STOmics

Stereo-seq ANALYSIS WORKFLOW FILE FORMAT MANUAL

Software Version: V5.5.0

Manual Version: A2

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:	Add cell bin GEF file demo and format specification; Add IPR file format specification.

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

Manual Version:	A2
Software Version:	V5.5.0
Date:	Jan. 2023
Description:	Update visit links for schematic diagram of GEF and IPR.

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/BGIResearch/SAW**

1.2. About Manual

This manual includes descriptions of key files format 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 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.
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
```

2.2 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
```

2.3 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.

2.3.1 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/BGIResearch/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.
/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	**tissueCut**	**SS200000135TL_ D1.gef**	`tissueCut` 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.

2.3.2 Cell Bin GEF

The first layer of Cell Bin GEF contains one required group “cellBin” and multiple optional datasets. “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/BGIResearch/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.
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.

/cellBin/cell:
Dataset "cell" is a 1D array which stores basic information and indices information of cells and expression.

Dataset Attributes	DataType	Example	Description
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 dataset.
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, and block count in y-axis, respectively.

2.4 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 show 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 2-4 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".
#BinSize	**uint16**	**1**	(Please check 1.3 Terminologies and Concepts Bin)
#STOmicsChip	**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
#BinSize=1
#STOmicsChip=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
```

2.5 Image Process Record File

Image process record (IPR) file is designed to recording the whole-life information of a microscopic staining image from photo-taking to processing. The six basic groups are “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/BGIResearch/SAW/tree/main/Documents/FileFormat** to get the schematic diagram of IPR.

Table 2-6 IPR File Format and Text Fields Description

File Attribute	
Attribute	**Description**
IPRVersion	IPR file format version.

/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.

/ImageInfo: Group records basic image information.	
Attributes	**Description**
Pitch	Physical pitch (nm) between neighbor spots.
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 for the color balance of the lighting.
Dataset DataType: 1D array	**Description**
RGBScale	RGB color.

/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	The staining type.
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).
Dataset DataType: 2D array	**Description**
