, thus unlikely to be a chimeric join. Each label is assigned a score, and the scores are typically output as a column in the genome map CMAP file by RefAligner during *de novo* assembly.

During hybrid scaffolding, if the chimeric quality scores surrounding the conflicting locus of a Bionano genome map are lower than a specified threshold, the cut_conflicts.pl script cuts the genome map at the conflict locus. If the scores are higher than the threshold, it cuts the corresponding region of the sequence. The score threshold (ranges from 0% to 100%) can be specified by the cut_conflicts.min_quality_score_threshold field in the XML configuration file. The script requires the presence of the ChimQuality column in the Bionano CMAP file. See Chimeric Score Generation below in case chimeric quality scores are absent.

The cut_conflicts.pl file outputs conflicts_cut_status.txt file (Table 8), which has a similar format as conflicts.txt, but with additional columns indicating whether cuts occurred (see document 30166). Continuing on the conflict example above, suppose that the labels surrounding 268,750 on genome map 1210 have high chimeric quality scores, then the sequence 18616 would be cut into two pieces: one to the left of 505,984 and one to the right.

Table 8. The conflicts_cut_status.txt file describes the status of the breakpoints. It indicates whether the sequence (ref) or the genome map (qry) should be cut at the conflicting locus. Here, the sequence is cut into two pieces while the genome map remains unchanged.

xMapId	refQry	refId	leftRefBkpt	rightRefBkpt	alignmentOrientation	ref_leftBkpt_toCut	ref_RightBkpt_toCut	ref_toDiscard
4273	ref	18616	505984.0	-1	+	cut	okay	okay

refQry	qryId	leftQryBkpt	rightQryBkpt	alignmentOrientation	qry_leftBkpt_toCut	qry_rightBkpt_toCut	qry_toDiscard
qry	1210	268750.0	-1	+	okay	okay	okay

Note that this file can be manually edited, allowing users to fine-tune how conflicts are resolved. The cut_conflicts.pl script outputs the conflict-resolved sequence and genome map CMAP files. Other potentially useful files are auto_cut_NGS_coord_translation.txt and auto_cut_BN_coord_translation.txt; they detail how a sequence or map is cut. BED files detailing the location of the cuts are produced too, and they are loaded into Bionano Access for display. These files are contained in the assignAlignType/cut_conflicts/ sub-directory.

STEP 3: MAP MERGING

Merging of sequence-derived maps with Bionano maps is performed by the MergeNGS_BN.pl script, which calls RefAligner to perform iterative pairwise merging. The input of the sequences or genome maps can be from one of the following sources:

1. Original assembly (the original input but may contain chimeric joins)
2. Conflict-resolved assembly (result from cut_conflicts.pl)
3. Conflict-free assembly (result from AssignAlignType.pl)

Option 1 performs pairwise merging between the original sequences or genome maps. Option 2 performs merging on the conflict-resolved ones; and Option 3 selects only the non-conflicting maps as input to the merge. One can a select different option values for sequence and genome maps; see Figure 7 and the How to Run the Hybrid Scaffold Pipeline section below.

Note that Step 2 (identification and resolution of conflicting alignments) is always performed. In other words, there are always output files in the assignAlignType/ and assignAlignType/cut-conflicts/ subdirectories when running the Hybrid Scaffold pipeline. For example, users may have selected -N 1 -B 1 (the original input assemblies) options when running the Hybrid Scaffold pipeline; however, conflict-identification and resolution are performed regardless. The pipeline simply ignores those results during this map-merging step, and it merges only the original input assemblies as indicated by the users.

Furthermore, if one runs the Hybrid Scaffold pipeline using a custom resolution file (-M option), assemblies are cut according to the file, and they are then passed as inputs to the MergeNGS_BN.pl. See Figure 7 and the section

Manual Conflict Resolution below.

The MergeNGS_BN.pl script generates CMAP files in the mergeNGS_BN/ output sub-directory. Particularly, it outputs step2.hybrid.cmap, which contains the scaffolds.

Handling of complex multi-path regions (CPMRs)

Most of the multi-path regions in a complex genome can be correctly assembled in Bionano maps because the underlying ultra-long molecules can span across them. However, there remain regions such as those from very large segmental duplications that are difficult to span across and can potentially result in chimeric assembly during hybrid scaffolding, and they require special handling. In the *de novo* assembly pipeline, a pairwise alignment between all consensus maps is performed, and large shared regions between two maps are detected. These are candidate complex multi-path regions and are marked in the final consensus map from the assembly pipeline (see Bionano Solve Theory of Operation, Structural Variant Calling PN# 30110 for more detail). During hybrid scaffolding, we check for these regions. A merge is not allowed between a Bionano map and a sequence contig if the sequence is only aligned to the marked regions of the Bionano map (see Figure 10a). A merge is allowed if the sequence contig spans a sufficiently large unique region (>50 kbp) of the Bionano map (Figure 10b).

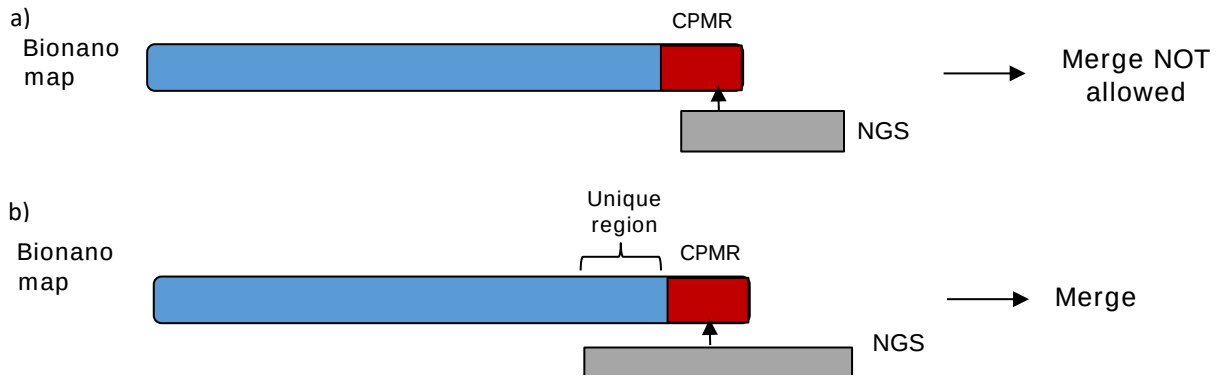


Figure 10. A schematic illustrating the handling of CPMRs during Hybrid Scaffold.

