STOmics

Stereo-seq ANALYSIS WORKFLOW FILE FORMAT MANUAL

Software Version: V7.0

Manual Version: A4

REVISION HISTORY

Manual Version: A0
Software Version: V4.1.0
Date: Apr. 2022
Description: Initial release

Manual Version: A0.1
Software Version: V4.1.0
Date: Jun. 2022
Description: Revised the data type of some data elements in the GEF file.

Manual Version: A1
Software Version: V5.1.3
Date: Sep. 2022
Description: Added cell bin GEF file demo and format specification; added IPR file format specification.

Manual Version: A1.1
Software Version: V5.1.3
Date: Dec. 2022
Description: Fixed some typos in GEF and IPR file format.

Manual Version: A2
Software Version: V5.5.3 - V5.5.4
Date: Jan. 2023
Description: Updated visit links for schematic diagram of GEF and IPR.

Manual Version: A3
Software Version: V6.0
Date: Mar. 2023
Description: Updated file versions of IPR (v0.1.0) and RPI (v0.0.2)

Manual Version: A3.1
Software Version: V6.1
Date: May. 2023
Description: Modification of fields in IPR and RPI; addition of FASTQ format description.

Manual Version: A4
Software Version: V7.0
Date: Oct. 2023
Description: Updated file version of IPR (v0.2.0); modification of fields in RPI.

Note: Please download the latest version of the manual and use it with the software specific for this manual.

TABLE OF CONTENTS

CHAPTER 1
OVERVIEW

1.1. About Software

Stereo-seq Analysis Workflow[1] (SAW) software suite is a set of pipelines that are bundled to position sequenced reads to their spatial location on the tissue section, and quantify spatial gene expression.

SAW download (Docker Hub): **https://hub.docker.com/r/stomics/saw**

SAW Github: **https://github.com/STOmics/SAW**

1.2. About Manual

This manual includes descriptions of key files formats generated from SAW, which help users better understand and make use of information from analysis results.

1.3. Terminologies and Concepts

Table 1-1 Terminologies and Concepts

Abbreviation	Full Name	Description
SN	**Serial Number**	Unique ID for Stereo-seq Chip T.
RIN	**RNA Integrity Value**	RNA integrity value measures the RNA degradation degree to indicate the integrity of RNA and evaluate the quality of the RNA sample. RIN values range from 1 (totally degraded) to 10 (intact). In Stereo-seq analysis, only tissue sample with a pre-measured RIN value greater than 7 should be used for further sequencing and bioinformatics analysis.
CID	**Coordinate ID**	Spatial position identifier, the artificially synthesized barcode sequence unique to each spot on the Stereo-seq Chip T.
MID	**Molecular ID**	Molecular identifier (same as UMI), the artificially synthesized sequence unique to each mRNA molecule captured from the sample which helps to differentiate the number of reads contributed by mRNA expression level due to amplification. Two copies of native transcripts from the same molecule captured on one DNB will result in two independent reads with the same CID but different MID. In contrast, two reads with identical CID and MID were originated from the same transcript but got amplified.
DNB	**DNA Nanoball**	DNA nanoball is the product of rolling-circle amplification (RCA) that is linearly amplified from the original circular single-stranded DNA template. DNB is the smallest capture unit on the Stereo-seq Chip T.
Bin	**Bin**	Bin (or Square Bin) is the analysis unit on the gene expression heat map. A bin is a fixed-sized square in which the expression value in this square is accumulated. Bins are not overlapped. The value followed by “Bin” represents the side length of the square. For bin 1, each DNB on the Stereo-seq Chip T is shown as a spot, which means one spot only contains the data from one DNB. Bin N means one spot on the heat map is an aggregation of data from N×N neighbor DNBs. For example, a spot of bin 100 covers data from 10,000 DNBs.
Cell Bin	**Cell Bin**	Similar to Square Bin, a cell bin stands for a region of cell on the expression map recognized by the algorithm (either from image or heat map). Expression within a cell bin region is accumulated, and neighbor cell bins are not over-lapped.

CHAPTER 2
FILE FORMAT

2.1 FASTQ

FASTQ is a common format for storing sequencing reads and corresponding quality evaluation. The Stereo-seq method is PE (paired-end) sequencing. Read 1 contains CID and MID information while read 2 contains captured mRNA sequencing data. During multi-sample sequencing, an additional sequence (sample barcode) is added to identify samples. When sequencing data is off, low-quality MID (containing N bases in MID or having two or more bases with quality value lower than 10) filtration is performed on read 1 and paired read 2. CID and MID are added to readID of read 2 that has been filtered by MID quality, and single strands in read 2, containing mRNA information, are written into the file in FASTQ format as the original sequencing data, where sample barcode is removed.

Q40 FASTQ and Q4 FASTQ are two optional output formats for the original sequencing data. Q40 adopts an evaluation system that describes the quality of sequenced bases with 41 quality values. Q4 means a similar evaluation system but with 4 quality values. Q40 FASTQ, consists of a pair of read files,
read 1 for CID, MID information and read 2 for captured mRNA sequencing data respectively.

Example of Q40 FASTQ:

```
# read 1
@E100026571L1C001R00300000000/1
TGTCCAACGGAGACGGCTCCGACAAGGCACTGGCA
+
>DG;<BGH=>*EFE8*G/3E@2:F0-GBGG188F<

# read 2
@E100026571L1C001R00300000000/2
GTCTCACCATACTTTTACAAAGTTATTTCAACCCAAATCACAATTTAAGAATTATTTGTTCTACCTATGCCACACT
TTAAATAAATGTCTATTAAAACCA
+
-GFEECG?ECBFF<=@A@<E@><;FGCF=>=E53FEF5>FGF@,0ADE9CEAG2GBE@
HF3EA<CE;G2F@=G8=?@G9FBGE.EG6G2;974E*D9DE9
```

Q4 FASTQ is an output format with only one read file, which is split from a data set (containing 16 or 64 parts). ReadID in the file starts with "@" and contains read name and encoded CID and MID information. The sequence part contains captured mRNA sequencing data. Because of the combined output format and fewer kinds of quality values, the file storage space is greatly saved.

