Bionano Solve Theory of Operation: Variant Annotation Pipeline

Document Number: 30190
Document Revision: J

Table of Contents

Legal Notice 3
Revision History 4
Introduction 4
Workflow 4
Molecule check 6
Strategy for deriving molecule coverage cutoff recommendations for nickase 6
Input files and parameters 9
Running Variant Annotation Pipeline 11
Expected output 11
CNV Annotation 12
Recommendations on selecting rare SVs in the results of a trio analysis 12
Recommendations on selecting for putative *de novo* SVs in the results of a trio analysis 14
Control SV database 16
Merging SV databases 17
Custom gene annotations 18
ISCN Annotation 19
CNV loss and gain 19
Deletions 20
Duplications 20
Insertions 20
Inversions 21
Inter-chromosomal translocations 21
Intra-chromosomal fusions 21
Unsupported cases 22
Technical Assistance 23

Legal Notice

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

Revision History

Revision	Notes
J	Update for Solve 3.7 release • Add description of CNV annotation • Add description of ISCN notation • Update cutoffs for high confidence variants

Introduction

The purpose of the variant annotation pipeline (VAP) is to enable users to determine if a Bionano structural variant (SV) or copy number variant (CNV) call is relevant to certain physical or disease traits in humans. It can help to identify if a variant is putative *de novo* or proband-specific in family studies, somatic in tumor-normal pair studies, rare among phenotypically "normal" individuals, overlaps with annotated genes, or is a potential false positive call.

Workflow

The pipeline was written in Perl and designed for three types of analyses: **single** sample, **dual** (e.g., tumor-normal), and **trio** (e.g., child-parents family) analyses. It annotates SV and CNV calls made against a reference genome.

Figure 1 describes the workflow for a trio analysis. It takes in the Bionano SV file (SMAP) and CNV (cnv_calls_exp.txt) of the sample of interest (the proband), and adds annotation information to each call. It gathers the coverage and assembly scores of contigs to determine if the SVs were called due to chimeric joins in assembly and were not true variants. For each variant, the pipeline searches for overlapping genes, neighboring genes, and potential fusion genes. Note that for the fusion genes annotation, the pipeline currently only checks if a variant can potentially bring two genes together based on genomic location. It does not consider the orientation of the gene transcript. Furthermore, to estimate the population frequency of the proband's variants, it queries them against a control sample SV database. This database is essential for estimating variant frequency as the calls stored in the database are found by Bionano's genome mapping, which can discover variants unidentifiable by other technologies. The pipeline can use Bionano's control sample database for human (hg19 and hg38) or mouse (mm10) data, or a user-generated database. See section Control SV database for more details. Human samples are also compared against the Database of Genomic Variants (DGV) to provide an additional population frequency for each SV.

When run with control samples (e.g., non-tumor sample or parents), the variant annotation pipeline checks whether the calls are sample-specific. For example, in the case of a trio study, it first checks whether the variants in the child are also found in the parents' assemblies. Next, it checks whether these variants are found in parents' molecules. Heterozygous variants may be missed in the parents' assemblies because the variant alleles are not assembled, and so checking parents' molecules would avoid incorrectly classifying the proband's variants as *de*

novo. Finally, this pipeline would perform this check on the proband's molecules as well. In principle, all calls in the proband are expected to be validated by the molecules from the same sample. However, in rare cases, interval sizing errors or chimeric join errors during assembly can generate false variants, and these can be eliminated by checking the sample's own molecules.

The variant annotation pipeline is fully integrated with Bionano Access™. The user can start the variant annotation pipeline with user-defined parameters, view the molecule alignments, and filter SV calls based on the annotation within Access. To streamline analysis, the user could also set up variant annotation when setting up a *de novo* assembly or a Rare Variant Pipeline run in Bionano Access.

