Geosci. Model Dev., 17, 3409–3431, 2024
https://doi.org/10.5194/gmd-17-3409-2024
© Author(s) 2024. This work is distributed under
the Creative Commons Attribution 4.0 License.

Geoscientific
Model Development
Open Access

`cfr` (v2024.1.26): a Python package for climate field reconstruction

Feng Zhu[1], **Julien Emile-Geay**[2], **Gregory J. Hakim**[3], **Dominique Guillot**[5], **Deborah Khider**[4], **Robert Tardif**[3], **and Walter A. Perkins**[6]

[1]Climate and Global Dynamics Laboratory, NSF National Center for Atmospheric Research, Boulder, CO, USA
[2]Department of Earth Sciences, University of Southern California, Los Angeles, CA, USA
[3]Department of Atmospheric Sciences, University of Washington, Seattle, WA, USA
[4]University of Southern California, Information Sciences Institute, Marina Del Rey, CA, USA
[5]Department of Mathematical Sciences, University of Delaware, Newark, DE, USA
[6]Allen Institute for Artificial Intelligence, Seattle, WA, USA

Correspondence: Feng Zhu (fengzhu@ucar.edu)

Received: 15 September 2023 – Discussion started: 26 September 2023
Revised: 27 January 2024 – Accepted: 19 March 2024 – Published: 30 April 2024

Abstract. Climate field reconstruction (CFR) refers to the estimation of spatiotemporal climate fields (such as surface temperature) from a collection of pointwise paleoclimate proxy datasets. Such reconstructions can provide rich information on climate dynamics and provide an out-of-sample validation of climate models. However, most CFR workflows are complex and time-consuming, as they involve (i) preprocessing of the proxy records, climate model simulations, and instrumental observations; (ii) application of one or more statistical methods; and (iii) analysis and visualization of the reconstruction results. Historically, this process has lacked transparency and accessibility, limiting reproducibility and experimentation by non-specialists. This article presents an open-source and object-oriented Python package called `cfr` that aims to make CFR workflows easy to understand and conduct, saving climatologists from technical details and facilitating efficient and reproducible research. `cfr` provides user-friendly utilities for common CFR tasks such as proxy and climate data analysis and visualization, proxy system modeling, and modularized workflows for multiple reconstruction methods, enabling methodological intercomparisons within the same framework. The package is supported with extensive documentation of the application programming interface (API) and a growing number of tutorial notebooks illustrating its usage. As an example, we present two `cfr`-driven reconstruction experiments using the PAGES 2k temperature database applying the last millennium reanalysis (LMR) paleoclimate data assimilation (PDA) framework and the graphical expectation–maximization (GraphEM) algorithm, respectively.

1 Introduction

Paleoclimate reconstructions provide critical context for recent changes and out-of-sample validation of climate models (Masson-Delmotte et al., 2013). Site-based reconstructions have been used to infer climate change with the assumption that site-based paleoclimate records (i.e., proxies) can represent the regional or even global climate variations to some extent (e.g., Shakun et al., 2012; Marcott et al., 2013), yet such approaches may lead to conclusions biased to specific sites (Bova et al., 2021; Osman et al., 2021). Climate field reconstruction (CFR), in contrast, aims to optimally combine the available proxy data and infer multivariate climate variability over an entire domain of interest, hence alleviating the biases in site-based reconstruction methods. It has become the emerging approach to studying spatiotemporal climate history and model–data comparisons over the past 2000 years (e.g., Tingley et al., 2012; Neukom et al., 2019a; Tierney et al., 2020; Zhu et al., 2020, 2022; Osman et al., 2021; King et al., 2021).

Existing CFR methods can be classified into three main categories. The first category is based on regression models, which fit climate variables (e.g., temperature, pressure, and precipitation) onto proxy observations using some variant of least squares (e.g., Mann et al., 1998, 1999; Evans et al., 2002; Cook et al., 2004, 2010; Tingley et al., 2012; Smerdon and Pollack, 2016). A more formal approach is based on Bayesian hierarchical models, which explicitly model the various levels of the relationship linking climate processes to proxy observations and invert the proxy–climate relation via Bayes' theorem (e.g., Tingley and Huybers, 2010a, b, 2013; Christiansen and Ljungqvist, 2017; Shi et al., 2017). Both approaches rely on the covariance structure between proxies and climate variables that is estimated from observational data over a calibration period. The third is based on paleoclimate data assimilation (PDA; e.g., Dirren and Hakim, 2005; Goosse et al., 2006; Ridgwell et al., 2007; Steiger and Hakim, 2016; Hakim et al., 2016; Tardif et al., 2019), which follows a similar idea to the previous approaches but with a critical difference in that the proxy–climate covariance matrix is estimated from climate model simulations instead of observational data. As such spatial, multivariate relationships are available and subject to dynamical constraints. As a result, PDA methods can, in principle, estimate any field simulated in the model prior, though the reconstruction quality will be a function of how strongly those variables can be constrained by paleo-observations.

