PixAR : Pixel Based Analysis of Rhythms

[Author: Nikhil |nikhil@wustl.edu|| Last Updated: 23 Nov 2021]

Navigate to:

INTRODUCTION

PixAR is python based and implements pixel-based time series analysis on tif image stacks/movies by incorporating multiple analysis algorithms including MetaCycle , pyBOAT, FFT and Crossover analysis to analyze both stationary and non-stationary time series.

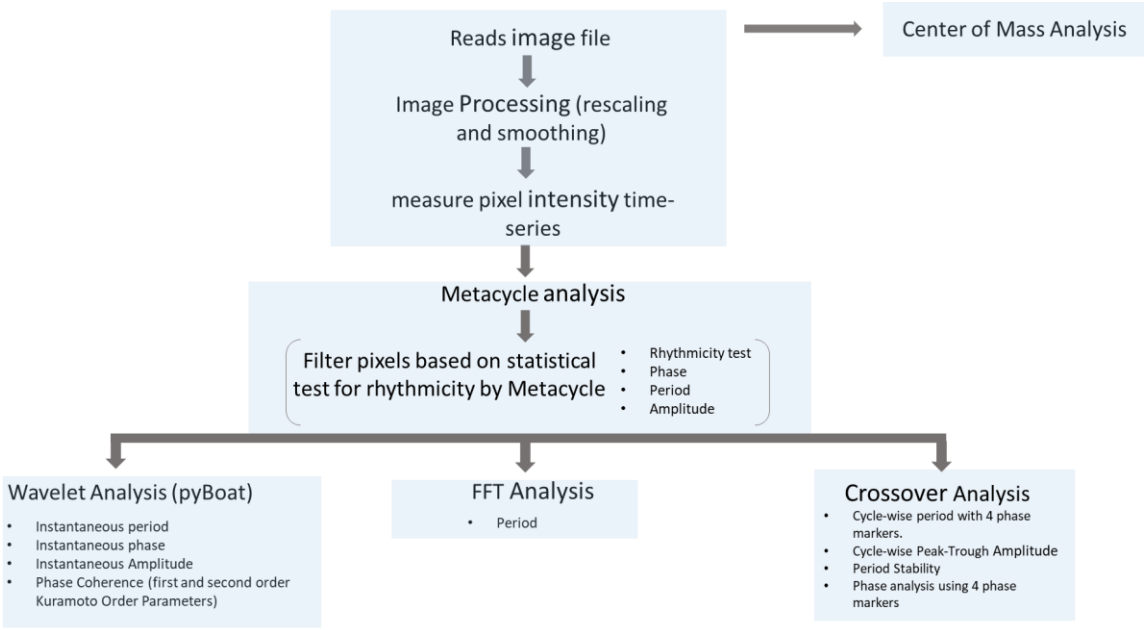
The user can analyse –

a) Rhythm stability and tissue dynamics using Center of mass and its trajectory over time.
b) Periods based on 10 different methods and phase markers.
c) Phases based on 7 different methods and phase markers.
d) Amplitude based on 3 different methods.
e) Instantaneous period, phase, and amplitudes for nonstationary time series.
f) Phase clustering using phase coherence analysis.
g) Phase evolution and phase wave propagation.
h) And visualize data in different ways including phase, period and amplitude maps, clustermaps, histograms, correlation plots and 2d scatter and polar/circular phase evolution plots.

ANALYSIS PIPELINE

There are four categories of analysis

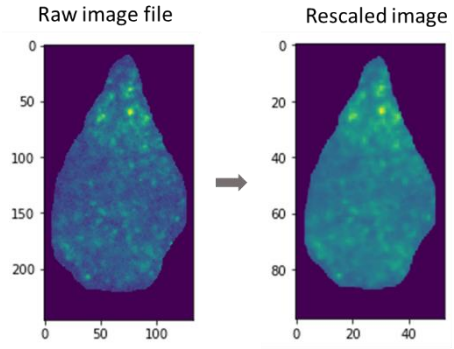
a) Center of Mass Analysis
b) Metacycle Analysis
c) Wavelet Analysis
d) FFT Analysis
e) Crossover Analysis



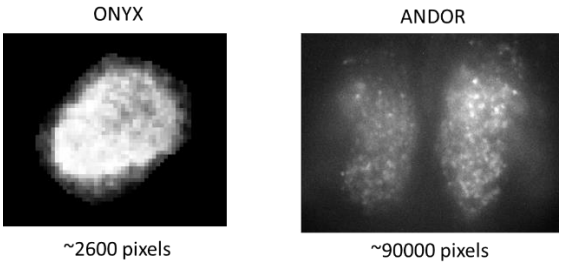
Note: If you are an experienced programmer, you will notice that the style of script writing is not the best because of a) my limited python experience and b) it was not written in one go but developed overtime by adding more modules based on others' feedback. I may improve the script overtime but that would mostly change the style of coding and not the analysis. Happy to collaborate if you wish to help improve the code or add more features.

IMAGE PROCESSING [1: Reading and rescaling image (source)]

When dealing with n-dimensional image analysis, the processing time can increase exponentially with the number of pixels. Rescaling images can speed up processing time by reducing the total pixels to be analyzed without significant loss in spatial or temporal resolution (depending on extent of rescaling).



However, extensive rescaling leads to loss of spatial resolution and so, different image sizes (due to different recording cameras for instance; see image below) may need different extent of rescaling to avoid loss of resolution.



How does the script decide the extent of rescaling factor?

Counts total number of valid pixels (pixels with sum of intensity across timeseries > 0)

⬇

Calculates amount of rescaling required to get ~6000 valid pixels

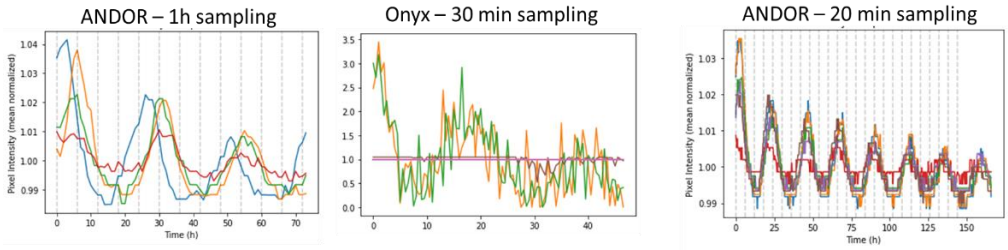
(User can decide how many valid pixels should be analysed.)

Tips to decide pixel number:

- Analysis time increases exponentially with pixel number.
- Reducing the number a lot can lead to extensive rescaling and reduce spatial resolution.

IMAGE PROCESSING [2: Smoothing (source)]

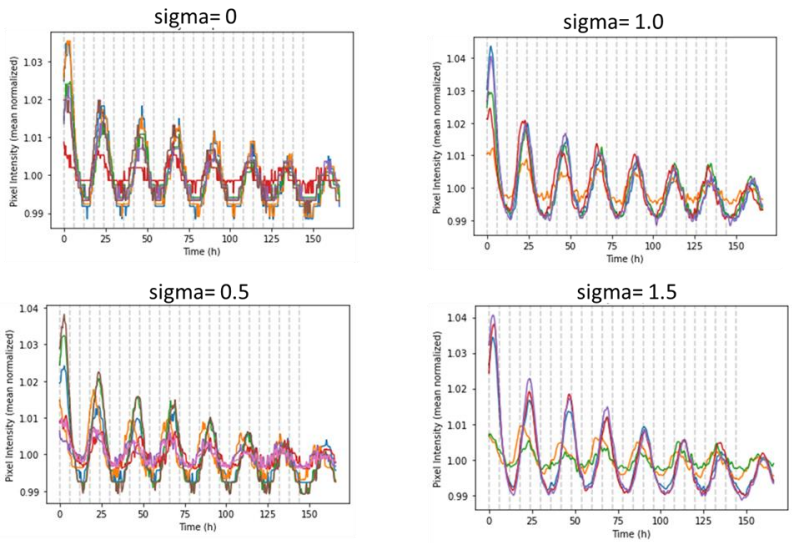
Noise levels in the signal (pixel intensities) can change with a) camera type and b) recording intervals possibly due to different exposure settings and other factors. Below is an example of different noise levels observed in different recordings cameras and intervals.



