A dynamic functional connectivity toolbox for multiverse analysis

Micha Burkhardt[1,*] **and Carsten Gießing**[1]

[1]Carl von Ossietzky University Oldenburg, Department of Psychology, 26121 Oldenburg
[*]micha.burkhardt@uol.de

ABSTRACT

Various methods for estimating dynamic functional connectivity from fMRI data and subsequent analyses through graph theoretic approaches have been introduced in recent years. But with many of the ground truths unknown, researchers are often faced with arbitrary yet defensible decisions for their analyses, which raises concerns about the reproducibility of results. We here aim to address this issue through the implementation of a wide array of dynamic functional connectivity methods in a unified Python software package, facilitating a diverse exploration of brain dynamics. Anchored in the framework of multiverse analysis, the present work further introduces a workflow for the systematic examination of various methodological decisions. The developed toolbox is supplemented by a graphical user interface for easy usability and accessibility if usage outside of a script-based Python analysis pipeline is desired. Further, extensive demo scripts are provided for researchers to easily adapt this approach for their own analyses.

Keywords: toolbox, dynamic functional connectivity, multiverse analysis, graph analysis, fMRI

1 Introduction

We introduce COMET (COnnectivity METhods toolbox), a user-friendly and efficient toolbox tailored to compute a variety of methodological approaches for estimating dynamic functional connectivity (dFC). In contrast to other packages, COMET is not limited to dFC estimation, but allows for the results of these approaches to be used for multiverse analysis. This enables researchers to systematically assess the variability in results caused by different dFC approaches and the multitude of further decisions made throughout an entire analysis pipeline. In addition, COMET includes specialized tools for graph-theoretical analyses, providing researchers with a broad array of analysis options. However, the multiverse analysis module of the toolbox is not restricted to dFC and graph-theoretical methods, but also enables multiverse analyses in other areas of research independent of the specific research question.

The tools included in the toolbox are particularly relevant as the estimation of dFC is still a developing field of research that promises to allow for novel insights into the temporally fluctuating pattern of brain interactions. This has, among others, led to new findings about various neurological and psychiatric disorders including Alzheimer's disease, schizophrenia, and mood disorders (Calhoun et al., 2014; Gonzalez-Castillo and Bandettini, 2018). Further, methods of estimating dFC also provide a framework to describe the dynamic nature of brain networks in response to environmental demands and internal cognitive shifts (Hutchison et al., 2013). As the field of dFC is a fledging area of research, many different methodological approaches have been proposed over the last few years (Hutchison et al., 2013; Preti, Bolton, and Van De Ville, 2017; Lurie et al., 2020). However, each

of these methods has its own strengths and weaknesses, with none of them being able to exploit the entire richness of the data. This raises questions regarding the robustness of the results, as different algorithms for calculating dynamic connectivity can lead to substantially different outcomes.

This general issue of reproducibility extends beyond the methods used for estimating dFC and has been a topic of ongoing discussion within the broader scientific community for many years (Ioannidis, 2005; Simmons, Nelson, and Simonsohn, 2011; Frias-Navarro et al., 2020). Previous research compared different methods by their test-retest reliability (Choe et al., 2017) or by evaluating their performance using simulated data (Thompson et al., 2018). Despite these efforts, the replication crisis is still present and possible solutions are increasingly being discussed (Fanelli, 2018; Frias-Navarro et al., 2020). In neuroimaging studies, this problem is particularly evident as neuronal data is noisy and requires a large amount of preprocessing and complex statistical analyses. This can lead to a considerable amount of variation in the results for the same set of hypotheses, depending on the chosen methods (Botvinik-Nezer et al., 2020). The present work aims to address these issues through the framework of multiverse analysis. As a recently proposed tool to improve robustness and transparency in data analysis (Steegen et al., 2016), the basic idea behind multiverse analysis is an aggregated evaluation and reporting of all possible combinations of arbitrary yet defensible decisions made in data construction and model building. In contrary to the selective reporting of only one analysis, this allows for an in-depth mapping of interrelations between analysis pipelines and provides an intuitive overview over the robustness of effects (Dafflon et al., 2022). With dFC methods being introduced at a high frequency and many of the ground truths being unknown (Choe et al., 2017; Thompson et al., 2018), these methods are an ideal subject for multiverse analysis.

COMET addresses the aforementioned issues by providing an open and versatile software tool for the estimation of dFC and multiverse analysis. A wide array of methods is provided to be used on their own or in combination with multiverse analysis. The following sections first introduce the software package and the included dFC methods. This is followed by a brief overview of how these methods can be integrated into a multiverse- and graph-based network analyses. Additionally, a concise usage guide and practical examples of using the software are presented, highlighting its current capabilities in comparison to existing solutions. We emphasize that active development is underway, ensuring regular updates and expansions in the near future to further enhance its utility.

2 The COMET Toolbox

The main contribution of the present work is a comprehensive open-source Python toolbox for the estimation of dFC. Users can access its functions through a graphical user interface (Figure 1) as well as a standard scripting API if more complex analyses are required. The software package is further extended by tools for performing multiverse analysis in combination with graph theoretical analyses, but is not limited to this. All software, installation instructions, and usage examples are available at `https://github.com/mibur1/dfc-multiverse`. The following subsections will provide a brief overview of the included methods.

2.1 Dynamic Functional Connectivity

In a systematic literature review, 372 papers were screened for their connectivity methods. For a detailed description of the review process, please refer to the figures in the supplementary material. The extracted methods can be broadly divided into two distinct classes at a higher level. The first class consists of *continuously varying* dFC methods, which provide a dFC estimate for each time point in the data. The second class includes *state-based* methods, which treat dFC as composed of a

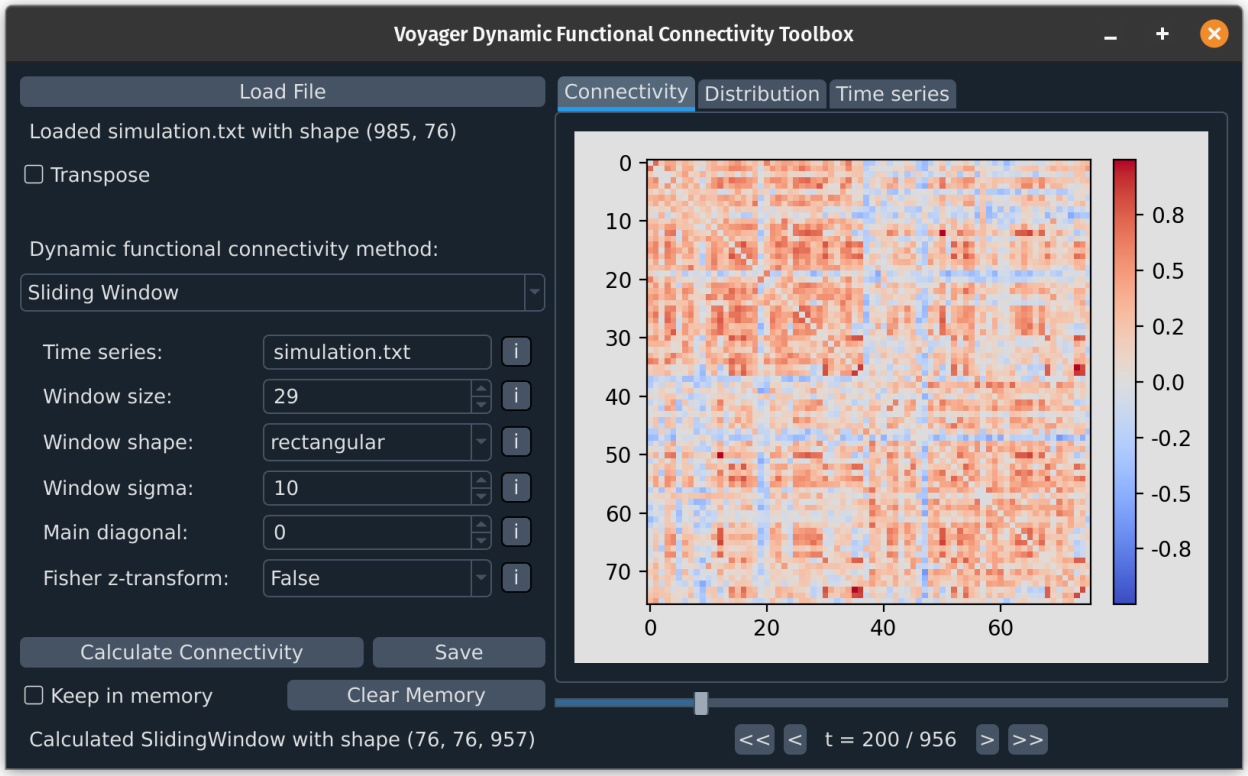

Figure 1. Graphical user interface. In addition to scripting, COMET also includes a graphical user interface for calculating dynamic functional connectivity and quick visualization purposes.

finite set of states and assume connectivity to be static within these states. Currently, COMET only includes continuously varying dFC methods, with the additional constraint that each of the included methods was used in more than one of the reviewed studies.

