SpindleUtil
Manual

By T47, *Dec 2014*

Contents

Introduction

- SpindleUtil is a set of MATLAB scripts to facilitate the process of spindle line-scan quantitation. It is aimed to improve the productivity by replacing the "traditional" way of manually draw lines in MetaMorph and process data by hand using Excel.
- SpindleUtil makes it much easier in rotating the image to position poles vertically. It also introduces box-scan, as well as the line-scan as a comparison.
- SpindleUtil automates the quantitation, data visualization, comparison, and data management, making it all done in MATLAB.

Introduction

- In this manual, we will demonstrate the basic setup and process. Using a test dataset, the interface and output files will be illustrated.
- For future expansions, the steps after the FITC/TexRd ratio plot needs to be developed and tested.
- All scripts are named with "`spindle_`" prefix, making it easier to identify in MATLAB
- You should "Set Path" first to include those scripts (*Add with subdirectories*).

Without SpindleUtil

- Before SpindleUtil became available, the image quantitation is done with MetaMorph and Excel. Lines are drawn in MetaMorph, suffering problems with inconsistent spindle boundaries. Data was exported to a huge Excel sheet, making plots and comparisons are thus hard and cumbersome.

MetaMorph® Line-scan Quantitation

MATLAB The Language of Technical Computing

Plots Comparison

5

With SpindleUtil

- SpindleUtil aims to take over the whole process using MATLAB, eliminating the complicated and time-consuming data transfer.
- SpindleUtil takes advantage of an interactive interface for spindle rotation and boundary setup, which is faster than in MetaMorph or ImageJ.
- SpindleUtil quantitates the intensity from the original TIFF, and overlays them to each other using normalized inter-pole distance. This is much faster than Excel.
- SpindleUtil automatically saves all figures, log, and workspace data to an “analysis” folder in an organized way.

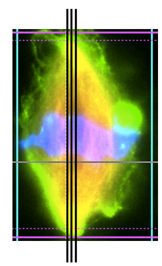
6

Box-scan Quantitation

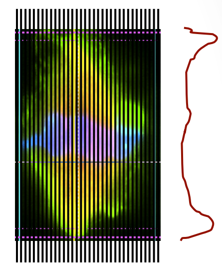
- One advantage of using SpindleUtil is its box-scan quantitation, especially when comes to deformed spindle geometry, e.g. splayed poles.
- Line-scan quantitation only takes account for the center line (pole-pole line) and 2 side-lines. This is representative in regular spindles, but may be inaccurate and losing information.
- Box-scan draws as many lines as possible in the spindle (including the 3 lines in line-scan), and thus averaging the whole spindle. This is useful in e.g. splayed pole case where the pole is not focused.

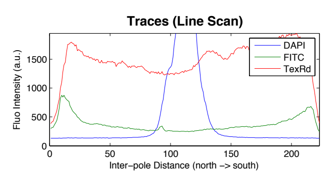
7

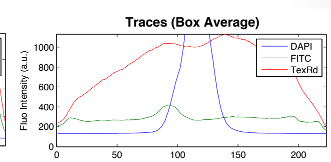
Box-scan Quantitation



- Line-scan (3 lines)
- Box-scan (every line)







8

Section 1: Image Processing

- In this section, we will go over the pipeline of image processing, which is from raw TIFF files to quantitated traces saved in MATLAB session.
- Related scripts are in folder:
  Scripts/SpindleQuantify/ImageProcess
- However, you will be using a master function `spindle_mat_analysis`, instead of any individual functions in the above folder.

9

`spindle_mat_analysis`

- This is an all-in-one function to perform the analysis pipeline.
- Usage:
  ```
  spindle_mat_analysis (dir_name, CEN_LINE_OFFSET, POLE_PORTION, y_lim, NUM_BIN)
  ```
- **`dir_name`** is required, all others are optional.
- Example:
  ```
  spindle_mat_analysis("3. M+F2MCAK");
  ```

10

`spindle_mat_analysis`

- Arguments:
  - **`dir_name`**: *(type string)*, the name of the folder containing all the TIFF images.
  - **`CEN_LINE_OFFSET`**: *(type int)*, the distance between side-lines and the center line used in line-scan. If not specified, default of 5 will be used.
  - **`POLE_PORTION`**: *(type double)*, the portion along the pole-pole axis considered as pole region. If not specified, default of 1/10 will be used.
  - **`y_lim`**: *(type double)*, the y axis upper limit in the ratio plots. If not specified, default of 2.0 will be used.
  - **`NUM_BIN`**: *(type int)*, the number of bins used to group ratio data points. If not specified, default of 100 will be used, meaning one ratio data point for every 1%.

11

More about the arguments

- `dir_name`:
  - First, make sure your "current working directory" is correct (see left panel in MATLAB window). Go to the parent directory of your image-file folder (so that the target folder is a child of the current working directory).
  - Second, use the target directory name as `dir_name`. For example, `"3. M+F2MCAK"` (exactly as is, including whitespaces).

