OGM File Format Specification Sheet

DOCUMENT NUMBER:
CG-00045

DOCUMENT REVISION:
C

Effective Date:
04/24/2024

bionano

Table of Contents

Revision History

REVISION	NOTES
A	Initial release
B	Updated for SMAP v.091 with information on ordering of RefContig ids
C	Updated to refer to CG-30110 for details on confidence score translation

OGM File Specifications

BNX v1.3 File Format Specification

The Bionano® BNX file is a raw data view of molecule and label information and quality scores per channel identified during a run or runs if data from multiple runs are merged. BNX v1.3 supports one or two label channels (colors). This section provides descriptions, with examples, of the BNX header and molecule information block format of the file.

The BNX file presents the general molecule information (data) in two sections: the BNX information header, which describes the specific format of the data; and the molecule information block, which contains the data values. For easy readability, BNX files can be opened in Excel or in any tab-delimited, text-based editor. However, raw BNX files can be gigabytes in size and may not be viewable on a computer with limited memory. BNX v1.3 is not supported in IrysView® or on Irys instruments. Only Saphyr® instruments can generate BNX v1.3 files. Previously generated BNX v1.2 files may be imported into Bionano Access®.

When molecules in a BNX file are aligned to a reference (typically in a CMAP file), the alignment tool, RefAligner, checks that the BNX label motif header line is consistent with that in the CMAP. This is to ensure that the input data are compatible, and that the resulting alignment data can be interpreted.

UNDERSTANDING RUN HEADERS

Run headers in the BNX file are used to identify the origin of the molecule data collected. This includes the software versions, label channels, enzyme recognition sites, information about the origin of molecules, and information about the molecule data that will follow.

For data generated on the Saphyr instrument, each cohort has a separate Run Data line. A cohort is defined as a subset of a flowcell for a given scan. For example, since ICS v4.8, each flowcell is divided into eight cohorts. The number of Run Data lines will be eight times the number of scans for that flowcell. Each of the Run Data lines is given a unique sequential RunID. Each molecule listed in the BNX file has a RunID associated with it, so each molecule can be traced back to its original cohort. The scan number of all molecules from a Saphyr Chip is always 1. Downstream Bionano software can continue to assume that all molecules sharing the same RunID and scan number share the same scaling factor. For this to work, it is not necessary to identify precisely which cohort is associated with a specific RunID. However, to support customer applications and to trace back individual molecules to the image data, the specific cohort location on each chip can be identified from the SourceFolder field on the Run Data line. In general, the **SourceFolder** field identifies the folder of images from which the molecules associated with this Run Data line are derived. **NOTE**: these image folders are deleted by default immediately after extracting the molecule information, unless the user requests that they be preserved and set it up before the run starts. For Saphyr Chips, this field ends in a 4- or 5-digit number. The lowest order digit indicates the cohort within the bank, the second lowest digit indicates the bank, and the high order 2 or 3 digits indicate the scan number. In the Run Data example below, the data came from scan 17, bank 3 and cohort 2.

FORMAT

The BNX file contains the following sections:

- BNX header
  - # BNX File Version
  - # Label Channels
  - # Nickase Recognition Site 1 and color
  - # Nickase Recognition Site 2 and color (optional)
  - # Software Version (optional)
  - # Bases per pixel (optional)
  - # Number of molecules (optional)
  - # rh
  - # Run Data
  - # 0h
  - # 0f
  - # 1h
  - # 1f
  - # 2h (optional)
  - # 2f (optional)
  - # Qh
  - # Qf
- Molecule information block
  - Molecule header (as defined in #0h)
  - Label position for Channel 1 (as defined in #1h)
  - Quality Score 1 (label SNR) for Channel 1
  - Quality Score 2 (label average intensity) for Channel 1
  - Label position for Channel 2 (as defined in #2h)
  - Quality Score 1 (label SNR) for Channel 2
  - Quality Score 2 (label average intensity) for Channel 2

NOTE: The data are broken down into sections. Each section is a group of data rows associated with a single molecule and is then repeated for all data.

EXAMPLE SINGLE COLOR BNX FILE

```
# BNX File Version:     1.3
# Software Version:     4.8.19085.2, merco 1.3.8041.8044 Tue Oct 30 14:23:20 PDT 2018
# Label Channels:       1
# Nickase Recognition Site 1:   CTTAAG;BNGFLGR001
# Bases per Pixel:      375
#rh     SourceFolder    InstrumentSerial        Time    NanoChannelPixelsPerScan        StretchFactor
BasesPerPixel   NumberofScans   ChipId  Flowcell        LabelSNRFilterType      MinMoleculeLength
MinLabelSNR1    RunId
# Run Data       /home/bionano/access/local/SAPHYR_F09/2017-06/SN_CDLERX6NPNRX7NWU,Run_6c1a79ef-141a-
4096-9d82-314634ab0357/FC1/Cohort1732       SAPHYR_F09      2019-05-07 01:24:35 PM  68819821
```

```
0.83    375     1       chips,SN_CDLERX6NPNRX7NWU,Run_6c1a79ef-141a-4096-9d82-314634ab0357,0    1
dynamic 15      3       1
…
# Number of Molecules:  12978111
#0h     LabelChannel    MoleculeId      Length  AvgIntensity    SNR     NumberofLabels
OriginalMoleculeId      ScanNumber      ScanDirection   ChipId  Flowcell        RunId   Column
StartFOV        StartX  StartY  EndFOV  EndX    EndY
#0f     int     int     int     float   float   int     int     int     int     string  int     int
int     int     int     int     int     int     int
#1h     LabelChannel    LabelPosition[N]
#1f     int     int
#Qh     QualityScoreID  QualityScores[N]
#Qf     string  float
# Quality Score QX11: SNR for channel 1
# Quality Score QX12: Intensity for channel 1
```

EXAMPLE TWO COLOR BNX FILE

```
# BNX File Version:     1.3
# Software Version:     4.9.19225.1, merco 1.3.8041.8044 Tue Oct 30 14:23:20 PDT 2018
# Label Channels:       2
# Nickase Recognition Site 1:   GCTCTTC;BNGFLRD001
# Nickase Recognition Site 2:   CTTAAG;BNGFLGR001
# Bases per Pixel:      375
#rh     SourceFolder    InstrumentSerial        Time    NanoChannelPixelsPerScan        StretchFactor
BasesPerPixel   NumberofScans   ChipId  Flowcell        LabelSNRFilterType      MinMoleculeLength
MinLabelSNR1    RunId
# Run Data              SAPHYR_D-BETA2  2019-08-14 06:25:57 PM  68819821        0.83    375     1
chips,SN_4MSLRTONPOAXZNWU,Run_2ca21a30-2c55-4cda-92f9-a6b6cd169ea0,0    2       dynamic 15      3
1
…
# Number of Molecules:  7198645
#0h     LabelChannel    MoleculeId      Length  AvgIntensity    SNR     NumberofLabels
OriginalMoleculeId      ScanNumber      ScanDirection   ChipId  Flowcell        RunId   Column
StartFOV        StartX  StartY  EndFOV  EndX    EndY
#0f     int     int     int     float   float   int     int     int     int     string  int     int
int     int     int     int     int     int     int
#1h     LabelChannel    LabelPosition[N]
#1f     int     int
#2h     LabelChannel    LabelPosition[N]
#2f     int     int

#Qh     QualityScoreID  QualityScores[N]
#Qf     string  float
# Quality Score QX11: SNR for channel 1
# Quality Score QX12: Intensity for channel 1
# Quality Score QX21: SNR for channel 2
# Quality Score QX22: Intensity for channel 2
```

HEADER SPECIFICATIONS

Table 1. BNX header fields - overview

Header Line Tag	Header Line Description
# BNX File Version:	Indicates the version of the BNX file.
# Label Channels:	Defines the number of label channels.
# Nickase Recognition Site 1:	Comma separated list of enzyme recognition sequences for channel 1 followed by semicolon and channel 1 color (or fluorescent dye part number or name). There can be no spaces in this string. Color or dye is optional. This can also refer to the label recognition sequence for a non-nicking enzyme (i.e., DLE-1).

# Nickase Recognition Site 2:	Comma separated list of enzyme recognition sequences for channel 2 followed by semicolon and channel 2 color or dye. There can be no spaces in this string. Color is optional. This can also refer to the label recognition sequence for a non-nicking enzyme (i.e., DLE-1).
# Software Version:	Indicates software tool and version that generated the BNX file. In merged BNXs, this field is optional. Tools that handle BNX may not fill this in.
# bases per pixel	This field is optional. Tools that handle BNX may not fill this in.
# Number of Molecules	This field is optional. Tools that handle BNX may not fill this in.
# rh	Defines required tab-separated columns for headers in the # Run Data section.
# Run Data	Defines data that conforms to tab-separated headers specified in # rh.
#0h	Defines required columns for backbone data rows (rows labeled "0").
#0f	Defines format for columns in backbone data rows (rows labeled "0").
#1h	Description of fields in label channel 1.
#1f	Defines format for data in label channel 1.
#2h	Description of fields in label channel 2.
#2f	Defines format for data in label channel 2.
#Qh	Description of a quality score fields (ID and scores).
#Qf	Description of the quality score line format.

HEADER SPECIFICATION DETAILS

Tables 2-16 provide the BNX header's descriptions (including any specific formatting, limitations, and requirements) and examples.

Table 2. BNX file version header fields

# BNX File Version	
Header	# BNX File Version:
Description	Indicates the version of the BNX file.
Example	# BNX File Version:<TAB>1.3

Table 3. Label channel header fields

# Label Channels	
Header	# Label Channels:
Description	Defines the number of label channels. Available values: [1,2].
Example	# Label Channels:<TAB>2

Table 4. Nickase recognition site 1 fields

# Nickase Recognition Site 1	
Header	# Nickase Recognition Site 1:
Description	Sequence of enzyme for channel 1, optionally followed by the laser color or dye name or dye part number for the channel and separated by a semicolon. This field is case insensitive.
Example	# Nickase Recognition Site 1:<TAB>CCTCAGC;BNGFLGR001

Table 5. Nickase recognition site 2 fields

# Nickase Recognition Site 2	
Header	# Nickase Recognition Site 2:
Description	Sequence of enzyme for channel 2, optionally followed by the laser color or dye name or dye part number for the channel and separated by a semicolon. This field is case insensitive.
Example	# Nickase Recognition Site 2:<TAB> CCTCAGC;BNGFLGR001

NOTE: If no color arguments are given, Access assumes site 1 is green and site 2 is red. A color argument may be the string “unknown.”

Table 6. Software version fields

# Software Version	
Header	# Software Version:
Description	Indicates the detection software and version that generated the BNX. For merged runs, this header may be dropped or replaced by a comment with the command used to generate the merged BNX.
Example	# Software Version:<TAB> 4.8.19085.2, merco 1.3.8041.8044 Tue Oct 30 14:23:20 PDT 2018

Table 7. Required header fields definition.

# rh		
Header	# rh	
Description	Description of the required tab-separated columns for headers specified in # Run Data rows. See Understanding BNX Headers above:	
	SourceFolder	The original images folder for the run
	InstrumentSerial	Instrument name or identifier (or UNKNOWN)
	Time	Beginning run time (or 1999, if unknown)
	NanoChannelPixelsPerScan	Effective pixels per nanochannel
	StretchFactor	Chip stretch factor
	BasesPerPixel	Estimated or calculated from stretch factor
	NumberofScans	Number of scans performed in the run
	ChipId	This field is used to identify the chip that was used. It may contain comma separated values with no spaces. It must contain ',' followed by an integer on the end that uniquely identifies the chip run (UID).
	Flowcell	Flowcell number
	LabelSNRFilterType	Type of filter applied to label SNR for individual run (static or dynamic)
	MinMoleculeLength	Minimum molecule length filter for individual run (in kilobases)
	MinLabelSNR	Value of minimum label SNR for individual run
	RunId	Unique run Id (optional if there is only one Run Data line, with a value of 1 implied)
Example	#rh<TAB>SourceFolder<TAB>InstrumentSerial<TAB>Time<TAB> NanoChannelPixelsPerScan<TAB>StretchFactor<TAB>BasesPerPixel <TAB>NumberofScans<TAB>ChipId<TAB>Flowcell<TAB> LabelSNRFilterType<TAB>MinMoleculeLength<TAB>MinLabelSNR <TAB>RunId	

NOTE: #rh entries must be in the order listed. Versions 1.0 and 1.1 only have the first nine entries up to Flowcell.

Table 8. Run data header fields definition

# Run Data	
Header	# Run Data
Description	Defines data that conforms to the tab-separated headers specified in #rh. Multiple Run Data lines will exist when BNX combines data from multiple irys runs or multiple Saphyr cohorts. ChipId entry must be a comma-delimited list of values, with at least one comma, and the last value must be an integer. Some values may be the string “UNKNOWN.”
Example	# Run Data <TAB>D:\Data\EColi\2012-08\PL-Ecoli\Detect_Molecules <TAB>SAPHYR_F09<TAB> 2019-05-07 01:24:35 PM<TAB> 68819821<TAB>0.83<TAB>375<TAB>1<TAB> chips,SN_CDLERX6NPNRX7NWU,Run_6c1a79ef-141a-4096-9d82-314634ab0357,0<TAB>2<TAB>dynamic<TAB>15<TAB>3<TAB>1

Table 9. Molecule backbone header fields definition

#0h		
Header	#0h	
Description	Description of the required tab-separated columns for molecule backbone data rows (rows labeled “0”):	
	LabelChannel	Channel 0 corresponds to the molecule backbone channel
	MoleculeId	Molecule ID
	Length	Molecule length in bases
	AvgIntensity	Average backbone molecule intensity
	SNR	Average backbone molecule SNR
	NumberofLabels	Total number of labels detected for this molecule on all channels
	OriginalMoleculeId	When multiple runs are merged, molecule IDs will be re-numbered and reported in the #0h MoleculeId column for each run. This field reports the molecule ID from the original BNX file. For a single run, this field reports the value reported in the #0hMoleculeId header line tag.
	ScanNumber	Scan number from the run. For Saphyr Chips, this value is always ‘1’.
	ScanDirection:	Description:

#0h		
	-1	Unknown
	0	Forward
	1	Backward
	ChipId	Serial number of the chip of the run
	Flowcell	Flow cell number
	RunId	A positive integer that identifies the # Run Data header line: A value N corresponds to the Nth header line.
	Column	The imaging column in which the molecule was detected.
	StartFOV	The field of view within the column in which the molecule begins.
	StartX	The X coordinate in which the molecule begins.
	StartY	The Y coordinate in which the molecule begins.
	EndFOV	The field of view within the column in which the molecule ends.
	EndX	The X coordinate in which the molecule ends.
	EndY	The Y coordinate in which the molecule ends.
	GlobalScanNumber	Unique global ID computed from RunId and ScanNumber. (Not present in all BNX files)
Example	#0h<TAB>LabelChannel<TAB>MoleculeId<TAB>Length<TAB>SNR <TAB> AvgIntensity<TAB>NumberofLabels<TAB>OriginalMoleculeIdA <TAB>ScanNumber<TAB>ScanDirection<TAB>ChipId <TAB>Flowcell<TAB>RunId<TAB>Column<TAB>StartFOV<TAB>StartX<TAB>StartY<TAB>EndFOV<TAB>EndX<TAB>EndY<TAB>GlobsalScanNumber	

NOTE: BNX 1.0 has only a single "# Run Data" header and does not include the RunId or any subsequent field.

NOTE: The 7 fields from Column to EndY may be absent from the header, in which case all molecule headers will be missing this information. This may be the case for early versions of BNX 1.3 data.

Table 10. Molecule backbone header fields format definition

#0f	
Header	#0f
Description	Defines the format for columns in backbone data rows (rows labeled "0").
Example	#0f<TAB>int<TAB>int<TAB>int<TAB>float<TAB>float<TAB>int <TAB>int<TAB>int<TAB>int<TAB>string<TAB>int<TAB>int<TAB>int<TAB>int<TAB>int<TAB>int<TAB>int<TAB>int<TAB>int

Table 11. Label channel 1 header definition

# 1h	
Header	#1h
Description	Description of the fields in label channel 1. For LabelPosition[N], [N] is variable [1...N]
Example	#1h<TAB>LabelChannel<TAB>LabelPosition[N]

Table 12. Label channel 1 header fields format definition

#1f	
Header	#1f
Description	Defines the format for data in label channel 1.
Example	#1f<TAB>int<TAB>int

Table 13. Label channel 2 header fields definition

# 2h	
Header	#2h
Description	Description of the fields in label channel 2. For LabelPosition[N], [N] is variable [1...N].
Example	#2h<TAB>LabelChannel<TAB>LabelPosition[N]

Table 14. Label channel 2 header fields format definition

#2f	
Header	#2f
Description	Defines the format for data in label channel 2.
Example	#2f<TAB>int<TAB>int

Table 15. Quality header fields definition

#Qh	
Header	#Qh
Description	Description of the fields in QX<m><n>; where QualityScoreID is QX[*m*=channel][*n*=score sequence]; and QualityScores[N], where [N] is variable [1...N].
Example	#Qh<TAB>QualityScoreID<TAB>QualityScores[N]

Table 16. Quality header fields format definition

#Qf	
Header	#Qf
Description	Defines the format for data in QX<m><n>.
Example	#Qf<TAB>string<TAB>float

MOLECULE INFORMATION BLOCK SPECIFICATION

Molecule information block rows are prefixed by the backbone (0) and channel (1 or 2) designations and the quality designation for labels (e.g., QX11, QX12, QX21, QX22). Each molecule information block adheres to the following convention:

- Molecule information block
  - Backbone data row (values for header #0h)
  - Channel 1 data row (values for header #1h)
  - Channel 1 quality score field ID and values for SNR (values for header # Quality Score QX11:)
  - Channel 1 quality score field ID and values for average intensity (values for header # Quality Score QX12:)
  - Channel 2 data row (values for header #2h)
  - Channel 2 quality score field ID and values for SNR (values for header # Quality Score QX21:)

- Channel 2 quality score field ID and values for average intensity (values for header # Quality Score QX22:)

NOTE: A molecule information block has the data rows for a single molecule. Molecule information blocks are repeated for each molecule's data.

Molecule Information Block Specification Details

Tables 17-23 provide the BNX molecule information block descriptions (including any specific formatting, limitations, and requirements) and examples.

Table 17. Backbone data row

0	
Header	Backbone data row (values for header #0h)
Description	Required
Example	0<TAB>1<TAB>898875<TAB>15795.7<TAB>438.111<TAB>224<TAB> 55226<TAB>1<TAB>-1<TAB>chips,SN_CDLERX6NPNRX7NWU,Run_6c1a79ef-141a-4096-9d82- 314634ab0357,0<TAB>1<TAB>1<TAB>32<TAB>1<TAB>103<TAB>217<TAB>2<TAB>100<TAB>760

Table 18. Channel 1 data row

1	
Header	Channel 1 data row (values for header #1h)
Description	Required
Example	1<TAB>15416<TAB>20446<TAB>22515<TAB>32389 <TAB>36148 38079<TAB>47018 <TAB>54062<TAB> 76852<TAB>80806<TAB>94432<TAB>96995<TAB>99773 <TAB>122543<TAB>138487<TAB>150504<TAB>153590 <TAB>154790<TAB>159545….

Table 19. Channel 1 quality scores – label SNR

QX11	
Header	QX11
Description	Channel 1 quality score field ID and values (label SNR)
Example	QX11<TAB>39.3287<TAB>6.2613<TAB>6.9517<TAB>15.9162<TAB>17.8695

Table 20. Channel 1 quality scores – average label intensity

QX12	
Header	QX12
Description	Channel 1 quality score field ID and values (label average intensity)
Example	QX12<TAB>0.1233<TAB>0.0431<TAB>0.0471<TAB>0.0641<TAB>0.0767<TAB>0.0489

Table 21. Channel 2 data row

2	
Header	Channel 2 data row (values for header #2h)
Description	Required
Example	2<TAB>15416<TAB>20446<TAB>22515<TAB>32389 <TAB>36148 38079<TAB>47018 <TAB>54062<TAB> 76852<TAB>80806<TAB>94432<TAB>96995<TAB>99773 <TAB>122543<TAB>138487<TAB>150504<TAB>153590 <TAB>154790<TAB>159545

Table 22. Channel 2 quality scores – label SNR

QX21	
Header	QX21
Description	Channel 2 quality score field ID and values (label SNR)
Example	QX21<TAB>39.3287<TAB>6.2613<TAB>6.9517<TAB>15.9162<TAB>17.8695

Table 23. Channel2 quality scores – average label intensity

QX22	
Header	QX22
Description	Channel 2 quality score field ID and values (average label intensity)
Example	QX22<TAB>0.1233<TAB>0.0431<TAB>0.0471<TAB>0.0641<TAB>0.0767<TAB>0.0489

MOLECULE INFORMATION BLOCK 2 COLOR EXAMPLE

```
0       183     241500  4321.49 80.67   66      12967   1       -1      chips,SN_CDLERX6NPNRX7NW
U,Run_6c1a79ef-141a-4096-9d82-314634ab0357,0    1       1       4       4       405     813
4       399     1456
1       8431    15317   18719   20337   22378   23461   26511   38148   43239   44940   46152
48618   50418   55040   58304   62986   65026   67284   73302   75041   79238   83161   86625
90744   93541   98059   100275  103807  106615  108569  113238  115342  119415  123874  127147
128836  132386  136947  139303  143796  146698  148737  153243  156795  158939  163425  166261
169608  171133  173298  180030  182590  187506  189308  191170  194120  199178  205613  212731
215843  218206  221351  224684  229039  233971  238808  241500

2       372     30454   33754   53263   68284   73545   75769   80890   85650   87569   121950
132445  134124  161983  164214  183781  205368  208468  213711  227576  233488  239277  244203
252851  256800  264283  273708  277056  279563  285553  290158  297320  310045  311687  326948
336634  339543  344722  348611  352278  357012  358578  362169  364573  366917  368454  372675
376614  385439  387369  389689  395302  399662  405791  409123  416338  418047  423318  425846
429461  438289  442342  444530  464606  467240  472875

QX11    10.08348        6.33244 6.25445 6.27364 4.90548 6.82648 3.94645 20.34334        8.32105
7.80591 9.32729 11.34582        9.53998 7.06546 11.75305        11.87323        5.79447 6.69021
8.47028 12.94812        14.33600        8.09798 21.59392        5.61965 21.69473        21.09607
19.16816        9.50938 34.36756        11.66572        30.09792        41.02215
73.67329        17.74068        29.42975        39.03440        28.29641        28.13300
21.77220        30.53836        12.88604        5.35750 41.67753        17.60822        16.60717
7.78603 13.97649        11.65518        21.52665        15.09029        17.83440
16.97878        14.53676        12.57979        15.61269        15.79564        7.96846 28.43458
16.56446        4.34261 8.42204 14.80289        15.17917        12.42019        4.78720 6.89824

QX21    7.64803 9.44464 14.81698        8.44297 12.80191        12.53211        9.51094 6.65061
10.88102        9.18963 11.08927        10.47844        18.68086        5.27135 8.22451 9.80301
7.41648 15.81695        14.93037        13.41137        12.47227        9.26461 32.88548
19.00489        13.21570        10.60119        10.19845        14.87359        14.36357
9.82766 14.30509        20.68182        8.14954 17.95187        10.97401        16.51873
10.16308        33.78429        17.33000        5.89374 21.92048        8.46575 22.60254
18.22422        18.64907        11.70299        23.17835        9.13149 8.74133 14.90674
8.69611 9.19728 9.79332 15.49938        15.17399        12.41004        13.67709        17.42826
22.22139        9.82052 11.87555        22.77458        12.69888        8.88651 12.80701

QX12    583.12  366.20  361.69  362.80  283.68  394.77  228.22  1176.44 481.20  451.41  539.39
656.12  551.69  408.59  679.67  686.62  335.09  386.89  489.83  748.78  829.04  468.30  1248.76
324.98  1254.59 1219.97 1108.48 549.92  1987.45 674.62  1740.54 2372.28 4260.47 1025.93 1701.90
2257.33 1636.36 1626.91 1259.07 1766.01 745.19  309.82  2410.18 1018.27 960.38  450.26  808.25
674.01  1244.87 872.66  1031.35 981.87  840.65  727.48  902.87  913.45  460.81  1644.35 957.91
251.13  487.04  856.04  877.80  718.25  276.84  398.92

QX22    449.87  555.55  871.56  496.63  753.03  737.16  559.45  391.20  640.04  540.55  652.29
616.36  1098.84 310.07  483.78  576.63  436.25  930.38  878.23  788.88  733.64  544.96  1934.38
1117.90 777.37  623.58  599.89  874.89  844.89  578.08  841.45  1216.54 479.37  1055.96 645.51
971.66  597.81  1987.25 1019.38 346.68  1289.40 497.97  1329.52 1071.98 1096.97 688.39  1363.39
537.13  514.18  876.84  511.52  541.00  576.06  911.70  892.56  729.98  804.51  1025.16 1307.10
577.66  698.54  1339.64 746.97  522.72  753.33
```

NOTE: For data generated on the Irys instrument, each chip run of a single flow cell is given a single Run Data line with only one RunID in the header. It describes data across all scans of a single flowcell; individual molecules identify the scan number they originated from. If multiple flowcells, multiple chip runs (reuse of the same chip), or multiple chips are run, and the BNX files are merged, there will be more than one Run Data line, each with a unique sequential RunID. Individual molecules now identify both the RunID and scan number they originated from. Downstream Bionano software assumes that all BNX molecules sharing the same RunID and scan number share the same scaling factor.