The included continuously varying dFC methods can further be separated into two sub-classes, namely *amplitude-based* and *time-frequency based* methods. On the one hand, amplitude-based methods operate in the time domain of the signal and derive their connectivity estimates directly from the amplitude of BOLD time series data. On the other hand, time-frequency based methods take frequency or phase information of the time series into account, providing insight into the temporal synchronization and rhythmic coordination between different brain regions (see e.g. Lurie et al. (2020) for a review of time-frequency but also amplitude-based approaches). Six amplitude-based methods and three time-frequency based methods are currently included in the toolbox and will be briefly introduced in the following subsections. An overview of all included methods is given in Table 1.

Sliding Window Correlation

To date, sliding window correlation (SWC) is the predominant method for the assessment of dFC (Shakil, Lee, and Keilholz, 2016; Lurie et al., 2020). The SWC method works by sliding a window with a specified size, shape, and step size over the time series and calculates the pairwise Pearson correlation between the resulting subsets of time series. Given two time series $x(t)$ and $y(t)$, the SWC for a window i corresponds to the Pearson correlation coefficient within that window:

Connectivity Method	Signal Domain	Specific Implementation
(Tapered) Sliding Window Correlation	Amplitude	Standard Implementation
Flexible Least Squares		Liao et al. (2014)
Multiplication of Temporal Derivatives		Shine et al. (2015)
Spatial Distance		Thompson et al. (2017)
Jackknife Correlation		Richter et al. (2015)
Dynamic Conditional Correlation		Lindquist et al. (2014)
Phase Synchronization	Time-frequency	Honari et al. (2021)
Wavelet Coherence		Billings et al. (2021)
Leading Eigenvector Dynamics		Cabral et al. (2017)

Table 1. Included dynamic functional connectivity methods and their respective implementation.

$$SWC_i = \frac{\mathrm{cov}(x_i, y_i)}{\sigma(x_i)\sigma(y_i)} \tag{1}$$

Here, x_i and y_i are the subset of $x(t)$ and $y(t)$ within the ith window, $\mathrm{cov}(x_i, y_i)$ is the covariance between x_i and y_i, and σ denotes the standard deviations of x_i and y_i, respectively. Considering data from N brain regions, this creates $N \times (N-1)$ unique dFC estimates for each window, which are usually summarized in connectivity matrices of shape $N \times N$. Inspecting the temporal fluctuations of values in these matrices at specific position reveals the time course of dFC estimates, which, due to the low-pass properties of the window, are smoothed time courses representing the time-varying connectivity between each pair of brain areas.

Sliding window correlations are a mathematically and computationally simple, but also come with significant caveats. The most discussed issue stems from the choice of the window length, which controls the a trade-off between temporal sensitivity and accuracy (Thompson et al., 2018). Small window sizes increase the risk of spurious fluctuations in dFC due to noise and also hinders a reliable computation of the Pearson correlation, while large windows result in the loss of temporal variations in the signal.

Multiplication of Temporal Derivatives

Multiplication of Temporal Derivatives (MTD) was first introduced by Shine et al. (2015) as a method to overcome the aforementioned shortcomings of the sliding window. The method first calculates the temporal derivative dt for each time series $x(t)$ at each time point t:

$$dt_{x,t} = x_{x,t} - x_{x,t-1} \tag{2}$$

Then, the coupling between each pair of derivative time series dt_x and dt_x is computed by multiplying them at each time point t and dividing by the product of their standard deviation.

$$MTD_t = \frac{dt_{x,t} dt_{y,t}}{\sigma_{dt_x} \sigma_{dt_y}} \tag{3}$$

Thus, MTD provides a measure of functional coupling between time series. If signals increase or decrease together MTD will be positive, if signals change in opposite direction MTD will be negative. As a frame-wise estimate operating on individual time points, MTD can, however, be susceptible to high-frequency noise. Thus, Shine et al. (2015) suggest to use temporal smoothing by using a simple moving average for the MTD time series with a window size of 7 time points.

Jackknife Correlation

The Jackknife Correlation (JC) method (Richter et al., 2015) computes the covariance of two time series $x(t)$ and $y(t)$ at time point t by using all time points except t. This therefore creates estimates of covariance on a single trial basis and can intuitively be seen as an inverse sliding window of length one:

$$JC_t = -\left(\frac{\sum_i^T (x_i - \bar{x}_t)(y_i - \bar{y}_t)}{\sum_i^T (x_i - \bar{x}_t)^2 \sum_i^T (y_i - \bar{y}_t)^2} \right) \quad i \neq t \tag{4}$$

The Jackknife Correlation JC_t is calculated for each time point t. The leave-1-out process results in an inversion of the direction of connectivity estimates, which is corrected by the minus sign. The primary advantage of the JC method is that it addresses the aforementioned challenges associated with small window sizes as mentioned above. In this context, the method has previously been described as an optimal sliding window, as it avoids the trade-off between temporal sensitivity and accuracy since the calculation of JC at each time point t is based on $N-1$ data points with N equal to the length of the time series (Thompson et al., 2018).

It is to note that this method is also not without its limitations. First, JC is highly sensitive to high-frequency noise in the data. This could, however, be reduced by excluding more than one time point for each estimate, resulting in a leave-n-out procedure. Second, it only provides relative estimates of connectivity that fluctuate around the static connectivity of each subject. And third, it leads to a compression of variance proportional to the length of the time series as each estimate differs by just one data point. To that avail, Thompson and Fransson (2018) propose further processing steps like standardisation of the resulting JC time series for direct comparison between subjects.

Spatial Distance

The previously introduced methods all use temporally adjacent time points to estimate the covariance between brain areas. Spatial Distance (SD) (Thompson, Brantefors, and Fransson, 2017) differs from that approach by using time points that have a similar spatial profile. Initially, a weight vector is computed for each time point t by calculating the spatial distance (e.g. Euclidean distance) to all other time points. As introduced by Thompson, Brantefors, and Fransson (2017), the present toolbox implements the inverse of the Euclidean Distance (ED) between the BOLD signal amplitudes at time point t and all other time points u:

$$w_u^{(t)} = \frac{1}{ED_{t,u}} \quad t \neq u \tag{5}$$

Assuming a time series containing 100 time points, SD will then create a vector of length 100 for each time point, with each vector containing the spatial distance of the activity pattern from the corresponding time point to all other time points. Each weight vector is then normalized, scaling the weights between 0 and 1, with the self-weight at time point u being set to 1. Subsequently, the functional connectivity between two brain regions at time point t can be calculated as the weighted Pearson correlation coefficient, utilizing the weight vector $w_u^{(t)}$. SD has shown to be effective in capturing dynamic changes in connectivity by prioritizing spatial similarities. However, it may be influenced by the choice of distance metric and normalization process (Thompson, Brantefors, and Fransson, 2017).