12

More about the arguments

- `dir_name`:
  - All TIFF files should meet the format of `“Prefix_XXX_fluroChannel.TIF”`:
    - Prefix should be exactly same as the parent folder name (i.e. `dir_name`).
    - ID number is 3-digit, (filled with 0s). It needs to be neither continuous (of files in the same folder), nor starting with `001`.
    - Fluorophore Channel should be one of the following: `“w1DAPI”`, `“w2FITC”`, `“w3Texas Red”`. This is same as default by microscope image output.
    - Extension is `“TIF”`, not `“TIFF”`.
    - Valid image name example: `“3. M+F2MCAK_001_w2FITC.TIF”`.
  - “nd” and “_thumb” files in the folder will be ignored.

13

More about the arguments

- `dir_name`:
  - The script is expecting that for each spindle (set), 3 images of each channel are present. Thus, the number of total legit TIF images should be multiples of 3.
  - Possible error messages:

```
ERROR: directory not found.

ERROR: TIFF images set (of 3) incomplete.
       14 non-thumb TIF files present.

ERROR: TIF and directory name mismatch.
       3. M+F2MCAK/5. M+715AA_001_w1DAPI.TIF
```

14

More about the arguments

- `CEN_LINE_OFFSET`:
  - This defines the distance between the 3 lines used in line-scan. In line-scan, one center line is drawn, and two side-lines (one on each side) `CEN_LINE_OFFSET` pixels away are automatically drawn.
  - Default is 5 (pixels), as inherited from older MetaMorph approaches.

15

More about the arguments

- `POLE_PORTION`:
  - This defines the portion of spindle considered as “pole region”. Default is 1/10, which means the inter-pole distance will be divided to 10 sections, where the 1^{st} and 10^{th} section are considered “north” and “south” pole, while the 2^{nd}-to-9^{th} sections are “body”.
  - This number is useful when comparing a distribution of “pole” (where MTs are enriched) to rest of spindle. Currently, no calculation in the output files is dependent on this argument. But it can be used in marking the boundary on plots.
  - *Use 1/10, not 10.*

16

More about the arguments

- `y_lim`:
  - This defines the upper limit of y-axis in all ratio plots. Default is 2.0, which means ratio of 0-to-2.0 will be shown. Ratios higher than 2.0 will appear “cut-off” on the plot. You may specifiy a lower number, e.g. 1.5, if max ratio of the trace is low.
- `NUM_BIN`:
  - This defines the number of bins used in grouping/ normalizing the ratio trace across different spindle length. See “calculate the ratio” page.

17

spindle_mat_analysis

- A successful launch will show the following message in MATLAB console:

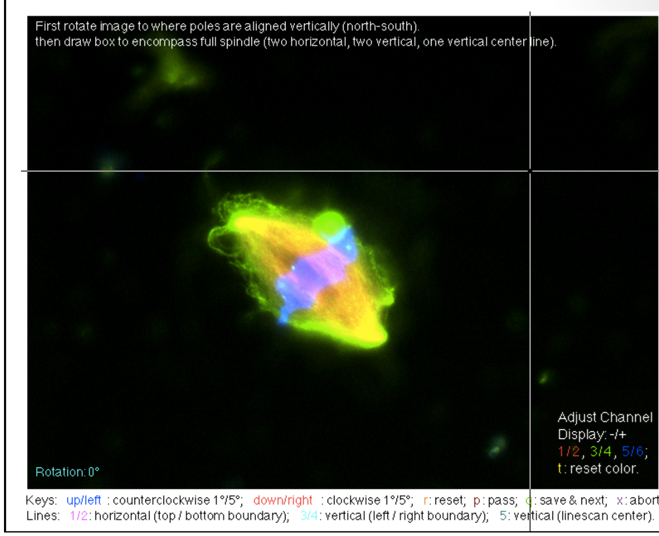
```
Load files from directory: 3. M+FMCAK
Load 15 non-thumb TIF images from directory.
Group into 5 spindles {DAPI, FITC, TexRd}.

Instructions:
First rotate image to where poles are aligned vertically (north-south),
then draw box to encompass full spindle
        (two horizontal, two vertical, one vertical center line).
Controls:
Keys: up/left: counterclockwise 1°/5°; down/right: clockwise 1°/5°;
      r: reset; p: pass; q: save & next;
      x: abort (premature terminate, data lost).
Lines: 1/2: horizontal (top / bottom boundary);
       3/4: vertical (left / right boundary);
       5: vertical (linescan center).
```

18

Define spindle from image

- For each spindle set (of 3 images representing each channel), we will select a region that defines the spindle. This is done using an interactive interface.
- A window will pop out (see next page), with a cross-cursor and key descriptions.
- We will first rotate the image, then draw 5 lines to define the spindle.

