Sapphire manual

1. Introduction

This document provides a manual of the software named Sapphire which is a part of DIAMonDs (Drosophila Individual Activity Monitoring and Detection System).
The software is a web application working on Dash framework in Python.
Installation and usage of Sapphire is described as follows.

2. System requirement

- Any OS (Ubuntu 16.04/18.04, OS X 10.11 El Capitan, Windows 7 are validated)
- Google chrome (web browser)
- Python 3.6

3. Installation and Usage

3-1. Python environment

3-1-1. Installation of Python
3-1-2. Creation of virtual environment by Anaconda
3-1-3. Installation of modules
3-1-4. Module dependencies

3-1. Python environment

Sapphire requires Python as base environment. Some operation systems such as Ubuntu and MacOSX includes Python, however, Anaconda or Miniconda is recommended because it provides convenience in creation and management of multiple virtual environments.

3-1-1. Installation of Python

Python could be automatically installed by installation of Anaconda or Miniconda.

Anaconda:
http://www.anaconda.com

Miniconda:
https://docs.conda.io/en/latest/miniconda.html

3-1-2. Creation of virtual environment by Anaconda

To create a new environment, you can type at the command in Terminal:

```
conda create -n sapphire python=3.6
```

The command specify that the name of the environment is `sapphire` and the version of Python is 3.6. The latest version of Sapphire requires Python 3.6 or later version.

You can activate the environment by typing:

```
source activate sapphire
```

Then, your terminal prompt shows like this:

```
(sapphire) your_machine:your_directory username$
```

3-1-3. Installation of modules

Modules are installed by `conda install` command.

(Example 1) Install of Numpy

```
conda install numpy
```

(Example 2) Install of Numpy and Scipy (multiple install)

```
conda install numpy scipy
```

(Example 3) Install of Numpy ver. 1.15.0 (version specification)

```
conda install numpy==1.15.0
```

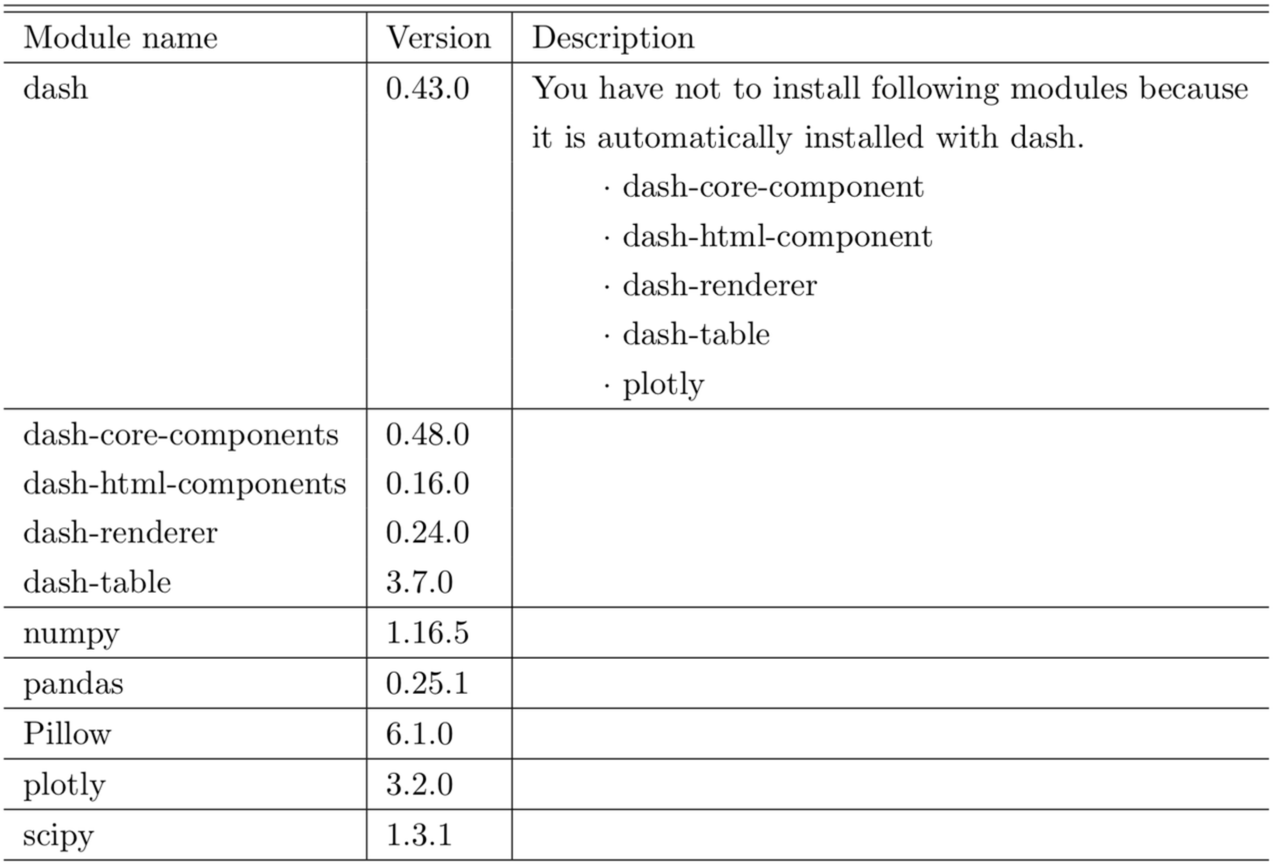
(Example 4) Installation of `dash` (and related modules).
Execute the install command specifying the version:

```
conda install -c conda-forge dash==0.43.0
```

3-1-4. Module dependencies

The application requires Python 3.6.
It also needs Python modules as described bellow.

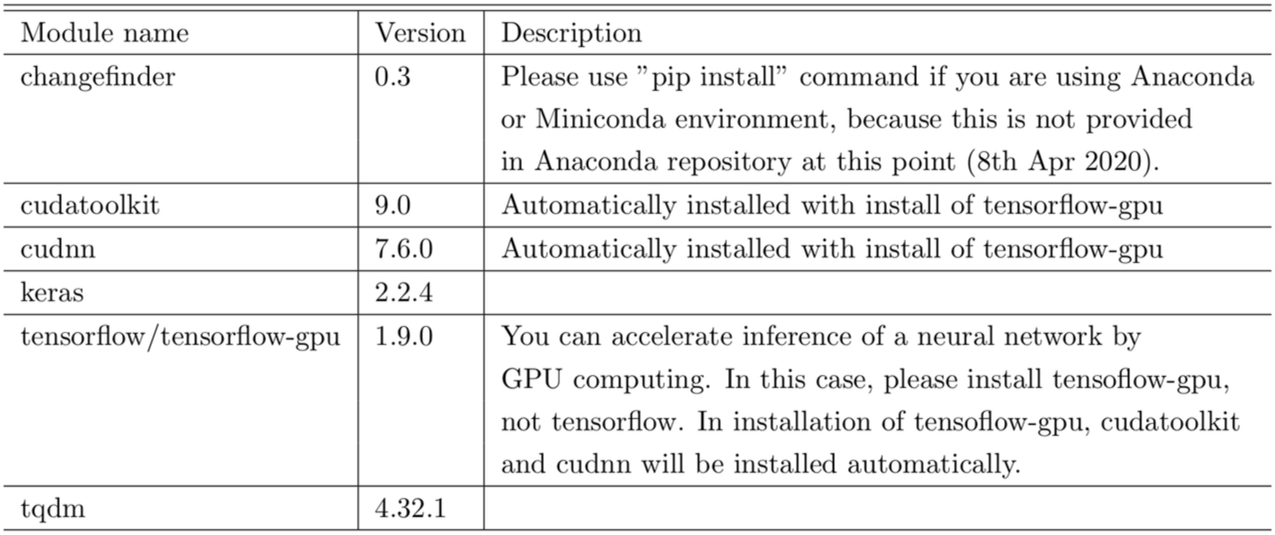
Following minimum modules are required if you want to use minimum functions of Sapphire such as viewing images, signals, etc (Table 1).



