

Analysis of chromosome movement with Chromo

Jesús Pinto Cruz

Materials

- **ImageJ (Fiji):** version 1.53q
- **MorphoLibJ_-1.4.3:** this .jar plugin file must be placed and installed in the ImageJ's plugins folder (e.g. `fiji-win64\Fiji.app\plugins` in windows OS)

Procedure

0. From movie to .csv file

This processing method requires some previous conditions in order to process the cells correctly.

Image requirements:

1. The raw images must be oriented vertically so the major part of the cell's movement is registered in the vertical axis in order to correctly analyze the movement in the Chromo app.

[Image]

[Image]

Image > Transform > Rotate

2. The cell must be as isolated as possible and not touch another cells or particles.
3. The cell can not touch the image border in any of its frames.
4. The signal intensity of the analyzed cell should be homogeneous along the entire series, high variations could cause the mask to be generated incorrectly.
5. The input image that we want to analyze must be a Z projection of our raw image.

These conditions can be met cropping the image manually in the case of multiple cells near to each other, and checking that the cell does not lose a significant amount of signal in any frame along

the time series.

In order to obtain the segmentation and measurements of the cell, we use a Fiji macro (.ijm) with simple code that allows us to segment and obtain some morphological parameters and the position of the center of mass along the time series.

1. ImageJ processing:

Steps:

We use a fixed path structure in order to prevent some pathing errors:

The main folder includes the subfolders: CSV, Mask, RoiMask, Rois and Z_projections.

1. Place the images in the processing folder batchmacro inside the Z_projections folder:

```
script_segmentacion/Z_projections/batchmacro
```

The images inside the batchmacro folder will be processed by the Fiji macro, and the output will be split between the main folders, generating the metrics data in a .csv file for each analyzed cell in the CSV folder.

[Image]

The segmented images will be generated in the RoiMasks folder so we can check if there is any problem with the segmentation.

The Roi analyzed in each frame are saved in the Rois folder, this way we can check if there is any unwanted particle that is affecting our detection and analysis and fix it manually.

2. We open the macro and set the path variable of our main folder:

```
path = "C:/User/script_segmentacion/";
```

[Image]

3. Run the macro.

[Image]

4. In the display window, set both "Input directory" and "Output directory" paths to the same folder, by default this is our main folder with the Z_projections\batchmacro subfolders.

The default format input of the Fiji macro is a .tif file, it can be changed in the macro input window. Possible errors may appear if the extension is .tiff instead of .tif.

It is recommended that even if we clearly see the correct path displayed in the window, we set it

again each time we run the macro.

Once we run the macro, we obtain:

- One .csv file per analyzed particle (*S. pombe* cells in our case) in the CSV folder with the filename "Results_" + filename.
- Masks of Roi for each frame in the time series in the folder RoiMasks.
- Roi data tables in the Rois folder.

The .csv files will be generated in the CSV folder.

2. From .csv file to Chromo

Once we have our .csv files generated by the Fiji macro, we need to process them with a simple R script, called **CSVprocess_standalone.R** in the main folder.

Inside the CSV folder we have our raw .csv files, our R script is going to process the .csv files inside the CSV_batch subfolder.

1. Copy the .csv files generated in the CSV folder to the CSV_batch subfolder.
2. Open the script with RStudio so we can configure it setting some variables values:
3. Change the **path** variable to the folder where our .csv files generated by the Fiji macro are located. e.g.:

```
path <- ("C:\\User\\script_segmentacion\\CSV\\CSV_batch\\")
```

4. Change the **x\$label** value to the name that we want to associate to our cell group, by default the strain name. e.g.:

```
x$label <- "AFA123"
```

5. Change the **csv_name** variable to the desired final name of the .csv with all the cells of the group merged. e.g.:

```
csv_name <- "AFA123.csv"
```

6. Once we have set the variables, we can run the code and the output will be a .csv file that can be used in the Chromo app.

If we want to compare multiple strains or groups, this process has to be repeated and the output .csv file must be merged manually or placing the several output .csvs with all the cells of the group in the CSVmatchmacro folder and running the last part of this script, changing the pattern variable in the line to a common string. e.g.:

```
files <- list.files(ruta, pattern = "Results")`
```

3. CSV processing with Chromo

1. Choose the Large tier instance of the last version of Chromo¹

[Image]

2. Click the **BROWSE** button and select our processed .csv file obtained from the previous R script.

[Image]

3. Check the boxes:

- ☒ Add vectorial and angular velocity
- ☒ Add cumulative distance

4. And select the option:

- ☐ None
- ☒ Last is common and zero
- ☐ Last is common

5. Download the data with the **DOWNLOAD DATA** button on the top side of the page.

[Image]

This .csv file contains the data of the whole time series until the first cell division. If we want to analyze only the horsetail movement (every frame until the cell stops its movement) we can subset our time series with the script `script_horsetail` using this downloaded file from Chromo with the added parameters from point 3.

4. Obtention of horsetail phase with R script

Using the `script_horsetail.R` script:

1. Set the `path` variable to our chosen directory where the .csv file was downloaded. e.g.:

```
path <- "C:\\Users\\CSV\\"
```

¹<https://chromo.cloud/>

2. Set the `chromo_csv_filename` with the name of the previous `.csv` file downloaded with Chromo. e.g.:

```
chromo_csv_filename <- "AFA123_chromo.csv"
```

3. Set the `stop_data_filename` with the desired name for the horsetail/post-horsetail's durations `.csv` file. e.g.:

```
stop_data_filename <- "AFA123_chromo_stop.csv"
```

4. Set the `output_filename` with the desired horsetail time series's name. e.g.:

```
output_filename <- "AFA123_chromo_horsetail.csv"
```

5. Run the script

The `.csv` file generated will have only the horsetail phase of the time series and can be analyzed with Chromo.

If we want to compare multiple strains or groups, this process has to be repeated and the output `.csv` file must be merged manually.

5. Chromo analysis

1. The default settings for the segmentation discovery in the `Segments > Discovery` tabs are:
 - Method: Lavielle
 - Columns to cluster: `cal.speed`
 - Cluster range: 1-4
 - Significance test: Likelihood Ratio
 - P-value adjustment: None

[Image]

2. The default Spectrum settings are:

- Spectrum comparison: 0.06-0.35

[Image]

3. The Distributions default settings are:

- Input column: cal.speed
- Smoothing: 4

[Image]

4. To run the motifs discovery, go to the `Per Cluster` tab, then in the bottom left corner press the **Run motif discovery** button. Motifs settings are by default:

- Variable to analyze: y
- Analysis group: both strains or groups
- Query group: one of the strains or groups
- Window size: 14
- Samples to compute: 1
- Correlation: 0.98
- Motifs to discover: 3

☐ Matrix Profile

☒ Motifs

☐ Discords

Explanation of the code

- From **line 7 to 30** of the script, we crop the time series based on the division frame where the cell enters meiosis II and a second particle appears.
- From **line 36 to 66** we subset some parameters and change their names so the Chromo app can understand them.
- From **line 70 to 89** we merge all the `.csv` files with a common string (default: "Results"), so we get another `.csv` file with all the processed cells.

This final `.csv` is the input for the Chromo app.

The last part of the script (**line 70 to 89**) can be customized so we can merge multiple groups of cells so we can compare them in Chromo, provided that they have a different group label.

Notes:

Make sure there are not any opened images or windows in Fiji before running the macro.

If we want to compare multiple groups of cells, this script must be run each time separately, making sure that the folder only contains those raw `.csv` files that we want to process, since running twice the script on the same files will get some errors.