Example of Q4 FASTQ:

```
@FP300000513L1C002R00400000218 CE242DF29A57 97D26
GTGTAGTGAACCCCATGGTAGTTTTCTGATTGTTGTTAAAAAAAATGACTTAACATATTACATGGACACTCAAT
AAAAATGTTTTATTTCCTGTTGAAAA
+
FFFFFFFFFFFF8F8FFFFFFFFFFFFF8FFFFFFFFF8FF8FFF8FFFFFFF,FFFFFFFFFFF8FFFFFF8F
8F,F8FFFFFF,FFFFFFFFF,FFF
```

2.2 BAM

The BAM[2] file format is a binary format for saving sequence alignment and gene annotation data. SAW **mapping** BAM adds custom tags in the BAM optional field to record reads coordinates, CID and MID information. **count** BAM adds annotation information in the tag field. Custom tags are described in Table 2-1.

Table 2-1 BAM custom tags

Tag	Description
Cx:i	x coordinate of CID.
Cy:i	y coordinate of CID.
UR:Z	The hexadecimal representation of uncorrected binary-encoded MID.
XF:Z	Mapping region on the reference genome. Valid value: 0=EXONIC, 1=INTRONIC, 2=INTERGENIC. 3=rRNA
GE:Z	Annotated gene name.
GS:Z	'+' or '-', indicating forward/reverse strand respectively.
UB:Z	The hexadecimal representation of count corrected binary-encoded MID.

Example of **mapping** BAM:

```
E100026571L1C009R00301275185    16      1       3000095 255
26M121066N74M   *       0       0       GGCTTTTTTTTTTTTTTTTTTTTTTTTTTCTAA
ATATTGGGTTTTATTAGCACCATGATAACTGTATATTAATTTGCACTGACTGTCATAACAAAATAC      G+
:GFFGGFGFFGFFGFGGFFGFFFFFCFGFCFGGGFGGFGFFFFGGFGGFGFFFGGFFGFFFGFGFGFFGFFGF
GFFFFGFFFFFFFFGGFFGGFFGEF    NH:i:1  HI:i:1    AS:i:88 nM:i:0  Cx:i:4826
Cy:i:11598      UR:Z:6FA29
```

Example of **count** BAM:

```
E100026571L1C002R00703943265    1040    1       3082766 255
11M132671N89M   *       0       0       CTGCTGCAGCTTTTTTTTCTTTGAGATTTA
TTTTTATGCTATGTGTATGGGTATTTTGCCTGCATATATGTCTATGCACCATGTGTGTGCAGTGCTTGAG
FFFFFECGFDCFGDGDFEE@EEGIBFGGCGFFGACGFCGFFDGDGFFFFFFEGCDFCGFFGG@FFF=EFFDGGG
GGFDGFFFGGGFGFFGGGFFGGGDFG    NH:i:1  HI:i:1    AS:i:88 nM:i:0  Cx:i:7767
Cy:i:18052      UR:Z:7AE49      XF:i:0  GE:Z:Xkr4       GS:Z:-  UB:Z:79E49
```

2.3 Mapped CID List with Reads Count File

mapping pipeline outputs mapped CID list file with reads counts for each CID. This file stores CID coordinates and reads count for each coordinate. The list does not have a header. The three columns are x coordinate, y coordinate and MID count.

Example of mapped CID list with reads count file:

```
14195   16619   1
19945   14450   2
14548   9438    1
```

2.4 Gene Expression File

Gene expression file (GEF) is a data management and storage format designed to support multi-dimensional data storage and high computation efficiency. Stereo-seq analysis workflow generates Square Bin GEF and Cell Bin GEF files. Square Bin file format is a hierarchically structured data model that stores one or bin combined gene expression matrices in different bin sizes. Cell Bin file format stores expression information within each cell.
Each GEF container organizes a collection of spatial gene expression matrices. It includes two primary data objects, Group and Dataset. A dataset is a multidimensional array of data elements. Group object is analogous to file system directory which organizes datasets and other groups in hierarchies.

2.4.1 Square Bin GEF

The first level of GEF includes four group objects: “geneExp” (required), “wholeExp” (optional), “wholeExpExon” (optional), and “stat” (optional). Group “geneExp” contains groups of gene spatial expression data in one or multiple bin size. Group “wholeExp” contains datasets that record expression level and gene type count of each coordinate in one or multiple bin sizes. Group “wholeExpExon” contains datasets that record exon level of each coordinate in one or multiple bin sizes. Group “stat” saves gene names, total MID count and spatial pattern enrichment score of each gene. “Attributes” of the file records the version of GEF format, software version, and omics information. “Attribute” in each dataset records the key metrics of that dataset. Check **https://github.com/STOmics/SAW/tree/main/Documents/FileFormat** to get the schematic diagram of square bin GEF. The field names and field data types of Square Bin GEF are described in Table 2-2. SAW outputs three GEF files in the whole process. Please check Table 2-3 to find the description.

Table 2-2 Square Bin Gene Expression File Text Fields Description

Attributes			
File Attributes	**DataType**	**Example**	**Description**
version	**uint32**	**2**	Gene expression file format version.
geftool_ver	**uint32[3]**	**0,7,11**	geftool version. It can be used as an individual tool to manipulate GEF files.
omics	**S32**	**b’Transcriptomics’**	Omics name.
gef_area	**float32**	**4.4410855E10**	Tissue or labeled tissue area in square nanometers.
bin_type	**S32**	**bin**	Bin type of the GEF file.
/geneExp/binN/expression: Dataset “expression” is a 1D array which stores coordinates and MID counts of each gene in the bin size of N, aggregated by gene name.			
Dataset Attributes	**DataType**	**Example (bin1)**	**Description**
minX	**int32**	**59820**	Minimum x coordinate in bin N.
minY	**int32**	**102086**	Minimum y coordinate in bin N.
maxX	**int32**	**73040**	Maximum x coordinate in bin N.
maxY	**int32**	**120539**	Maximum y coordinate in bin N.
maxExp	**uint32**	**28**	Maximum MID count in a spot when the bin size is N. Data type for “maxExp” is dynamically changed for each sample.
resolution	**uint32**	**500**	Physical pitch (nm) between neighbor spots.
Dataset DataType: compound	**DataType**	**Example (bin1)**	**Description**
x	**int32**	**71032**	x coordinate in bin N.
y	**int32**	**103180**	y coordinate in bin N.
count	**uint8/uint16/ uint32**	**1**	MID count at (x, y) when bin size is N. Data type for “count” is consistent with “maxExp” in the “Attributes.”

