

DRAGEN v4.5.2

For Illumina TruPath Genome prep

Software Release Notes

Introduction

These release notes detail the key features the Illumina® DRAGEN™ Secondary Analysis Software v4.5.2 for Illumina TruPath™ Genome Prep.

DRAGEN™ Installers, Resource Files, and Release Notes are available here:

https://support.illumina.com/sequencing/sequencing_software/dragen-bio-it-platform.html

DRAGEN™ v4.5 User Guide is now available here:

<https://help.dragen.illumina.com>

The software package includes downloadable installers for Phase 4 on-site servers:

- DRAGEN™ SW for x86 Oracle 8 - dragen-4.5.2-12.multi.el8.x86_64.run

The following configurations containing DRAGEN™ 4.5.2 are also available on request:

- AlmaLinux 8 Amazon Machine Images (AMIs) for f instances, available in 12 regions
- AlmaLinux 8 Microsoft Azure Image (VM) available in West US 2 for BYOL
- el8 compatible RPM packages for use with Amazon Web Services (AWS) f instances, for customer generated AMIs or customer generated docker images
- DRAGEN™ Kernel drivers for el8, for use with customer generated AMIs or QuickStart

Contents

Overview.....	4
DRAGEN Germline Pipeline for Illumina TruPath Genome.....	4
Summary.....	4
Pipeline Overview.....	4
Proximity Mode Analysis in DRAGEN.....	5
Key Benefits of TruPath vs standard Illumina SBS.....	6
• Phased, High-Quality Small Variant Calls in Clinically Relevant Gene Families.....	6
• Improved STR Expansion Length and Classification Accuracy ..	7
• Improved BND Filtering	8
Usage.....	8
Proximity Linking Model.....	9
• Sample Collection.....	9
Map/Align.....	11
Template Tagging.....	11
▪ Outputs.....	12
Phasing.....	16
• Phasing Options.....	17
• Phasing Output Files.....	17
Structural Variant Calling.....	20
• Leveraging TruPath proximity-linked features.....	20
• SV VCF Outputs.....	21
Multi-Region Joint Detection.....	21
• MRJD Method.....	22
• MRJD Outputs.....	23
• Visualize MRJD results in IGV.....	26
• Visualize MRJD results in DRAGEN Reports.....	27
• MRJD Notes.....	27
STR Calling.....	28
• In-Repeat Read (IRR) Recovery.....	28
• Phasing.....	29
• Sequencing Efficiency Correction.....	29
Colocation Maps.....	30
• Colocation Map Generation.....	31
• Cooler File.....	31
• Colocation Filter.....	31
Targeted Calling from TruPath Data.....	33
Proximity Coverage Reports.....	33
DRAGEN-Reports.....	34
Pipeline Limitations.....	35
TruPath Genome Licensing.....	35
Resource Files.....	37
Reference Genome Recommendations.....	37
Known Issues.....	38
SW Installation Procedure.....	39
Release History.....	39

Overview

v4.5.2 is the initial release of the DRAGEN Secondary Analysis Software that supports Germline WGS analyses on data generated by the Illumina TruPath™ Genome Prep. This release is intended only for this analysis.

DRAGEN Germline Pipeline for Illumina TruPath Genome

DRAGEN's Germline pipeline integrates proximity mapped reads from the Illumina TruPath Genome prep to enhance genomic analysis using long-range information encoded on the flowcell. This proximity-aware workflow supports highly accurate read mapping, phasing, and variant detection, including structural variants, paralog-resolved small variants, short tandem repeat (STR) genotyping, and colocation analysis. By modeling and applying read-to-read linkage probabilities, the pipeline enables more confident interpretation of complex and low-mappability genomic regions using standard short-read data.

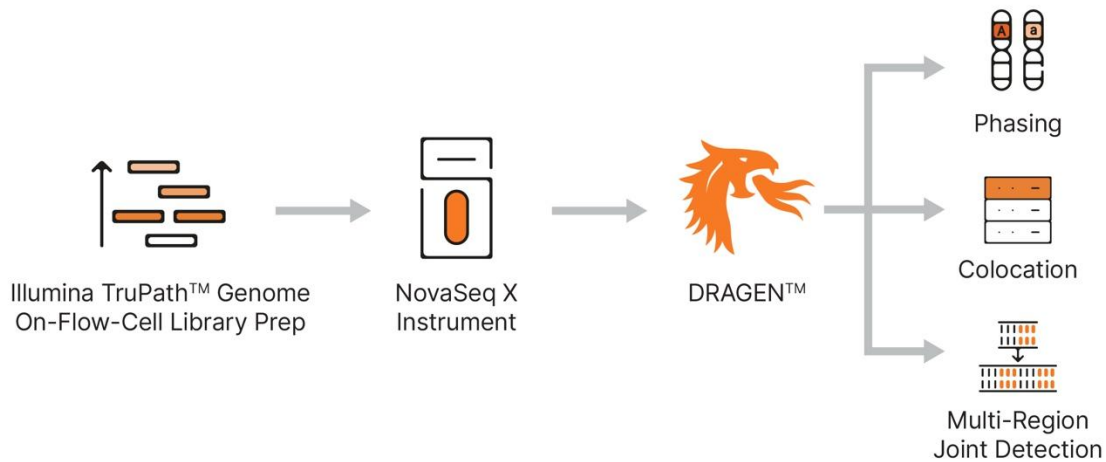
Summary

- **Integrated TruPath proximity mapping:** Enabling `--enable-proximity=true` activates proximity-aware modeling and analysis across the DRAGEN Germline pipeline, allowing reads that are spatially close on the flowcell to be probabilistically linked as originating from the same DNA template.
- **Proximity model-driven mapping and alignment:** DRAGEN performs a preliminary mapping pass to collect high-confidence alignments and fits a non-linear proximity linking model that relates flowcell spatial distance and genomic distance to read-to-read linkage probability. The resulting Phred-scaled linkage probability lookup table is applied during `map/align` to resolve ambiguous mappings and improve read placement accuracy in repetitive and complex genomic regions.
- **Enhanced phasing support:** Proximity information strengthens read phasing by associating reads from the same original template molecule, enabling longer and more reliable phasing blocks that propagate into variant calling and assembly-based analyses.
- **Structural variant calling:** The Germline SV caller leverages proximity-derived phasing to support phased assemblies, haplotype-aware machine-learning features, and haplotype-resolved genotyping for single-sample TruPath whole-genome analyses.
- **Haplotype-resolved small variant detection in paralogs:** For a given set of clinically relevant paralogous regions, Multi-Region Joint Detection (MRJD) uses read depth to estimate total copy number, constructs all paralog copies using the estimated total copy number, read sequences, and proximity information, assign copies to genomic regions or haplotypes, and calls small variants based on the constructed copies.
- **STR genotyping with IRR recovery:** Proximity linking enables recovery and placement of in-repeat reads (IRRs) that would otherwise be unmapped, improving detection and sizing of large STR expansions and supporting phasing-aware genotyping.
- **Colocation analysis and filtering:** Colocation maps summarize long-range genomic interactions using proximity-linked reads and are used to visualize structural features and filter SV breakends lacking proximity support.
- **Specialized outputs and reporting:** The pipeline generates proximity-aware BAM/CRAM files, VCFs, JSON summaries, cooler files, and TruPath-specific DRAGEN Reports with dedicated QC metrics and visualizations.

Pipeline Overview

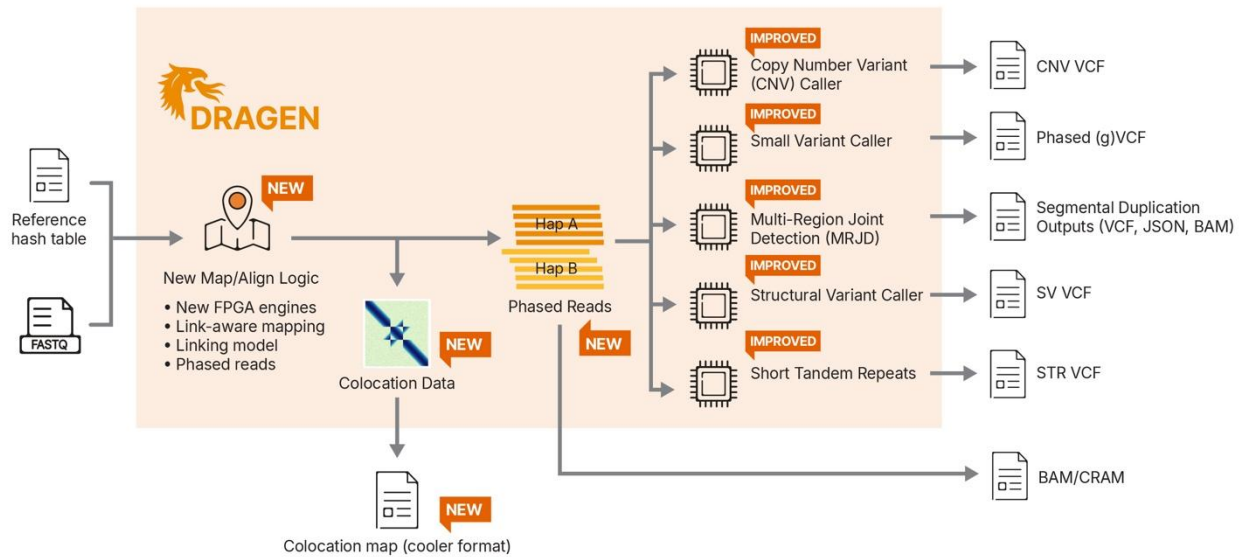
Short-read DNA sequencing typically captures genomic variation at high accuracy but lacks long-range context needed to confidently resolve complex regions such as repeats, paralogs, and structural variants. The **Illumina TruPath Genome Prep** encodes long-range molecular information directly on the flowcell by preserving spatial proximity between reads derived from the same original DNA molecule. When combined with DRAGEN's proximity-aware algorithms, this information enables long-range analysis that extends the power of standard short-read data.

The **DRAGEN Germline pipeline for Illumina TruPath Genome** leverages this flowcell-encoded proximity information through a probabilistic proximity linking model that assigns read-to-read linkage probabilities based on spatial and genomic distance. When proximity mode is enabled, DRAGEN automatically fits this model, generates Phred-scaled proximity link probability distributions, and applies them across mapping, phasing, and variant calling workflows. These proximity linkage probabilities serve as a foundational signal reused throughout the pipeline—informing alignment scoring, phasing blocks, candidate assemblies, machine-learning features, and variant filtering—to improve accuracy and confidence in repetitive and structurally complex genomic regions while remaining compatible with standard short-read sequencing workflows and formats.



Proximity Mode Analysis in DRAGEN

When proximity mode is enabled, DRAGEN automatically performs additional modeling and downstream analyses that integrate proximity information throughout the Germline pipeline. TruPath-specific proximity analysis is activated by enabling proximity during a DRAGEN Germline run setting `--enable-proximity=true`. This proximity-aware processing supports the following workflow and features:



- High-accuracy read mapping using linkage-informed alignment scoring
- Enhanced phasing via read-to-template association
- Structural variant calling using phased assemblies and haplotype-aware algorithms
- Paralog-resolved small variant detection with Multi-Region Joint Detection (MRJD)
- Improved STR genotyping through in-repeat read (IRR) recovery
- Long-range genomic interaction analysis and SV filtering using colocation maps

Key Benefits of TruPath vs standard Illumina SBS

DRAGEN Germline with proximity mode enabled for TruPath Genome provides several key benefits, including but not limited to improved small variant calling, ultra-long phase blocks, phased genes, and improved structural variant recall. See table below for additional details.

