

wormshape_viewer/appR 使用法

ライブラリ Shiny, colorRamps, EBImage を使うのでインストールしておく（初回だけに必要な作業）。R の コ ン ソ ー ル か ら `install.packages("shiny")`、`install.packages('colorRamps')`、`install.packages("BiocManager")`、`BiocManager::install("EBImage")`でインストールできるはず。

1) プレートごとにフォルダを用意し、それぞれに、軌跡のデータと輪郭のデータがひとつずつ入った形にしておく。軌跡データファイルの名前は共通にし、輪郭データファイルの名前も同じにする。プレートごとの名前はフォルダで区別する（あるいは同じフォルダにすべてのデータを入れることもできる。TrackAnalyzer 使用法 1.3 参照）。

TrackAnalyzer 使用法 1.3 参照）。

2) TrackAnalyzer1.3.r で軌跡データと輪郭データを読み込む。（TrackAnalyzer 使用法 1.3 参照）

3) ボックスにすべてチェックを入れて **Execute selected commands** を実行。

4) Diffusion simulator データを読み込む。

5) `calc.C` を行う。

6) `calc.Bearing` を行う。

（念のため一度データを.RData 形式で保存してもよい。）

7) 以上の処理によりデータがそろっている状態から、`App_prepare_turns.r` を走らせる。`source` で読み込んだのちに、`App_prepare_turns(turn.select="T1",anglebin=2)`を入力して実行。おわると自動的に `turns.RData` として一部のデータが保存される。このスクリプトは Wormtrack viewer (App.r) で表示するターンイベントを選択するもの。T1 を指定するとピルエットの中の最初のターンだけを表示する。さらに、`anglebin` を指定すると、T1 ターン直後の `Bearing` が一定範囲内のターンだけが表示される。

8) さらに、アプリでの表示を高速化するため、予め線虫の姿勢の情報を解析しておくとい。上と同様に `calc.Roundness.r` を読み込んで `calc.Roundness()`で走らせる。作られたデータ (`Roundness` と `Area`) は `Roundness.RData` に保存される。

9) R コンソールから `save.image()`または”作業スペースの保存”で全データを保存しておくとい。次回以降はこのデータをロードするか、6) で作ったファイルをロードした上で `turns.RData`、`Roundness.RData` をロードして用いる。なお、後者がなくても、表示に時間がかかるだけで問題はない。

10) データがロードされている状態で、作業フォルダ上に `App.r` が存在する状態で R コンソールから `shiny::runApp()`と入力するとビューアが起動する。

11) ターンの番号（最初は1）が左上のボックスに表示される。表示させたいターンの番号を入力することができる。（新しいターンの表示には少し時間がかかる）。あるいは、次の番号、あるいは前の番号を `next`, `previous` ボタンで選択する。`random` を押すとランダムに選択される。