STEP 4: SEQUENCE- AND BIONANO-HYBRID ALIGNMENTS

In the forth step, RefAligner aligns conflict-resolved sequences and Bionano maps (output from Step 2) to the hybrid scaffolds. This enables users to visualize how the sequence data and Bionano maps contributed to the hybrid scaffolds.

A two-stage alignment procedure is implemented to maximize both speed and sensitivity. During the first step, we use alignment parameters optimized for speed so large sequence contigs can be quickly aligned to the hybrid scaffolds. Then, all unaligned contigs were aligned to hybrid scaffolds a second time using alignment parameters aim optimized for sensitivity. This enables shorter NGS contigs to be anchored and improves the sequence completeness of the final scaffold. These are output to alignment files (xmap, _r.cmap representing the hybrid

scaffolds, and _q.cmap representing the sequences) containing the suffixes “1st_pass” and “2nd_pass” respectively in the align_final/ directory.

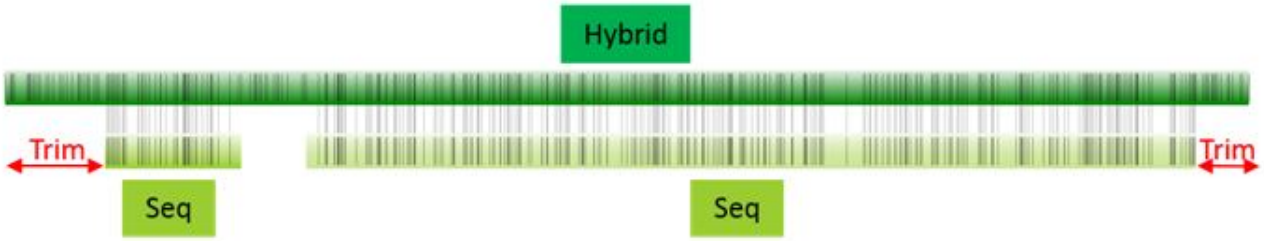
The alignment data are used in the subsequent generation of the AGP and FASTA output. Similarly, alignment is done between the hybrid scaffolds and the Bionano genome maps.

STEP 5: AGP/FASTA FILE GENERATION

The pipeline provides AGP and FASTA representations of the hybrid scaffolds. The AGP output includes information about how sequences are scaffolded. Details about the AGP v2.0 specification can be found in the National Center for Biotechnology Information website (https://www.ncbi.nlm.nih.gov/assembly/agp/AGP_Specification/). The sequence IDs in the AGP file correspond to the original input sequence FASTA IDs. However, there may be name changes to conflict-resolved sequences from Step 2 above; their names would contain the original IDs and the “subseq” keyword. Conflict-resolved NGS contigs are stored in the FASTA file with an extension “cut.fasta”. To be compliant with NCBI requirements, the left and right ends of the scaffolds begin with non-N sequences. The Hybrid Scaffold pipeline trims the scaffolds such that they always begin and end with defined sequences (Figure 11). In other words, if the termini of a scaffold were constructed by genome maps, they would be trimmed until the leftmost and the rightmost sequence.

Neighboring sequence contigs in the same hybrid scaffold are usually separated by gaps whose sizes are estimated from the alignment of sequences to hybrids from Step 4. When two sequence contigs are inferred to, we insert a small N-base gap of *G* bp in the AGP and FASTA export. If the gap size between two contigs is estimated to be less than *G + 10* bp, a fixed gap size of *G +10* bp would be add between the two contigs. In other cases, the gap size will be reported as estimated from the alignment. The gap size *G* is currently set at 13 bp and can be changed as a parameter.

Figure 11. Example of the trimming of termini of a hybrid scaffold.



Finally, based on the information in the AGP file, the Hybrid Scaffold pipeline outputs two FASTA files. One contains the sequences of the hybrid scaffolds, and one contains sequences not used in the hybrid scaffold. See details in Output Files section. At loci contributed by sequences, the individual nucleotides are output. At loci constructed by genome maps, N bases and the motif sequences are output. In the AGP file, loci constructed by genome maps are annotated as gaps between neighboring sequence contigs. Because NCBI allows only N bases in the gap region, an additional FASTA file is generated where only N bases are output in the gap regions (the motif sequences are not output in the FASTA). This FASTA file has an extension “_NCBI.fasta”.

Trimming overlapping sequence contigs

Due to how sequence assembly algorithms construct sequence contigs, adjacent NGS contigs in the same scaffold sometimes contain overlapping sequences. Starting from Bionano Solve 3.6, there is an “-g” option in the main Hybrid Scaffold script which enables trimming of the overlapping part of the NGS sequence contigs during export of AGP and FASTA files. The overlapping sequence of two adjacent NGS contigs is determined by their alignment to the hybrid scaffolds. This overlapping sequence would be trimmed from one of the NGS contigs (the contig to the right) and exported as a singleton contig during the AGP and FASTA generation process.

Molecule Alignment to hybrid scaffolds and Bionano maps (optional)

The -x option enables molecule alignment to the hybrid scaffolds and the genome maps. This step is optional and can be time-consuming. Its purpose is to enable users to visualize the molecule coverage of genome maps and hybrid scaffolds. This optional step utilizes the same scripts used by the Bionano Solve *de novo* assembly pipeline. In addition to the -x flag and the molecule BNX file (-m option), users also need to provide the directory containing the scripts (-p option) and the parameters XML file (-q option) used by the *de novo* assembly pipeline. Finally, the output files reside in the alignmol_bionano/ and alignmol_hybrid/ directories.

Note that molecule alignment to the original Bionano genome maps has already been performed by the Bionano Solve *de novo* assembly pipeline when the genome maps were initially assembled.

Chimeric quality score generation (for backward compatibility)

The -y option enables the calculation of chimeric quality scores, when they are not present in the input genome map file. This step is optional and can be time-consuming, but it is critical for conflict resolution with an older genome map assembly. Chimeric quality scores are required for conflict resolution (with -N 2 or -B 2 option). With the -y option, the Hybrid Scaffold pipeline would run this at the very beginning – even before conversion of sequences to CMAP (Step 1) – and it requires the molecule BNX file (-m option), and the ERRBIN or ERR file generated of AutoNoise1 during the *de novo* assembly. The AutoNoise1 file contains important noise parameters for scoring alignments. Although not recommended, this file may be omitted, in which case, the Hybrid Scaffold pipeline would use default noise values. The output files reside in the chim_qual/ subdirectory.