Benefit	TruPath, high molecular weight input DNA	TruPath, standard molecular weight input DNA	Standard Illumina SBS on DRAGEN 4.4
Best-in-class small variant calling performance	36,717 FP+FN	40,267 FP+FN	61,288 FP+FN
Multi-megabase phasing blocks	8.1 Mbp	649 kbp	NA
Fully phased genes	98.4%	87.6%	0%
Improved SV recall	94.0%	93.7%	80.7%

- **Phased, High-Quality Small Variant Calls in Clinically Relevant Gene Families**

TruPath enables haplotype-resolved, copy-number aware small variant calls in 10 clinically relevant gene sets with MRJD

Paralogous gene	Disease relevance
PMS2	Lynch Syndrome
SMN1-SMN2	Spinal Muscular Atrophy
NCF1	Chronic Granulomatous Disease
CYP21A2	Congenital Adrenal Hyperplasia
TNXB	Ehlers-Danlos syndrome
STRC	Recessive Nonsyndromic Hearing Loss
CYP2D6	Pharmacogenetics
CYP11B1-CYP11B2	Glucocorticoid-remediable Aldosteronism
CFHR1-CFHR2-CFHR3-CFHR4	Atypical Hemolytic Uremic Syndrome
USP18	Type I Interferonopathy

As an example, MRJD uses TruPath proximity information to generate haplotype-resolved variant calls in both PMS2 and PMS2CL (shown as copy 1 and copy 2 for each locus), with on-market long-read data shown for comparison (bottom).

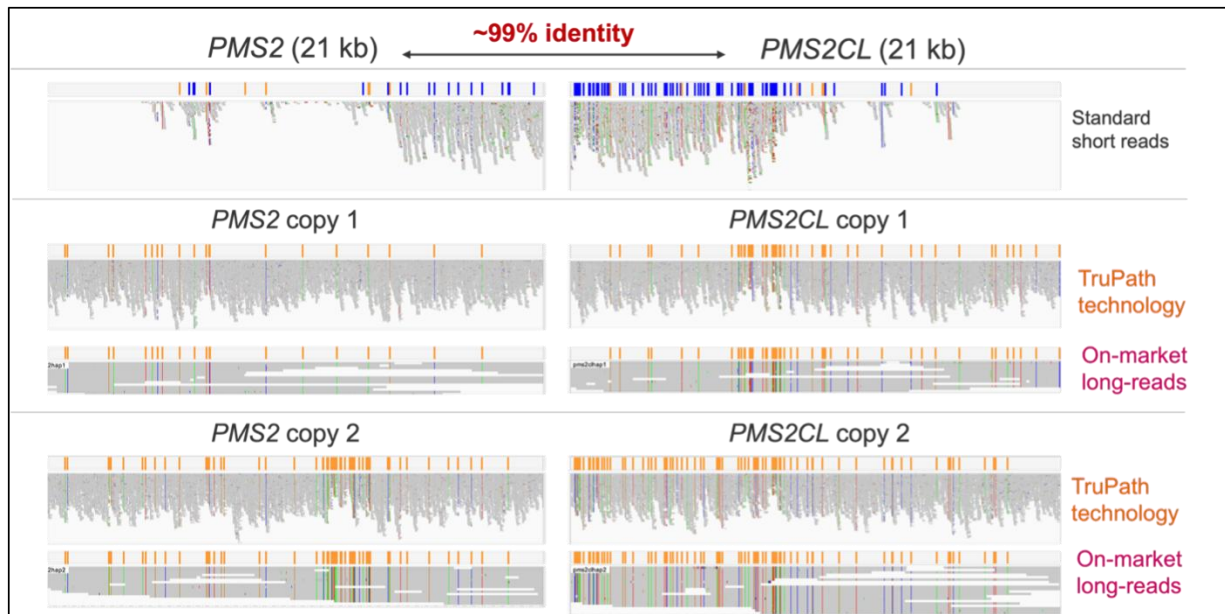


Figure 1. Example of MRJD usage of TruPath proximity information to generate haplotype-resolved variant calls, compared to SBS and on-market long read technologies

- **Improved STR Expansion Length and Classification Accuracy**

TruPath uses proximity information to recover fragments composed solely of STR sequence and employs a sequencing efficiency correction to account for uneven coverage in some repeat loci. With these improvements, TruPath can more accurately estimate STR expansion length and therefore

accurately classify expansions. Below is a comparison of STR expansion length estimation for standard Illumina sequencing and TruPath.

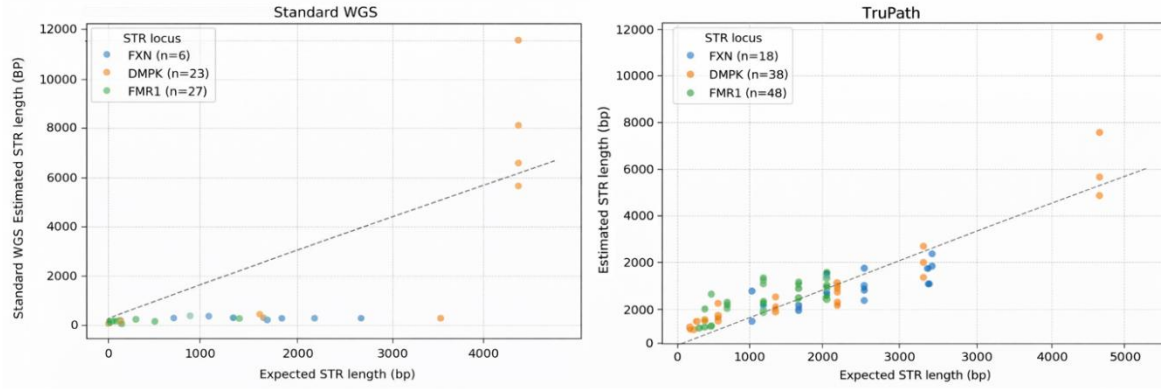


Figure 2. Improved STR expansion length and classification accuracy

- **Improved BND Filtering**

TruPath can use its novel colocation property to drastically reduce the number of interchromosomal and intrachromosomal large (>200 kbp) BND calls produced by DRAGEN-SV while maintaining its current recall.

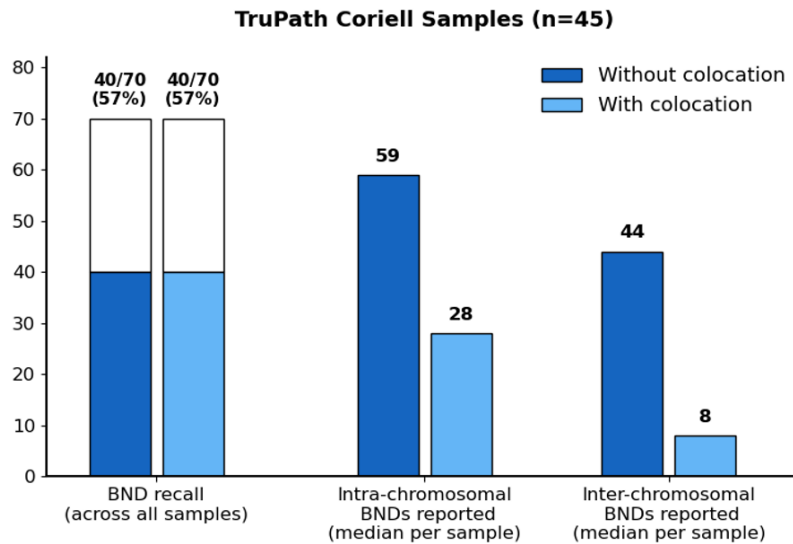


Figure 3. Improved BND filtering of structural variant calls

Usage

Below is the recommended dragen command line to use for TruPath analysis with v4.5.2

```

/opt/dragen/$VERSION/bin/dragen          #DRAGEN install path
--ref-dir $REF_DIR                       #path to DRAGEN pangenome hashtable
--output-directory $OUTPUT
--intermediate-results-dir $PATH
--output-file-prefix $PREFIX
# Inputs                                  # FASTQ-list or FASTQ pair
--fastq-list $PATH
--fastq-list-sample-id $STRING
# Illumina TruPath Genome
--enable-proximity true
# Mapper
--enable-map-align true
--enable-map-align-output true
--enable-sort true
--enable-duplicate-marking true
# Small variant caller
--enable-variant-caller true
# SV
--enable-sv true
# CNV
--enable-cnv true
--cnv-enable-self-normalization true
# Short tandem repeats
--repeat-genotype-enable true
# Multi-Region Joint Detection (MRJD)
--enable-mrjd true
--mrjd-enable-high-sensitivity-mode true

```

Proximity Linking Model

In the Illumina TruPath Genome, read pairs that are proximal on the flowcell have an increased likelihood of originating from the same template molecule, i.e., of being linked. To be able to quantify this proximity link likelihood, a probabilistic model relating genomic and flowcell proximity to the probability of two reads originating from the same input DNA molecule is required. DRAGEN includes such a proximity linking model. When DRAGEN map/align is run with `--enable-proximity=true`, the following steps are taken to estimate the parameters of the linking model and generate a proximity link probability distribution for each read group of a TruPath FASTQ input:

- Sample Collection
- Proximity Analysis
- Model Fitting
- Link Probability Distribution Generation

• Sample Collection

To fit the proximity model for a given read input, a representative subset of the input must be selected for the subsequent Link Analysis step. To this end, DRAGEN performs a preliminary mapping pass of the FASTQ input, generating groups of preliminary alignments one flowcell tile at a time. For each tile of alignments emitted during this preliminary mapping pass, proximal reads meeting a set of suitability requirements are gathered. These requirements are:

- MAPQ \geq 60
- Mapped
- Primary alignment
- Non-duplicate

- If paired-end: first in pair
- If paired-end: mapped mate
- If paired-end: properly-paired

DRAGEN continues to gather model input until 1 million preliminary alignments have been collected or the entire FASTQ input has been processed. If DRAGEN is unable to gather at least 1 million alignments, execution will still proceed to the following steps, but a warning is logged indicating a possible increased risk of failure due to an insufficient sample collection. If no suitable alignments were found at all, across the entire TruPath input, DRAGEN shall exit with an error.

- **Proximity Analysis**

Once a sufficiently large set of preliminary alignments has been gathered, an analysis is performed to discover pairs of reads that are spatially proximal on the flowcell as well as genomically proximal on the reference genome. Any read pairs that satisfy both proximity criteria have a high likelihood of originating from the same input DNA molecule. Every alignment has a spatial coordinate (X, Y) in addition to a mapped position on the reference genome. To find all pairs of likely-linked reads, the spatial and genomic displacements between all pairs of preliminary alignments are determined using these spatial and genomic coordinates.

A spatial flowcell displacement between two reads is represented as (XD, YD), where:

- XD (nm): x-component of flowcell displacement
- YD (nm): y-component of flowcell displacement

If the absolute spatial displacement between two reads is within X nm on the flowcell, the read pair meets the spatial proximity requirement. Likewise, a genomic displacement between two reads is represented as GDIST, where:

- GDIST (bp): (absolute) genomic distance, rounded to nearest 1000 bp

If the absolute genomic displacement between two reads is within Y bp, the read pair meets the genomic proximity requirement. Likely-linked read pairs whose displacements meet both thresholds can be represented using their displacements values (XD, YD, GDIST). The displacement values of all linked pairs are aggregated to give the counts N of read pairs with each XD, YD and GDIST. These counts of linked reads are used for the subsequent Model Fitting step.

As part of this analysis, a second set of counts is also tabulated using read pairs that are spatially proximal but further apart in the reference genome. These read pairs are considered spatially collocated by chance as opposed to being likely-linked reads. This set of counts is thus collected for the purpose of fitting the background/noise component of the model.

Before moving to Model Fitting, both sets of counts are analyzed to determine if the data exhibit trends that are consistent with TruPath data. If the input read data is deemed not to be TruPath data at this point, DRAGEN shall raise an error and exit.

- **Model Fitting**

The proximity linking model is non-linear and has ~ 20 parameters that allow N to be predicted from XD, YD and GDIST. The link counts from the link analysis are submitted to a non-linear least

squares solver to estimate the model parameters. If the solver fails to converge upon a solution for the model parameters, DRAGEN shall raise an error and exit.