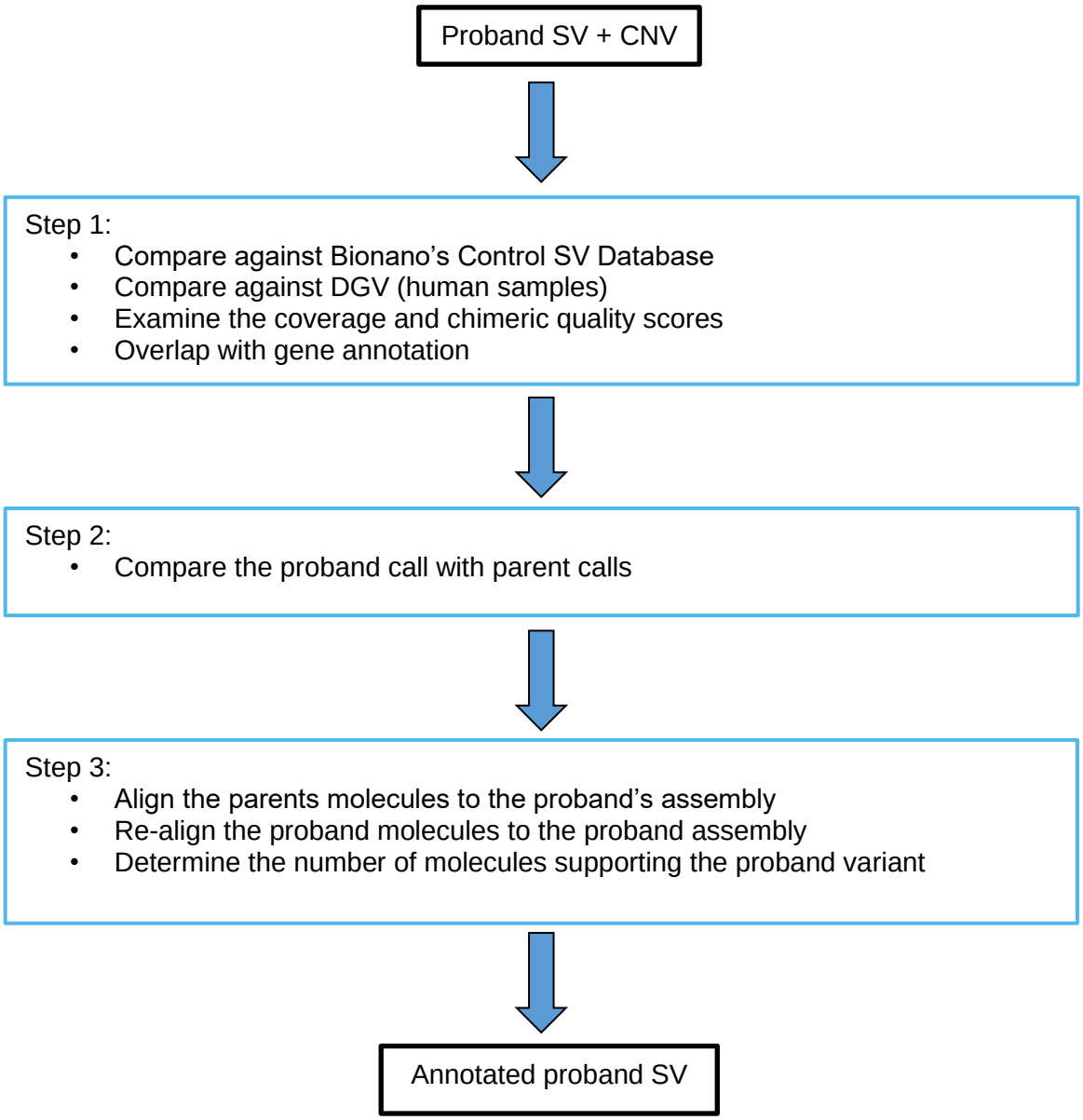


Figure 1 – Workflow for a trio analysis in the variant annotation pipeline.

Molecule check

The molecule check process determines if there are sufficient molecules that support the genome maps on which SVs have been identified. For example, if the molecules of a hypothetical Sample A are aligned to the assembled genome map of Sample A containing a SV of interest, it is expected that the SV region is supported by many molecules. If not, then this is an indication that Sample A's genome map may be incorrectly constructed. The molecule support quantity should not be used for estimation of variant allele fraction (VAF) because this test requires a high burden of proof and does not include a complete counting of covering molecules, refer to the VAF calculated instead. Alternately, when aligning molecules from a different sample to Sample A's genome maps, for example, aligning the molecules from Sample A's parent to Sample A's genome maps, then a lack of support for the genome map's SV structure would indicate inter-individual allelic difference.

For a variant to be confirmed in the molecule check process, by default, at least five molecules are required to align +/- five labels across each variant breakpoint on the genome map in DLE-1 data (Figure 2). For nickase, at least nine molecules are required to align +/- two labels across each variant breakpoint on the genome map. The number of molecules and the number of labels can be adjusted by the user. The following section describes how to choose a cutoff for the number of supporting molecules for DLE-1.