If a user specified conflict-resolution (-N 2 or -B 2 option), input a genome map CMAP without chimeric quality scores, and did not turn on this chimeric quality score generation (-y) option, the pipeline would revert to merging conflict-free only genome maps and sequences, thus turning Option 2 to Option 3 in Step 3 above.

Summary statistics

Summary statistics such as N50 and total scaffold length are calculated for the input Bionano genome maps, sequences, the final hybrid scaffolds and others by calc_cmap_stats.pl. The output is printed to screen and to a redirected log file in the sub-directory hybrid_scaffolds/ (with a file name suffix of HYBRID_ SCAFFOLD_log.txt).

Percent coverage of sequence and genome maps to hybrid scaffolds is calculated by calc_xmap_stats.pl. The statistics are printed in align_final/calc_xmap_stats.log.

Log files

The runtime output of each step is printed to screen and to a redirected log file (with a file suffix of HYBRID_ SCAFFOLD_log.txt) in the hybrid_scaffolds/ subdirectory. The summary log file "status.txt" informs Bionano Access the progress. A brief message would be printed indicating the step where an error occurs. Users could then trace the issue by examining the log file of the referenced step (e.g., fa2cmap/*log, align0/*.stdout, align1/*.stdout, assignAlignType/*log, assignAlignType/cut_conflicts/*log, mergeNGS_BN/*log, align_final/*.stdout, align_final/*log, and auto_noise/*log).

Manual conflict resolution

Note: It is recommended that manual conflict resolution be done via Bionano Access' graphical interface. For IrysView and command line users, manual manipulation of the conflict resolution files is available.

Suppose that a user ran the Hybrid Scaffold pipeline requiring that conflicting sequences and Bionano genome maps be cut (i.e. -N 2 and -B 2 option), and after examining the conflict_cut_status.txt file, disagreed with the way the conflicts were handled, perhaps because the cutting decisions contradicted with some prior knowledge. The user could indicate alternate cutting decisions and rerun the pipeline. Based on the example illustrated in Figure 8 and Table 8, after running the pipeline with the -N 2 -B 2 option, the user decided that the cut of sequence 18616 was inappropriate, and that Bionano map 1210 contained a chimeric join. The user could edit the assignAlignType/cut_ conflicts/conflict_cut_status.txt file (Table 9), and rerun the hybrid scaffold pipeline using the -M option along with the new edited status file. The pipeline would then follow the new instructions and break the genome map at the position indicated. See the section on How to Run Hybrid Scaffold Pipeline for directions on input requirements to execute manual conflict resolution.

Table 9. A manual conflict resolution text file with the automatic break decisions modified. This file can be inputted to rerun the hybrid scaffold pipeline.

xMapId	refQry	refId	leftRefBkpt	rightRefBkpt	alignmentOrientation	ref_leftBkpt_toCut	ref_RightBkpt_toCut	ref_toDiscard
4273	ref	18616	505984.0	-1	+	~~cut~~ **okay**	okay	okay

refQry	qryId	leftQryBkpt	rightQryBkpt	alignmentOrientation	qry_leftBkpt_toCut	qry_rightBkpt_toCut	qry_toDiscard
qry	1210	268750.0	-1	+	~~okay~~ **cut**	okay	okay

Alternatively, suppose a user determined that the genome map and the sequence were both correct, as could be the case at large heterozygous structural variant loci, then to prevent collapsing the two alleles, the user could exclude one of the two entries from participating in the merge step. When the Hybrid Scaffold pipeline is rerun (with -M option), in the conflict_cut_status.txt file, the columns ref_ toDiscard and qry_toDiscard could be changed from "okay" to "exclude". Table 10 shows how one could exclude a sequence contig from subsequent scaffolding.

Table 10. A manual conflict resolution text file indicating that conflicting sequence is to be excluded from merging.

xMapId	refQry	refId	leftRefBkpt	rightRefBkpt	alignmentOrientation	ref_leftBkpt_toCut	ref_RightBkpt_toCut	ref_toDiscard
4273	ref	18616	505984.0	-1	+	okay	okay	~~okay~~ **exclude**

refQry	qryId	leftQryBkpt	rightQryBkpt	alignmentOrientation	qry_leftBkpt_toCut	qry_rightBkpt_toCut	qry_toDiscard
qry	1210	268750.0	-1	+	okay	okay	okay

Note that an equivalent set of output files as in the automatic conflict-resolution step is produced in the cut_conflicts_M*/ sub-directory.

How to run the single-enyme Hybrid Scaffold pipeline

Using within Bionano Access or IrysView

Please see the Bionano Access Software User Guide (PN# 30142) for how to run the Hybrid Scaffold pipeline using Bionano Access.

Using Bionano Solve command line

The Hybrid Scaffold pipeline can be run using the command:

perl hybridScaffold.pl
-n < sequence file in FASTA format>
-b <Bionano CMAP file>
-c <hybrid scaffold configuration file in XML format>
-r <RefAligner binary file>
-o <output directory>
-B <conflict filter level genome maps; 1, 2 or 3>
-N <conflict filter level for sequences; 1, 2 or 3>
-f <a flag to overwrite existing files; optional>
-x <a flag to align molecules to hybrid scaffolds and genome maps>
-y <a flag to generate chimeric quality score for the input genome maps>
-M <a conflict resolution text file; optional>
-m <molecule BNX file to align molecules to maps and hybrid scaffolds; optional>
-p <de novo assembly pipeline script; optional but needed for the -x option>
-q <de novo assembly optArguments XML file; optional but needed for the -x option>
-e <de novo assembly noise parameter ERRBIN or ERR file; recommended for -y option>
-v <a flag to print the pipeline version>

Here is an example:

perl scripts/HybridScaffold/hybridScaffold.pl
-n data/seq/input.fa
-b data/Bionano/exp_refineFinal1_contigs.cmap
-c data/hybridScaffold_config.xml
-r bin/RefAligner
-o data/hybridscaffold/output
-f
-B 2
-N 2
-x

-y
-m data/Bionano/molecules.bnx
-p Pipeline/scripts
-q Pipeline/scripts/optArguments_small.xml
-e data/assembly/output/contigs/auto_noise/autoNoise1.errbin

When a user specifies a conflict resolution file when rerunning the Hybrid Scaffold pipeline, the input sequence and genome map assemblies MUST be the same as the initial run. Also, the output directory specified MUST be the same as the initial output directory, because the pipeline requires certain specific files in the initial output directory. If these conditions are not met, the pipeline is terminated with an error. Here is an example of rerunning the pipeline using the -M option:

perl scripts/HybridScaffold/hybridScaffold.pl
-n data/seq/input.fa
-b data/Bionano/exp_refineFinal1_contigs.cmap
-c data/hybridScaffold_config.xml
-r bin/RefAligner
-o data/hybridscaffold/output
-M myConflictResFile.txt
-m data/Bionano/molecules.bnx
-p Pipeline/scripts
-q Pipeline/scripts/optArguments_small.xml

Output sub-directories will have a _M1 suffix. If a user re-ran the pipeline with the -M option, the -B, -N, -y and -f parameters would be ignored. Finally, one could rerun with -M a second time; the output sub-directories would have a suffix of _M2. Each time the hybrid scaffold pipeline is rerun with the -M option, the suffix number increases by 1. See the section on output files below.