Module name	Version	Description
dash	0.43.0	You have not to install following modules because it is automatically installed with dash. · dash-core-component · dash-html-component · dash-renderer · dash-table · plotly
dash-core-components	0.48.0	
dash-html-components	0.16.0	
dash-renderer	0.24.0	
dash-table	3.7.0	
numpy	1.16.5	
pandas	0.25.1	
Pillow	6.1.0	
plotly	3.2.0	
scipy	1.3.1	

Table 1. Minimum module requirement of Sapphire

Additional modules are required if you want to advanced functions of Sapphire such as the inference and/or further analysis with the scripts in this repository (Table 2).



Module name	Version	Description
changefinder	0.3	Please use "pip install" command if you are using Anaconda or Miniconda environment, because this is not provided in Anaconda repository at this point (8th Apr 2020).
cudatoolkit	9.0	Automatically installed with install of tensorflow-gpu
cudnn	7.6.0	Automatically installed with install of tensorflow-gpu
keras	2.2.4	
tensorflow/tensorflow-gpu	1.9.0	You can accelerate inference of a neural network by GPU computing. In this case, please install tensoflow-gpu, not tensorflow. In installation of tensoflow-gpu, cudatoolkit and cudnn will be installed automatically.
tqdm	4.32.1	

Table 2. Additional module requirement for advanced operation of Sapphire

3-2. Data preparation

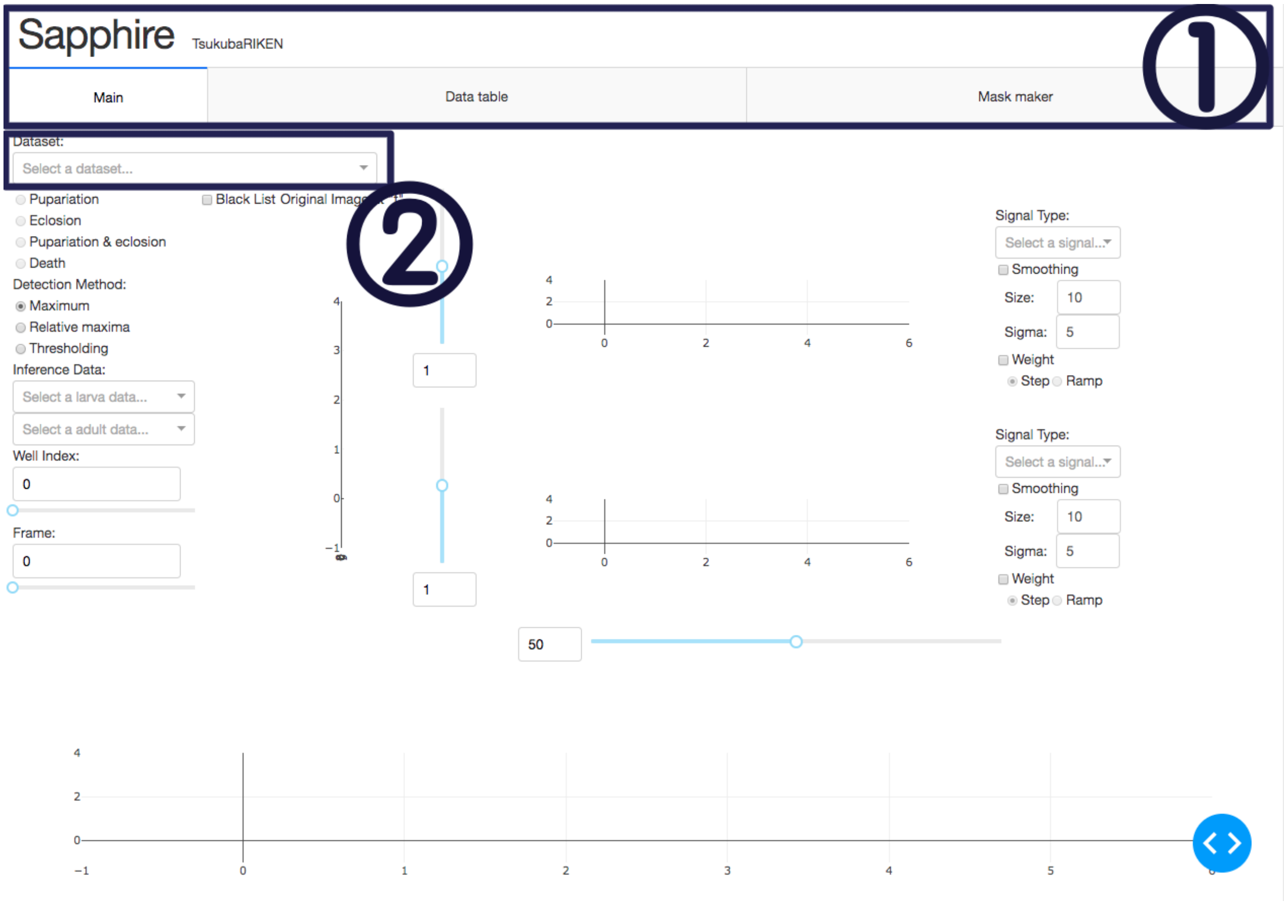
3-2-1. Original images

Sequential raw images should be gray-scale and individual well including animals also should be under 56 x 56 pixel size.