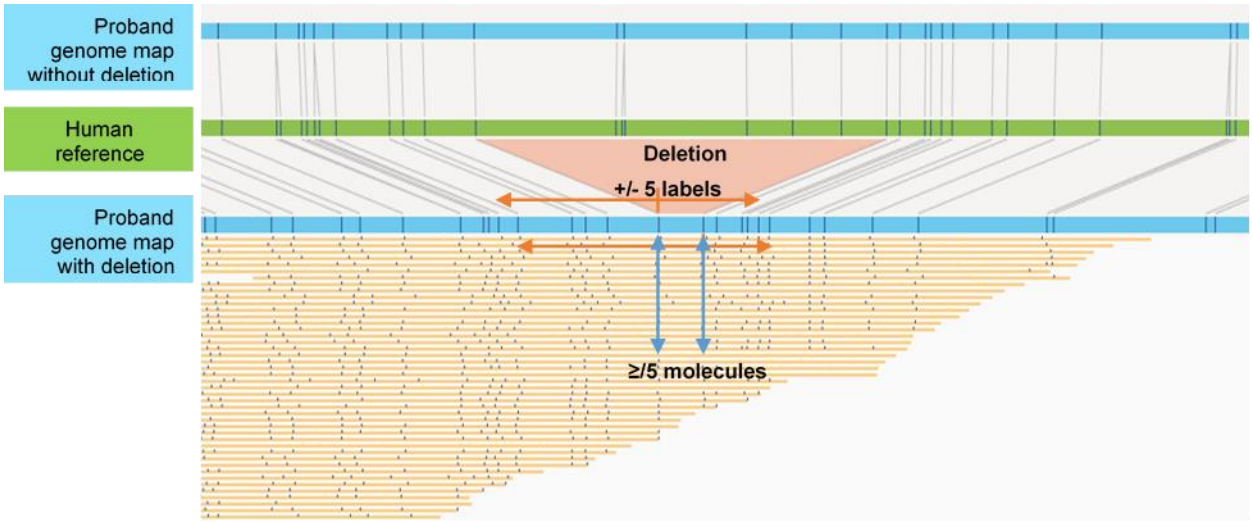


Figure 2 – Visual concept of molecule check in DLE-1 data. Every breakpoint of a SV will be checked for molecule support. That is, by default at least 5 molecules are required to span two labels around each breakpoint. Failure to have the required number of molecules at either breakpoint implies that the variant allele is not present among the molecules.

Strategy for deriving molecule coverage cutoff recommendations for nickase

NOTE: This section is only suitable for nickase data, such as BspQI or BssSI. We sought to determine optimal cutoffs and to maximize our ability to differentiate between TP (True Positive) and FP (False Positive) SV calls. Taking advantage of simulated genomes with known events, we estimated the number of molecules that would support a TP SV call and the number of molecules that would support a FP SV call (due to false positive alignment).

We created two genomes (called Genome A and Genome B for ease of discussion) that contained two sets of simulated random insertions and deletions. We aligned Genome A and Genome B molecules to Genome B genome maps in order to check whether the molecules supported insertion and deletion calls detected in Genome B (the calls were pre-filtered to eliminate false calls). The expectation was that Genome A molecules would align poorly to Genome B SV regions because the two genomes contained distinct sets of insertions and deletions. Genome A molecules may align due to false positive alignment, but this is expected to be rare. Genome B molecules would align well to Genome B SV regions.

The count distribution for FP SV calls is represented by the distribution of the number of Genome A-to-Genome B alignments for each Genome B SV; similarly, the distribution for TP SV calls is represented by the number of Genome B-to-Genome B alignments. Based on the TP and FP distributions, we constructed a Receiver Operating Characteristic (ROC) curve, identified the threshold corresponding to the breakeven point (where sensitivity was equal to Positive Predictive Value (PPV), and computed the expected sensitivity and PPV for the threshold. In this context, sensitivity refers to how likely a TP call is confirmed, and PPV refers to how likely a FP call is rejected.

The distribution was expected to be impacted by the input molecule coverage; different input coverage levels were tested. Also, the procedure was applied for other SV types. We further stratified the analysis based on the size of the insertion and deletion calls. Those bigger than 5 kbp were considered large.

We performed a linear regression analysis and based on the equation (Cutoff = -0.3 + 0.13*Input coverage[1]) from the fit, users could compute the optimal cutoff across SV types based on the input coverage (Figure 3 and Table 1). Smaller insertions and deletions appeared to behave differently and were excluded from the regression. Thus, we encourage users to be cautious when using the recommended cutoff and validate the results for those calls.

[1] Estimated during the assembly process and recorded in assembly informatics report.