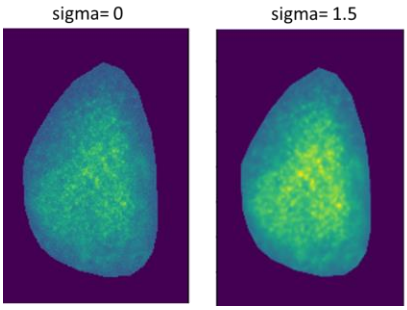
Is smoothing required for any of the analysis that the script implements?

- Metacycle and CWT analysis **does not require** data extensive smoothing.
- Crossover analysis **requires** extensive data smoothing
- COM analysis does not require data smoothing

The script implements Gaussian filter to smooth data. Gaussian filter is parametrized by a sigma value (SD of the Gaussian kernel; somewhat equivalent to the radius of a rolling ball). In other words, sigma non-linearly affects the extent of smoothing. Below is an example of how increasing sigma affects the signal [Note: the traces across images are not from the same pixel. So, traces with color cannot be compared across images].



However, very high sigma values can lead to extensive image blurring and loss of spatial resolution (see image below).



How does the script decide the optimal sigma?

It does not! The default sigma is set to 1 for Metacycle and CWT analysis and 3 for crossover analysis. This works for most cases (based on my limited empirical observation) but the user can change it if required.

Tips to decide smoothing:

- Noise level in the data.
- Mostly values between 1 and 3 works
- Avoid smoothing more than 3 as it leads to severe blurring of data and loss of spatial resolution

CENTER OF MASS ANALYSIS [source]

The script reads individual raw images from the stack and calculates COM [weighted average pixel intensities] for every image. For more details and how to interpret COM, refer to this and this.

Outputs:

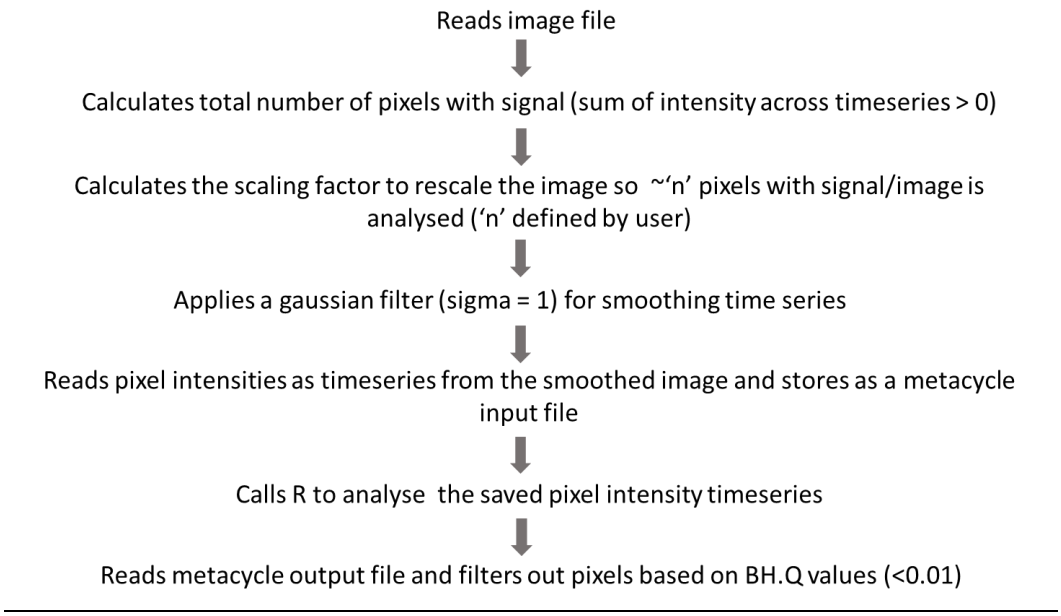
- The X and Y coordinates (in pixels) for every image's COM is saved as 'COM coordinates.csv'.
- Another file – 'COM Image Dimension.csv' is also saved. This has the total X and Y coordinates of the image from which COM is calculated. This may be helpful for some COM trajectory analysis such as percentage of area covered by COM along the time series.
- Plot of the tissue (COM.tif) with the COM marked in red at a user defined time along the recording.
- Plot of the COM trajectory (COM_Trajectory.tif) showing how COM moves with time.
- Image Stack (Stack Plots/COM) with COMs marked for all images in the stack.

METACYCLE ANALYSIS [source]

[Note: This script mainly uses Metacycle to statistically test for rhythmicity.]

The MetaCycle package is used for detecting rhythmic signals from large scale time-series data. Depending on features of each time-series data, MetaCycle incorporates ARSER(ARS), JTK_CYCLE(JTK), and Lomb-Scargle(LS) properly for periodic signal detection, and also output integrated analysis results.

Analysis Pipeline:



Note: Following metacycle analysis, the script uses BH.Q < 0.01 as the criterion to call a pixel rhythmic or not. None of the other timeseries method implemented here have a powerful statistical test for rhythmicity. So, all analysis following metacycle will use timeseries only from pixels that were called rhythmic based on the BH.Q criterion set.

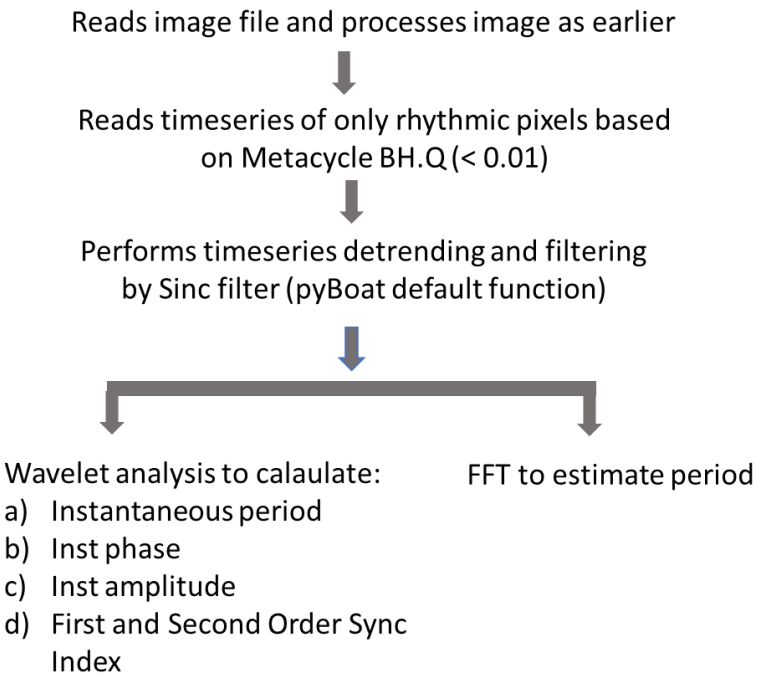
Outputs:

- 'metadata.csv' file which has the timeseries from all pixels and is used as input for metacycle analysis.
- 'metaout' folder which has the output files from metacycle analysis.

WAVELET (CONTINOUS WAVELET TRANSFORMS) and FFT ANALYSIS [source]

The script calls pyBoat functions for wavelet and FFT analyses and uses some default parameters for analysis. However, the script allows you to change some parameters if required. The current settings work well for most cases and may not need to be modified. More details about pyBoat implementation and things that you need to know before changing analysis parameters can be found here.