Once fitted, the model enables the calculation of $\mu(XD, YD, GDIST)$, which is the prediction of the number of proximal reads with values XD, YD and GDIST. (This is much smoother than the empirical values counted in 'Link Analysis'.)

A simple background/noise model enables the calculation of $(\mu_{\text{chance}}(XD, YD, GDIST))$, which is the prediction of the number of proximal reads due to chance with values XD, YD and GDIST. The linking probability can be calculated as $1 - \frac{\mu_{\text{chance}}(XD, YD, GDIST)}{\mu(XD, YD, GDIST)}$. Typically, this is represented on a 'Phred'-scale: $-10\log_{10} \left(\frac{\mu_{\text{chance}}(XD, YD, GDIST)}{\mu(XD, YD, GDIST)} \right)$, where large (positive) numbers mean very-likely linked.

- **Link Probability Distribution Generation**

After the proximity linking model has been successfully fitted, a distribution of link probabilities across all practical spatial and genomic displacements (XD, YD, GDIST) is evaluated by the model and stored in a lookup table. This lookup table is continuously generated until link probability scores fall below a minimum threshold value. In rare cases where the fitted model fails to predict any meaningful link probability scores that surpass the minimum threshold, an empty lookup table will be generated. In such cases, DRAGEN shall raise an error and exit. This lookup table representing the link probability distribution for the provided TruPath read input is the primary output of the TruPath proximity linking model, allowing for downstream components in the DRAGEN Germline pipeline to leverage link probabilities.

Map/Align

The proximity linking model described above is used by the mapper to improve read mapping accuracy for a TruPath sample. In regions of high homology, standard Illumina sequencing reads will match the reference genome equally well, or nearly so, in multiple places. This would lead to mapping ambiguity, but in TruPath there may be a proximity-linked read-pair that maps uniquely to the genome enabling both read-pairs to be mapped uniquely.

A batch of read-pairs from a flow-cell region of interest is processed through the usual mapping process. Various alignment candidates are produced and scored and key information such as this score, genomic position, and flowcell position for each plausible candidate alignment is stored in an indexed data structure.

Each read pair X that might benefit from linking information is then revisited and the data structure is searched for other read pairs Y and alignment candidates that reveal the possibility that X and Y came from the same original template molecule. The proximity linking model quantifies the likelihood of this possibility. A Phred-scaled score derived from this likelihood can be added to the particular joint alignment hypothesis.

Template Tagging

During alignment, the mapper emits each read with link probability scores that estimate the likelihood of links existing between the current read and other nearby reads on the flowcell. Template tagging uses these link scores to reconstruct the original template DNA molecules from which the read pairs originated.

Template tagging begins by grouping reads into fragments, where each fragment consists of two reads (paired-end sequencing). For each fragment, outgoing link probability scores are collected from its constituent reads. Any links scoring below the minimum Phred-scaled quality threshold specified by `--proximity-min-linkq-threshold` (default: 10) are filtered out.

The remaining high-quality links are used to connect fragments into templates, where each connected set of fragments represents a reconstructed template molecule. All reads belonging to the same template receive a globally unique template identifier via the BAM tag `BX:Z`, allowing reads from the same original molecule to be identified by their shared tag value.

▪ Outputs

Template tagging generates several metrics reports detailing characteristics of all templates and links discovered during the DRAGEN run. Reports are generated for whole-genome ("wgs") and any specified QC regions. A template or link is included in QC region metrics if any part of its genomic span overlaps with the QC region.

• Template Metrics

The following template metrics reports are generated

Template Report	Output File	Description
Subpair Count	<code>_template_subpairs.csv</code>	Histogram of all discovered templates binned by the number of fragments (subpairs) they contain. A "subpair" refers to a read pair fragment within a template.
Genomic Distance	<code>_template_gdist.csv</code>	The distribution of template genomic lengths from 0-100th percentile. Template lengths are defined as the genomic distance between the smallest and largest mapped genomic positions represented in the template, <i>i.e.</i> , the span from the start of the first fragment to the end of the last fragment
Spatial Distance	<code>_template_xdist.csv</code> <code>_template_ydist.csv</code>	The distribution of template spatial lengths in X and Y flowcell units (FCU) from 0-100th percentile. Template spatial lengths are defined as the distance between the smallest and largest flowcell coordinates represented in the template along a given axis. The X distance report captures the spatial extent along the flowcell's X axis, while the Y distance report captures the extent along the flowcell's Y axis.
Length Thresholds	<code>_template_thresholds.csv</code>	Summarizes the count and percentage of all discovered templates above specified genomic length thresholds

• Template Subpair Count Report

- File `<prefix>.<qc-region>_template_subpairs.csv`
- Provides a histogram of all discovered templates binned by the number of fragments (subpairs) they contain. A "subpair" refers to a read pair fragment within a template.
- Subpair counts
 - Format
`TEMPLATE SUBPAIR COUNT,<prefix>,<count>,<value>,<percentage>`

Where:

- <prefix>: Output file prefix
- <count>: Number of fragments in the template
- <value>: Number of templates with this fragment count
- <percentage>: Percentage of all templates with this fragment count

- o Summary statistics
 - Mean and percentile distributions

- o Example

```
TEMPLATE SUBPAIR COUNT,example,2,524,42.02
TEMPLATE SUBPAIR COUNT,example,3,294,23.58
TEMPLATE SUBPAIR COUNT,example,4,172,13.79
...
TEMPLATE SUBPAIR COUNT SUMMARY,example,MEAN SUBPAIR COUNT,3.34
TEMPLATE SUBPAIR COUNT SUMMARY,example,5TH PERCENTILE SUBPAIR COUNT,2.00
TEMPLATE SUBPAIR COUNT SUMMARY,example,25TH PERCENTILE SUBPAIR COUNT,2.00
TEMPLATE SUBPAIR COUNT SUMMARY,example,50TH PERCENTILE SUBPAIR COUNT,3.00
TEMPLATE SUBPAIR COUNT SUMMARY,example,75TH PERCENTILE SUBPAIR COUNT,4.00
TEMPLATE SUBPAIR COUNT SUMMARY,example,95TH PERCENTILE SUBPAIR COUNT,7.00
```

- *Template Genomic Distance Report*

- o File <prefix>.<qc-region>_template_gdist.csv
- o Details the distribution of template genomic lengths from 0-100th percentile. Template lengths are defined as the genomic distance between the smallest and largest mapped genomic positions represented in the template, *i.e.*, the span from the start of the first fragment to the end of the last fragment.
- o Format
PROXIMITY TEMPLATE GENOMIC DISTANCE DETAILS,<prefix>,<percentile>,<value>

Where:

- <prefix>: Output file prefix
- <percentile>: Percentile rank (0-100)
- <value>: Template genomic length in base pairs at this percentile

- o Example

```
PROXIMITY TEMPLATE GENOMIC DISTANCE DETAILS,example,0,189.00
PROXIMITY TEMPLATE GENOMIC DISTANCE DETAILS,example,1,1476.00
PROXIMITY TEMPLATE GENOMIC DISTANCE DETAILS,example,2,1583.00
...
PROXIMITY TEMPLATE GENOMIC DISTANCE DETAILS,example,98,52699.00
PROXIMITY TEMPLATE GENOMIC DISTANCE DETAILS,example,99,65494.59
PROXIMITY TEMPLATE GENOMIC DISTANCE DETAILS,example,100,257772.00
```

- o **Note:** Percentile values are interpolated from the distribution of all discovered template lengths, which can result in non-integer base pair values

- *Template Spatial Distance Reports*

- o File <prefix>.<qc-region>_template_xdist.csv and <prefix>.<qc-region>_template_ydist.csv
- o Details the distribution of template spatial lengths in X and Y flowcell units (FCU) from 0-100th percentile. Template spatial lengths are defined as the distance between the smallest and largest flowcell coordinates represented in the template along a given axis. The X distance report captures the spatial extent along the flowcell's X axis, while the Y distance report captures the extent along the flowcell's Y axis.
- o Format
PROXIMITY TEMPLATE n DISTANCE DETAILS,<prefix>,<percentile>,<value>

Where:

<prefix>: Output file prefix

<percentile>: Percentile rank (0-100)

<value>: Template spatial length in flowcell units (FCU) at this percentile

- o Example

```
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 0, 0.00
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 1, 0.00
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 2, 9.00
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 3, 9.00
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 4, 9.00
```

...

```
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 96, 176.00
```

```
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 97, 194.00
```

```
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 98, 213.00
```

```
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 99, 249.00
```

```
PROXIMITY TEMPLATE X DISTANCE DETAILS, example, 100, 4077.00
```

- o **Note:** Percentile values are interpolated from the distribution of all discovered template lengths, which can result in non-integer flowcell unit values

- *Template Length Thresholds Report*

- o File <prefix>.<qc-region>_template_thresholds.csv
- o Summarizes the count and percentage of all discovered templates above specified genomic length thresholds.

- o Format

```
threshold, count_above_threshold, percentage_above_threshold
```

Where:

<threshold>: Genomic length threshold in base pairs

<count_above_threshold>: Number of templates with genomic length greater than or equal to threshold

<percentage_above_threshold>: Proportion of templates above threshold (0.0 to 1.0)

- o Example

```
threshold, count_above_threshold, percentage_above_threshold
```

```
10000, 1031, 0.83
```

```
20000, 879, 0.7
```

```
60000, 395, 0.32
```

- o The thresholds in this report are defined by `--template-gdist-thresholds` (default: 10000, 20000, 60000)

- **Link Metrics**

Link metrics are generated for each Phred-scaled link quality threshold specified via the following command-line options:

- `--proximity-min-linkq-threshold` (default: 10): The primary link quality threshold used to accept or reject a link hypothesis during template tagging.
- `--proximity-additional-linkq-thresholds` (default: 25, maximum 2 values): Additional link quality thresholds used for deriving the following metrics reports at higher link quality cutoffs.

The following link metrics reports are generated

Link Report	Output File	Description
Genomic Distance	_proximity_gdist.csv	Details the distribution of link genomic lengths from 0-100th percentile for links that meet or exceed each specified link quality threshold. Link lengths are defined as the genomic distance between the two fragments connected by the link.
Spatial Distance	_proximity_xdist.csv _proximity_ydist.csv	Detail the distribution of link spatial lengths in X and Y flowcell units (FCU) from 0-100th percentile for links that meet or exceed each specified link quality threshold. Link spatial lengths are defined as the distance between the two flowcell coordinates of the fragments connected by the link along a given axis. The X distance report captures the spatial extent along the flowcell's X axis, while the Y distance report captures the extent along the flowcell's Y axis

- *Link Genomic Distance Report*

- File <prefix>.<qc-region>_proximity_gdist.csv
- Details the distribution of link genomic lengths from 0-100th percentile for links that meet or exceed each specified link quality threshold. Link lengths are defined as the genomic distance between the two fragments connected by the link.
- Format
PROXIMITY LINK GENOMIC DISTANCE DETAILS,<linkq-threshold>,<percentile>,<value>

Where:

<linkq-threshold>: Phred-scaled link quality threshold (0-63)
 <percentile>: Percentile rank (0-100)
 <value>: Link genomic length in base pairs at this percentile

- Example

```
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,0,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,1,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,2,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,3,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,4,517.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,5,1579.00
...
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,98,222531.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,99,239261.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,10,100,264674.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,0,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,1,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,2,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,3,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,4,0.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,5,906.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,6,1601.00
...
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,98,102292.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,99,107208.00
PROXIMITY LINK GENOMIC DISTANCE DETAILS,25,100,122872.00
```

- **Note:** Percentile values are interpolated from the distribution of all discovered link lengths, which can result in non-integer base pair values.
- *Link Spatial Distance Reports*
 - File `<prefix>.<qc-region>_proximity_xdist.csv` and `<prefix>.<qc-region>_proximity_ydist.csv`
 - Detail the distribution of link spatial lengths in X and Y flowcell units (FCU) from 0-100th percentile for links that meet or exceed each specified link quality threshold. Link spatial lengths are defined as the distance between the two flowcell coordinates of the fragments connected by the link along a given axis. The X distance report captures the spatial extent along the flowcell's X axis, while the Y distance report captures the extent along the flowcell's Y axis.
 - Format


```
PROXIMITY LINK n DISTANCE DETAILS,<linkq-threshold>,<percentile>,<value>
```