19

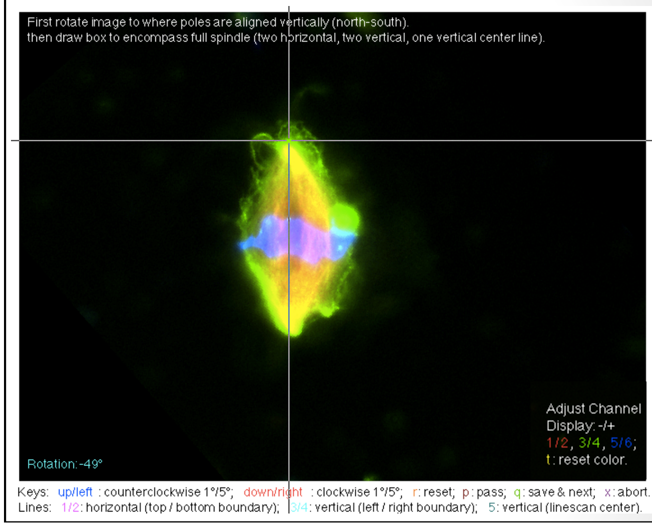


20

Rotate the spindle

- This is to make the line drawing easier. The goal is to rotate the image so that the spindle is oriented/aligned north-to-south (vertical).
- To do so, use the following control keys:
  - **Up**: rotate counterclockwise by 1 degree;
  - **Down**: rotate clockwise by 1 degree;
  - **Left**: rotate counterclockwise by 5 degree;
  - **Right**: rotate clockwise by 5 degree.
- Use left/right first to rotate the image, then refine by up/down. You can check the orientation by overlaying the cursor, which has vertical cross-line.
- The current rotation angle is shown in bottom-left corner (cyan).

21



22

Draw 2 horizontal lines

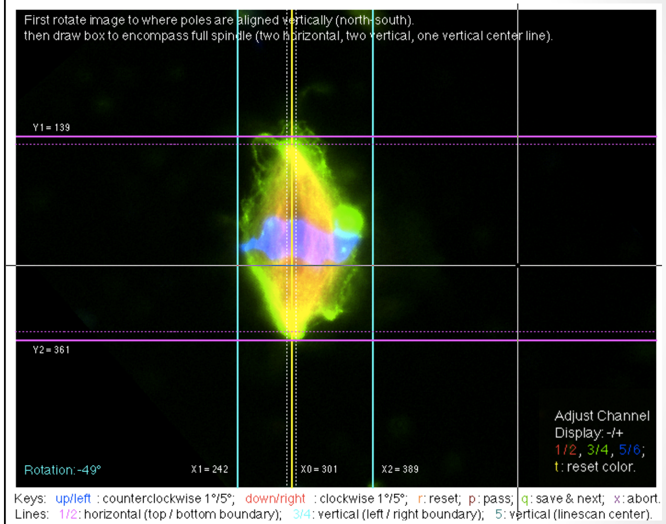
- These first 2 lines are horizontal and defines the north and south boundary. They are shown in magenta. It should be drawn exactly at the outer tip of the pole (names `y1` and `y2`).
- The `POLE_PORTION` defined pole region will be automatically shown by dashed lines inside the 2 horizontal lines you drawn.

23

Draw 3 vertical lines

- These next 2 lines are vertical and defines the left and right boundary. They are shown in cyan. It should be drawn exactly at the widest point of the spindle (named `x1` and `x2`).
- The last vertical line mimics the line-scan, where it should be through both poles. So it's not necessarily at the center. It is shown in yellow (named `x0`), and the `CEN_LINE_OFFSET` defined two side-lines are shown in dashed lines.

24



25

Interactive interface

- At this point, the line drawing is complete. Hit "**q**" to save and quit, which leads you to the next spindle.
- Other keys:
  - **r**: reset, restores the original image and erases all lines;
  - **p**: pass, the current spindle is considered bad (quality), thus it will be excluded from the following analysis (does not count in ratio calculation); Any rotation or lines drawn will be ignored.
  - **x**: abort: quit the `spindle_mat_analysis` function without saving any data (all will be lost).

26

Interactive interface

- Note:
  - The "close window" button is disabled to avoid mistaken clicks. Neither does "`command+w`" works.
  - The MATLAB console window may gain focus when you click keys too fast. Use "`command+``" to switch back to it.
  - **You can't go back to a spindle once "q" or "p".** So think twice before passing/finishing a spindle.
  - You can use ***1/2***, ***3/4***, and ***5/6*** key pairs to adjust the display of R, G, B channels. This only alters the display (contrast), not affecting the data in the image to be quantified. Use ***t*** to reset color display (not reset rotation or lines).

27

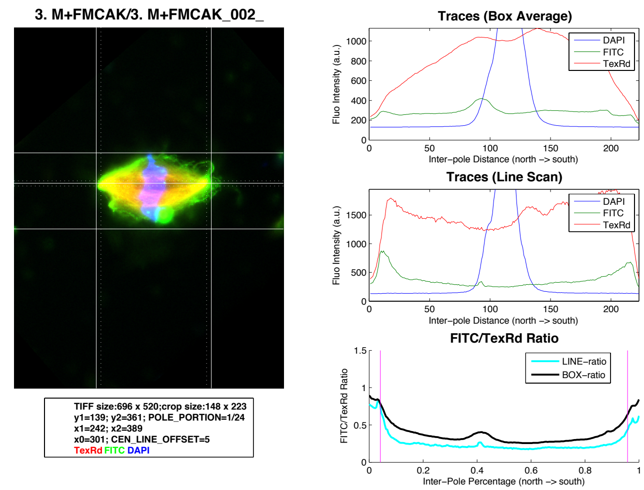
Spindle output

- Once you've done the definition for all spindles in the folder, an output file (.eps) will be saved for each spindle. See example next page.
- The output file includes the image of the spindle (rotated), the 5 lines drawn on top of the image, parameters used, and the quantitated traces. Although printing out the output is costly (color printing, black background), it is a good record.