Output files

Here is an example of the output directory structure after running hybrid scaffold once and rerunning it with the manual modification option (-M) twice.

data/hybridscaffold/output/
hybrid_scaffolds/
hybrid_scaffolds_M1/
hybrid_scaffolds_M2/
chim_qual/
fa2cmap/
align0/
align1/
assignAlignType/
cut_conflicts/
cut_conflicts_M1/
cut_conflicts_M2/
mergeNGS_BN/
mergeNGS_BN_M1/
mergeNGS_BN_M2/
align_final/

align_final_M1/
align_final_M2/
agp_fasta/
agp_fasta_M1/
agp_fasta_M2/
auto_noise/
auto_noise_M1/
auto_noise_M2/
alignmol_bionano/
alignmol_bionano_M1/
alignmol_bionano_M2/
alignmol_hybrid/
alignmol_hybrid_M1/
alignmol_hybrid_M2/

Note that these output files reside in the directory specified by the -o command. The final results of the pipeline reside in the hybrid_scaffolds subdirectory, which contains the following files (* denotes file prefix):

- *_log.txt - a log file of the pipeline.
- *_HYBRID_SCAFFOLD.cmap - the hybrid scaffolds in CMAP format.
- *_BNGcontigs_NGScontigs.xmap, _q.cmap, _r.cmap - the alignment files of the input Bionano maps (query) and sequences (reference). These files are duplicates of align1.xmap, align1.q.cmap, and align1.r.cmap in the align1/ subdirectory.
- *_BNGcontigs_HYBRID_SCAFFOLD.xmap, _q.cmap,
- _r.cmap - the alignment files of the hybrid scaffolds (reference) and Bionano maps (query). The files can also be found in the align_final/ subdirectory. See STEP 4: SEQUENCE-HYBRID OR BIONANO-HYBRID ALIGNMENT.
- *_NGScontigs_HYBRID_SCAFFOLD.xmap, _q.cmap,
- _r.cmap - the alignment files of the hybrid scaffolds (reference) and the sequences (query). The files can also be found in the align_final/ subdirectory. See STEP 4: SEQUENCE-HYBRID OR BIONANO-HYBRID ALIGNMENT.
- hybridScaffold_config.xml - a copy of the XML configuration file detailing the parameters used.
- *_HYBRID_SCAFFOLD.agp - the hybrid scaffolds in AGP format. This file describes how the contigs are ordered in the scaffolds.
- *_HYBRID_SCAFFOLD.fasta - the hybrid scaffolds in FASTA format. This file should be considered as the final output of the scaffold sequences.
- *_HYBRID_SCAFFOLD_NCBI.fasta – the hybrids scaffold in FASTA format without enzyme motif in gap regions.
- *_HYBRID_SCAFFOLD_NOT_SCAFFOLDED.fasta - the remaining sequences that did not contribute

to the scaffolds, or overlapping sequences that have been trimmed. This file should be considered as the final output of the leftover sequences.

- *_HYBRID_SCAFFOLD.gap –information about gaps between neighboring NGS sequence contigs on hybrid scaffolds.
- *_HYBRID_SCAFFOLD_trimmedTailGap.coord – a file detailing how much of the termini of hybrid scaffold was removed. See STEP 5: AGP/FASTA FILE GENERATION.
- conflicts.txt – a file showing the locations of the conflicts between the sequences and genome maps.
- conflicts_cut_status.txt – a file showing how conflicts are handled. This file only appears if -B 2, -N 2, or -M options are selected.
- bn_pre_cut_projected_ngs_coord_annotations.bed - if Bionano genome maps were cut during conflict-resolution (-B 2 or -M), the coordinates of the cuts are shown. Note that the coordinates are not based on the genome maps but on the corresponding (conflicting) sequences.
- ngs_pre_cut_annotations.bed - if sequences were cut during conflict-resolution (-N 2 or -M), the coordinates of the cuts are shown. The coordinates are based on the sequence coordinates.
- auto_cut_BN_coord_translation.txt - if Bionano genome maps were cut during conflict-resolution (-N 2 or -M), the coordinates of the maps after the cuts are shown. The coordinates are based on the sequence coordinates.
- auto_cut_NGS_coord_translation.txt - if sequence contigs were cut during conflict-resolution (-N 2 or -M), the coordinates of the sequences after the cuts are shown. The coordinates are based on the sequence coordinates.
- hybrid_scaffold_informatics_report.txt - the statistics of original Bionano genome maps and NGS sequences, conflicts, cuts, and hybrid scaffolds.
- ncbi_manifest.txt - manifest for NCBI submission.
- *.fasta - the original input FASTA.
- *._key.txt - the original sequence IDs and the ones used in hybrid scaffold.
- *_key.txt.cut.txt - the sequence IDs before and after cuts.

The chim_qual/ subdirectory contains the original input genome maps but with the addition of chimeric quality scores associated with each label.

The fa2cmap/ subdirectory contains the results of the FASTA-to-CMAP conversion (Step 1).

The align0/ and align1/ subdirectories contain the results of the initial alignment between sequence and Bionano

assemblies (Step 2).

The assignAlignType/ subdirectory contains the filtered (conflicting-free) Bionano genome maps and sequences (Step 2). Within this assignAlignType/ subdirectory, the cut_conflicts/subdirectory contains the post-cut or post-exclude sequences and genome maps (Step 2).

The mergeNGS_BN/ subdirectory contains the results of the merge process (Step 3). The file step2.hybrid.cmap has the resulting hybrid scaffolds.

The align_final/ subdirectory contains the results of the final alignments between the used sequences in the merge process and the hybrid scaffolds, as well as the alignments between the used Bionano genome maps and the hybrid scaffolds (Step 4).