[optional] /geneExp/binN/exon: Dataset "exon" is a 1D array which stores exon expression of each gene in the bin size of N, aggregated by gene name.			
Dataset Attributes	**DataType**	**Example (bin1)**	**Description**
maxExon	**int32**	**21**	Max exon expression in binN.
Dataset DataType: 1D array	**DataType**	**Example (bin1)**	**Description**
count	**uint8/uint16/ uint32**	**0**	Exon expression in binN at coordinate (x,y), the index is same to the index in the "expression" dataset. Data type for "count" is dynamically changed for each sample.

/geneExp/binN/gene: Dataset "gene" is a 1D array which stores the gene names, the starting row indexes in dataset "expression", and row counts.			
Dataset DataType: compound	**DataType**	**Example (bin1)**	**Description**
gene	**S32**	**b'Gm16045'**	Gene name.
offset	**uint32**	**21**	The starting row index in dataset "expression" for the gene. In this example, the gene expression data for gene "Gm16045" starts from row 21 in the dataset "expression."
count	**uint32**	**2**	Row count. In this example, expression data for gene "Gm16045" is recorded in row 21 and 22 (2 rows) in the dataset "expression."

[optional] /wholeExp/binN: Dataset "binN" is a 2D array (matrix) which stores the MID count and gene type count at each spot.			
Dataset Attributes	**DataType**	**Example (bin1)**	**Description**
number	**uint64**	**22879557**	Number of non-zero spots in the dense matrix.
minX	**int32**	**59820**	Minimum x coordinate in bin N.
lenX	**int32**	**13221**	Length of x.
minY	**int32**	**102086**	Minimum y coordinate in bin N.
lenY	**int32**	**18454**	Length of y.
maxMID	**uint32**	**2155**	Maximum MID count in a spot.
maxGene	**uint32**	**846**	Maximum gene type count in a spot.
resolution	**uint32**	**500**	Pitch (nm) between neighbor spots.
Dataset DataType: 2D array (XxY), compound	**DataType**	**Example (bin1)**	**Description**
MIDcount	**uint8/uint16/ uint32**	**1**	MID count in the spot. The spot coordinate can be identified from the row and column index of the 2D matrix plus the "minX" and "minY" specified in the attributes. Data type for "MIDcount" is dynamically changed for each sample.
genecount	**uint16**	**1**	Gene count in the spot. The spot coordinate can be identified from "Attributes" and the indexes of the 2D array.

[optional] /wholeExpExon/binN: Dataset "binN" in "/wholeExpExon/" Group is a 2D array (matrix) which stores the exon expression count at each spot.			
Dataset Attributes	**DataType**	**Example (bin1)**	**Description**
maxExon	**uint32**	**21**	Maximum exon expression count in a spot when the bin size is N.
Dataset DataType: 2D array	**DataType**	**Example (bin1)**	**Description**
MIDcount	**uint8/uint16/ uint32**	**0**	MID count in the spot. The spot coordinate can be identified from the row and column index of the 2D matrix plus the "minX" and "minY" specified in the attributes. Data type for "MIDcount" is dynamically changed for each sample.

[optional] /stat/gene: Dataset "gene" is a 1D array which stores the MID count and spatial pattern enrichment score (E10) of each gene. The array is order by the MID count in descending order.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxE10	**float32**	**65.53**	Maximum E10 score.
minE10	**float32**	**0.**	Minimum E10 score.
cutoff	**float32**	**0.1**	Threshold for filtering spots that will be used for computing E10. In this example, 0.1 means that the spots whose MID count is in the top 10% are used for calculating the spatial enrichment score.
Dataset DataType: compound	**DataType**	**Example**	**Description**
gene	**S32**	**b'Ptgds'**	Gene name.
MIDcount	**uint32**	**229502**	MID count for the gene.
E10	**float32**	**65.53**	The spatial pattern enrichment score (E10) for the gene.

The distinctions of each SAW output GEF files are explained in Table 2-3.

Table 2-3 SAW Output GEF Files Description

GEF Name	SAW Pipeline	Example	Description
SN.raw.gef	**count**	**SS200000135TL_ D1.raw.gef**	**`count`** output raw GEF, it only includes geneExp Group for the bin size of 1. The origin of expression matrix has been calibrated to (0,0).
SN.gef	**count**	**SS200000135TL_ D1.gef**	**`count`** output full GEF file. It contains geneExp Group and wholeExp Group for the bin size of 1, 10, 20, 50, 100, 200, and 500. SN.gef is also the only one that includes stat Group. The origin of expression matrix has been calibrated to (0,0), and its offsets are the same with SN.raw.gef. SN.gef is the input file for visualization.
SN.tissue.gef	**tissueCut**	**SS200000135TL_ D1.tissue.gef**	**`tissueCut`** output GEF file for the tissue-covered region. It only includes geneExp Group for the bin size of 1. The coordinates in the matrix and the offsets are all same with SN.raw.gef.

GEF Name	SAW Pipeline	Example	Description
SN.<stainType>.gef	tissueCut	SS200000135TL_D1.ssDNA.gef	`tissueCut` output full GEF file for the tissue-covered region. The coordinates in the matrix and the offsets are all same with SN.tissue.gef. It contains geneExp Group and wholeExp Group for the bin size of 1, 10, 20, 50, 100, 200, and 500.
SN.<label>.raw.label.gef	tissueCut	B01020C2.Label01.raw.label.gef	`tissueCut` output GEF file for the labeled ('aa' in the example) tissue-covered region. It only includes geneExp Group for the bin size of 1. The coordinates in the matrix and the offsets are all the same with SN.raw.gef.
SN.<label>.label.gef	tissueCut	B01020C2.Label01.label.gef	`tissueCut` output full labeled tissue-covered GEF file.

2.4.2 Cell Bin GEF

The first layer of Cell Bin GEF contains one required group “cellBin” and multiple optional datasets. The second layer "codedCellBlock" is optional, which stores precomputed data used in the rendering of StereoMap. “Attri-butes” of the file records the version of GEF format, software version, and omics information. “Attribute” in each dataset records the key metrics of that dataset. Check https://github.com/STOmics/SAW/tree/main/Documents/FileFormat to get the schematic diagram of cell bin GEF. The field names and field data types of Cell Bin GEF are described in Table 2-4.