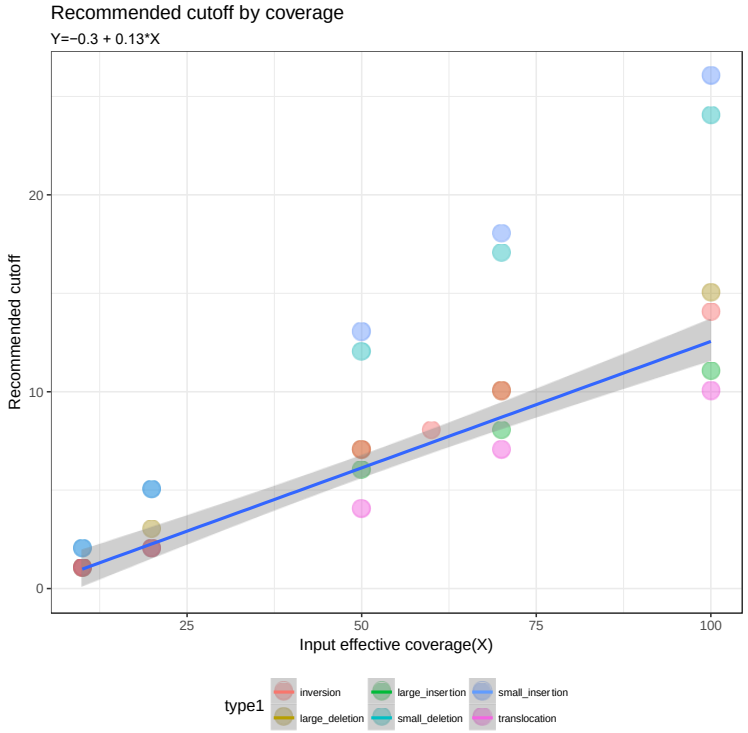


Figure 3 – Molecule check cutoff by effective coverage[2]. The individual cutoffs for each SV type and at each coverage level are based on the breakeven point according to sensitivity and PPV values. Linear regression was performed by pooling data from large insertions and deletions, inversion, and translocations. The shaded area in grey represents the 95% confidence interval for the regression fit; the solid blue line represents the recommended molecule number cutoff. The regression equation correlating the input effective coverage (Variable X) and the recommended cutoff (Variable Y) is shown at the top.

[2] Estimated during the assembly process and recorded in assembly informatics report.

Table 1 – Expected performance based on the recommended cutoffs. This table illustrates the sensitivity and PPV of SV confirmation by molecule check. Large and small insertions and deletions are > or ≤ 5 kbp, respectively. Inversions and translocations were not size-stratified.

SV type	Coverage[3] (X)	Recommended cutoff	Molecule confirmation sensitivity	Molecule confirmation PPV
Deletions > 5 kbp	50	6	0.99	0.94
	70	9	0.99	0.96
	100	13	0.99	0.97
Deletions ≤ 5 kbp	50	6	0.99	0.71
	70	9	0.98	0.76
	100	13	0.99	0.77
Insertions > 5 kbp	50	6	0.97	0.98
	70	9	0.98	0.99
	100	13	0.98	0.99
Insertions ≤ 5 kbp	50	6	0.98	0.72
	70	9	0.99	0.76
	100	13	0.99	0.77
Inversions	50	6	0.99	0.98
	70	9	0.99	0.99
	100	13	1.00	0.99
Translocations	50	6	0.99	0.98
	70	9	0.99	0.99
	100	13	1.00	0.99

Suppose that 105X, 80X, and 75X (effective coverage against hg19) of data was collected for the proband, mother and father, respectively. We recommend users to apply the linear regression equation of Figure 3 ($Y = -0.3 + 0.13X$) to determine the molecule check cutoffs. Here, they are 13.35, 10.1 and 9.45, respectively. These values should be input as parameters to the variant annotation pipeline. If the molecule coverages of all three samples at a proband's SV are higher than the cutoffs, then that variant is deemed to be present in all three genomes.

Input files and parameters

The input to the variant annotation pipeline is a single parameter text file, which contains the **full path** of necessary assembly or SV files, as well as the cutoffs. Template files for command line usage containing default parameters are as follows:

trio analysis: variant_annotation_param_db.txt
dual analysis: variant_annotation_param_dual_db.txt

[3] Estimated during the assembly process and recorded in assembly informatics report.

single analysis: variant_annotation_param_single_db.txt

The following are some of the files and parameters needed for a trio analysis. Note that the parameters required for single and dual analyses are subsets of the trio analysis.

Files needed from the proband:

1) The SV SMAP file (e.g., exp_refineFinal1_merged_filter_inversions.smap)
2) The SV alignment XMAP file (e.g., exp_refineFinal1_merged.xmap)
3) The genome map assembly CMAP file (e.g., EXP_REFINEFINAL1.cmap)
4) The molecule BNX file preferably from auto noise (e.g., autoNoise1_rescaled.bnx)
5) The molecule noise parameter ERRBIN file (e.g., autoNoise1.errbin)
6) CNV calls (optional - e.g. cnv_calls_exp.txt)
7) CNV stats (optional – e.g. cnv_chr_stats.txt)

Files needed from the parents:

1) The SV SMAP file
2) The molecule BNX file preferably from auto noise
3) The molecule noise parameter ERRBIN file
4) CNV calls (optional)
5) CNV stats (optional)

Control sample SV database file: preprocessed database files are available for hg19, hg38, and mm10 and for the *de novo* assembly pipeline and Rare Variant Pipeline.