The auto_noise/ subdirectory contains intermediate results needed to align molecules to genome maps and to hybrid scaffolds. The alignmol_bionano/ and alignmol_hybrid/ subdirectories contain final results of aligning molecules to the genome maps and hybrid scaffolds, respectively.

Configuration file and parameters

The configuration file defines the parameters used in each step of hybrid scaffold. Here is a brief description of some of the parameters.

- fa2cmap: run fa2cmap_multi_color.pl to convert sequence FASTA file into a CMAP file (Step 1):
  - 'enzyme' – define enzymes for *in-silico* digestion, where the available enzymes are BspQI, BbvCI, BsmI, BsrDI, BseCI, BssSI, and DLE-1. Review the Bionano Solve release notes for any changes to supported enzymes. Note that if more than one enzyme was used, the enzymes can be specified using the val0 and val1 fields.
  - 'minLabels' – specify minimum number of label sites per sequence
  - 'minLength' – specify minimum length in kbp of each sequence
- align1: run RefAligner to align sequences with Bionano genome maps (align0 and align1 in Step 2):
  - ‘T’ – p-value required in alignment. The default is set to 1e-10.
  - RefAligner parameters
- assignAlignType: run AssignAlignType.pl to flag conflicting alignments (AssignAlignType in Step 2)*:*
  - 'T_cutoff' – minimum confidence value used to flag chimeric/conflicting alignments. The default is set to 1e-13.
  - 'max_overhang' – maximum number of overhang labels used to flag chimeric/conflicting alignments. The default is set to 5.

- cut_conflicts: run cut_conflicts.pl to flag conflicting alignments (cut_conflicts in Step 2)*:*
  - ‘min_coverage_threshold’ – minimum number of molecules aligning to the conflicting loci on the genome maps. If the coverage at a conflicting locus on the genome map is below this threshold, the genome map is cut. Otherwise, the sequence is cut. The default is set to 10.
  - ‘min_quality_score_threshold’ – minimum percentage of aligned molecules spanning the left and right of the conflicting loci on the genome maps. If the score at a conflicting locus on the genome map is below this threshold, the genome map is cut. Otherwise, the sequence is cut. The default is set to 35.
- mergeNGS_BN: run MergeNGS_BN.pl for map merging (Step 3):
  - 'id_shift' – ID shift value for shifting Bionano genome map IDs to prevent ID collision with sequences. Note that if this value is still too small, the pipeline would iteratively increase it by multiples of ten until there is no ID collision. The default is set to 100000.
  - ‘merge_Tvalue’ – p-value cutoff required to perform pairwise merge. The default is set to 1e-13.
  - ‘pairmerge’ – the minimum length of alignment in kbp required for a merge. The default is set to 160 kbp.
  - Other RefAligner parameters
- align_final: run RefAligner to align sequences participated in the scaffold process to the hybrid scaffolds (Step 4):
  - ‘T’ – p-value required in alignment. The default is set to 1e-10.
  - RefAligner parameters for alignment
- refineFinal1: run RefAligner to generate chimeric quality scores
  - Changes are not recommended.

Suggested parameters

Depending on the complexity of the genome of interest, the values of certain parameters could be adjusted accordingly. The hybridScaffold_config.xml provided with the pipeline contains the default parameters targeted for human-sized genomes. The file hybridScaffold_DLE1_config.xml is designed specifically for scaffolding DLS genome maps, while hybridScaffold_DLE1_HiC_config.xml is designed for scaffolding HiC-data.-

The '-T' p-value determines the stringency of the initial alignment between sequences and Bionano genome maps (align1 in Step 2), and the final alignment between sequences and hybrid scaffolds (Step 4). The 'merge-Tvalue', and the pairmerge length (kbp) in the map merging step are the required p-value and alignment length thresholds (Step 3) that determine the stringency of the merging between sequences and genome maps. It is important to note that while the use of more aggressive parameters could increase incorporation of shorter sequence contigs and contiguity of the hybrid scaffolds, doing so could increase the risk of making more chimeric joins. Therefore, users should carefully evaluate their hybrid scaffold results with other experimental data.

Section II. Two-enzyme workflow

The two-enzyme Hybrid Scaffold pipeline combines two sets of Bionano maps and a sequence assembly to generate two-enzyme hybrid scaffolds. It utilizes various components from the single-enzyme pipeline and implements several new functionalities specific to two-enzyme scaffolds in R. Conceptually, the pipeline can be separated into three steps (Figure 12): 1) two-enzyme conflict resolution, 2) merging of Bionano maps and NGS contigs to construct two-enzyme hybrids, and 3) anchoring of sequence to hybrid scaffolds and export. This workflow can be used on NLRS and DLS data in any combination.

Figure 12. Detailed workflow of the two-enzyme Hybrid Scaffold pipeline. Ovals represent data entity such as BNG maps, NGS contigs or hybrid scaffolds. Rectangles represent operations that process various data. It contains three steps: 1) first, conflicts between NGS and each set of BNG maps as well as conflicts between the two BNG maps are detected and resolved; 2) next, the BNG maps and NGS are separately merged to generate two sets of single-enzyme hybrid scaffolds, and the two set of single-enzyme hybrids are further merged into a two-enzyme hybrid scaffolds; 3) lastly, NGS contigs are aligned back to the two-enzyme hybrid scaffolds utilizing label patterns from both enzyme simultaneously.

Controller

The two-enzyme pipeline is controlled by the script “runTGH.R”. Similar to the single-enzyme pipeline, it accepts as input two CMAP files, which represent the two sets of input Bionano maps, one FASTA file, which represents the NGS assembly, and an XML file that specifies the parameters used for scaffolding.

Step 1. Combined conflict resolution

Step 1a. Detecting NGS-BNG conflicts

The pipeline starts by detecting conflicts between NGS and each set of Bionano maps from input. This is done by calling the single-enzyme hybrid scaffold script “hybridScaffold.pl” with the “-S” options, which runs the pipeline up to the conflict resolution step (Steps 1 and 2 in single-enzyme pipeline). Since each set of BNG-NGS alignments can detect a subset of conflicts, the pipeline merges the conflict cuts from each single-enzyme hybrid run by merging the “conflict_cut_status.txt” file found in the “assignAlignType/ConflictCut/” folder in the single-enzyme output to maximize sensitivity of detecting conflicts.