3-2-2. 'Mask maker': making of mask file

Mask file should be made for access to individual well. Sapphire provides semi-automatic mask creation whose procedure is as follows:

(A) Target dataset selection
(B) Well selection
(C) Mask parameter tuning
(D) Conformation and save a proper parameters



Dataset:

○ Pupariation
○ Eclosion
○ Pupariation & eclosion
○ Death

□ Black List Original Image

Detection Method:

● Maximum
○ Relative maxima
○ Thresholding

Inference Data:

Well Index: 0

Frame: 0

1

1

Signal Type:

□ Smoothing

Size: 10

Sigma: 5

□ Weight

● Step ○ Ramp

Signal Type:

□ Smoothing

Size: 10

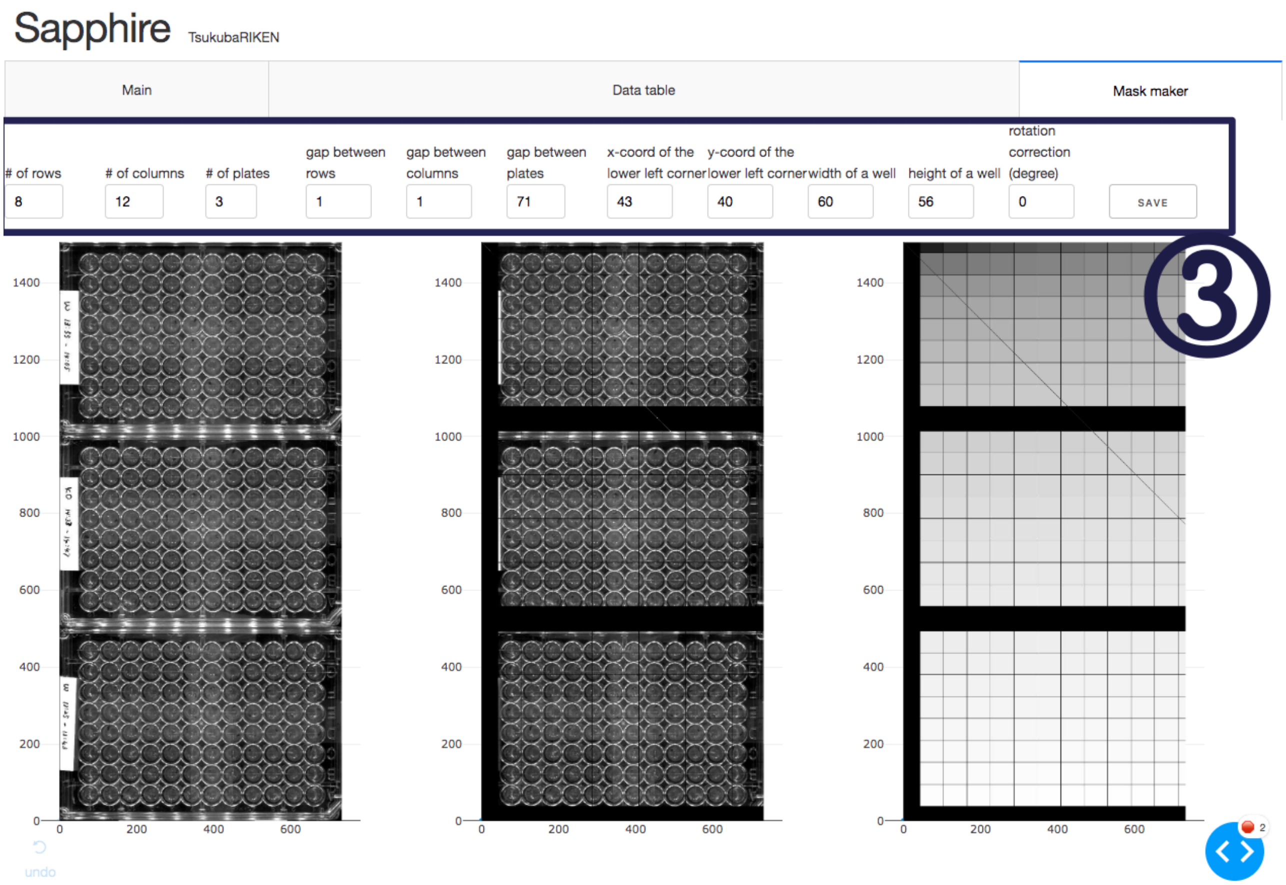
Sigma: 5

□ Weight

● Step ○ Ramp

50

Fig 1. Layout of 'Main' tab



Sapphire TsukubaRIKEN

Main | Data table | Mask maker

# of rows	# of columns	# of plates	gap between rows	gap between columns	gap between plates	x-coord of the lower left corner	y-coord of the lower left corner	width of a well	height of a well	rotation correction (degree)	
8	12	3	1	1	71	43	40	60	56	0	SAVE

Fig 2. Layout of 'Mask maker' tab

(A) Target dataset selection

Select target datasets in 'Main' tab, and click 'Mask maker' tab after the selection of dataset.

(B) Well selection

A first image of sequential images is displayed after dataset selection.
Users can choose an arbitral well within a population image and frame on a well as a reference by right click.

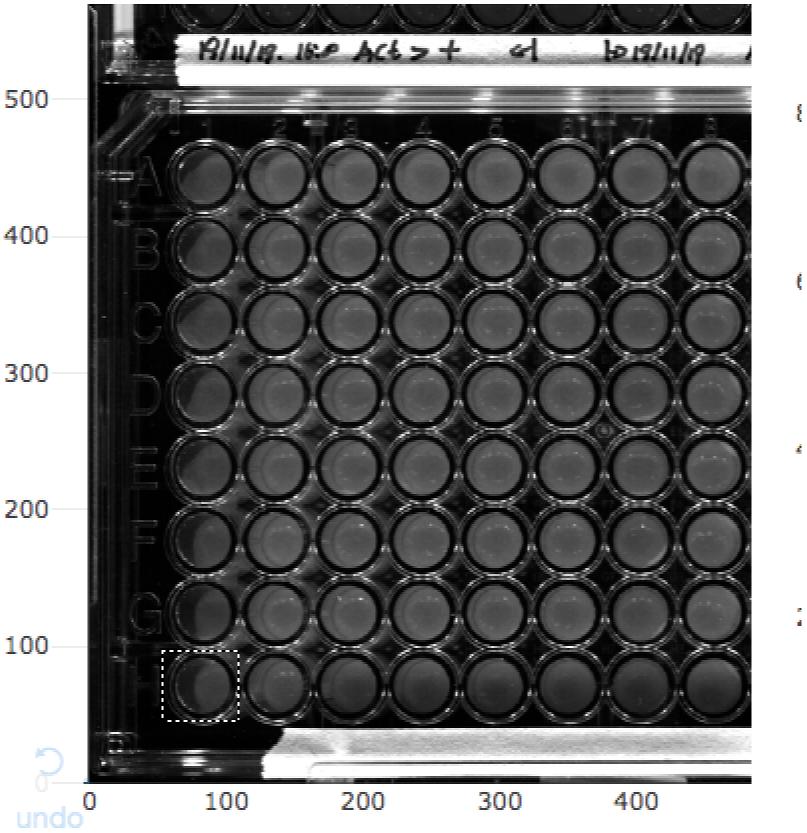
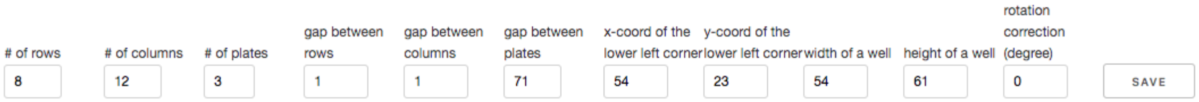