Analysis Pipeline:



Outputs:

- The following csv files are generated: 'CWT Instantaneous Amplitude', 'CWT Instantaneous Period',
  'CWT Instantaneous Period Power', 'CWT Instantaneous Phase_h', 'CWT Instantaneous Phase_rad'
  'Ensemble Dynamics', 'FFT Period', 'FFT Spectrum' and 'Normalized Traces'
- The following images are generated: 'Ensemble Dynamics' the tissue level dynamics, 'Daywise Amplitude.' showing daywise changes in amplitude, 'Sample CWT Phases' showing phase traces of randomly sampled traces, stack plots of phase evolution, 'period and phase histograms'. See 'Results' section for more details.

CROSSOVER ANALYSIS

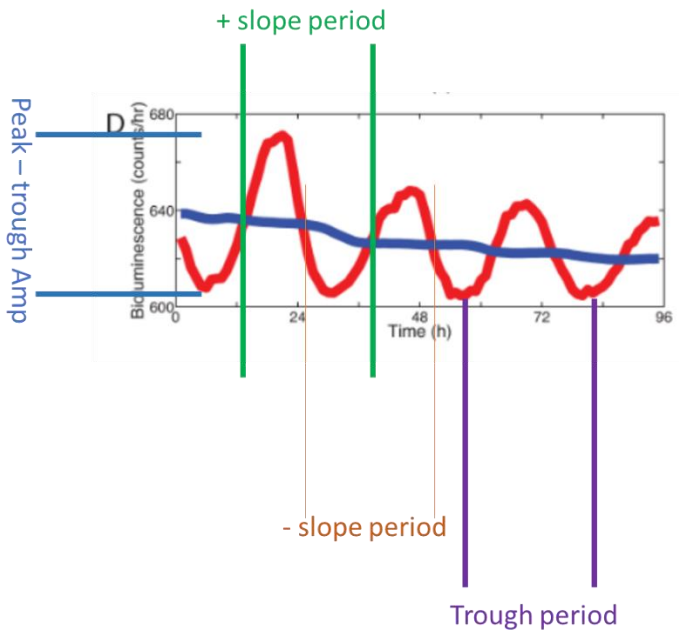
Crossover analysis implemented here is an extension of methods discussed in articles 1 and 2.

Principle (modified after ref 1): Unlike metacycle and FFT, crossover analysis does not assume constancy of period in the data. Crossover approach calculates two running averages from the raw data, with a 3-h (see note below) and a 24-h window, respectively. The 24-h smooth removes non-circadian trends and provides a baseline, while the 3-h smooth reveals the circadian oscillations around this baseline. The period is calculated from the daily intersections of these two lines. However, the method can be sensitive to missing or noisy data, but it is fast, intuitive, and provides details about how the rhythm changes from day to day.

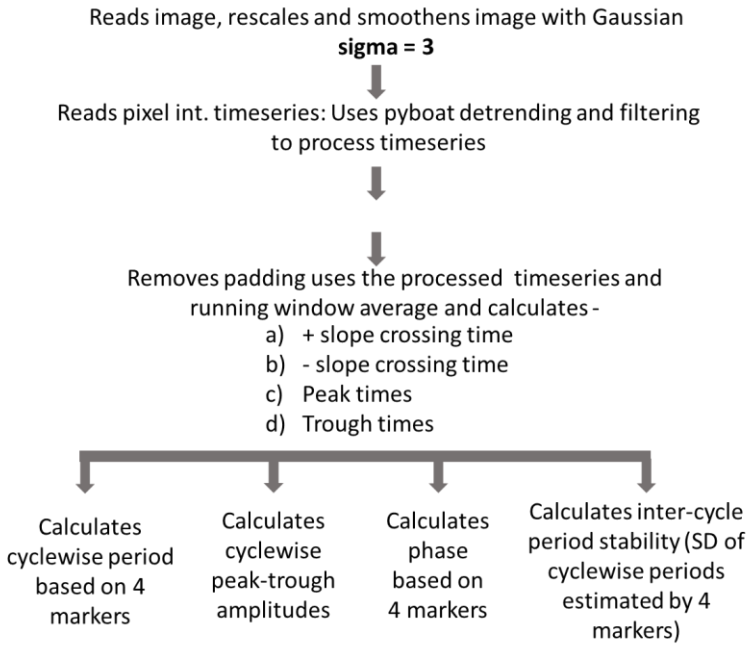
In this script, the 3h running window average is not calculated. Instead, the timeseries is smoothed using a gaussian sigma of 3. In addition, four phase markers are used:

a) Positive Slope Crossing (upward crossing of signal to the 24h average line)
b) Negative Slope Crossing (downward crossing of signal to the 24h average line)
c) Peak
d) Trough

Cycle-wise periods are calculated based on these 4 phase-markers and amplitude is calculated as peak – trough. Phases are calculated every day as the time at which each of the phase marker occurs and is presented on a scale of 0-24 h.



Analysis Pipeline:



Outputs:

- The following csv files are generated: 'Crossover cyclewise amp, 'Crossover negslope per',' Crossover posslope per', 'Crossover peak-peak per', 'Crossover trough-trough per', 'Crossover negslope phase',
  'Crossover posslope phase', 'Crossover peak phase', 'Crossover trough phase'
- The following images are generated: 'period and phase histograms' based on 4 markers, 'daywise amplitude'. See 'Results' section for more details.

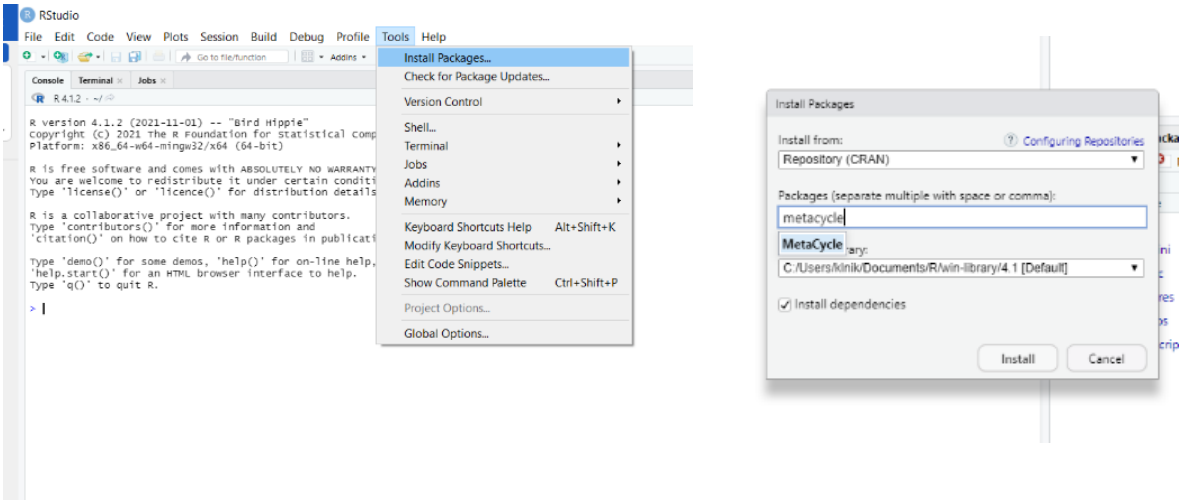
INSTALLATION

Windows:

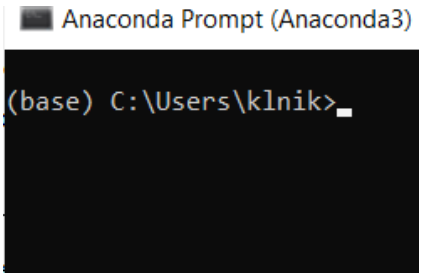
1. Install Anaconda with python version > 3.x. Installation instructions can be found here.

2. Install R and R Studio. Installation instructions can be found here.

3. Install Metacycle using R studio (this is the easier way): Open R studio and navigate to 'Tools > Install Packages'. Search for MetaCycle in 'Packages' and install.