Gene BED file: gene BED files are available for commonly studied species.

DGV SV database file: when the sample of interest is a human sample (hg19 or hg38), then the SV calls would be compared against the Database of Genomic Variants (DGV) SVs. The DGV supporting variants (release date 2020-02-25) were obtained from http://dgv.tcag.ca/dgv/app/downloads?ref=GRCh37/hg19.

Criteria of comparison:

1) Insertion: any overlap with DGV gains

2) Deletion: position overlap by at least 50% of both Bionano calls and DGV loss calls (specified by the parameter ins_del_size_percent_similarity)
3) Duplication: position overlap by at least 50% of both Bionano calls and DGV gain calls (specified by the parameter duplication_size_percent_similarity)
4) Inversion breakpoint: position overlap +/- 50kb (specified by the parameter inversion_position_overlap)
5) Translocation: none

General SV overlap criteria:

1) ins_del_position_overlap
   The maximum distance in basepairs allowed between two insertions or two deletions when looking for overlap (default 10,000 bp)

2) ins_del_size_percent_similarity
   The minimum percent size similarity required to confirm two insertions or two deletions (default 50%)

3) inversion_position_overlap

   The maximum distance in basepairs allowed between two inversion breakpoints when looking for overlap (default 50,000 bp)

4) translocation_position_overlap
   The maximum distance in basepairs allowed between two translocation breakpoints when looking for overlap (default 50,000 bp)

Running Variant Annotation Pipeline

Variant Annotation Pipeline can be run with a parameters file as input. Bionano Access will automatically create these input files according to user input. Example run commands:

trio analysis: perl variant_annotation.pl variant_annotation_param_db.txt
dual analysis: perl variant_annotation_dual.pl variant_annotation_param_dual_db.txt
single analysis: perl variant_annotation_single.pl variant_annotation_param_single_db.txt

Expected output

Results from the Variant Annotation Pipeline are written to an annotated SMAP file with annotation for each SV detected in the sample of focus and an annotated version of CNV output file (cnv_calls_exp_annotation-results.txt). The output file formats are similar to the standard Bionano SMAP and CNV files with additional annotation columns appended. Refer to SV Annotation Pipeline File Format Specification Sheet (PN 30168) and

Copy Number Variant Annotation Pipeline Format (PN 3046) for descriptions of both formats. Below are recommendations on how to filter for rare and putative *de novo* (proband-specific) variants in a trio analysis. Finally, four BED files, one for each variant type detected in the sample of focus, are also created, and these files can be readily uploaded to external genome browsers such as the UCSC Genome Browser.

CNV Annotation

Solve 3.7 adds annotation for copy number gains and losses. CNV annotation is performed as part of the main variant annotation pipeline if CNV input files are supplied. CNVs are annotated with nearest and overlapping gene(s), links to the CNV region in the UCSC genome browser and ISCN notation. Human samples are cross-referenced with the Database of Genomic Variants (DGV). As described above, CNV input files for all samples include:

1) CNV calls (cnv_calls_exp.txt)

2) CNV stats (e.g. cnv_chr_stats.txt)

Recommendations on selecting rare SVs in the results of a trio analysis

A rare variant is defined as a variant that is present in no more than 1% of the samples in the Bionano control sample SV database.

Table 2 shows the columns and the recommended cutoffs for both one- and two-enzyme workflows. For the two-enzyme workflow, the recommended cutoffs are the same, except that the cutoffs have to be applied to each enzyme individually. The expected output after applying these filters is a subset of rare SVs.

Table 2 – Recommendations on selecting rare insertions, deletions, duplications, inversion breakpoints and translocations in the results of a trio analysis of a one-enzyme workflow Note that for the two-enzyme workflow, the cutoffs should be applied to each enzyme individually.

Insertion and deletion		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Present_in_%_of_BNG_control_samples	≥ 0 and ≤ 1	Low % occurrence in control SV database
Present_in_%_of_BNG_control_samples_with_the_same_enzyme	≥ 0 and ≤ 1	Low % occurrence among database samples labelled using the same enzyme
Found_in_self_molecules	≠ “no”	Supported by self molecules
Confidence	≥ 0.5	High confidence
Type	Contains “ins” or “del”	Type is insertion or deletion

Type	Not contain “nbase”	Does not overlap N-base gaps

Duplication		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Present_in_%_of_BNG_control_samples	≥ 0 and ≤ 1	Low % occurrence in control SV database
Present_in_%_of_BNG_control_samples_with_the_same_enzyme	≥ 0 and ≤ 1	Low % occurrence among database samples labelled using the same enzyme
Found_in_self_molecules	≠ “no”	Supported by self molecules
Type	Contains “dup”	Type is duplication, duplication_split, or duplication_inverted