Fig 3. Reference well surrounded by white dots square

(C) Mask parameter tuning

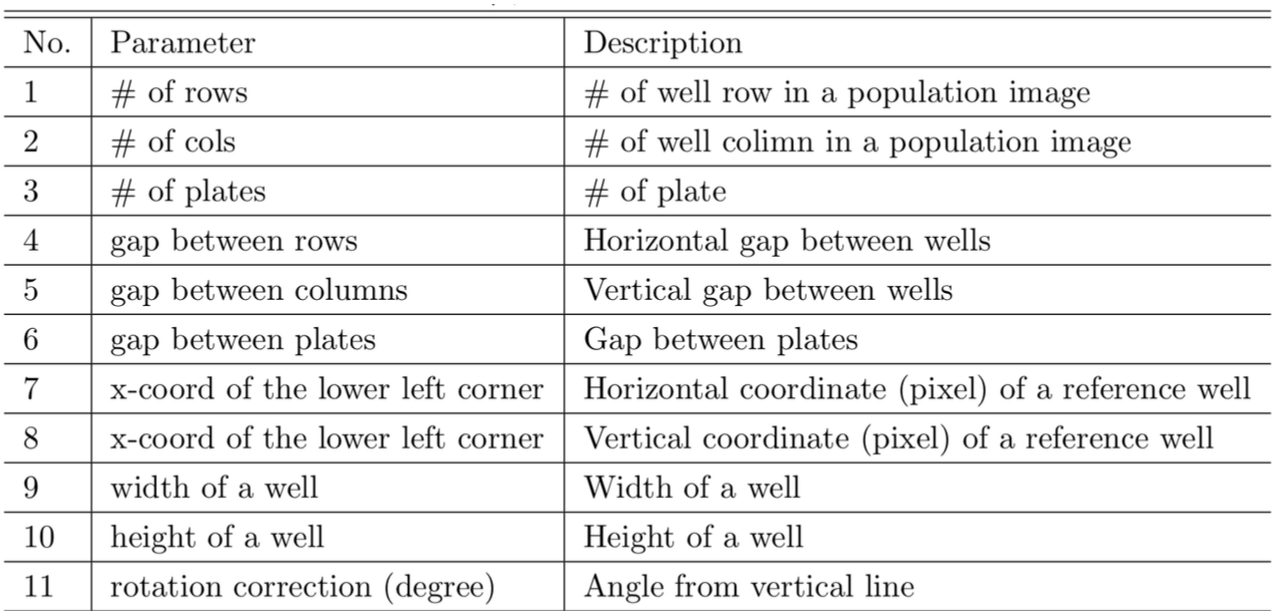
Users can modify the mask parameters with monitoring a generated mask image (Fig. 4)



# of rows	# of columns	# of plates	gap between rows	gap between columns	gap between plates	x-coord of the lower left corner	y-coord of the lower left corner	width of a well	height of a well	rotation correction (degree)
8	12	3	1	1	71	54	23	54	61	0

Fig 4. Parameters for mask creation

Each parameter descriptions are as follows:



No.	Parameter	Description
1	# of rows	# of well row in a population image
2	# of cols	# of well colimn in a population image
3	# of plates	# of plate
4	gap between rows	Horizontal gap between wells
5	gap between columns	Vertical gap between wells
6	gap between plates	Gap between plates
7	x-coord of the lower left corner	Horizontal coordinate (pixel) of a reference well
8	x-coord of the lower left corner	Vertical coordinate (pixel) of a reference well
9	width of a well	Width of a well
10	height of a well	Height of a well
11	rotation correction (degree)	Angle from vertical line

Table 3. Parameters for mask creation

(D) Confirmation and save a proper parameters

If mask parameters were proper and all the wells were correctly surrounded by square-shape reference box, save the parameters by press SAVE button.

3-2-3. Data directory structure

Sapphire loads data such as config files, original images, etc. from dataset directory tree. The directory tree is rigorously defined as follow. Note that the files/directories beginning with asterisk can be arbitrarily named, although the others should have the names exactly same as given below.

```
*data_root_for_demo
├── *dataset1
│   ├── blacklist.csv
│   ├── config.json
│   ├── grouping.csv
│   ├── inference
│   │   ├── adult
│   │   │   └── *profile1
│   │   │       ├── *cf_r0.003_signals.npy
│   │   │       ├── probs
│   │   │       │   ├── 000.npz
│   │   │       │   ├── 001.npz
│   │   │       │   └── 002.npz
│   │   │       ├── probs.npz
│   │   │       └── *signals.npy
│   │   └── larva
│   │       └── *profile1
│   │           ├── probs
│   │           │   ├── 000.npz
│   │           │   ├── 001.npz
│   │           │   └── 002.npz
│   │           ├── probs.npz
│   │           └── *signals.npy
│   ├── mask.npy
│   ├── mask_params.json
│   ├── *network
│   │   ├── adult
│   │   │   ├── *profile1
│   │   │   └── *profile2
│   │   └── larva
│   │       ├── *profile1
│   │       └── *profile2
│   └── original
│       ├── 0001.jpg
│       ├── 0002.jpg
│       ├── 0003.jpg
│       ├── 0004.jpg
│       ├── 0005.jpg
│       ├── eclosion.csv
│       └── pupariation.csv
└── *dataset2
```

- `dataset1`, `dataset2`: Dataset directory.
- `blacklist.csv`: written in CSV (comma separated value) format.
- `config.json`: Configuration file for a dataset.
- `grouping.csv`: Defines groups of flies (CSV format).
- `inference`: Directory for storing results of inference by a trained neural network.
- `inference/adult` or `inference/larva`: Stores inference results for adult/larva flies.
- `inference/*/profile1`: The name of training profile indicating which trained network is used for the inference. The directory name is same as `network/*/profile1`.
- `inference/*/cf_r0.003_signals.npy`: ChangeFinder signal.
- `inference/*/probs`: Stores inference results of each fly in Numpy archive format. The number in file names indicates fly ID.
- `inference/*/probs.npz`: Numpy archive including inference results of all the flies.
- `inference/*/signals.npy`: Label diference signal.
- `mask.npy`: Definition of pixels of each fly in an original image. You can create this file with Sapphire's mask maker tab.
- `mask_params.json`: Parameters for creating the mask. You can create this file with Sapphire's mask maker tab.
- `network`: Stores neural networks trained for semantic segmentation.
- `network/adult` or `network/larva`: Stores networks for adult/larva flies.
- `network/*/profile1`: The name of training profile.
- `original`: Stores original images.

3-3. Automatic detection

3-3-1. Animal body segmentation (inference by trained neural network)

To perform animal body inference by trained network, you can type as follows:

```
python inference.py PATH_OF_TARGET_DATASET FILE_OF_TRAINED_NETWORK
```

3-3-2. Getting change point signal

You can get ChangeFinder signal by running make_CF_signals.py script. The script requires signals.py file which has signal data calculated by sequential segmentation data.

```
python make_CF_signals.py PATH/signal.py TARGET_EVENT
```

TARGET_EVENT takes an event name ‘pupariation’, ‘eclosion’, or ‘death’

The script also accepts the parameter of ChangeFinder, r.