4. Windows users: Copy the pixar_env.yml file (in the 'Read_Me/Install' folder) to "C:/Users/<your username>" folder.



5. After copying the yml file, open 'anaconda prompt' with admin privileges (right click and run as administrator) and you will see a terminal window as shown above but the path will be different. You will have to navigate to the directory where the yml file is saved. To do this, type *cd C:\Users\<your username>*

6. After navigating to the directory, execute the command:
   conda env create -f pixar_env.yml

7. The new pixar environment will be set up. This might take a couple of minutes.

8. After all dependencies are installed execute:
   conda activate pixar

9. You will notice that (base) in the terminal will change to (pixar). You are now in the new environment created for pixar analysis. You will need to install a few more modules. Make sure you have completed step 8 before proceeding.

10. Install some more modules using the following commands:
    pip install pyboat
    pip install rpy2
    pip install pytest-shutil

11. This should set up the environment for pixar analysis. However, if you notice that Jupyter notebook or spyder is not running when you launch it, you can try this tip. Open Anaconda Prompt and execute the following sequentially:

    conda remove qt
    conda remove pyqt
    conda install qt
    conda install pyqt

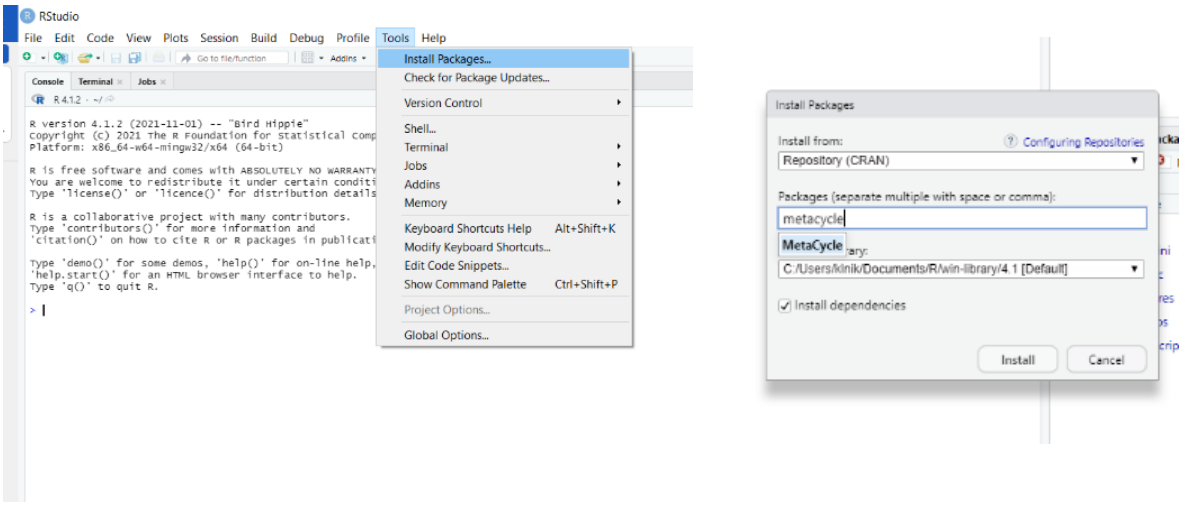
12. You should now be able to use Pixar.

OS X:

1. Install Anaconda with python version > 3.x. Installation instructions can be found here.

2. Install R and R Studio. Installation instructions can be found here. Note that some of the R versions are OSX version and processor type (apple or intel) specific. Download and install the version that is compatible with your system.

3. Install Metacycle using R studio (this is the easier way): Open R studio and navigate to 'Tools > Install Packages'. Search for MetaCycle in 'Packages' and install.



4. Download pixar and locate the pixar_env_osx.yml file within the Read_Me folder.

5. Open Terminal and execute:
   conda env create -f /<path to pixar_env.yml file>/pixar_env_osx.yml

6. This will setup the pixar environment and install all dependencies. This might take a couple of minutes.

7. After all dependencies are installed, execute the following line in the terminal:
   conda activate pixar

8. You will notice that (base) in the terminal will change to (pixar). You are now in the new environment created for pixar analysis. You will need to install a few more modules.

9. Using a text editor, open the file 'meta2d_script.R'. You will find it inside the 'Imports' folder within the downloaded Pixar folder.

10. Check whether the first line is -
    #! /usr/local/bin/Rscript

11. If not, add < #! /usr/local/bin/Rscript > to the first line of the script and save the file. If you see the line already there, close the file without modifying it.

12. Open the terminal and execute the following commands sequentially:
    chmod 755 /<path to file>/meta2d_script.R
    ln -s /<path to meta2d_script.R>/meta2d_script.R /usr/local/bin/Rscript

13. You should now be able to use Pixar.

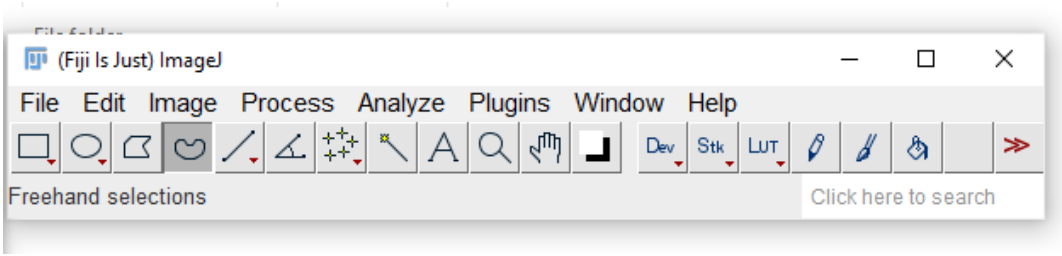
HOW TO USE PixAR?

A) Preparing the movie file

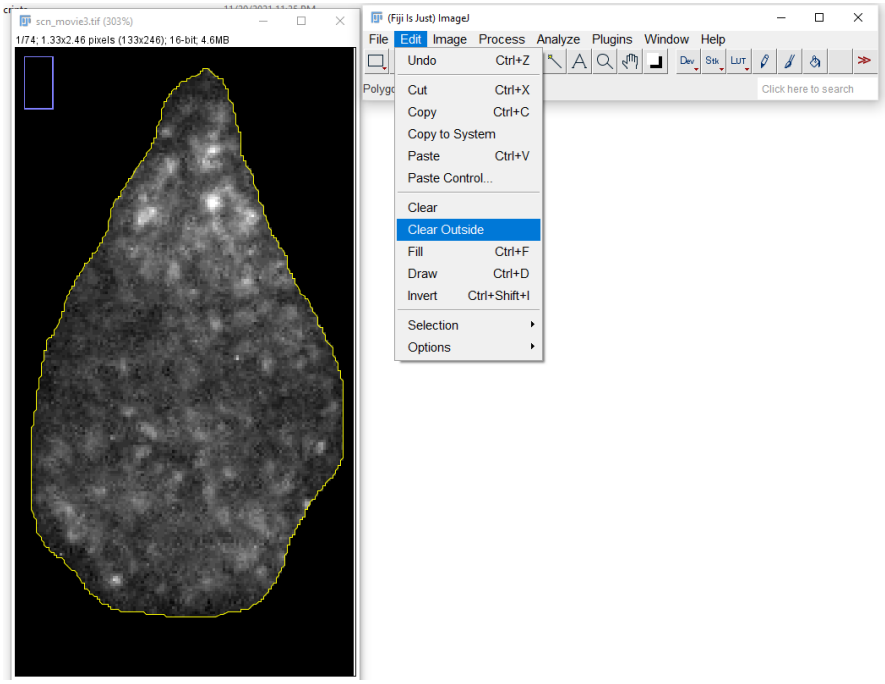
1. Process the movie/stack using adjacent frame minimization.