CMAP v0.1 File Format Specification

The Bionano CMAP file is a data file which provides location information for label sites within a genome map or an *in silico* digestion of reference or sequence data. The CMAP is a tab-delimited text-based file and can be opened in Excel for easy readability. Although the CMAP most commonly contains data from FASTA reference digestion

and a *de novo* Assembly, a BNX file (which typically contains raw molecule data) can also be converted to a CMAP.

A CMAP file contains two sections:

- the CMAP information header, which describes the format of the data, and
- the map information block, which contains the data values. This file format specification sheet provides descriptions, with examples, of the CMAP header and map information block format of the file.

FORMAT

The CMAP file contains the following sections:

- CMAP header
  - # CMAP File Version
  - # Label Channels
  - # Nickase Recognition Site
  - # Number of Consensus Maps
  - #h
  - #f
- Map information block
  - First label site in map
  - Next label site in map (repeated for all label sites)
  - Last label site is end of map.

HEADER SPECIFICATIONS

Header rows are prefixed by the pound sign (#). "*" Denotes required header line tags.

Table 24. CMAP header fields – overview

Header Line Tag	Header Line Description
# CMAP File Version	Version of CMAP*
# Label Channels	The number of label channels (integer)*
# Nickase Recognition Site 1	Comma separated list of label motif recognition sequences for channel 1 followed by semicolon and channel 1 color. There can be no spaces in this string. Color is optional. This can also refer to the label recognition sequence for a non-nicking enzyme (i.e., DLE-1).
# Number of Consensus Maps	The total number of consensus genome maps in the CMAP file (integer)
#h	The columns for each data row
#f	The numerical data type for each data column

HEADER SPECIFICATION DETAILS

Tables 25-31 provide the CMAP header's descriptions (including any specific formatting, limitations, and requirements) and examples. CMAP currently supports up to two label channels. Additional columns may be present but are not defined. Certain columns may be absent in earlier versions of the CMAP format.

Table 25. CMAP File version header fields

# CMAP File Version	
Header	# CMAP File Version:
Description	Version of CMAP, auto generated.
Example	# CMAP File Version:<TAB>0.2

Table 26. Label channel header fields

# Label Channels	
Header	# Label Channels:
Description	The number of label channels (integer). Available values are: [1, 2].
Example	# Label Channels:<TAB>1

Table 27. Nickase recognition site 1 header fields

# Nickase Recognition Site 1	
Header	# Nickase Recognition Site 1:
Description	Comma separated list of label motif recognition sequences for channel 1 followed by semicolon and channel 1 color. There can be no spaces in this string. Color is optional. This can also refer to the label recognition sequence for a non-nicking enzyme (i.e., DLE-1).
Example	# Nickase Recognition Site 1:<TAB>gctcttc,cctcagc;green_01

Table 28. Nickase Recognition Site 2 header fields

# Nickase Recognition Site 2 (optional)	
Header	# Nickase Recognition Site 2:

# Nickase Recognition Site 2 (optional)	
Description	Comma separated list of label motif recognition sequences for channel 2 followed by semicolon and channel 2 color. There can be no spaces in this string. Color is optional. This can also refer to the label recognition sequence for a non-nicking enzyme (i.e., DLE-1).
Example	# Nickase Recognition Site 2:<TAB>cctcagc;red_01

Table 29. Number of consensus maps header fields

# Number of Consensus Maps	
Header	# Number of Consensus Maps:
Description	The total number of consensus genome maps in the CMAP file (integer).
Example	# Number of Consensus Maps:<TAB>81
#h	

Table 30. Header fields definition

# h		
Header	#h	
Description	Defines the columns for each data row in #h rows:	
	CMapId	Map ID, ordinal number
	ContigLength	Map length in basepairs
	NumSites	Total number of label sites in map
	SiteID	Label ID, ordinal number
	LabelChannel	Label channel of label sites The last LabelChannel field of each map is always 0.
	Position	Position of label on map [0-based from map start] in basepairs
	StdDev	Theoretical standard deviation in bases of label site interval between the current and next site. Value will be 0 for FASTA digestion of a reference.

# h		
	Coverage	Weighted coverage of aligned molecules across an interval. The values may be fractional. How much an alignment to a map contributes to the weighted coverage depends on whether the alignment is unique to that map. If a molecule aligns equally well to two maps, it will contribute 0.5 in coverage to each of the maps. Since Solve 3.5, coverage refers to the interval between the current and the next label. The header of the CMAP now includes a comment on whether coverage is based on the interval between labels.
	Occurrence	Number of molecules with a label aligned to a given label. This is also weighed. If a molecule spans an interval but its labels do not align to the label of interest, it will contribute to coverage but not occurrence. Generally, occurrence should be less than coverage. However, this may not be true in corner cases.
	ChimQuality	Percent of molecules that align to both sides of the label out of all molecules that align on either side near this label.
	SegDupL	See Note.
	SegDupR	See Note.
	FragileL	See Note.
	FragileR	See Note.
	OutlierFrac	Fraction of number of molecules with internal outlier which overlaps this site.
	ChimNorm	This is the quantity (N1+N2+N3) described below.
	Mask	64-bit hex value: each bit flags a possible attribute for each label. See below for currently used flags.
Example	#h CmapId<TAB>ContigLength<TAB>NumSites<TAB>SiteID<TAB> LabelChannel<TAB>Position<TAB>StdDev<TAB>Coverage<TAB>Occurrence<TAB> ChimQuality<TAB>SegDupL<TAB>SegDupR<TAB>FragileL<TAB>FragileR<TAB>OutlierFrac<TAB>ChimNorm <TAB>Mask	

Table 31. Header fields format definition

#f	
Header	#f
Description	Defines the numerical data type for each data column.
Example	#f<TAB>int<TAB>float<TAB>int<TAB>int<TAB>int<TAB>float<TAB>float<TAB>float <TAB>float<TAB>float<TAB>float<TAB>float<TAB>float<TAB>float<TAB>float<TAB>float<TAB>Hex

GENOME MAP QUALITY SCORES

Based on the molecule-to-genome map alignment, we compute the following genome map quality scores for each label of the genome map. In **Figure 1**, N1-N5 are representative molecules which align to the genome map. They all contain the label of interest, for which the score is computed. The numbers may be fractional, since coverage is typically weighted (a molecule that aligns to two regions of the genome gets a weight of 0.5 for each location).

- First, the following quantities are computed for each label in the genome map. For N2 through N5, up to two missing and one extra label are allowed next to the label for which the score is computed.
- N1: the number of molecules which align over both left and right flanks. Each flank is 36 kbp (see CovTrimLen in refineFinal section of optArguments.xml)
- N2/N3: number of molecules which align on one flank, but have an endoutlier (unaligned portion, shown in red below) which spans the second flank.
- N4/N5: same as N2/N3 but no endoutlier is present.
- The genome map quality scores are defined by the following (they are expressed as fractions):
  - ChimQuality = N1/(N1+N2+N3)
  - SegDupL = N2/(N1+N2+N3)
  - SegDupR = N3/(N1+N2+N3)
  - FragileL = N4/(coverage)
  - FragileR = N5/(coverage)

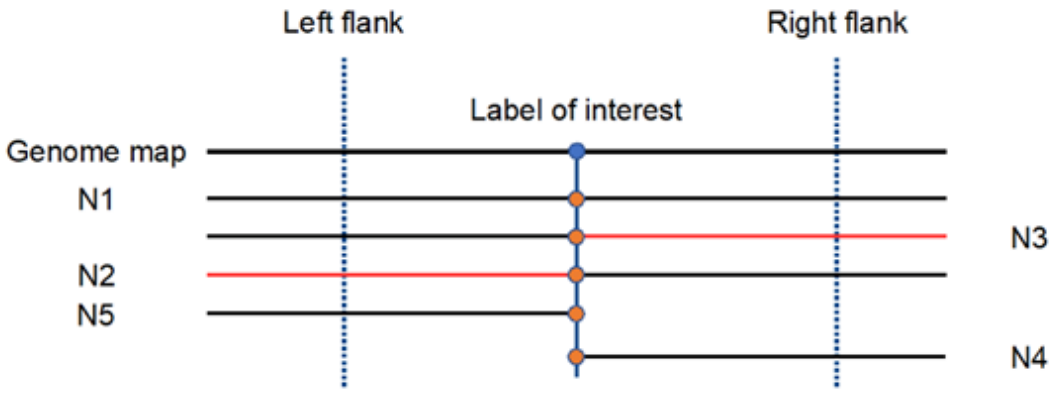

Figure 1. Genome map label attributes encoded in Mask column.

The following bits are currently used to flag attributes of labels in the genome map (the default bit value is 0):

- Bit 0 (Value 1) is set for end labels to mark a broken end when a genome map is broken at an ambiguous CMPR (complex multi-path region). See *Bionano Solve Theory of Operation: Structural Variant Calling* (CG-30110) for detail.
- Bit 1 (Value 2) is set for end labels to mark the end of an alternate allele map (like assembly graph bubbles). Typically, such a map consists of the alternate region plus 300 kbp at either end of the shared homozygous region. They are generated when haplotype-aware assembly is performed. For a haplotype-aware assembly, most of these alternate maps are assigned to one of the two allelic maps, but any alternate maps that could not be assigned to either of the two dominant alleles will have their ends marked with this Bit 1.

- Bit 2 (Value 4) is set for all labels in a region that is a suspected CMPR (complex multi-path region): these are genome map regions that closely resemble regions in other genome maps (other than the matching allelic map pair) and could be mediated by segmental duplications. By default, such regions over 140 kbp are likely to be broken with both pieces sharing the CMPR region and the broken ends marked with Bit 0 (see above). We also provide the option to not break them. Currently, CMPR regions under 140 kbp are NOT broken but marked with Bit 2.
- Bit 3 (Value 8) is used in Hybrid Scaffold to mark ends derived from a Bionano genome map.
- Bit 4 (Value 16 OR 0x10) is used in Hybrid Scaffold to mark ends derived from an NGS sequence (or sequence scaffold).

NOTE: a Hybrid Scaffold can have one end derived from a Bionano genome map and the other end derived from an NGS sequence. During Hybrid Scaffold, Mask bits 3 and 4 are used to prevent merging scaffold ends that are both derived from an NGS sequence.

GENOME MAP INFORMATION BLOCK SPECIFICATION

The data is grouped per genome map represented in the CMAP file. Each group starts with the first label site, followed by each label site in the map, and ends with the map length. Each group follows this convention:

- Genome map information block
  - First label site in map
  - Next label site in map (repeated for all label sites)
  - End location of genome map. This position encodes the final coordinates of the map.

EXAMPLE

```
# CMAP File Version:  0.2 # Label Channels: 1
# Nickase Recognition Site 1: cttaag;green_01 # Number of Consensus Maps: 459
# Values corresponding to intervals (StdDev, HapDelta) refer to the interval between current site and
next site
#h CMapId     ContigLength    NumSites       SiteID  LabelChannel    Position       StdDev  Coverage
      Occurrence     ChimQuality SegDupL SegDupR FragileL  FragileR       OutlierFrac    ChimNorm
      Mask
#f int float int int int float float float float float float float float float float float float Hex
  182   58474736.7  1023  1  1   58820.9    13   13.  -    -    -    3.  0.00  0.00  -     0
                    5           35.4       .5   5    1.0  1.0  1.0  63             1.00
                                                     0    0    0
  182   58474736.7  1023  2  1   70333.1    13   13.  -    -    -    0.  0.00  0.00  -     0
                    5           36.5       .6   6    1.0  1.0  1.0  00             1.00
                                                     0    0    0
  182   58474736.7  1023  3  1   84845.3    14   13.  -    -    -    0.  0.00  0.00  -     0
                    5           30.7       .6   7    1.0  1.0  1.0  31             1.00
                                                     0    0    0
  182   58474736.7  1023  4  1   87470.9    14   14.  -    -    -    0.  0.00  0.04  -     0
                    5           36.7       .6   6    1.0  1.0  1.0  31             1.00
                                                     0    0    0
  182   58474736.7  1023  5  1   106152.6   34   14   14.  -1.00 -1.00  0.00  0.00  0.10  -
                    5                      .9   .6   5    -1.00                          1.00
                                                                                        0
  182   58474736.7  1023  6  1   119659.3   30   14   13.  100.00 0.00  0.03  0.00  0.00  13.6
                    5                      .7   .6   2    0.00                           4 0
  182   58474736.7  1023  7  1   122330.5   29   15   14.  96.66 3.34    5.33  0.00  0.00  14.9
                    5                      .9   .1   1    0.00                           9 0
```

XMAP v0.2 File Format Specification

The Bionano .xmap file is a cross-comparison between two maps (see **Figure 2**), and version 0.2 is backwards-compatible with XMAP v0.1. The .xmap file reports the comparison derived from the alignment between an

anchor .cmap file and a query .cmap file. The data line displays the map start and end coordinates and the locations of the labels on the map using a tab-delimited text-based file.

The .xmap file presents the information in two sections: the XMAP information header, which describes the specific format of the data; and the map alignment information block, which contains the data rows. This section provides descriptions, with examples, of the XMAP header and map alignment information block format of the file. When imported into Bionano Access, the .xmap file is automatically filtered and ready for downstream analysis. XMAP files can be opened in Excel for easy readability or in any tab-delimited, text-based editor.

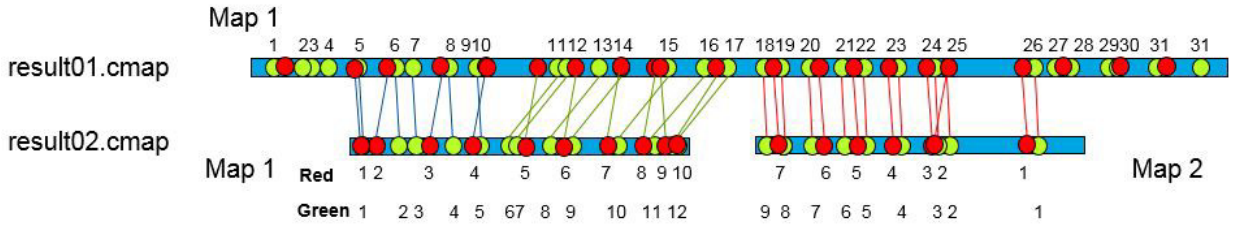

Figure 2. Visualization of alignment between two CMAPs

FORMAT

The XMAP file contains the following sections:

- XMAP header
  - # XMAP File Version:
  - # Reference Maps From:
  - # Query Maps From:
  - #h
  - #f
- Alignment information block (each row as defined by the column headers in #h)
  - After the 3 IDs, is the first alignment of a reference map label to a query map label with orientation and confidence.
  - Then the (pseudo)-CIGAR string displays in HitEnum, followed by query and reference length and label channel.
  - The final string shows the alignment label site in the map and is repeated for all label sites indexed per label color channel.

HEADER SPECIFICATIONS

Header rows are prefixed by the pound sign (#) and follow an order, as seen in **Table 32**. **NOTE**: *Denotes the required header line tags for Bionano Access to read an XMAP file. Required header line tags must be present and must precede the Alignment Information Block to read an XMAP file. Header lines which are not required are optional and may be omitted. **Denotes the required header line tags for importing into Bionano Access.

Table 32. XMAP header fields - overview

Header Line Tag	Header Line Description
# XMAP File Version:	Indicates the version of the XMAP file*
# Reference Maps From:	A string denoting the path to the corresponding r.cmap*
# Query Maps From:	A string denoting the path to the corresponding q.cmap*
#h	Defines the columns for each data row**
#f	Defines the numerical data type for each data column**

HEADER SPECIFICATION DETAILS

Table 33 provides the XMAP header's descriptions (including any specific formatting, limitations, and requirements) and examples.

Table 33. XMAP File version header fields

# XMAP File Version	
Header	# XMAP File Version:
Description	Indicates the version of the XMAP file.
Example	# XMAP File Version:<TAB>0.2

Table 34. Reference Maps header fields

# Reference Maps From	
Header	# Reference Maps From:
Description	A string denoting the path to the corresponding reference map, which contains the reference or anchor data.
Example	# Reference Maps From:<TAB> ExampleXmap_r.cmap

Table 35. Query maps header fields

# Query Maps From	
Header	# Query Maps From:
Description	A string denoting the path to the corresponding query map, which contains the query data.
Example	# Query Maps From:<TAB>ExampleXmap_q.cmap

Table 36. Header fields definition

Description	Description of the required tab-separated columns in #h:	
#h		
Header	#h	
	XmapEntryID	A unique line number for the data lines in the XMAP file. **NOTE**: For 2-color, the XmapEntryID will begin with the number 2.
	QryContigID	Map ID of query map (Contig ID from .cmap file for query)
	RefContigID	Map ID of the reference map from the .cmap reference file (the .cmap file may contain multiple reference maps). **NOTE**: RefContigIDs must be integers, but they need not be sequential.
	QryStartPos	Coordinates of the first aligned label on the query map (Start position of hit on query map)
	QryEndPos	Coordinates of the last aligned label on the query map (Stop position of hit on query map)
	RefStartPos	Coordinates of the first aligned label on the reference or anchor map
	RefEndPos	Coordinates of the last aligned label on the reference or anchor map
	Orientation	The relative orientation of the query map relative to the reference: forward (+) or reverse (-). The convention is that the reference is always positive orientation, so if the query aligns in reverse, it is shown as having negative (-) orientation. **NOTE**: For 2-color, the orientation will be the same.

Description	Description of the required tab-separated columns in #h:	
	Confidence	Statistical Confidence of result: Negative Log10 of p-value of alignment (without Bonferroni Correction for multiple experiments). **NOTE**: For 2-color, the confidence number is the combined confidence of the alignment for both colors.
	HitEnum	Pseudo-CIGAR string representing matches (M), insertions (I), or deletions. (D) of label sites with respect to the reference or anchor map. Count begins at the leftmost anchor label of that color. **NOTE**: When 2 or more anchor sites resolve into a single query site, only the rightmost anchor site is shown matched with the query site and the leftmost associated anchor sites are shown as deletions.
	QryLen	Length of query map from q.cmap.
	RefLen	Length of reference map from r.cmap.
	LabelChannel	Color channel of alignment from cmap files. • For 1-color data, LabelChannel is 1. • For 2-color data: • Using -usecolor N, the LabelChannel is *N* (*N* = 1 or 2), and there is only one XMAP entry per alignment for the color channel specified by N. Without -usecolor N, LabelChannel is 1 or 2. In this case, there are two XMAP entries (two lines), one for each color channel.
	Alignment	Indices of the aligned site ID pairs. (When the query orientation is reversed ("-"), the query IDs are in descending order.) Count begins at the leftmost anchor label of that color. **NOTE**: When two sites in the reference align with the same site in the query, it is an indication that the two sites in the reference failed to resolve. Alignment provides a view of aligned pairs which would normally be ignored by HitEnum (CIGAR string).
Example	#h XmapEntryID<TAB>QryContigID<TAB> RefContigID <TAB>QryStartPos<TAB>QryEndPos<TAB>RefStartPos <TAB>RefEndPos<TAB>Orientation<TAB>Confidence <TAB>HitEnum <TAB>QryLen<TAB>RefLen<TAB>LabelChannel<TAB>Alignment	

NOTE: Additional columns may be present but are not defined by XMAP Version 0.2.

Table 37. Header fields format definition

#f	
Header	#f
Description	Defines the numerical data type for each data column.
Example	#f int<TAB>int<TAB>int<TAB>float<TAB>float<TAB> float<TAB>float<TAB>string<TAB>float<TAB>string

ALIGNMENT INFORMATION BLOCK SPECIFICATION

The data is grouped such that each data row (see **Figure 3**) represents an alignment between one reference or anchor map and one query contig/map.