Since some CFR methods have been shown to depend sensitively on the input data (Wang et al., 2015), it is imperative to apply more than one method to the same problem to establish a result's robustness. This has heretofore been difficult, as most CFR workflows are complex, possibly involving the selection and processing of proxy records, the processing of gridded climate model simulation and instrumental observation data, the calibration of proxy system models (PSMs; Evans et al., 2013), and the setup and execution of the specific reconstruction algorithms, as well as the validation and visualization of the reconstruction results. Each of these steps can lead to a large decision tree (Bürger et al., 2006; Büntgen et al., 2021), complicating comparisons. Furthermore, these steps require a comprehensive knowledge of proxies, data analysis and modeling, and reconstruction methods, as well as scientific programming and visualization, and can be obscure and error-prone for climatologists not familiar with all of them. Here we present a Python package called `cfr` that is designed to make CFR workflows easy to understand and perform, with the aim of improving the efficiency and reproducibility of CFR-related research.

The paper is organized as follows: Sect. 2 presents the design philosophy of the package. Section 3 provides a primer for the two classes of reconstruction methods presently included in the package. Section 4 introduces the architecture and major modules. Sections 5 and 6 describe the reconstruction workflows of two methods, namely paleoclimate data assimilation and graphical expectation–maximization, respectively. A summary and an outlook are provided in Sect. 7.

2 The `cfr` design philosophy

`cfr` aims to provide intuitive and effective workflows for climate field reconstruction with the following features:

1. *Reproducible computational narratives.* Although a traditional script-based style is also supported, `cfr` is natively designed to be used in the context of reproducible computational narratives known as Jupyter Notebooks (Kluyver et al., 2016), which provides an interactive programming laboratory for data analysis and visualization and has become the new standard for analysis-driven scientific research.

2. *Intuitive.* `cfr` is coded in the object-oriented programming (OOP) style, which provides intuitive ways to manipulate data objects. For instance, each proxy record object (`ProxyRecord`) supports a collection of methods for analysis and visualization; proxy record objects can be added together (by the operator "+") to form a proxy database object (`ProxyDatabase`) that supports another collection of methods specific to a proxy database. So-called "method cascading" is also supported to allow for smooth processing of the data objects in one line of code combining multiple processing steps.

3. *Flexible.* `cfr` provides multiple levels of modularization. It supports object-specific workflows, which provide in-depth operations of the data objects, and workflows specific to reconstruction tasks, enabling macroscopic manipulations of the reconstruction-related tasks.

4. *Community-based.* `cfr` is built upon community efforts on (paleoclimate) data analysis, modeling, and visualization, including but not limited to NumPy (van der Walt et al., 2011), SciPy (Virtanen et al., 2020), Pandas (McKinney, 2010), Xarray (Hoyer and Hamman, 2017), Matplotlib (Hunter, 2007), Cartopy (Met Office, 2010), Seaborn (Waskom, 2021), Plotly (Plotly Technologies Inc., 2015), Statsmodels (Seabold and Perktold, 2010), Pyleoclim (Khider et al., 2022), and PRYSM (Dee et al., 2015). This makes the codebase of `cfr` not only concise and efficient but also ready for grassroots open development.

5. *User-friendly.* `cfr` is designed to be easy to install and use. It is supported with an extensive documentation on the installation and the essential application programming interface (API), as well as a growing number of tutorial notebooks illustrating its usage. In addition, the commonly used proxy databases and gridded climate data for CFR applications can be remotely loaded from the cloud with `cfr`'s data fetching functionality.

3 Reconstruction methods

At the moment, cfr supports two classes of reconstruction methods, though it is designed to accommodate many more.