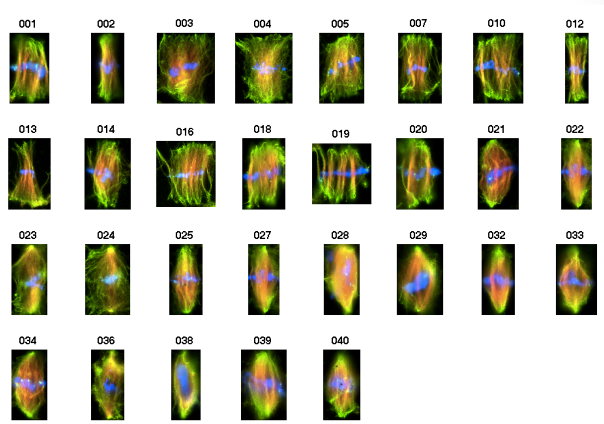
28

Spindle output

- Outputs are saved to `"Prefix_analysis/summary"` with same name as spindle, e.g. `"3. M +F2MCAK_analysis/summary/3. M +F2MCAK_002.eps"` (not same folder as TIF data).
- Additionally, a snapshot output file (`"Prefix_analysis/summary.eps"`) is present for an overview of all "good" spindle shapes.



Output file



Spindle output

- First 2 plots are the fluorescence intensity from pole-to-pole, averaged by box-scan and line-scan, respectively.
- X-axis is the pole length in pixels; Y-axis is the averaged fluorescence. Again, box-scan averaged all columns while line-scan only samples 3 columns.
- The 3rd plot is the FITC/TexRd ratio (assuming that FITC channel is protein of interest and Texas Red is labeled MTs). Ratio at each pixel position is calculated by dividing the FITC intensity by TexRd. X-axis is normalized to 0%-to-100% and the `POLE_PORTION` line is drawn for reference.

Section 2: Data Processing

- In this section, we will go over the pipeline of data processing, which is from quantitated traces to normalized FITC/TexRd ratio.
- Related scripts are in folder:
  Scripts/SpindleQuantify/DataProcess
- However, you will still be using a master function `spindle_mat_analysis`, instead of any individual functions in the above folder (continuously from section 1).

33

Calculating the ratio

- As described in the previous section, FITC/TexRd ratio of each spindle is calculated by simple division.
- However, the ratio of each spindle cannot be averaged directly since the length of each spindle (pole-to-pole distance in pixels) are different, thus the number of points in the (0%-to-100%) range is different.
- To solve the problem, we group the data points in the ratio trace to 100 bins first, then average the binned ratio trace across spindles.
- This is done by `spindle_mat_average_ratio` function, with a default `NUM_BIN = 100`.

34

Calculating the ratio

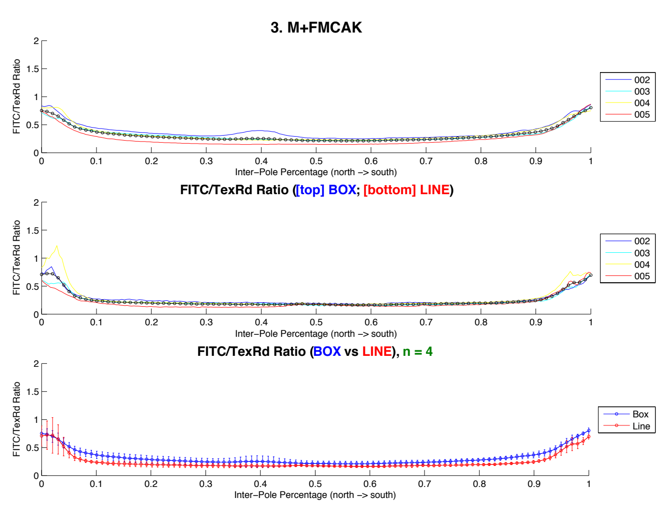
- Briefly, the grouping into bins is simple:
  - Divide 0%-100% into 100 bins (1% each);
  - Find the data points in the original ratio in each bin;
  - Assign ratio of each (1%) bin by averaging the ratio data points fall in the bin.
- Usually, the pole-to-pole distance in pixels is larger than 100. So each 1% bin will have at least one ratio data points.
- In the end, an average of ratio of all spindles is calculated and plotted (as well as standard deviation).

35

Calculating the ratio

- An example output is in next page.
- For box-scan and line-scan, the averaged ratio (across spindles) is colored in black and overlaid to all individual ratio traces.
- The averaged ratio is plotted against each other (box-scan vs. line-scan) with error bars.
- Overall, the box-scan and line-scan agrees in trends, while box-scan has smaller errors.