Step 1b. Detecting conflicts between two sets of Bionano maps

In addition to BNG-NGS conflicts, conflicts between the two sets of Bionano maps can be identified by identifying inconsistency in multiple pairs of NGS-BNG alignments. As illustrated in Figure 13, operationally, assume BspQI map1 and BssSI map1 are a pair of genome maps that aligned to the same NGS contig, NGS1. Bionano maps generated from different enzyme cannot be aligned directly to each other as their label positions have no correspondence. However, because they are aligned to the same NGS contig, by inference, we can establish a corresponding region L on BspQI map1 and BssSI maps that represent the same genomic region (see Figure 13). Within this region, any NGS contigs that align to BspQI map1 should also be aligned to the BssSI map1. We would thus search for any NGS contig in region L that align to BspQI map1 but another BssSI map instead of BssSI map1 (i.e. BssSI map2 in the example). This would indicate potential assembly errors either in the BspQI map1 and BssSI map1. In this case, we cannot determine which map may be incorrect, and thus, we would cut both maps at the conflicting location. The conflict cuts detected in both steps were merged in a single conflict cut table and can be found in the output sub-folder “/CombinedConflictsCut” (see output section below for more details).

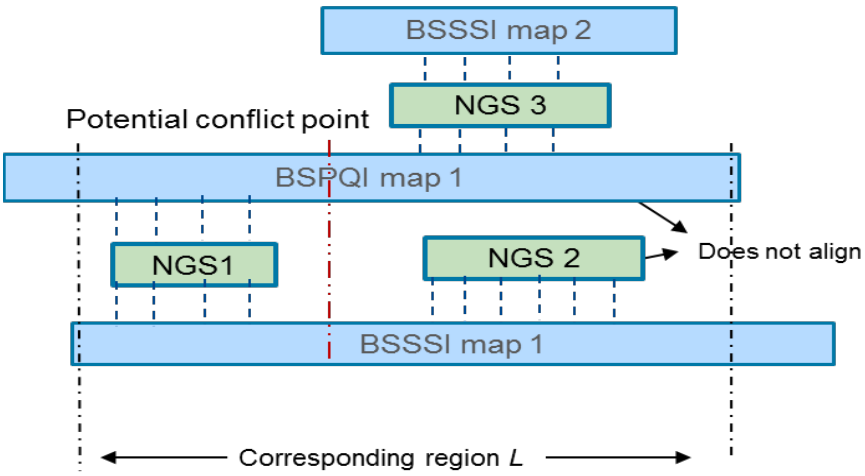


Figure 13. An example illustrating that conflicts between a pair of BspQI and BssSI maps can be detected by identifying inconsistency in multiple NGS-BNG alignments

Step 2. Map merging

After conflict resolution, two sets of BNG maps and the NGS contigs are merged together to construct combined scaffolds. This is done in two steps. First, each set of BNG maps are separately merged with the corresponding NGS-derived in-silico maps iteratively (see Step 3 from single-enzyme section) to create two sets of single-enzyme hybrid scaffolds. This is done by calling the single-enzyme pipeline with "-M" and passing the combined conflict table file that were generated from the step one.

Next, the two sets of single-enzyme hybrid scaffolds are further merged into two-enzyme hybrid scaffolds. Each NGS contig is first aligned to the best-scoring single-enzyme hybrids from each set. To filter out spurious alignments, the NGS must be aligned with a stringent p-value of at least 1e-13, and the p-value must also be hundred times smaller than the P-value of the second best alignment in each set. Then, a graph is constructed where each node represents a single-enzyme hybrid scaffold and an edge is formed between two nodes if at least one NGS contig aligns to both single-enzyme hybrid scaffolds. Finally, we cluster all the single-enzyme hybrid scaffolds together by finding all the connected components in the graph. Each connected component represents a two-enzyme hybrid scaffold. The relative positions of single-enzyme hybrid scaffolds from different enzymes can be computed using the NGS contigs that link them together. Using the NGS contigs as the reference point, the workflow computes the relative positions of the labels on each Bionano maps. This then provides a common coordinate system to merge the label positions from different single-enzyme hybrids into combined hybrid maps that contain labels patterns from both enzymes. The two-enzyme hybrid maps can be found in the sub-folder "/two_enzyme_hybrid_scaffold_M1/Sandwich2/".

Similar to Step 4 of the single-enzyme pipeline, additional rules were applied to handle potential complex multi-path regions in the genome. During merging of two single-enzyme maps into a two-enzyme map, if an NGS contig is only aligned to non-unique region of any of the Bionano map (Figure 14a), it cannot be used as evidence for merging the two single-enzyme maps. The merge is only allowed if the NGS sequence is aligned to a sufficiently

large (>80 kbp) unique region on both single-enzyme maps (Figure 14b).

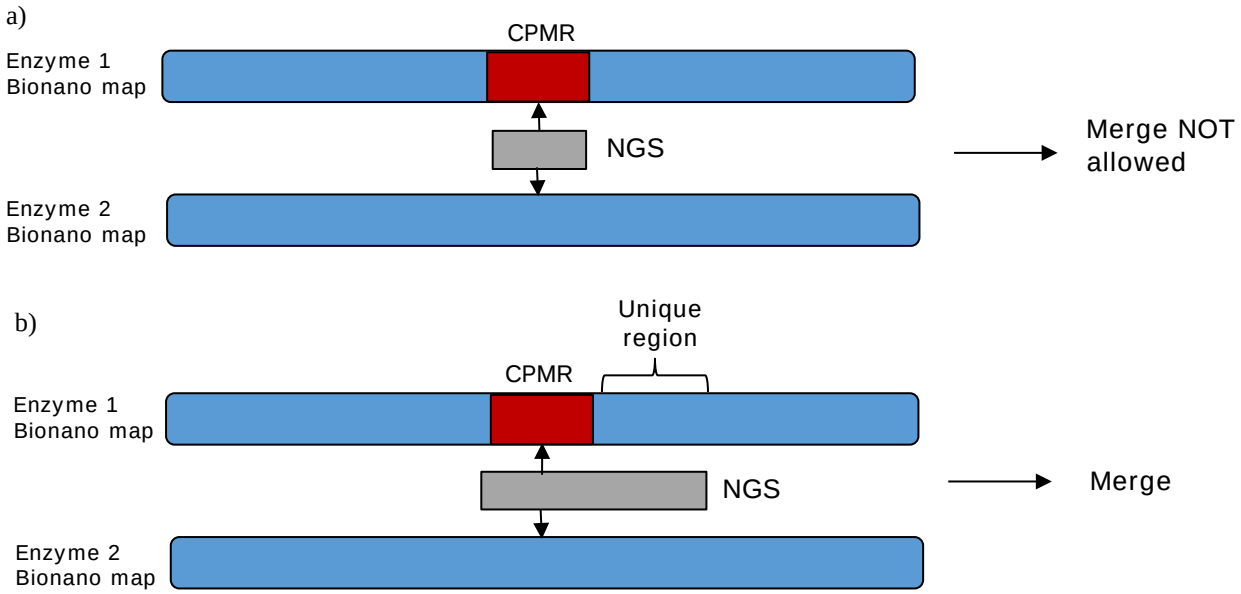


Figure 14. A schematic illustrating the handling of repeat regions in two-enzyme hybrid scaffolds