(Example) Calculation with r=0.009

```
python make_CF_signals.py PATH/signal.py TARGET_EVENT -r0.009
```

Note that r should take between zero and one.

3-4. Launching of Sapphire as a viewer

Sapphire can be launched in arbitral web browser.
The web application has three tabs.

'Main' tab

- Viewing original image data
- Viewing inference data
- Visualization of individual activity signals and summary data

'Data table' tab

- Browsing all the event timing obtained by automatic detection algorithm, and exporting the result as csv file.

'Mask maker' tab

- Making mask file

3-4-1. Usage

You can launch Sapphire by typing:

```
python sapphire.py
```

Sapphire is a web application, so you can open Sapphire in your web browser by accessing `localhost:8050` (Fig 4).

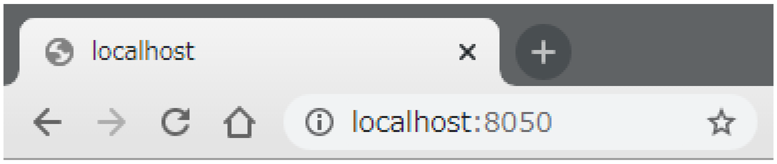
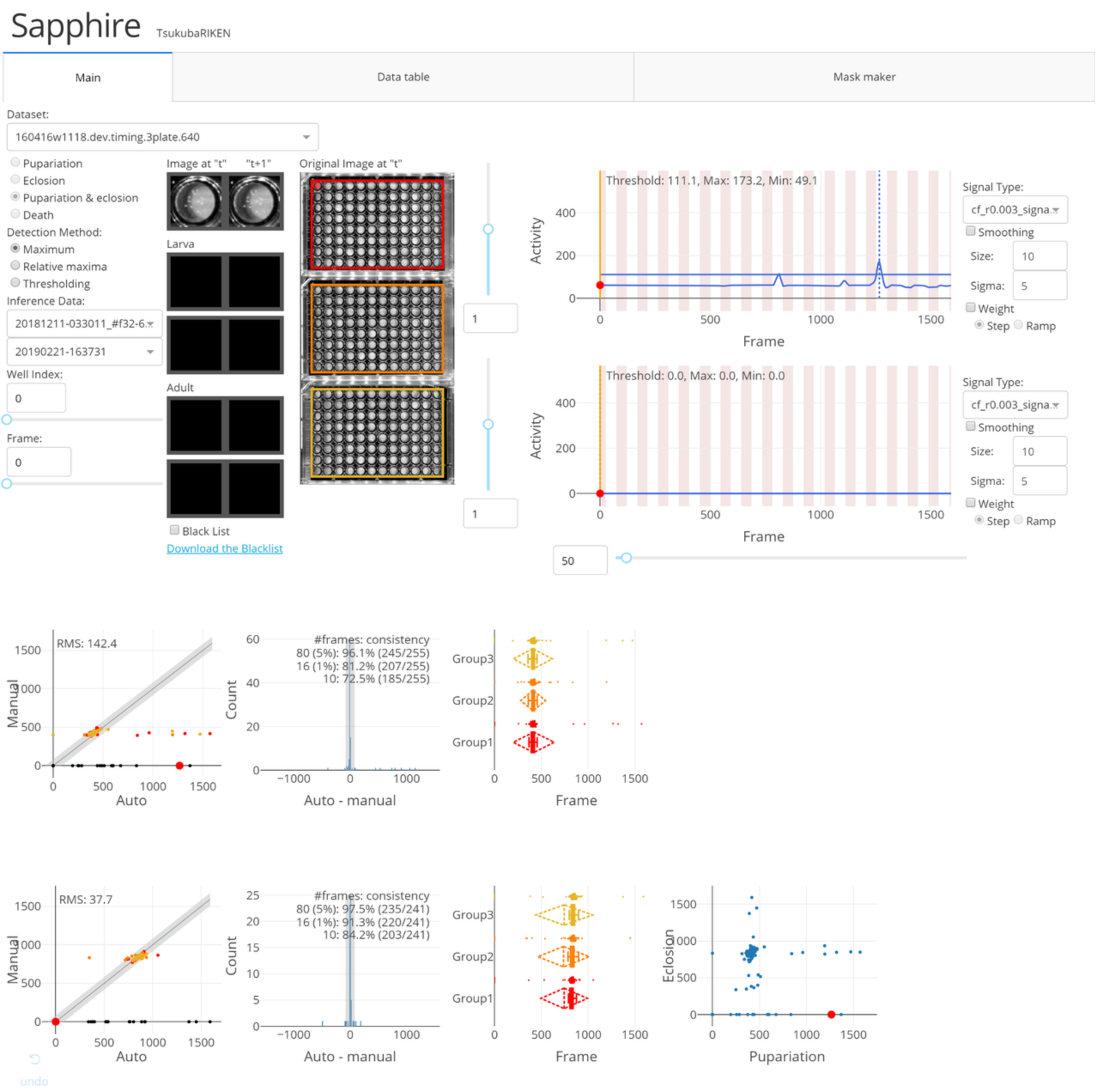


Fig 4. URL to Sapphire

3-4-2. ‘Main’ tab

Original images, inference results, summary could be visualized in ‘Main’ tab (Fig 5).



Sapphire TsukubaRIKEN

Main | Data table | Mask maker

Dataset: 160416w1118.dev.timing.3plate.640

○ Pupariation
○ Eclosion
● Pupariation & eclosion
○ Death

Detection Method:
● Maximum
○ Relative maxima
○ Thresholding

Inference Data: 20181211-033011_#f32-6...
20190221-163731

Well Index: 0

Frame: 0

□ Black List
Download the Blacklist

Threshold: 111.1, Max: 173.2, Min: 49.1

Signal Type: cf_r0.003_signa...
□ Smoothing
Size: 10
Sigma: 5
□ Weight
● Step ○ Ramp

Threshold: 0.0, Max: 0.0, Min: 0.0

Signal Type: cf_r0.003_signa...
□ Smoothing
Size: 10
Sigma: 5
□ Weight
● Step ○ Ramp

RMS: 142.4

#frames: consistency
80 (5%): 96.1% (245/255)
16 (1%): 81.2% (207/255)
10: 72.5% (185/255)

RMS: 37.7

#frames: consistency
80 (5%): 97.5% (235/241)
16 (1%): 91.3% (220/241)
10: 84.2% (203/241)

Fig 5. Screenshot of ‘Main’ tab

3-4-3. Dataset and target event

By the pull-down list at left top in ‘Main’ tab, users can choose the dataset to be visualized. A dataset directory includes a configuration file (`config.json`) that defines which type of event should be detected by the system (e.g. pupariation, eclosion, and death). Such target event will be automatically shown in the corresponding radio button.