Inversion breakpoint		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Present_in_%_of_BNG_control_samples **Present_in_%_of_BNG_control_samples**	≥ 0 and ≤ 1	Low % occurrence in control SV database
Present_in_%_of_BNG_control_samples_with_the_same_enzyme	≥ 0 and ≤ 1	Low % occurrence among database samples labelled using the same enzyme
Found_in_self_molecules	≠ “no”	Supported by self molecules
Fail_assembly_chimeric_score	≠ "fail"	Not chimeric assembly
Confidence	≥ 0.7	High confidence
Type	Contains “inversion”	Type is inversion
Type	Not contain “partial”	No partial breakpoint

Translocation		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Present_in_%_of_BNG_control_samples **Present_in_%_of_BNG_control_samples**	≥ 0 and ≤ 1	Low % occurrence in control SV database
Present_in_%_of_BNG_control_samples_with_the_same_enzyme	≥ 0 and ≤ 1	Low % occurrence among database samples labelled using the same enzyme
Found_in_self_molecules	≠ “no”	Found in self molecules
Fail_assembly_chimeric_score	≠ “fail”	Not chimeric assembly
Confidence	≥ 0.15 for intrachromosomal ≥ 0.65 for interchromosomal	High confidence
Type	Contains “trans”	Type is translocation
Type	Not contain “common” and not contain “segdupe”	Not common translocation and not associated with segmental duplication

Recommendations on selecting for putative *de novo* SVs in the results of a trio analysis

A *de novo* variant is defined as proband-specific, thus not present in the parents. Table 3 shows the columns and the recommended cutoffs for both one- and two-enzyme workflows. For the two-enzyme workflow, the recommended cutoffs are the same, except that the cutoffs have to be applied to each enzyme individually. The expected output of applying these filters is a subset of putative *de novo* SV.

Table 3 – Recommendations on selecting for putative *de novo* (proband-specific) insertions, deletions, inversion breakpoints and translocations based on results of a trio analysis of an one-enzyme workflow. Note for the two-enzyme workflow, the cutoffs should be applied to each enzyme individually.

Insertion and deletion		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Found_in_self_molecules	≠ “no”	Supported by self molecules
Found_in_parents_assemblies	“none” or “-“	Not found in parents’ assemblies
Found_in_parents_molecules	“none" or “-“	Not found in parents' molecules
Confidence	≥ 0	High confidence
Type	Contains “ins” or “del”	Type is insertion or deletion
Type	Not contain “nbase”	Does not overlap N-base gap

Duplication		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Found_in_self_molecules	≠ “no”	Supported by self molecules
Found_in_parents_assemblies	“none” or “-“	Not found in parents’ assemblies
Found_in_parents_molecules	“none" or “-“	Not found in parents' molecules
Type	Contains “dup”	Type is duplication, duplication_split or duplication_inverted

Inversion breakpoint		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Found_in_self_molecules	≠ “no”	Supported by self molecules
Fail_assembly_chimeric_score	≠ “fail”	Not chimeric assembly
Found_in_parents_assemblies	“none” or “-“	Not found in parents’ assemblies
Found_in_parents_molecules	“none” or “-“	Not found in parents' molecules
Confidence	≥ 0.7	High confidence
Type	Contains “inversion”	Type is inversion
Type	Not contain “partial”	No partial breakpoint

Translocation		
Column in one-enzyme workflow	**Cutoff**	**Effect**
Found_in_self_molecules	≠ “no”	Supported by self molecules
Fail_assembly_chimeric_score	≠ “fail”	Not chimeric assembly

Found_in_parents_assemblies	"none" or "-"	Not found in parents' assemblies
Found_in_parents_molecules	"none" or "-"	Not in parents' molecules
Confidence	≥ 0.05 for intrachromosomal ≥ 0.05 for interchromosomal	High confidence
Type	Contains "trans"	Type is translocation
Type	Not contain "common" and not contain "segdupe"	Not common translocation and not associated with segmental duplication

Control SV database

By default, to estimate the population frequency of the proband's variants, the pipeline queries variants against Bionano's human control sample SV database containing variants collected from ethnically-diverse mapped human genomes with no reported disease phenotypes. This database is essential for estimating variant frequency as the calls stored in the database are found by Bionano's genome mapping, which can discover variants unidentifiable by other technologies. Separate databases are available for the *de novo* assembly pipeline and Rare Variant Pipeline (RVP), and for hg19 and hg38.