Step 3. Aligning NGS to two-enzyme hybrids

To anchor sequence contigs to the combined two-enzyme scaffolds, NGS contigs are first aligned to the scaffolds using label patterns from only one enzyme. Then, NGS contigs that cannot be aligned initially are aligned a second time using label patterns from both motifs. This is done in two steps because the computational complexity for aligning genome maps with two sets of label patterns simultaneously is considerably higher, thus resulting in much longer runtime particularly for large NGS contigs. Therefore, only shorter NGS contigs are aligned using two-enzyme alignment. All alignments can be found in the output subfolder "/two_enzyme_hybrids_hybrid_scaffold_M1/alignfinal/".

Export to AGP and Fasta

This step is identical to the last step in single-enzyme. Please see the previous section for details. The option to turn on the trimming of overlapping NGS contig is "--trim"; it is not turned on by default.

Statistical calculations

Summary statistics for input and scaffold results can be found in the file "hybridScaffold_informatic_report.txt" found in the folder two_enzyme_hybrid_scaffold_M1. A brief report on the current status of the pipeline can be found in the file "status.txt". A more detailed log of the pipeline run can also be found in the files "TGH.log" and "TGH.log.errlog".

Manual conflict resolution

Note: It is recommended that manual conflict resolution be done via the graphical interface within Bionano Access. For IrysView and command line users, the users can manually modify the conflict cut files using a text editor.

The two-enzyme Hybrid Scaffold pipeline enables users to examine the conflict cuts determined by the pipelines and manually modify, add or delete conflict cuts, as is done for single-enzyme hybrid scaffold. This is done by running the pipeline with the "-m1" and "-m2" options in the pipeline. Note that two sets of conflicts cut files are needed, one for each enzyme, since we need one conflict file for each set of BNG maps to generate the single-enzyme hybrids (see Step 2 above). It is important that the NGS conflict cuts are consistent in both conflicts files; otherwise, when merging the two sets of single-enzyme hybrid scaffolds into the two-enzyme hybrid scaffolds, we may be merging parts of maps that do not belong to the same genomic regions, together. Fortunately, this is done automatically for the user modifying the conflicts cut files through the Bionano Access interface (please see the Bionano Access Software User Guide, document 30142).

Running two-enzyme Hybrid Scaffold pipeline

Using Bionano Access

Please refer to the Bionano Access Software User Guide, document 30142.

Software Dependencies

Please see the Bionano Solve Installation Guide (document 30172) for software dependencies.

Using command line

The pipeline can be called by running the script runTGH.R using the following syntax and options:

```
Rscript  /<hybrid scaffold path>/runTGH.R

runTGH.R [--] [--help] [--override] [--opts OPTS] [--BNGPath1 BNGPATH1] [--BNGPath2 BNGPATH2] [--
NGSPath NGSPATH] [--OutputDir OUTPUTDIR] [--RefAlignerPath REFALIGNERPATH] [--RunFlags
RUNFLAGS] [--Enzyme1 ENZYME1] [--Enzyme2 ENZYME2] [--ManualCut1 MANUALCUT1] [--
ManualCut2 MANUALCUT2] [--tar TAR] [--status STATUS] paramFile

positional arguments:

paramFile                                   Parameter file (xml)

flags:
  -h, --help                                Show this help message and exit
  -f, --override                            Override output folder [default: FALSE]

optional arguments:
  -b1, --BNGPath1 BNGPATH1          Path to BNG maps for enzyme1
```

```
-b2, --BNGPath2 BNGPATH2          Path to BNG maps for enzyme2
-N, --NGSPath NGSPATH                  Path to NGS sequence (fasta file)
-O, --OutputDir OUTPUTDIR              Output directory [default: ./]
-R, --RefAlignerPath REFALIGNERPATH    Path to RefAligner
-e1, --Enzyme1 ENZYME1                 Enzyme used in BNG maps specified by --bng1
-e2, --Enzyme2 ENZYME2                 Enzyme used in BNG maps specified by --bng2
-m1, --ManualCut1 MANUALCUT1           Manual cut file (for enzyme 1)
-m2, --ManualCut2 MANUALCUT2           Manual cut file (for enzyme 2)
-t, --tar TAR                          Result tar file to be import to Bionano Access.  This tar
                                       file contains only results file relevant to relevant to two-
                                       enzyme Hybrid Scaffold.  [default: TGH.tar]
-s, --status STATUS               Path to status file [default: status.txt]
--trim                            Turn on trimming of overlapping NGS contigs during AGP export
```

Note that “-O” and “-R” options are capitalized. An example command line is shown below:

```
Rscript scripts/HybridScaffold/runTGH.R
    -N  data/seq/NGS.fa
    -b1 data/Bionano_enzyme1/exp_refineFinal1_contigs.cmap
    -b2 data/Bionano_enzyme2/exp_refineFinal1_contigs.cmap
    -e1 BSPQI
    -e2 BSSSI
    -O data/hybridscaffold/output
    -R data/tools/RefAligner/RefAligner
    hybridSscaffold/TGH/hybridScaffold_two_enzymes.xml
```

Note that the last argument, which is the configuration xml file, is a positional argument, so hence “-“ option is not needed.

Below is another example of running the pipeline with manual conflict cut options:

```
Rscript scripts/HybridScaffold/runTGH.R
    -N data/seq/NGS.fa
    -b1 data/Bionano_enzyme1/exp_refineFinal1_contigs.cmap
    -b2 data/Bionano_enzyme2/exp_refineFinal1_contigs.cmap
    -e1 BSPQI
    -e2 BSSSI
    -m1 result/Conflict_cut_status_enzyme1.txt
    -m2 result/Conflict_cut_status_enzyme2.txt
    -O data/hybridscaffold/output
    -R data/tools/RefAligner/RefAligner
    hybridSscaffold/TGH/hybridScaffold_two_enzymes.xml
```

Alternatively, all the command-line parameters can be also specified in an xml file and pass to the main control script by running:

```
Rscript /<hybrid scaffold path>/runTGH.R hybridScaffold_two_enzymes.xml
```

See “configuration file” section below for explanation of the configuration file.

