

# UCSD Microanalysis

Overview	2
Internals	2
Experimental Setup	2
Hardware and Software Requirements	3
CellTrack Segmentation & tracking module	4
Operation	4
Inputs	4
Output	4
CellQuant Quantification & Parameterization	7
Operation	7
Inputs	7
Output	7
CellQuery Quantification & Parameterization	9
Operation	9
Inputs	9
Output	9

---

# 1. Overview

---

Track and segment phase-contrast photomicrographs of primary and secondary cells. Specifically, the software is targeted at segmenting BMDMs, microglia, BV2, RAW and CHO cells.

The software has 3 components; tracking/segmentation, parameterization and analysis.

Matlab is required to run the software. The typical workflow is as follows:

1. Acquire and save images following naming conventions.
2. Copy images from microscope to a local directory accessible to MATLAB.
3. Run CellTrack and select proper directory. CellTrack will automatically process all the images, and create intermediate files and directories for further processing.
4. Run CellQuant and select proper directory.
5. Run CellQuery and select proper directory.

## 1.1. Internals

### 1.1.1. Settings

Because of differences between microscopes, the software will have distinct settings for the microscopes in the Hasty and Simon labs. The settings are hardcoded and are selected at run-time by a command-line option.

### 1.1.2. Usability

CellTrack is driven from the command-line. It requires some training, but is still intended for non-experts to use.

### 1.1.3. Processing

CellTrack processes the images serially, keeping 3 time points in memory at all times.

### 1.1.4. Portability

CellTrack will be portable between different computers and OS'es, requiring only a simple command to recompile a few functions.

### 1.1.5. Memory usage

In order to reduce memory usage, CellTrack processes the images serially, only keeping 3 time points in memory at any time.

## 1.2. Experimental Setup

### 1.2.1. Microscope Settings

Adjust the microscope so that the background and foreground levels are between 5% and 70%.

Magnification	20x
Background level	5%
Peak exposure level	70%

### 1.2.2. Image Channels

There are two obligatory channels of information that must be included in every data set, phase-contrast images and a fiducial channel, typically a nuclear-localized fluorescent protein. Any number of additional channels may be processed.

#### 1.2.2.1. Channel Numbering

The phase-contrast images must be identified as channel zero (0). The nuclear fiducial marker must be identified as channel one (1).

### 1.2.3. Multiple Locations

For an experiment with multiple locations on a single dish, CellTrack will have to be rerun for each location. In other words, each location is treated as an independent experiment. Processing data from multiple locations also requires the images from each location to be saved in separate locations.

### 1.2.4. Other Settings

What other settings/setup do we want to standardize?

## 1.3. Hardware and Software Requirements

### 1.3.1. MATLAB

Any requirements for MATLAB?

### 1.3.2. Analysis Computer

CellTrack can be run on any OS that has MATLAB. Additionally, the recommended requirements are a 2.5GHz processor and 8GB of RAM.

## 2. CellTrack Segmentation & tracking module

### 2.1. Operation

CellTrack is used to identify the margins of cells from phase-contrast images.

#### 2.1.1. Commands and options

MATLAB command and options described here.

### 2.2. Inputs

CellTrack requires requires specific suffixes and strings as part of the image filenames in order to properly operate. The expected nomenclature is detailed in Figure 1.

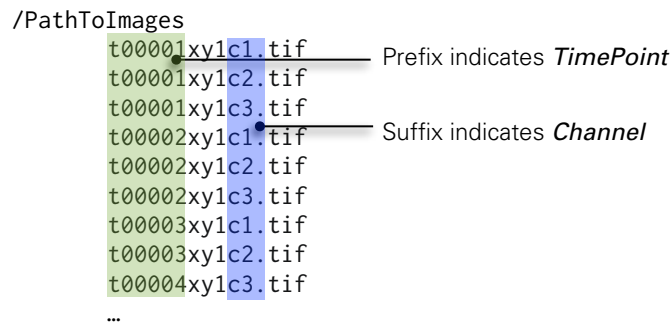


Figure 1. File naming convention for input directory.

### 2.1. Output

#### 2.1.1. Filename structure

The CellTrack module produces intermediate data suitable for parameterization and analysis by other modules. The intermediate data is organized and stored in files and subdirectories that are created during the segmentation and tracking process.