36



37

Calculating the ratio

- Notes:
  - The averaged ratio is across all "good" spindles in the folder, not discriminating "regular vs. splayed". For such selection, see "pick spindles to plot" page.
  - Ratio plot is saved as `"Prefix_analysis/ratio.eps"`.
  - Screen log is saved as `"Prefix_analysis/log.txt"`.
  - MATLAB data is saved as `"Prefix_analysis/save.mat"`. (see "MATLAB data structure" page)

38

Pick spindles to plot

- You may need to pick a subset of spindles in a folder and calculate statistics of the subset, instead of the mixture of all "good" spindles.
- To do so, you will use an all-in-one function:
- Usage:

```
obj_select = spindle_mat_pick (spd_data,
str_title, file_name, y_lim,
NUM_BIN)
```

- **`spd_data`** is required, all others are optional.
- Example:

```
obj_splayed = spindle_mat_pick (spd_data,
"splayed");
```

39

`spindle_mat_pick`

- Arguments:
  - **`spd_data`**: *(type spd_data struct)*, the `spd_data` variable from `save.mat`.
  - **`str_title`**: *(type string)*, the title of the plot, e.g. "MCAK-715AA splayed". If not specified, default of folder name will be used.
  - **`file_name`**: *(type string)*, the saved .eps file name. If not specified, default of "ratio_pick" will be used.
  - **`y_lim`**: *(type double)*, the y axis upper limit in the ratio plots. If not specified, default of 2.0 will be used.
  - **`NUM_BIN`**: *(type int)*, the number of bins used to group ratio data points. If not specified, default of 100 will be used, meaning one ratio data point for every 1%.

40

More about the arguments

- `spd_data`:
  - First, load the `save.mat` from any "prefix_analysis" folder. This loads the previously saved data into MATLAB session. The `spd_data` variable in `save.mat` is the data structure holding all spindle information and quantitated data.
- `str_title` and `file_name`:
  - These allows you to specify the figure title and saved figure name for easier comparison when printed out.
- `y_lim` and `NUM_BIN`:
  - Same as before.

41

More about the arguments

- `obj_select`
  - This is the return variable that holds the data. It is a struct with following fields:
    - `list_chosen`: all spidnle IDs used in the average ratio calculation;
    - `ratio_box_mean`: averaged ratio of box-scan;
    - `ratio_box_std`: standard deviation of box-scan ratio;
    - `ratio_box_mean`: averaged ratio of box-scan;
    - `ratio_box_std`: standard deviation of box-scan ratio;
  - Use a variable name that is identifiable, e.g. "`obj_splayed`". DO NOT use "`obj_chosen`".
  - See "MATLAB data structure" page.

42

spindle_mat_pick

- First, you will be asked to type in the selection:

```
Type in spindle IDs (MATLAB expression, e.g. [1:10,12].
Only spindles not marked as is_bad will be selected.
Input:
```

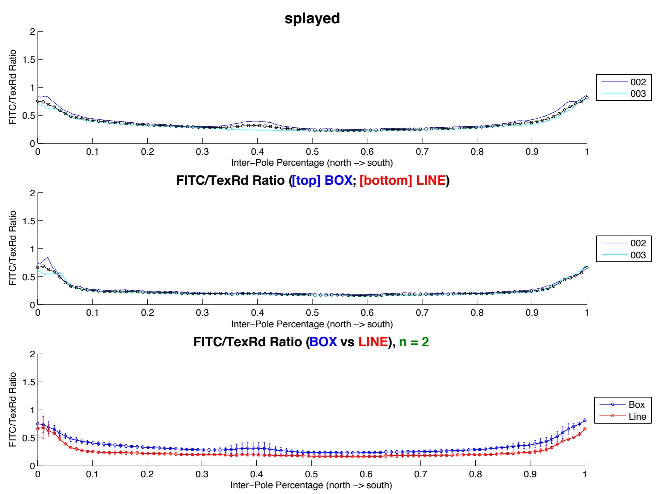
- Use MATLAB expression for ranges, e.g. [1:10], or [1:10, 12], or [1:10, 12:15], etc. <u>Do not forget the square brackets</u>.
- The function will automatically go through `spd_data`, and get all good (not "`is_bad`") spindles in your range, and put together a new `list_select` for averaging and plotting.

43

spindle_mat_pick

- The result figure is similar to the calculated ratio figure, but only with the spindles you just picked.
- It is saved as the name you input (or "`ratio_pick.eps`" if left blank), at the same folder as current working directory. Also, a "`ratio_pick_summary.eps`" will be provided as snapshot for all chosen spindles.
- The number of picked spindles is shown in the title of 3rd panel ("` n = `");
- Remember to **<u>SAVE</u>** the result to the same "`save.mat`" before quit MATLAB!