NOTE: Depending on the parameters used during alignment, there may be more than one alignment for each reference and/or query map. Even for the same query and reference ID pair, different local alignments (alignments of the same region of the query with different regions of the reference) can be present.

```
# XMAP File Vers                 0.2
# Label Channels:                2
# Reference Map twocolor_r.cmap
# Query Maps Frc twocolor_q.cmap
#h XmapEntryID  QryContigID        RefContig QryStartPo QryEndPo RefStartPo RefEndPo Orientatic Confidenc HitEnum  QryLen   RefLen   LabelChar Alignment
#f int          int                int       float      float    float      float    string     float     string    float    float    int       string
          2     106000337          1         1897.5     173467.3 9749       178762   +          23.74     1M1I2M1I  180295.8 5139685           1 (1,1)(2,3)(3,4)(4,6)(6,7)(7,9)(8,10)(9,14)(10,15)
          3     106000337          1         2920.1     177124.8 10905      183608   +          23.74     2M1D2M2   180295.8 5139685           2 (3,1)(4,2)(6,3)(7,4)(9,5)(10,5)(13,6)(14,6)(15,7)(16,8)(17,9)(18,10)(19,11)(20,12)(22,13)(23,14)(26,15)
          4     101000333          1         3923.1     196137.5 9749       276866   +          23.13     1M3I1M1I  198762.3 5139685           1 (1,3)(2,7)(3,9)(7,11)(9,15)(10,18)(12,20)(13,21)(15,25)(16,29)
          5     101000333          1         2674.4     194285   82367      275962   +          23.13     4M1D8M1   198762.3 5139685           2 (17,1)(18,2)(19,3)(20,4)(22,5)(23,6)(24,7)(25,8)(26,9)(27,10)(28,11)(29,12)(31,13)(32,14)(33,15)(34,16
          6     101000298          1         137939.7   98.7     28180      165956   -          19.25     1M1I1M1I  164521.1 5139685           1 (2,12)(3,10)(4,8)(5,7)(6,6)(7,4)(8,3)(9,1)
          7     101000298          1         146990     2080.3   18846      163433   -          19.25     1M1D1M2   164521.1 5139685           2 (5,16)(7,15)(9,14)(10,14)(11,13)(13,11)(14,11)(15,10)(16,9)(17,8)(18,7)(19,6)(20,5)(22,3)(23,2)(24,1)
          8     101000145          1         134855.2   485.1    50004      183657   -          18.89     1M1I3M1I  171060.3 5139685           1 (3,13)(4,11)(5,10)(6,9)(7,7)(8,6)(9,5)(10,2)(11,1)
          9     101000145          1         165750.6   156.8    10905      183608   -          18.89     5M1D2M3   171060.3 5139685           2 (3,15)(4,14)(5,13)(6,12)(7,11)(9,10)(10,9)(12,8)(14,8)(15,7)(17,6)(18,5)(19,4)(22,3)(23,2)(27,1)
         10     101000188          1         168377.7   4425.8   50004      264214   -          13.41     1M1I1M3I  183211.1 5139685           1 (3,24)(4,22)(5,18)(6,17)(7,15)(11,11)(12,10)(13,9)(14,5)(15,1)
         11     101000188          1         143176.5   9913.7   124414     257851   -          13.41     1M1D3M1   183211.1 5139685           2 (20,16)(22,15)(23,14)(24,13)(26,12)(27,11)(28,10)(30,8)(31,8)(32,6)(33,5)(34,4)(36,3)(37,3)(38,2)(39,1
         12     104000242          1         1979.9     267311.5 50004      314303   +          38.35     5M1I2M1I  273601.5 5139685           1 (3,1)(4,2)(5,3)(6,4)(7,5)(8,7)(9,8)(11,9)(12,10)(13,11)(14,14)(15,16)(16,17)(19,18)(20,19)(21,20)(22,21
         13     104000242          1         11564.3    265770.1 62228      314757   +          38.35     6M1I5M1I  273601.5 5139685           2 (13,2)(14,2)(15,3)(16,4)(17,5)(18,6)(19,7)(20,9)(21,10)(22,11)(23,12)(24,13)(26,14)(27,15)(29,16)(30,1
         14     102000221          1         48638.8    162957.6 76188      187880   +          20.28     2M3I1D6M  178351.9 5139685           1 (4,1)(5,2)(7,6)(8,7)(9,8)(10,9)(11,10)(12,11)
         15     102000221          1         3089.7     142885.9 32214      168206   +          20.28     2M1D1M3   178351.9 5139685           2 (7,1)(8,2)(9,3)(10,3)(12,4)(14,4)(15,6)(16,7)(17,8)(18,9)(19,10)(20,11)(22,12)(23,13)(24,14)(26,15)
         16     103000534          1         153496.7   14711.6  89565      227268   -          27.61     2M3I1M1I  163286.2 5139685           1 (5,25)(6,24)(7,20)(8,18)(9,14)(10,11)(11,10)(12,9)(13,8)(14,4)
```

Figure 3. Results map or .xmap

SMAP v0.91 File Format Specifications

The Bionano SMAP file contains a list of structural variants (SV) detected between query maps and reference maps. Detailed information about each SV call is output in a tab-delimited, text-based format.

The SMAP file presents the information in two sections: 1) the SMAP information header, which describes the specific format of the data, and 2) the SV information block, which contains the data rows. This file format specification sheet provides descriptions, with examples, of the SMAP header and SV information block format of the file.

When the data are imported into Bionano Access, the SMAP file is automatically processed and ready for downstream analysis and visualization. SMAP files can also be opened in Excel for easy readability or in any tab-delimited, text-based editor.

FORMAT

The SMAP file contains the following sections:

- Header. Contains metadata and description of the contents in the SV information block. There are both mandatory and optional lines.
- # SMAP File Version:
- # Reference Maps From:
- # Query Maps From:
- # XMAP Entries From:
- # Confidence scores:
- # VAF:
- #h
- #f
- SV information block. The content of each row is defined by the column titles in the header line #h. It is an open-ended, tab-delimited text format with no maximum number of columns defined, but there must be correspondence between the number of columns and the column names in #h. Columns:
  - After the 4 IDs [SmapEntryID, QryContigID, RefcontigID1, and RefcontigID2] are the positions for query and reference of each SV [QryStartPos, QryEndPos, RefStartPos, RefEndPos].
  - Followed by the confidence scores of the SV calls and their corresponding SV type [Confidence, Type].
  - Then the fields XmapIDs provide the ID of XMAP entries used to make the SV calls.
  - The next field LinkID references a SmapEntryID when linked SMAP entries define a single SV call, especially for inversion breakpoints.
  - The Idxs [QryStartIdx, QryEndIdx, RefStartIdx, and RefEndIdx] are the label indices for query and reference labels for each SV call.
  - Other columns are pipeline and postprocessing dependent. For example, Zygosity, Genotype, GenotypeGroup, RawConfidence, RawConfidenceLeft, RawConfidenceRight, RawConfidenceCenter, SVsize, SVfreq, orientation, and VAF.

HEADER SPECIFICATIONS

Table 38 lists header rows which are prefixed by the pound sign (#).

Table 38. SMAP header fields - overview.

Header Line Tag	Header Line Description
# SMAP File Version:	Indicates the version of the SMAP file
# Reference Maps From:	A string denoting the path to the corresponding _r.cmap
# Query Maps From:	A string denoting the path to the corresponding _q.cmap
# XMAP Entries From:	A string denoting the path to the corresponding .xmap
#h	Defines the columns for each data row
#f	Defines the data type for each data column

NOTE: The above are required header line tags for Bionano Access to import SV data from an SMAP file. Required header line tags must be present and must precede the SV Information Block. Other header lines may contain auxiliary information and are optional.

OPTIONAL HEADER LINES

The confidence scores and VAF algorithms add lines to the header with details about versions and parameters used. To help with parsing, the values are stored in JSON format.

HEADER SPECIFICATION DETAILS

Tables 39-46 provide the SMAP header's descriptions (including any specific formatting, limitations, and requirements) and examples.

Table 39. SMAP File version header fields

# SMAP File Version	
Header	# SMAP File Version:
Description	Indicates the version of the SMAP file.
Example	# SMAP File Version:<TAB>0.91

Table 40. Reference Maps header fields

# Reference Maps From	
Header	# Reference Maps From:
Description	Denotes the path to the corresponding reference map, which contains the reference or anchor data.
Example	# Reference Maps From:<TAB>Example_r.cmap

Table 41. Query maps header fields

# Query Maps From	
Header	# Query Maps From:
Description	Denotes the path to the corresponding file of query maps, which contains the query data.
Example	# Query Maps From:<TAB>Example_q.cmap

Table 42. Xmap entries header fields

# Xmap Entries From	
Header	# Xmap Entries From:
Description	A string denoting the path to the corresponding xmap file, which contains information about the map alignments.
Example	# Xmap Entries From:<TAB>Example_.xmap

Table 43. Confidence scores header fields

# Confidence scores	
Header	# Confidence scores:
Description	A JSON entry recording details on confidence scoring model and parameters.
Example	`# Confidence scores: {"translocations_score": {"pipeline_name": "denovo", "model_file": "/home/bionano/models/translocations_v0.5.2.joblib", "model_version": "0.5.2", "features": "LabelOccurrenceFrac,RegionCoverageFrac,RawConfidenceLeft,RawConfidenceRight,MeanCoverage,MeanOccurrence,MeanChimQuality,MeanSegDupL,MeanSegDupR,MeanFragileL,MeanFragileR,MeanOutlierFrac,InCentromere,InSegDup,StartRegion,EndRegion,Type"}, "inversions_score": {"pipeline_name": "denovo", "model_file": "/home/bionano/models/inversions_v0.5.0.joblib", "model_version": "0.5.0", "features": "MinAlignConfidence,MinLabelsAligned,BpPositionGap,RefLabelsUnaligned,LabelsInCMPR,MeanOutlierFrac,RegionCoverageFrac,LabelOccurrenceFrac,LabelsQryInvertedRegion,RefMatchGroupsOverlap,QryMatchGroupsOverlap,Conflict,BpLowerBoundInCentromere,BpLowerBoundInSegDup,BpUpperBoundInCentromere,BpUpperBoundInSegDup,BpLowerBoundRegion,BpUpperBoundRegion"}}`

Table 44. VAF header fields

# VAF	
Header	# VAF:
Description	A JSON entry recording algorithm used for variant allele fraction
Example	`# VAF: {"algorithm for non-duplications": "BetaPriorBayesian", "version": "1.0","algorithm for duplications": "CoverageRatio", "pipeline": "de novo"}`

Table 45. Header fields definition

#h		
Header	#h	
Description	Description of the required tab-separated columns in #h:	
	SmapEntryID	A unique number for an entry in the SMAP file.
	QryContigID	Map ID of query map (Contig ID from .cmap). Both XmapID1 and XmapID2 contain alignments to this map.
	RefcontigID1	Reference contig ID (XmapID1). Map ID of the reference map from the .cmap reference file (the .cmap file may contain multiple reference maps). For interchromosomal translocations, this contig aligns to the upstream (or predominantly upstream, if overlapping) region of the query map relative to that of RefconfigID2. **NOTE:** RefContigIDs must be integers, but they need not be sequential.
	RefcontigID2	Reference contig ID (XmapID2). Map ID of the reference map from the .cmap reference file (the .cmap file may contain multiple reference maps). For interchromosomal translocations, this contig aligns to the downstream (or predominantly downstream, if overlapping) region of the query map relative to that of RefcontigID1. **NOTE:** These RefContigIDs are always the same for insertions, deletions, duplications, and inversion breakpoints.
	QryStartPos	Start of SV on the query map. It is always the case for Indels (or anytime the 2 alignment match groups are not overlapped) that QryStartPos <= QryEndPos.
	QryEndPos	End of SV on the query map.
	RefStartPos	Coordinate of reference contig ID1 aligned position (typically a site but can be between misresolved sites) which borders this SV. This position is either a start or end of XmapID1 (or the start or end of a contiguous subset of label matches in XmapID1 called a *matchgroup*, for cases where XmapID1 supports multiple SVs and cannot be so trimmed). It may correspond to either the query start position (QryStartPos) OR query end position (QryEndPos). For translocations, the match to RefStartPos borders the matchgroup aligning to the upstream (or predominantly upstream, if overlapping) region of the query map.
	RefEndPos	Coordinate of reference contig ID2 aligned position which borders this SV. This position is either a start or end of XmapID2 (or the start or end of a matchgroup, for cases where XmapID2 supports multiple SVs and cannot be so trimmed). It matches the position of either the query end position (QryEndPos) or query start position (QryStartPos). For translocations, the match to RefEndPos borders the matchgroup aligning to the downstream (or predominantly downstream, if overlapping) region of the query map.
	Confidence	Estimate of probability of being correct for insertions, deletions over 500bp, inversions, and translocations. Other SVs are given a placeholder value of '-1.00'. See Bionano Solve Theory of Operation: Structural Variant Calling (document 30110).
	Type	Type of SV (See definitions in **SV Types Definitions** below).

#h		
	XmapID1	XmapEntryID in the .xmap file of the first alignment from which this SV is derived.
	XmapID2	XmapEntryID in the .xmap file of the second alignment from which this SV is derived.
	LinkID	For some SV types, two SMAP entries may be linked using this field (e.g., inversion-partial, inversion-paired).
	QryStartIdx	Index in query map of site nearest to QryStartPos.
	QryEndIdx	Index in query map of site nearest to QryEndPos.
	RefStartIdx	Index in reference map of site nearest to RefStartPos.
	RefEndIdx	Index in reference map of site nearest to at RefEndPos.
	Zygosity	One of 'homozygous', 'heterozygous', or 'unknown' based on overlap with other SVs and alignments.
	Genotype	'1' for homozygous SVs, typically '2' for heterozygous SVs (in general number of distinct SV clusters overlapping current SV), and '-1' for unknown zygosity and SVs which are not indels or translocations
	GenotypeGroup	Indels which overlap one another and belong to the same size cluster are assigned the same group.
	RawConfidence	Minimum of next three columns for indels. '-1' for other SV types.
	RawConfidenceLeft	Confidence of alignment to the left (on reference) of indel or translocation.
	RawConfidenceRight	Confidence of alignment to the right (on reference) of indel or translocation.
	RawConfidenceCenter	Indels only: outlier confidence.
	SVsize	The estimated size of the SV (this is output for insertion, deletion, duplication, and inversion breakpoint calls.)
	SVfreq	See *Calculation of SVfreq* below.
	Orientation	This is computed only for translocation breakpoints and indicates the orientation of the pair of matchgroups supporting translocation events. It can be “+/+,” “+/-“, “-/+,” or “-/-“. The first sign corresponds to the matchgroup bordered by the (RefContigID1, RefStartPos) match and the second sign to the matchgroup bordered by the (RefContigID2, RefStopPos) match. See *Theory of Operation: Structural Variant Calling* (CG-30110) for detail.
	VAF	Variant allele fraction as calculated in Bionano Solve 3.7. See *Theory of Operation: Structural Variant Calling* (CG-30110) for details.

Table 46. Header field format definition

#f	
Header	#f
Description	Defines the numerical data type for each data column.
Example	#f int<TAB>int<TAB>int<TAB>int<TAB>float<TAB>float<TAB> float<TAB>float<TAB>float<TAB>string<TAB>int<TAB>int<TAB>int <TAB>int<TAB>int<TAB>int<TAB>int<TAB>string<TAB>int<TAB>int<TAB> float<TAB>float<TAB>float<TAB>float<TAB>float<TAB>float<TAB>string

SV TYPES DEFINITIONS

Structural variants (SVs) are defined as any significant difference between, typically, a *de novo* Assembly of Bionano molecules and a reference. The assembly pipeline includes an SV detection stage. SVs are detected either as pairs of local alignments (MatchGroups) on the genome map or within a single alignment for indels **Table 47** provides an overview of the SV types currently included in the SMAP and describes the rules by which they are classified.

Table 47. SV types currently included in the SMAP.

SV Types	Definition
insertion	Size difference which is larger on the query than on the reference. Query length - Reference length <= 5 Mbp.
insertion_nbase	Insertion with an N-base reference gap in between the insertion breakpoints covering at least 40% of the reference breakpoint interval. Must have a .bed file supplied to specify the gap.
insertion_tiny	Insertion smaller than 5% of smaller of reference or query range AND both reference and query ranges include at least two misaligned labels: possibly a balanced indel, like a small inversion.
deletion	Size difference which is larger on the reference than query.
deletion_nbase	Deletion with N-base reference gap in between the deletion breakpoints. Must have a .bed file supplied.
deletion_tiny	Deletion smaller than 5% of smaller of reference or query range AND both reference and query ranges include at least two misaligned labels: possibly a balanced indel, like a small inversion.
inversion	Two local alignments that have opposite orientation and no overlap. This is an inversion breakpoint, not a full inversion event.
inversion_paired	Two inversion events which are linked and form a full inversion. LinkID will point to other paired inversion.

SV Types	Definition
inversion_partial	Extra information about inversion events. Not an independent event. LinkID will point to an inversion, inversion_nbase, or inversion_repeat event.
inversion_nbase	Inversion with N-base reference gap in between the inversion breakpoints. Must have bed file supplied.
inversion_repeat	Inversion call in which at least one matchgroup primarily consists of a simple repeat (adjacent regularly spaced label intervals) on the reference, which may occur in multiple locations in the Genome.
translocation_intrachr	Two local alignments which align to the same reference contig (chromosome) and are typically (but not always) separated by more than 5 Mbp on the reference. The minimum confidence (and size) of each local alignment is around one-third lower than for other SVs. They must also satisfy the translocation criteria described below.
translocation_interchr	Two local alignments which align to different reference contigs (chromosomes). The minimum confidence (and size) of each local alignment is around one-third lower than for other SVs. They must also satisfy the translocation criteria described below.
trans_intrachr_common, trans_interchr_common	A translocation_intrachr / translocation_interchr with a breakpoint which overlaps a list of common translocation calls in euploid samples as specified in the .bed file argument to the Pipeline; presumed false positive call which is not displayed in Bionano Access by default.
trans_intrachr_overlap, trans_interchr_overlap	A translocation_intrchr / translocation_interchr with a breakpoint that overlaps another distinct SV call for the same Map: Indicates a possible incorrect call.
trans_intrachr_segdupe, trans_interchr_segdupe	A translocation_intrachr / translocation_interchr with a breakpoint which overlaps an annotated segmental duplication (50kb or larger) in the reference as specified in the .bed file argument to the Pipeline; presumed false positive call which is not displayed in Bionano Access by default.
duplication	A region of the reference which aligns to two places on a genome map.
duplication_inverted	A duplication with MatchGroups in opposite orientation.
duplication_split	A duplication inferred by the rearrangement of MatchGroups, but the Map does not include complete copies of both duplicates.
end	Unaligned region of at least five sites and 50 kbp at one end of the genome map : possibly an incomplete insertion or translocation region.
complex	A pair of MatchGroups which do not satisfy any of the above criteria. For example, translocations which fail the criteria below.

NOTE: Translocation criteria - If the two local alignments (MatchGroup) do not overlap, they must be no further than 500 kbp apart on the query (or have an intermediate MatchGroup on the query). When they do overlap, they must not overlap by more than 30% (of the minimum MatchGroup size) and by no more than 140 kbp.

CALCULATION OF SVFREQ

NOTE: Bionano Solve 3.7 provides a new way of estimating the allele fraction with results stored in the VAF column. See *Bionano Solve Theory of Operation Structural Variant Calling* (CG-30110) for information about the VAF calculation. The column SVfreq is kept for backwards compatibility. Details on the original algorithm are presented here for historical purposes but will be removed in future releases.

SVfreq provides information about the prevalence of an allele in a sample relative to other alleles. This is most relevant for *de novo* Assembly pipeline data. Conceptually, SVfreq reflects the ratio between the number of molecules that are unique to a given allele map and the number of molecules that align to a particular reference region.

SVfreq is calculated based on the weighed molecule coverage during the final refinement stage (refineFinal1; output/contigs/exp_refineFinal1/EXP_REFINEFINAL1.cmap) of the assembly. The molecule coverage data are saved in the **Coverage** column on the consensus genome map CMAP. If a molecule could align to two maps, the coverage it contributes would be halved accordingly. Currently, the molecule alignment counts towards coverage of the CMAP from the first to last aligned label, but the coverage is recorded for label intervals, so it would correspond to the first through the second last aligned label in CMAP. SVfreq is then computed during SV calling and output in the SMAP.

The number of molecules that align to a particular reference region (overall coverage on the reference map regardless of the alleles) is computed by averaging the coverage of all consensus maps that align to the reference region. For each SV, the coverage of the allele-specific consensus map that called the SV (averaged for the map region where the SV is called) is divided by the coverage of the reference.

Because coverage is weighted, if the same SV is called by two or more consensus maps, SVfreq across the maps need to be summed to get the overall variant allele frequency. For example, if a homozygous SV event is observed and called in two maps (only one map is shown), each SV call is expected to have an SVfreq of 0.5. In **Figure 4**, Cov2 to Cov6 on the reference are expected to be zero. The sum of Cov1 and Cov2 on the map is expected to be roughly half of the sum of Cov1 and Cov7 on the reference because molecules would align to both maps containing the deletion. The “SV coverage” (weighed coverage of the labels on the consensus maps) would then be roughly half of the “Ref coverage.”

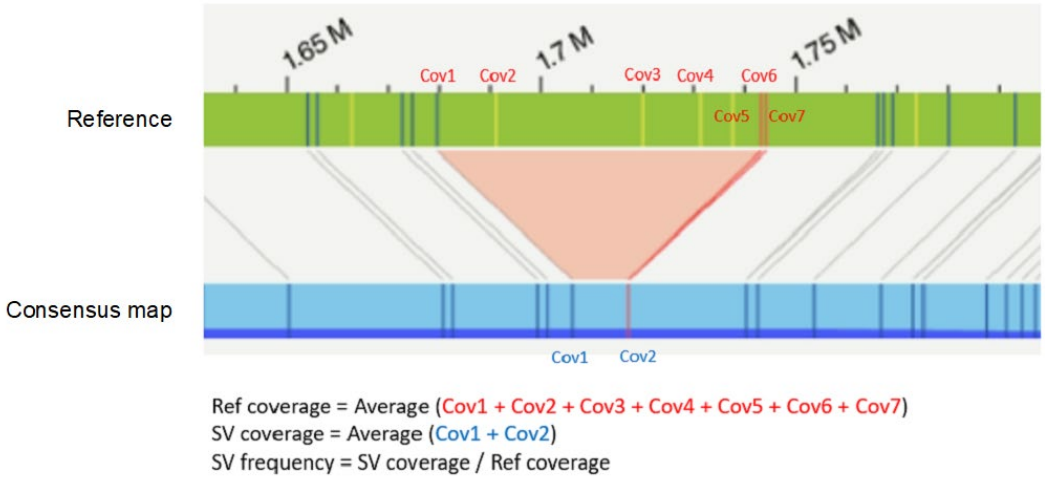

Figure 4. Calculation of SVFreq

BED File Format Specifications

A BED (Browser Extensible Data) file, typically tab-delimited, contains a list of genomic regions. Three required fields specify the regions of interest and additional optional fields. All rows in the BED file are expected to contain the same number of fields. The format for the BED files that Bionano uses is generally consistent with that described on the UCSC Genome Browser website (https://genome.ucsc.edu/FAQ/FAQformat#format1). Exceptions are noted below. Currently, BED files are used for annotating structural variant calls. BED files can be opened in Excel or any text editor for easy readability and editing.

