

CellH5: An R-package for easy access to ch5-files

December 8, 2015

1 Getting started

The cellh5-source code is available on [github](#). Download and unpack the zip file. For installation follow the instructions in the README.md. After installation load the package:

```
> library(cellh5)
> # to show the help page
> help(CellH5)
```

Open a file (note: the source package does not include demo data).

```
> c5f <- CellH5("../data/_all_positions.ch5")
```

1.1 Demo scripts

There are a some demo scripts in the repository. Each script can be downloaded [here](#).

'demo_cellh5.R'	Shows basic functionality and the use of the introspection methods such C5HasObjects(...) or C5HasTimelapse(...).
'demo_image.R'	Load an image and plot it using the native image(...) method.
'demo_classimage.R'	render an image with color coded object contours and labels.
'demo_gallery.R'	Export Event gallery images to a user defined directory. Script is incredibly slow. Exporting galleries is always slow.

2 Reading basic information

Read out specific information such as plate names, segmentation (channel-regions pairs), positions or timelapse tables.

```
> # segmentation information
> chreg <- C5SegementationMasks(c5f)
> plates <- C5Plates(c5f)
> positions <- C5Positions(c5f, plates[[1]])
> timelapse <- C5Timelapse(positions[[1]])
> # available features for a particular segmentation region (primary__primary)
> clfeatures <- C5FeatureNames(c5f, chreg[[1]])
```

Print data to the console:

```
> head(timelapse)

  frame timestamp_abs timestamp_rel
1     1     1353838842             0
2     2     1353838719            -123
3     3     1353838720            -121
4     4     1353838722            -119
5     5     1353838740            -101
6     6     1353838821            -20
```

```
> # plate names
> print(plates)

[1] "H2b_aTub_MD20x_exp911"
```

2.1 Example: object counts

Read the number of cells per frame and class name:

```
> object_counts <- C5ObjectCounts(c5f, positions$W0_P0013, "primary__primary")
> head(object_counts)
```

	frame	inter	pro	prometa	meta	earlyana	lateana	telo	apo
1	1	37	2	7	0	0	1	3	1
2	2	31	5	6	0	2	0	2	1
3	3	31	4	6	0	3	0	2	0
4	4	32	4	9	0	0	2	0	0
5	5	30	6	9	0	0	1	1	0
6	6	31	4	5	0	2	1	2	1

2.2 Example: read image

Be careful and don't overwrite the built-in function `image`. Set the option `useRaster=TRUE` to increase the performance.

```
> colors = grey.colors(256)
> image_ <- C5ReadImage(c5f, positions$W0_P0013, "primary__primary", frame=1, zstack=1)
> #plot the image
> image(image_, col=colors, axes=FALSE, useRaster=TRUE)
```

2.3 Example: read object details (main features)

In this tutorial we refer as main features the mean intensity, its standard deviation and the size of an object.

```
> object_details <- C5ObjectDetails(c5f, positions$W0_P0013, "primary__primary")
> head(object_details)
```

	frame	obj_id	class_name	class_label	mean	stddev	size
1	1	2945	telo	7	57.67866	22.868649	778
2	1	260	inter	1	17.90005	5.027965	1881
3	1	3462	apo	8	86.54470	55.998842	2304
4	1	2567	prometa	3	80.09554	31.644412	1570
5	1	3083	inter	1	53.55691	10.454321	2047
6	1	2318	inter	1	25.84809	6.302346	3094

3 File introspection

Finally, the method `C5FileInfo` gives you an overview of the file contents. Alternatively you can use [HDFView](#). Some functions start with a prefix "C5Has...". These methods are meant to determine if the hdf5 file contains e.g. Objects, Events, Tracks or if the objects have been classified at all.

```
> if (C5HasEvents(positions$W0_P0013)) {
+   events <- C5Events(c5f, positions$W0_P0013, "primary__primary",
+                     include_branches=TRUE, return_indices=TRUE)
+   event_features <- C5EventFeatures(c5f, positions$W0_P0013,
+                                     "primary__primary", TRUE, c("n2_avg", "roisize"))
+ } else {
+   warning('no events found!')
+ }
```

Although `C5EventFeatures(...)` would return `NULL` if the file contains no events, it is recommended to use these functions to improve code readability.

4 Currently implemented functions

All methods start with the prefix **C5**. For detailed information of each single function type in the *R-terminal* `help(<C5FunctionName>)` or `?<C5FunctionName>`. Below is a list of available functions.

<code>C5BoundingBoxes</code>	read the bounding boxes of the object
<code>C5Center</code>	return the centers of the object
<code>C5ChannelRegions</code>	list of the segmentation regions
<code>C5ClassifierDefinition</code>	read the classifier definition from the file
<code>C5Close</code>	close cellh5 instance and release hdf5 resources
<code>C5ContourImage</code>	plot a image with color coded contours on the screen
<code>C5Contours</code>	read object contours from file
<code>C5EventFeatures</code>	read feature vectors per track found by the event selection
<code>C5Events</code>	read tracks found by the event selection
<code>C5ExportGallery</code>	export gallery images of the track to the hard disk
<code>C5FeatureNames</code>	get a list of the feature names used for a particular classifier
<code>C5FeaturesByName</code>	load features vectors. Features are selected by name
<code>C5FileInfo</code>	print basic information of the content of a cellh5 file
<code>C5GalleryImageByIndex</code>	load a gallery image of an object by index
<code>C5HasClassifiedObjects</code>	returns 'TRUE' if a position contains objects that has been labeled by a classifier
<code>C5HasEvents</code>	test if a position contains events found by event selection
<code>C5HasObjects</code>	test if a position contains objects found by segmentation
<code>C5HasTimelapse</code>	test if a position contains timelapse information
<code>C5ObjectCounts</code>	read out the number of cells per frame and class label
<code>C5ObjectDetails</code>	read detailed object table. Table contains class labels, mean intensity, etc...
<code>C5ObjectLabels</code>	list of all object
<code>C5Orientation</code>	angle and eccentricity of an ellipse fit
<code>C5Plates</code>	list of processed plates names from the file
<code>C5Positions</code>	list of positions for a certain plate
<code>C5PredictionProbabilities</code>	prediction probabilities of a classification (if provided by the classifier)
<code>C5Predictions</code>	read out the predicted class names
<code>C5ReadImage</code>	read an raw image as array
<code>C5TimeIdx</code>	return a list of frame numbers for each object
<code>C5Timelapse</code>	read the timelapse table for a given position