Currently, 179 DLE-1 datasets are included in the database and are classified according to https://www.internationalgenome.org/faq/which-populations-are-part-your-study/:

Classification	Count
African (AFR)	45
Admixed American (AMR)	16
East Asian (EAS)	17
European (EUR)	44
South Asian (SAS)	15
Unknown	43

Note that we incorporated data from 30 COVID-positive samples. They had no known severe genetic conditions otherwise. Contact Bionano Support if you require custom control databases that do not include the COVID-positive samples.

Control databases for mouse are also available for the *de novo* assembly pipeline and RVP (based on the mm10 reference). We incorporated data from 11 B6 mice, only one of which was considered a true control. Other mice had various phenotypes. Users need to exercise caution when using the mouse control data and when interpreting the annotation results.

If custom control data are available, the user can generate a custom control SV database using the following command-line script (provided in the Bionano Solve package):

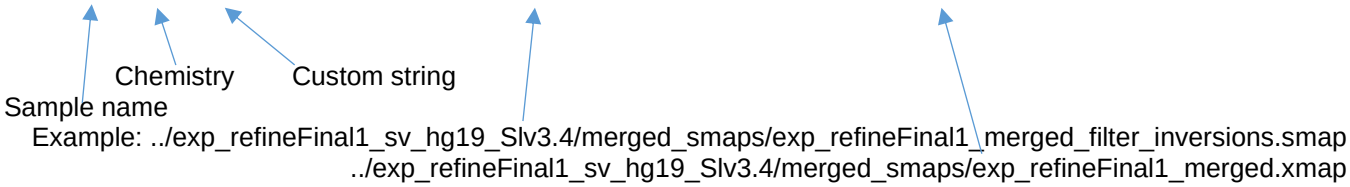
```
perl config/ctrl_sv_create_custom_db.pl <ctrl_sv_list_file> <reference_build> <exclude_sample_list>
<type_of_sv_algorithm>
```

For example, to generate a control SV database for hg19, please use the following command-line:

```
perl config/ctrl_sv_create_custom_db.pl /path/to/data/ctrl_sv_list.txt hg19
/path/to/data/exclude_sample_list.txt moleSV
```

The *ctrl_sv_list* file is a tab-delimited text file in the following format:

Sample1_bspqi_sop	/path/to/smap	/path/to/xmap
Sample2_bspqi_sop	/path/to/smap	/path/to/xmap
Sample2_dle1_sop	/path/to/smap	/path/to/xmap



In case of a sample occurring multiple times with different labeling chemistries, the sample will only be counted once.

The *exclude_sample_list* file is a text file with a list of samples in the same format as the first column above. An empty text file may be passed as input if no samples should be excluded. The following is an example of a valid file:

Sample2_bspqi_sop
Sample2_dle1_sop

The parameter *type_of_sv_algorithm* is optional and is used to provide a custom string that will be appended to the output filename.

Running the example command will generate a database file called *ctrl_sv_db_anonymize_hg19.txt*, and a file containing the control SV list annotated with the mapping between the original sample name and the anonymized sample name, called *ctrl_sv_list_anonymize_hg19.txt*.

Finally, edit the appropriate variant annotation parameter file to point to the database file generated above. Now, the variant annotation pipeline can be run as described previously.

Note: In previous Bionano Solve versions, the control database was split into multiple files. In the current version, a single control database file is expected.

Merging SV databases

To merge two control SV databases, e.g. to add additional human samples to the control SV database provided by Bionano, run the following command-line.

```
perl config/ctrl_sv_merge_dbs.pl.pl <ctrl_sv_db_file1> <ctrl_sv_db_file2> <ctrl_sv_db_merged>
```

For example, to merge a custom control SV database with the Bionano database, run the following command-line.

```
perl config/ctrl_sv_merge_dbs.pl config/data/homo_sapiens/ctrl_sv_db_anonymize_hg38.txt
/path/to/data_custom/ctrl_sv_db_anonymize_hg38.txt
/path/to/data_merged/ctrl_sv_db_merged_anonymize_hg38.txt
```

Note: it is assumed that there is no sample overlap between the two databases. If overlapping sample names are found, then the sample names in the second database will be renamed. Running the script will generate a new database file, as well as a file containing the mapping between the original sample name and the anonymized sample name.

Custom gene annotations

To enable species-specific gene annotation, the user will need to provide a file containing known genes that follows the BED format specifications (link). The file should be tab-delimited and describe a single gene per row, with the chromosomes provided as integer cmap IDs, the gene name in the fourth column, and no header. For animals, gene positions and gene names can be downloaded from the UCSC genome browser; for most species, the required information will be found in fields chromosome, txStart, txEnd, and name2. For plants, reference genome FASTA files and gene annotation GFF3 files are often available from Gramene (link), and GFF3 files can be converted to a BED file using a tool such as BEDOPS gff2bed (link).