Flexible Least Squares

Flexible Least Squares (FLS) is a time-varying flexible regression approach introduced by Kalaba and Tesfatsion (1989), which gained popularity in the domain of dFC through its inclusion in the DynamicBC toolbox (Liao et al., 2014). FLS estimates changing relationships between time series through a β coefficient that varies over time to minimize data fitting errors. It is a data-driven and distribution-free method that accounts for both measurement and dynamic errors to capture evolving associations between the time series. First, a general regression model is assumed in the form of:

$$y(t) = x(t)\beta(t) + u(t) \tag{6}$$

In this equation, $x(t)$ and $y(t)$ are two time series, $\beta(t)$ determines the strength of the covariance between them, and $u(t)$ is the approximation error. Opposed to static linear regression, which estimates a single beta coefficient, FLS estimates a time-varying sequence of betas by minimizing 1) a measurement error between the actual and estimated observation at each time point, and 2) a dynamic error which represents the discrepancy due to incorrect specification of the state transition equations. The measurement fit error is described by:

$$r_M^2(\beta, T) = \sum_{t=1}^{T} (y(t) - x(t)\beta(t))^2 \tag{7}$$

where $x(t)$ and $y(t)$ represent two time series of length T at time t and β denotes the model fit coefficient vector. The dynamic error is described by:

$$r_D^2(\beta, T) = \sum_{t=1}^{T-1} (\beta(t+1) - \beta(t))^T (\beta(t+1) - \beta(t)) \tag{8}$$

This dynamic error also has the objective of smoothing the $\beta(t)$ coefficients by penalizing large variations between time points and thus preventing overfitting to noise. Both errors are then combined into a single cost-incompatibility function to estimate the time-varying $\beta(t)$ sequence:

$$C(\beta, \mu, \mathrm{T}) = \mu * r_D^2(\beta, T) + r_M^2(\beta, T), \tag{9}$$

This function is then minimized through ordinary least squares, with μ forming a weighting parameter between the measurement fit error r_M^2 and the dynamic error r_D^2. The resulting β coefficient between two signals at time point t is then interpreted as the functional connectivity. As previously mentioned for JC and SD, the FLS method also has the advantage of not relying on a specified window length and thus avoids the aforementioned issues that come with this choice. However, as the optimization procedure has to be performed for each pair of signals, the method is computationally the most demanding out of all included methods.

Dynamic Conditional Correlation

Dynamic Conditional Correlation (DCC) (Engle, 2002) is an extension of GARCH models predominantly used in economics and finance. DCC models time-varying correlation between time series by adapting to transient fluctuations in data. Its use was recently introduced in the context of fMRI analyses by Lindquist et al. (2014), who found DCC to outperform previously established methods for estimating dFC. A GARCH model is described through the following equations:

$$y_t = \sigma_t \varepsilon_t \tag{10}$$

Here, it is first assumed that a time series y_t can be expressed as the product of a time-varying variance term σ_t and random variable $\varepsilon_t \sim \mathcal{N}(0,1)$ referred to as innovation. In a GARCH(1,1) model, which takes into account a single autoregressive term and a single moving average term, the conditional variance is then expressed as:

$$\sigma_t^2 = \omega + \alpha y_{t-1}^2 + \beta \sigma_{t-1}^2 \tag{11}$$

with parameters $\omega > 0$, $\alpha, \beta \geq 0$ and $\alpha, \beta < 1$. α modulates the influence of the squared previous observation y_{t-1}^2 and β modulates the influence of the previous estimate of the conditional variance σ_{t-1}^2. These parameters are then optimized through maximum likelihood estimation to best match the time series.

It should be noted that this can be generalized to GARCH(p,q) processes, with p denoting the number of previous observations and n denoting the number of previous estimates of the conditional variance to be included in the model. This could especially prove fruitful when using data acquired with a small TR, however this has yet to be explored in the context of fMRI dFC analyses.

As an extension to GARCH, the DCC method has been shown to be superior to other multivariate GARCH models that try to estimate dynamic correlations (Engle, 2002). To show the general working of the model, $y_t = \varepsilon_t$ is considered as a bivariate time series with mean zero and a conditional covariance matrix Σ_t. DCC is then expressed through the following set of equations:

$$\sigma_{i,t}^2 = \omega_i + \alpha_i y_{i,t-1}^2 + \beta_i \sigma_{i,t-1}^2 \text{ for } i = 1,2 \tag{12}$$

$$\mathbf{D}_t = \operatorname{diag}\left\{\sigma_{1,t}, \sigma_{2,t}\right\} \tag{13}$$

$$\varepsilon_t = \mathbf{D}_t^{-1} \mathbf{e}_t \tag{14}$$

$$\mathbf{Q}_t = (1 - \theta_1 - \theta_2)\overline{\mathbf{Q}} + \theta_1 \varepsilon_{t-1} \varepsilon_{t-1}' + \theta_2 \mathbf{Q}_{t-1} \tag{15}$$

$$\mathbf{R}_t = \operatorname{diag}\{\mathbf{Q}_t\}^{-1/2} \mathbf{Q}_t \operatorname{diag}\{\mathbf{Q}_t\}^{-1/2} \tag{16}$$

$$\sum_t = \mathbf{D}_t \mathbf{R}_t \mathbf{D}_t \tag{17}$$

These equations describe two general steps. First, univariate GARCH(1,1) models are fit to each time series in y_t (Eq. 12), and standardized residuals are calculated (Eq. 14). Second, Q_t, a non-normalized version of the time-varying correlation is derived using an EWMA method on standardized residuals (Eq. 15). The matrix $\bar{Q}$ is the unconditional covariance of ε_t, with scalars θ_1, θ_2 constrained as $0 < \theta_1 + \theta_2 < 1$. They are adjusted for scale to form a valid correlation matrix (Eq. 16), and the time-varying covariance matrix is established (Eq. 17). The model parameters, including ω_1, α_1, β_1, ω_2, α_2, β_2, θ_1, θ_2, are estimated through a bifurcated process: initially, by estimating time-varying variances for each time series, and subsequently by utilizing standardized residuals to estimate dynamic correlations as shown by (Engle, 2002) .

The strength of the DCC model is its capability to detect rapid changes in data correlation structures with relatively few assumptions, and it has been shown to outperform sliding window techniques but also faces challenges such as computational complexity, potential numerical issues

during optimization, as well as a risk of overfitting to noise (Lindquist et al., 2014). The latter is particularly problematic in fMRI data, where the signal-to-noise ratio is generally low.

Phase Synchronization

Phase Synchronization (PS) models originate from the field of physics, and have initially been used to examine the behaviour of weakly coupled oscillators (Rosenblum, Pikovsky, and Kurths, 1996). The general idea behind PS methods is to convert a real signal into its complex analytic version to obtain instantaneous phase and amplitude information, and then derive pairwise coherence estimates. While this can be achieved in various ways, phase synchronization methods usually apply the Hilbert transform, which is described as follows:

$$x_a(t) = x(t) + j\mathrm{H}[x(t)] \tag{18}$$

Here, the real signal $x(t)$ is translated into its analytic form $x_a(t)$ with H being the Hilbert transform and j being the imaginary unit. The instantaneous phase is then defined as:

$$\phi(t) = \arg(x_a(t)) = \arctan\left(\frac{\mathrm{H}[x(t)]}{x(t)}\right) \tag{19}$$

Given the instantaneous phase ϕ_i and ϕ_j of two analytic signals, their coherence can be estimated in multiple ways. The toolbox includes two different approaches, namely the PSc (Honari, Choe, and Lindquist, 2021; Maltbie et al., 2022) and and PSp (Honari, Choe, and Lindquist, 2021). These approaches are implemented as follows:

$$PSc_{ij} = cos(\phi_i - \phi_j) \tag{20}$$

$$PSp_{ij} = 1 - \mid sin(\phi_i - \phi_j) \mid \tag{21}$$

Here, instantaneous PS is estimated by either taking the cosine of the instantaneous phase difference between a pair of signals, or by subtracting the absolute of the sine of the instantaneous phase difference from one. This results in different measures of PS, as it produces a coherence matrix with estimates ranging from -1 (anti-phase) to 1 (in phase) for PSc, and coherence estimates ranging between 0 (no phase coherence) and 1 (perfect phase coherence) for PSp.

The main limitation of these phase synchronization methods is that the Hilbert transform only produces meaningful results on a narrowband signal, and the fMRI time series therefore needs to be filtered accordingly. While fMRI studies usually filter signal in the range of 0.01-0.1 Hz, this is band is still too broad and literature suggests smaller bands (e.g. 0.03-0.07 Hz) are required to obtain meaningful results (Sheng et al., 2023; Honari, Choe, and Lindquist, 2021; Glerean et al., 2012).