Where:

 - `<linkq-threshold>`: Phred-scaled link quality threshold (0-63)
 - `<percentile>`: Percentile rank (0-100)
 - `<value>`: Link spatial length in flowcell units (FCU) at this percentile
 - Example


```
PROXIMITY LINK X DISTANCE DETAILS,10,0,0.00
PROXIMITY LINK X DISTANCE DETAILS,10,1,0.00
PROXIMITY LINK X DISTANCE DETAILS,10,2,0.00
PROXIMITY LINK X DISTANCE DETAILS,10,3,0.00
PROXIMITY LINK X DISTANCE DETAILS,10,4,0.00
PROXIMITY LINK X DISTANCE DETAILS,10,5,0.00
PROXIMITY LINK X DISTANCE DETAILS,10,6,9.00
PROXIMITY LINK X DISTANCE DETAILS,10,7,9.00
...
PROXIMITY LINK X DISTANCE DETAILS,10,98,148.00
PROXIMITY LINK X DISTANCE DETAILS,10,99,167.00
PROXIMITY LINK X DISTANCE DETAILS,10,100,222.00
PROXIMITY LINK X DISTANCE DETAILS,25,0,0.00
PROXIMITY LINK X DISTANCE DETAILS,25,1,0.00
PROXIMITY LINK X DISTANCE DETAILS,25,2,0.00
PROXIMITY LINK X DISTANCE DETAILS,25,3,0.00
PROXIMITY LINK X DISTANCE DETAILS,25,4,0.00
PROXIMITY LINK X DISTANCE DETAILS,25,5,0.00
PROXIMITY LINK X DISTANCE DETAILS,25,6,0.00
PROXIMITY LINK X DISTANCE DETAILS,25,7,9.00
PROXIMITY LINK X DISTANCE DETAILS,25,8,9.00
...
PROXIMITY LINK X DISTANCE DETAILS,25,98,84.00
PROXIMITY LINK X DISTANCE DETAILS,25,99,93.00
PROXIMITY LINK X DISTANCE DETAILS,25,100,111.00
```
 - **Note:** Percentile values are interpolated from the distribution of all discovered link lengths, which can result in non-integer flowcell unit values.

Phasing

When working with TruPath data, DRAGEN can phase reads upstream of variant calling and then use the haplotype-phased reads to make phased variant calls. Read phasing is performed as an extension of DRAGEN's personalization feature, whereby the sample's ancestral haplotypes are inferred and, simultaneously, reads are phased to the inferred haplotypes. Using phased reads during variant calling then yields improved variant calling accuracy and allows distant variants to be phased with each other.

DRAGEN read phasing combines information from the reads' alignment scores for all of the haplotypes in the hash-table database to phase reads and impute the sample's ancestral haplotypes. Phasing is enhanced by the long-range proximity-linking information provided by the TruPath library preparation. As in the regular personalization workflow, DRAGEN will also impute variants for the sample from the variants present in the haplotype database.

To understand the read phasing output, it is useful to understand how DRAGEN phases reads:

- The genome is divided into small, contiguous bins (typically 4096 bases long).
- Haplotypes are imputed and reads are phased within each bin using the haplotype database in the reference hash table.
- The proximity-linking information is used to transfer phasing information between reads in different bins.
- For each bin:
 - The phased reads are tagged with the haplotype information in the output BAM file,
 - The best haplotype pair is output in a BED file,
 - The variants are imputed and saved in a VCF file,
 - Bins are grouped in contiguous, non-overlapping phase blocks when there is strong evidence of co-phasing.

The bin is the minimal unit of phasing in DRAGEN. Each bin is phased in the context of the ancestral haplotypes inferred in neighboring bins, and the phasing of linked reads in other bins.

• Phasing Options

Phasing is enabled by default when proximity mode is enabled via `--enable-proximity=true`, no other explicit options are required. Default settings are recommended but users can adjust the parameters controlling the phasing behavior using the following options:

Command Line Argument	Description	Value
<code>--personalization-phase-block-threshold</code>	Threshold used by the read phasing algorithm to group personalization bins into phase blocks (Default=20)	[0.0, inf)
<code>--read-phasing-gene-list</code>	Optional GTF file that, if specified, will be used to calculate a metric that counts genes that are fully contained within phase blocks	

Lowering the phase-block threshold parameter will reduce the amount of co-phasing evidence required to group adjacent personalization bins into a single-phase block, and vice versa.

• Phasing Output Files

○ BAM/CRAM Output

The phased reads in the map/align output file are annotated with the following tags:

Tag	Description	Values
pp	Phasing probability in Phred-scale log odds: $10 * \log_{10} (P(H_1)/P(H_2))$	[-127,127]
HP	Haplotype tag for all reads where $ pp \geq 10$	1,2
PS	Phase block tag	[0, 2 ³²)

- **Personalized Haplotypes**

The personalized haplotypes for each phased bin are output in tab-delimited format (TSV). A summary of the phase blocks defined in the TSV file is also written in GTF format.

TSV (<sample_id>.personal_haplotypes.tsv.gz)

The personalized haplotypes TSV file contains the following columns:

Column	Description
CHROM	Chromosome name
START	Start position of the phased bin (0-based)
END	End position of the phased bin (1-based)
PHASE_BLOCK	Phase block ID for the bin, bin with the same ID are confidently co-phased
PHASING_CONFIDENCE	Phasing confidence for the bin: the lower the confidence, the higher the chance of haplotype switching

GTF (<sample_id>.phase_blocks.gtf.gz)

The regions covered by the phase blocks, as defined in the personalized TSV file's PHASE_BLOCK column, are output in a GTF file with the following fields:

Column	Description
seqname	Chromosome name
source	Always 'dragen'
feature	Always 'phaseblock'
start	Start position of the phase block (1-based)
end	End position of the phase block (1-based)
score	Unused ('.')
strand	Unused ('.')
frame	Unused ('.')
attribute	Always 'phase_block n'

- **Imputed Variants**

The imputed variants for each phased bin are output in a VCF file. It is important to note that the imputed VCF:

- Contains only variants imputed from the haplotype database in the reference hash table,
- Does not contain any novel variants from the sample,
- Splits multiallelic variants into multiple records, one for each alternate allele.

VCF (<sample_id>.personal.vcf.gz)

The VCF follows the 4.2 standard, below is the description of relevant fields:

Tag	Description
QUAL	Phred-scale score for the marginal probability of ALT. For example, for a diploid variant: $-10 * \log_{10}(P(GT='0 0'))$
INFO:HAPS	Two best haplotype pairs for the bin the variant belongs to
INFO:PGP	Marginal probability for $P(GT='0 0'), P(GT='1 0') + P(GT='0 1'), P(GT='1 1')$
FORMAT:PS	Phase block ID for the bin the variant belongs to

○ Phasing Metrics

DRAGEN reports a set of standard phasing metrics for each TruPath analysis, and outputs these metrics to a phasing summary statistics CSV file.

CSV (<sample_id>.phasing_summary_stats.csv)

Metric	Description
Phasing chromosomes	A list of the chromosomes used to calculate the metrics. Only autosomes with phased reads are considered.
N50	The length of the shortest phase block where all phase blocks of at least that length account for $\geq 50\%$ of the cumulative phase block length.
L50	The smallest number of phase blocks that account for 50% of the cumulative phase block length.
NG50	The length of the shortest phase block where all phase blocks of at least that length account for $\geq 50\%$ of the cumulative length of the phasing chromosome set.
LG50	The smallest number of phase blocks that account for 50% of the cumulative length of the phasing chromosome set.
Total phase block length for L50/N50	The cumulative length of the phase-block assembly.
Total phase block length for LG50/NG50	The cumulative length of the chromosome set.
Number of fully phased 300 kbp windows	After partitioning each chromosome into 300 kbp windows, the number of such windows that are each fully contained within a single-phase block.
Number of fully phased genes	The number of genes that are each fully contained within a single-phase block.

Gene list	The filename of the gene list used to calculate the number of fully phased genes
-----------	--

Note that the two gene-related metrics are only computed if a gene list is specified using the `-read-phasing-gene-list` command-line argument.

Structural Variant Calling

TruPath-specific structural variant (SV) calling is only supported in a single-sample whole-genome germline SV discovery scenario. The DRAGEN SV caller leverages proximity information indirectly through phasing information in the reads. This approach provides several key advantages:

- **Phased assemblies** — Each haplotype in a candidate region is assembled separately, reducing the complexity of the assembly graph and producing higher-quality assembled contigs.
 - **Haplotype-aware ML features** — Some features are segregated by haplotype, enabling better training and inference for the ML model.
 - **Haplotype-resolved genotyping** — Heterozygous SVs can be distinguished and assigned to specific local haplotypes.
- **Leveraging TruPath proximity-linked features**

The DRAGEN SV caller currently leverages proximity information indirectly by utilizing the phasing information in the reads during candidate assembly as well as ML filtering. It is highly recommended to keep the ML filtering enabled for best accuracy results.

- **Phased Assemblies**

Reads collected for assembling a candidate are segregated into two haplotypes using the phasing information available in the reads, and each set is assembled separately, leading to at most one contig each. Two contigs per candidate are processed in the rest of the downstream pipeline.

- **ML Processing**

DRAGEN-SV with TruPath data uses an ML model that has been trained on TruPath leveraging features dependent on phasing information in the reads, in addition to features being used in DRAGEN-SV with Illumina standard sequencing. Enabling ML is crucial for obtaining the best accuracy.

- **Collapsing, Deduplication, Regenotyping**

Structural variants of certain types (only insertions, deletions, duplications) coming from phased rounds of assemblies are collapsed and deduplicated if they are deduced to be representing the same event before outputting them to the VCF file. The type of SV, its length, its location, its genotype scores, and the haplotype of the reads used for assembly are used to decide whether two SVs represent the same event.

The genotype (INFO field GT) of a HET SV is changed to 1/0 if the SV was yielded by assembly of reads phased to be the first haplotype. Similarly, if two SVs being collapsed are from different haplotypes' reads, the resulting SV will be re-genotyped as 1/1.

- **SV VCF Outputs**

The following VCF fields are added for TruPath

Field Type	ID	Description
INFO	PHASEDASM	Haplotype of the reads used for the assembly yielding the SV (only with <code>--enable-proximity=true</code>)
	ML_UPDATED	The FILTER status has changed from PASS to non-PASS or non-PASS to PASS after QUAL being recalibrated by ML
FORMAT	MLQS	ML recalibrated QUAL for indels
FILTER	MLFail	Record Level Filter Prob(TP) is less than SV_ML_MIN_PASS_DEL_PROB for deletions or Prob(TP) is less than SV_ML_MIN_PASS_INS_PROB

Multi-Region Joint Detection

DRAGEN Multi-Region Joint Detection (MRJD) is a *de novo* germline small variant caller for paralogous regions. When combined with TruPath data, MRJD produces haplotype-resolved small variant calls in paralogous regions by exploiting long-range information enabled by the proximity properties of TruPath. This variant calling process does not rely on known population haplotypes.

Currently, MRJD with TruPath data covers nine sets of paralogous regions that include 15 clinically relevant genes. Table 1 below lists the hg38 region coordinates covered by MRJD (note that MRJD currently is **only compatible with the hg38 reference genome**).