Table 2-4 Cell Bin Gene Expression File Text Fields Description

Attributes			
File Attributes	**DataType**	**Example**	**Description**
geftool_ver	uint32[3]	0,7,11	geftool version. It can be used as an individual tool to manipulate GEF files.
offsetX	int32	0	Minimum x coordinate in bin 1.
offsetY	int32	0	Minimum y coordinate in bin 1.
omics	S32	b‘Transcriptomcis’	Omics name.
resolution	uint32	500	Pitch (nm) between neighbor spots.
version	uint32	2	Gene expression file format version.
/cellBin/cell: Dataset “cell” is a 1D array which stores basic information and indices information of cells and expression.			
Dataset Attributes	**DataType**	**Example**	**Description**
averageArea	float32	494.666	Average area for cells in pixel.
averageDnbCount	float32	194.299	Average number of mRNA-captured DNBs in a cell.
averageExpCount	float32	541.715	Average MID count in cell.
averageGeneCount	float32	310.157	Average gene count in cell.
maxArea	uint16	1925	Maximum area for cells in pixel.
maxDnbCount	uint16	883	Maximum number of mRNA-captured DNBs in a cell.
maxExpCount	uint16	3018	Maximum MID count in cell.
maxGeneCount	uint16	1415	Maximum gene count in cell.
maxX	int32	17658	Maximum x coordinate of the cell’s center of mass.
maxY	int32	19422	Maximum y coordinate of the cell’s center of mass.

/cellBin/cell: Dataset "cell" is a 1D array which stores basic information and indices information of cells and expression.			
medianArea	**float32**	**474.**	Median area for cells in pixel.
medianDnbCount	**float32**	**183.**	Median number of mRNA-captured DNBs in a cell.
medianExpCount	**float32**	**491.**	Median MID count in cell.
medianGeneCount	**float32**	**289.**	Median gene count in cell.
minArea	**uint16**	**2**	Minimum area for cells in pixel.
minDnbCount	**uint16**	**0**	Minimum number of mRNA-captured DNBs in a cell.
minExpCount	**uint16**	**0**	Minimum MID count in cell.
minGeneCount	**uint16**	**0**	Minimum gene count in cell.
minX	**int32**	**2933**	Minimum x coordinate of the cell's center of mass.
minY	**int32**	**5568**	Minimum y coordinate of the cell's center of mass.
Dataset DataType: compound	**DataType**	**Example**	**Description**
id	**uint32**	**10**	Cell ID index, the start ID is 0. In the Example, 10 represent the 10th cell in the data-set.
x	**int32**	**541**	The x coordinate of the cell's center of mass. In the Example, the x coordinate of the 10th cell's center of mass is 541.
y	**int32**	**190**	The y coordinate of the cell's center of mass. In the Example, the x coordinate of the 10th cell's center of mass is 190.
offset	**uint32**	**494**	The start row index of the cell in the "/cellBin/cellExp" dataset. The example represents that the gene ID index and total MID count information of the 10th cell in the "/cellBin/cellExp" dataset start from the 494th row.
geneCount	**uint16**	**100**	Gene count in the cell. In the example, 100 represents that the 100 rows in the "/cellBin/cellExp", start from the 494th to the 593th row, contains the gene ID indices and total MID count of the gene for the 10th cell in "/cellBin/cell" dataset.
expCount	**uint16**	**500**	Cell MID count.
dnbCount	**uint16**	**200**	mRNA-captured DNBs of the cell.
area	**uint16**	**474**	Cell area in pixel.
cellTypeID	**uint32**	**0**	Cell type ID.
clusterID	**uint32**	**20**	Cell cluster ID.

/cellBin/cellBorder: Dataset “cellBorder” is a 3D array which stores the lists of points for the bounding polygons of the cell.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxX	**int32**	**16127**	Maximum x coordinate of the bounding box of the cell.
maxY	**int32**	**16663**	Maximum y coordinate of the bounding box of the cell.
minX	**int32**	**11129**	Minimum x coordinate of the bounding box of the cell.
minY	**int32**	**12784**	Minimum y coordinate of the bounding box of the cell.
Dataset DataType: 3D array	**DataType**	**Example**	**Description**
-	**32*(int16,int16)**	**[[-17,-11],[-15,-5]... [32767,32767]]**	A list of 32 coordinates recording the differences between cell bounding points and the cell’s center of mass (0,0). The real coordinate of cell’s center of mass (x, y) can be obtained from “cell” dataset using cellID.

/cellBin/cellExp: Dataset “cellExp” is a 1D array which stores the expression information of each cell.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxCount	**uint16**	**336**	Maximum MID count of a gene in a cell.
Dataset DataType: compound	**DataType**	**Example**	**Description**
geneID	**uint32**	**1610**	Gene IDs of the genes detected in the cell. ID is the index of “gene” dataset. In the example, 1610 represents the 1610th item in the “gene” dataset, and the name of the gene can be acquired in “gene” dataset.
count	**uint16**	**3**	MID count for the gene. In the example, (assume this is the 0th item in the “cellExp” dataset, from the “offset” and “geneCount” record in the “cell” dataset we can know that the 0th item in the “cellExp” belongs to the cell whose cellID=0) the MID count for the gene (geneID=1610) in the cell (cellID=0) is 3.

[optional] /cellBin/cellExon: Dataset “cellExon” is a 1D array which stores the exon information for each cell.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxExon	**uint16**	**5793**	Maximum exon count of a gene in all cells.
minExon	**uint16**	**0**	Minimum exon count of a gene in all cells.
Dataset DataType: 1D array	**DataType**	**Example**	**Description**
-	**uint16**	**16**	Exon count in a cell, the index of the array is same to the cellID in the “cell” dataset.

[optional] /cellBin/cellExpExon: Dataset “cellExpExon” is a 1D array which stores exon expression information for each cell.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxExon	**uint16**	**336**	Maximum exon count of a gene in a cell.
Dataset DataType: 1D array	**DataType**	**Example**	**Description**
-	**uint16**	**3**	Exon count (MID) for the gene. The index is same to the “cellExp” dataset. In the example, (assume this is the 0th item in the “cellExpExon” dataset, since the index is same to “cellExp” dataset, from the “offset” and “geneCount” record in the “cell” dataset we can know that the 0th item in the “cellExpExon” belongs to the cell whose cellID=0) the exon count (MID) for the gene (geneID=1610) in the cell (cellID=0) is 3.

/cellBin/cellTypeList: Dataset “cellTypeList” is a 1D array which stores cell types of each cell.			
Dataset DataType: 1D array	**DataType**	**Example**	**Description**
-	**S32**	**b'default'**	Cell type, “default” stands for undefined cell type.