Leading Eigenvector Dynamics

Leading Eigenvector Dynamics (LeiDA) extends the idea of the PS method by deriving the leading eigenvector from the coherence matrix and is implemented as described by Cabral et al. (2017) and Olsen et al. (2022). As previously described for the PS method, LeiDA relies on the Hilbert transform to extract the instantaneous of an analytic signal. From the instantaneous phase, a coherence matrix C is then computed analogous to PSc with i and j indexing over brain regions:

$$C_{ij} = \cos(\phi_i(t) - \phi_j(t)) \tag{22}$$

The eigendecomposition of this coherence matrix then provides its eigenvectors and eigenvalues. The leading eigenvector, which corresponds to the matrix's largest eigenvalue, is of particular interest:

$$\Lambda, V = \text{eigh}(C) \tag{23}$$

where Λ is the diagonal matrix of eigenvalues and V are the corresponding eigenvectors with the leading eigenvector V_1. V_1 is then interpreted as a measure of functional connectivity, with nodes of equal sign being considered as coherent, and the size of V_1 representing the strength of this coherence. This reduces the dimensionality of each dFC estimate from $(N \times (N-1))/2$ to N, which is usually the preferred output. However, a full connectivity matrix can be reconstructed as the outer product of the leading eigenvectors with their transpose, which has been shown to still explain most of the variance of the original coherence matrix (Cabral et al., 2017):

$$R(t) = V_{i1} \otimes V_{j1} \tag{24}$$

Wavelet Coherence

The last approach included in the toolbox is Wavelet Coherence (WCOH), which is implemented as described by Billings et al. (2021). In contrast to phase synchronization methods which focus on a a single, narrow frequency band, WCOH makes use of the continuous wavelet transform to transform the real signal $f(t)$ into time-frequency spectrograms and calculates pairwise coherence between them. The wavelet transform is given by:

$$W_t(s) = \sum_{t=0}^{T-1} f(t) \frac{1}{S} \psi^* \left(\frac{t-u}{s} \right) \delta t \tag{25}$$

Here, $W_t(s)$ represents the wavelet coefficient at time t and frequency scale s. The function ψ^* is the complex conjugate of the wavelet function, while u serves as the translation parameter that facilitates the shifting of the wavelet across the time series. The squared wavelet coherence $R_t^2(s)$ between two time series X and Y can then be computed as:

$$R_t^2(s) = \frac{\left| \left\langle \frac{1}{s} W_t^{XY}(s) \right\rangle \right|^2}{\left\langle \frac{1}{s} \left| W_t^X(s) \right|^2 \right\rangle \left\langle \frac{1}{s} \left| W_t^Y(s) \right|^2 \right\rangle} \tag{26}$$

In this equation, $R_t^2(s)$ denotes the squared coherence at time t and scale s. The term $W_t^{XY}(s)$ is the cross-wavelet transform between X and Y, while $W_t^X(s)$ and $W_t^Y(s)$ are their individual wavelet transforms. The operator $\langle \cdot \rangle$ indicates a smoothing operation in both time and scale, and the factor s^{-1} serves as a normalization in the wavelet transform.

While it is in theory possible to perform subsequent analyses for separate frequency bands, it is often useful to obtain a unified dFC measure. To achieve this, Billings et al. (2021) proposed to take a weighted average over all scales:

$$AWC_t^{XY} = 1 - \sum^{s} \frac{\left| W_t^{XY}(s) \right|}{\max_s \left| W_t^{XY}(s) \right|} R_t^2(s) \tag{27}$$

Here, the cross-wavelet power $W_t^{XY}(s)$ is normalized and used for the calculation of a weighted mean of $R_t^2(s)$ over all frequency bands, resulting in an average wavelet coherence (AWC) time series for each pair of signals.

In contrast to the previously introduced methods of PS and LeiDA, WCOH has the benefit of being applicable beyond narrow-band signals. However, a peculiarity with WCOH are 'cone of influence artifacts' which refer to the distortions at the edges the spectrograms, caused by the finite length of the data. A common approach to address this is by removing the outer scales and time points, but this leads to a loss of potentially important data, especially in the case of the latter. Moreover, WCOH necessitates variable smoothing across both time and frequency domains (Grinsted, Moore, and Jevrejeva, 2004), making it the most heavily smoothed method in the toolbox. This needs to be considered when fast fluctuations in dFC are of interest.

2.2 Multiverse Analysis

For multiverse analysis, COMET follows a similar approach to the existing packages like Boba (Liu et al., 2020). Here, the multiverse of potential analysis pipelines can be created through a modular approach which will in detail be described and showcased with a usage example in Section 3.2. In short, users must define the possible options for each decision, i.e. they define the forking paths in the analysis pipelines. The toolbox then automatically creates analysis scripts for all possible combinations of decision options and can also evaluate these scripts. This approach was chosen so that users have an easy way of checking individual scripts and subsequently have the flexibility of running individual scripts sequentially one after the other or as large batches in parallel, depending on the available computational resources.

2.3 Graph Analysis

A diverse set of graph analytical options is offered for multiverse analysis. For this, COMET allows users to use functions included in the Python version of the Brain Connectivity Toolbox (Rubinov and Sporns, 2010). Further, for performance related reasons, COMET also offers custom graph measure implementations which compile to machine code at run time using the Numba library (Lam, Pitrou, and Seibert, 2015). This significantly increases calculation speed, which is particularly helpful when dealing with large graphs. An example for this is shown for the calculation of local efficiency in a relatively large network containing 379 nodes, in which performance is increased by a factor of 3 to 10 compared to previous implementations of the method (Figure 2).

3 Usage

The COMET toolbox can be installed via the Python Packaging Index by typing following commands in the command prompt/terminal. As this will automatically install or update required packages, we recommend users to set up a dedicated conda[1] environment to avoid potential version conflicts with existing projects.

Command line

```
pip install comet-toolbox
```

A main design goal of the toolbox is flexible and modular usage independent of the analysis pipeline. More specifically, researchers are not required to run their entire analysis pipelines in the toolbox, but can also use it as a standalone tool to estimate dFC or run multiverse analyses independent of their specific research question. For the dFC module this means only a two dimensional matrix

[1] https://conda.io/projects/conda/en/latest/index.html

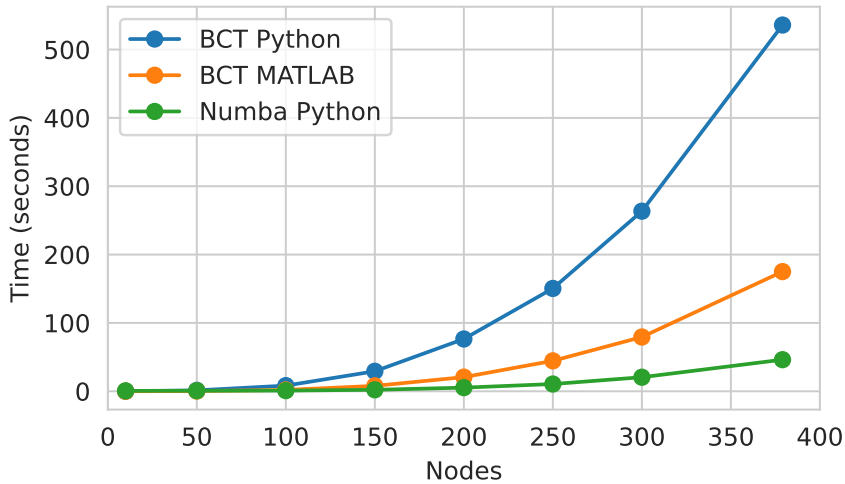

Figure 2. Calculation time for local efficiency. The custom Numba implementation included in COMET is compiled to native machine code at run time, which significantly improves performance. For a weighted network with 379 nodes and 50% density, the calculation of local efficiency shows improved calculation speed compared to the Python version (> 10x) and MATLAB version (> 3x) of the Brain Connectivity Toolbox.

of time series data (T time points x N brain regions) is required as input and a three dimensional functional connectivity matrix ($N * N * T$) is returned for further use. This allows researchers running analyses in different programming languages or frameworks and, in a second step, to use the toolbox as a stand-alone module to derive dFC estimates that can be further analysed with their own software. For the multiverse analysis module, forking paths are not limited to the functions introduced in the toolbox, but can flexibly be created for any kind of variable or function.

Generally speaking, interfacing with the toolbox is possible through either a GUI or a standard scripting API. On the one hand, the GUI accepts time series data in all popular data formats, and allows the user to save dynamic functional connectivity estimates of the included methods. The GUI further allows for visual exploration of the results through the tabs above the plot. This enables users to switch between dFC matrices, histograms of the distribution of dFC values, and further also the dFC time series of individual edges. On the other hand, the scripting approach is more flexible and currently the only option for performing multiverse analyses. It is however also possible to use the GUI within scripts, combining the advantages of both approaches.

3.1 Graphical User Interface

If preprocessed 2D matrices of time series data with $N_{time\ points}$ rows and $N_{brain\ regions}$ columns are already available, users can chose to use the GUI to calculate the time series of the dFC estimators, check them visually and visualise their distribution properties. If the package was installed via pip, the GUI can be invoked through the command prompt/terminal:

Command line

```
comet-gui
```

As shown in Figure 3, the basic usage of the GUI is as follows:

1. 2D time series with shape $N_{time\ points} * N_{brain\ regions}$ needs to be loaded. The resulting shape of the data will be displayed and data can be transposed if it does not match the required shape.