Table 1. Paralogous regions covered by MRJD

Chromosome	Start	End	Region name	Paralog set name	Paralog type
chr1	196786972	196827189	CFHR3-CFHR1	CFHR3-CFHR1-CFHR4-CFHR2	Non-tandem
chr1	196911497	196951222	CFHR4-CFHR2	CFHR3-CFHR1-CFHR4-CFHR2	Non-tandem
chr5	70924941	70966375	SMN1	SMN1-SMN2	Non-tandem
chr5	70049523	70090528	SMN2	SMN1-SMN2	Non-tandem
chr6	32037415	32045473	CYP21A2-TNXB	CYP21A2	Tandem
chr6	32004679	32012619	CYP21A1P-TNXA	CYP21A2	Tandem
chr7	5969485	5987844	PMS2	PMS2-PMS2CL	Non-tandem
chr7	6736851	6755308	PMS2CL	PMS2-PMS2CL	Non-tandem
chr7	74771000	74791999	NCF1	NCF1-NCF1B-NCF1C	Non-tandem
chr7	73217606	73238630	NCF1B	NCF1-NCF1B-NCF1C	Non-tandem
chr7	75153934	75174978	NCF1C	NCF1-NCF1B-NCF1C	Non-tandem
chr8	142873164	142879856	CYP11B1	CYP11B1-CYP11B2	Tandem

chr8	142910764	142917883	CYP11B2	CYP11B1-CYP11B2	Tandem
chr15	43599563	43618800	STRC	STRC-STRCP1	Tandem
chr15	43699418	43718260	STRCP1	STRC-STRCP1	Tandem
chr22	18159724	18174315	USP18	USP18-USP41P	Non-tandem
chr22	20362649	20377695	USP41P	USP18-USP41P	Non-tandem
chr22	42123192	42132193	CYP2D6	CYP2D6-CYP2D7	Tandem
chr22	42135344	42145873	CYP2D7	CYP2D6-CYP2D7	Tandem

• **MRJD Method**

MRJD begins by collecting all primary alignments within the paralogous regions of interest, regardless of mapping quality. For each set of paralogous regions (such as *SMN1* and *SMN2*), MRJD estimates the total copy number by leveraging read depth in all the regions of interest and pre-selected stable regions across the genome. Using the estimated total copy number, read sequences, and proximity information, MRJD then constructs all copies within the paralogous region set.

For non-tandem paralog sets (see Table 1), MRJD uses proximity information to assign each constructed copy to the genomic location where it most likely originated (e.g., *PMS2* vs. *PMS2CL*). For tandem paralog sets, MRJD uses proximity information to assign each constructed copy to either the maternal or paternal haplotype. Finally, MRJD calls small variants based on the constructed copies and reports the variant calls together with their assigned genomic locations or haplotypes.

See workflow diagram below for an overview of the MRJD method.

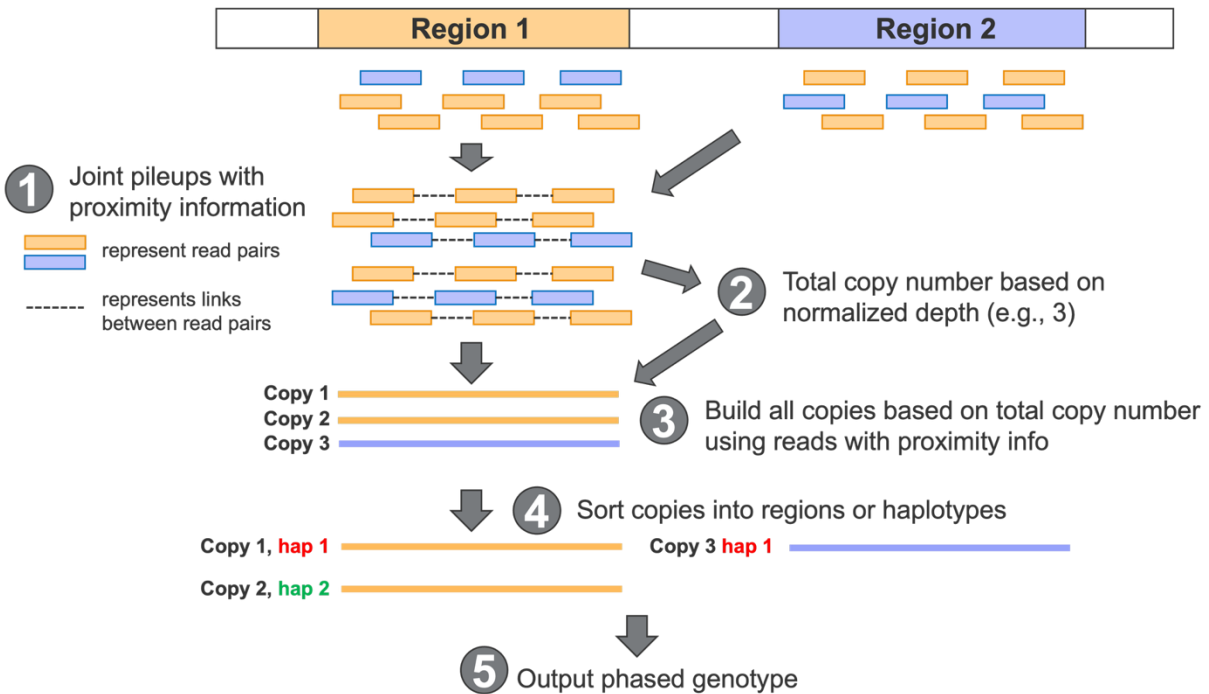


Figure 4. MRJD workflow using TruPath data

- MRJD Outputs

Once DRAGEN has completed, the following files will be produced in the output directory:

Output File(s)	Description
<prefix>.mrjd.hard-filtered.vcf.gz	VCF file containing small variants called by MRJD in paralogous regions
<prefix>.mrjd.json	JSON file for MRJD containing copy number estimates, region/haplotype assignments, and run status for each paralogous region
<prefix>.mrjd.phased.bam	MRJD BAM file containing phased alignments within paralogous regions
mrjd_supporting_files/	Directory containing supporting files for MRJD that could be useful for downstream analysis and visualization. The files included are below
mrjd_supporting_files/ <prefix>.mrjd.<paralog_name>.vcf.gz	VCF file containing small variants called by MRJD for each paralogous region in multi-column format (one column for each copy). One file is generated for each set of paralogous regions
mrjd_supporting_files/ <prefix>.mrjd. reference_region_alignments.sam	SAM file containing reference region alignments for MRJD

- MRJD VCF output

The MRJD caller generates a <prefix>.mrjd.hard-filtered.vcf.gz file in the output directory. The output file is a gzip-compressed VCFv4.2 formatted file that contains the VCF representation of the small variants from the identified genotype. Importantly, for a given set of paralogous regions, all copies will be reported under all regions, each copy will be annotated with its assigned genomic region or haplotype in the `REGION_PLACEMENT` or `PSL` field in the `FORMAT` column, respectively.

Here is an example of the VCF records called by MRJD in the non-tandem paralogous region. The `REGION_PLACEMENT` field in the `FORMAT` column indicates the genomic region assignment for each copy reported in the genotype (following the order in the genotype field), with `I` indicating assignment to the current region, `A` indicating assignment to other or alternate regions, and `U` indicating unplaced assignment.

Chr	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	<prefix>
chr5	70052190	.	C	CA	500	.	regionGroupName=SMN1-SMN2;REF_DIFF_SITE	GT:REGION_PLACEMENT:RPQL:PQ:JAD:JAF:JDP:PS	1 0 0 0:A,A,I,I:..500:90,30:0.250:120:70052190
chr5	70052613	.	T	C	500	.	regionGroupName=SMN1-SMN2	GT:REGION_PLACEMENT:RPQL:PQ:JAD:JAF:JDP:PS	1 0 0 0:A,A,I,I:..500:86,35:0.289:121:70052190
chr5	70052881	.	C	CAAAAA	500	.	regionGroupName=SMN1-SMN2;REF_DIFF_SITE	GT:REGION_PLACEMENT:RPQL:PQ:JAD:JAF:JDP:PS	1 0 0 0:A,A,I,I:..500:93,28:0.231:121:70052190
chr5	70053733	.	TC	T	500	.	regionGroupName=SMN1-SMN2	GT:REGION_PLACEMENT:RPQL:PQ:JAD:JAF:JDP:PS	0 1 0 0:A,A,I,I:..500:85,32:0.274:117:70052190
chr5	70053985	.	CT	C	500	.	regionGroupName=SMN1-SMN2	GT:REGION_PLACEMENT:RPQL:PQ:JAD:JAF:JDP:PS	0 1 0 1:A,A,I,I:..500:67,65:0.492:132:70052190
chr5	70054456	.	TA	T	500	.	regionGroupName=SMN1-SMN2	GT:REGION_PLACEMENT:RPQL:PQ:JAD:JAF:JDP:PS	0 1 1 1:A,A,I,I:..500:22,105:0.827:127:70052190

Here is an example of the VCF records called by MRJD in the tandem paralogous region. The `PSL` field in the `FORMAT` column indicates the haplotype assignment for each copy reported in the genotype (following the order in the genotype field), with `hap1` indicating assignment to the first haplotype and `hap2` indicating assignment to the second haplotype. Note that

the `REGION_PLACEMENT` field is not applicable for tandem paralogous regions and is thus filled with `U` (unplaced) for all copies.

Chr	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	<prefix>
chr6	32004754	.	T	C	63.01	.	regionGroupName=CYP21A2;REF_DIFF_SITE	GT:PSL:REGION_PLACEMENT:AGQL:PQ:3AD:JAF:JDP:PS	1 0 1 0:copy1_hap1,copy2_hap1,copy3_hap2,copy4_hap2:U,U,U,U:0.78:1:57,54:0.486:111:32004754
chr6	32004791	.	G	A	63.01	.	regionGroupName=CYP21A2;REF_DIFF_SITE	GT:PSL:REGION_PLACEMENT:AGQL:PQ:3AD:JAF:JDP:PS	1 0 1 0:copy1_hap1,copy2_hap1,copy3_hap2,copy4_hap2:U,U,U,U:0.78:1:62,56:0.475:118:32004754
chr6	32004857	.	C	T	63.01	.	regionGroupName=CYP21A2;REF_DIFF_SITE	GT:PSL:REGION_PLACEMENT:AGQL:PQ:3AD:JAF:JDP:PS	1 0 1 0:copy1_hap1,copy2_hap1,copy3_hap2,copy4_hap2:U,U,U,U:0.78:1:51,53:0.510:104:32004754
chr6	32004862	.	C	T	63.01	.	regionGroupName=CYP21A2;REF_DIFF_SITE	GT:PSL:REGION_PLACEMENT:AGQL:PQ:3AD:JAF:JDP:PS	1 0 1 0:copy1_hap1,copy2_hap1,copy3_hap2,copy4_hap2:U,U,U,U:0.78:1:48,55:0.534:103:32004754
chr6	32004868	.	G	A	63.01	.	regionGroupName=CYP21A2;REF_DIFF_SITE	GT:PSL:REGION_PLACEMENT:AGQL:PQ:3AD:JAF:JDP:PS	1 0 1 0:copy1_hap1,copy2_hap1,copy3_hap2,copy4_hap2:U,U,U,U:0.78:1:49,55:0.529:104:32004754
chr6	32005002	.	G	A	63.01	.	regionGroupName=CYP21A2	GT:PSL:REGION_PLACEMENT:AGQL:PQ:3AD:JAF:JDP:PS	1 0 0 0:copy1_hap1,copy2_hap1,copy3_hap2,copy4_hap2:U,U,U,U:0.78:1:102,30:0.227:132:32004754

• MRJD JSON output

The MRJD caller generates a `<prefix>.mrjd.json` file in the output directory. This JSON-formatted file contains detailed information for each paralogous region, including copy number estimates, genomic region assignment for each copy, and haplotype assignment for each copy.

The total copy number of the paralogous set is reported under `jointCopyNumber`. The `mrjdRunStatus` indicates whether MRJD ran successfully for the region, with `Success` indicating a successful run, and `Aborted` indicating a failure.

