PySyncDyn: Python Package for Synchronous Dynamics maps from CPMG Relaxation Dispersion Data

Manu Veliparambil Subrahmanian
(mvelipar@umn.edu)

February 11, 2025

Quick Notes

- **User Input:**
  - Edit `UserInput.ini` to set your protein name, experimental list file paths, CPMG parameters, and other analysis options.
  - Ensure all required fields (e.g., `ProteinName`, `list_folder`, `list_files`, `vcpmg`, and `tcpmg_ms`) are correctly configured.
- **Running the Workflow:**
  - Open a terminal in the PySyncDyn directory.
  - Run the main workflow using:

```
python GenSyncDyn.py -i UserInput.ini
```

  - This will sequentially execute:
    1. `GenR2Eff.py` (calculates R_2^{eff} values from list files)
    2. `PairFit.py` (performs pairwise fitting using the Carver-Richards model)
    3. `PlotSyncDyn.py` (generates dynamic correlation maps)
- **SyncDyn Score Calculation:**
  - The score for each residue is computed as:

$$\text{SyncDyn Score}_i = \frac{N_i - 1}{N_{\text{total}}},$$

    where N_i is the number of correlations (above a set threshold) in the i^{th} row of the correlation matrix, and N_{total} is the total number of residues.
  - The score is saved as a CSV file (e.g., `Score_R_squared_min_Caxis0.85-1.0_kex0-5000_full`
- **Visualizing in PyMOL:**

- Configure `PymolParams.ini` with the correct Score file location, PDB ID, and visualization settings.
- Run:

```
python Score2Pymol.py -i PymolParams.ini
```

- This will launch PyMOL, apply the score-based color mapping, and save both a PyMOL session file and a PNG image.

- **Dependencies:**
  - Ensure Python 3.6+ is installed.
  - Install required packages with:

```
pip install pandas numpy scipy matplotlib seaborn openpyxl tqdm numba
```

Contents

1 Introduction

Carr-Purcell-Meiboom-Gill Relaxation Dispersion (CPMG-RD) experiments are highly effective for probing micro- to millisecond conformational exchange processes in proteins. By performing experiments at multiple magnetic field strengths (B_0), one can extract dynamic parameters such as exchange rates, population fractions, and chemical shift differences.

PySyncDyn is a comprehensive Python-based toolkit that automates the entire workflow from raw data processing to the generation of Dynamic Correlation (SyncDyn) maps. The workflow includes the calculation of effective transverse relaxation rates (R_2^{eff}), pairwise fitting using the Carver-Richards model, generation of correlation maps, and computation of a SyncDyn Score that quantifies the extent of correlated dynamics across the protein. In addition, the `Score2Pymol.py` script allows visualization of these scores on the three-dimensional structure of the protein in PyMOL.

Figure 1 provides an overview of the workflow.

2 Workflow Overview

The PySyncDyn workflow comprises the following steps:

1. **CPMG-RD Data Acquisition:** List files generated by resonance assignment software contain intensity and signal-to-noise data at various CPMG frequencies.

2. **Calculation of R_2^{eff}:** The script `GenR2Eff.py` processes the list files to compute R_2^{eff} values and their uncertainties using the formula:

$$R_2^{\text{eff}} = -\frac{\ln\left(\frac{I}{I_0}\right)}{t_{\text{cpmg}}}, \tag{1}$$

   where I is the intensity at a given CPMG frequency, I_0 is the reference intensity (at $\nu_{\text{cpmg}} = 0$), and t_{cpmg} is the total CPMG evolution time.

3. **Pairwise Fitting:** `PairFit.py` applies the Carver-Richards model to perform pairwise fitting of the R_2^{eff} data, yielding kinetic parameters such as rate constants and population fractions.

4. **Dynamic Correlation Mapping:** `PlotSyncDyn.py` (using the `PlotPairParameterNew` function) generates a symmetric heatmap of a chosen correlation parameter (e.g., R^2_{min}) for all residue pairs.

5. **SyncDyn Score Calculation:** For each residue, the SyncDyn Score is computed as:

$$\text{SyncDyn Score}_i = \frac{N_i - 1}{N_{\text{total}}},$$

   where N_i is the number of correlation values in the i^{th} row that meet or exceed a threshold T (with the self-correlation subtracted), and N_{total} is the total number of residues. The scores are saved in a CSV file.

6. **Visualization in PyMOL:** `Score2Pymol.py` loads the score CSV file and applies a custom colormap to the protein structure in PyMOL, saving both a session file and a high-quality image.

3 User Input File (`UserInput.ini`)

The analysis is configured via the `UserInput.ini` file, which includes the following sections:

3.1 General Configuration

```
[General]
ProteinName = RNaseA
```

- **ProteinName** is used to name output directories and files.