2. Open movie using ImageJ

3. Using the 'free hand selection' (or any other similar) tool, draw a boundary around the tissue of interest.



4. Click on 'Edit > Clear Outside' [Apply this setting for all frames]. This should set all regions outside the tissue boundary to black. Note: In some cases, the 'clear outside' command turns the outside region white. If this happens, navigate to 'Edit > Options > Colors' and set background color to black. This step helps clear all the unwanted pixels outside the ROI and speeds up computation.



5. Save the movie file in the PixAR directory

B) Running the script

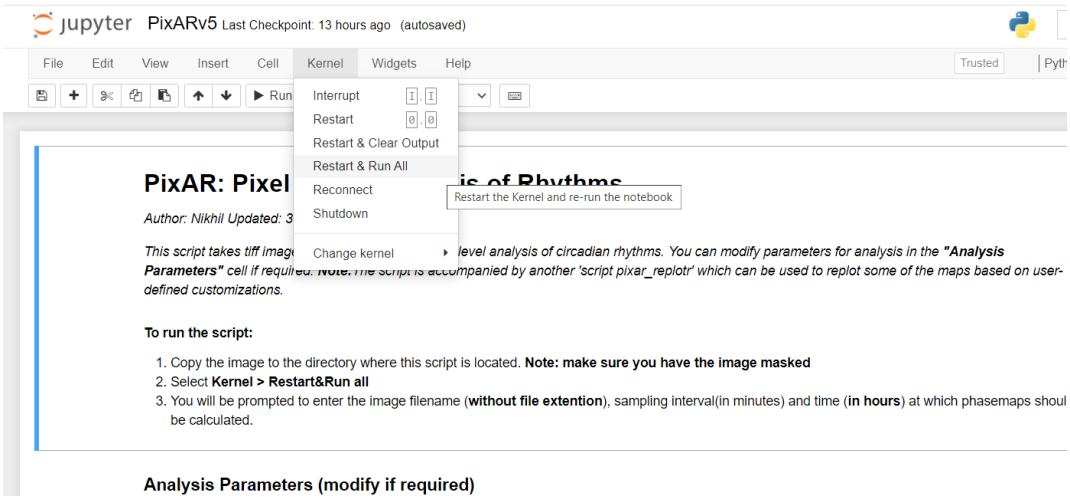
1. Open Anaconda Prompt and type the following commands sequentially:
   conda activate pixar
   jupyter notebook

2. This will open a page on your default web browser

3. Navigate to the PixAR directory and open 'PixAR.ipynb'. This will open the script in a new tab in the browser.

4. If required, at this point, you can change some analysis and plotting parameters.

5. Select 'Kernel > Restart & Run All'.



6. You will be prompted thrice to –
    a) Enter the movie file name. Enter just the file name without the file extension (.tif).
    b) Enter sampling intervals (**in minutes**). Example: Type 10 for 10 min sampling and 60 for 1 hour sampling.
    c) Enter the time (**in hours**) at which you need the phases to be calculated. You can use the two timeseries plots presented just before the third prompt to decide what time do you want the phases to be calculated at. Note: You cannot enter a time beyond the range of your recording or lesser than the sampling interval resolution. For instance, if your sampling intervals is 60 min and for 70 hours, you cannot calculate phase at 71 hours (beyond the range of your recording) or at 20.5 hours (lower resolution than your sampling rate). You can either choose 20 or 21. You will receive a warning message and the script won't be executed until you provide a valid time.

7. After all the three entries, the script will continue, and you will be notified at the end of the run. The results will be saved in 'Results_<your filename>' folder in the PixAR directory.

HOW TO CHANGE ANALYSIS PARAMETERS?

The 'Analysis Parameters' cell in the script lets you change parameters for analysis.

a) How to change period range for analysis?

Default period ranges considered for analysis is 18 h to 30 hours. Changing this range is a 2-step process –

1) Change *min_per* and *max_per* values in the 'Analysis Parameters' cell of the script.
2) In the 'Imports' folder, open 'meta2d_script.R' file using a text editor and change *minper* and *maxper.* Caution: In the meta2d script, the period values are in minutes and not hours.

b) How to speed up analysis time?

Analysis time can be sped up by reducing the number of pixels to be analysed per image. A value of 6000 pixels per image works fine for most cases but can take several hours and sometimes overnight if the tissue has a lot of arrhythmic cells. In such cases, you can reduce analysis time by changing the number of pixels assigned to *pixels_per_image*

Caution: reducing the pixel number can affect your spatial resolution and is not suggested for smaller images. 4000-6000 is a reasonable range based on my experience.

c) How to increase period resolution for wavelet and FFT analysis?

Default number of periods that the authors of wavelet and FFT analysis suggest is 200. This can be changed in the line *nper*.

d) How to change data smoothing?

Extent of gaussian smoothing is set to a default value of Gaussian sigma = 1. If you have extremely noisy data and want to smooth it more, you can change the 'default' value assigned to *smooth* to 2 or 3 (without quotes).

Caution: By default, metacycle and wavelet analysis does not need extensive smoothing and crossover uses gaussian sigma of 3 for smoothing. Also, increased smoothing can lead to loss of spatial resolution and is especially detrimental for smaller image sizes. So, avoid changing this unless it is really needed.

HOW TO CUSTOMIZE PLOTS?

Only maps and clustermap plots can be customized. This can be done in two ways:

1) **Using the main PixAR script before running the analysis** (gives you fewer options compared to the next method)
   Before running the script, you can change values in the 'Plot Customization' cell of PixAR script to modify plotting parameters.

   a) *periodmap_color = "jet"*
   phasemap_color = "hsv"
   Change color maps for the phase, period and amplitude maps. You can find the list of available colors here. Also, if you want a inverted color scheme, add _r to the end of the colormap name. For instance, if you like 'jet' but prefer the colors to be inverted, use 'jet_r'.

   b) *map_background_color = 'black'*
   Change the background color of maps to a different color. List of available colors can be found here.

   c) *maps_minper = 'auto'*
   maps_maxper = 'auto'
   You set min and max periods for period maps. If set to auto, time from your original pixar analysis will be used. If you want the periods to be plotted between 20 and 30 hours, just set min and max values to 20 and 30 (without quotes). Please note that changing these values wont change any of the original analysis. It just modifies the plots.

   d) If you want to use one of the maps for publication, you can change the figure dpi based on journal requirement using the line:
   dpi = 300

2) **Using PixAR Replotr after running the analysis** (gives you more options)
   Refer this section for details.

RESULTS

The table below provides a list of output files along with their description.

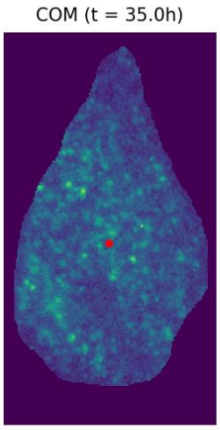
Most files will have X and Y columns which indicate the x and y coordinates of the pixels in the image. The CycID/Pixel is just ID/number assigned to a pixel.	
Output File/ Folder Name	**Description**
cache	Do not delete this folder. It stores some values that are used by replotr script.
metaout	Folder with metacycle analysis output
COM Cordinates	X and Y coordinates of COM for every image in the stack provided.
COM Image Dimension	Total pixels in X and Y dimensions for the image used for COM analysis. Can be used to calculate % of area covered by COM trajectory.
Compiled Amplitude	Amplitude calculated by Metacycle, wavelet (CWT)and crossover (XOV)
	1. Metacyc Amp = relative amplitude of pixels/cells reported by metacycle
	2. CWT Amp = Averaged instantaneous amplitudes across all timepoints excluding 6 h at the beginning and 6 h at the end of the timeseries to avoid edge artifacts.
	3. XOV Amp: peak - trough amplitude averaged across all cycles.
Compiled Period	Periods estimated by all the implemented methods.
	Metacycle per (h) and FFT per (h) : Period of pixels/cells reported by metacycle and FFT respectively
	Avg CWT Per (h): Averaged instantaneous periods across all timepoints excluding 6 h at the beginning and 6 h at the end of the timeseries to avoid edge artifacts.
	Crossover periods: Periods calculated from the 4 phase markers and averaged across cycles.
Compiled Phase	Same as above but for phases.
Crossover cyclewise Amp	Cyclewise peak-trough amplitudes
Crossover negslope per	Cyclewise periods calculated from consecutive negative slope crossing
Crossover posslope per	Cyclewise periods calculated from consecutive positive slope crossing
Crossover peak per	Cyclewise periods calculated from consecutive peaks
Crossover trough per	Cyclewise periods calculated from consecutive troughs
Crossover negslope phase	Time (in hours) of consecutive negative slope crossing.
	Ex: if the column values for a cell/pixel are 10, 30 and 50, the signal crossed the baseline at 10, 30 and 50 hours with time 0 being the first timepoint of the timeseries