Here is an example of the JSON output for a non-tandem paralogous region. For each copy reported in the VCF file (following the order in the genotype field), the placement information under `regionPlacement` indicates which genomic region the copy is assigned to.

```
{
  "regionGroupName": "SMN1-SMN2",
  "region1Coord": "chr5:70924941-70965975",
  "region1Name": "SMN1",
  "region2Coord": "chr5:70049523-70090528",
  "region2Name": "SMN2",
  "jointCopyNumber": "4",
  "jointCopyNumberFloat": "3.972865",
  "regionPlacement": {
    "SMN1": [
      "copy1",
      "copy2"
    ],
    "SMN2": [
      "copy3",
      "copy4"
    ]
  },
  "mrjdRunStatus": "Success"
}
```

Here is an example of the JSON output for a tandem paralogous region. For each copy reported in the VCF file (following the order in the genotype field), the haplotype information under `locusStructure` indicates which haplotype the copy is assigned to. Note that all copies are listed under unplaced in `regionPlacement` since tandem copies cannot be assigned to specific genomic locations.

```
{
  "regionGroupName": "CYP21A2",
  "region1Coord": "chr6:32037415-32045473",
  "region1Name": "CYP21A2-TNXB",
```

```
"region2Coord": "chr6:32004679-32012619",
"region2Name": "CYP21A1P-TNXA",
"jointCopyNumber": "4",
"jointCopyNumberFloat": "3.892923",
"locusStructure": {
  "hap1": [
    [
      "copy1"
    ],
    [
      "copy2"
    ]
  ],
  "hap2": [
    [
      "copy3"
    ],
    [
      "copy4"
    ]
  ]
},
"regionPlacement": {
  "unplaced": [
    "copy1",
    "copy2",
    "copy3",
    "copy4"
  ]
}
},
"mrjdRunStatus": "Success"
}
```

- **MRJD phased BAM output**

The MRJD caller generates a `<prefix>.mrjd.phased.bam` file in the output directory. The output file is a BAM formatted file that contains the phased alignments within paralogous regions. Same as the MRJD VCF file, for a given set of paralogous regions, all copies will be reported under all regions.

The following tags are added to the BAM records in the phased BAM file:

- **HP:** The copy label assigned to the read. For non-tandem paralogs, the copy labels correspond to the genomic regions (e.g., `copy1_SMN1`, `copy2_SMN2`, etc.). For tandem paralogs, the copy labels correspond to the haplotypes (e.g., `copy1_hap1`, `copy2_hap1`, etc.).
- **PC:** Confidence score (Phred-scaled) of the read-to-copy assignment.
- **PS:** Phasing set identifier.
- **BX:** Template ID based on proximity linking information. Fragments with the same **BX** tag are likely to originate from the same original DNA molecule.

Note that the output format can be either BAM, CRAM or SAM, depending on the `--output-format` option specified in the DRAGEN run.

- **MRJD supporting files**

The MRJD caller generates a `mrjd_supporting_files/` directory in the output directory. This directory contains supporting files for MRJD:

- `<prefix>.mrjd.<paralog_name>.vcf.gz`: VCF file containing small variants called by MRJD for each paralogous region in multi-column format (one column per copy). This file is useful for visualizing haplotype-resolved variants in genome browsers (such as IGV) that support multi-column VCF format.
- `<prefix>.mrjd.reference_region_alignments.sam`: SAM file containing reference region alignments for MRJD. This file helps interpret the reference sequence differences between paralogous regions and aids in understanding variant calls (e.g., identifying gene conversion events).

- **Visualize MRJD results in IGV**

Here is an example of inspecting MRJD results in *SMN1*-*SMN2* regions by loading the multi-column VCF file (`mrjd_supporting_files/<prefix>.mrjd.SMN1-SMN2.vcf.gz`), phased BAM file (`<prefix>.mrjd.phased.bam`), and reference region alignments SAM file (`<prefix>.mrjd.reference_region_alignments.sam`) in IGV.

- All *SMN1* and *SMN2* copies are reported under the *SMN1* region (and also under the *SMN2* region) in the multi-column VCF file. The copy-to-region assignment is indicated in the "sample" column. In the IGV screenshot below, copy1 and copy2, and copy3 are assigned to the *SMN1* region, while copy4 is assigned to the *SMN2* region.
- The phased BAM file shows reads assigned to each copy (load the BAM and select "Group alignments by" - "phase")
- The reference region alignments SAM file highlights the reference sequence differences between *SMN1* and *SMN2*.

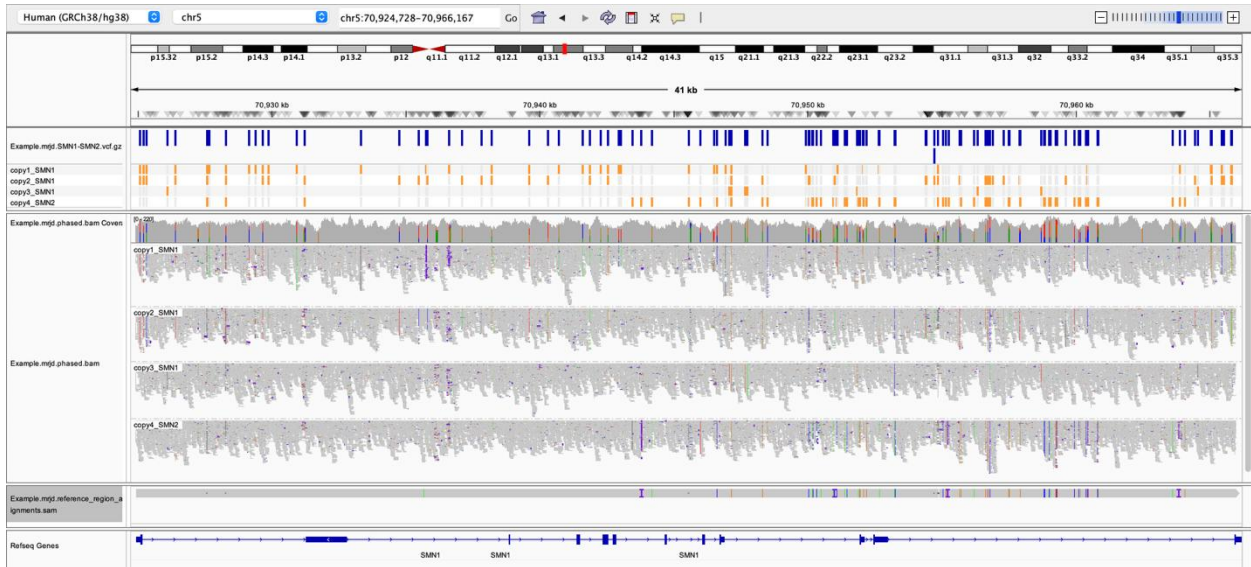


Figure 5. Example of visualizing MRJD results in IGV

- **Visualize MRJD results in DRAGEN Reports**

MRJD results are also integrated into DRAGEN Reports, specifically, the MRJD results for sample-level reports will be available under the "Paralogs" tab. The "Paralog Sets" table provides an overview of each paralogous region analyzed, including the estimated total copy number. The "Paralogous regions" view provides visualization of the haplotype-resolved variant calls within each paralogous region. The following example shows the MRJD phased variant calls for *PMS2-PMS2CL*.

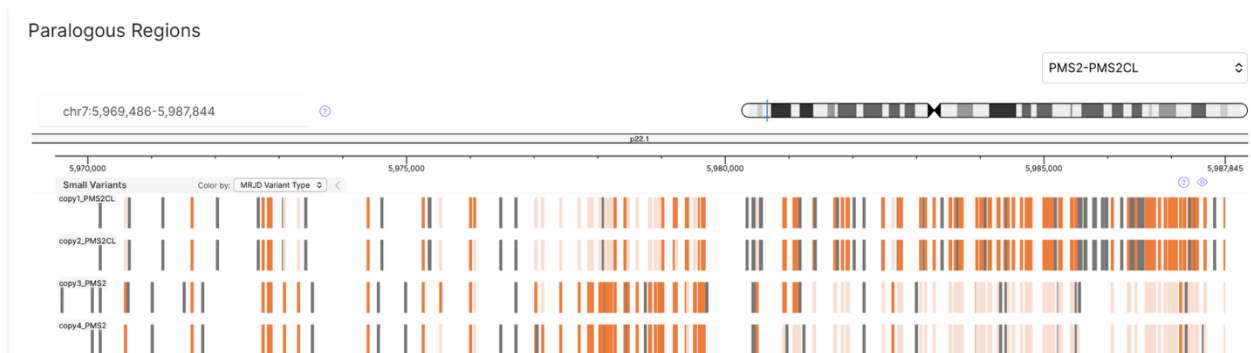


Figure 6. Example of visualizing MRJD results in DRAGEN Reports. Here the dark orange represents alternative allele for a reference difference site between the paralogous regions, the light orange represents reference allele for a reference difference site, the grey represents non-reference difference site variant

- **MRJD Notes**

- MRJD currently only supports paralogous region calling when the total copy number is less than 8. Regions with higher copy numbers will be skipped, and no variants will be called. However, the total copy number estimation will still be reported in the JSON output.
- MRJD currently only supports the hg38 reference genome.

- MRJD currently only supports variant calling when the sample average linked coverage (without duplicates) is $\geq 16X$.
- MRJD currently only supports small variant calling.

STR Calling

TruPath data improves mapping accuracy for long STRs by leveraging proximity linking information to map repetitive read pairs (In Repeat Reads, IRRs) to their true genomic locations, enabling more accurate sizing of STR expansions. DRAGEN can also leverage phasing information to improve STR genotyping accuracy, especially in large heterozygous expansions. IRR recovery, the relative improvements to STR calling with proximity linking and phasing-aware genotyping are automatically enabled when running the DRAGEN Germline pipeline. All necessary resource files are autodetected for the supported references.

- **In-Repeat Read (IRR) Recovery**

Currently IRR recovery is only supported for patterns from 2 to 6 bases in length. Patterns outside of this range will not be evaluated by IRR recovery even if the catalog contains them. DRAGEN can leverage proximity information to recover In-Repeat Reads (IRRs) that would otherwise be unmapped or misaligned. This feature is particularly useful for detecting large repeat expansions that exceed the fragment length. Although the mapper already accounts for proximity information for better mapping, IRRs need special handling due to their low-complexity sequence content. IRR recovery is enabled by default when DRAGEN is in proximity mode and DRAGEN-STR will automatically adjust its parameters accordingly. It is highly recommended to not disable this feature when analyzing samples for repeat expansions.

IRR recovery uses a BED catalog of regions to index the possible locations of IRRs in the genome. The catalog contains the location of the STR and its motif. It is important to note that the catalog admits multiple entries for the same region, which allows multiple motifs to be specified for the same STR locus.

For example, the *RFC1* locus can be represented in the catalog as follows:

Chromosome	Start	End	Sequence	Name
4	39348424	39348479	AAAAG	RFC1
4	39348424	39348479	AAAGG	RFC1
4	39348424	39348479	AAGGG	RFC1
4	39348424	39348479	AAGAG	RFC1
4	39348424	39348479	AACGG	RFC1
4	39348424	39348479	ACGGG	RFC1
4	39348424	39348479	ACAGG	RFC1
4	39348424	39348479	AAAGGG	RFC1

DRAGEN provides BED catalogs for IRR recovery that cover all the locus of the default DRAGEN-STR catalogs. The default BED catalogs are in the `<INSTALL_PATH>/resources/irr_recovery/` directory. The catalog detection is automatic for supported references so there is no need to specify any extra command-line arguments to get the benefit of TruPath proximity information in DRAGEN-STR when using a supported reference genome and the defaults catalogs.

- **IRR Custom Catalogs**

DRAGEN accepts custom BED catalogs, for IRR recovery via the `--irr-recovery-str-bed` command-line argument. The custom catalog must follow the same format as the default catalogs provided by DRAGEN. When a custom catalog is provided, DRAGEN will use it instead of the default catalog for the selected reference genome. It is key to ensure that the custom catalog covers all the loci of interest for repeat expansion detection. If a locus is missing from the catalog, IRR recovery will not be performed for that locus, which may lead to reduced sensitivity for large expansions.