2. The desired method for estimating dFC can be chosen in a drop-down menu. All available parameters will automatically be displayed below. Continuous parameters are set by default to the values selected in the original publications (Table 1). The *i* symbol also provides additional information about the parameters.

3. Once the model is specified, dFC estimates can be calculated and afterwards also saved to disk. The *Keep in memory* checkbox allows users to keep the calculated dFC estimates in working memory to enable quick switching and comparison between different methods. For computationally expensive dFC methods, a progress bar will be be displayed in the terminal from which the GUI was invoked. Once calculations are finished, the results will be displayed and can be explored by using the slider at the bottom of the window to inspect the connectivity at different time points.

4. After dFC is estimated, it will automatically be displayed in the connectivity tab on the right hand side of the GUI. Users can further also inspect the distribution of dFC values for a given time point or the time series for single pairs of brain regions by using the other tabs. This helps to achieve an intuitive understanding about the derived results and can also be an indicator for potentially erroneous dFC estimates.

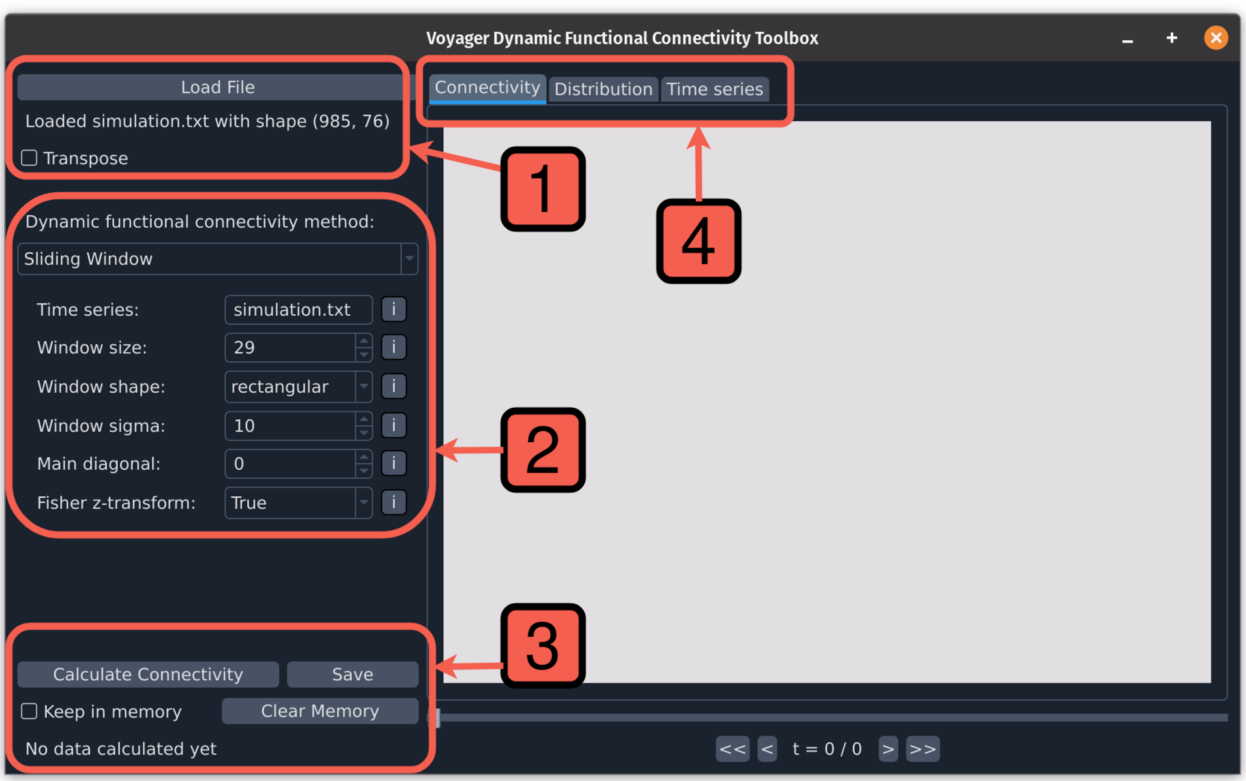

Figure 3. Basic GUI usage. The left side of the GUI is used for data handling and method selection, while the right side of the GUI will subsequently provide visual information for estimated dFC data.

3.2 Scripting API

If more complex analyses are required or a scripting approach is preferred in general, the dFC methods can be used as shown in Figure 4. In this example, simulated time series data is loaded from the data module and subsequently band-pass filtered. Then, dFC estimates for phase synchronization are calculated by first creating a PhaseSynchrony() object from the connectivity module, and then by invoking its connectivity() method for the actual calculation. For quick visualization purposes, the calculated dFC data can then also be inspected through the GUI. A list containing all available classes, methods, and parameters can be found in the documentation of the package.

```
from comet import methods, data, gui

ts = data.load_example()
ts_clean = data.clean(ts, low_pass=0.07, high_pass=0.03)

ps = methods.PhaseSynchrony(ts_clean, method="crp")
dfc = ps.connectivity()

gui.run(dfc_data=dfc)
```

Mock time series data is loaded from the package and subsequently band-pass filtered.
All dFC methods require **2D time series data as input.**

The included dFC methods are instantiated as classes and connectivity can be estimated through the **connectivity()** method.

The **GUI** is also accesible from the script for quick visualization and quality assessment purposes

Figure 4. dFC calculation and visualization.

If the estimation of dFC is to be included into multiverse analysis, a workflow which uses the multiverse module is required. The general structure of a multiverse analysis starts with the definition of the forking paths/decision points, which are then incorporated into the analysis pipeline via a template script. For this, we will use the following terminology: In an analysis pipeline, the user can arrive at forking paths at which a decision between two or more possible analysis options has to be made. An example for this with two decision points is shown in Figure 5. Here, the individual forking paths are defined in a python dictionary, with key-value pairs indicating the type of decision and the possible options. For example, the first forking path *"low_pass"* contains three different options for the cutoff frequency of the low pass filter, while the second forking path *"dfc_method"* contains two options for the dFC method to be applied on the cleaned data. These decisions can then be included into a template analysis pipeline. This template is a standard analysis pipeline, but instead of containing individual decisions, the forking paths are defined by double brackets as a placeholder (e.g. *{{low_pass}}*). This template has to either be encapsulated in a Python function or can, for larger analysis pipelines, be loaded from a dedicated Python script (shown in the example scripts in the GitHub repository). Once the complete analysis pipeline with all forking paths is created, the *multiverse.create()* function can be called to create an individual Python script for each universe (every possible combination of decisions). Here, with two decisions containing $3 * 2$ options, this would create analysis scripts for six different universes. Users can then run the analysis pipeline of all (or also individual) universes with the *multiverse.run()* function.

4 Analysis Example

We now showcase the toolbox and multiverse analysis workflow with simulated fMRI data. For this, the SimTB package (Erhardt et al., 2012) was used to create simple example time series data consisting of 10 nodes whose connectivity structure varies between two discrete states in a block-like design. The time series consist of 100 blocks with a duration of 7.2 seconds each, in which one of the