The current version of Sapphire provides three types of detection methods. Maximum, relative maxima, and simple thresholding of ChageFinder signal can be chosen at the radio button named ‘Detection method’.

3-4-4. Inference data

By pull-down list in ‘Inference data’, users could select an inference data if they had several inference data obtained by various methods or conditions.

3-4-5. Visualization of well example

Individual well image and inference result at specific time step (*t*) and consecutive time step (*t*+1) are displayed (Fig 6). The well example can be selected by textbox named ‘Well index’ and sliderbar under the textbox, or direct clicking of target well on a population image.

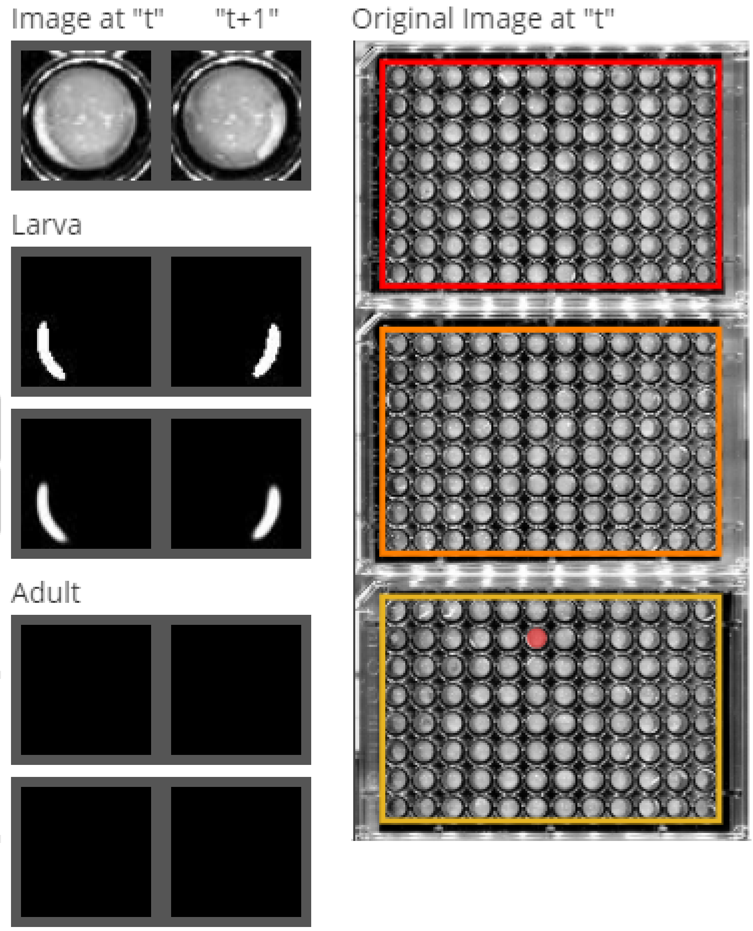
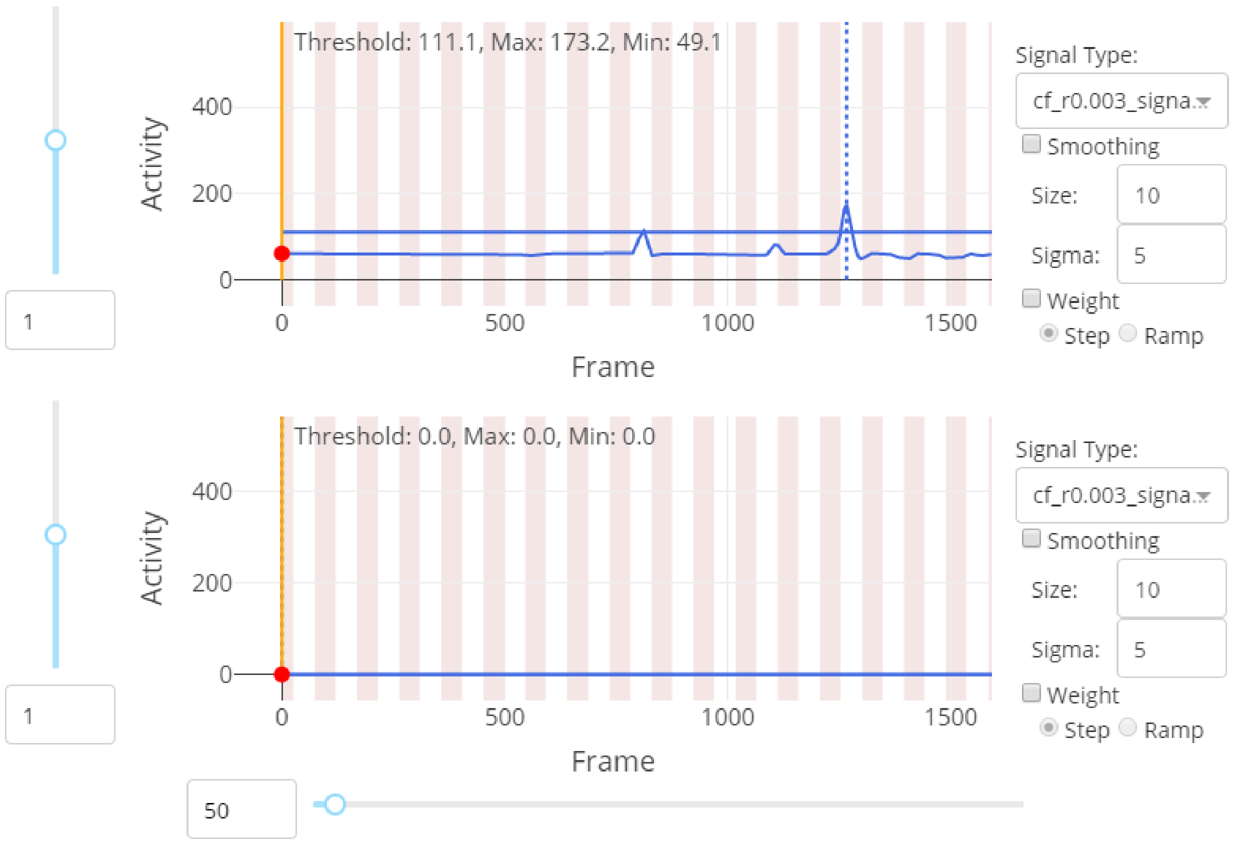


Fig 6. Well example

Time step *t* is also modifiable by textbox named 'Frame' and sliderbar under the textbox, or direct clicking on time series of activity signal desplayed in right top panel. The time series data will be explained in following section.

3-4-6. Visualization of time series data

Sapphire can visualize several types of time series data (Fig 7). The signal users would like to show can be selected by pull-down list named 'Signal Type'. Visualized time series data are modifiable by 'smoothing' checkbox and textbox defining the smoothing parameters. The signals can be weighted by step or ramp function. Usually, upper panel describes the signal obtained from larva data and lower panel shows the signal from adult animal.