Each BED file entry contains information about a genomic region. This file format specification sheet provides descriptions, with examples, of the fields in the BED file.

FORMAT

The BED file contains three required fields: chrom, chromStart, and chromEnd. Additional optional fields include name, score, strand, thickStart, thickEnd, and itemRgb. The BED files that Bionano provides currently contain no header lines.

EXAMPLE

```
1    1           10000       gap   1    +    1           10000       100,0,150
1    207667      257666      gap   2    +    207667      257666      100,0,150
1    297969      347968      gap   3    +    297969      347968      100,0,150
1    535989      585988      gap   4    +    535989      585988      100,0,150
1    2702782     2746290     gap   5    +    2702782     2746290     100,0,150
1    12954385    13004384    gap   6    +    12954385    13004384    100,0,150
1    16799164    16849163    gap   7    +    16799164    16849163    100,0,150
1    121976460   122026459   gap   8    +    121976460   122026459   100,0,150
1    125184588   143184587   gap   9    +    125184588   143184587   100,0,150
1    223558936   223608935   gap   10   +    223558936   223608935   100,0,150
1    228558365   228608364   gap   11   +    228558365   228608364   100,0,150
1    248946423   248956422   gap   12   +    248946423   248956422   100,0,150
```

FIELD SPECIFICATIONS

Table 48 provides brief descriptions for each field. The data is grouped per genomic region represented in the BED file. Each group includes three required fields (the chromosome, the chromosome start, and the chromosome end for the region of interest) and additional fields.

Table 48. BED file fields - overview

Field	Field Descriptions
chrom	Name of the chromosome, scaffold, or contig
chromStart	Starting position
chromEnd	Ending position
name	Name

score	Score (not used for visualization)
strand	Strand/orientation (not used for visualization)
thickStart	Starting position (not used for visualization)
thickEnd	Ending position (not used for visualization)
itemRgb	Display color in RGB space

FIELD SPECIFICATION DETAILS

Tables 49-57 provide details and examples for each field (including any specific formatting, limitations, and requirements). **NOTE**: Additional fields may be present.

Table 49. Chrom field

chrom	
Description	Name of the chromosome, scaffold, or contig. This should be numeric. There should be no "chr" prefix. This field is required.
Example	"1" and "2"

Table 50. chromStart field

chromStart	
Description	Starting position. This should be numeric. This field is required.
Example	"100"

Table 51. chromEnd field

chromEnd	
Description	Ending position. This should be numeric. This field is required.
Example	"10000"

Table 52. name field

name	
Description	Name or type. This can be any string. Currently, three types of regions are recognized for annotating structural variant calls: "gap," "common," and "segdupe."
Example	"gap" and "common"

Table 53. score field

score	
Description	Score. This can be any numeric value. This is currently used as a region ID field. The UCSC definition requires that the range be between 0 and 1000. Bionano does not enforce this requirement. This field is not used for visualization.
Example	“1”

Table 54. strand field

strand	
Description	Strand. This can be either “+” or “-“. This field is not used for visualization.
Example	“+”

Table 55. thickStart field

thickStart	
Description	Starting position. This should be numeric but is currently considered a dummy field. It is expected to be consistent with chromStart. This field is not used for visualization.
Example	“100”

Table 56. thickEnd field

thickEnd	
Description	Ending position. This should be numeric but is currently considered a dummy field. It is expected to be consistent with chromEnd. This field is not used for visualization.
Example	“10000”

Table 57. itemRgb field

itemRgb	
Description	Display color in RGB space. An RGB value should be in the form “R, G, B.” The three-color components are comma delimited.
Example	“100,0,150”

SVMerge Output File Format Specifications

The Bionano SVMerge tool can merge structural variant (SV) calls from two single-enzyme genome map assemblies of the same sample into a single integrated set of calls (*Bionano Solve Theory of Operation: Structural Variant Calling* (CG-30110). SVMerge is currently recommended for NLRS data only. It merges

insertion, deletion, inversion breakpoint, translocation breakpoint, and duplication calls. The merged calls are output to a text file with the suffix _mergedSV.txt.

The _mergedSV.txt file reports merged SV calls as well as SV calls detected only in one of the single-enzyme assemblies. Each data line contains the merged SV start and end coordinates and their locations in each individual enzyme as shown in each SMAP file using a tab-delimited, text-based file.

The _mergedSV.txt file presents the information in two sections: 1) the information header, which describes the specific format of the data, and 2) the merged SV information block, which contains the data rows. This file format specification sheet provides descriptions, with examples, of the _mergedSV.txt header and merged SV information block format of the file.

When imported into Bionano Access along with proper SMAP and XMAP files, the _mergedSV.txt file is automatically processed and ready for visualization. The _mergedSV.txt files can be opened in Excel for easy readability or in any tab-delimited, text-based editor.

FORMAT

The _mergedSV.txt file header contains the following sections:

- The *_mergedSV.txt* file header:
  - # SVMergeVersion:
  - # SMAP of Enzyme 1:
    - *# <headers copied from SMAP file of Enzyme 1>* (See *SMAP File Format Specification Sheet* (CG-30041)
  - # SMAP of Enzyme 2:
    - *# <headers copied from SMAP file of Enzyme 2>* (See *SMAP File Format Specification Sheet* (CG-30042)
    - *# "data column names"*
- The *_mergedSV.txt* file information block (each row as defined by the *data column names*). The information columns can be grouped into three categories:
  - Column 1-12 and 37, 38: [SVIndex, Type, RefcontigID1, RefcontigID2, RefStartPos, RefEndPos, Confidence, RawConfidence, Size, Zygosity, E1Id, E2Id, Orientation1, Orientation2] - information for merged call
  - Columns 13-24: [Type1, Confidence1, RawConfidence1, QryContigID1, QryStartPos1, QryEndPos1, QryStartIdx1, QryEndIdx1, RefStartPos1, RefEndPos1, RefStartIdx1, RefEndIdx1] – SMAP entry information of the first enzyme used for SV merging. Copied from the original SMAP entry.
  - Columns 25-36: [Type2, Confidence2, RawConfidence2, QryContigID2, QryStartPos2, QryEndPos2, QryStartIdx2, QryEndIdx2, RefStartPos2, RefEndPos2, RefStartIdx2, RefEndIdx2] – SMAP entry information of the second enzyme used for SV merging. Copied from the original SMAP entry.

HEADER SPECIFICATIONS

Table 58 lists header rows that are prefixed by the pound sign (#). **NOTE**: "*****" denotes the required header line tags for _mergedSV.txt file. Required header line tags must be present and must precede the SV Information Block to read _mergedSV.txt file. Other header lines are optional and may be omitted.

Table 58: SVMerge header fields - overview

Header Line Tag	Header Line Description
# SVMergeVersion:	Indicates the version of the _mergedSV.txt file*
# SMAP of Enzyme 1:	A string denoting the path to the first SMAP file used for SVMerge*
# <headers copied from SMAP file of Enzyme 1>	Header information copied from the first SMAP file excluding the column header (#h) and data type (#f) lines (see SMAP file format for detail)
# SMAP of Enzyme 2:	A string denoting the path to the second SMAP file used for SVMerge*
# <headers copied from SMAP file of Enzyme 2>	Header information copied from the second SMAP file excluding the column header (#h) and data type (#f) lines (see SMAP file format for detail)
# SVIndex	Defines the columns for each data row*

HEADER SPECIFICATION DETAILS

Tables 59 through **62** provide the *_mergedSV.txt* file header's descriptions (including any specific formatting, limitations, and requirements) and examples.

Table 59. SVMerge file version header fields

# SVMergeVersion:	
Header	# SVMergeVersion:
Description	Indicates the version of the _mergedSV.txt file.
Example	# SVMergeVersion:<TAB>0.9.5

Table 60. SMAP of enzyme 1 header fields

# SMAP of Enzyme 1:	
Header	# SMAP of Enzyme 1:
Description	A string denoting the path to the first SMAP file used for SV Merge
Example	# SMAP of Enzyme 1:<TAB>output/contigs/exp_refineFinal1_sv/merged_smaps/exp_refineFinal1_merged_filter_inversions.smap

Table 61. SMAP of Enzyme 2 header fields

# SMAP of Enzyme 2:	
Header	# SMAP of Enzyme 2:
Description	A string denoting the path to the second SMAP file used for SV Merge
Example	# SMAP of Enzyme 2:<TAB>output/contigs/exp_refineFinal1_sv/merged_smaps/exp_refineFinal1_merged_filter_inversions.smap

Table 62. SVIndex header fields

# SVIndex		
Header	# SVIndex	
Description	Description of the required tab-separated columns in # SVIndex	
	SVIndex	A unique line number for the data lines in the SMAP file.
	Type	Type of SV (insertion, deletion, inversion, translocation. See definitions in 30041, SMAP File Format Specification Sheet).
	RefcontigID1	Reference contig ID. **NOTE**: RefcontigIDs must be integers, but they need not be sequential.
	RefcontigID2	Reference contig ID.
	RefStartPos	SV breakpoint coordinate on reference map ID1 (RefcontigID1).
	RefEndPos	SV breakpoint coordinate on reference map ID2 (RefcontigID2).
	Confidence	Probability of an insertion or deletion call being correct, and a quality metric for translocation breakpoints. '-1.00' for other SV types.
	RawConfidence	Only applies to indels. '-1' for other SV types.
	Size	Size for insertion, deletion, inversion, and duplication calls. Maintained for backwards compatibility. SVSize column should be preferred over this column.
	Zygosity	One of 'homozygous', 'heterozygous', or 'unknown'. For merged calls, if one of the single-enzyme calls is 'heterozygous', the merged call is 'heterozygous'.

# SVIndex	
LinkID	For some SV types, two SMAP entries may be linked using this field (e.g., inversion-partial).
E1Id	The first SV SMAP entry id participated in the merging (For file name see header "# SMAP of Enzyme 1:"). '-1' if the SV is only detected in the second single-enzyme assembly.
E2Id	The second SV SMAP entry id participated in the merging (For file name see header "# SMAP of Enzyme 2:"). '-1' if the SV is only detected in the first single-enzyme assembly.
[Type1, Confidence1, RawConfidence1, QryContigID1, QryStartPos1, QryEndPos1, QryStartIdx1, QryEndIdx1, RefStartPos1, RefEndPos1, RefStartIdx1, RefEndIdx1, LinkID1]	The first SV SMAP entry information participating in SV merging. Copied from the original SMAP file as indicated in the header - "# SMAP of Enzyme 1:". (See 30041, SMAP File Format Specification Sheet for detail.) All values would be -1 except Type1 (NA) if the SV is only detected in the second single-enzyme assembly.
[Type2, Confidence2, RawConfidence2, QryContigID2, QryStartPos2, QryEndPos2, QryStartIdx2, QryEndIdx2, RefStartPos2, RefEndPos2, RefStartIdx2, RefEndIdx2, LinkID2]	The second SV SMAP entry information participating in SV merging. Copied from the original SMAP file as indicated in the header - "# SMAP of Enzyme 2:". (See 30041, SMAP File Format Specification Sheet for detail.) All values would be -1 except Type2 (NA) if the SV is only detected in the first single-enzyme assembly.
Orientation1	Only applies to translocation and inversion breakpoints; -1 for other SV types. For translocation and inversion orientation, see **Figure 1** and **Figure 2**, respectively.
Orientation2	Only applies to translocation and inversion breakpoints; -1 for other SV types. For translocation and inversion orientation, see **Figure 1** and **Figure 2**, respectively.

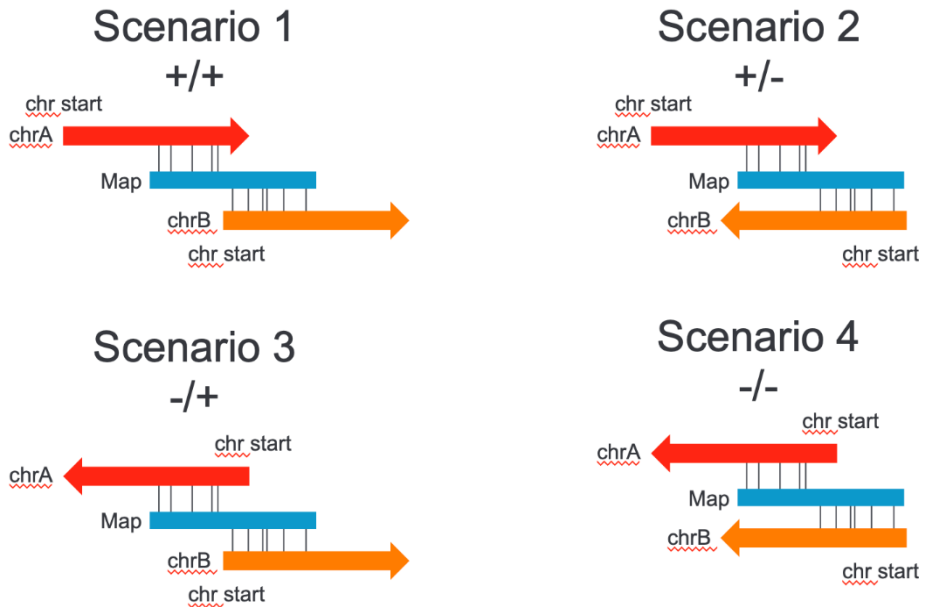

Figure *5*. Determination of translocation orientation.

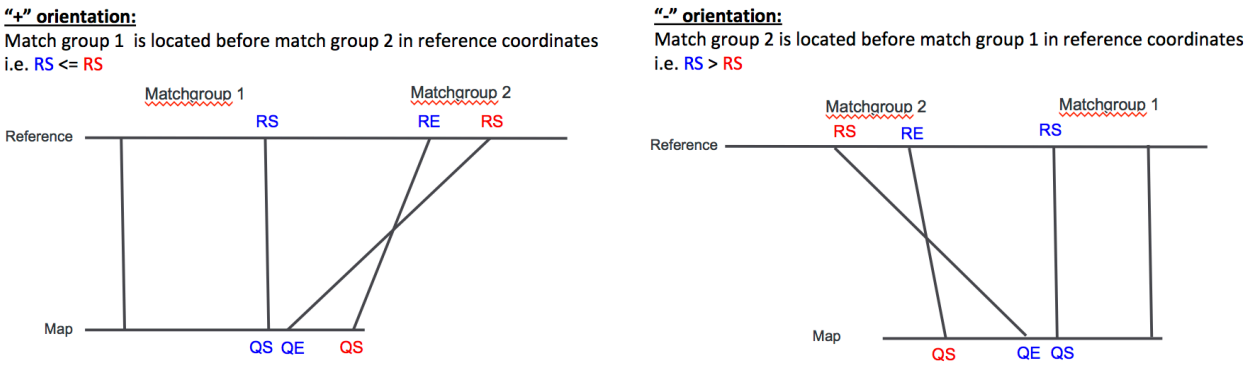

Figure 6. Determination of inversion orientation.

MERGED SV INFORMATION BLOCK SPECIFICATION

The data is grouped such that each data row represents one structural variant – merged or detected in one of the single-enzyme assemblies.

MERGING SV WITH VARIANT ANNOTATION

Variant annotation of SVs is performed for each enzyme, and SVMerge combines the annotation of the two sets with minor edits. The output file name, when run using the command line has the suffix *_mergedSV_genes.txt. **NOTE**: when one downloads the results from Bionano Access, the output file name's suffix is *_mergedSV.txt. The following describes the addition of variant annotation information to the basic SVMerge output, and it assumes that the two enzymes used were Nt.BspQI and Nb.BssSI. Refer to the *Bionano Solve Theory of Operation: Variant Annotation Pipeline* (CG-30190).

There are two additional header lines denoting the sample name given to the two single enzyme experiments.

Table 63. SVIndex variant annotation fields

# SVIndex		
Header	# SVIndex	
Description	Description of the required tab-separated columns in # SVIndex	
	Present_in_%_of_BNG_control_samples	The percentage of samples in the Bionano control SV database that also carry that SV. SVMerge takes the maximum number between the two enzymes' variant annotation results.
	Present_in_%_of_BNG_control_samples_with_the_same_enzyme	Same as above, but the percentage is calculated only based on those database samples having the same enzyme as the sample being annotated. SVMerge takes the maximum number between the two enzymes' variant annotation results.
	Algorithm_BspQI Algorithm_BssSI	The calls are based on comparing the *de novo* assembly of the sample with the reference, and so the algorithm is called "assembly comparison."
	Fail_BspQI_assembly_chimeric_score Fail_BssSI_assembly_chimeric_score	A flag used to denote whether a potential chimeric join occurred during *de novo* assembly at the variant locus. This denotes whether a minimal chimeric quality score of 35 and coverage of 10 have been achieved around each SV breakpoint. A value of 'pass' means that the two criteria have been met; a 'fail' denotes the criteria not met; a 'not_applicable' value denotes that the check has not been performed. Notice that this check is performed only for inversion and translocation calls. **NOTE:** a chimeric quality score of a label on a genome map is the percent of molecules that align to both sides of the label out of all molecules that align on either side near this label.

The next sets of columns vary depending on whether trio, dual or single analyses have been selected upon execution of the Variant Annotation Pipeline (VAP) for each enzyme (**Tables 64-66**).

TRIO ANALYSIS

Table 64. SVIndex annotation fields – trio analysis

# SVIndex	
Header	# SVIndex
Description	Description of the required tab-separated columns in # SVIndex

Found_in_parents_BspQI_assemblies Found_in_parents_BssSI_assemblies	Whether the SV call is also identified in the father's or mother's assembly. The possible values are 'mother,' 'father,' 'both' and 'none.' Since inversion partial calls are not annotated, a value of '-' is shown for any inversion partial call.
Found_in_parents_BspQI_molecules Found_in_parents_BssSI_molecules	These columns show whether there are enough parent molecules supporting the proband's genome map at the SV breakpoints. The possible values are 'mother,' 'father,' 'both' and 'none.' Since inversion partial calls are not annotated, a value of '-' is shown for inversion partial calls. **NOTE** that the minimum numbers of molecules required are defined as parameters by the users upon running the variant annotation pipeline.
Found_in_self_BspQI_molecules Found_in_self_BssSI_molecules	These columns denote whether there are enough proband's molecules supporting the proband's genome map at the SV breakpoints. Since inversion partial calls are not annotated, a value of '-' is shown for inversion partial calls. **NOTE**: The minimum number of molecules required is defined as a parameter by the users upon running the variant annotation pipeline.

DUAL ANALYSIS

Table 65. SVIndex annotation fields – dual analysis

# SVIndex		
Header	# SVIndex	
Description	Description of the required tab-separated columns in # SVIndex	
	Found_in_self_BspQI_molecules Found_in_self_BssSI_molecules	These columns denote whether there are enough case sample molecules supporting the case sample's genome map at the SV breakpoints. The possible values are 'yes' and 'no.' Since inversion partial calls are not annotated, a value of '-' is shown for inversion partial calls. **NOTE**: The minimum number of molecules required is defined as a parameter by the users upon running the variant annotation pipeline.
	Found_in_control_sample_BspQI_assembly Found_in_control_sample_BssSI_assembly	Whether the SV call is also identified in the control sample's assembly. The possible values are 'yes' or 'no.' Since inversion partial calls are not annotated, a value of '-' is shown for any inversion partial call.
	Found_in_control_sample_bspqi_molecules Found_in_control_sample_bspqi_molecules	These columns show whether there are enough control sample molecules supporting the case sample's genome map at the SV breakpoints. The possible values are 'yes' and 'no.' Since inversion partial calls are not annotated, a value of '-' is shown for inversion partial calls. **NOTE**: The minimum number of molecules required is defined as a parameter by the users upon running the variant annotation pipeline.

SINGLE ANALYSIS

Table 66. SVIndex annotation fields – single analysis

# SVIndex		
Header	# SVIndex	
Description	Description of the required tab-separated columns in # SVIndex	
	Found_in_self_BspQI_molecules Found_in_self_BssSI_molecules	These columns denote whether there is enough case sample molecules supporting the case sample's genome map at the SV breakpoints. The possible values are 'yes' and 'no.' Since inversion partial calls are not annotated, a value of '-' is shown for inversion partial calls. **NOTE**: The minimum number of molecules required is defined as a parameter by the user upon running the variant annotation pipeline.

GENE OVERLAP

The variant breakpoints are refined during SVMerge, so the gene-overlap is recomputed using the refined positions. The last few columns show the genes overlapping or closest to merged variants.

Table 67. SVIndex gene annotation fields

# SVIndex		
Header	# SVIndex	
Description	Description of the required tab-separated columns in # SVIndex	
	OverlapGenes	A semi-colon separated list indicating which genes overlap with the SV.
	NearestNonOverlapGene	The next closest gene to the SV.
	NearestNonOverlapGeneDistance	The distance between the SV and the next closest gene.

VCF File Format Specifications

The Bionano Variant Call Format (VCF) file contains structural variation (SV) and copy number variation (CNV) calls in a VCF version 4.2 format. It is generated by a Python-based VCF converter from the SMAP and CNV output from the main SV analysis pipelines. More details can be found in the *SMAP File Format Specification* (CG-30041) and *Bionano Solve Theory of Operation Structural Variant Calling* (CG-30110) as well as in the annotated versions of these files produced by the Variant Annotation Pipeline: *Structural Variant Annotation Pipeline File Format Specification Sheet* (CG-30168) and *Copy Number Variant Annotation Pipeline File Format Specification Sheet* (CG-30461). Bionano's VCF output has been validated by VCFtools' VCF validator (https://vcftools.github.io). For more information on how the SV and CNV calls are generated by the analysis

pipelines as well as more detailed descriptions of how variants are represented in VCF, see *Bionano Solve Theory of Operation Structural Variant Calling* (CG-30110).

The VCF output is automatically generated at the end of each analysis pipeline and imported into Bionano Access. Here, after the analysis results have been imported, users have the option to select variant calls of interest and output a VCF with the selected calls.