44



45

Get Mean Ratio

- Now you can get the average ratio of the pole vs. body for all the subset (obj_good, obj_splayed, etc.).
- Usage:

```
    [ratio_poles, std_poles, ratio_body,
std_body] =
spindle_mat_compare_ratio(dir_name,
POLE_PORTION, y_lim)
```

- **`dir_name`** is required, all others are optional.
- Example:

```
    spindle_mat_compare_ratio ([]);
```

46

`spindle_mat_compare_ratio`

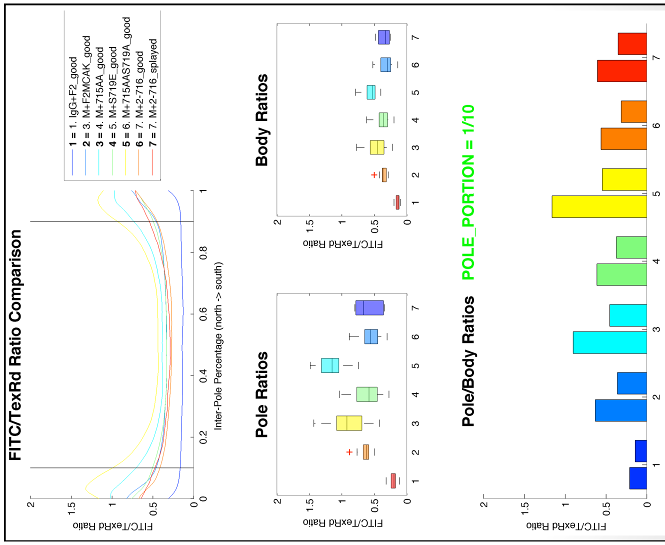
- Arguments:
  - **`dir_name`**: *(type string cell)*, all the folder names. Use `[]` (empty) to let the program fetch all folders with `"_analysis"` suffix in the current working directory. Make sure you are at the right path. Otherwise, specify the folders to include, e.g. `{'3. M+F2MCAK', '4. M+715AA'}`. Note, curly braces `{}` are required for cell type; you do not need to type the `"_analysis"` suffix in the cell.
  - **`POLE_PORTION`**: *(type double)*, the portion along the pole-pole axis considered as pole region. If not specified, default of 1/10 will be used.
  - **`y_lim`**: *(type double)*, the y axis upper limit in the ratio plots. If not specified, default of 2.0 will be used.

47

`spindle_mat_compare_ratio`

- The function will go over all the folders listed (with `"_analysis"` suffix appended), and look for and load `"save.mat"`. Make sure the MATLAB workspace file name is not changed.
- Then it will automatically identify all the `obj_*` variables in save.mat. There should be at least `obj_` (`obj_good`), and as many as `obj_*`s (e.g. `obj_splayed`). All `obj_*`s will be plotted individually.
- The following figure named `"ratio_compare.eps"` will be saved, as well as individual ratio plots for each subset group (e.g. `"pole_ratio_3. F2+MCAK_good.eps"`)

48



49

spindle_mat_compare_ratio

- The plot includes:
  - FITC/TexRd ratio traces for each experimental condition (e.g. 3. F2+MCAK), with specified POLE_PORTION overlaid as vertical black lines.
  - Separate plots of averaged FITC/TexRd ratio within both pole regions (e.g. 0~1/10 and 9/10~1), and averaged FITC/TexRd ratio within body region (e.g. 2/10~9/10). Used boxplot.
  - Combined plot of pole and body together as bar plot. Numbering and coloring are the same as in legend.

50

spindle_mat_compare_ratio

- Returned data: there are 4 variables returned from `spindle_mat_compare_ratio`:
  - **`ratio_poles`**: *(type double array)* averaged pole ratios of all subsets (as plotted in figure, same numbering as in figure).
  - **`std_poles`**: *(type double array)* standard deviation of `ratio_poles`.
  - **`ratio_body`**: *(type double array)* averaged body ratios of all subsets.
  - **`std_body`**: *(type double array)* standard deviation of `ratio_body`.

- For convenience, all the data above are automatically saved to "`ratio_compare.txt`". **Watch out for overwriting**! You can load it into excel using "import". File is comma delimited.

51

Manually average ratios

- You may need to tailor a specific set of commands for your special need in averaging pole/body ratios from data. Here is a guideline of how-to.
- These requires understanding of basic MATLAB operation, as well as knowledge of the data structure of the `save.mat`. Read "MATLAB data structure" section first.

52

Manually average ratios

- **1) Load the save.mat that contains data of interest.**
  - Remember, all results are named samely as `save.mat`, and the variable names inside each `save.mat` is the same. Loading a second `save.mat` will overwrite the previous one in the open MATLAB session (not the file itself).
  - One way is to load into a variable, e.g. `spd_715AA = load('5. M+715AA/save.mat')`. In this way, you can access all data of 715AA inside the variable `spd_715AA`, e.g. `spd_715AA.spd_data{i}`. Read more using `help load`.
  - Another way is to rename the variables before loading the next `save.mat`. For instance, rename `obj_good` to `obj_715AA_good` preserves its values from being overwritten.