3.2 Experimental Configurations

Each magnetic field strength (e.g., `[600]`, `[800]`) specifies:

```
[600]
list_folder = /path/to/list/files/600
list_files =
    RNaseA-T17A_15N-CPMG_5D2O_25C_600_00.000ms_Final.sp.list,
    RNaseA-T17A_15N-CPMG_5D2O_25C_600_00.625ms_Final.sp.list,
    ...
vcpmg = 0,1600,1400,1400,1000,800,600,500,400,400,300,200,100
tcpmg_ms = 40
```

- **list_folder**: Directory containing the list files.
- **list_files**: Comma-separated names of the list files.
- **vcpmg**: CPMG frequencies (Hz) corresponding to each list file.
- **tcpmg_ms**: Total CPMG evolution time (ms).

3.3 PairFit Configuration

```
[PairFit]
save_plot = False
save_plot_type = pdf
Chi2Plot = False
Chi2_PlotVar = kab
ComputeCores = 4
```

- **save_plot**: If `True`, saves the fitting plots.
- **save_plot_type**: File format for saving plots (e.g., pdf, png).
- **Chi2Plot**: If `True`, generates a chi-squared plot for the specified parameter.
- **Chi2_PlotVar**: Parameter name for generating the chi-squared plot.
- **ComputeCores**: Number of CPU cores for parallel processing.

3.4 PlotSyncDyn Configuration

This section controls the generation of SyncDyn maps using `PlotPairParameterNew`.

```
[PlotSyncDyn]
PlotPairParameterNew_Parameter2Plot = R_squared_min
PlotPairParameterNew_Colors_Order = white, red
PlotPairParameterNew_Colors_Value = 0,1
PlotPairParameterNew_Caxis = 0,1
PlotPairParameterNew_PlotXRegionFromTo = 1,50
PlotPairParameterNew_PlotYRegionFromTo = 1,50
save_plot_type = pdf
```

3.5 RunOptions Configuration

This section specifies which scripts are executed. The details for each option are as follows:

```
[RunOptions]
run_GenR2Eff = True    ; When True, executes GenR2Eff.py to calculate R2Eff
   from list files.
run_PairFit = True     ; When True, executes PairFit.py to perform pairwise
   fitting using the Carver-Richards model.
run_PlotSyncDyn = True  ; When True, executes PlotSyncDyn.py to generate
   dynamic correlation maps.
```

4 Detailed Script Descriptions

4.1 `GenR2Eff.py`

This script processes list files from CPMG-RD experiments, calculates R_2^{eff} (using the formula below) and its uncertainty, and saves the results in Excel workbooks.

$$R_2^{\text{eff}} = -\frac{\ln\left(\frac{I}{I_0}\right)}{t_{\text{cpmg}}}$$

where I is the measured intensity and I_0 is the reference intensity.

4.2 `PairFit.py`

This script performs pairwise fitting using the Carver-Richards model. It:

- Loads assignment data from Excel workbooks.
- Uses a Numba-accelerated function to simulate R_2^{eff} values.
- Applies least squares optimization to extract kinetic parameters.
- Saves the fitting results to Excel workbooks.

4.3 `PlotSyncDyn.py`

This script generates dynamic correlation maps. The `PlotPairParameterNew` function creates a symmetric heatmap of the selected parameter (e.g., R^2_{min}) with user-defined filtering and color mapping. It can also export the correlation matrix as a CSV file.

4.4 SyncDyn Score Calculation

The SyncDyn Score is computed for each residue as:

$$\text{SyncDyn Score}_i = \frac{N_i - 1}{N_{\text{total}}},$$

where N_i is the number of correlation values in the i^{th} row that are greater than or equal to a threshold T (excluding the self-correlation) and N_{total} is the total number of residues. The scores are saved in a CSV file for further analysis.

4.5 `Score2Pymol.py`

This script visualizes the SyncDyn Score in PyMOL by:

- Reading the score CSV file and PyMOL parameters from `PymolParams.ini`.
- Loading the specified PDB file (or fetching it if not available locally).
- Mapping each residues score to an RGB color using a custom colormap.
- Coloring the protein structure and optionally adding labels.
- Saving a PyMOL session file (`.pse`) and a high-quality PNG image.

4.6 `GenSyncDyn.py`

This is the main orchestrating script. It reads `UserInput.ini` and sequentially executes:

- `GenR2Eff.py` (to calculate R_2^{eff}),
- `PairFit.py` (to perform pairwise fitting), and
- `PlotSyncDyn.py` (to generate SyncDyn maps).

The output files are organized in a directory named after the protein (e.g., `RNaseA/`).