Output files

Below is an example output structure from two-enzyme hybrid scaffold after one round of manually-edited conflict cuts:

data/hybridscaffold/output/
BSPQI/
BSSSI/
CombinedConflictCuts/
Sandwich1/
two_enzyme_hybrid_scaffold_M1/
    fa2cmap/
     CombinedConflictCuts
    align1/
    alignfinal/
    Sandwich2/
    AGPExport/
TGH_M2/
    fa2cmap/
     CombinedConflictsCuts
    align1/
    alignfinal/
    Sandwich2/
    AGPExport/

Note that these outputs reside in the directory specified by the “-o” option.

- The sub-folder “BSPQI” and “BSSSI” contain outputs for the single-enzyme hybrid scaffold pipeline for each set of BNG maps respectively. These folders are named according the enzyme name that was passed to the pipeline (i.e “-e1” and “-e2” option).
- The folder “CombinedConflictCuts” contains the merged conflicts that are generated from automatic conflict detection in the pipeline (see Step2 from single-enzyme workflow section for a description of the conflict cut file).
- The folder “Sandwich1” contains intermediate results files that are needed to perform the “BNG vs BNG” conflicts detections, users can safely ignore it.
- The folder “TGH_M*” folder contains results that are specific to the two-enzyme scaffolds.
- The “fa2cmap” folder contains output files from in silico digestions of the sequence FASTA file using label patterns from the two enzymes.
- The “align1” folder contains alignments of NGS to Bionano maps before conflict-resolution (basically a copy of the align1 folder from the single-enzyme hybrid scaffold pipeline output).
- The “alignfinal” folder contains final alignments of NGS contigs and Bionano maps to the two-enzyme hybrid scaffold, as well as alignments of NGS contigs to BNG maps. The NGS contigs and BNG maps used in these alignments are those after the conflict-resolution step. They have been cut at detected

conflict sites. The filenames of the alignment files follow a naming convention based on what type of alignment has been performed: “E_” is followed by the name of the enzyme used, “Q_” is followed by the data used as query and “A_” is followed by the data used as the anchor/reference. For example, the xmap file “E_BSPQI_Q_NGScontigs_A_HYBRID.xmap” contains alignment of NGS contigs to hybrid scaffolds using label patterns from BSPQI. Similarly, the file “E_BSPQI_E_BSSSI_ Q_NGScontigs _A_HYBRID.xmap” contains alignment of NGS contigs to hybrid scaffolds using label patterns from both enzymes.

Note that in the single-enzyme Hybrid Scaffold workflow, the output folder of each round of scaffolding with manual cuts is indicated by “_M*” suffix. (i.e. the first round of manual cut has output folder suffix with “_M1” and so on). For the two-enzyme Hybrid Scaffold workflow, an additional round of single-enzyme hybrid scaffold with the manual cut option is performed when passing in the combined conflict files. Thus, the results for automatic conflict resolution are stored in folder with suffix “_M1” in the two-enzyme hybrid output. In order to be consistent with this naming convention, as shown in the example above, the first round of two-enzyme hybrid scaffold run with manually edited cuts would output a folder with suffix of “_M2” and so on.

Configuration file and parameters

In the Hybrid Scaffold program subfolder “/TGH”, there is an example parameters file “hybridScaffold_two_enzymes.xml” that contains recommended parameters for running the two-enzyme pipeline. It is composed of three sections, a “TGH” section that contains parameters specific to the two-enzyme pipeline, and “hybridScaffold1” and “hybridScaffold2” sections, which are identical to the parameters for single-enzyme hybrid scaffold. The single-enzyme parameter sets are intended for Step 1 of the pipeline (see the single-enzyme section for more details). The recommended parameters are shown to be robust across different genomes and a wide variety of NGS data. Therefore, in general, we encourage users to run the pipeline using the recommended parameters. For users who wish to further fine-tune the parameters, we will highlight parameters below that control important aspects of the pipeline.

Conflict resolution

Users can decide how aggressive they want to be in the conflict-resolution step. Two parameters mainly control this: the “T_cutoff” parameter, which determines the p-value threshold of the NGS-BNG alignment and the “max_overhang” parameter, which determines the number of labels in the unaligned regions, in the “assignAlignType” subsection in either “hybridScaffold1” or “hybridScaffold2” section. A larger “T_cutoff” value or a smaller “max_overhang” value will increase the sensitivity of conflict detection, but it may decrease scaffold contiguity as more contigs/maps are cut during conflict resolution.

Merging

Users can also control how aggressively they want to merge BNG maps and NGS contigs to generate the hybrid scaffolds. The parameters “merge_Tvalue” and “pairmerge” in the subsection “mergeNGS_BN” in either “hybridscaffold1” or “hybridScaffold2” section determine how aggressively the BNG maps and NGS contigs are merged to generate the single-enzyme hybrid scaffolds. The parameter “twoEnzymeMergeT” in the “TGH” section control how stringent the NGS-hybrid alignment needs to be for the pipeline to merge two single-enzyme hybrid scaffolds into a two-enzyme hybrid scaffolds. Higher values in “merge_Tvalue” and “twoEnzymeMergeT”, or a smaller value in “pairmerge” means the pipeline will be more aggressive in merging the maps, but this might also increase the number of chimeric joins in the final scaffolds.

Anchoring NGS to final scaffolds

The parameter “T” in the “alignfinal” subsections in either “hybridScaffold1” or “hybridScaffold2” determines the p-value of anchoring an NGS contig to final hybrid using labels from only one motif. The parameter “alignFinal2Enzyme” in the “TGH” section determines the p-value of aligning an NGS contigs to the final hybrid scaffolds using labels from both motifs. Increasing the values of these parameters allows for anchoring of more NGS contigs into the final hybrid scaffolds, but some small contigs may be misplaced.

Technical Assistance

For technical assistance, contact Bionano Genomics Technical Support.

You can retrieve documentation on Bionano products, SDS's, certificates of analysis, frequently asked questions, and other related documents from the Support website or by request through e-mail and telephone.

Type	Contact
Email	**support@bionanogenomics.com**
Phone	**Hours of Operation:** **Monday through Friday, 9:00 a.m. to 5:00 p.m., PST** **US: +1 (858) 888-7600**
Website	**www.bionanogenomics.com/support**

Bionano Genomics, Inc.
9540 Towne Centre Drive, Suite 100
San Diego, CA 92121