/cellBin/gene: Dataset “gene” is a 1D array which stores the indices of cell and expression information of each gene.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxCellCount	**uint32**	**5718**	Maximum number of cells a gene can be detected.
maxExpCount	**uint32**	**55361**	Maximum MID count of a gene.
minCellCount	**uint32**	**1**	Minimum number of cells a gene can be detected.
minExpCount	**uint32**	**1**	Minimum MID count of a gene.
Dataset DataType: compound	**DataType**	**Example**	**Description**
geneName	**S32**	**b'AC149090.1'**	Gene name.
offset	**uint32**	**0**	The start row index of the gene in “/cellBin/geneExp” dataset. In the example, 0 means that start from the 0th item in “/cellBin/geneExp” dataset records the cellIDs and total MID count information of ”AC149090.1”.
cellcount	**uint32**	**60**	Number of cells a gene can be detected. In the example, 60 represents that start from the 0th item to the 59th item records the information of gene ”AC149090.1”.
expCount	**uint32**	**100**	Sum of MID count for the gene. In the example, the total MID count of ”AC149090.1” is 100.
maxMIDcount	**uint16**	**4**	Maximum MID count of a gene in a cell. In this case, the maximum MID count of gene ”AC149090.1” in a cell is 4.

/cellBin/geneExp: Dataset “geneExp” is a 1D array which stores cell and expression information of each gene.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxCount	**uint16**	**10**	Maximum MID count of a gene.
Dataset DataType: compound	**DataType**	**Example**	**Description**
cellID	**uint32**	**1247**	cellID that contains the gene whose index is same to the index in “gene” dataset. In the example, (assume we use the 0th item in “geneExp” dataset) 1247 shows that the gene ”AC149090.1” appears in the cell whose cellID is 1247.
count	**uint16**	**3**	The MID count of the gene, whose index is same to the index in “gene” dataset, in the cellID. In the example, the MID count of gene ”AC149090.1” in the cell (cellID=1247) is 3.

[optional] /cellBin/geneExon: Dataset “geneExon” is a 1D array which stores the exon expression information of each gene.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxExon	**uint32**	**55361**	Maximum exon count of a gene.
minExon	**uint32**	**0**	Minimum exon count of a gene.
Dataset DataType: 1D array	**DataType**	**Example**	**Description**
-	**uint32**	**97**	Total exon count of a gene, the index of “geneExon” dataset is same to the “gene” dataset. In the example, (assume this is the 0th item in the “geneExon” dataset, and gene ”AC149090.1” is the 0th item in the “gene” dataset) the exon count of gene ”AC149090.1” is 97.

[optional] /cellBin/geneExpExon: Dataset “geneExpExon” is a 1D array which stores the exon expression information in cells of each gene.			
Dataset Attributes	**DataType**	**Example**	**Description**
maxExon	**uint16**	**336**	Maximum exon expression of a gene in a cell.
Dataset DataType: 1D array	**DataType**	**Example**	**Description**
-	**uint16**	**3**	Exon count of a gene in a cell. The index of “geneExpExon” dataset is same to the “geneExp” dataset. In the example, (assume this is the 0th item in the “geneExpExon” dataset, since the index is same to “geneExp” dataset, from the “offset” and “cellCount” record in the “gene” dataset we can know that the 0th item in the “geneExpExon” dataset belongs to the gene ”AC149090.1”) 3 stands for the exon count of gene ”AC149090.1” in cell 1247 is 3.

/cellBin/bockIndex:
Dataset “bockIndex” is a 1D array which stores the matrix block partition information.

Dataset DataType: 1D array	DataType	Example	Description
-	uint32	0	Cell count in each partition block. cnt=blockIndex[i+1]-blockIndex[i]

/cellBin/bockSize:
Dataset “bockSize” is a 1D array which stores the block size of partition.

Dataset DataType: 1D array	DataType	Example	Description
-	uint32	256, 256, 104, 104	4-element array. The 4 items represent the block length in x-axis, block length in y-axis, block count in x-axis,

[optional]/codedCellBlock:
Group "codedCellBlock" stores pre-computed data for rendering in StereoMap.

Group Attributes	DataType	Example	Description
info	string	{"@type": "neuroglancer_annotations_v1", ...}	Metadata of encoded precomputed data in JSON.

[optional]/codedCellBlock/L0/0_1:
Dataset "0_1" is an example chunk encoded pre-computed data, including id, geometry, and so on.

Dataset Data Type:Bytes	DataType	Example	Description
	H5T_OPAQUE	1F 8B 08 00 …	Bytecode of the chunk.

2.5 Gene Expression Matrix

Gene expression matrix stores genes spatial expression data. SAW generates multiple gene expression matrix files in the workflow, the basic format requires four columns with a header row that shows the column names. The four columns are gene name, x coordinate, y coordinate, and MID count. The origin of `tissueCut` generated gene expression matrices have been calibrated to (0, 0). The header of expression matrix for maximum area enclosing rectangle region has six annotation rows start with "#" before the column rows. The header field names and field types are described in Table 2-5.

Table 2-5 Gene Expression Matrix Header Fields Description

Fields	DataType	Example	Description
#FileFormat	**string**	**GEMv0.1**	Gene expression matrix file format version.
#SortedBy	**string**	**None**	Gene expression matrix sorting strategy. Valid values: "geneID", "x", "y", "MIDCount", "None".
#BinType	**string**	**Bin**	Bin type of the GEM file.
#BinSize	**string**	**1**	(Please check 1.3 Terminologies and Concepts Bin)
#Omics	**string**	**Transcriptomics**	(Omics name.
#Stereo-seqChip	**string**	**SS200000135TL_D1**	Stereo-seq Chip T serial number.
#OffsetX	**uint32**	**1**	X coordinate of the origin before calibration.
#OffsetY	**uint32**	**1**	Y coordinate of the origin before calibration.
geneID	**string**	**Cr2**	Gene name.
x	**uint32**	**16809**	X coordinate of the spot.
y	**uint32**	**8546**	Y coordinate of the spot.
MIDCount	**uint32**	**1**	Number of MIDs at (x, y) for the gene in the corresponding row.
ExonCount	**uint32**	**0**	(Optional) Number of exon count at (x, y) for the gene in the corresponding row.
CellID	**uint32**	**55892**	(Optional) CellID for (x, y).