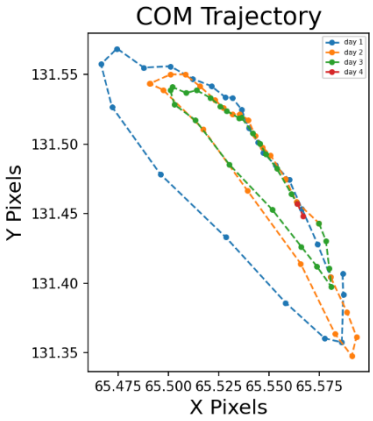
Crossover posslope phase	Time (in hours) of consecutive positive slope crossing.
Crossover peak phase	Time (in hours) of consecutive peaks.
Crossover trough phase	Time (in hours) of consecutive troughs.
CWT Instantaneous Amplitude	Instantaneous amplitude calculated by wavelet analysis
	Note: even though average CWT amplitude is calculated by excluding first and last 6 h, this file stores data for all time points
CWT Instantaneous Period	Instantaneous periods calculated by wavelet analysis
	Note: even though average CWT amplitude is calculated by excluding first and last 6 h, this file stores data for all time points
CWT Instantaneous Period Power	Power spectrum for all 200 periods between min and max periods set by the user. The period with maximum power is used as CWT period in the script. This file stores powers associated with other periods as well.
CWT Instantaneous Phase_h	Instantaneous phases (in hours) calculated by wavelet analysis
CWT Instantaneous Phase_h	Same values as above but in radians instead of hours
Detrended Traces	Detrended timeseries (implemented using pyBoat) after Gaussian smoothing of sigma = 1
Ensemble Dynamics	Instantaneous period and amplitude averaged across all cells/pixels for a given time point.
	Sync Index 1: First order Kuramoto Order Parameter
	Sync Index 2: Second order Kuramoto Order Parameter
FFT Period	Periods calculated from FFT analysis

FFT Spectrum	Power spectrum of all 200 periods between min and max period set by user. Period with max power is used in the script.
metadata	Timeseries data of all pixels in the image (after Gaussian smoothing sigma = 1) used as input for metacycle analysis
NormalizedTraces	Amplitude normalized detrended timeseries which removes damping (implemented using pyBoat).
Percent Rhythmic	Details of percent rhythmic pixels in the analysed image
Smoothed Traces	Smoothes timeseries after Gaussian smoothing of sigma = 3

PLOTS

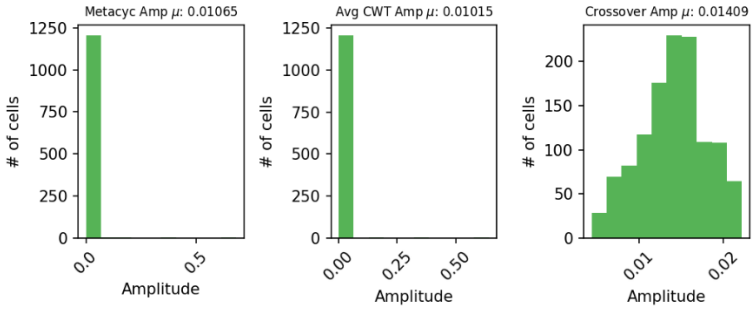
COM and COM trajectory





- Plot of the tissue (COM.tif) with the COM marked in red at a user defined time along the recording.
- Plot of the COM trajectory (COM_Trajectory.tif) showing how COM moves with time and is color coded to depict different days. 'Day' in the plot is a 24 h window and not related to the cell/pixels period.

Amplitude Histograms

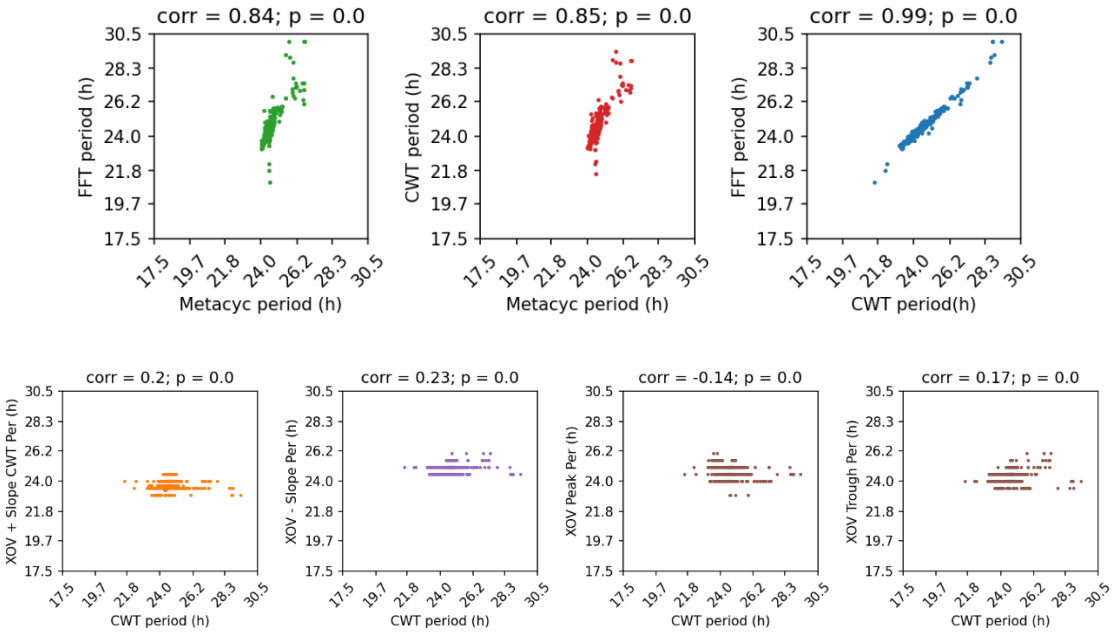


Histograms of amplitudes calculated by different methods.

- Metacyc amplitude is the relative amplitude (relative to mesor) calculated by metacycle.
- Avg CWT Amplitude: Averaged instantaneous amplitudes across all timepoints excluding 6 h at the beginning and 6 h at the end of the timeseries to avoid edge artifacts.
- Crossover Amp: peak - trough amplitude averaged across all cycles.