In addition, the VCF converter is a standalone tool that may be run on the command line. It is packaged as part of Bionano Solve. The tool requires Python 3.7 and Python libraries including pandas and numpy. The required input is the SMAP output from one of the SV analysis pipelines. The SMAP may be annotated by the Variant Annotation Pipeline. The command line tool optionally takes the output from the CNV analysis pipeline; the CNV data is included by default when the VCF is generated as part of the pipeline runs. See Appendix G in *Bionano Solve Theory of Operation Structural Variant Calling* (CG-30110).

FORMAT

VCF is a text-based format; VCF files may be opened in Excel for easy readability or in any text-based editor. Each VCF file contains a meta-information section. Each line in this section starts with "##." It is followed by a single header line that starts with "#" and then the data lines, each containing information about a given variant call. Aspects of the representation of variant calls that are unique to Bionano are presented. See the *Bionano VCF version 4.2 File Format Specification Sheet* (CG-30459) for reference and additional information.

META-INFORMATION

In the meta-information section, information lines are presented as key-value pairs. Fields tagged with "INFO" such as SVTYPE, SVLEN and END provide basic information about each variant call. Variant calls are listed with SVTYPE and ALT alleles that follow VCF standard conventions. Original variant types from the Bionano SV and CNV callers are preserved in the BNGTYPE INFO field. Additional fields such as OVERLAPGENES, NEARGENE, and DGVOVERLAPS come from the VAP, which annotates variant calls based on, for example, genome and gene annotations. See *Structural Variant Annotation Pipeline File Format Specification Sheet* (CG-30168) and *Copy Number Variant Annotation Pipeline File Format Specification Sheet* (CG-30461) for information about the annotations. VAP runs automatically for human and mouse datasets; those meta-information lines will be present regardless of whether the VAP fields are present in the input SMAP.

HEADER

The single header line has nine fixed fields: CHROM, POS, ID, REF, ALT, QUAL, FILTER, INFO, and FORMAT. The last field is variable, and it depends on the sample name provided. The default is "Sample1".

VARIANT DATA

CHROM: the chromosome on which the variant is called. The set of possible chromosomes is indicated in the meta-information section. For human datasets, the chromosome IDs in the SMAP are automatically converted into the "chrN" format. Chromosomes "23" and "24" are converted to "chrX" and "chrY", respectively. For translocations, inversions, and inversion_partial variants, each breakpoint is listed as a separate entry where CHROM lists one of the chromosomes with the partner breakpoint referenced in the MATEID INFO field.

- **POS**: the starting position of the variant interval is indicated. The ending position is indicated as END in the INFO field. When converting the SMAP into VCF, the VCF converter attempts to estimate breakpoint uncertainty and adjusts positions accordingly. Therefore, the coordinates in POS and END may not

correspond to coordinates in the SMAP. The uncertainty of the breakpoints is indicated in the CIEND and CIPOS fields. See Appendix G in *Bionano Solve Theory of Operation Structural Variant Calling* (CG-30110) for information about the calculation.

- **ID**: these are output based on the SMAP entry IDs ("SMAP" followed by SMAP ID) and the CNV entry IDs ("CNV" followed by CNV ID). Breakpoints for intra-chromosomal fusions, inter-chromosomal translocations and inversions will be listed with a 'bnd_' prefix and a unique numeric suffix. For example, a translocation called with SMAP ID 4122 will have two breakpoint entries with IDs bnd_SMAP4122_1 and bnd_SMAP4122_2.
- **REF**: because the Bionano optical mapping platform does not provide single-base level resolution, the precise base for a given variant is not relevant. "N" is the output for all variants.
- **ALT**: the variant type (defined in the meta-information section) is output.
- **QUAL**: the variant confidence scores from the SMAP and CNV output are converted into Phred scale. See *Bionano Solve Theory of Operation Structural Variant Calling* (CG-30110) for more information on the conversion.
- **FILTER**: "Masked" is the output for masked calls. "LowConfidence" is the output for variants that do not meet the minimum recommended confidence score. "PoorMoleculeSupport" is output for variants that do not meet VAP self-molecule checks or assembly chimeric quality score checks. "PASS" is the output for all other calls. Masking is performed by default during the SV and CNV calling steps using separate masks. The SV mask includes regions where false positive translocation calls were made in control samples with no known translocations. These regions are often segmental duplication loci and cannot be aligned uniquely. The CNV mask includes regions with elevated coverage noise, defined based on control samples with no known large CNV events. False positive CNV calls are more common in high coverage noise regions. SV and CNV calls overlapping with the masks are masked and of lower confidence. See *Bionano Solve Theory of Operation Structural Variant Calling* (CG-30110) for information about the masks and the masking procedure.
- **INFO**: the fields are defined in the meta-information section.
- **FORMAT field definition**: Structural variant calls will have "GT" and "VAF" to record the detected genotype and variant allele fraction as a floating point number. Genotype can be hemizygous ("1", inferred based on copy number data), heterozygous ("0/1"), homozygous ("1/1"), or unknown ("./."). If zygosity is not present in the input SMAP, "./." is output. Hemizygous chromosomes are defined to be the ones where the average chromosome copy number is between 0.9 to 1.1. CNV calls will have "CN" and "CNF" fields that record the rounded copy number for the gain or loss as an integer and the fractional copy number as a floating point. Variants that fall in a region where absence or loss of heterozygosity was detected will have an "ABK" field defined where the value is a unique identifier for the AOH/LOH region. Variants in an AOH/LOH region will all share the same ABK identifier.

NOTES:

- Both intra-chromosomal fusions and inter-chromosomal translocations are represented by paired breakend (BND) entries linked by MATEID. The SMAP orientation column is used to produce the break-end continuation direction in the VCF ALT field. However, the SMAP data does not fully characterize the SV, as the orientation column is encoded based on the map while the semantics of the VCF specification are defined in terms of the reference. The SMAP does not currently encode the relationship between the aligning segments on the map and the aligning segments on the reference. Therefore, directionality is encoded assuming that the upstream side of the map always corresponds to RefContigID1 and RefStartPos columns in the SMAP. When needed, the complete orientation and continuation direction for translocations can be visualized in Bionano Access.

- Unlike the uncertainty for other SV types, the uncertainty for CNV calls is fixed at 30 kbp and set based on empirical data. It is used for both CIPOS and CIEND. It is subject to change, as new methods for estimating the breakpoint uncertainty become available.
- The two-entry inversion_paired calls in the input SMAP are converted into single-entry VCF lines. The smallest coordinate in the two entries is taken to be POS, and the largest coordinate is taken to be END. POS and END then represent the outer bounds of where the inversion of interest might be.

SAMPLE HEADER

Following is a sample header that includes definitions for all INFO and FILTER entries.

```
##fileformat=VCFv4.2
##fileDate=2023-05-13
##source=Bionano Solve 3.8 variant annotation pipeline
##command=bionano_vcf_converter.py <args>
##sample=<ID=sample_id,sex=male>
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##INFO=<ID=BNGTYPE,Number=.,Type=String,Description="Original BNG variant type from SMAP or CNV">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant described in this
record">
##INFO=<ID=MATEID,Number=.,Type=String,Description="ID of mate breakends">
##INFO=<ID=SVLEN,Number=1,Type=Integer,Description="Difference in length between REF and ALT alleles">
##INFO=<ID=CIPOS,Number=2,Type=Integer,Description="Breakpoint uncertainty for start position POS">
##INFO=<ID=CIEND,Number=2,Type=Integer,Description="Breakpoint uncertainty for end position END">
##INFO=<ID=CT,Number=1,Type=String,Description="Breakpoint connection type">
##INFO=<ID=EXPERIMENT,Number=1,Type=String,Description="experiment_id from dbVar submission of the
experiment that generated this call">
##INFO=<ID=SAMPLE,Number=1,Type=String,Description="sample_id from dbVar submission. Each call must
have only one of either SAMPLE or SAMPLESET">
##INFO=<ID=SAMPLESET,Number=1,Type=Integer,Description="sampleset_id from dbVar submission. Each call
must have only one of either SAMPLE or SAMPLESET">
##INFO=<ID=OVERLAPGENES,Number=.,Type=String,Description="Set of genes overlapped by structural
variant">
##INFO=<ID=NEARGENE,Number=1,Type=String,Description="Nearest non-overlapping gene">
##INFO=<ID=NEARGENEDIST,Number=1,Type=Integer,Description="Distance to nearest non-overlapping gene">
##INFO=<ID=DGVOVERLAPS,Number=1,Type=Integer,Description="Number of overlapped variants in DGV
database">
##INFO=<ID=INPARENTS,Number=1,Type=String,Description="Found in parents' datasets">
##INFO=<ID=ISCN,Number=.,Type=String,Description="ISCN annotation">
##INFO=<ID=UCSC1,Number=1,Type=String,Description="UCSC web link 1">
##INFO=<ID=UCSC2,Number=1,Type=String,Description="UCSC web link 2">
##INFO=<ID=SAMPLETYPE,Number=1,Type=String,Description="SV sample type in VAP pipeline">
##INFO=<ID=ALG,Number=1,Type=String,Description="Algorithm used in VAP">
##INFO=<ID=IMPRECISE,Number=0,Type=Flag,Description="Imprecise structural variation">
##INFO=<ID=PCNTBNG,Number=1,Type=Float,Description="Percent of BNG control samples with SV">
##INFO=<ID=PCNTBNGENZ,Number=1,Type=Float,Description="Percent of BNG control samples with the same
enzyme with SV">
##INFO=<ID=PCNTBNGHOM,Number=1,Type=Float,Description="Percent of BNG control samples with homozygous
SV">
##INFO=<ID=PCNTBNGHET,Number=1,Type=Float,Description="Percent of BNG control samples with heterozygous
SV">
##INFO=<ID=PCNTBNGSVAFR,Number=1,Type=Float,Description="Percent of AFR BNG control samples with SV">
##INFO=<ID=PCNTBNGSVAMR,Number=1,Type=Float,Description="Percent of AMR BNG control samples with SV">
##INFO=<ID=PCNTBNGSVEUR,Number=1,Type=Float,Description="Percent of EUR BNG control samples with SV">
##INFO=<ID=PCNTBNGSVEAS,Number=1,Type=Float,Description="Percent of EAS BNG control samples with SV">
##INFO=<ID=PCNTBNGSVSAS,Number=1,Type=Float,Description="Percent of SAS BNG control samples with SV">
##INFO=<ID=PCNTBNGSVUNK,Number=1,Type=Float,Description="Percent of unknown BNG control samples with
SV">
##INFO=<ID=FAILCHIM,Number=1,Type=String,Description="Fail assembly chimeric score">
##INFO=<ID=GENFUS,Number=0,Type=Flag,Description="Putative gene fusion">
##INFO=<ID=INCTRLASSM,Number=0,Type=Flag,Description="Found in control sample assembly">
##INFO=<ID=INPAIRCTRL,Number=0,Type=Flag,Description="Found in paired control sample CNVs">
##INFO=<ID=INPARASSM,Number=1,Type=String,Description="Found in parents' assemblies">
##INFO=<ID=ZYG,Number=1,Type=String,Description="Zygosity of SV">
##INFO=<ID=ZYGPAIRASSM,Number=.,Type=String,Description="Zygosity in paired control sample assembly">
##INFO=<ID=ZYGMASSM,Number=.,Type=String,Description="Zygosity in mother assembly">
##INFO=<ID=ZYGFASSM,Number=.,Type=String,Description="Zygosity in father assembly">
```

```
##INFO=<ID=SELFMOL,Number=0,Type=Flag,Description="Found in self molecules">
##INFO=<ID=INCTRLMOL,Number=0,Type=Flag,Description="Found in paired control sample molecules">
##INFO=<ID=PARMOL,Number=1,Type=String,Description="Found in parents' molecules">
##INFO=<ID=SELFMOLCNT,Number=1,Type=String,Description="Self molecule count">
##INFO=<ID=CTRLMOLCNT,Number=1,Type=String,Description="Paired control sample molecule count">
##INFO=<ID=MMOLCNT,Number=1,Type=String,Description="Mother molecule count">
##INFO=<ID=FMOLCNT,Number=1,Type=String,Description="Father molecule count">
##ALT=<ID=DEL,Description="Deletion">
##ALT=<ID=INS,Description="Insertion">
##ALT=<ID=INV,Description="Inversion">
##ALT=<ID=DUP,Description="Duplication">
##ALT=<ID=DUP:INVERTED,Description="Inverted duplication">
##ALT=<ID=BND,Description="Breakend">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=CN,Number=1,Type=Integer,Description="Copy number genotype for imprecise events">
##FORMAT=<ID=CNF,Number=1,Type=Float,Description="Fractional copy number for imprecise events">
##FORMAT=<ID=VAF,Number=1,Type=Float,Description="Variant allele fraction">
##FORMAT=<ID=ABK,Number=1,Type=Integer,Description="Identifier for block of absence of heterozygosity
in an individual that a variant falls in">
##FILTER=<ID=LowConfidence,Description="Does not meet minimum recommended confidence score for variant
type">
##FILTER=<ID=Masked,Description="Masked due to low quality in problematic calling regions">
##FILTER=<ID=PoorMoleculeSupport,Description="Does not meet molecule support criteria">
```

OGM BAM File Specifications

With Solve 3.8, molecule-to-reference alignments are saved in the standard Binary Alignment Map (BAM) format in addition to the native Bionano _r.cmap, _q.cmap and .xmap files. Bionano OGM BAM files conform to the Sequence Alignment/Map Format Specification v1.6 as described at https://samtools.github.io/hts-specs/SAMv1.pdf and can be used with any software that supports this format. Some OGM specific details of the BAM file follow:

- Each XMAP data line describes a linear alignment of a molecule segment to a reference sequence segment. Each such line is presented as a single BAM record, except for occasions where it may be split on large deletions into multiple BAM records (described below).
- Start and end alignment positions are based on the full detected size of a molecule and not restricted to label positions only. That is, if an XMAP data line indicates that the first or last label of a molecule aligns to a reference label, the BAM file will encode the alignment starting at the beginning or end of the of the molecule. Similarly, if an XMAP data line indicates the alignment starts (or ends) before (or after) an interior label, the BAM record will encode as aligning half the distance to the adjacent unaligned label.
- No base sequence is recorded for molecules.
- Base quality scores are not provided.
- Mapping quality encodes the original XMAP confidence value with Bonferroni correction, except in the case of ambiguous mapping, where it is set to 0.
- Unaligned molecules are not included in the OGM BAM
- Each alignment is assigned a read group based on its label channel in the XMAP. The sample name for a read group is assigned based on the experiment ID command line parameter.
- If a samtools (http://www.htslib.org/) sequence dictionary is provided, it will be used to map CMAP IDs to chromosome names for use as reference sequence names. Otherwise, the CMAP IDs will be used as reference sequence names. Sequence dictionaries are provided with Solve for human reference genomes and automatically used by the BAM conversion process in the Bionano pipelines.

CIGAR Alignment Encoding

CIGAR strings recording the alignment between the molecule and reference are created to encode the putatively matching region around resolved, aligning labels as well as regions where interval/site differ from the reference.

More specifically, XMAP alignments are encoded in the BAM as CIGAR strings considering each consecutive pair of label matches, where a label match is an aligned label ID pair referencing one reference label and one query label. Each consecutive pair of label matches is considered as an independent portion of the overall XMAP alignment. There are four scenarios that are considered, depending on possible combinations of one label of the second match being a) the same as the first, b) the label immediately following the first, or c) a further label beyond the immediately following label on its respective sequence. See **Figures 7-10**.

SCENARIO 1: THE SECOND MATCH COMPRISES THE SUBSEQUENT LABEL ON BOTH THE REFERENCE AND THE QUERY.

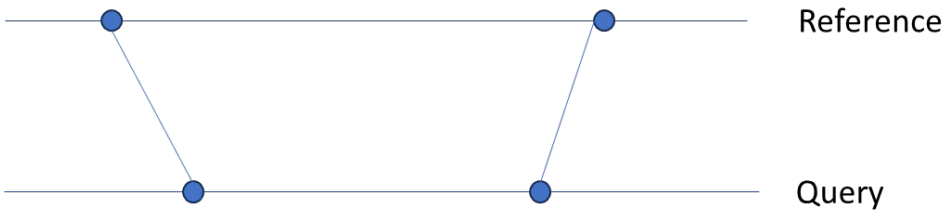

Figure 7. Consecutive matches with no shared or interposing labels

A portion of an alignment spanning two label matches will be encoded as a continuous CIGAR match (M) the length of the reference interval if a.) there are no interposing unaligned sites on either the query or reference and b.) any difference in the sizes of the reference and query intervals is less than a *sizing_error_tolerance* parameter (defaults to 5,000 bp). If there are no interposing unaligned sites on either sequence but the size difference is greater than a *sizing_error_tolerance* parameter, then the alignment portion will be encoded as a CIGAR insertion (I) or deletion (D) the size of the difference. The insertion or deletion will be encoded in the center of the alignment with CIGAR matches flanking it.

SCENARIO 2: THE SECOND LABEL MATCH ALIGNS TO THE SAME REFERENCE SITE AS THE FIRST LABEL MATCH.

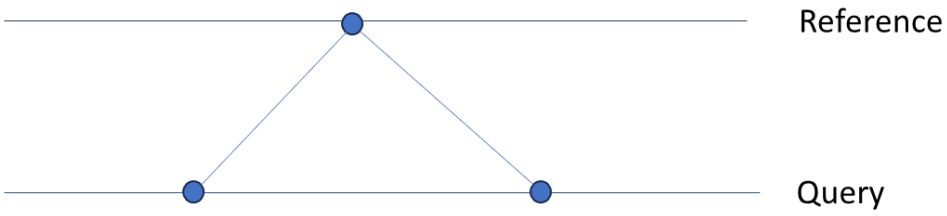

Figure 8. Consecutive label matches with a shared reference label

If two query sites align to a single reference site, a CIGAR insertion (I) the size of the query interval is encoded.

SCENARIO 3: THE SECOND LABEL MATCH ALIGNS TO THE SAME QUERY SITE AS THE FIRST LABEL MATCH.

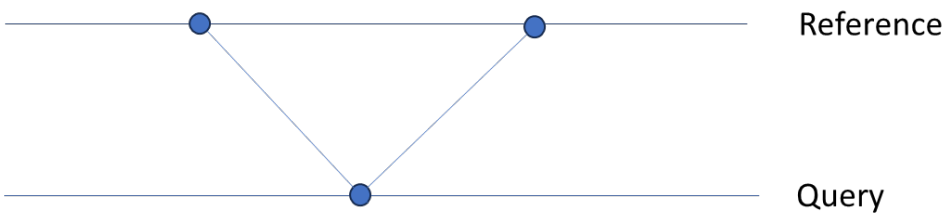

Figure 9. Consecutive label matches with a shared query label

If two reference sites align to a single query site and the size of the reference interval is less than a *minimum_resolvable_interval* parameter (defaulting to 50,000, effectively disabling this encoding), then it is assumed the interval exists on the actual molecule but was below the imaging resolution limit of detection, and it is encoded as a CIGAR match (M) the length of the interval reference. If the reference interval is greater than *minimum_resolvable_interval*, then it is encoded as a CIGAR deletion (D) the size of the reference interval.

SCENARIO 4: EITHER THE REFERENCE OR QUERY ALIGNMENT OF THE SECOND LABEL MATCH ALIGNS TO A FURTHER LABEL ON ITS SEQUENCE.

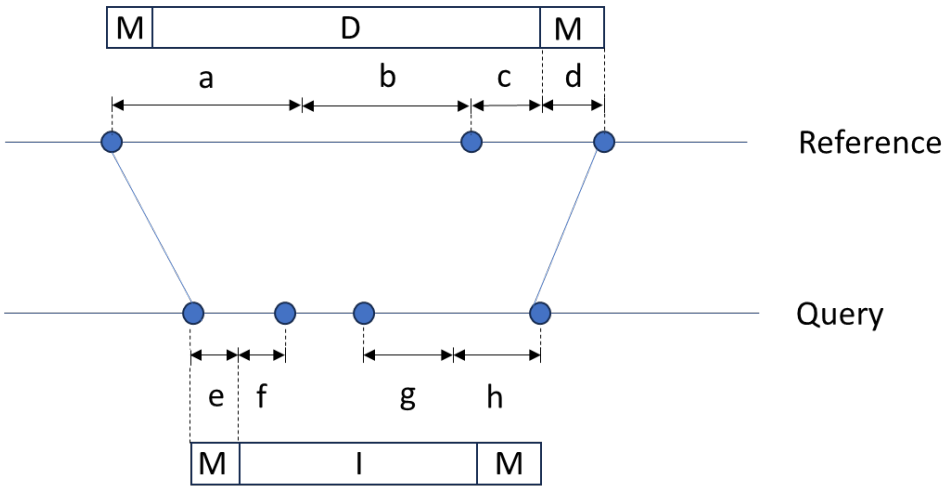

Figure 10. Consecutive label matches with interposing unmatched labels

If there are interposing unaligned sites between the label matches on either sequence, there is a clear indication of mismatched bases between the matches. A single base mismatch span could be as small as to cover the longer span of mismatched labels across the two sequences or as large as the distance between match labels. This leads to an inevitable representation error: the BAM specifies an alignment in base-pair resolution, but OGM data has much lower resolution, averaging about fifteen labels per 100kb for the DLE-1 enzyme on human samples. To limit the magnitude of the error, a midpoint is identified on each sequence between each match label and its closest unaligned label between the matches (or the center of the interval if there are no unaligned sites). A CIGAR matching (M) span is encoded between each match label and the shorter distance to a midpoint (chosen between the reference and query midpoints). For example, in **Figure 10**, *e* is chosen over *a* and *d* is chosen over *h* as the matching (M) spans. The remaining DNA is accounted for by a CIGAR insertion (I) in the query and a CIGAR deletion (D) of the reference. Thus, the alignment in the above figure might be encoded as the CIGAR string `2000M19000D11000I2200M`. The labeled blocks indicate how each segment of the reference

and query sequence contribute, where the matching segments are the same and the insertion and deletion are unique to each sequence.

After each pair of label matches across all four scenarios is converted to its independent CIGAR representation, a final CIGAR string for the whole alignment is produced, collapsing runs of consecutive CIGAR matches (M) into single matches of combined length.

Molecule Label Alignment Tags

Each aligned segment also has a BAM tag with the identifier “ls” containing a list of the reference label positions that had an aligned query label. If there are multiple query labels that aligned to a single reference label, then its position is encoded as a negative integer. This is useful for algorithms that detect SNPs at label positions. When a reference label position is close enough to another label position that it is mis-resolved in some molecules, it may be less suitable (i.e., noisier) for detecting SNPs. This is because a molecule with only one true label may be aligned to both reference sites, and there may be two true labels that cannot be resolved, giving rise to an ambiguous alignment pattern.