Example of `GEM`:

```
#FileFormat=GEMv0.1
#SortedBy=None
#BinType=Bin
#BinSize=1
#Omics=Transcriptomics
#Stereo-seqChip=SS200000135TL_D1
#OffsetX=0
#OffsetY=0
geneID        x        y        MIDCount        ExonCount        CellID
Ptgds        7585        19729        1        1        55892
Cdk8        7582        19730        2        0        55892
1500011K16Rik        7585        19730        2        2        55892
```

2.6 Image Process Record File

Image process record (IPR) file is designed to record the whole-life information of a microscopic staining image from photo-taking to processing. Each staining image (ssDNA/DAPI, IF, H&E) includes six basic groups, “ImageInfo”, “QCInfo”, “Stitch”, “TissueSeg”, “CellSeg”, and “Register”, which are used to store microscopy photo-taking information, image quality control information, image stitching records, tissue segmentation records, cell segmentation records (optional), and registration records. Check **https://github.com/STOmics/SAW/tree/main/Documents/FileFormat** to get the schematic diagram of IPR.

Table 2-6 IPR File Format and Text Fields Description

File Attribute & Dataset	
Attributes	**Description**
IPRVersion	IPR file format version.
Dateset	**Description**
Preview	A 2D matrix merges stitched image, tissue segmentation boundary and cell segmentation boundary.

/<stainType>/ImageInfo: Group records basic image information.	
Attributes	**Description**
AppFileVer	Microscope software version.
BackgroundBalance	Background balance.
BitDepth	The bit-depth of a camera sensor describes its ability to transform the analog signal coming from the pixel array into a digital signal.
Brightness	Relative intensity affecting a person or sensor.
ChannelCount	Number of RGB channels.
ColorEnhancement	Whether enhanced image color display or not.
Contrast	The difference in color and intensity of the depicted object from its background.
DeviceSN	Microscope device serial number.
DistortionCorrection	Whether fixed distortion or not.
ExposureTime	Exposure time in ms.
FOVHeight	Height of an individual FOV in pixel.
FOVWidth	Width of an individual FOV in pixel.
Gain	Amplification applied to the signal by the image sensor.
Gamma	The coefficient links between the human eye and the digital camera.
GammaShift	Whether adapt the digital image taken with the help of a linearly recording camera to the nonlinear perception of the human eye or not.
Illuminance	Intensity of light.
Manufacture	Microscope manufacture.
Model	Microscope model.
Overlap	Overlapping pixels between single tiles.
Pitch	Physical pitch (nm) between neighbor spots.

/<stainType>/ImageInfo: Group records basic image information.	
PixelSizeX	Size of pixel in x direction.
PixelSizeY	Size of pixel in y direction.
QCResultFile	Prefix of ImageQC/ImageStudio result file, the unique identifier of the image.
ScanChannel	Fluorescence channel.
ScanCols	Number of columns scanned.
ScanObjective	Magnification power of the scan objective lens.
ScanRows	Number of rows scanned.
ScanTime	Scan date and time.
Sharpness	Degree of clarity of the edge(s) of the image.
StereoResepVersion	Stereo-resep version.
StitchedImage	Whether the corresponding image is a panorama image (true) or a set of tiled images (false).
STOmicsChipSN	Stereo-seq Chip T serial number.
WhiteBalance	An adjustment in electronic and film imaging that corrects the color balance of the lighting.

Dataset DataType: 1D array	Description
RGBScale	RGB color.

/<stainType>/QCInfo: Group records the QC information of the image.	
Attributes	**Description**
ClarityScore	Reference score for evaluating the clearness of cell boundaries.
Experimenter	Email of the experimenter who did QC for the image.
GoodFOVCount	Number of FOVs that have identified more than 3 track cross points.
ImageQCVersion	ImageQC/imageStudio version.
QCPassFlag	Whether the corresponding image passed QC.
RemarkInfo	Any remarks, notes, comments on the image.
StainType	Stain type of microscopy image.
TotalFOVCount	Total number of FOVs.
TrackLineScore	Reference score for evaluating whether the detected track lines can be used as references for image stitching and registering with gene expression matrix. (This score only evaluate whether the program detected track lines on the image, it does not infer the clarity of the lines or the images).
TrackLineChannel	Shooting channel of track lines.
TrackCrossQCPassFlag	Whether the QC of the track-cross point is passed.
ScopeStitchQCScore	Score to assess the stability of microscope overlap.
ScopeStitchQCMatrix	Matrix of overlap deviation of each microscope FOV.
ScopeStitchQCPassFlag	Whether the QC of microscope stitch is passed.

/<stainType>/QCInfo: Group records the QC information of the image.	
Attributes	**Description**
TemplateValidArea	Proportion of encircled area by the detected points that match the global track line template (within the error range of 5 pixels), to the entire imagearea.
TemplateRecall	Proportion of detected points, which match the global track line template (within the error range of 5 pixels), to the points derived from the chip track line rules.
ClarityCutSize	Size of cut images (based on FOV) for clarity evaluation.
ClarityCounts	Counts of cut images in different categories.
ClarityOverlap	Overlap of cut images for clarity evaluation.
Dataset DataType: 2D array	**Description**
ScopeStitchQCMatrix	Matrix of overlap deviation of each microscope FOV.
CrossPoints/row_col*n	Group of datasets for each FOV that records the track cross point coordinates. (Row and col stand for the FOV row and column index number, and n stands for number of FOVs). Each dataset is a 2D array records, (x, y) coordinates of track cross points in each FOV.
ClarityArr	Clarity prediction of each cut image.

/<stainType>/Calibration Group records the calibration information (only <protein>_IF).	
Attributes	**Description**
CalibrationQCPassFlag	Whether the calibration QC of IF image is passed.

/<stainType>/Calibration/Scope	
Attributes	**Description**
Confidence	Calibration confidence of microscope-stitched tiled image vs. IF.
OffsetX	Horizontal offset.
OffsetY	Vertical offset.

[optional]<stainType>/Calibration/BGI	
Attributes	**Description**
Confidence	Calibration confidence of BGI-stitched image vs. IF.
OffsetX	Horizontal offset.
OffsetY	Vertical offset.

/<stainType>/Stitch: Group records the stitching information.	
Attributes	**Description**
StitchingScore	Reference score for stitching.
TemplateSource	The reference FOV for deriving the template used for rotating and scaling the microscopic images.
WhichStitch	Stitching method for tiled images, including microscope, template and ripple stitching. Tiled image only.
Dataset DataType: 2D array	**Description**
TemplatePoint	Center coordinates for deriving template lines.
TransformTemplate	Coordinates of template points registered with expression matrix.

/<stainType>/Stitch/BGIStitch: Group records the image stitching information processed by BGI program.	
Attributes	**Description**
StitchedGlobalHeight	Height of stitched tiled images using BGI stitching algorithm. Tiled image only.
StitchedGlobalWidth	Width of stitched tiled images using BGI stitching algorithm. Tiled image only.
Dataset DataType: 2D array	**Description**
StitchedGlobalLoc	Coordinates for the BGI stitched tiled image.