/QCInfo/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 track cross point coordinates in the FOV, [x, y, row, col].

/Stitch: Group records the stitching information.	
Attributes	**Description**
StitchingScore	Reference score for stitching.
TemplateSource	The reference FOV for deriving the template that used for rotating and scaling the microscopic images.

/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**
/Stitch/BGIStitch/StitchedGlobalLoc	List of coordinates for the BGI stitched tiled image.

/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**
/Stitch/ScopeStitch/GlobalLoc	List of coordinates for the stitched tiled image (either program stitched or microscope stitched).

/Stitch/StitchEval: Group records the evaluation result of stitching.	
Attributes	**Description**
MaxDeviation	Maximum stitching deviation.
Dataset DataType: 2D array	**Description**
/Stitch/StitchEval/GlobalDeviation	Dataset stores the global stitching deviation matrix.
/Stitch/StitchEval/StitchEvalH	Dataset stores the stitching deviation matrix for the horizontal axes.
/Stitch/StitchEval/StitchEvalV	Dataset stores the stitching deviation matrix for the vertical axes.

/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**
/TissueSeg/TissueMask	Encoded tissue segmentation mask file (before register with gene expression matrix).

/CellSeg: Group records the cell segmentation information.	
Attributes	**Description**
CellSegShape	Image shape for cell segmentation mask image.
Dataset DataType: 2D array	**Description**
/CellSeg/CellMask	Encoded cell segmentation mask file (before register with gene expression matrix).

/Register: Group records the information that align images with gene expression matrix.	
Attributes	**Description**
CounterRot90	Count of counter clockwise rotation of 90 degree.
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).
Dataset DataType: 2D array	**Description**
/Register/MatrixTemplate	List of track cross point derived from gene expression matrix.

/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.

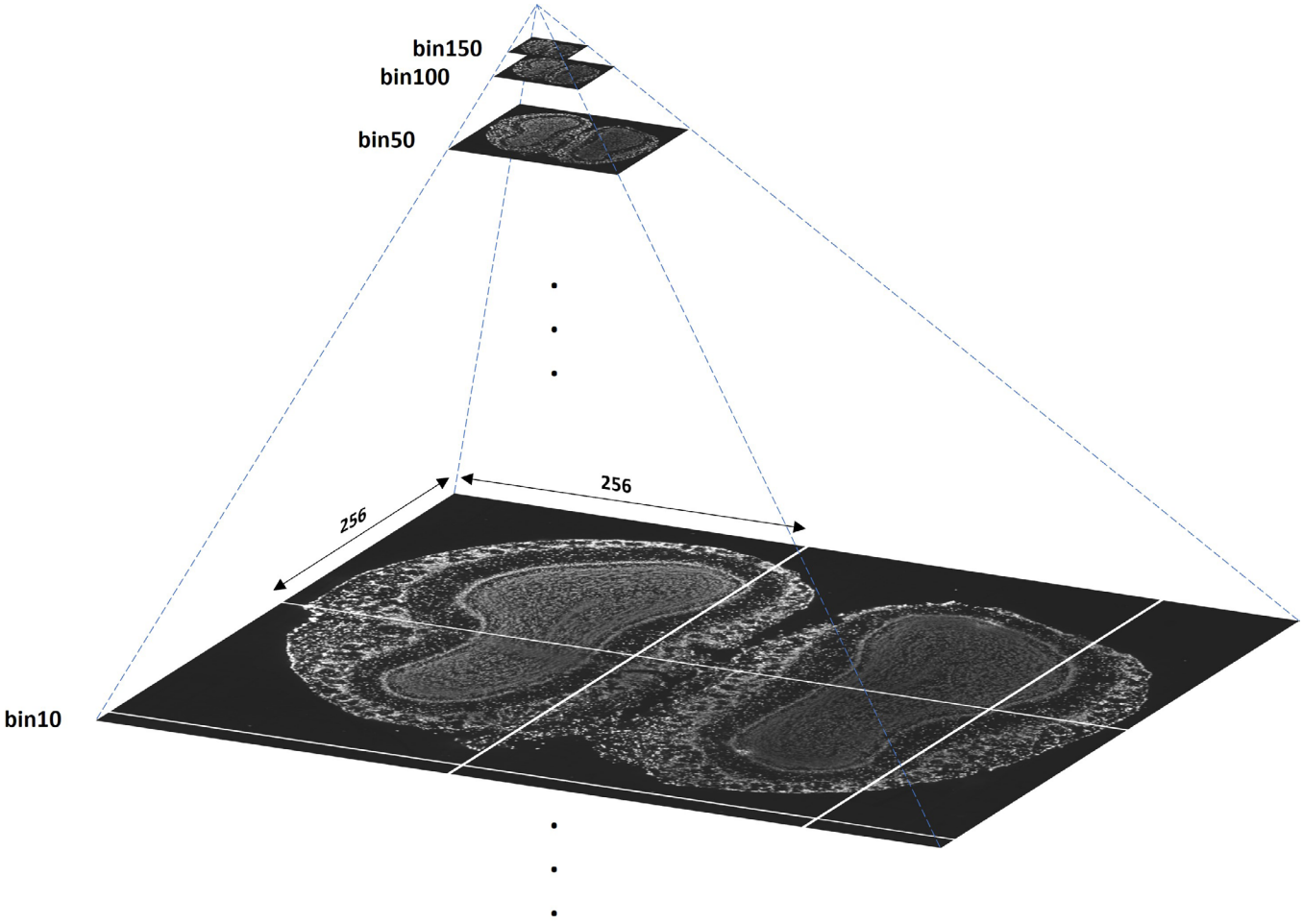
/StereoResepSwitch: Group stores the state of each module that whether the module need to be performed.	
Attributes	**Description**
stitch	Switch for performing stitching.
tissueseg	Switch for performing tissue segmentation.
cellseg	Switch for performing cell segmentation.
register	Switch for performing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

2.6 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.

A standard RPI file usually includes a ssDNA group (registered ssDNA image), a TissueMask group (registered mask boundary for the tissue coverage area), and a CellMask group (registered mask boundaries for the cell coverage area, optional).

Schematic diagram of image pyramid:



References

1. BGIResearch/SAW. Accessed October 13, 2021. https://github.com/BGIResearch/SAW
2. *Sequence Alignment/Map Format Specification.*; 2021. Accessed May 21, 2021. https://github.com/samtools/hts-specs.

Contact Us

BGI-Research, Shenzhen

https://www.stomics.tech/EN/

Email: support@stereomics.com

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