To convert the first column of the BED file from chromosome IDs to cmap IDs, users will need to provide the key file generated during *in-silico* digestion and run the following command-line script:

```
perl /path/to/variant_annotation/config/bed_map_chr.pl <bed_file> <key_file>
```

The mapped BED file output will be generated in the same directory as the input BED file. The overlap_database parameter in the variant annotation parameter file will need to point to this file.

An example script for creating a mapped BED file for a model plant species is provided below.

```
gff2bed < Sorghum_bicolor.Sorghum_bicolor_NCBIv3.45.chr.gff3 | awk '$8 == "gene"' >
```

```
Sorghum_bicolor.Sorghum_bicolor_NCBIv3.45.chr.bed

perl /path/to/variant_annotation/config/bed_map_chr.pl
Sorghum_bicolor.Sorghum_bicolor_NCBIv3.45.chr.bed
sorghum_bicolor.sorghum_bicolor_ncbiv3.dna.toplevel_DLE-1_0kb_0labels_key.txt
```

ISCN Annotation

As of Solve 3.7, variant annotation includes notation for each variant using the International System for Cytogenomic Nomenclature (ISCN) recommendations. Bionano Genomics is working with standards bodies to define an official nomenclature for optical genome mapping data, however we have adopted interim conventions that approximate most cases for OGM structural variants based on existing guidelines at https://varnomen.hgvs.org. Following are examples of variant types and how they are represented in Solve 3.7.

All ISCN annotations produced by the Variant Annotation Pipeline will include a prefix that specifies optical genome mapping as the technology and the reference genome that variants are being reported against. Example:

ogm[GRCh38]

is used to specify variants reported on the hg38/GRCh38 reference genome.

CNV loss and gain

CNV losses and gains are reported with the chromosome and cytoband of the CNV along with the basepair coordinates for the start and end of the event. Copy number will be specified as 'x' followed by the detected copy number; fractional copy number state is notated as a range between the lower and upper bounds of the copy number. If a fractional copy number of a gain is greater than 4, a generalized amplification notatation of 'amp' is used. Examples:

variant	description
ogm[GRCh38] 1p36.33(710374_711817)x0~1	CNV loss with fractional copy number state between 0 and 1
ogm[GRCh38] 1p36.33(710374_711817)x2~3	CNV gain with a fractional copy number between 2 and 3
ogm[GRCh38] 1p36.33(710374_711817)amp	CNV gain with a fractional copy number greater than 4

Deletions

Deletion structural variants use the same notation as CNV loss with the difference being that copy number state is reported as an integer rather than as fractional copy number range. Examples:

variant	description
ogm[GRCh38] 1p36.33(710374_711817)x1	Heterozygous deletion
ogm[GRCh38] 1p36.33(710374_711817)x0	Homozygous deletion

Duplications

Duplication structural variants are notated with the chromosome and cytoband positions of the duplication along with the keyword ‘dup’ and the detected copy number of the variant. In the case of inverted duplications, the cytoband positions will be listed with the end position of the duplication first followed by the start position.

variant	description
ogm[GRCh38] dup(8)(q24.21q24.22)x3	Duplication on chromosome 8 duplication with a reference start position at q24.21, reference stop of q24.22 and with a copy number state of 3
ogm[GRCh38] dup(8)(q24.22q24.21)x3	Inverted duplication in the same position as above. Inverted position is notated by reference start (q24.22) listed first.

Insertions

Insertion variants are notated with the keyword ‘ins’ followed by chromosome and cytoband position of the

insertion. The insertion of unknown sequence at the noted position is indicated by a question mark ('?'). Examples:

variant	description
ogm[GRCh38] ins(4;?)(q28.3;?)	Insertion of unknown sequence to chromosome 4 at cytoband q28.3

Inversions

Inversion variants are listed with the keyword 'inv' followed by the chromosome and cytoband positions of the start and end of the inversion.

variant	description
ogm[GRCh38] inv(6)(p25.3q16.1)	Inversion on chromosome 6 from p25.3 to q16.1

Inter-chromosomal translocations

Translocations between different chromosomes are notated with listed with the keyword 't' followed by the two chromosomes and cytoband positions of each translocation breakpoint. Example:

variant	description
ogm[GRCh38] t(2;11)(p25.1;p15.2)	Translocation between chromosome 2 p25.1 and chromosome 11 p15.2

Intra-chromosomal fusions

Fusions between regions on the same chromsome are notated with listed with the keyword 'fus' followed by the chromosome and cytoband positions of each translocation breakpoint.

variant	description
ogm[GRCh38] fus(4;4)(28.3;p22.1)	Fusion between chromosome 4 p22.1 and q28.3

Unsupported cases

The Solve 3.7 ISCN model does not currently support notation for chromosome aneuploidy or regions of AOH/LOH.

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) 888-7663**
Website	**www.bionanogenomics.com/support**

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