- **IRR Recovery BAM Tags**

Remapped IRRs are annotated in the output BAM file with the `tr` tag. The `tr` tag is a packed representation of a motif in 16bits, where the lower 12 bits represent the motif bases in 2-bit representation [A=00,C=01,G=10,T=11] and the upper 4 bits represent the motif length. The bases will be ordered from least significant to most significant. For example, the motif AAGGG of length 5 will be represented as follows:

```
Motif: AAGGG
Length: 5 = 0101
Packed: 0101 00 10 10 10 00 00
        5  - G G G A A
```

To avoid redundant motifs, the packed representation of a tag is always the minimum length pattern and the smallest (lexicographical) rotation between the forward and reverse complement of the motif sequence. For example, the motif CACA will be represented as AC. The `tr` tag will be applied to all the IRRs recovered via proximity information. The remapped reads will be mapped to a single position, on the first base of the corresponding STR region in the reference genome and set as unmapped with MAPQ 0.

- **Phasing**

When proximity mode is enabled, DRAGEN will make use of phasing information, if available, to improve the accuracy of repeat expansion genotyping. Phasing information can help to resolve ambiguities in the assignment of reads to haplotypes in diploid regions, which is particularly important for accurately estimating the repeat sizes in large heterozygous expansions. The output calls will still be unphased, maintaining the same VCF format as standard SBS runs, but the underlying genotyping model will leverage the phasing information to improve accuracy.

- **Sequencing Efficiency Correction**

Some loci can be affected by sequencing biases that lead to uneven coverage across different alleles. This can impact the accuracy of repeat expansion genotyping, especially in cases where one allele is significantly larger than the other. When proximity mode is enabled, DRAGEN will apply a sequencing efficiency correction to account for these biases. This correction adjusts the expected coverage for each locus based on empirical data, improving the accuracy of repeat size estimates. It is hard to distinguish between sequencing efficiency bias and mapping bias so this feature is only enabled for TruPath samples where the impact of mapping bias is minimized. The sequencing efficiency correction can be applied on a per-locus basis in the catalog file by adding the `SequencingEfficiencyCorrection` field to the respective catalog entry. For example:

```
{
  "LocusId": "DMPK",
```

```
"LocusStructure": "(CAG)*",  
"ReferenceRegion": "chr4:3076600-3076625",  
"VariantType": "Repeat",  
"SequencingEfficiencyCorrection": 1.2345 # example correction factor  
}
```

The correction factor should be determined empirically based on a set of control samples with known repeat sizes through orthogonal methods. DRAGEN has precomputed correction factors in the default catalogs that were calibrated for the following loci:

- *FMR1*
- *DMPK*
- *FXN*

Colocation Maps

Colocation maps extract proximity information to gain insight into long-range interactions in a sample. The output of the colocation module is a matrix with counts. Each cell in the matrix represents the number of interactions between two regions of the genome. The following figure is one instance of a small region on chr5 visualized as a heatmap.

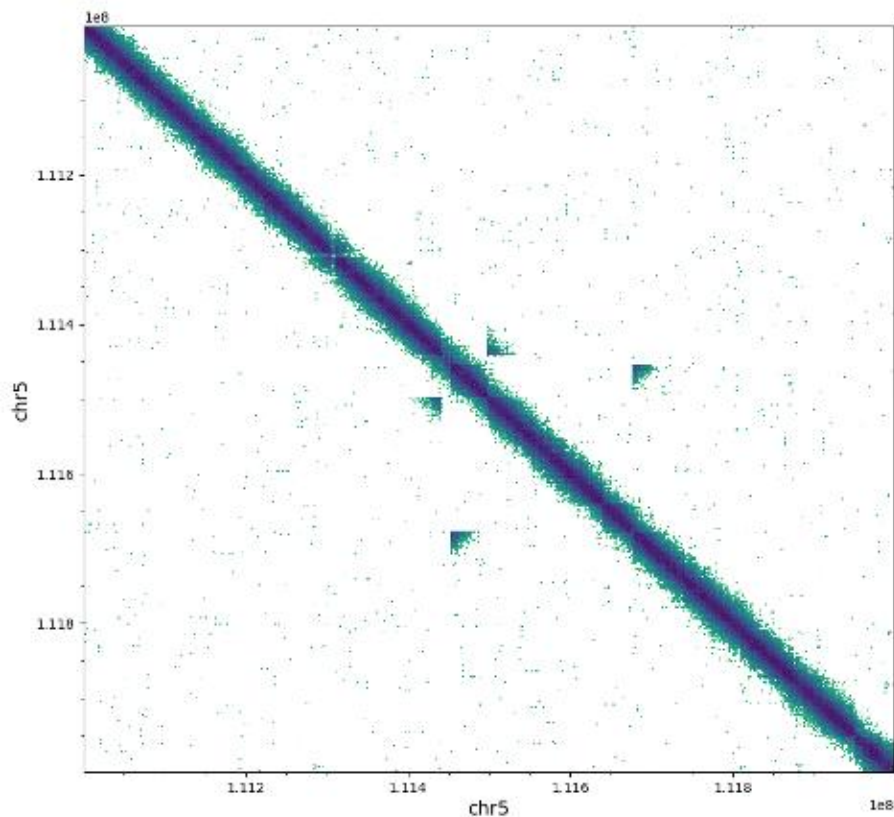


Figure 7. Example of a Colocation Plot

The darker a pixel, the greater the interactions exist between the two regions indicated on the axes. The main diagonal is formed by many fragments sequenced from each individual long template that land nearby on the flowcell. The triangles represent structural variants such as large deletions and breakends. Most off-diagonal pixels are either empty (white) or noise (green).

- **Colocation Map Generation**

Generating a colocation map is a three-step process.

- Collect all relevant alignments
- Compute the matrix
- Generate output

First, DRAGEN collects all reads to be included in the analysis. Any alignment that lands on a decoy contig, is below a mapping quality threshold or is a duplicate is ignored. The reads are separated into bins, where each bin represents about 2000bp of the genome.

The core algorithm for building the matrix is as follows:

```
for every read1:  
  for every read2 nearby:  
    matrix[read1.bin][read2.bin] += 1
```

A crucial detail here is the definition of *nearby*. For performance reasons we define a read2 to be nearby a read1 if it falls into a rectangle centered on read1. The size of this rectangle is determined by the proximity linkage of the sample.

Additional Options:

- For the colocation matrix the genome is split into bins of equal size. An alignment is assigned to the bin of its starting position. The bin size can be adjusted using `--colocation-bin-size`.
- It may be beneficial to exclude reads with specific flags from the analysis. The `--colocation-alignment-filter-flags` option can be used to signal DRAGEN to ignore any read with any of these BAM flags set. The value is the integer representation of bitflags.
- Use `--colocation-alignment-filter-min-mapq` to set a minimum MAPQ for any read.

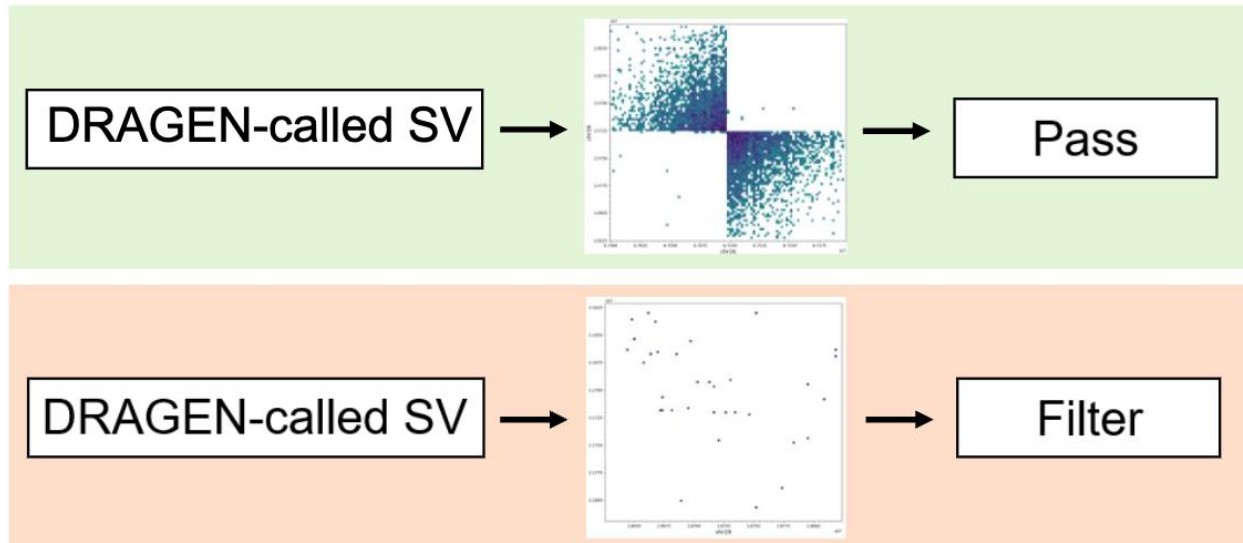
- **Cooler File**

The Colocation output is a cooler file containing a sparse representation of the colocation matrix. The file follows schema 3 of the [official cooler specification](#). DRAGEN produces a single-resolution cooler file. The matrix is stored in square mode but should be mostly symmetrical. Each pixel contains only one count field which is an int32. The file can easily be processed using the cooler CLI and Python API.

- **Colocation Filter**

The colocation filter uses the colocation map output by the colocation module to identify and filter breakends that are not supported by proximity data. For a candidate breakend between chrom1:pos1 and chrom2:pos2, the filter computes the sum of all bins in the colocation map within a bounding box (200kbp by default) centered on (chrom1:pos1, chrom2:pos2). To account for

samples with different depths and qualities, the colocation filter normalizes the region sum using the median non-zero diagonal value in the colocation map. If the normalized sum of the region is less than the threshold (1.0 by default), the `ColocationSum` filter will be applied to the candidate breakend in the VCF output. Note that if the `ColocationSum` filter is applied to one breakend in a pair, it will also be applied to the mate breakend record.



- **Running DRAGEN SV with Colocation Filter**

Colocation filtering is enabled by default if `enable-colocation` and `enable-sv` are both true. To enable the filter manually, set `--sv-enable-colocation-filter` to true when starting a TruPath DRAGEN analysis.