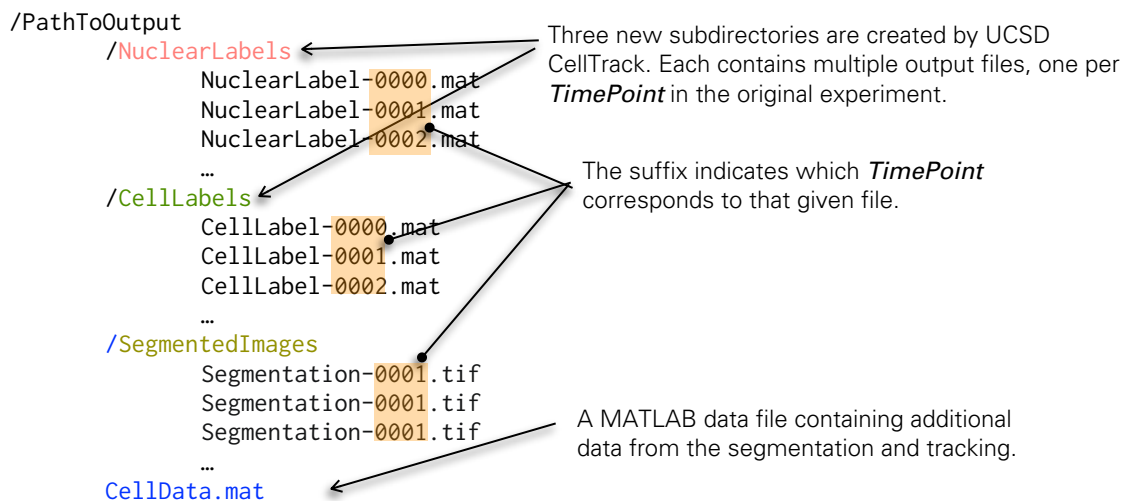


Figure 2. CellTrack stores output data in a distinct directory from the input images.

## 2.1.2. Data Types

The following data is created by CellTrack during segmentation and tracking of cells. Each of the ***NuclearMask*** and ***NuclearMask*** .MAT files contain a single variable. The variable name is the same in all of the files, consequently loading a ***CellLabel*** or ***NuclearLabel*** .MAT file overwrites the previously loaded ***CellLabel*** or ***NuclearLabel*** .MAT file.

### 2.1.2.1. ***NuclearLabel***

Description	Time point-specific monochrome mask of nuclei.
Data Type	MAT File
Size	Same dimensions as input images. There are as many <b><i>NuclearLabel</i></b> files as there are timepoints.
Location	Saved in the /NuclearMasks subdirectory.
Notes	Produced by segmentation of the phase contrast images. The indexes assigned to cells in <b><i>NuclearLabel</i></b> are invariant, i.e. once a cell is identified, its index in subsequent <b><i>NuclearLabel</i></b> images will never change,

### 2.1.2.2. ***CellLabel***

Description	Time point-specific label matrix for cells.
Data Type	MAT File
Size	Same pixel dimensions as input images. There are as many <b><i>CellLabel</i></b> files as there are timepoints.
Location	Saved in the /CellLabels subdirectory.
Notes	Produced by segmentation of the phase contrast images. The indexes assigned to cells in <b><i>CellLabel</i></b> are invariant, i.e. once a cell is identified, its index in subsequent <b><i>CellLabel</i></b> images will never change.

### 2.1.2.3. ***SegmentedImage***

Description	Time point-specific label matrix for cells.
Data Type	24-bit TIFF
Size	Same pixel dimensions as input images. There are as many <b><i>SegmentedImage</i></b> files as there are timepoints.
Location	Saved in the /CellLabels subdirectory.
Notes	False-colored images for displaying results of tracking and segmentation.

### 2.1.2.4. ***CellData***

Description	Stores object information.
Data Type	Structure
Size	One element per cell identified by CellTrack.
Location	Saved in the CellTracking.mat file.
Notes	The index of an object in the CellData array is used to identify objects uniquely. This index corresponds to the indexes from both <b><i>CellLabel</i></b> and <b><i>CellData.Parent</i></b> .

#### 2.1.2.5. *CellData.FrameIn*

Description	An integer indicating the earliest frame in which this cell is identified.
Data Type	Integer
Location	Part of the <i>CellData</i> structure.
Notes	Each row corresponds to a single cell. The first row is the lineage for the object number 1.

#### 2.1.2.6. *CellData.FrameOut*

Description	An integer indicating the last frame in which this cell is identified.
Data Type	Integer
Location	Part of the <i>CellData</i> structure.
Notes	Each row corresponds to a single cell. The first row is the lineage for the object number 1.

#### 2.1.2.7. *CellData.Parent*

Description	An integer indicating the index of the parent cell.
Data Type	Integer
Location	Part of the <i>CellData</i> structure.
Notes	Each row corresponds to a single cell. The first row is the lineage for the object number 1. Original cells have a <i>CellData.Parent</i> set to zero (0).

#### 2.1.2.8. *CellData.Edge*

Description	An boolean indicating the whether this object ever touches the boundary of the image.
Data Type	Boolean
Location	Part of the <i>CellData</i> structure.
Notes	Each row corresponds to a single cell. The first row is the lineage for the object number 1. True indicates the object touched an edge at some point during the experiment.