Assigning Linear Alignments to Read Alignments

There may be multiple linear alignments (one per XMAP row) between segments of a molecule and the reference, both due to structural variants as well as repeats in the genome. The BAM conversion process uses a weighted interval scheduling algorithm to choose a subset of these alignments that best cover the full molecule, preferring long, high scoring alignments with minimal overlap. These alignments are marked as comprising the primary read alignment, with all other alignments marked as secondary. If two alignments cover the same segment of a molecule and differ in quality score by less than 10, the quality score is set to 0 for both.

Split Alignments

The BAM format can represent deletions either encoded within a single linear alignment in the CIGAR string, or as an unaligned segment of the reference flanked by a split pair of linear alignments. Because OGM molecules are much longer than even the longest NGS reads, linear alignments can span deletions that would normally only be detected in NGS data by copy number analysis. Such copy number analysis algorithms may not consider CIGAR encoded deletions. To make such deletions detectable by such algorithms, the BAM converter will split linear alignments with large internal deletions into separate linear alignments.

Converting OGM Alignments to BAM Format from the Command Line

The BAM converter can be run from the command line in two steps. The first step converts all the alignments between molecules and a single reference sequence in isolation of any molecule alignments to other reference sequences. For human OGM molecule alignments to chromosome one on an existing *de novo* or rare variant analysis output (assumed to be in directory $output), one could run:

```
python3 -m xmap_to_bam \
  -x $output/contigs/alignmolvref/merge/exp_refineFinal1_contig1.xmap  \
  -c $output/ref/hg38.fa.dict \
  -o $output/contigs/alignmolvref/merge/exp_refineFinal1_contig1.bam
```

This would need to be repeated on the remaining 23 XMAP files.

The second step merges all the chromosome-wise alignments, groups linear alignments into a primary read alignment and secondary read alignment components, computes Bonferroni corrected MAPQ confidence

adjustments, and splits linear alignments on long internal deletions. It can be run on a directory of BAM files produced in the first step as follows:

```
python3 -m merge_and_refine_bams \
  -i $output/contigs/alignmolvref/merge \
  -c $output/ref/hg38.fa.dict  \
  -o $output/contigs/alignmolvref/exp.ogm.bam

For Rare Variant Analysis (RVA) output conversion, substitute “data” for “contigs”
```

NOTE: The XMAP files (and accompanying CMAP files) in $output/contigs/alignmolvref are considered intermediate files and can be cleaned up by the pipeline unless otherwise specified when running the pipeline.

Hybrid Scaffold Conflict Cut Status File Format Specifications

This file format specification sheet details the file format specifications for Conflict Cut Status File Version 0.1. It is highly recommended that conflicts are viewed and managed within Bionano Access. Bionano Access provides advanced visual features and an audit trail to simplify the management of automatic cuts and introduction of new cuts in both the maps and NGS contigs.

In the Hybrid Scaffold pipeline, one important step is to identify and resolve conflicts. A conflict junction is the start position of a region where multiple data sources (e.g., NGS versus BNG) disagree, thus indicating possible assembly errors (see **Figure 11**). The Bionano conflict cut status file is a tab-delimited data file, which provides location information and the resolution strategy for each conflict junction detected in hybrid scaffold. The file can be opened in Excel for easy readability or in any tab-delimited, text-based editor.

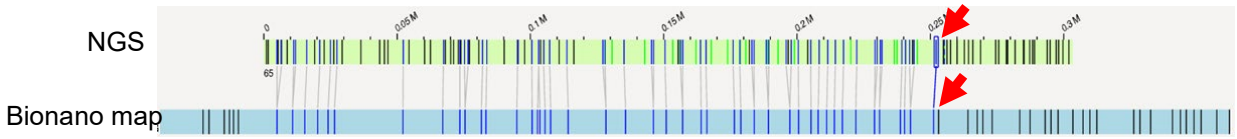

Figure 11. Example conflict between sequence and Bionano map. A significant number of unaligned labels outside the aligned region (left of the red arrows) indicate the presence of conflict between the two assemblies. Red arrows indicate the position of the conflict junction on the sequence and the genome map.

FORMAT

The conflict cut status file contains the following sections:

- Header
  - #
  - #
- Conflict cuts information block
  - First conflict
  - Next conflict [repeated for all conflicts]
  - Last conflict

HEADER SPECIFICATIONS

Header rows are prefixed by the pound sign (#), seen in **Table 68**. **NOTE**: *Denotes the required header line tag to read a conflict cut status file.

Table 68. General header field specifications

Header Line Tag	Header Line Description
#*	Defines the column name
#*	Defines the possible valid value for each column

HEADER SPECIFICATION DETAILS

Tables 69 and **70** provide the conflict cut status header's descriptions (including any specific formatting, limitations, and requirement) and examples.

Table 69. Header fields definition

#		
Header	#	
Description	Defines the columns for each data row in # rows:	
	xMapId	Each conflict is detected by an NGS-BNG alignment. This specifies the alignment id from which the conflicts are detected. Alignment Id is a non-negative number.
	refQry	This column indicates whether an action is going to be performed on the NGS contig or BNG map. By convention, 'ref' refers to NGS contigs and 'qry' refers to BNG maps.
	refId	The Id of the NGS contig where the conflict is detected. This can either be a positive Id number or '-1', which indicates this conflict is not associated with any NGS contig.
	leftRefBkpt	A conflict is designated as "left" or "right" depending on whether the unaligned region is to the left or right of the conflict junction. A particular NGS-BNG alignment can have a conflict junction to the left or to the right or both. This column specifies the coordinate of a conflict junction on the NGS contig.
	rightRefBkpt	The NGS coordinate of a conflict junction where the unaligned region is to the right of the site.
	alignmentOrientation	'+' indicates the BNG map is aligned to the NGS contig with same orientation, '-'indicates opposite orientation.
	ref_leftBkpt_toCut	This specifies whether to cut the NGS contig at the position specifies by 'leftRefBkpt.' The keyword 'cut' means cut the NGS contig and the keyword 'okay' means DO NOT cut the NGS contig.
	ref_rightBkpt_toCut	This specifies whether to cut the NGS contig at the position specified by 'rightRefBkpt' (see previous column).

#		
	ref_toDiscard	The keyword 'exclude' means excluding the NGS contig specified by 'refId' from hybrid-scaffold (i.e., this NGS contig will not be used in any subsequent steps in hybrid scaffold). The keyword 'okay' means to keep the NGS contig in hybrid scaffolding.
	refQrv	Same as column 2. see above.
	qryId	The Id of the BNG map from which the conflict is detected
	leftQryBkpt	Like 'leftRefBkpt,' this specifies the coordinate of the conflict junction on the BNG map.
	rightQryBkpt	Like 'rightRefBkpt' but specifies the coordinate on BNG map instead.
	alignmentOrientation	See column 6.
	qry_leftBkpt_toCut	Like 'ref_right_Bkpt_toCut' but specifies the action performed for BNG map.
	qry_rightBkpt_toCut	Like 'ref_right_Bkpt_toCut;' see above.
	qry_toDiscard	Discard the BNG maps specified by 'qryId' (see ref_toDiscard').
Example	`# xMapId<TAB>refQry<TAB> refId<TAB>leftRefBkpt<TAB>rightRefBkpt<TAB> alignmentOrientation<TAB>ref_leftBkpt_toCut<TAB>ref_rightBkpt_toCut<TAB> ref_toDiscard<TAB>refQry<TAB>qryId<TAB>leftQryBkpt<TAB>rightQryBkpt<TAB> alignmentOrientation<TAB>qry_leftBkpt_toCut<TAB>qry_rightBkpt_toCut<TAB> qry_toDiscard`	

Table 70. Definition of valid values for header fields

#	
Header	#
Description	Defines valid values allowed in each column. For each column, the set of valid values is separated by '/.' By convention, the last value in the set is a special value that indicates that this column is not relevant for this conflict. For example, in the first column the values 'id/-1' indicates that this column can either be a valid xmap ID or '-1' which indicates this conflict is not associated with any NGS-BNG alignment (i.e., the user may have found this conflict using external data).
Example	`# id/-1<TAB>ref<TAB>id/-1<TAB>position/-1<TAB>position/-1<TAB>+/-<TAB> okay/cut/-<TAB> okay/cut/-<TAB>okay/exclude/-<TAB>qry<TAB>id/-1<TAB> position/-1<TAB>position/-1<TAB>+/-<TAB>okay/cut/-<TAB>okay/cut/-<TAB>`

CONFLICT CUTS INFORMATION BLOCK

- Conflict cuts information block
  - First conflict

- Next conflict [repeated for all conflicts]
- Last conflict

EXAMPLE

# xMapId	refQry	refId	leftRefBkpt	rightRefBkpt	alignmentOrientation	ref_leftBkpt_toCut	ref_rightBkpt_toCut	ref_toDiscard	refQry	qryId	leftQryBkpt	rightQryBkpt	alignmentOrientation	qry_leftBkpt_toCut	qry_rightBkpt_toCut	qry_toDiscard
# id/-1	ref	id/-1	position/-1	position/-1	+/-	okay/cut/-	okay/cut/-	okay/exclude/-	qry	id/-1	position/-1	position/-1	+/-	okay/cut/-	okay/cut/-	okay/exclude/-
140	ref	96	-1	294731	-	okay	cut	okay	qry	5	-1	3541452	-	okay	okay	okay
660	ref	623	134613	-1	-	cut	okay	okay	qry	7	2790265	-1	-	okay	okay	okay
596	ref	548	-1	180307	-	okay	cut	okay	qry	27	-1	2853242	-	okay	okay	okay
326	ref	262	310534	-1	+	cut	okay	okay	qry	70	652954	-1	+	okay	okay	okay
105	ref	71	-1	329334	+	okay	cut	okay	qry	121	-1	906727	+	okay	okay	okay
89	ref	61	-1	189738	+	okay	cut	okay	qry	2	-1	2286421	+	okay	okay	okay
181	ref	130	469199	-1	+	cut	okay	okay	qry	107	1134675	-1	+	okay	okay	okay
532	ref	475	-1	322138	+	okay	cut	okay	qry	4	-1	5326556	+	okay	okay	okay

Figure 12. An Example of Conflict Cut Status File

Variant Annotation Pipeline

The Variant Annotation Pipeline enables users to determine if a Bionano structural variant (SV) call is relevant to certain phenotypes or disease traits. For more information about the pipeline, please refer to *Bionano Solve Theory of Operation: Variant Annotation Pipeline* (CG-30190). This document describes only the additional annotation columns. The output file of the Variant Annotation Pipeline is an annotated SMAP file, a general format to describe SVs detected by Bionano, with additional annotation columns appended. **NOTE**: The last few columns can vary depending on whether a trio, dual or single analysis has been performed upon execution of the Variant Annotation Pipeline (see **Tables 71** through **73**).

ANNOTATION COLUMNS – ALL ANALYSES

Table 71. Annotation columns common to all analyses

Annotation	Description
sample	The sample name.
algorithm	This is based on comparing the *de novo* assembly or the Rare Variant Pipeline SV calls of the sample with the reference; this is typically output as “assembly comparison.”
size	The size (in base pair) of insertion and deletion calls. It would be -1 for other variant types.
Present_in_%_of_BNG_control_samples	The percentage of samples in the Bionano control SV database that also carry the SV.
Present_in_%_of_BNG_control_samples_with_the_same_enzyme	Like above, but the percentage is calculated based on database samples having *the same enzyme* as the sample being annotated.
%_BNG_control_sample_with_homozygous_SV	The percentage of samples in the Bionano control SV database that are homozygous for the SV
%_of_BNG_control_sample with_heterozygous_SV	The percentage of samples in the Bionano control SV database that are heterozygous for the SV

Annotation	Description
%_of AFR_BNG_control_sample_with_SV	The percentage of African samples in the Bionano control SV database that carry the SV
%_of AMR_BNG_control_sample_with_SV	The percentage of Admixed American samples in the Bionano control SV database that carry the SV
%_of_EUR_BNG_control_sample_with_SV	The percentage of European samples in the Bionano control SV database that carry the SV
%_of_EAS_BNG_control_sample_with_SV	The percentage of East Asian samples in the Bionano control SV database that carry the SV
%_of_SAS_BNG_control_sample_with_SV	The percentage of South Asian samples in the Bionano control database that carry the SV
%_of_unknown_BNG_control_sample_with_SV	The percentage of Unknown population samples in the Bionano control database that carry the SV
Fail_assembly_chimeric_score	A flag used to denote whether there might be a potential chimeric join at the variant locus. This denotes whether a minimal chimeric quality score of 35 and coverage of 10X have been achieved around each SV breakpoint. A value of 'pass' means that the two criteria have been met; a ''failure'' denotes the criteria not met; and a 'not_applicable' value denotes that the check has not been performed. Notice that this check is performed only for inversion and translocation calls. **NOTE**: A chimeric quality score of a label on a genome map is the percent of molecules that align to both sides of the label out of all molecules that align on either side near this label. See *CMAP File Format Specification Sheet* (CG-30039) for details.
num_overlap_DGV_calls	If the sample is human, then the SVs would be compared against the Database of Genomic Variants (DGV), and the number of DGV variants overlapping the call is outputted.
OverlapGenes	A semi-colon separated list indicating which genes overlap with the SV.
NearestNonOverlapGene	The next closest gene to the SV.
NearestNonOverlapGeneDistance	The distance between the SV and the next closest gene.
PutativeGeneFusion	The list of fusion genes that may be created by the SV.

Annotation	Description
Found_in_self_molecules	This column denotes whether there are enough case/proband sample's molecules supporting the case/proband sample's genome map at the SV breakpoints. The possible values are 'yes' and 'no.' Since inversion_partial calls are not annotated, a value of '-' is shown for inversion_partial calls. **NOTE**: The minimum number of molecules required is defined as a parameter by the users upon running the variant annotation pipeline.
Self_molecule_count	The number of molecules supporting the SV.
UCSC_web_link1	If the sample is either human or mouse, then a weblink to the SV region in the UCSC genome browser would be created. If the SV is a translocation breakpoint, then the weblink goes to one side of the translocation breakpoint.
UCSC_web_link2	The weblink goes to the other side of the translocation or fusion breakpoint.
ISCN	International System for Human Cytogenomic Nomenclature notation for SV

TRIO ANALYSIS

Table 72. Annotation columns – trio analysis

Annotation	Description
Found_in_parents_assemblies	Whether the SV call is also identified in the father's or mother's assembly. The possible values are 'mother,' 'father,' 'both' and 'none.' Since inversion_partial calls are not annotated, a value of '-' is shown for any inversion_partial call.
Found_in_parents_molecules	This column shows whether there are enough parents' molecules supporting the proband's genome map at the SV breakpoints. The possible values are 'mother,' 'father,' 'both' and 'none.' Since inversion_partial calls are not annotated, a value of '-' is shown for inversion_partial calls. **NOTE**: The minimum numbers of molecules required are defined as parameters by the users upon running the variant annotation pipeline.
Mother_molecule_count	Number of molecules supporting the SV in the mother's assembly.
Father_molecule_count	Number of molecules supporting the SV in the father's assembly

DUAL ANALYSIS

Table 73. Annotation columns – dual analysis

Statistic	Description
Found_in_control_sample_assembly	Whether the SV call is also identified in the control sample's assembly. The possible values are 'yes' or 'no.' Since inversion_partial calls are not annotated, a value of '-' is shown for any inversion_partial call. See *Bionano Solve Theory of Operation: Variant Annotation Pipeline* (CG-30190).
Found_in_control_sample_molecules	This column shows whether there are enough control sample molecules supporting the case sample's genome map at the SV breakpoints. The possible values are 'yes' and 'no.' Since inversion_partial calls are not annotated, a value of '-' is shown for inversion_partial calls. **NOTE**: The minimum number of molecules required is defined as a parameter by the users upon running the variant annotation pipeline. See *Bionano Solve Theory of Operation: Variant Annotation Pipeline* (CG-30190).
Control_molecule_count	The number of molecules supporting the SV in the control sample

EnFocus™ FSHD Analysis JSON (*.json) file version 1.0.1.

The Bionano EnFocus™ FSHD Analysis Pipeline generates a JSON (JavaScript Object Notation) file that includes information about the analysis and summarizes the results. JSON is a generic open-standard file format, which relates keys (or attributes) to values. Bionano has adapted this format to store summary information from the FSHD analysis pipeline. For easy readability, JSON files can be opened in a text editor or specialized JSON viewers.

FORMAT

The data are organized in a hierarchy of key-value pairs (see **Figure 13**). The top level has two main sections: "sections" and "additional_info." The section "sections" contains data that Bionano Access uses for visualization and report generation. The section "additional_info" contains data that Bionano Access uses to generate a PDF report. The report version (from the key report_version) is also contained in this section. The keys are numbered (0, 1, 2, and so forth; see example in "Example JSON Output" section) to define the order in which the sections should appear in the PDF report.

The JSON contains the following sections:

- sections
  - Experiment information
    - Sample name
    - Enzyme used
    - Instrument serial number
    - Chip ID
    - Run ID
    - Date of data collection

        - Version of ICS software
    - Overall sample quality metrics
        - Inferred sex of sample
        - Assessment of molecule quality
        - Assessment of stable regions
    - Analysis information
        - Analysis performed.
        - Job ID
        - Job name
        - Operator name
        - Date of analysis
        - Version of assembly pipeline
        - Version of FSHD analysis pipeline
    - Detailed results
    - Screenshots
    - Additional information
    - Background information
        - Methods and limitations
        - References
- additional_info
    - report_version
    - display_columns
    - display_headers
    - display_widths
    - report_name

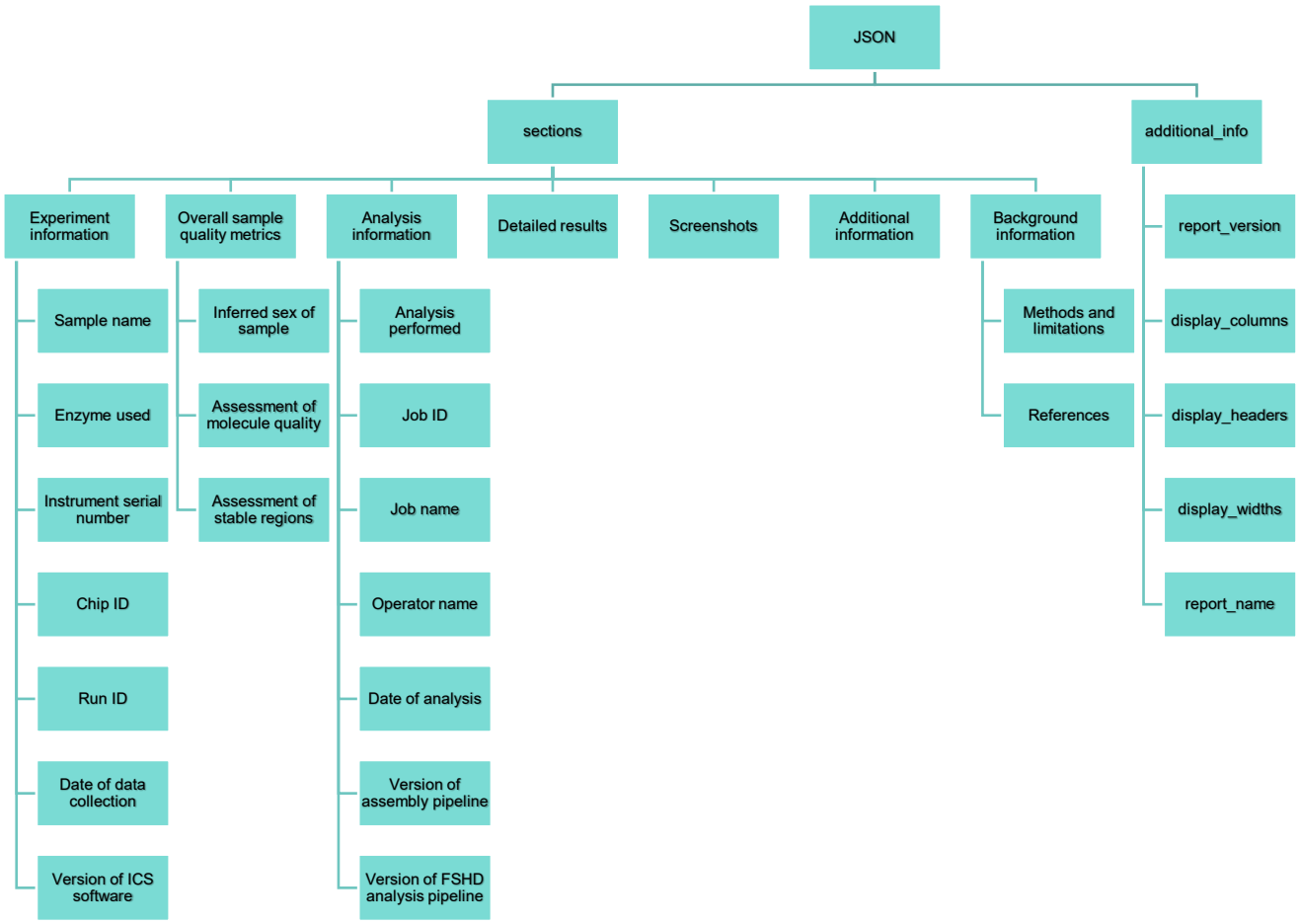

Figure 13. JSON document hierarchy of EnFocus FSHD Analysis output

SPECIFICATIONS: "SECTIONS"

In **Table 74**, there are seven sub-sections under "sections": "Experiment information," "Overall sample quality metrics," "Analysis information," "Detailed results," "Screenshots," "Additional information," and "Background information."

The "Experiment information" section includes information about the extracted and labeled DNA sample ("Sample name" and "Enzyme used"), the map data collection process ("Instrument serial number", "Chip ID", "Run ID", and "Date of data collection"), and the version of the imaging analysis software used to convert the image data into molecule data ("Version of ICS software"). Some of the information is passed into the pipeline by Bionano Access, so they may be absent if the pipeline is run on the command line.

Table 74. JSON properties - Experiment information section

Key	Description	Format	Example
Sample name	Name of the sample; corresponds to "Name" in Bionano Access. Defaulted to <sample_name> if not provided.	string	Sample_1

Key	Description	Format	Example
Enzyme used	Enzyme used to label the DNA; only DLE-1 is supported in Bionano Access	string	DLE-1
Instrument serial number	Serial number of the Bionano Saphyr instrument	string	SAPHYR_A1
Chip ID	Serial number of the chip followed by the flowcell number in parentheses	string	3RSBCYWNPMKXRNWU (Flowcell 2)
Run ID	Unique identifier for a chip run	string	4ba6a250-c593-41fe-b8bf-fd56ecee9e33
Date of data collection	Date and time when the data from the first scan is generated	datetime	2019-07-29 10:20:39 AM
Version of ICS software	Version of the ICS software used for analyzing the image data. Defaulted to “unknown” if unable to get information from input bnx file.	string	ICS 4.8.19085.2

The FSHD analysis pipeline assesses sample quality metrics to provide users with information about the data quality; the data is summarized as seen in **Table 75**, Overall sample quality metrics. The metrics and the results are divided into three subsections: “Inferred sex of the sample,” “Assessment of molecule quality,” and “Assessment of stable regions.” For more information, see *Bionano Solve Theory of Operation Bionano EnFocus™ FSHD Analysis* (CG-30321).