1

Threshold: 111.1, Max: 173.2, Min: 49.1

Signal Type: cf_r0.003_signa...

□ Smoothing

Size: 10

Sigma: 5

□ Weight

● Step ○ Ramp

1

Threshold: 0.0, Max: 0.0, Min: 0.0

Signal Type: cf_r0.003_signa...

□ Smoothing

Size: 10

Sigma: 5

□ Weight

● Step ○ Ramp

50

Fig 7. Time series data

A vertical orange line indicates in the graph is a timing of manual detection if it was loaded. A horizontal blue line describes a threshold value which is manually modifiable by textbox at left side and sliderbar above it. A vertical blue dot line corresponds to a timing of automatic detection by the system.

3-4-7. Visualization of summary data

Sapphire provides summary visualization by several styles (Fig 8).
Scatter plot describes a comparison between automatic- and manual-detections (left panel). A colored dot corresponds to an individual well and the black line is $y = x$.

The gray region indicates 5% consistency between auto and manual for entire frames. The frame difference between auto and manual is displayed as a histogram (middle panel). Summary for individual group, defined by user via `group.csv`, is shown as box plots (right panel).

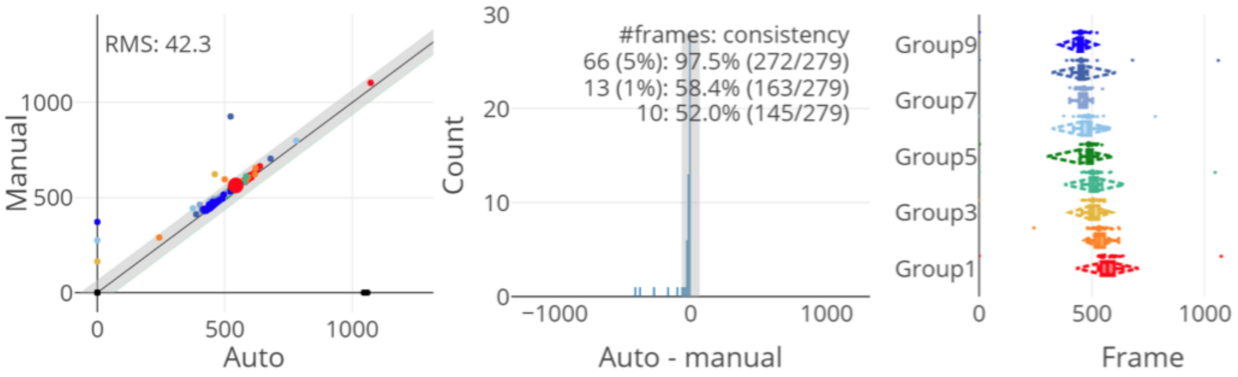
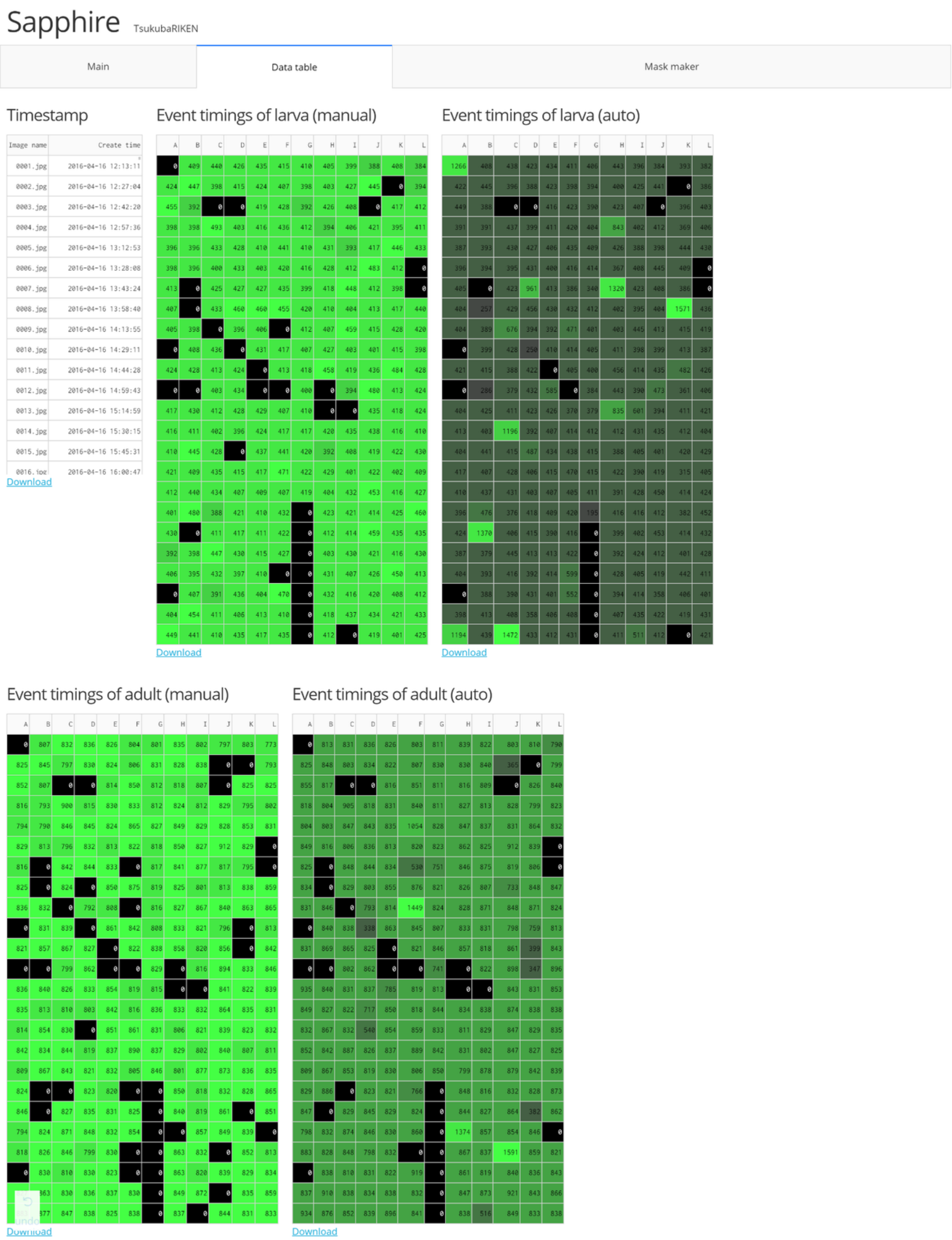


Fig 8. Comparison between automatic- and manual-detections

3-4-8. ‘Data table’ tab

This tab provides some table data (Fig 9). The ‘Timestamp’ table shows the correspondence of an image file name to the captured date and time. ‘Event timings’ tables exhibit animal event (pupariation, eclosion, and death) timings evaluated by manual and auto. You can get CSV files of the tables by clicking ‘download’ link.