#### 2.1.2.9. *CellData.ImagePath*

Description	The path to the original images
Data Type	Strings
Location	Part of the <i>CellData</i> structure.
Notes	The directory separator is hardcoded into the path. It is a "/"

---

## 3. CellQuant Quantification & Parameterization

---

### 3.1. Operation

CellQuant is used to quantify cellular parameters.

#### 3.1.1. Commands and options

MATLAB command and options described here.

### 3.2. Inputs

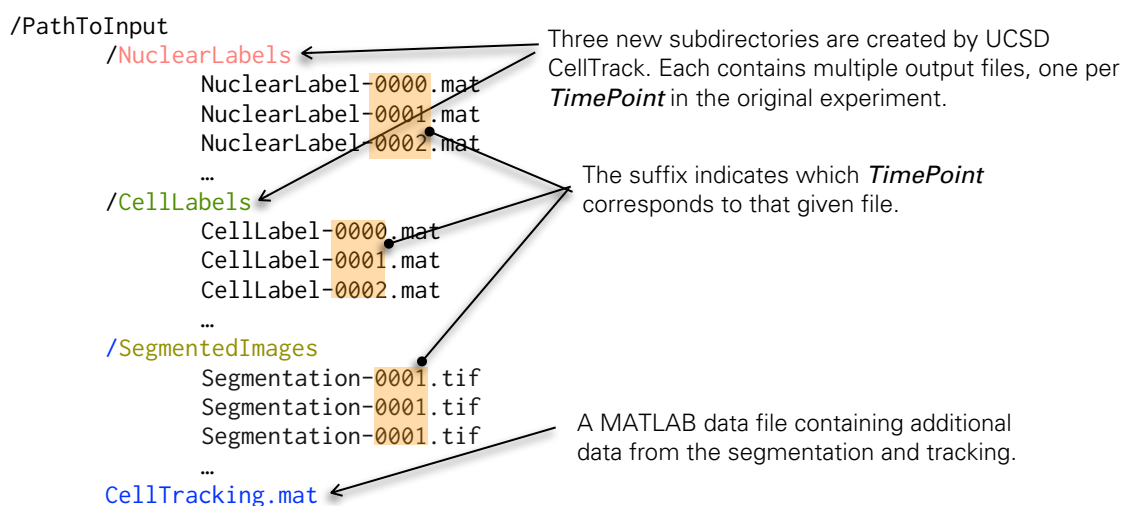
CellQuant requires requires images as well as data from CellTrack in order. The expected nomenclature and directory structure is detailed in Figure 3.

*Figure 3. File naming convention for input directory.*

### 3.1. Output

#### 3.1.1. Filename structure

The CellQuant module produces intermediate data suitable for analysis by other modules. The intermediate data is organized and stored in files and subdirectories that are created during the segmentation and tracking process.



*Figure 4. CellTrack stores output data in the same directory as the input images.*

#### 3.1.2. Data Types

The following data is created by CellTrack during segmentation and tracking of cells.

##### 3.1.2.1. CellParam.CentroidX

Description	description
Data Type	data type
Size	size
Location	location
Notes	Rows are cells. Columns are timepoints.

#### **3.1.2.1. *CellParam.CentroidY***

Description	description
Data Type	data type
Size	size
Location	location
Notes	Rows are cells. Columns are timepoints.

#### **3.1.2.1. *CellParam.Area***

Description	description
Data Type	data type
Size	size
Location	location
Notes	Rows are cells. Columns are timepoints.

#### **3.1.2.1. *CellParam.Intensity***

Description	description
Data Type	data type
Size	size
Location	location
Notes	Rows are cells. Columns are timepoints. Third-dimension is channel starting with channel 3. Intensity is integrated.

#### **3.1.2.1. *CellParam.Morphology***

Description	description
Data Type	data type
Size	size
Location	location
Notes	Rows are cells. Columns are timepoints.

#### **3.1.2.1. *CellParam.Texture***

Description	description
Data Type	data type
Size	size
Location	location
Notes	Rows are cells. Columns are timepoints.



---

## 4. CellQuery Quantification & Parameterization

---

### 4.1. Operation

CellQuery is used to classify and cluster cells, as well as to generate plots of raw data.

#### 4.1.1. Commands and options

MATLAB command and options described here.

### 4.2. Inputs

CellQuery requires data from CellQuery. The expected nomenclature and directory structure is detailed in Figure 5.

*Figure 5. File naming convention for input directory.*

### 4.1. Output

#### 4.1.1. Filename structure

The CellQuery module produces intermediate data suitable for analysis by other modules. The intermediate data is organized and stored in files and subdirectories that are created during the segmentation and tracking process.

*Figure 6. CellQuery stores output data in the same directory as the input images.*

#### 4.1.2. Data Types

The following data is created by CellQuery during segmentation and tracking of cells.

##### 4.1.2.1. Data1

Description	description
Data Type	data type
Size	size
Location	location
Notes	notes

##### 4.1.2.2. Data2

Description	description
Data Type	data type
Size	size
Location	location
Notes	notes