Table 75. JSON properties – Overall sample quality metrics section

Key	Description	Format	Example
Inferred sex of sample	Sex of the sample as inferred from the copy number analysis pipeline based on the molecule alignment (“coverage”) data. “NULL” if data is not available; otherwise, “male” or “female.”	string	female
Assessment of molecule quality	Quality of the molecules based on three criteria: molecule N50 (> 150 kbp) must be at least 200 kbp, effective coverage must be at least 75X, and map rate must be at least 70%. “NULL” if data is not available; otherwise, “PASS” or “FAIL.”	string	PASS
Assessment of stable regions	Quality of the consensus based on evaluation of regions considered stable. “NULL” if data is not available; otherwise, “PASS” or “FAIL.”	string	PASS

The “Analysis information” section includes information about the analysis being performed, shown in **Table 76**. Some of the information is passed into the pipeline by Bionano Access, so they may be absent if the pipeline is run on the command line.

Table 76. JSON properties - Analysis information section

Key	Description	Format	Example
Analysis performed	Name of the analysis	string	Bionano EnFocus™ FSHD Analysis
Job ID	Unique Job ID assigned by Bionano Access when the analysis is run. Defaulted to <job_id> if not provided.	string	123456
Job name	Name of the FSHD analysis job when the analysis is run in Bionano Access. Defaults to <object_name> if not provided.	string	Sample_1 DLE1 - FSHD Analysis_Solve3.5_11212019"
Operator name	Name of the user when the analysis is run in Bionano Access. Defaulted to <operator_name> if not provided.	string	John Doe
Date of analysis	The date and time when the FSHD analysis is run	datetime	2019-11-21 15:11
Version of assembly pipeline	Version of the assembly pipeline used for targeted assembly of the regions of interested	string	Bionano Solve 3.5
Version of FSHD analysis pipeline	Version of the Bionano EnFocus™ FSHD Analysis pipeline	string	Bionano EnFocus™ FSHD Analysis 1.0

The “Detailed results” section contains the necessary data for generating the results table in the PDF output report (see **Table 77**). The dataframe/table-like data is represented in a list of key-value pairs format. The keys correspond to column names in the table; the values correspond to the cell entries in the table. Each row contains data for a particular map that represents an allele.

The columns of the data are subject to change; the specific columns that are used in report generation are defined in the *additional_info* section as documented below. Selected columns are described below.

Table 77. JSON properties – Detailed results section

Key	Description	Format	Example
MapID	Identifier of a particular map from the assembly	int	22
Chr	Chromosome which the map (referenced in MapID) is from; either 4 or 10	int	4
Haplotype	Haplotype of the allele; 4qA or 4qB if the map is from chr4, and 10qA or 10qB if the map is from chr10. “unknown” if undermined	string	4qA
Count_repeat	Repeat count. If repeat is fully spanned, the pipeline would output an integer value. If not, the pipeline would output a lower bound value (for example >= 20).	int or string	5
Repeat_spanning_coverage	Number of molecules spanning the repeat region	int	30
Start_repeat	Label ID for the repeat start	int	56
End_repeat	Label ID for the repeat end	int	57

Key	Description	Format	Example
Start_haplotype	Position of label in basepairs for the haplotype start	float	600000.0
End_haplotype	Position of label in basepairs for the haplotype end	float	690000.0
Confidence	Deprecated	NA	NA
Map_alignment_confidence	Statistical confidence of alignment: Negative Log10 of p-value of alignment which is the same as the confidence value in XMAP	float	100.0
Anchor_to_mapend_map	Distance from the anchor label to the end of map. The anchor label refers to the label before repeat start.	float	10000.0
Array_length	Length of the repeat array interval	float	16.11
Count_length_consistency	Ratio of repeat count between before and after counting shift (See theory of operation for repeat shift)	float	1.01
Contains_SV	Indicates whether the map contains SVs proximal to the D4Z4 region; true/false	bool	true
ImageText	Text to be displayed in PDF report	string	Chromosome 4, Map 22 whose haplotype is 4qA has a calculated repeat count of 5
Count_repeat_mol	Deprecated	NA	NA
Merged	If other redundant maps have the same repeat number	bool	false
truncated_bool	If the map is truncated or not	bool	false
parsed_repeat_counts	Only applicable for the truncated map. Convert string “>= repeat number” to a numeric value (see theory of operation for “>=” sign in the truncated map)	int	10

The “Screenshots” section indicates where the screenshots (shown in PDF report) should be inserted. It does not contain data.

The “Additional information” section (see **Table 78**) includes statements indicating whether there may be additional SVs and/or CNVs of interest. The text can vary depending on whether there is presence or absence of such SVs and/or CNVs. The first statement is related to the presence or absence of SVs and/or CNVs proximal to the chr4 D4Z4 region; the second statement is related to the presence or absence of CNVs proximal to the SMCHD1 gene.

The “Background information” section has two subsections: “Methods and limitations,” which briefly describes the methods, and “References,” which lists publications that introduce FSHD and its analysis. The same text is shown in Bionano Access when a user sets up the FSHD analysis.

SPECIFICATIONS: "ADDITIONAL_INFO"

There are five key-value pairs under "Additional information": "report_version," "display_columns," "display_headers," "display_widths," and "report_name." These are used by Bionano Access, and they impact the PDF report generation.

Table 78. JSON properties – Additional information section

Key	Description	Format	Example
report_version	Version of the FSHD/JSON report	string	1.0.1
display_columns	Columns to be displayed in PDF report	list of string	["Chr", "MapID", "Count_repeat", "Haplotype", "Repeat_spanning_coverage"]
display_headers	Column names to be used in PDF report	list of string	["Chr", "Map ID", "Calculated repeat count (units)", "Haplotype", "Repeat-spanning coverage (X)"]
display_width	Column widths to be used in PDF report	list of int	[35, 40, 80, 60, 80]
report_name	Report name to be used in PDF report	string	Bionano EnFocus™ FSHD Analysis Report

EXAMPLE JSON OUTPUT


```
{
  "sections": {
    "0": {
      "Experiment information": {
        "0": {
          "Sample name": "Sample_1"
        },
        "1": {
          "Enzyme used": "DLE-1"
        },
        "2": {
          "Instrument serial number": "SAPHYR_A1"
        },
        "3": {
          "Chip ID": "3RSBCYWNPMKXRNWU (Flowcell 2)"
        },
        "4": {
          "Run ID": "4ba6a250-c593-41fe-b8bf-fd56ecee9e33"
        },
        "5": {
          "Date of data collection": "2019-07-29 10:20:39 AM"
        },
        "6": {
          "Version of ICS software": "ICS 4.8.19085.2"
        }
      }
    },
    "1": {
      "Overall sample quality metrics": {
        "0": {
          "Inferred sex of sample": "male"
        },
        "1": {
          "Assessment of molecule quality": "PASS"
        },
        "2": {
          "Assessment of stable regions": "PASS"
```

```
        }
      }
    },
    "2": {
      "Analysis information": {
        "0": {
          "Analysis performed": "Bionano EnFocus™ FSHD Analysis"
        },
        "1": {
          "Job ID": 123456
        },
        "2": {
          "Job name": "Sample_1 DLE1 - FSHD Analysis_Solve3.5_11212019"
        },
        "3": {
          "Operator name": "John Doe"
        },
        "4": {
          "Date of analysis": "2019-11-21 15:11"
        },
        "5": {
          "Version of assembly pipeline": "Bionano Solve 3.5"
        },
        "6": {
          "Version of FSHD analysis pipeline": "Bionano EnFocus™ FSHD Analysis 1.0"
        }
      }
    },
    "3": {
      "Detailed results": [
        {
          "MapID": 22,
          "Chr": 4,
          "Haplotype": "4qA",
          "Count_repeat": 5,
          "Repeat_spanning_coverage": 27,
          "Start_repeat": 110,
          "End_repeat": 111,
          "Start_haplotype": 769862.1,
          "End_haplotype": 787920.9,
          "Confidence": -1,
          "Map_alignment_confidence": 124.07,
          "Anchor_to_mapend_map": 32504,
          "Array_length": 16.11,
          "Count_length_consistency": 0.98,
          "Contains_SV": true,
          "ImageText": "Chromosome 4, Map 22 whose haplotype is 4qA has a calculated repeat count of
5",
          "Count_repeat_mol": -1,
          "Merged": false,
          "truncated_bool": false,
          "parsed_repeat_counts": -1
        },
        {
          "MapID": 290,
          "Chr": 4,
          "Haplotype": "4qB",
          "Count_repeat": 17,
          "Repeat_spanning_coverage": 23,
          "Start_repeat": 51,
          "End_repeat": 52,
          "Start_haplotype": 361995.5,
          "End_haplotype": 388109.3,
          "Confidence": -1,
          "Map_alignment_confidence": 54.83,
          "Anchor_to_mapend_map": 65341.09999999998,
          "Array_length": 56.85,
          "Count_length_consistency": 1.01,
          "Contains_SV": true,
          "ImageText": "Chromosome 4, Map 290 whose haplotype is 4qB has a calculated repeat count of
17",
          "Count_repeat_mol": -1,
          "Merged": false,
```

```
          "truncated_bool": false,
          "parsed_repeat_counts": -1
        },
        {
          "MapID": 11,
          "Chr": 10,
          "Haplotype": "10qA",
          "Count_repeat": 6,
          "Repeat_spanning_coverage": 43,
          "Start_repeat": 1017,
          "End_repeat": 1018,
          "Start_haplotype": 7713036.3,
          "End_haplotype": 7731665.1,
          "Confidence": -1,
          "Map_alignment_confidence": 1208.81,
          "Anchor_to_mapend_map": 35881.89999999944,
          "Array_length": 20.04,
          "Count_length_consistency": 1.01,
          "Contains_SV": true,
          "ImageText": "Chromosome 10, Map 11 whose haplotype is 10qA has a calculated repeat count of
6",
          "Count_repeat_mol": -1,
          "Merged": true,
          "truncated_bool": false,
          "parsed_repeat_counts": -1
        },
        {
          "MapID": 260,
          "Chr": 10,
          "Haplotype": "10qA",
          "Count_repeat": 15,
          "Repeat_spanning_coverage": 25,
          "Start_repeat": 40,
          "End_repeat": 41,
          "Start_haplotype": 417067.2,
          "End_haplotype": 435718.9,
          "Confidence": -1,
          "Map_alignment_confidence": 51.28,
          "Anchor_to_mapend_map": 66380.79999999999,
          "Array_length": 50.54,
          "Count_length_consistency": 1.02,
          "Contains_SV": true,
          "ImageText": "Chromosome 10, Map 260 whose haplotype is 10qA has a calculated repeat count of
15",
          "Count_repeat_mol": -1,
          "Merged": false,
          "truncated_bool": false,
          "parsed_repeat_counts": -1
        }
      ]
    },
    "4": {
      "Screenshots": "Screenshots to be inserted here"
    },
    "5": {
      "Additional information": "Structural variants and other copy number variants were detected in
the proximal chr4 region. No copy number variants were detected proximal to SMCHD1."
    },
    "6": {
      "Background information": {
        "0": {
          "Methods and limitations": "The Bionano EnFocus™ FSHD Analysis is performed based on whole-
genome optical mapping data collected on the Bionano Saphyr Genome Imaging Instrument. Based on
specific labeling and mapping of ultra-high molecular weight DNA in nanochannel arrays, optical mapping
enables high-resolution analysis of the D4Z4 repeat array.\n\nMolecules aligning to regions of interest
in chr4 and chr10 are extracted and assembled. The resulting consensus maps are used for the Bionano
EnFocus™ FSHD Analysis. The repeat arrays are sized, and the permissive and non-permissive alleles (4qA
and 4qB) assigned. Additional structural variants and copy number gains and losses are noted in the
proximity of the D4Z4 repeat array on chr4. Copy number gains and losses in the proximity of the SMCHD1
gene on chr18 are also noted.\n\nThe analysis data can be imported into Bionano Access, a graphical
user interface tool for visualization and curation. This method cannot detect single-nucleotide
variants that do not impact sequence motif sites and may miss small variants with potential functional
impacts."
```

```
        },
        "1": {
          "References": "Wijmenga et al. Chromosome 4q DNA rearrangements associated with
facioscapulohumeral muscular dystrophy. Nature Genetics (1992).\nDeidda et al. Direct detection of 4q35
rearrangements implicated in facioscapulohumeral muscular dystrophy (FSHD). J Med Genetics
(1996).\nZhang et al. Clinical application of single-molecule optical mapping to a multigeneration
FSHD1 pedigree. Molecular Genetics and Genomic Medicine (2019)."
        }
      }
    }
  },
  "additional_info": {
    "0": {
      "report_version": "1.0.1"
    },
    "1": {
      "display_columns": [
        "Chr",
        "MapID",
        "Count_repeat",
        "Haplotype",
        "Repeat_spanning_coverage"
      ]
    },
    "2": {
      "display_headers": [
        "Chr",
        "Map ID",
        "Calculated repeat count (units)",
        "Haplotype",
        "Repeat-spanning coverage (X)"
      ]
    },
    "3": {
      "display_widths": [
        35,
        40,
        80,
        60,
        80
      ]
    },
    "4": {
      "report_name": "Bionano EnFocus™ FSHD Analysis Report"
    }
  }
```


EnFocusTM Fragile X Analysis JSON v1.0.1 File Format Specifications

The Bionano EnFocusTM Fragile X Analysis Pipeline generates a JSON file that includes information about the analysis and summarizes the results (see **Figure 14**). JSON (JavaScript Object Notation) is a generic open-standard file format, which relates keys (or attributes) to values. Bionano has adapted this format to store summary information from the Fragile X analysis pipeline. For easy readability, JSON files can be opened in a text editor or specialized JSON viewers.

FORMAT

The data are organized in a hierarchy of key-value pairs. The top level has two main sections: “sections” and “additional_info.” The section “sections” contains data that Bionano Access uses for visualization and report generation. The section “additional_info” contains information that Bionano Access uses to generate a PDF report. The report version (from the key report_version) is also contained in this section. The keys are numbered (0, 1, 2, and so forth; see example in “Example JSON Output” section) to define the order in which the sections should appear in the PDF report.

The JSON contains the following sections:

- sections
  - Experiment information
    - Sample name
    - Enzyme used
    - Instrument serial number
    - Chip ID
    - Run ID
    - Date of data collection
    - Version of ICS software
  - Overall sample quality metrics
    - Inferred sex of sample
    - Assessment of molecule quality
    - Assessment of stable regions
  - Analysis information
    - Analysis performed
    - Job ID
    - Job name
    - Operator name
      - Date of analysis
      - Version of Bionano Access
      - Version of Bionano Solve
  - Detailed results
  - Screenshots
  - Additional information
  - Background information
    - Methods and limitations
    - References
- additional_info
  - report_version
  - display_columns
  - display_headers
  - display_widths
  - report_name

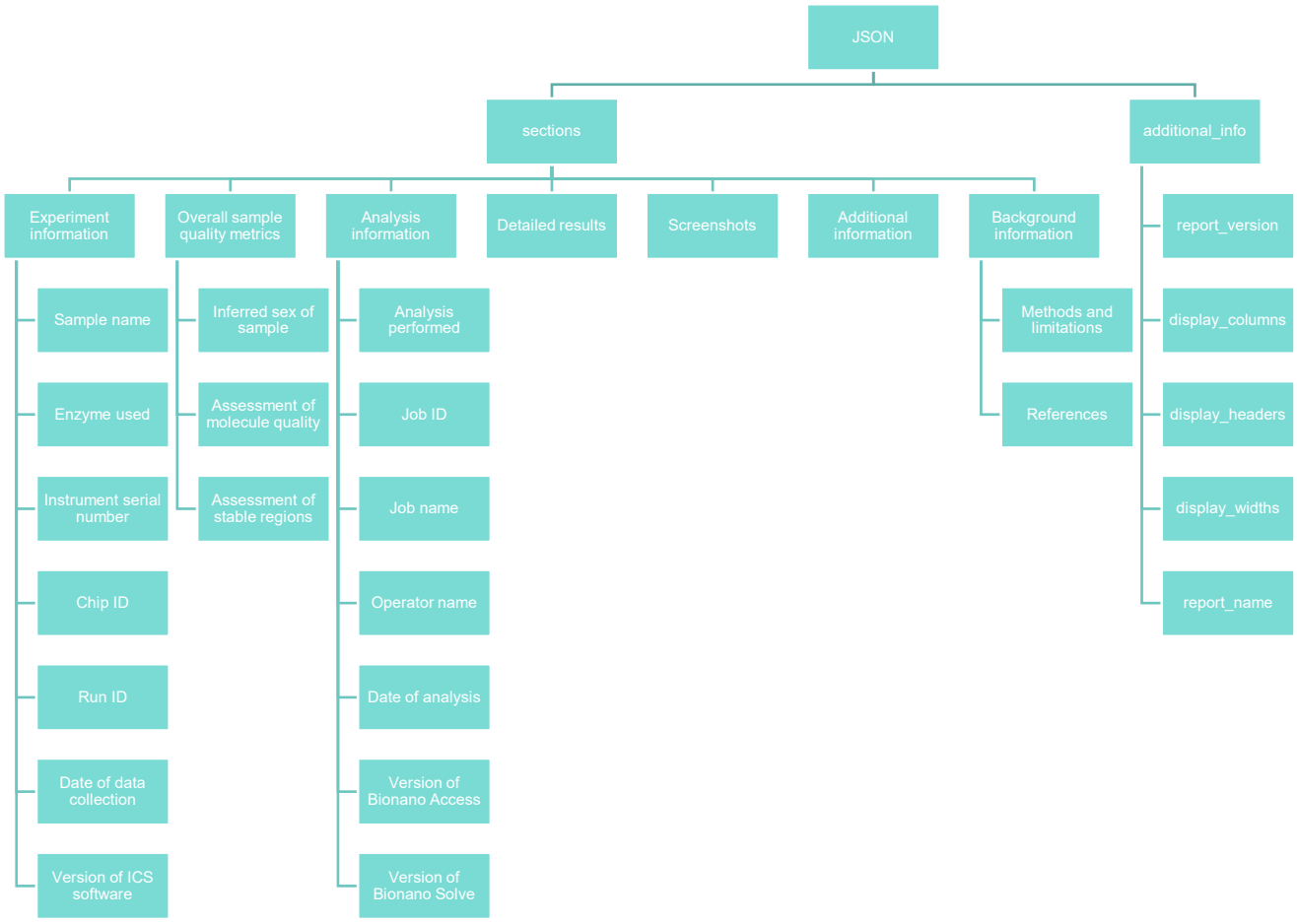

Figure 14. JSON document hierarchy of EnFocusä Fragile X Analysis output

SPECIFICATIONS: "SECTIONS"

There are seven sub-sections under "sections": "Experiment information," "Overall sample quality metrics," "Analysis information," "Detailed results," "Screenshots," "Additional information," and "Background information."

The "Experiment Information" section, seen in **Table 79**, includes information about the extracted and labeled DNA sample ("Sample name" and "Enzyme used"), the map data collection process ("Instrument serial number", "Chip ID", "Run ID", and "Date of data collection"), and the version of the imaging analysis software used to convert the image data into molecule data ("Version of ICS software"). Some of the information is passed into the pipeline by Bionano Access, so they may be absent if the pipeline is run on the command line.

Table 79. JSON properties - Experiment information section

Key	Description	Format	Example
Sample name	Name of the sample; corresponds to "Name" in Bionano Access. Defaulted to <sample_name> if not provided.	string	Sample_1
Enzyme used	Enzyme used to label the DNA; only DLE-1 is supported in Bionano Access	string	DLE-1

Key	Description	Format	Example
Instrument serial number	Serial number of the Bionano Saphyr instrument	string	SAPHYR_A1
Chip ID	Serial number of the chip followed by the flowcell number in parentheses	string	3RSBCYWNPMKXRNWU (Flowcell 2)
Run ID	Unique identifier for a chip run	string	4ba6a250-c593-41fe-b8bf-fd56ecee9e33
Date of data collection	Date and time when the data from the first scan is generated	datetime	2019-07-29 10:20:39 AM
Version of ICS software	Version of the ICS software used for analyzing the image data. Defaulted to “unknown” if unable to get information from input bnx file.	string	ICS 4.8.19085.2

Table 80 summarizes the Fragile X analysis pipeline which assesses sample quality metrics to provide users information about the data quality; the data is summarized in “Overall sample quality metrics.” The metrics and the results are divided into three subsections: “Inferred sex of the sample,” “Assessment of molecule quality,” and “Assessment of stable regions.” For more information, see *Bionano Solve Theory of Operation: Bionano EnFocus™ Fragile X Analysis* (CG-30457).

Table 80. JSON properties – Overall sample quality metrics section

Key	Description	Format	Example
Inferred sex of sample	Sex of the sample as inferred from the copy number analysis pipeline based on the molecule alignment (“coverage”) data. “NULL” if data is not available; otherwise, “male” or “female.”	string	female
Assessment of molecule quality	Quality of the molecules based on three criteria: molecule N50 (> 150 kbp) must be at least 200 kbp, effective coverage must be at least 75X, and map rate must be at least 70%. “NULL” if data is not available; otherwise, “PASS” or “FAIL.”	string	PASS
Assessment of stable regions	Quality of the consensus based on evaluation of regions considered stable. “NULL” if data is not available; otherwise, “PASS” or “FAIL.” Detailed information on the assessment of stable regions can be found in *Bonano Solve Theory of Operation: Bionano EnFocus™ Fragile X Analysis* (CG-30457).	string	PASS

The “Analysis information” section includes information about the analysis being performed, shown in **Table 81**. Some of the information is passed into the pipeline by Bionano Access, so they may be absent if the pipeline is run on the command line.