Timestamp

Image name	Create time
0001.jpg	2016-04-16 12:13:11
0002.jpg	2016-04-16 12:27:04
0003.jpg	2016-04-16 12:42:20
0004.jpg	2016-04-16 12:57:36
0005.jpg	2016-04-16 13:12:53
0006.jpg	2016-04-16 13:28:08
0007.jpg	2016-04-16 13:43:24
0008.jpg	2016-04-16 13:58:40
0009.jpg	2016-04-16 14:13:55
0010.jpg	2016-04-16 14:29:11
0011.jpg	2016-04-16 14:44:28
0012.jpg	2016-04-16 14:59:43
0013.jpg	2016-04-16 15:14:59
0014.jpg	2016-04-16 15:30:15
0015.jpg	2016-04-16 15:45:31
0016.jpg	2016-04-16 16:00:47

Event timings of larva (manual)

A	B	C	D	E	F	G	H	I	J	K	L
0	409	440	426	435	415	410	405	399	388	408	384
424	447	398	415	424	407	398	403	427	445	0	394
455	392	0	0	419	428	392	426	408	0	417	412
398	398	493	403	416	436	412	394	406	421	395	411
396	396	433	428	410	441	410	431	393	417	446	433
398	396	400	433	403	420	416	428	412	483	412	0
413	0	425	427	427	435	399	418	448	412	398	0
407	0	433	460	460	455	420	410	404	413	417	440
405	398	0	396	406	0	412	407	459	415	428	420
0	408	436	0	431	417	407	427	403	401	415	398
424	428	413	424	0	413	418	458	419	436	484	428
0	0	403	434	0	0	400	0	394	480	413	424
417	430	412	428	429	407	410	0	0	435	418	424
416	411	402	396	424	417	417	420	435	438	416	410
410	445	428	0	437	441	420	392	408	419	422	430
421	409	435	415	417	471	422	429	401	422	402	409
412	440	434	407	409	407	419	404	432	453	416	427
401	480	388	421	410	432	0	423	421	414	425	460
430	0	411	417	411	422	0	412	414	459	435	435
392	398	447	430	415	427	0	403	430	421	416	430
406	395	432	397	410	0	0	431	407	426	450	413
0	407	391	436	404	470	0	432	416	420	408	412
404	454	411	406	413	410	0	418	437	434	421	433
449	441	410	435	417	435	0	412	0	419	401	425

Event timings of larva (auto)

A	B	C	D	E	F	G	H	I	J	K	L
1266	408	438	423	434	411	406	443	396	384	393	382
422	445	396	388	423	398	394	400	425	441	0	386
449	388	0	0	416	423	390	423	407	0	396	403
391	391	437	399	411	420	404	843	402	412	369	406
387	393	430	427	406	435	409	426	388	398	444	430
396	394	395	431	400	416	414	367	408	445	409	0
405	0	423	961	413	386	340	1320	423	408	386	0
404	257	429	456	430	432	412	402	395	404	1571	436
404	389	676	394	392	471	401	403	445	413	415	419
0	399	428	250	410	414	405	411	398	399	413	387
421	415	388	422	0	405	400	456	414	435	482	426
0	286	379	432	585	0	384	443	390	473	361	406
404	425	411	423	426	370	379	835	601	394	411	421
413	403	1196	392	407	414	412	412	431	435	412	404
404	441	415	487	434	438	415	388	405	401	420	429
417	407	428	406	415	470	415	422	390	419	315	405
410	437	431	403	407	405	411	391	428	450	414	424
396	476	376	418	409	420	195	416	416	412	382	452
424	1370	406	415	390	416	0	399	402	453	414	432
387	379	445	413	413	422	0	392	424	412	401	428
404	393	416	392	414	599	0	428	405	419	442	411
0	388	390	431	401	552	0	394	414	358	406	401
398	413	408	358	406	408	0	407	435	422	419	431
1194	439	1472	433	412	431	0	411	511	412	0	421

Event timings of adult (manual)

A	B	C	D	E	F	G	H	I	J	K	L
0	807	832	836	826	804	801	835	802	797	803	773
825	845	797	830	824	806	831	828	838	0	0	793
852	807	0	0	814	850	812	818	807	0	825	825
816	793	900	815	830	833	812	824	812	829	795	802
794	790	846	845	824	865	827	849	829	828	853	831
829	813	796	832	813	822	818	850	827	912	829	0
816	0	842	844	833	0	817	841	877	817	795	0
825	0	824	0	850	875	819	825	801	813	838	859
836	832	0	792	808	0	816	827	867	840	863	865
0	831	839	0	861	842	808	833	821	796	0	813
821	857	867	827	0	822	838	858	820	856	0	842
0	0	799	862	0	0	829	0	816	894	833	846
836	840	826	833	854	819	815	0	0	841	822	839
835	813	810	803	842	816	836	833	832	864	835	831
814	854	830	0	851	861	831	806	821	839	823	832
842	834	844	819	837	890	837	829	802	840	807	811
809	867	843	821	832	805	846	801	877	873	836	835
824	0	0	823	820	0	0	850	818	832	828	865
846	0	827	835	831	825	0	840	819	861	0	851
794	824	871	848	832	854	0	0	857	849	839	0
818	826	846	799	830	0	0	863	832	0	852	813
0	830	810	830	823	0	0	863	820	839	829	834
827	863	830	836	837	830	0	849	872	0	835	859
883	877	847	838	825	838	0	837	0	844	831	833

Event timings of adult (auto)

A	B	C	D	E	F	G	H	I	J	K	L
0	813	831	836	826	803	811	839	822	803	810	790
825	848	803	834	822	807	830	830	840	365	0	799
855	817	0	0	816	851	811	816	809	0	826	840
818	804	905	818	831	840	811	827	813	828	799	823
804	803	847	843	835	1054	828	847	837	831	864	832
849	816	806	836	813	820	823	862	825	912	839	0
825	0	848	844	834	530	751	846	875	819	806	0
834	0	829	803	855	876	821	826	807	733	848	847
831	846	0	793	814	1449	824	828	871	848	871	824
0	840	838	338	863	845	807	833	831	798	759	813
831	869	865	825	0	821	846	857	818	861	399	843
0	0	802	862	0	0	741	0	822	898	347	896
935	840	831	837	785	819	813	0	0	843	831	853
849	827	822	717	850	818	844	834	838	874	838	838
832	867	832	540	854	859	833	811	829	847	829	835
852	842	887	826	837	889	842	831	802	847	827	825
809	867	853	819	830	806	850	799	878	879	842	839
829	886	0	823	821	766	0	848	816	832	828	873
847	0	829	845	829	824	0	844	827	864	382	862
798	832	874	846	830	860	0	1374	857	854	846	0
883	828	848	798	832	0	0	867	837	1591	859	821
0	838	810	831	822	919	0	861	819	840	836	843
837	910	838	834	838	832	0	847	873	921	843	866
934	876	852	839	896	841	0	838	516	849	833	838

Fig 9. Summary table in ‘Data table’ tab