KAMeL Documentation

Contents

1 KAMeL Overview

This document details the usage of the KAMeL package (Kymograph Analysis with Machine Learning). KAMeL automatically segments kymographs of neuronal transport and reports the average displacement and speed of the particles in one kymograph. It assumes that the moving particles are traveling at a constant velocity, and hence creating fairly straight lines. The software uses an SVM which needs to be trained with annotated data. This document details how to both train KAMeL and use it to segment lines.

1.1 Citation

If you use this code in your research, please cite it as:

R. Cooper, D. Lipton, S. Yogev, K. Shen, and M. Horowitz, *KAMeL: Kymograph Analysis with Machine Learning*, in preparation, 2016.

1.2 Setup & System Requirements

KAMeL and its supporting code can be downloaded at:

`http://roscoope.github.io/KAMeL`

Furthermore, the SVM tool `liblinear` is required to run KAMeL, and can be downloaded here:

`https://www.csie.ntu.edu.tw/ cjlin/liblinear/`

Throughout this document, we assume a basic working knowledge of MATLAB: strings, cell arrays of strings, comments, using the command line, etc. Thus far, MTQuant has been tested on Windows 7 and 8, using MATLAB versions R2014b and R2013b and `liblinear-2.1`.

The code can be downloaded as a ZIP file or checked out as a Github repository. Download all of the files and keep them together. The either run the code in the top-level KAMeL directory, or elsewhere by typing the following commands into MATLAB to add the appropriate directories to the path:

```
>> addpath('C:/path/to/KAMeL')
>> addpath('C:/path/to/KAMeL/em')
>> addpath('C:/path/to/KAMeL/merge')
>> addpath('C:/path/to/KAMeL/misc')
>> addpath('C:/path/to/KAMeL/radonTransform')
>> addpath('C:/path/to/KAMeL/segmentationSVM')
>> addpath('C:/path/to/KAMeL/segmentationSVM/predict')
>> addpath('C:/path/to/KAMeL/segmentationSVM/train')
```

If you copy and paste commands from this document into MATLAB directly, some characters, such as curly braces ({}) might not copy correctly; be sure to review copied commands before running them.

2 Training KAMeL

To train the SVM in KAMeL, we first need to annotate a training dictionary. When KAMeL is run in training mode, the software randomizes the list of files, and sequentially presents them to the user. The user can select to add that image to the training dictionary or not, or to stop adding images to the training dictionary altogether. For each image that is added to the dictionary, the user must first select the dynamic range compression parameter. The display prompt asks the user to select a new value for the parameter or to enter the number -999 when the displayed image is satisfactory. If the image is dim, *decrease* the value of this parameter, and if the image is saturated, *increase* its value. The parameter value must be between 0 and 100.

Then, the user must click on the lines in the image. For each line, the user clicks on one endpoint, and then the second, and then hits “enter.” After all of the lines are selected, the user clicks “enter” again to save the selected lines.

A few notes:

- We recommend annotating ∼1000 lines.
- For every image you add to the training dictionary, you must annotate *all* lines in the image. If you skip lines, then during the retraining process, any lines that were not part of the training dictionary will be considered false positives, even if they are true positives.