Additional Options:

- `--sv-colocation-filter-normalize-by-median`: If true, colocation filter will normalize the region sum by the median diagonal value of the colocation matrix (default: true)
- `--sv-colocation-filter-threshold`: Minimum (normalized) sum of region in colocation matrix to pass filter (default: 1.0)
- `--sv-colocation-filter-region-width`: Width (in bp) of square region in colocation matrix to compute sum of (default: 200kbp)
- `--sv-colocation-filter-min-svlen`: If true, Colocation filter will not run on intra-chromosomal breakend pairs that are within this distance of each other (default: 200kbp)
- `--sv-colocation-filter-inter-bnd`: If true, colocation filter will be applied to inter-chromosomal breakends (default: true)
- `--sv-colocation-filter-intra-bnd`: If true, colocation filter will be applied to intra-chromosomal breakends (default: true)

- **SV Output**

The SV VCF file will have the additional headers if the colocation filter is enabled:

```
##INFO=<ID=NORMALIZED_COLOC_SUM,Number=1,Type=Float,Description="The sum of the
square region in the colocation matrix centered on variant coordinates with
width 200000 and normalized by the median diagonal count">
##FILTER=<ID=ColocationSum,Description="The sum of the square region in the
colocation matrix centered on variant coordinates with width 200000 and
normalized by the median diagonal count does not meet the threshold of 1">
```

Examples VCF records can be seen below. The first breakend pair has the ColocationSum filter applied, as there was no colocation signal at all (NORMALIZED_COLOC_SUM=0.0000).

```
chr1 94900000 DRAGEN:BND:12587:0:1:0:0:0:0 A
A[chr2:39900000[ 280 ColocationSum
SVTYPE=BND;MATEID=DRAGEN:BND:12587:0:1:0:0:0:1;BND_DEPTH=52;MATE_BND_DEPTH=54;N
ORMALIZED_COLOC_SUM=0.0000 GT:GQ:PL:PR:MLQS:VF:VF1:VAF1:VF2:VAF2
0/1:280:330,0,637:38,3:..:38,3:23,3:0.115385:15,3:0.166667
chr2 39900000 DRAGEN:BND:12587:0:1:0:0:0:1 C ]chr1:94900000]C
280 ColocationSum
SVTYPE=BND;MATEID=DRAGEN:BND:12587:0:1:0:0:0:0;BND_DEPTH=54;MATE_BND_DEPTH=52;N
ORMALIZED_COLOC_SUM=0.0000 GT:GQ:PL:PR:MLQS:VF:VF1:VAF1:VF2:VAF2
0/1:280:330,0,637:38,3:..:38,3:15,3:0.166667:23,3:0.115385
chr3 52000000 DRAGEN:BND:65926:0:1:0:0:0:1 C
C]chr3:72000000] 955 PASS
SVTYPE=BND;MATEID=DRAGEN:BND:65926:0:1:0:0:0:0;BND_DEPTH=53;MATE_BND_DEPTH=54;N
ORMALIZED_COLOC_SUM=40.1980
GT:GQ:PL:PR:SR:SB:FS:MLQS:VF:VF1:VAF1:VF2:VAF2
0/1:715:999,0,712:29,8:38,23:21,17,1,22:44.774:..:48,31:21,19:0.475000:27,20:0.4
25532
chr3 72000000 DRAGEN:BND:65926:0:1:0:0:0:0 A
A]chr3:52000000] 955 PASS
SVTYPE=BND;MATEID=DRAGEN:BND:65926:0:1:0:0:0:1;BND_DEPTH=54;MATE_BND_DEPTH=53;N
ORMALIZED_COLOC_SUM=40.1980
GT:GQ:PL:PR:SR:SB:FS:MLQS:VF:VF1:VAF1:VF2:VAF2
0/1:715:999,0,712:29,8:38,23:21,17,1,22:44.774:..:48,31:27,20:0.425532:21,19:0.4
75000
```

Targeted Calling from TruPath Data

For WGS TruPath data, only lpa, hba, and smn will run when the Targeted Caller is enabled, but a custom list of supported targets can be enabled via the command line.

Proximity Coverage Reports

When proximity mapping is enabled, DRAGEN generates a parallel set of coverage reports filtered to include only linked reads. During template reconstruction, each read pair fragment is assigned a link quality score equal to the highest-quality link connecting it to other fragments. Only reads from fragments with link quality scores meeting or exceeding a specified threshold are included in the proximity coverage reports.

Proximity coverage reports are generated for each link quality threshold specified via `--proximity-min-linkq-threshold` (default: 10) and `--proximity-additional-linkq-thresholds` (default: 25, maximum 2 values). These reports are available for WGS and all specified QC coverage regions.

Report Name	Output File	Notes
-------------	-------------	-------

Proximity coverage metrics	<code>_proximity_linkqual<linkq-threshold>_coverage_metrics.csv</code>	Coverage statistics for linked reads
Proximity fine histogram coverage	<code>_proximity_linkqual<linkq-threshold>_fine_hist.csv</code>	Detailed coverage histogram for linked reads
Proximity histogram coverage	<code>_proximity_linkqual<linkq-threshold>_hist.csv</code>	Binned coverage histogram for linked reads
Proximity overall mean coverage	<code>_proximity_linkqual<linkq-threshold>_overall_mean_cov.csv</code>	Overall mean coverage for linked reads
Proximity per contig mean coverage	<code>_proximity_linkqual<linkq-threshold>_contig_mean_cov.csv</code>	Per-contig mean coverage for linked reads

These reports use the same format and metrics as standard coverage reports but reflect statistics computed exclusively from linked reads meeting the specified threshold.

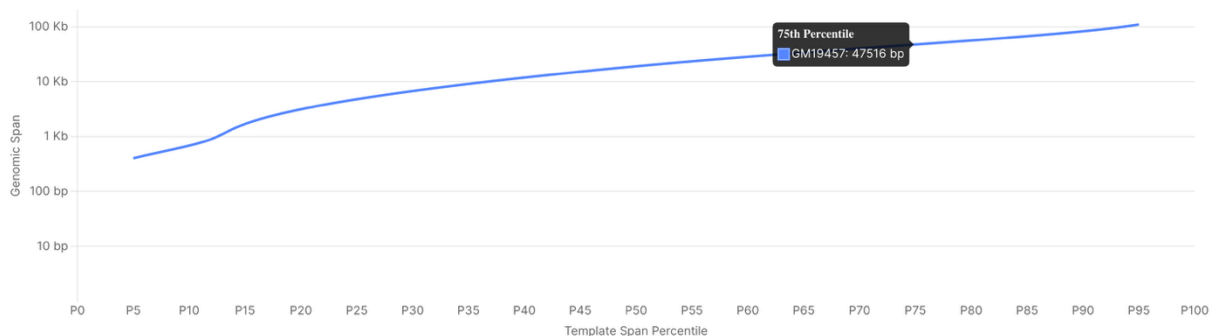
DRAGEN-Reports

DRAGEN-Reports now includes a manifest file to generate a report specific for TruPath WGS analysis at `trupath/germline_wgs.json` in the `/opt/dragen-reports/manifests` directory. In addition to all of the standard QC metrics and visualizations from the standard WGS reports, this report includes an additional Proximity tab to highlight data specific to the TruPath proximity mode analysis, such as:

- **Fit RMSE** - How well the proximity model fits the data
- **Q25 Proximity Rate** - Percentage of read-pairs with at least 1 neighbor above Q25
- **Q25 Proximity Coverage** - Average coverage over the genome of read-pairs with link-quality above Q25
- **P75 Template Size** - Size of the linked template molecules at the 75th percentile
- **NG50** - Size of the smallest phasing block needed to cover at least 50% of the genome

The Proximity tab also includes a number of visualizations of useful proximity data, such as: The distribution of template genomic lengths from `<prefix>.wgs_template_gdist.csv`

Template Genomic Span



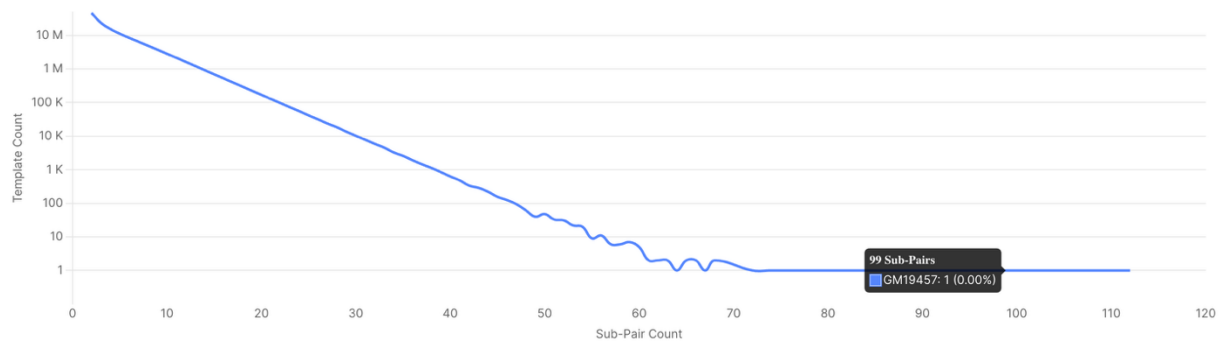
The reverse cumulative sum of variant phasing blocks by size, from `<prefix>.phase_blocks.gtf`

Cumulative Phase Block Sizes



The distribution of templates by sub-read count from `<prefix>.<qc-region>_template_gdist.csv`

Template Sub-Pair Counts



Pipeline Limitations

Illumina TruPath proximity enabled analysis has the following limitations in DRAGEN v4.5.2:

- Illumina TruPath proximity mode is currently supported for the DRAGEN Germline pipeline. The Somatic, RNA, UMI, MRD, and Methylation pipelines are not supported.
- DRAGEN downsampling is not supported. In order to maintain the proximity property of the TruPath assay, FASTQs should not be randomly downsampled.
- Only human samples using hg38 have been verified.
- Only TruPath data inputs from the Illumina TruPath Genome prep are supported at this time. Running `--enable-proximity=true` with non-TruPath data inputs will halt analysis.
- Phasing requires the use of a pangenome reference hash table with personalization enabled. Analysis will halt with low coverage to support personalization.
- For on-premises analyses, TruPath analysis requires a v4 DRAGEN server due to FPGA memory limitations. For reference, v4 servers have a server serial number which begins with the letters "AC". Cloud analysis is supported.
- MRJD requires at least 16x coverage to make calls; the caller will abort any attempt to call genes with insufficient aligned read coverage.

TruPath Genome Licensing

For DRAGEN OnPrem Servers and DRAGEN FPGA Cloud BYOL customers, this pipeline requires the 'Proximity' license. At this time, the cost of the Proximity license is included with the Illumina TruPath Genome prep and has been automatically assigned. For OnPrem Server users, due to hardware limitations the Proximity License has only been assigned to those with Phase 4 servers. For reference, v4 servers have a server serial number which begins with the letters "AC".

Resource Files

DRAGEN™ v4.5 requires updates to key resource files to function correctly and achieve optimum performance. All resource files are available for download at the Illumina DRAGEN™ Product Files support site here: https://support.illumina.com/sequencing/sequencing_software/dragen-bio-it-platform/product_files.html

- The hash table interface has been updated to format version 12 (HTv12). Hash tables must be updated to use v4.5.2. Existing hash tables built for v4.4 or older are not supported.
- Pangenome reference must be used for TruPath analysis
- The pipeline has been validated with hg38 only

Resource files applicable to Germline WGS TruPath analysis are listed below

Resource	Description	File name(s)
Hash Tables v12	Pre-built HTv6 format version 12 hash table for hg38.	Pangenome: hg38-alt_masked.cnv.graph.hla.methyl_cg.rna-12-r6.0-1.tar.gz
Pangenome Reference Builder Collection v6	HT mask BED, Graph BED, Graph exclusion BED, Graph msVCF and FASTA files for building hg38 reference.	hg38-pangenome-reference-collection-v6-1.tar.gz

Reference Genome Recommendations

Reference Support and Recommended Use for Germline TruPath Human Data

Human		hg38	Recommended Reference Type
Germline	SNV	Yes	Pangenome
	CNV	Yes	Pangenome
	SV	Yes	Pangenome
	Expansion Hunter	Yes	Pangenome
	Targeted Callers	Yes	Pangenome
	CNV	Yes	Pangenome
	SV	Yes	Pangenome
Annotation	Nirvana	Yes	n/a

Known Issues

Known issues of the DRAGEN™ v4.5.2 release

Component	Summary	Resolution/Workaround
MRJD	MRJD analysis can run out-of-memory on non-MRJD data. We observe < 0.5% failure rate on v4.5.2	None. Fixed in next release
MRJD	MRJD analysis can segfault or encounter watchdog termination on some data. We observe < 0.5% failure rate on v4.5.2	None. Fixed in next release
MRJD	MRJD occasionally exhibits inter-replicate variability in calls	None. Fixed in next release
STR	A small number of templates with extremely long genomic lengths will be created and tagged when IRR is enabled, impacting template and link length metrics as well as tagging some reads with BAM template tags (BX:Z) when they otherwise shouldn't have been included in a template. Negligible impact.	None. Ignore any link length distributions with extreme lengths in the template and link metrics output. Fixed in next release

SW Installation Procedure

- Download the desired installer from the Illumina support website and unzip the package.
- The archive integrity can be checked using: `./<DRAGEN 4.5.2 .run file> --check`
- Install the appropriate release based on your Linux OS with the command: `sudo sh <DRAGEN 4.5.2 .run file>`

Release History

Revision	Release Reference	Originator	Description of Change
00	1132881	Cobus De Beer	Initial release