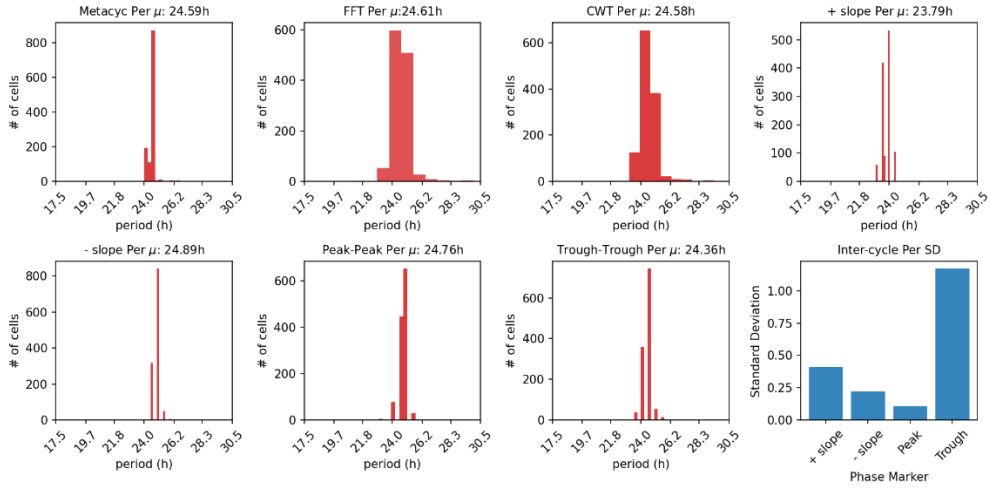
Correlation Plots 1 and 2

- Correlations between periods estimated by different methods. **Corr = pearson correlation coefficient. p = significance level.**
- CWT Period is the period of every cell averaged across all time points excluding 6 h in the beginning and end of time series to avoid edge artifacts.
- XOV periods are periods calculated from respective phase markers and averaged across cycles.



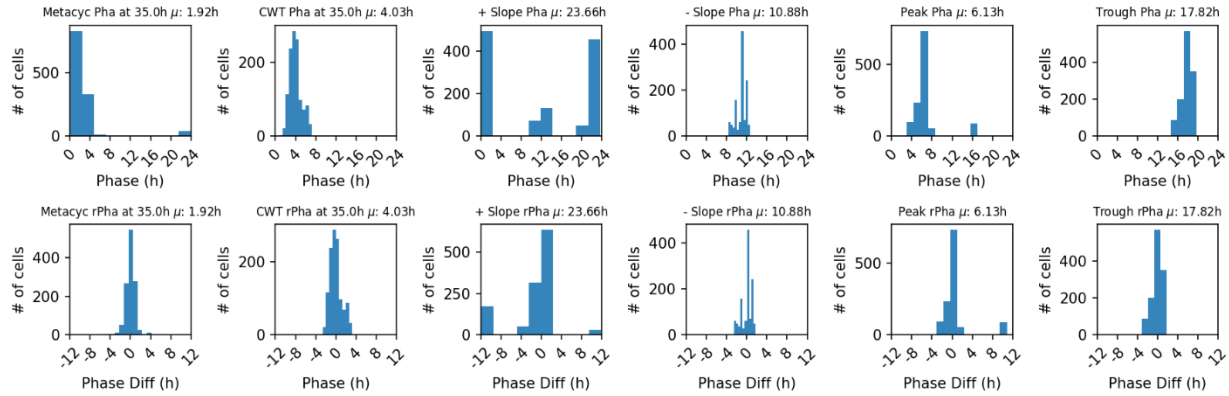
Period Histograms

- Histograms of period calculated by different methods.
- CWT Period is the period of every cell averaged across all time points excluding 6 h in the beginning and end of time series to avoid edge artifacts.
- XOV periods are periods calculated from respective phase markers and averaged across cycles.
- The last plot is SD of inter-cycle periods estimated by the 4 phase markers. The marker with least SD is generally preferred.



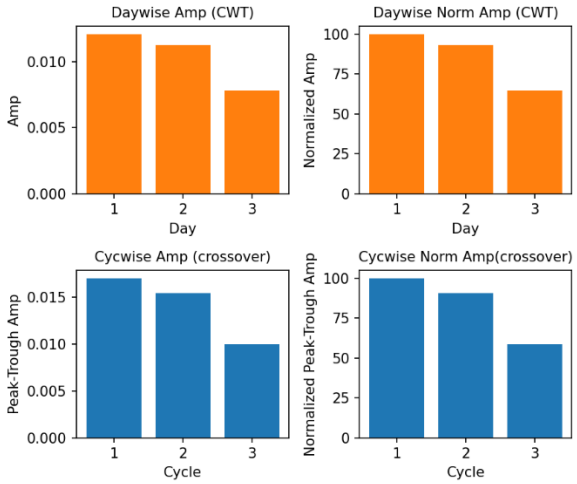
Phase Histograms

- Histograms of phases calculated by different methods.
- Metacycle and CWT phases are calculated for the time provided by the user. Phases by crossover analysis cannot be calculated for a specific time and therefore is calculated as the average of all cycles. See calculations section for more detail.



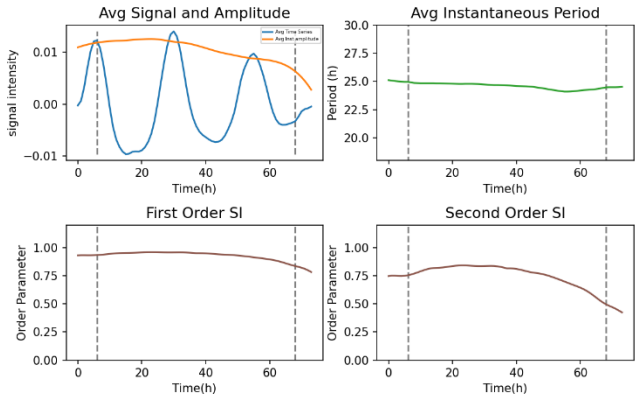
Daywise Amplitude

- Daywise CWT amplitude (top panel) is calculated as average instantaneous amplitude every 24 hours.
- Cyclewise crossover amplitude (bottom panel) is calculated as the peak-trough amplitude between every peak and its successive trough. Depending on the period of the timeseries, sometimes number of cycles for crossover amplitudes (bottom panel) may be different from number of days for CWT amplitude (top panel). Left figures are raw amplitude values and right-side plots are amplitudes normalized to the amplitude on day/cycle1.



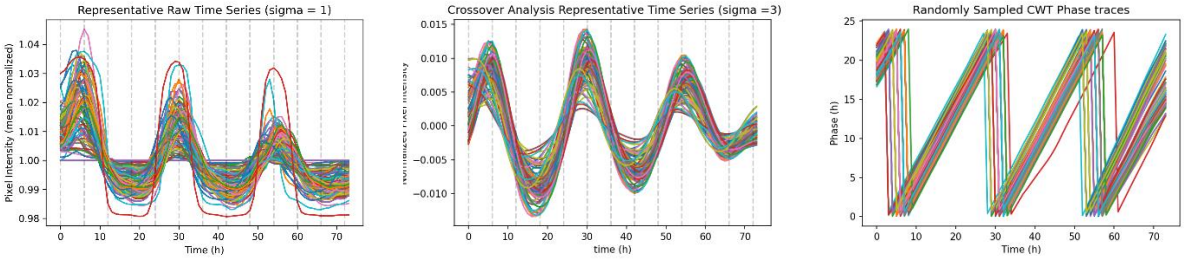
Ensemble Dynamics

- Plots a) averaged timeseries and amplitude b) averaged instantaneous period and c) first and second order Kuramoto order parameters of all cells/pixels.
- The vertical dashed gray lines indicate the 6 h window from the beginning and end of the time series. Averaged CWT amplitude and CWT period presented in other plots are values withing the vertical lines.
- The average instantaneous period plotted here are smoothed using a 6h window. The first and second order parameters are not smoothed.



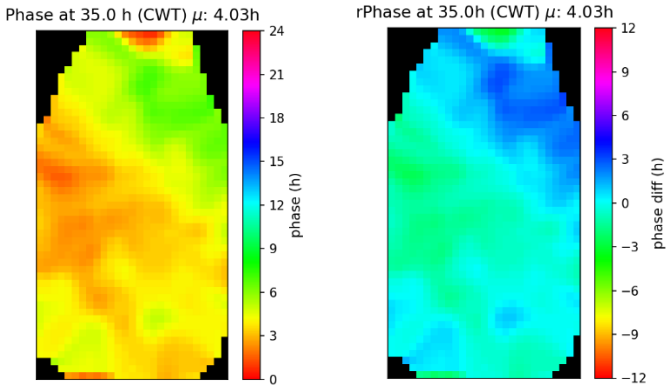
Representative time series plots

- Sample Time series: 100 randomly selected time series after Gaussian smoothing (sigma = 1). This data is presented before metacycle analysis and sometimes you will find arrhythmic cells as well. The purpose is to give the user a sense of how the raw data looks.
- Smoothed Time series: 100 randomly selected time series after Gaussian smoothing (sigma = 3).
- CWT Phase Traces: 100 randomly sampled phase traces to give the user a sense of phase distribution of the tissue across time.



