Class: Logable — *python class that can create log entries*

Class: Project — *base projects expect input data in the format: CXY*

Class: TimecourseProject — *time course projects expect input data in the format: NCXY*

project classes: *provide code framework to load and save data and to execute different processing steps*

Class: ProcessingStep — *python class that can load a configuration file to read parameters and create a subdirectory under the project for results*

Segmentation

segmentation classes

individual segmentation workflows

Class: Segmentation — subclasses: workflows

Class: ShardedSegmentation — subclasses: workflows

Class: TimecourseSegmentation — subclasses: workflows

Class: MultithreadedSegmentation — subclasses: workflows

segmentation classes: *provide segmentation framework for different datatypes with or without parallel processing without providing a segmentation algorithm*

segmentation workflows: *extend the existing segmentation classes with different segmentation algorithms*

Extraction

extraction classes

Class: HDF5CellExtraction

Class: TimecourseHDF5CellExtraction

extraction classes: *provide code framework to extract single-cell images for all cells segmented within the previously calculated segmentation masks*

Classification

classification classes

Class: Classification

Class: MLClusterClassifier — *performs an inference using a pretrained pytorch model on all cells*

Class: CellFeaturizer — *calculates classic image features e.g. mean intensity on a chosen channel from all cells*

classification classes: *provide code framework to classify or featurize the generated single-cell image datasets*

Selection

selection classes

Class: LMDSelection

selection classes: *provide code framework to select cells for subsequent laser microdissection*