2.1 Example Usage & Output

```
>> mode = 'train';
>> dirList = {'C:/Users/Roshni/Desktop/KAMeLTest/textquotesingle};
>> pathToLiblinear = ('../code/codeV2/liblinear-2.1/MATLAB/ ');
>> dictID = 'C:/Users/Roshni/Desktop/KAMeLTest/svm/test_';
>> KAMeL(mode,dirList,pathToLiblinear,'dictID',dictID);
The training dictionary has 3 files.
There are 324 files left in this dataset
The current file is C:/Users/Roshni/Desktop/KAMeLTest/images/224_BlReg.tif
Add image to training dictionary?(0 = yes, 1 = no, 2 = stop adding files) 0
Current pThresh = 75.  Enter a new pThresh or -999 to stop.  90
Current pThresh = 90.  Enter a new pThresh or -999 to stop.  -99
Current pThresh = 90.  Enter a new pThresh or -999 to stop.  -999
The training dictionary has 4 files.
```

```
There are 323 files left in this dataset
The current file is C:/Users/Roshni/Desktop/KAMeLTest/images/118_BlReg.tif
Add image to training dictionary?(0 = yes, 1 = no, 2 = stop adding files) 0
Current pThresh = 90.  Enter a new pThresh or -999 to stop.  -999
The training dictionary has 5 files.
There are 324 files left in this dataset
The current file is C:/Users/Roshni/Desktop/KAMeLTest/images/160_BlReg.tif
Add image to training dictionary?(0 = yes, 1 = no, 2 = stop adding files) 2
Creating training data...
file 1 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/64_BlReg.tif
file 2 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/76_BlReg.tif
file 3 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/294_BlReg.tif
file 4 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/224_BlReg.tif
file 5 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/118_BlReg.tif
Training SVM...
Best C = 1.000000 CV accuracy = 89.4901%
Retraining SVM...
Run KAMeL on each image in the training dictionary...
Image 1 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/64_BlReg.tif
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Image 2 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/76_BlReg.tif
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Image 3 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/294_BlReg.tif
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Image 4 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/224_BlReg.tif
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Calculating Radon Transform...
Segmentation SVM...
```

```
Merging lines...
Image 5 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/118_BlReg.tif
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Calculating Radon Transform...
Segmentation SVM...
Merging lines...
Creating new training data...
file 1 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/64_BlReg.tif
file 2 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/76_BlReg.tif
file 3 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/294_BlReg.tif
file 4 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/224_BlReg.tif
file 5 out of 5:  C:/Users/Roshni/Desktop/KAMeLTest/images/118_BlReg.tif
Training the SVM with augmented data...
Best C = 0.031250 CV accuracy = 98.8216%
Model saved in:  C:/Users/Roshni/Desktop/KAMeLTest/svm/test_model_2016_06_27.mat
```

3 Using KAMeL to trace lines

KAMeL uses the model file described in the previous section to trace lines. Again, it requires as input a list of directories in which to search, the mode, and the path to the liblinear installation. Then it saves a '*workspace*.mat'file for each image file in the same location as the image file, with the same name as the image file, just with the word "workspace" and the date appended to it. When loaded, the variable diagLinesOutC contains the output lines. This variable is organized as such: to speed up the processing of the kymographs, the kymographs are segmented into hStep $\times$ wStep blocks (hStep and wStep are set to 300 and 500 by default, but can be changed). Then, each entry in diagLinesOutC is a cell array containing the lines traced for the corresponding block of the kymograph. Each entry of the cell array is a sparse binary mask of one traced line.

KAMeL saves two CSV files. The first CSV file, with the name of the input argument fileOut, contains an entry for every image analyzed. For each image file, it stores the file name, the name of the saved workspace file, the average vesicle run length, and the average vesicle speed.

The second file is has the suffix '*_grouped.csv.'The program assumes that, if the input directory list is only one element long, that each sub-folder of that directory is a separate group of worms. KAMeL assigns each group a number, which is included in fileOut, and used for both the distribution refinement. The '*_grouped.csv'file calculates the mean and standard deviation for the run length and speed for each separate group of worms.

3.1 Example Usage

```
>> dirList = 'C:/Users/Roshni/Desktop/KAMeLTest/images/predict';
>> mode = 'predict';
>> pathToLiblinear = '../code/codeV2/liblinear-2.1/MATLAB/';
>> fileOut = 'C:/Users/Roshni/Desktop/KAMeLTest/data';
>> modelFile ='C:/Users/Roshni/Desktop/KAMeLTest/svm/test_model_2016_06_27.mat';
>> KAMeL(mode,dirList,pathToLiblinear,'fileOut',fileOut,'modelFile',modelFile);
```

4 Directory Organization Naming Conventions

Every analyzed line scan is given a group number, titled `DirNum`: The group number of the line scan file. If the `dirList` sent to `KAMeL` had one entry, then each sub-folder inside of `dirList` is considered a separate group, and receives a distinct `DirNum`. Otherwise, each directory in `dirList` is considered a distinct group. An example of the assignment of `DirNum` is described below. First, consider the following directories:

- `C:/path/to/data`
- `C:/path/to/data/wild_type`
- `C:/path/to/data/mutant1`
- `C:/path/to/data/mutant2`

Assume that `C:/path/to/data` contains the three subdirectories `wild_type`, `mutant1`, and `mutant2`, and no other images. The three subdirectories can contain images, subdirectories with more images, or a combination of both. Depending on the input `dirList` to `MTQuant`, the assignment of the output `DirNum` varies.