3.1 Offline paleoclimate data assimilation (PDA)

In recent years, paleoclimate data assimilation (PDA) has provided a novel way to reconstruct past climate variations (Jones and Widmann, 2004; Goosse et al., 2006, 2010; Gebhardt et al., 2008; Widmann et al., 2010; Annan and Hargreaves, 2013; Steiger et al., 2014, 2018; Shi et al., 2019; Hakim et al., 2016; Franke et al., 2017; Acevedo et al., 2017; Tierney et al., 2020; Osman et al., 2021; King et al., 2021; Lyu et al., 2021; Zhu et al., 2022; Valler et al., 2022; Annan et al., 2022; Fang et al., 2022). PDA proceeds by drawing from a prior distribution of climate states, which it updates by comparison with observations (Wikle and Berliner, 2007). The prior comes from physics-based model simulations, typically from general circulation model experiments covering past intervals (e.g., Otto-Bliesner et al., 2016).

The observations, in this case, are the values of climate proxies, which indirectly sense the climate field(s) of interest (e.g., surface temperature, precipitation, and wind speed). The link between latent climate states and paleo-proxy observations is provided by observation operators (one for each proxy type), which in this case are called proxy system models (PSMs; Evans et al., 2013). PSMs are low-order representations of the processes that translate climate conditions to the physical or chemical observations made on various proxy archives, such as tree rings, corals, ice cores, lake and marine sediments, or cave deposits (speleothems).

cfr implements a version of a data assimilation algorithm known as an offline ensemble Kalman filter (EnKF), popularized in the paleoclimate context by Steiger et al. (2014); Hakim et al. (2016); Tardif et al. (2019). The code for the EnKF solver is derived from the Last Millennium Reanalysis (LMR) codebase (https://github.com/modons/LMR, last access: 24 April 2024) and was streamlined with utilities from the cfr package. The output of this algorithm is a time-evolving distribution (the "posterior") quantifying the probability of particular climate states over time.

PDA can reconstruct as many climate fields as are present in the prior, though they will not be equally well constrained by the observations. Prior values are usually obtained by drawing n (100 to 200) samples at random from a model simulation at the outset. This precludes sampling variations, ensuring that all variability in the posterior is driven by the observations.

Uncertainty quantification is carried out by random sampling of a subset of the proxy set (typically 75 %) for separate training and validation. This process is typically repeated Monte Carlo (MC) (20–50) times, yielding so many MC "iterations" of the reconstruction, each of which contains a unique set of n samples from the prior. The posterior is therefore composed of $n \times \mathrm{MC}$ trajectories, typically numbering in the thousands. For scalar indices like the global mean surface temperature or the North Atlantic Oscillation index, the entire ensemble is exported. However, for climate fields, this added dimension would yield unacceptably large files. As a compromise, for each field variable, cfr by default exports the ensemble mean of the n reconstructions during each Monte Carlo iteration, and the final output thus contains MC ensemble members. In case the full ensemble is needed, cfr also provides a switch to export the entire $n \times$ MC members of the reconstructed fields.

The framework treats each target year for reconstruction in a standalone manner, i.e., the reconstruction of a certain year will not affect that of the others. This design is a key feature of offline DA. It alleviates the issue of non-uniform temporal data availability and facilitates the parallel reconstruction of multiple years.

3.2 Graphical expectation–maximization (GraphEM)

GraphEM (Guillot et al., 2015, hereafter G15) is a variant of the popular regularized expectation–maximization (RegEM) algorithm introduced by Schneider (2001) and popularized in the paleoclimate reconstruction context by Michael Mann and co-authors (Rutherford et al., 2005; Mann et al., 2007, 2008, 2009; Emile-Geay et al., 2013a, b). Like RegEM, GraphEM mitigates the problems posed to the traditional EM algorithm (Dempster et al., 1977) by the presence of rank-deficient covariance matrices, due to the "large p, small n" problem; climate fields are observed over large grids (e.g., a 5° × 5° global grid contains nearly 2500 grid points) but a comparatively short time (say 170 annual samples for the period 1851–2020). GraphEM addresses this issue by exploiting the conditional independence relations inherent to a climate field (Vaccaro et al., 2021a); i.e., surface temperature at a given grid point is conditionally independent of the vast majority of the rest of the grid, except for a handful of neighbors, which is typically a fraction of a percent of the total grid size. Effectively, this reduces the dimensionality of the problem to the point that the estimation is well-posed, and the various submatrices of the covariance matrix $\mathbf{\Sigma}$ are invertible and well estimated.