Manually average ratios

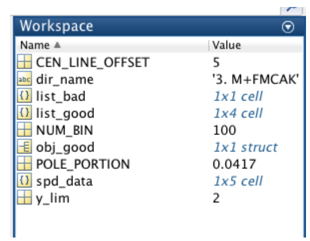
- **2) For each obj_ of interest, calculate averaged ratios.**
  - Call function `pole_ratios = spindle_mat_divide_pole(spd_data, list_chosen, POLE_PORTION)`.
  - You need the corresponding `spd_data`, and `list_chosen` (which is a field of `obj_`).
  - It returns a variable `pole_ratios` *(type double matrix)*:
    - 1st column is label of spindle ID (e.g. 001);
    - 2nd column is the averaged FITC/TexRd ratio of left pole (specified by POLE_PORTION) for each spindle.
    - 3rd column is the averaged FITC/TexRd ratio of body.
    - 4th column is the averaged FITC/TexRd ratio of right pole.

Manually average ratios

- **3) Calculate statistics of ratios**
  - Now you can calculate the average FITC/TexRd ratio of poles in the spindle group by averaging all values of left pole and right pole, e.g. `mean([pole_ratios.ratios(:,2); pole_ratios.ratios(:,4)])`; as well as errors (see `help std`).
  - Do the same for body region.
  - This gives you a pair of averaged FITC/TexRd ratios for one experimental condition (as one pair of bars in figure output).
  - Then you can plot the result as you want. Basically the above manual steps can be identified in `spindle_mat_compare_ratio` code.

MATLAB data structure

- There are several variables in save.mat:



Workspace

Name ▲	Value
CEN_LINE_OFFSET	5
dir_name	'3. M+FMCAK'
list_bad	*1x1 cell*
list_good	*1x4 cell*
NUM_BIN	100
obj_good	*1x1 struct*
POLE_PORTION	0.0417
spd_data	*1x5 cell*
y_lim	2

- Input arguments of `CEN_LINE_OFFSET`, `POLE_PORTION`, `y_lim`, `NUM_BIN` and `dir_name` are saved.

MATLAB data structure

- `spd_data` is the main variable holding all data. It is a 1xn cell, where n is the total number of spindles (good and bad together) originally in the folder.
- For each entry of `spd_data` (e.g. `spd_data{2}`), it is a struct as below:

57

MATLAB data structure

Variables – spd_data{1, 2}

spd_data | spd_data{1, 2}

1x1 struct with 16 fields

Field ▲	Value	Min	Max
raw_file	'3. M+FMCAK/3. M+FMCAK_002_'		
rotation_angle	-48	-48	-48
box_coord	[144.0000,363,244,381,298]	144.0...	381
img_raw	*520x696x3 uint16*	<Too ...	<Too ...
is_bad	0		
data_box	*220x3 double*	129.3...	2.0009...
data_line	*220x3 double*	130	3.6003...
data_label	*1x3 cell*		
img_box	*220x138x3 uint16*	121	4077
CEN_LINE_OFFSET	5	5	5
POLE_PORTION	0.0417	0.0417	0.0417
ratio_box	*220x1 double*	0.2470	0.8577
ratio_line	*220x1 double*	0.1740	0.8477
plot_x	*220x1 double*	0	1
ratio_box_100	*100x1 double*	0.2474	0.8374
ratio_line_100	*100x1 double*	0.1754	0.8293

58

`spd_data` entry

- **`raw_file`:** *(type string)* the prefix of the spindle image file names.
- **`rotation_angle`:** *(type int)* the final rotation angle (in degrees).
- **`box_coord`:** *(type array double)* the coordinates of 5 drawn lines after image rotated, i.e. `[y1, y2, x1, x2, x0]`.
- **`img_raw`:** *(type matrix uint16)* the raw TIFF image, height-by-width-by-3, 3 as in (R,G,B) channels. This serves as a backup of the TIFF files.
- **`img_box`** *(type matrix uinit16)* the cropped TIFF image, only showing the spindle, i.e. `[y1:y2, x1:x2, :]` of `img_raw`.

59

`spd_data` entry

- **`CEN_LINE_OFFSET`:** *(type int)* the input argument.
- **`POLE_PORTION`:** *(type double)* the input argument.
- **`data_label`:** *(type cell string)* the label of color channels, i.e. `{'TexRd', 'FITC', 'DAPI'}` as in (R,G,B).
- **`is_bad`:** *(type boolean)* the flag for whether this spindle is considered bad quality.
- If it were "passed" in interactive interface, `is_bad = True (1)`, and no quantitation data (`img_box, rotation_angle, box_coord, data_box, data_line, ratio_box, ratio_line, ratio_box_100, ratio_line_100, plot_x`) is available (all empty).

60

spd_data entry

- **`data_box`** and **`data_line`:** *(type matrix double)* the averaged/projected fluorescence intensity from pole to pole in 3 channels, using box-scan and line-scan respectively; same column order as `data_label`.
- **`ratio_box`** and **`ratio_line`:** *(type array double)* the FITC/TexRd ratio from corresponding projected data, i.e. `data_box(:,2)./data_box(:,1)`.
- **`plot_x`**: *(type array double)* the 0-to-100% normalized x-axis positions, used for plotting ratio_box and ratio_line on 0-100% scale.
- **`ratio_box_100`** and **`ratio_line_100`:** *(type array double)* ratios of `ratio_box_100` and `ratio_line_100` grouped into `NUM_BIN` bins.