Table 81. JSON properties – Analysis information section

Key	Description	Format	Example
Analysis performed	Name of the analysis	string	Bionano EnFocus™ Fragile X Analysis

Key	Description	Format	Example
Job ID	Unique Job ID assigned by Bionano Access when the analysis is run. Defaulted to <job_id> if not provided.	string	123456
Job name	Name of the Fragile X analysis job when the analysis is run in Bionano Access. Defaulted to <object_name> if not provided.	string	Sample_1 DLE1 – FragileX Analysis_Solve3.7_09012021
Operator name	Name of the user when the analysis is run in Bionano Access. Defaulted to <operator_name> if not provided.	string	John Doe
Date of analysis	The date and time when the FSHD analysis is run	datetime	2021-09-14 22:46
Version of Bionano Access	Version of Access	string	1.7
Version of Bionano Solve	Version of bioinformatics tools	string	Bionano Solve 3.7

The “Detailed results” section contains the necessary data for generating the results table in the PDF output report. The dataframe/table-like data is represented in a list of key-value pairs format. The keys correspond to column names in the table; the values correspond to the cell entries in the table. Each row contains data for a particular map that represents an allele.

The columns of the data are subject to change; the specific columns that are used in report generation are defined in the “additional_info” section as documented in **Table 82** where selected columns are described.

Table 82. JSON properties – Detailed results section

Key	Description	Format	Example
Gene	Identifier of repeat gene	string	FMR1
Sample	Sample name	string	95552_6
Sex	Sex of sample	string	female
Chr	Chromosome where the repeat gene is located	string	X
Start_ref	Position of label in base pairs for the start of the interval of interest in the reference	int	147910189
End_ref	Position of label in base pairs for the end of the interval of interest in the reference	int	147918814
Interval_ref	Length of the interval of interest in the reference	int	8625
Count_repeat_ref	Repeat count in the reference	int	25
Repeat_unit_size	Repeat unit size	int	3

Key	Description	Format	Example
Irrelevant_ref	Length of the flanking irrelevant region in the reference	int	8550
MapID	Identifier of a particular map from the assembly	int	231
Start_repeat	Label ID for the repeat start	int	253
End_repeat	Label ID for the repeat end	int	254
I_2_start_ref	To correct for the irrelevant region, extra space needs to be added from the start of the observed interval (See FAQs in theory of operation for flanking region correction)	float	500.0
I_2_end_ref	To correct for the irrelevant region, extra space needs to be added from the end of the observed interval	float	500.0
Array_length	Length of the observed interval in kilo-base pairs	float	8.92
Unmatched_labels	Number of unmatched labels within the observed interval	int	1
Count_repeat_observed	Repeat count estimated by (observed interval – irrelevant_ref)/Repeat_unit_size	int	200
Count_repeat	Estimated repeat count with the maximum posterior probability	int	200
P >= expansion_cutoff	Probability that sample repeat number is greater than or equal to 200 units	string	99%
Expanded_repeat	Repeat cutoff for the full expansion	int	200
Realigned	If boundary labels are realigned	bool	False
CI_lower	Lower bound of repeat count for 99% credible interval	int	100
CI_upper	Upper bound of repeat count for 99% credible interval	int	300
Percentile	Percentage of negative control samples which have repeat number lower than the estimated repeat count	int	50
Repeat_spanning_coverage	Number of the molecules spanning the repeat region	int	30
Qry_contig_length	Total length of the consensus map	float	100000.0
ImageText	Text to be displayed in PDF report	string	Chromosome X, Map231 has a calculated repeat count of 548.

The “Screenshots” section indicates where the screenshots (shown in PDF report) should be inserted. It does not contain data.

The “Background information” section has two subsections: “Methods and limitations,” which briefly describes the methods, and “References,” which lists publications that introduce Fragile X and its analysis. The same text is shown in Bionano Access when a user sets up the Fragile X analysis.

SPECIFICATIONS: “ADDITIONAL_INFO”

Shown in **Table 83**,there are five key-value pairs under “additional_info”: “report_version,” “display_columns,” “display_headers,” “display_widths,” and “report_name.” These are used by Bionano Access, and they impact the PDF report generation.

Table 83. JSON properties – Additional info section

Key	Description	Format	Example
report_version	Version of the Fragile X/JSON report	string	1.0.1
display_columns	Columns to be displayed in PDF report	list of string	[“Gene”,”Sample,” “Chr,” “Count_repeat,” “P >= expansion_cutoff,” “CI_lower,” “CI_upper,” “Repeat_spanning_coverage”]
display_headers	Column names to be used in PDF report	list of string	[“Gene,” “Sample,” “Chr,” “Calculated repeat count,” “Probability >= 200 repeat units”, “99% credible interval lower bound”, “99% credible interval upper bound”, “Repeat-spanning coverage (X)”]
display_width	Column widths to be used in PDF report	list of int	[40, 80, 25, 55, 55, 50, 50, 50]
report_name	Report name to be used in PDF report	string	Bionano EnFocus™ Fragile Analysis Report

EXAMPLE JSON OUTPUT


```
{
  "sections": {
    "0": {
      "Experiment information": {
        "0": {
          "Sample name": "95552_6"
        },
        "1": {
          "Enzyme used": "DLE1"
        },
        "2": {
          "Instrument serial number": "SAPHYR_D08"
        },
        "3": {
          "Chip ID": "C7B2OJGNPPRSJNWU (Flowcell 3)"
        },
        "4": {
          "Run ID": "be595a62-82d3-4f8a-a95b-f097f720ba7b"
        },
        "5": {
          "Date of data collection": "2021-04-14 11:16:50 PM"
        },
        "6": {
          "Version of ICS software": "ICS 5.1.21018.2"
```

```
      }
    }
  },
  "1": {
    "Overall sample quality metrics": {
      "0": {
        "Inferred sex of sample": "female"
      },
      "1": {
        "Assessment of molecule quality": "PASS"
      },
      "2": {
        "Assessment of stable regions": "PASS"
      }
    }
  },
  "2": {
    "Analysis information": {
      "0": {
        "Analysis performed": "Bionano EnFocusTM Fragile X Analysis"
      },
      "1": {
        "Job ID": "123456"
      },
      "2": {
        "Job name": "Sample_1 DLE1 - FragileX Analysis_Solve3.7_09012021"
      },
      "3": {
        "Operator name": "John Doe"
      },
      "4": {
        "Date of analysis": "2021-09-14 22:46"
      },
      "5": {
        "Version of Bionano Access": "1.7"
      },
      "6": {
        "Version of Bionano Solve": "Bionano Solve 3.7"
      }
    }
  },
  "3": {
    "Detailed results": [
      {
        "Gene": "FMR1",
        "Sample": "95552_6",
        "Sex": "female",
        "Chr": "X",
        "Start_ref": 147910189,
        "End_ref": 147918814,
        "Interval_ref": 8625,
        "Count_repeat_ref": 25,
        "Repeat_unit_size": 3,
        "Irrelevant_ref": 8550,
        "MapID": 231,
        "Start_repeat": 213,
        "End_repeat": 214,
        "I_2_start_ref": -1,
        "I_2_end_ref": -1,
        "Array_length": 8.92,
        "Unmatched_labels": 0,
        "Count_repeat_observed": 122,
        "Count_repeat": 122,
        "P >= expansion_cutoff": "0.08%",
        "Expanded_repeat": 200,
        "Realigned": false,
        "CI_lower": 63,
        "CI_upper": 184,
        "Percentile": 99,
        "Repeat_spanning_coverage": 50,
        "Qry_contig_length": 2025503.5,
        "ImageText": "Chromosome X, Map231 has a calculated repeat count of 122"
      },
```

```
        {
          "Gene": "FMR1",
          "Sample": "95552_6",
          "Sex": "female",
          "Chr": "X",
          "Start_ref": 147910189,
          "End_ref": 147918814,
          "Interval_ref": 8625,
          "Count_repeat_ref": 25,
          "Repeat_unit_size": 3,
          "Irrelevant_ref": 8550,
          "MapID": 232,
          "Start_repeat": 214,
          "End_repeat": 215,
          "I_2_start_ref": -1,
          "I_2_end_ref": -1,
          "Array_length": 8.92,
          "Unmatched_labels": 0,
          "Count_repeat_observed": 122,
          "Count_repeat": 122,
          "P >= expansion_cutoff": "0.08%",
          "Expanded_repeat": 200,
          "Realigned": false,
          "CI_lower": 63,
          "CI_upper": 184,
          "Percentile": 99,
          "Repeat_spanning_coverage": 49,
          "Qry_contig_length": 2024381,
          "ImageText": "Chromosome X, Map232 has a calculated repeat count of 122"
        }
      ]
    },
    "4": {
      "Screenshots": "Screenshots to be inserted here"
    },
    "5": {
      "Background information": {
        "0": {
          "Methods and limitations": "The Bionano EnFocusTM Fragile X Syndrome analysis is performed
based on optical genome mapping (OGM) data collected on the Bionano Saphyr genome imaging instrument.
Based on specific labeling, alignment, and assembly of ultra-long DNA molecules in nanochannel arrays,
OGM enables for assessment of CGG expansions in the FMR1 gene locus.\n\nMolecules aligning to the
region of interest on chrX are extracted and assembled. The resulting consensus maps are used for the
Bionano EnFocusTM Fragile X syndrome analysis. The size of the CGG repeat array in the FMR1 gene is
inferred based on the measured distance between two neighboring labels on the assembled map(s) that
contain the FMR1 gene. By incorporating a known set of control map measurements, the pipeline estimates
the most likely CGG repeat count as well as the credible intervals for the uncertainty of the repeat
count. We also compute the probability that the repeat count exceeds the pathogenic threshold of >200
repeat unit. The expanded CGG segment silences the FMR1 gene expression, which in turn disrupts nervous
system functions leading to learning and cognitive impairment seen in Fragile X syndrome.\n\nThe
analysis data can be imported into Bionano Access, a graphical user interface tool for visualization
and curation. This method cannot detect single-nucleotide variants that do not impact sequence motif
sites and may miss small variants with potential functional impacts. Because it is impossible to
exclude other sequence insertions within the expansion interval or other repeat expansion (i.e.,
containing AGG interruptions), repeat expansion is inferred and any increase in length is assumed to be
CGG expansion. \n\nOptical Genome mapping is intended for research use only; it is not a diagnostic
test.  "
        },
        "1": {
          "References": "Hunter JE, Berry-Kravis E, Hipp H, Todd PK. FMR1 Disorders. 1998 Jun 16
[updated 2019 Nov 21]. In: Adam MP, Ardinger HH, Pagon RA, Wallace SE, Bean LJH, Mirzaa G, Amemiya A,
editors. GeneReviews® [Internet]. Seattle (WA): University of Washington, Seattle; 1993-2021. Available
from http://www.ncbi.nlm.nih.gov/books/NBK1384/\nSahajpal, N. et. al. Optical Genome Mapping as a Next-
Generation Cytogenomic Tool for Detection of Structural and Copy Number Variations for Prenatal Genomic
Analyses. Genes (Basel). 2021 Mar; 12(3): 398."
        }
      }
    }
  },
  "additional_info": {
    "0": {
      "report_version": "1.0.1"
    },
```

```
    "1": {
      "display_columns": [
        "Gene",
        "Sample",
        "Chr",
        "Count_repeat",
        "P >= expansion_cutoff",
        "CI_lower",
        "CI_upper",
        "Repeat_spanning_coverage"
      ]
    },
    "2": {
      "display_headers": [
        "Gene",
        "Sample",
        "Chr",
        "Calculated repeat count",
        "Probability >= 200 repeat units",
        "99% credible interval lower bound",
        "99% credible interval upper bound",
        "Repeat-spanning coverage (X)"
      ]
    },
    "3": {
      "display_widths": [
        40,
        80,
        25,
        55,
        55,
        50,
        50,
        50
      ]
    },
    "4": {
      "report_name": "Bionano EnFocusTM Fragile X Analysis Report"
    }
  }
```

Absence/Loss of Heterozygosity Pipeline File Format Specifications

The Bionano AOH/LOH detection pipeline detects regions with absence or loss of heterozygosity based on the zygosity of structural variants detected by the *de novo* Assembly pipeline. For more information about AOH/LOH detection, please refer to *Bionano Solve Theory of Operation: Structural Variant Calling* (CG-30110). The output of the pipeline consists of two tab-delimited, text-based files; one file describes each call, and one file provides detailed information about the structural variants that were used during detection. This file format specification sheet provides descriptions of the AOH/LOH output file headers and descriptions of the columns in each file.

When the data are imported into Bionano Access, the AOH/LOH output files are automatically processed and visualized. AOH/LOH output files can also be opened in Excel for easy readability, or in any tab-delimited, text-based editor. See **Tables 84** through **90**.

AOH/LOH Calls

FORMAT

The Bionano Absence/Loss of Heterozygosity output file loh_calls.txt contains the following sections:

- AOH/LOH file header
  - # SMAP Entries From:

- # Package:
- # HMM parameters:
- #h
- #f

- AOH/LOH calls block (each row as defined by the column headers in #h)
  - Unique ID for the AOH/LOH call [Id]
  - The position along the reference and width of each call [Chromosome, Start, End, Width]
  - Followed by the confidence score of the AOH/LOH call and their corresponding SV type [Confidence].

HEADER SPECIFICATIONS

Header rows are prefixed by the pound sign (#). "*" Denotes required header line tags.

Table 84. AOH/LOH header fields - overview

Header Line Tag	Header Line Description
# Smap Entries From:	SMAP file used to make AOH/LOH calls*
# Package:	Package name, version, and installation date*
# HMM Parameters:	Parameters used by Hidden Markov Model
#h	The columns for each data row
#f	The numerical data type for each data column

HEADER SPECIFICATION DETAILS

Table 85. Smap entries from header fields

# Smap Entries From	
Header	# Smap Entries From:
Description	SMAP file used to make AOH/LOH calls, auto generated.
Example	# Smap Entries From:<TAB>merged_smaps/exp_refineFinal1_merged_filter_inversions.smap

Table 86. Package header fields

# Package	
Header	# Package:
Description	Package name, version, and installation date, auto-generated.
Example	# Package: lohdetection. branch: master. Commit hash f47e05d21e761bda477230058f793. Installation date: 2021-10-01 09:56:38.621762

Table 87. Hidden Markov Model header fields

# HMM parameters	
Header	# HMM parameters:
Description	Parameters used for Hidden Markov Model.
Example	# HMM parameters: transm_bg2bg = 0.959011191677458; transm_loh2loh = 0.8042549108141759; startprob_bg = 0.8215713321165973; emissionprob_bg_hom = 0.31788854922897; emissionprob_loh_hom = 0.975137538627914

Table 88. Header field format definitions

#f	
Header	#f
Description	Defines the numerical data type for each data column.
Example	#f<TAB>int<TAB>int<TAB>float<TAB>float<TAB>float<TAB>float

Table 89. Header field format definition

#h		
Header	#h	
Description	Defines the columns for each data row in #h rows:	
	Id	
	Chromosome	Map ID, ordinal number
	Start	Start position of AOH/LOH call on map [0- based from map start] in basepairs

#h		
	End	End position of AOH/LOH call on map [0- based from map start] in basepairs
	Width	AOH/LOH call width in basepairs
	Confidence	The probability that the AOH/LOH call is a true event. More specifically, this is the model's precision when detecting simulated AOH/LOH events of a similar width.
Example	#h Id<TAB>Chromosome<TAB>Start<TAB>End<TAB>Width<TAB>Confidence	

ANNOTATED AOH/LOH FILE

Annotated AOH/LOH calls can be found in the loh_calls_annotation_results.txt file. In addition to the columns present in the base AOH/LOH results file, the following additional columns are present.

Table 90. AOH/LOH annotation fields

Annotation	Description
OverlapGenes	A semi-colon separated list indicating which genes overlap with the AOH/LOH region.
NearestNonOverlapGene	The next closest gene to the AOH/LOH region.
NearestNonOverlapGeneDistance	The distance between the AOH/LOH region and the next closest gene.
UCSC_web_link1	If the sample is either human or mouse, then a weblink to the AOH/LOH region in the UCSC genome browser would be created.
ISCN	International System for Human Cytogenomic Nomenclature notation for AOH/LOH

AOH/LOH per SV Info

FORMAT

The Bionano Absence/Loss of Heterozygosity output file loh_per_sv_info.txt contains details about the subset of SVs from the input SMAP file that were used in AOH/LOH detection, with a few additional columns containing information about the AOH/LOH calls (see **Tables 91** through **96**). The file contains the following sections:

- AOH/LOH file header

- # SMAP Entries From:
- # Package:
- # HMM parameters:
- #h
- #f

- AOH/LOH calls block (each row as defined by the column headers in #h)
  - After the 2 IDs [SmapEntryID, RefcontigID1] are the positions along the reference of each SV [RefStartPos, RefEndPos].
  - Followed by the confidence scores of the SV calls, their corresponding SV type, and zygosity of the SV [Confidence, Type, Zygosity].
  - The final columns indicate whether the SV was determined to be in an AOH/LOH region, and the calculated probability that the SV is in an AOH/LOH region [In_AOH_LOH, AOH_LOH_prob].

HEADER SPECIFICATIONS

Header rows are prefixed by the pound sign (#). "*" Denotes required header line tags.

Table 91. AOH/LOH Per SV header fields - overview

Header Line Tag	Header Line Description
# Smap Entries From:	SMAP file used to make AOH/LOH calls*
# Package:	Package name, version, and installation date*
# HMM Parameters:	Parameters used by Hidden Markov Model
#h	The columns for each data row
#f	The numerical data type for each data column

Table 92. Smap Entries From header fields

# Smap Entries From	
Header	# Smap Entries From:
Description	SMAP file used to make AOH/LOH calls, auto generated.
Example	# Smap Entries From:<TAB>merged_smaps/exp_refineFinal1_merged_filter_inversions.smap

Table 93. Package header fields

# Package	
Header	# Package:

Description	Package name, version, and installation date, auto-generated.
Example	# Package: lohdetection. branch: master. Commit hash f47e05d21e761bda477230058f793. Installation date: 2021-10-01

Table 94. Hidden Markov Model header fields

# HMM parameters	
Header	# HMM parameters:
Description	Parameters used for Hidden Markov Model.
Example	# HMM parameters: transm_bg2bg = 0.959011191677458; transm_loh2loh = 0.804254910 8141759; startprob_bg = 0.8215713321165973; emissionprob_bg_hom = 0.317888549228 97; emissionprob_loh_hom = 0.975137538627914

Table 95. Header fields definition

#h		
Header	#h	
Description	Defines the columns for each data row in #h rows:	
	SmapEntryID	A unique number for an entry in the input SMAP file
	RefcontigID1	Reference contig ID (XmapID1). Map ID of the reference map from the .cmap reference file (the .cmap file may contain multiple reference maps). **NOTE**: RefContigIDs must be integers, but they need not be sequential.
	RefStartPos	Coordinate of reference contig ID1 aligned site which borders this SV. This site is always either a start or end of XmapID1 and it matches the site at the query start position (QryStartPos).
	RefEndPos	Coordinate of reference contig ID2 aligned site which borders this SV. This site is always either a start or end of XmapID2 and it matches the site at the query end position (QryEndPos).
	Confidence	Estimate of probability of being correct for insertions and deletions, and a quality metric for inversion and translocation breakpoints. Other SVs are given a placeholder value of '-1.00'. See *Bionano Solve Theory of Operation: Structural Variant Calling* (CG-30110).
	Type	Type of SV. For definitions, see *SMAP File Format Specification Sheet* (CG-30041).

#h		
	Zygosity	One of 'homozygous', 'heterozygous' or 'unknown' based on overlap with other SVs and alignments.
	In_AOH_LOH	Whether this SV was determined to be in an AOH/LOH region. '1' for yes, '0' for no.
	AOH_LOH_prob	The probability of the SV being in an AOH/LOH region as calculated by the Hidden Markov Model.
Example	#h SmapEntryID<TAB>RefcontigID1<TAB>RefStartPos<TAB>RefEndPos<TAB>Confidence<TAB>Type <TAB>Zygosity<TAB>In_AOH_LOH<TAB>AOH_LOH_prob	

Table 96. Header fields format definition

#f	
Header	#f
Description	Defines the numerical data type for each data column.
Example	#f<TAB>int<TAB>int<TAB>int<TAB>float<TAB>float<TAB>string<TAB>string<TAB>int<TAB>float

Copy Number Variant Annotation Pipeline File Format Specifications

The Variant Annotation Pipeline enables users to determine if a Bionano copy number variant (CNV) call is relevant to certain phenotypes or disease traits (see **Tables 97** through **99**). For more information about the pipeline, please refer to *Bionano Solve Theory of Operation: Variant Annotation Pipeline* (CG-30190). The output file of the Variant Annotation Pipeline is an annotated CNV results file, with additional annotation columns appended. The CNV file format is a general format to describe CNVs detected by Bionano; please refer to *Bionano Solve Theory of Operation: Structural Variant Calling* (CG-30110) for details on the CNV calling algorithm and output files.

This document describes only the additional annotation columns. **NOTE**: The last few columns can vary depending on whether a trio, dual or single analysis has been performed upon execution of the Variant Annotation Pipeline.

ANNOTATION COLUMNS – ALL ANALYSES

Table 97. CNV annotation fields – all analyses

Statistic	Description
OverlapGenes	A semi-colon separated list indicating which genes overlap with the CNV.

Statistic	Description
NearestNonOverlapGene	The next closest gene to the CNV.
NearestNonOverlapGeneDistance	The distance between the CNV and the next closest
num_overlap_DGV_calls	If the sample is human, then the CNVs would be compared against the Database of Genomic Variants (DGV), and the number of DGV variants overlapping the call is outputted.
UCSC_web_link1	If the sample is either human or mouse, then a weblink to the CNV region in the UCSC genome browser would be created.
ISCN	If the sample is human, then the CNVs would be annotated with the International System for Human Cytogenomic Nomenclature (ISCN) notation for CNV

TRIO ANALYSIS

Table 98. CNV annotation fields – trio analysis

Statistic	Description
Found_in_parents	Whether the CNV call is also identified in the father's or mother's assembly. The possible values are 'mother,' 'father,' 'both' and 'none.'

DUAL ANALYSIS

Table 99. CNV annotation columns – dual analysis

Statistic	Description
Found_in_control_paired	Whether the CNV call is also identified in the control sample's assembly. The possible values are 'yes' or 'no.'

Technical Assistance

For technical assistance, contact Bionano 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@bionano.com
Phone	Hours of Operation: Monday through Friday, 9:00 a.m. to 5:00 p.m., PST US: +1 (858) 888-7663 Monday through Friday, 9:00 a.m. to 5:00 p.m., CET UK: +44 115 654 8660 (UK) France: +33 5 37 10 00 77 Belgium: +32 10 39 71 00
Website	www.bionano.com/support
Address	Bionano, Inc. 9540 Towne Centre Drive, Suite 100 San Diego, CA 92121

Legal Notice

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

Patents

Products of Bionano Genomics® may be covered by one or more U.S. or foreign patents.

Trademarks