The degree of regularization is determined by the density of non-zero entries in the inverse covariance matrix $\mathbf{\Omega}$, which can be used to construct a graph adjacency matrix (Lauritzen, 1996). The more zero entries in this matrix, the sparser the graph, and the more damped the estimation. The fuller the graph, the more connections are captured between distant locales, but the more potential there is for a non-invertible covariance matrix, resulting in unphysical values of the reconstructed field. Therefore, the quality of the GraphEM estimation hinges on an appropriate choice of graph, which is a compromise between the two poles described above. In practice, it is helpful to split $\mathbf{\Omega}$ into the following three parts:

- $\mathbf{\Omega}_{\mathrm{FF}}$ stands for "field–field" and encodes covariances intrinsic to a given field, for instance covariance between temperature at different grid points.
- $\mathbf{\Omega}_{\mathrm{FP}}$ stands for "field–proxy" and encodes the way the proxies covary with the target field (here, temperature).
- $\mathbf{\Omega}_{\mathrm{PP}}$ stands for "proxy–proxy" and encodes how the values of one proxy may be predictive of another.

This last part is the identity matrix because a proxy cannot provide any more information about another proxy than the underlying climate field can (Guillot et al., 2015). Therefore, the only parts of $\mathbf{\Omega}$ that need be specified are $\mathbf{\Omega}_{\mathrm{FF}}$ and $\mathbf{\Omega}_{\mathrm{FP}}$. `cfr` supports the following two ways to do so:

- Neighborhood graphs are the simplest method, identifying as neighbors all grid points that lie within a certain radius (say 1000 km) of a given location. The smaller the radius, the sparser the graph; the larger the radius, the fuller the graph. Neighborhood graphs take no time to compute, but their structure is fairly rigid, as its sole dependence on distance means that it cannot exploit anisotropic features like land–ocean boundaries, orography, or teleconnection patterns.
- Graphical LASSO (GLASSO) is an ℓ_1 penalized likelihood method introduced by Friedman et al. (2008) in the context of high-dimensional covariance estimation. A major advantage of GLASSO is that it can extract non-isotropic dependencies from a climate field (Vaccaro et al., 2021a). In `cfr`, the level of regularization is controlled by two sparsity parameters, which explicitly target the proportion of non-zero entries in $\mathbf{\Omega}$ (see G15 for details). `sp_FF` controls the sparsity of submatrix $\mathbf{\Omega}_{\mathrm{FF}}$, and `sp_FP` controls the sparsity of submatrix $\mathbf{\Omega}_{\mathrm{FP}}$. The smaller the sparsity parameters, the sparser the graph, and the more damped the estimate.

G15 found that for suitable choices of the sparsity parameters, the GLASSO approach yielded much better estimates of the temperature field than the neighborhood graph method. However, the GLASSO approach has two main challenges: (1) the graph optimization can be computationally intensive, and (2) it requires a complete data matrix to operate. In the case of paleoclimate reconstructions, this means that proxy series and instrumental observations of the climate field of interest must not contain any missing values over the entire calibration period (e.g., 1850–2000).

To address the first challenge, `cfr` uses a greedy algorithm to find the optimal `sp_FP` and `sp_FF`, as proposed by G15. To address the second challenge, we use GraphEM with neighborhood graphs to obtain a first guess for the climate field, use it to run GLASSO, and obtain a more flexible graph, which is then used within GraphEM on the original (incomplete) data matrix (Sect. 6). Another advantage of this method is that it provides GraphEM (an iterative method) with a reasonable first guess for the parameters of the distribution (μ_0, Σ_0), which can considerably speed up convergence. We call this approach the "hybrid" method in `cfr`.

3.3 Generalization

The `cfr` framework is designed to be quite general and can in principle accommodate any CFR method. PDA and GraphEM are only two of the methods used, for instance, in Neukom et al. (2019b), which are themselves a subset of all possible CFR methods. The two options included at this stage, based on completely different assumptions and methodologies, provide a way to test the method-dependence of a given result. This is critical, as some CFR methods have been shown to depend sensitively on the input data (Wang et al., 2015). More methods will be added into the package in subsequent versions. We also welcome contributions from the community for the inclusion of more other Python-based CFR methods. A contributor's guide can be accessed at https://fzhu2e.github.io/cfr (last access: 24 April 2024) (Zhu et al., 2024).

4 The `cfr` architecture

To make CFR workflows intuitive and flexible, `cfr` takes an object-oriented programming (OOP) approach and defines Python classes describing and organizing the essential data objects along with their connections, such as proxy records, proxy databases, and gridded climate fields, as well as the PSMs and reconstruction method solvers, under different modules, which are described in the following sections.