/<stainType>/Stitch/ScopeStitch: Group records the image stitching information processed by microscope imaging software.	
Attributes	**Description**
GlobalHeight	Height of panorama image.
GlobalWidth	Width of panorama image.
Dataset DataType: 2D array	**Description**
GlobalLoc	Coordinates for the stitched tiled image (either program stitched or microscope stitched).
ScopeJitterDiff	Jitter offset of microscope stitching.
ScopeHorizontalJitter	Horizontal jitter offset of microscope stitching. Tiled image only.
ScopeVerticalJitter	Vertical jitter offset of microscope stitching. Tiled image only.

/<stainType>/Stitch/StitchEval: Group records the evaluation result of stitching.	
Attributes	**Description**
MaxDeviation	Maximum stitching deviation. Tiled image only.
Dataset DataType: 2D array	**Description**
GlobalDeviation	Global stitching deviation matrix.
StitchEvalH	Stitching deviation matrix for the horizontal axes.
StitchEvalV	Stitching deviation matrix for the vertical axes.

/<stainType>/TissueSeg: Group records the tissue segmentation information.	
Attributes	**Description**
TissueSegScore	Reference score for tissue segmentation.
TissueSegShape	Image shape for tissue segmentation mask image.
Dataset DataType: 2D array	**Description**
TissueMask	Encoded tissue segmentation mask file (before registration with gene expression matrix).

/<stainType>/TissueSeg/Labeling/<label_name>: Group records the labeling tissue segmentation information, specifically labeled areas.	
Attributes (within Dataset)	**Description**
Color	Color of the labeling area.
Description	Description for the labeling area.
Dataset DataType: 2D array	**Description**
Canvas	Serialization string of canvas.

/<stainType>/TissueSeg/Labeling/<label_name>/Element_<num>: Group records the detailed information of each labeling element.	
Attributes (within Dataset)	**Description**
Shape	Shape of the labeling area.
Offset	Offset of the labeling area.
Dataset DataType: 2D array	**Description**
LabelMask	Encoded label mask file.

/<stainType>/CellSeg: Group records the cell segmentation information.	
Attributes	**Description**
CellSegShape	Reference score for cell segmentation.
Dataset DataType: 2D array	**Description**
CellMask	Encoded cell segmentation mask file (before register with gene expression matrix).
CellSegTrace	Cell contour attributes, including height, width and area.

/<stainType>/Register: Group records the information that align images with gene expression matrix.	
Attributes	**Description**
CounterRot90	Count of counter-clockwise rotation of 90 degrees.
Flip	Whether horizontally flipped or not.
MatrixShape	Height and width of the gene expression matrix.
OffsetX	Offset between microscope image and gene expression matrix in x-axis.
OffsetY	Offset between microscope image and gene expression matrix in y-axis.
RegistrationScore	Reference score for registration.
Rotation	Rotation degree between raw image and deviation template.
ScaleX	Scale between raw image and deviation template in horizontal direction.
ScaleY	Scale between raw image and deviation template in vertical direction.
XStart	Gene expression matrix offset x (GEF geneExp/binN/expression attribute minX).
YStart	Gene expression matrix offset y (GEF geneExp/binN/expression attribute minY).
ManualRotation	Manual rotation degree of the raw image around the center point.
ManualScaleX	Manual scale of the raw image in horizontal direction based on center point.
ManualScaleY	Manual scale of the raw image in vertical direction based on center point.
Dataset DataType: 2D array	**Description**
MatrixTemplate	List of track cross points derived from gene expression matrix.

/StereoResepSwitch: Group stores the state of each module that whether the module need to be performed.	
Attributes	**Description**
stitch	Switch for performing stitching.
tissueseg	Switch for performing tissue segmentation.
cellseg	Switch for performing cell segmentation.
register	Switch for performing registration.

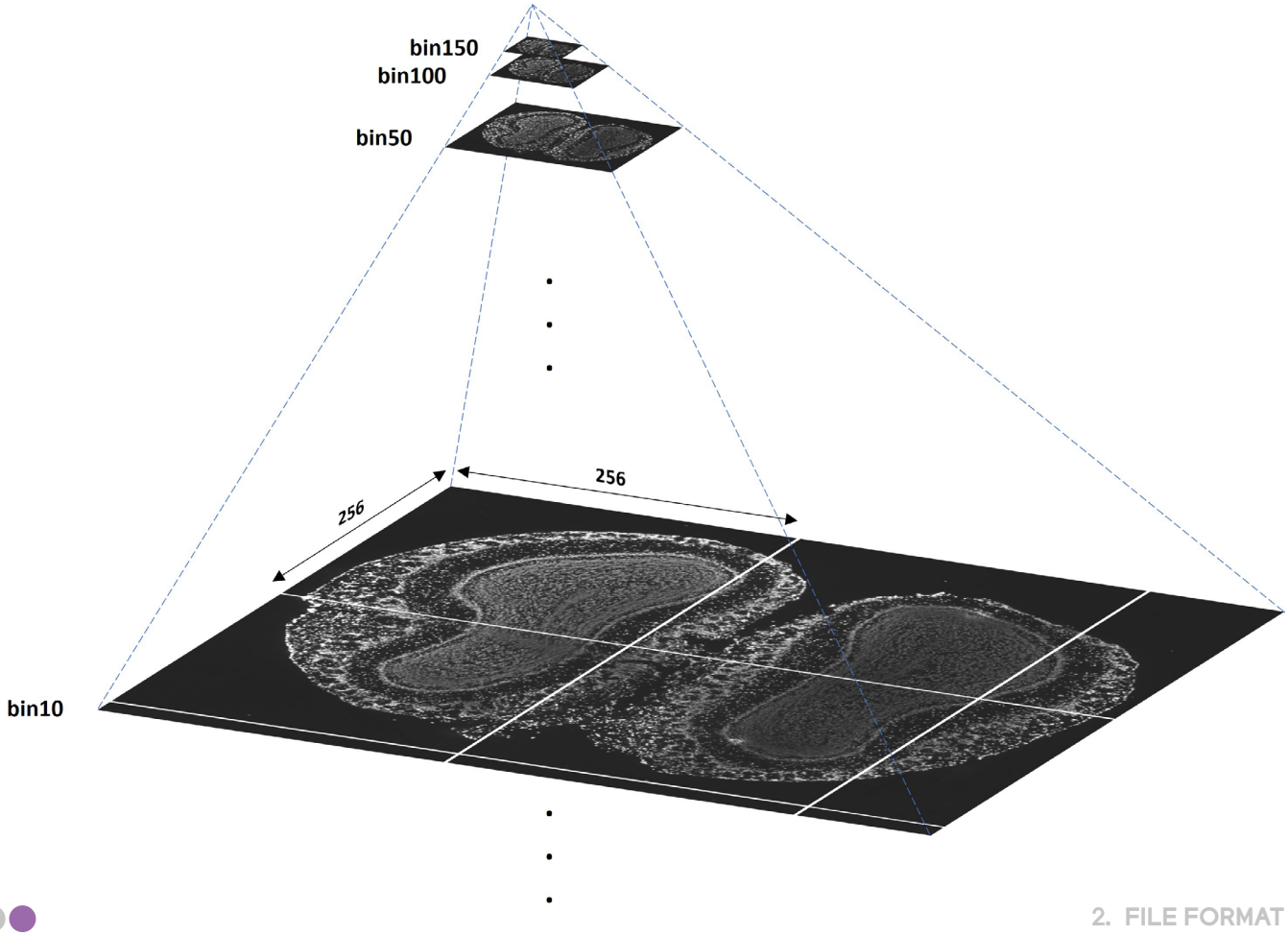
/ManualState: Group stores the state of each module that whether the module has been manually processed.	
Attributes	**Description**
stitch	Whether manually stitched the tiled images.
tissueseg	Whether manually delineated the tissue coverage region.
cellseg	Whether manually delineated the cell coverage regions.
register	Whether manually aligned microscope image and gene expression matrix.
calibration	Whether manually calibrated two images. For example, manually match IF image with DAPI image.

