

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

May 19, 2014

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 <- C5ChannelRegions(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      1228404065             0
2     2      1228404358            293
3     3      1228404638            573
4     4      1228404913            848
5     5      1228405191           1126
6     6      1228405468           1403
```

```
> # 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	32	3	4	0	2	0	3	5
2	2	29	3	5	0	2	0	1	5
3	3	28	4	7	0	1	0	1	5
4	4	27	3	7	1	1	0	0	5
5	5	27	5	6	1	2	0	0	4
6	6	30	3	6	0	1	0	0	5

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	2	1	1	17.10562	4.593563	1903
2	1	3	1	1	46.09320	12.703094	1867
3	1	4	1	1	39.76294	10.240670	3092
4	1	5	1	1	18.52007	4.706463	1719
5	1	6	1	1	31.70925	8.513526	1871
6	1	8	1	1	36.03890	8.838551	2108

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 recommend to use these function to improve code readability.

```
> head(C5FileInfo(c5f))
```

```
[1] "File: ../data/_all_positions.ch5"
      name                                     dclass    dim
0  /feature                                   group
1  /feature/primary__primary                 group
2  /feature/primary__primary/bounding_box    COMPOUND  4
3  /feature/primary__primary/center          COMPOUND  2
4  /feature/primary__primary/crack_contour   COMPOUND  1
5  /feature/primary__primary/object_classification group
6  /feature/primary__primary/object_classification/class_labels COMPOUND  8
7  /feature/primary__primary/object_classification/classifier COMPOUND  1
8  /feature/primary__primary/object_classification/features COMPOUND 239
9  /feature/primary__primary/object_features COMPOUND 239
10 /feature/primary__primary/orientation     COMPOUND  2
11 /image                                    group
12 /image/channel                           COMPOUND  1
13 /image/region                           COMPOUND  1
14 /image/time_lapse                        STRING   3
15 /object                                  group
16 /object/event                           COMPOUND  1
17 /object/primary__primary                 COMPOUND  1
18 /object/tracking                         COMPOUND  1

      name    dclass dim
0          /feature    group
1      /feature/primary__primary    group
2      /feature/primary__primary/bounding_box COMPOUND  4
3      /feature/primary__primary/center COMPOUND  2
4      /feature/primary__primary/crack_contour COMPOUND  1
5 /feature/primary__primary/object_classification    group
```

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.

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