Maps

- Maps folder has 3 subfolders – Period Maps, Amplitude maps and Phase maps.
- Period and amplitude maps plots period values estimated using different methods. Avg CWT period and CWT amplitudes are averaged instantaneous periods and amplitudes excluding first and last 6 hours of the timeseries.
- Crossover periods are averaged cyclewise periods based on respective phase markers.
- Phase maps: Phase maps are same as period and amplitude maps. There are two kinds of phase maps plotted. **Absolute phase and Relative phase.**
- Absolute phase is the actual phase (between 0 h and 24 h) of the cell/pixel at a given time. Relative Phase is the phase of pixel/cell relative to the mean phase of the tissue at that time and is plotted within 12 – 12 h range. So, a cell with a relative phase 4h means that it is 4 hours ahead of the average phase of the tissue at that time.
- While Absolute and Relative phases for metacycle and CWT analysis is plotted based on user chosen time, this cannot be calculated for crossover analysis. Phases calculated based on crossover analysis is the averaged phase across all cycles calculated based on the respective phase marker.



Stack Plots

Stack Plots folder includes different kinds of plots for same data and can are basically different ways of visualizing phase dynamics as a movie.

1) COM
   COM of every image (all time points) in the image stack provided. Can be used to visualize COM trajectory.

2) CWT Abs Phase Evolution Map
   Instantaneous CWT phase values for all timepoints in the timeseries. Can be used to visualize phase waves.

3) CWT Abs Phase Evol Polar
   Same plot as (2) but on a circular scale.

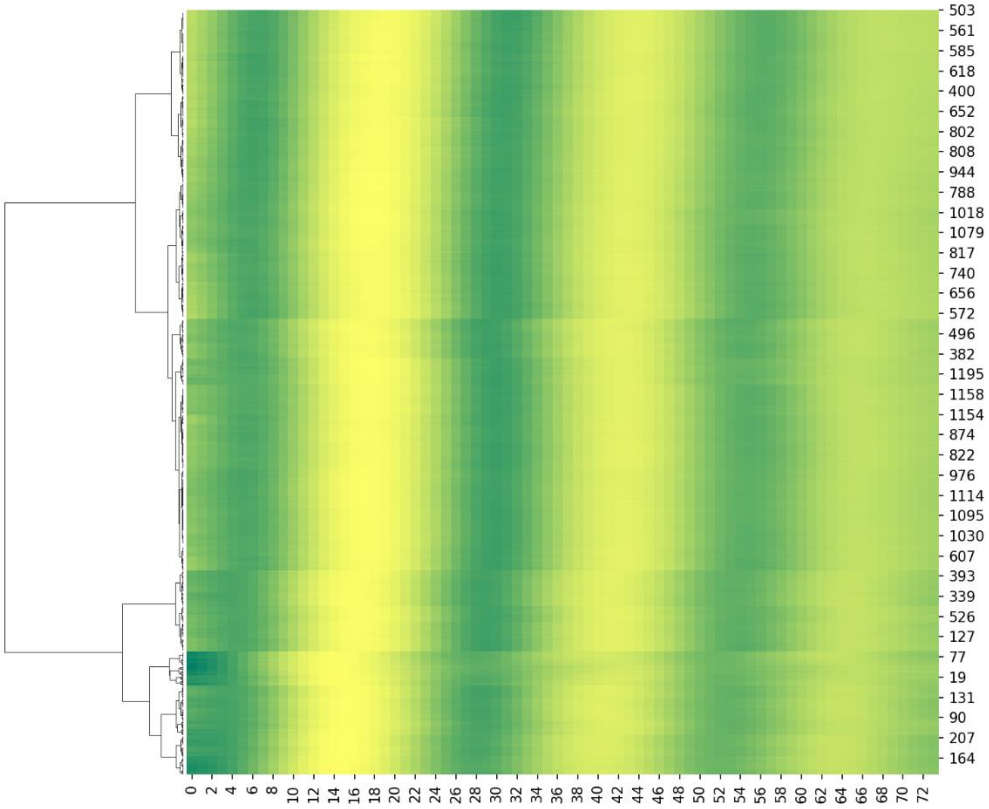
4) CWT Rel Phase Evol Polar
   Same as (3) but plotted as relative phase. In 3, you would see phases move around the circle with time but in this plot, you will notice that the points remain stable around 0 unless there is phase clustering or other sorts of phase dynamics observed in the tissue.

5) Polar histograms
   Same as (3) but values are plotted as histograms on a circular scale.

Note: None of the polar plots have circular statistics analysis implemented. These are just phase values plotted as different projections.

Clustermap

The script automatically implements hierarchical clustering on smoothed traces from all pixels/cells based on some default parameters (you can modify this is Replotr). Can be helpful to preliminarily observe some interesting features in the data.



HOW TO USE PixAR Replotr?

Sometimes, you may want to change the way maps are generated. In such cases, running the entire script again can be time consuming. So, I have included another script called the PixAR Replottr which can be used to replot some of the maps without running the entire script again.

With PixAR_Replotr:

1. **You can modify plots using the 'Plot Customization" cell:**

e) *phasemap_time = 'auto'*
Change time (in h) for phase map plotting if required. If set to auto, time from your original pixar analysis will be used.
To set a different time change *phasemap_time* from *'auto'* to any other value (without quotes)

f) *periodmap_color = "jet"*
phasemap_color = "hsv"
Change color maps for the phase, period and amplitude maps. You can find the list of available colors here. Also, if you want a inverted color scheme, add _r to the end of the colormap name. For instance, if you like 'jet' but prefer the colors to be inverted, use 'jet_r'.

g) *map_background_color = 'black'*
Change the background color of maps to a different color. List of available colors can be found here.

h) *maps_minper = 'auto'*
maps_maxper = 'auto'
You set min and max periods for period maps. If set to auto, time from your original pixar analysis will be used. If you want the periods to be plotted between 20 and 30 hours, just set min and max values to 20 and 30 (without quotes). Please note that changing these values wont change any of the original analysis. It just modifies the plots.

i) You can also set the min and max values for amplitude maps. Using lines below:

cwt_minamp = 'auto' #cwt minimum amp
cwt_maxamp = 'auto' #cwt maximum amp
metacyc_minamp = 'auto' #metacycle minimum amp
metacyc_maxamp = 'auto' #metacycle maximum amp
xov_minamp = 'auto' #crossover minimum amp
xov_maxamp = 'auto' #crossover maximum amp

j) If you want to use one of the maps for publication, you can change the figure dpi based on journal requirement using the line:
dpi = 300

2. You can modify clustermap using the 'Clustermap Customization" cell:

I have set it to default values which in my experience works for most cases. You might want to have a little understanding of clustering before you can decide on the parameters.

a) Choosing a clustering method using this link.
cluster_method= 'average'

b) Choose a distance metric using this link.
distance_metric='correlation'

c) Z-score decides whether or not to calculate z-scores for rows (0) or columns (1). Z scores are: z = (x - mean)/std, so values in each row (column) will get the mean of the row (column) subtracted, then divided by the standard deviation of the row (column). This ensures that each row (column) has mean of 0 and variance of 1.
z_score = 0

d) Standar_scale decides whether to or not to standardize that dimension, meaning for each row or column, subtract the minimum and divide each by its maximum.
standard_scale = None

e) You can change color scheme of the cluster map. Choose any of the colors from here or here.
clustermap_color = "YlGnBu"