4.1 `proxy`: proxy data processing and visualization

The `proxy` module contains a `ProxyRecord` class and a `ProxyDatabase` class, providing a collection of operation methods for proxy data processing and visualization, which is fundamental for paleoclimate data analysis.

The `ProxyRecord` class provides a structure to store, process, and visualize a proxy record. The essential attributes and methods are listed in Table 1. For instance, suppose we have a `ProxyRecord` object named `pobj`, then calling `pobj.plot()` will visualize the proxy time series along with its location depicted on a map (Fig. 1). Basic metadata stored as its attributes will be displayed on the plot, including the proxy ID, the location in latitude and longitude, the proxy type, and the proxy variable with the unit if available. There are methods for basic processing; calling `pobj.slice()` will slice the record based on the given time span ("index slicing" is also supported; for the details, please refer to the notebook tutorial at https://fzhu2e.github.io/cfr/notebooks/proxy-ops.html#Slice-a-ProxyRecord, last access: 24 April 2024; see Zhu et al., 2024), calling `pobj.annualize()` will annualize/seasonalize the record based on the specified list of months, and calling `pobj.center()` or `pobj.standardize()` will center or standardize the value axis of the record. There are also methods related to proxy system modeling; the `get_clim()` method helps to get the grid point value from a gridded climate field nearest to the record, after which the `get_pseudo()` method can be called to generate the pseudoproxy estimate. To support intuitive conversions between data objects, multiple `ProxyRecord` objects can be added together with the "+" operator to form a `ProxyDatabase` that we introduce in the next section (i.e., `pdb = pobj1 + pobj2 + ...`).

The `ProxyDatabase` class provides a structure to organize multiple `ProxyRecord` objects at once. The essential attributes and methods are listed in Table 2. For example, suppose we have a `ProxyDatabase` object named `pdb`, then calling `pdb.plot()` will visualize the proxy database on a static map (Fig. 2 top), along with the count of the records by proxy type if the argument `plot_count=True` is set (Fig. 2 middle). In contrast, calling `pdb.plotly()` will visualize the proxy database on an interactive map that allows query of the location and proxy type of each record by hovering a mouse pointer (Fig. 2 bottom). The `from_df()` and `to_df()` methods allow data input and output in the form of a `pandas.DataFrame`, a common format for tabular data. The `filter()` method offers a handy tool to filter the proxy database in flexible ways, such as by proxy types, latitude/longitude ranges, center location and radius distance, and arbitrary tag combinations. For more details, please refer to the notebook tutorial on this topic at https://fzhu2e.github.io/cfr/notebooks/pp2k-pdb-filter.html (last access: 24 April 2024) (Zhu et al., 2024). To support intuitive conversions between data objects, multiple `ProxyRecord` objects and/or `ProxyDatabase` objects can be added together with the "+" operator to form a `ProxyDatabase` (i.e., `pdb = pdb1 + pobj1 + ...`), and multiple `ProxyRecord` objects can be removed from a `ProxyDatabase` with the "−" operator (i.e., `pdb = pdb1 - pobj1 - ...`). `ProxyDatabase` also comes with `find_duplicates()` and `squeeze_dups()` to locate and remove duplicate records when adding multiple databases together. In addition, with the `fetch` method, the commonly used proxy databases can be conveniently fetched from the cloud. For instance, `pdb = cfr.ProxyDatabase.fetch('PAGES2kv2')` will remotely load the PAGES 2k phase 2 database (PAGES 2k Consortium, 2017), and `pdb = cfr.ProxyDatabase.fetch('pseudoPAGES2k/ppwn_SNRinf_rta')` will fetch and load the base version of the "pseudoPAGES2k" database (Zhu et al., 2023a, b). Its argument can also be an arbitrary URL to a supported file stored in the cloud to support flexible loading of the data. If the `fetch` method is called without an argument, the supported predefined entries will be printed out to give a hint to the users.

4.2 `climate`: gridded climate data processing and visualization

The `climate` module comes with a `ClimateField` class that is essentially an extension of `xarray.DataArray` to better fit CFR applications. It contains a collection of methods for processing and visualizing the gridded climate model simulation and instrumental observation data.