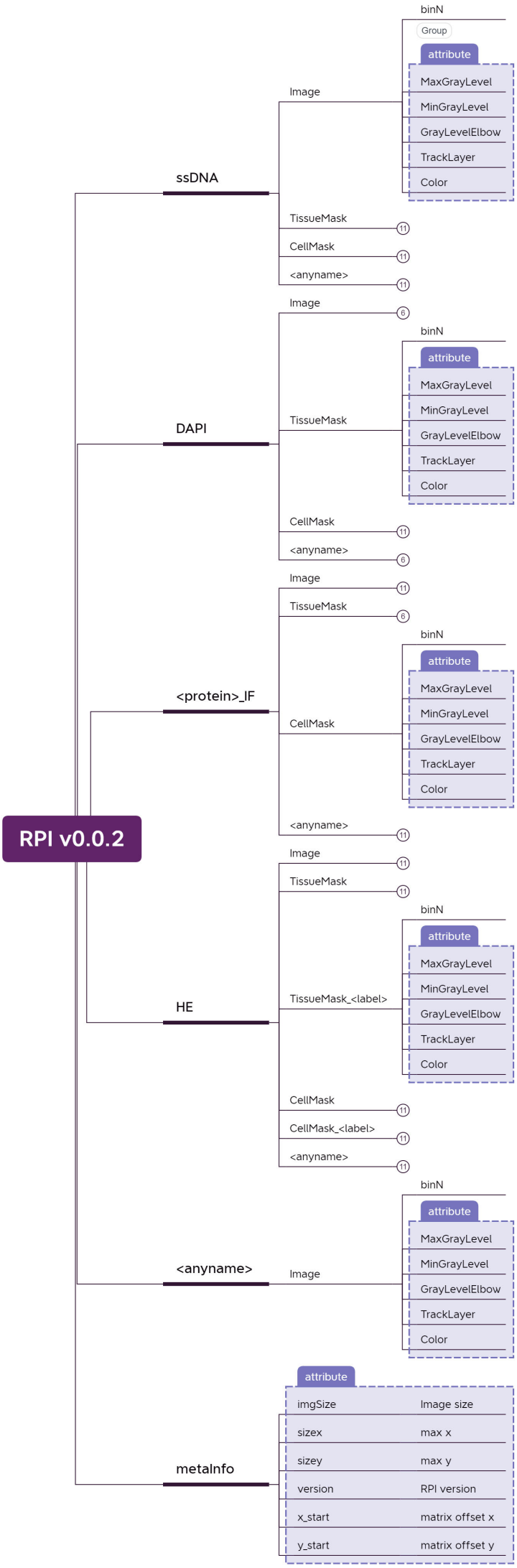
2.7 Image Pyramid

The image pyramid model is a multi-resolution hierarchical model that is used to store and display images in different resolutions. For the same field of view, the layer of the image pyramid that is closest to the bottom includes the most detailed information and has the largest scale. `register` pipeline performs the down-sampling step on the registered image, and the resulted images are layered to construct a pyramid with the suffix ".rpi". For each resolution layer, the intact registered image is split into 256 pixels x 256 pixels tiles. If the size of a layer is smaller than 256 x 256, the image will then remain intact.

The outer layer group of RPI file is defined according to its stain type, generally including ssDNA, DAPI, IF (immunofluorescence image especially for protein, group name as <protein_IF>), and HE. In each group of stain types, multiple image results, including an image subgroup (registered microscopy image), a TissueMask subgroup (registered mask boundary for the tissue coverage area), and a CellMask subgroup (registered mask boundaries for the cell coverage area, optional), could be saved respectively.

Schematic diagram of image pyramid:




RPI v0.0.2
ssDNA
Image
binN
Group
attribute
MaxGrayLevel
MinGrayLevel
GrayLevelElbow
TrackLayer
Color
TissueMask
CellMask
<anyname>
DAPI
Image
TissueMask
CellMask
<anyname>
<protein>_IF
Image
TissueMask
CellMask
<anyname>
HE
Image
TissueMask
TissueMask_<label>
CellMask
CellMask_<label>
<anyname>
<anyname>
Image
metaInfo
attribute
imgSize
Image size
sizex
max x
sizey
max y
version
RPI version
x_start
matrix offset x
y_start
matrix offset y

References

1. STOmics/SAW. Accessed April 17, 2023. https://github.com/STOmics/SAW
2. Sequence Alignment/Map Format Specification. Accessed May 21, 2021. https://github.com/samtools/hts-specs.

Contact Us

Beijing Genomics Institute (Shenzhen)

https://en.stomics.tech

Email: info_global@stomics.tech

Please raise GitHub issues for reporting bugs and requesting features:
SAW GitHub Issue Page: https://github.com/STOmics/SAW/issues