If `dirList` is just a single directory (e.g., `dirList = {'C:/path/to/data'};`), then all subdirectories contained in `dirList` are considered a separate group. Hence, images in the `wild_type` subdirectory (and its subdirectories) would be in group 1, `mutant1` images in group 2, and `mutant2` images in group 3. The output `DirNum` is 1 for the images in group 1 (`wild_type`), 2 for images in group 2 (`mutant1`), and 3 for images in group 3 (`mutant2`).

If `dirList` is a list of directories (e.g., `dirList = {'C:/path/to/data/mutant2'; 'C:/path/to/data/mutant1'};`), then each directory in the list is considered a separate group, and numbered based on the order it appears in `dirList`. Therefore, images in the `mutant2` subdirectory (and its subdirectories) would be in group 1 and `mutant1` images in group 2. Again, `DirNum` is 1 for the images in group 1 (`mutant2`) and 2 for images in group 2 (`mutant1`).

5 Full List of Input Arguments

Input Argument	Description	Default Value
REQUIRED ARGUMENTS		
`model`	`'train'` or `'predict'`	
`dirList`	List of directories in which to find the images to process The program searches the directories in `dirList`, as well as all subdirectories, for the files to analyze. ***Must be a cell array of strings.***	
`pathToLiblinear`	String containing the path to the MATLAB subdirectory of the `liblinear` installation	
`fileOut`	Name of the file to store the data. Must ***not*** have an extension.	
TRAINING (REQUIRED)		
`dictID`	Name of the file to store the dictionary or look for existing dictionary. Must ***not*** have an extension.	
TRAINING (OPTIONAL)		
`addToDictionary`	1 = search for an existing training dictionary and add files to it 0 = create a blank training dictionary	1
`retrainManual`	During SVM REtraining, ... 1 = ask the user to select mislabeled lines 0 = automatically determined mislabeled lines	0
PREDICTION (REQUIRED)		
`modelFile`	String containing name of file storing SVM model. This string was printed out during training.	
PREDICTION (OPTIONAL)		
`toEM`	0 = do not implement distribution refinement 1 = implement distribution refinement	1
GENERAL		
`fileExt`	String to use when searching for files to process	'*.tif'
`verbose`	1 = display figures after processing each image and display detailed status updates 0 = display no figures and only minimal status updates	0
`hStep`	Vertical size of blocks to run KAMeL on; to run on entire image at once (can be slow), set hStep to be larger than the height of the largest image in the dataset	300
`wStep`	Horizontal size of blocks to run KAMeL on; to run on entire image at once (can be slow), set wStep to be larger than the width of the largest image in the dataset	500

Input Argument	**Description**	**Default Value**
`pThresh`	Dynamic range parameter; when empty, `KAMeL` uses the value from the `modelFile`. Override if a new set of images are especially dim or bright	[]
`wThresh`	Minimum distance (in pixels) a vesicle must travel in order to be considered a line	10
`vThresh`	Potential line segments are ignored if more than `vThresh` of their pixels are part of the vertical mask	0.5
`fvThresh`	Connect segments predicted simultaneous on the same $< \theta, \rho >$ line if they are are fewer than 10 pixels apart and more than `fvThresh` pixels fall on the vertical mask	0.5
`dThresh`	Minimum average value of the decision values of pixels in a line segment; any line segments with an average decision value below `dThresh` are ignored	-1
`fThresh`	Maximum fraction of overlap a new line segment can have with the existing line traces; any line segments with an overlap above texttttfThresh are ignored	0.75
`rtThresh`	Maximum separation of two $< \theta, \rho >$ pairs for the line segments to be merged	5

Table 1: Input Arguments to KAMeL