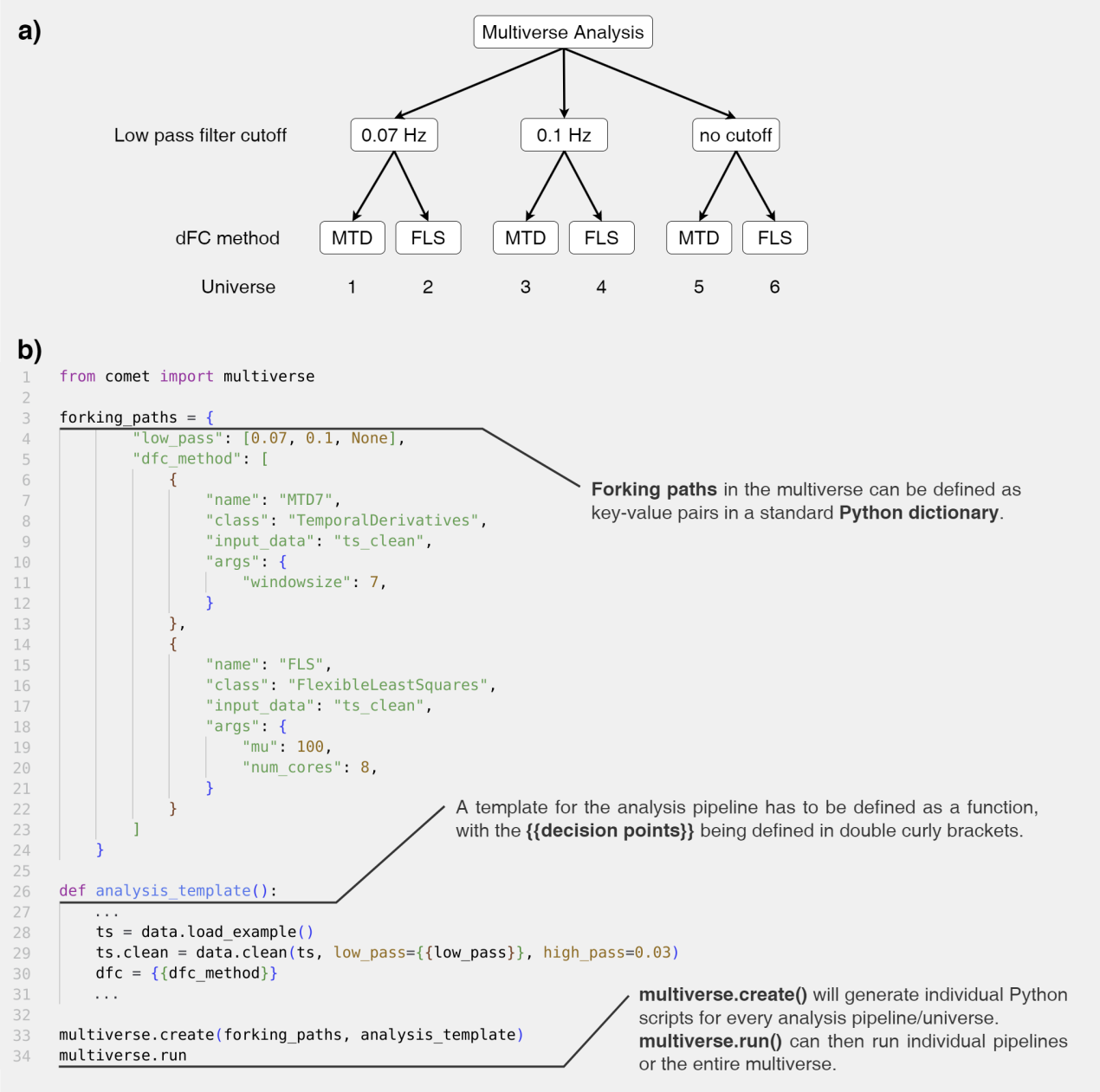

Figure 5. Multiverse analysis example. a) A multiverse analysis with a decision point for the low pass filter (0.07 Hz, 0.1 Hz, None) as well as for the dFC method (Multiplication of Temporal Derivatives, MTD; Flexible Least Squares, FLS) is to be constructed, resulting in 6 different analysis pipelines (universes). b) Implementation of the described multiverse analysis in the COMET toolbox.

two states was randomly chosen as the dominant pattern. Each node has an independent 20% chance of experiencing a unique event at each time point and is further subject to Gaussian noise, adding to the complexity of the signal and creating a more challenging scenario for the dFC methods and analysis.

The main goal of this analysis example is to perform block-by-block classification by using graph measures derived from the connectivity structure estimated by different dFC methods. For this, we constructed a multiverse analysis containing 96 $(6*2*2*2*2)$ universes which include analytical

Forking Path	Decision
dFC method	Sliding Window Correlation
	Flexible Least Squares
	Multiplication of Temporal Derivatives
	Phase Synchronization
	Spatial Distance
	Dynamic Conditional Correlation
Temporal lag	6 TR
	10 TR
Graph density	strongest 50%
	strongest 25%
Graph weights	Weighted
	Binarized
SVC kernel	Linear
	Radial basis function

Table 2. Forking paths and decisions for the example analysis. The multiverse consists of 96 ($6*2*2*2*2$) unique combinations of decisions (universes). For every universe, a support vector classifier (SVC) performed trial-based predictions of the connectivity state, with the results shown in Figure 6

.

decisions for dFC estimation (Sliding Window Correlation, Flexible Least Squares, Multiplication of Temporal Derivatives, Phase Synchronization, Spatial Distance, and Dynamic Conditional Correlation), graph construction (temporal lag of 6 or 10 TR, density of 50% or 25%, and binarized or weighted edges), and the prediction model (linear or radial basis function kernel for the support vector classifier). A summary of these forking paths is provided in Table 2. Notably, we emphasize that we do not want to evaluate the outcome of the multiverse analysis in terms of these decisions and options, but rather want to provide users with a simple to use analysis example. A Jupyter Notebook for this example can be found in the demo folder of the GitHub repository, and we encourage interested readers to explore this resource.

In detail, we applied six distinct dFC methods, each based on different theoretical assumptions. We segmented the resulting dFC time series into 100 trials, in line with our simulation design, and added a temporal lag which shifted the resulting segments by 6 or 10 TR to account for the hemodynamic delay in the BOLD response. Before constructing graphs, we averaged these segments for each trial. The graph construction involved two decisions: selecting the graph density to retain either the top 50% or 25% of edges, and choosing between binarizing the graph or keeping it weighted. From these graphs, we calculated three nodal measures: the clustering coefficient, participation coefficient, and nodal efficiency. These measures were then used as features for classification. We chose a support vector classifier, with a decision path for either a linear or a non-linear radial basis function kernel. After performing 5-fold cross-validation, we visualized the prediction accuracies for all configurations using a specification curve (Figure 6). This visualization arranged individual pipelines and their corresponding accuracies in ascending order, providing a clear overview of the results. Notably, for about half of the pipelines, the 95% confidence interval did not include the chance level of 50% accuracy. Given the known ground truth of two distinct connectivity patterns in the simulation data, this suggests that many of the pipelines could incorrectly lead to the conclusion of no distinct pattern

in the connectivity structure.

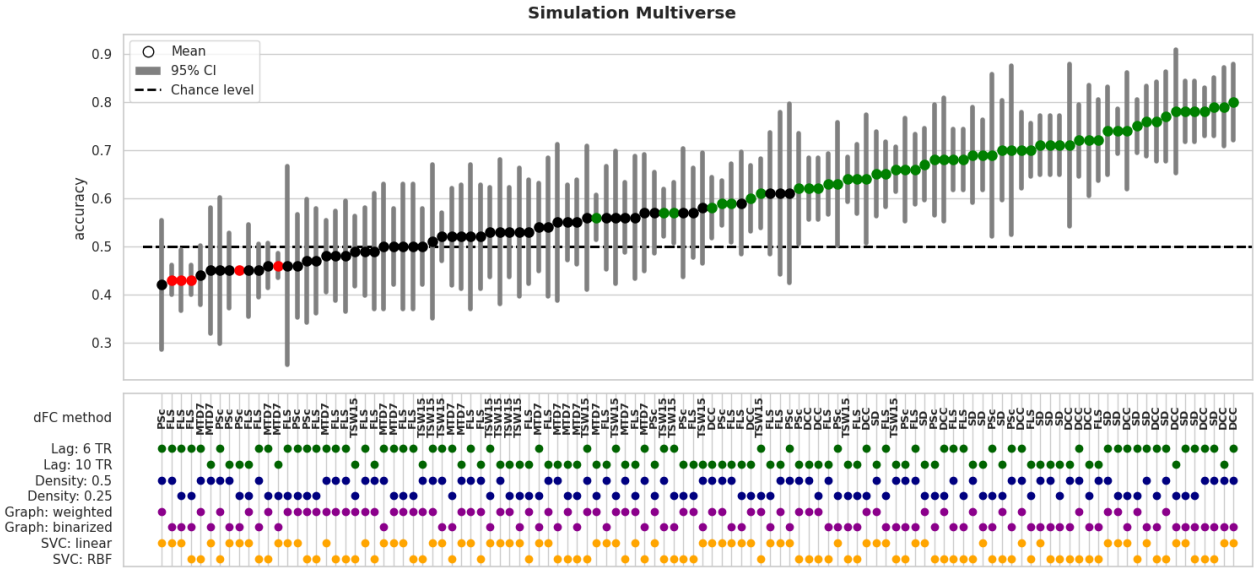

Figure 6. Specification curve for the example analysis. The classification accuracies for the individual universes are displayed in ascending order. The upper plot displays the classification accuracies, with red and green colors indicating universes whose 95% confidence intervals resulting from 5-fold cross validation do not include the random chance level of 50% accuracy. The bottom plot displays the individual decisions chosen in the analysis pipeline leading to the corresponding classification accuracy above.