5 How to Run PySyncDyn

5.1 Prerequisites

Ensure that Python (version 3.6 or higher) and the following packages are installed:

- numpy: For numerical computations
- pandas: For data manipulation and Excel file handling

- matplotlib: For plotting and visualization
- seaborn: For statistical data visualization
- scipy: For scientific computations and optimization
- numba: For performance optimization
- tqdm: For progress bars
- configparser: Built-in Python library for reading configuration files
- openpyxl: For reading/writing Excel files (.xlsx)
- xlsxwriter: For creating Excel files with advanced formatting
- scikit-learn: For Gaussian Mixture Model fitting in distribution analysis

Install dependencies via:

```
pip install numpy pandas matplotlib seaborn scipy numba tqdm openpyxl
    xlsxwriter scikit-learn
```

For PyMOL installation (required for structure visualization):

- **macOS:**

```
conda install -c conda-forge pymol-open-source
# or
brew install pymol

```

- **Linux (Ubuntu/Debian):**

```
sudo apt-get install pymol
# or
conda install -c conda-forge pymol-open-source

```

- **Windows:**

```
conda install -c conda-forge pymol-open-source

```

Note: The recommended approach is to use conda with the conda-forge channel. First install Miniconda or Anaconda, then install PyMOL using conda.

5.2 Visualizing Scores in PyMOL

1. **Important:** Before running Score2Pymol.py, ensure PyMOL is properly installed and accessible from your system's command line. You can verify this by running `pymol --version` in your terminal. If this command fails, PyMOL is either not installed or not properly added to your system's PATH.

2. **Configure `PymolParams.ini`:** Specify the Score CSV file location, PDB identifier, and visualization options.

3. **Run `Score2Pymol.py`:** Execute:

```
python Score2Pymol.py -i PymolParams.ini

```

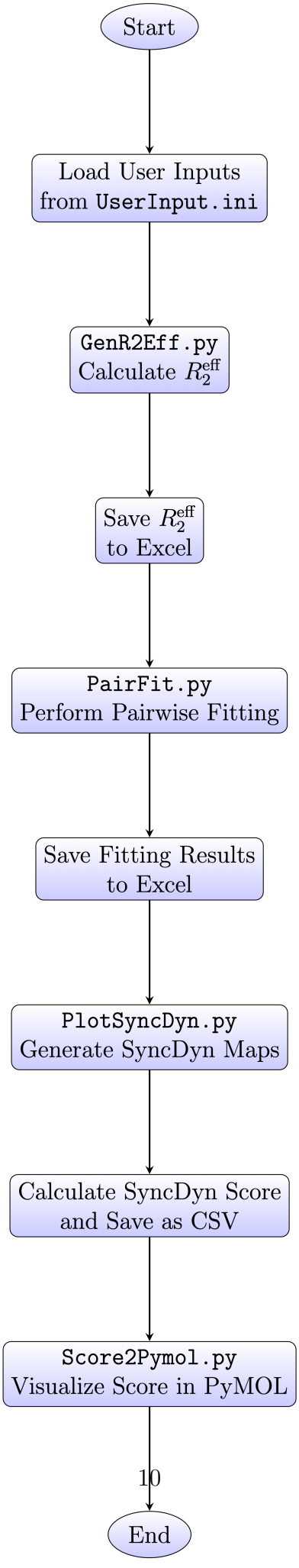
   This script will launch PyMOL in quiet mode, load the structure, apply the score-based color mapping, and save both a PyMOL session file and a PNG image.

4. If you encounter any PyMOL-related errors, verify that:

   - PyMOL is installed correctly using one of the methods above
   - PyMOL is accessible from your command line
   - You have appropriate permissions to execute PyMOL
   - Your system meets PyMOL's graphics requirements

6 Conclusion

PySyncDyn streamlines the analysis of CPMG-RD data by automating the calculation of R_2^{eff}, pairwise fitting of residues, generation of dynamic correlation maps, and computation of a SyncDyn Score. The integrated `Score2Pymol.py` script further enables direct visualization of these scores on the three-dimensional protein structure using PyMOL. By following the quick cheat codes and detailed instructions provided in this manual, users can efficiently obtain comprehensive reports and high-quality visualizations to enhance the understanding of protein dynamics.


Start
Load User Inputs from UserInput.ini
GenR2Eff.py
Calculate R_2^{eff}
Save R_2^{eff} to Excel
PairFit.py
Perform Pairwise Fitting
Save Fitting Results to Excel
PlotSyncDyn.py
Generate SyncDyn Maps
Calculate SyncDyn Score and Save as CSV
Score2Pymol.py
Visualize Score in PyMOL
End