61

obj_good

Variables - obj_good

obj_good

1x1 struct with 5 fields

Field	Value	Min	Max
list_chosen	1x4 cell		
ratio_box_mean	1x100 double	0.2085	0.8017
ratio_box_std	1x100 double	0.0300	0.1094
ratio_line_mean	1x100 double	0.1604	0.7205
ratio_line_std	1x100 double	0.0105	0.3181

- **`obj_`** variables are objects that hold data for the result of ratio averaging of a selected set of spidles.
- **`obj_good`** is saved after image analysis automatically. As the name suggests, it is the result of all "good" spindles.
- **`obj_select`** is returned by `spidnle_mat_pick` to reflect the subset the user picked.

62

obj_good

- **`list_chosen`:** *(type cell string)* the list of all chosen spindle IDs of this object. For each individual entry, the 3-digit ID is saved as string, i,e. '001', not (int) 001.
- **`ratio_box_mean`** and **`ratio_line_mean`**: *(type array double)* the averaged ratio trace across all spindles in `list_chosen`, using box-scan or line-scan, respectively.
- **`ratio_box_std`** and **`ratio_line_std`**: *(type array double)* the standard deviation of the averaged ratio trace across all spindles in `list_chosen`, using box-scan or line-scan, respectively.

- Note: `ratio_box_mean,ratio_line_mean, ratio_box_std` and `ratio_line_std` are all of length `NUM_BIN`.

63

Function Description

- Image processing:

  - `spd_data = spindle_read_folder(dir_name, CEN_LINE_OFFSET, POLE_PORTION);`

  Scan through and checks all TIF files in a input folder, generate and match a list of TIF files, call `spindle_box_select` for each spindle.

  - `im_input = spindle_read_TIFF(file_id, file_id2, file_id3);`

  Load 3 TIF files into MATLAB and save as matrix.

64

Function Description

- Image processing:
  - `spd_data = spindle_box_select(file_id, CEN_LINE_OFFSET, POLE_PORTION);`
  Calls `spidnle_read_TIFF` to load image, calls `spindle_draw_box` to interactively define spindle, calls `spindle_quantitate` to project data traces, and builds each `spd_data` entry.
  - `[y1, y2, x1, x2, x0, rot_angle, is_pass] = spindle_draw_box(im_input, rot_angle, file_id, CEN_LINE_OFFSET, POLE_PORTION);`
  Interactive interface for rotation and line drawing.

65

Function Description

- Image processing:
  - `[data_box, data_line] = spindle_quantitate(im_input, rot_angle, coord_xy, CEN_LINE_OFFSET);`
  Quantitates and projects box-scan and line-scan to get 1D fluorescence intensity.
  - `spindle_mat_display(spd_data, list_good);`
  Print snapshot of all “good” spindles in one-page.

66

Function Description

- Data processing:
  - `spindle_mat_analysis(dir_name, CEN_LINE_OFFSET, POLE_PORTION, y_lim, NUM_BIN);`
  Calls `spindle_read_folder` to process all images in folder, calls `spidnle_mat_summary` to print out summary for each spindle, calls `spindle_mat_calc_ratio` to group ratio traces into bins, calls `spindle_mat_plot_ratio` to plot average ratio, and saves log and workspace file.
  - `idx = spindle_mat_find_ID(spd_data, spindle_ID);`
  Finds index of target spindle ID in `spd_data`.

67

Function Description

- Data processing:
  - `spindle_mat_summary(spd_data_sub);`
  Print out one-page summary for a specific spd_data entry, including raw image, lines, intensity plots, and ratio.
  - `spd_data = spindle_mat_calc_ratio(spd_data, list_good);`
  Calculates FITC/TexRd ratio for all “good” entries in `spd_data`.
  - `[spd_data, ratio_box_mean, ratio_box_std, ratio_line_mean, ratio_line_std] = spindle_mat_average_ratio(spd_data, list_good, NUM_BIN)`

68

Function Description

- Data processing:

  Groups ratio trace data points into `NUM_BIN` of bins.

  - `[spd_data, ratio_box_mean, ratio_box_std, ratio_line_mean, ratio_line_std] = spindle_mat_plot_ratio(spd_data, list_plot, y_lim, NUM_BIN, str_title, file_name, dir_name);`

  Plots a selected list of spindles for averaged ratio.

  - `obj_chosen = spindle_mat_pick(spd_data, str_title, file_name, y_lim, NUM_BIN)`

  Let user choose a subset of spindles and calls `spindle_mat_plot_ratio`.

69

Function Description

- `pole_ratios = spindle_mat_divide_pole(spd_data, list_good, POLE_PORTION);`

  Calculates and plots averaged ratio of poles and body regions for a group of spindles.

- `[ratio_poles, std_poles, ratio_body, std_body] = spindle_mat_compare_ratio(dir_name, POLE_PORTION, y_lim)`

  Calculates and plots averaged ratio of poles and body regions for all experimental conditions. Calls `spindle_mat_divide_pole`.

70

Function Description

- For historical reasons, data processing scripts has prefix "`_mat_`". This is due to a previous version processing data output from MetaMorph, with a set of scripts with prefix "`_excel_`". Those are no longer in use.

71