5 Comparison to existing software

The development of the COMET toolbox is a significant contribution to the field, as such integrated and flexible software packages currently do not exist, particularly when combined with multiverse analysis. In Python, to the best of our knowledge, the most comprehensive software package for estimating dFC is teneto (Thompson, Brantefors, and Fransson, 2017). While teneto includes the computationally simple methods SW, JC, SD, and MTD, it is heavily specialized on working with data in the BIDS format and does not offer support for multiverse analysis. Other popular software packages can also be found in MATLAB. Popular examples in that regard are the DynamicBC toolbox (Liao et al., 2014) which includes the SW and FLS methods. Independent of the programming framework, many other software packages only include individual dFC methods related to corresponding publications. Examples of this are e.g. the timecorr package with kernel based dynamic correlation (Owen, Chang, and Manning, 2021) and the Dynamic Correlation toolbox (Lindquist et al., 2014) which only includes various DCC implementations. Other dFC methods like LeiDA offer source code tailored to the analysis of the specific paper, which requires heavy modifications to be applicable for other analyses (Olsen et al., 2022). While not exhaustive, this brief overview further underlines the necessity of the newly introduced toolbox, as the application of a diverse array of dFC methods is, due to the aforementioned challenges, currently infeasible for many analyses.

Similar issues also arise in the domain of multiverse analysis, with only few software packages being available. Here, the most popular options include Boba for Python and R (Liu et al., 2020), or the multiverse R package (Sarma et al., 2023). These packages currently have much more

sophisticated capabilities compared to COMET, however also have their own shortcomings in the context of neuroimaging analyses as they are mostly tailored to a broader set of simple numerical decisions or statistical analyses. An example of this is if the the complexity of forking paths extends beyond simple parameter options and for example includes entire functions or classes, which is not natively possible in the aforementioned packages, but necessary for the inclusion of complex dFC methods or graph analysis functions.

6 Future Directions

The COMET toolbox enables the application of a diverse range of dFC methods and facilitates multiverse analysis. As it stands, the toolbox is still in an early stage of development, with an ongoing process of enhancement and expansion. In the near future, we plan to expand the dFC methods to include state-based approaches to provide a more comprehensive mapping of the method space. Additionally, we intend to enhance the flexibility of multiverse analysis by adding some of its essential features to the GUI.

Long term, we aim to expand the toolbox with new foundational ideas. In addition to the standard application of multiverse analysis to assess the robustness of results across various analytical decisions through e.g. specification curve analysis (Simonsohn, Simmons, and Nelson, 2020), we also aim to explore new perspectives which understand multiverse analysis as an exploratory tool to identify key covariates that significantly impact study outcomes (Bowring, Nichols, and Maumet, 2022) and as a framework for abduction and inferential decisions (Engzell and Mood, 2023). This can help in excluding ineffective methods in new dataset analyses and provide insights into conditions for expected effects. Another example is the recently proposed approach of using active learning to effectively map the multiverse if an exhaustive exploration is not feasible (Dafflon et al., 2022). The development and integration of such advanced statistical and machine learning techniques aims to contribute to more integrated, transparent, and adaptable analysis approaches. Thus, our vision for the future of this toolbox is not just as a static software package, but as a dynamic, evolving platform that adapts and grows in response to the changing landscape of neuroimaging research.

7 Conclusion

In this work, we introduced COMET, a novel Python toolbox that aims to be a significant contribution to the field of neuroimaging by providing a unified, flexible platform for exploring the dynamic interactions within the brain through various methodological lenses. An important aspect of the toolbox is its systematic integration of continuously varying dFC methods which range from the widely-used Sliding Window Correlation to more sophisticated approaches like Dynamic Conditional Correlation, Wavelet Coherence, and Leading Eigenvector Dynamics, accommodating a diverse set of potential research needs. The multiverse analysis feature of the toolbox addresses reproducibility concerns in neuroimaging research by enabling systematic examination of different analytical decisions across a spectrum of possible analysis pipelines. Its outcome facilitates a comprehensive understanding of how methodological choices impact research outcomes. Additionally, the inclusion of graph analytical options, adds another layer of analytical depth. This feature not only enhances the utility of the toolbox in complex network analyses but also ensures efficiency in processing large graphs, a critical requirement in handling high-dimensional neuroimaging data.

The user-friendly nature of COMET is underlined by its graphical user interface and simple scripting API. This makes it accessible to a broad range of users and independent of a specific analysis pipeline, creating a valuable resource for both experienced researchers and newcomers to

the field. The toolbox addresses the evolving needs of the field of neuroimaging research for more integrated and transparent analysis approaches, and further paves the way for comprehensive and reliable insights into the dynamic nature of brain connectivity.

8 Acknowledgements

This work was supported by a grant from the German Research Foundation (DFG) to Carsten Gießing (GI 682/5-1), Christiane Thiel (TH 766/9-1) and Andrea Hildebrandt (HI 1780/7-1) as part of the DFG priority program "META-REP: A Meta-scientific Programme to Analyse and Optimise Replicability in the Behavioural, Social, and Cognitive Sciences" (SPP 2317).

References

Billings, Jacob, Manish Saggar, Jaroslav Hlinka, Shella Keilholz, and Giovanni Petri (2021). "Simplicial and topological descriptions of human brain dynamics". In: *Network Neuroscience* 5.2, pp. 549–568. DOI: https://doi.org/10.1162/netn_a_00190.

Botvinik-Nezer, Rotem et al. (2020). "Variability in the analysis of a single neuroimaging dataset by many teams". In: *Nature* 582, pp. 84 –88. DOI: https://doi.org/10.1038/s41586-020-2314-9.

Bowring, Alexander, Thomas E Nichols, and Camille Maumet (2022). "Isolating the sources of pipeline-variability in group-level task-fMRI results". In: *Human brain mapping* 43.3, pp. 1112–1128. DOI: https://doi.org/10.1002/hbm.25713.

Cabral, Joana, Diego Vidaurre, Paulo Marques, Ricardo Magalhães, Pedro Silva Moreira, José Miguel Soares, Gustavo Deco, Nuno Sousa, and Morten L Kringelbach (2017). "Cognitive performance in healthy older adults relates to spontaneous switching between states of functional connectivity during rest". In: *Scientific reports* 7.1, p. 5135. DOI: https://doi.org/10.1038/s41598-017-05425-7.

Calhoun, Vince D., Robyn Miller, Godfrey Pearlson, and Tulay Adalı (2014). "The Chronnectome: Time-Varying Connectivity Networks as the Next Frontier in fMRI Data Discovery". In: *Neuron* 84.2, pp. 262–274. DOI: https://doi.org/10.1016/j.neuron.2014.10.015.

Choe, Ann S., Mary Beth Nebel, Anita D. Barber, Jessica R. Cohen, Yuting Xu, James J. Pekar, Brian Caffo, and Martin A. Lindquist (2017). "Comparing test-retest reliability of dynamic functional connectivity methods". In: *NeuroImage* 158, pp. 155–175. DOI: https://doi.org/10.1016/j.neuroimage.2017.07.005.

Dafflon, Jessica et al. (2022). "A guided multiverse study of neuroimaging analyses". In: *Nature Communications* 13.1, p. 3758. ISSN: 2041-1723. DOI: https://doi.org/10.1038/s41467-022-31347-8.

Engle, Robert (2002). "Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models". In: *Journal of Business & Economic Statistics* 20.3, pp. 339–350. DOI: https://doi.org/10.1198/073500102288618487.

Engzell, Per and Carina Mood (2023). "Understanding Patterns and Trends in Income Mobility through Multiverse Analysis". In: *American Sociological Review* 88.4, pp. 600–626. DOI: https://doi.org/10.1177/00031224231180607.

Erhardt, Erik B, Elena A Allen, Yonghua Wei, Tom Eichele, and Vince D Calhoun (2012). "SimTB, a simulation toolbox for fMRI data under a model of spatiotemporal separability". In: *Neuroimage* 59.4, pp. 4160–4167. DOI: https://doi.org/10.1016/j.neuroimage.2011.11.088.

Fanelli, Daniele (2018). "Is science really facing a reproducibility crisis, and do we need it to?" In: *Proceedings of the National Academy of Sciences* 115.11, pp. 2628–2631. DOI: https://doi.org/10.1073/pnas.1708272114.

Frias-Navarro, Dolores, Juan Pascual-Llobell, Marcos Pascual-Soler, Jose Perezgonzalez, and Jose Berrios-Riquelme (2020). "Replication crisis or an opportunity to improve scientific production?" In: *European Journal of Education* 55.4, pp. 618–631. DOI: https://doi.org/10.1111/ejed.12417.

Glerean, Enrico, Juha Salmi, Juha M Lahnakoski, Iiro P Jääskeläinen, and Mikko Sams (2012). "Functional magnetic resonance imaging phase synchronization as a measure of dynamic functional connectivity". In: *Brain connectivity* 2.2, pp. 91–101. DOI: https://doi.org/10.1089/brain.2011.0068.

Gonzalez-Castillo, Javier and Peter A. Bandettini (2018). "Task-based dynamic functional connectivity: Recent findings and open questions". In: *NeuroImage* 180, pp. 526–533. DOI: 10.1016/j.neuroimage.2017.08.006.

Grinsted, Aslak, John C Moore, and Svetlana Jevrejeva (2004). "Application of the cross wavelet transform and wavelet coherence to geophysical time series". In: *Nonlinear processes in geophysics* 11.5/6, pp. 561–566. DOI: https://doi.org/10.5194/npg-11-561-2004.

Honari, Hamed, Ann S Choe, and Martin A Lindquist (2021). "Evaluating phase synchronization methods in fMRI: A comparison study and new approaches". In: *NeuroImage* 228. , p. 117704. DOI: https://doi.org/10.1016/j.neuroimage.2020.117704.

Hutchison, R. Matthew et al. (2013). "Dynamic functional connectivity: Promise, issues, and interpretations". In: *NeuroImage* 80, pp. 360–378. DOI: https://doi.org/10.1016/j.neuroimage.2013.05.079.

Ioannidis, John P. A. (2005). "Why Most Published Research Findings Are False". In: *PLoS Medicine* 2.8, e124. DOI: 10.1371/journal.pmed.0020124.

Kalaba, Robert and Leigh Tesfatsion (1989). "Time-varying linear regression via flexible least squares". In: *Computers & Mathematics with Applications* 17.8-9, pp. 1215–1245. DOI: https://doi.org/10.1016/0898-1221(89)90091-6.

Lam, Siu Kwan, Antoine Pitrou, and Stanley Seibert (2015). "Numba: A llvm-based python jit compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6. DOI: https://doi.org/10.1145/2833157.2833162.

Liao, Wei, Guo-Rong Wu, Qiang Xu, Gong-Jun Ji, Zhiqiang Zhang, Yu-Feng Zang, and Guangming Lu (2014). "DynamicBC: a MATLAB toolbox for dynamic brain connectome analysis". In: *Brain connectivity* 4.10, pp. 780–790. DOI: https://doi.org/10.1089/brain.2014.0253.

Lindquist, Martin A., Yuting Xu, Mary Beth Nebel, and Brain S. Caffo (2014). "Evaluating dynamic bivariate correlations in resting-state fMRI: A comparison study and a new approach". In: *NeuroImage* 101, pp. 531–546. DOI: https://doi.org/10.1016/j.neuroimage.2014.06.052.

Liu, Yang, Alex Kale, Tim Althoff, and Jeffrey Heer (2020). "Boba: Authoring and visualizing multiverse analyses". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2, pp. 1753–1763. DOI: https://doi.org/10.1109/TVCG.2020.3028985.

Lurie, Daniel J. et al. (Feb. 2020). "Questions and controversies in the study of time-varying functional connectivity in resting fMRI". In: *Network Neuroscience* 4.1, pp. 30–69. ISSN: 2472-1751. DOI: https://doi.org/10.1162/netn_a_00116.

Maltbie, Eric, Behnaz Yousefi, Xiaodi Zhang, Amrit Kashyap, and Shella Keilholz (2022). "Comparison of Resting-State Functional MRI Methods for Characterizing Brain Dynamics". In: *Frontiers*

in Neural Circuits 16, p. 681544. DOI: https://doi.org/10.3389/fncir.2022.681544.

Olsen, Anders S et al. (2022). "Psilocybin modulation of time-varying functional connectivity is associated with plasma psilocin and subjective effects". In: *Neuroimage* 264, p. 119716. DOI: https://doi.org/10.1016/j.neuroimage.2022.119716.

Owen, Lucy LW, Thomas H Chang, and Jeremy R Manning (2021). "High-level cognition during story listening is reflected in high-order dynamic correlations in neural activity patterns". In: *Nature Communications* 12.1, p. 5728.

Preti, Maria Giulia, Thomas AW Bolton, and Dimitri Van De Ville (2017). "The dynamic functional connectome: State-of-the-art and perspectives". In: *Neuroimage* 160, pp. 41–54. DOI: https://doi.org/10.1016/j.neuroimage.2016.12.061.

Richter, Craig G., William H. Thompson, Conrado A. Bosman, and Pascal Fries (2015). "A jackknife approach to quantifying single-trial correlation between covariance-based metrics undefined on a single-trial basis". In: *NeuroImage* 114, pp. 57–70. DOI: 10.1016/j.neuroimage.2015.04.040.

Rosenblum, Michael G, Arkady S Pikovsky, and Jürgen Kurths (1996). "Phase synchronization of chaotic oscillators". In: *Physical review letters* 76.11, p. 1804. DOI: https://doi.org/10.1103/PhysRevLett.76.1804.

Rubinov, Mikail and Olaf Sporns (2010). "Complex network measures of brain connectivity: uses and interpretations". In: *Neuroimage* 52.3, pp. 1059–1069. DOI: https://doi.org/10.1016/j.neuroimage.2009.10.003.

Sarma, Abhraneel, Alex Kale, Michael Jongho Moon, Nathan Taback, Fanny Chevalier, Jessica Hullman, and Matthew Kay (2023). "multiverse: Multiplexing alternative data analyses in R notebooks". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–15. DOI: https://doi.org/10.1145/3544548.3580726.

Shakil, Sadia, Chin-Hui Lee, and Shella Dawn Keilholz (2016). "Evaluation of sliding window correlation performance for characterizing dynamic functional connectivity and brain states". In: *NeuroImage* 133, pp. 111–128. DOI: https://doi.org/10.1016/j.neuroimage.2016.02.074.

Sheng, Dan, Weidan Pu, Zeqiang Linli, Guo-Liang Tian, Shuixia Guo, and Yu Fei (2023). "Aberrant global and local dynamic properties in schizophrenia with instantaneous phase method based on Hilbert transform". In: *Psychological Medicine* 53.5, pp. 2125–2135. DOI: https://doi.org/10.1017/S0033291721003895.

Shine, James M., Oluwasanmi Koyejo, Peter T. Bell, Krzysztof J. Gorgolewski, Moran Gilat, and Russell A. Poldrack (2015). "Estimation of dynamic functional connectivity using Multiplication of Temporal Derivatives". In: *NeuroImage* 122, pp. 399–407. DOI: https://doi.org/10.1016/j.neuroimage.2015.07.064.

Simmons, Joseph P., Leif D. Nelson, and Uri Simonsohn (2011). "False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant". In: *Psychological Science* 22.11, pp. 1359–1366. DOI: https://doi.org/10.1177/0956797611417632.

Simonsohn, Uri, Joseph P. Simmons, and Leif D. Nelson (2020). "Specification curve analysis". In: *Nature Human Behaviour* 4.11, pp. 1208–1214. DOI: https://doi.org/10.1038/s41562-020-0912-z.

Steegen, Sara, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel (2016). "Increasing Transparency Through a Multiverse Analysis". In: *Perspectives on Psychological Science* 11.5, pp. 702–712. DOI: https://doi.org/10.1177/1745691616658637.

Thompson, William Hedley, Per Brantefors, and Peter Fransson (2017). "From static to temporal network theory: Applications to functional brain connectivity". In: *Network Neuroscience* 1.2. , pp. 69–99. DOI: https://doi.org/10.1162/NETN_a_00011.

Thompson, William Hedley and Peter Fransson (2018). "A common framework for the problem of deriving estimates of dynamic functional brain connectivity". In: *NeuroImage* 172, pp. 896–902. DOI: https://doi.org/10.1016/j.neuroimage.2017.12.057.

Thompson, William Hedley, Craig Geoffrey Richter, Pontus Plavén-Sigray, and Peter Fransson (2018). "Simulations to benchmark time-varying connectivity methods for fMRI". In: *PLOS Computational Biology* 14.5. DOI: https://doi.org/10.1371/journal.pcbi.1006196.