The essential attributes and methods are listed in Table 3. For instance, suppose we have a `ClimateField` object named `fd`, then calling `fd.get_anom()` will return the anomaly field against a reference time period, and calling `fd.get_anom().plot()` (an example of method cascading) will plot the anomaly field on a map (Fig. 3). The spatial resolution and coverage of the field can be altered by calling `fd.regrid()` and `fd.crop()`. Since `ClimateField` is based on `xarray.DataArray`, the slice method applies similarly; however, we augmented the original method to more robustly account for time and calendars in the paleoclimate context. For more details on time slicing, please refer to the notebook tutorial at https://fzhu2e.github.io/cfr/notebooks/climate-ops.html#Time-slicing (last access: 24 April 2024) (Zhu et al., 2024). Similar to a `ProxyRecord`, a `ClimateField` with monthly time resolution can be annualized/seasonalized by calling the `annualize()` method with a specified list of months. To support convenient calculation of many popular climate indices such as global mean surface temperature (GMST) and NINO3.4, the `geo_mean()` method calculates the area-weighted geospatial mean over a region defined by the latitude and longitude range. To support quick validation of a reconstructed field, the `compare()` method facilitates the comparison against a reference field in terms of a metric such as the grid-point-wise linear correlation (r), coefficient of determination (R^2), or coefficient of efficiency (CE). Last, the `load_nc()` and `to_nc()` methods allow loading and writing gridded datasets in the netCDF format (Rew and Davis, 1990), while the `fetch` method can be used to conveniently load gridded datasets from the cloud. For instance, `fd = cfr.ClimateField.fetch('iCESM_past1000historical/tas')` will fetch and load the iCESM-simulated surface temperature field since the year 850 CE (Brady et al., 2019). Similar to that of `ProxyDatabase`, the argument can also be an arbitrary URL to a netCDF file hosted in the cloud, and calling without an argument will print out supported predefined entries.

4.3 `ts`: time series processing and visualization

The `ts` module is for time series processing and visualization in general, and it comes with an `EnsTS` class to handle ensemble time series. Specifically, the `EnsTS` class is mainly designed to visualize and validate the reconstructed time series such as GMST and NINO3.4, and its essential attributes and methods are listed in Table 4. The

Table 1. The essential attributes and methods of `cfr.ProxyRecord`.

Attributes	Description	Methods	Description
pid	The unique proxy ID.	plot()	Plot the record time series.
time	The time axis.	slice()	Slice the record based on a give time span.
value	The value axis.	+	Combine multiple `ProxyRecord` items into a `ProxyDatabase`.
ptype	The proxy type.	annualize()	Annualize/seasonalize the proxy record.
lat, lon, elev	The latitude, longitude, and elevation.	get_clim()	Get climate data nearest to the record.
seasonality	The seasonality.	get_pseudo()	Generate the pseudoproxy estimate.
dt	The median of time differences.	center()	Center the record against a reference time span.
tags	A tag set for filtering purposes.	standardize()	Standardize the value axis of the record.

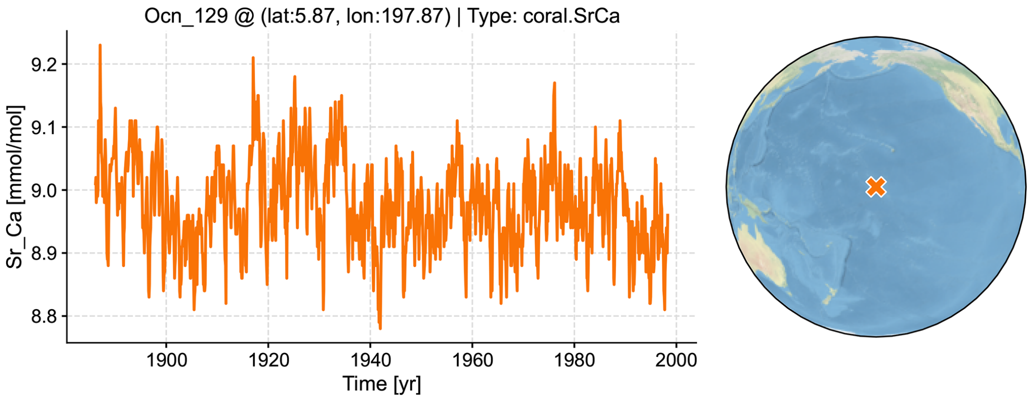

Figure 1. A visualization example of a coral Sr/Ca record using `cfr.ProxyRecord.plot()`.

`plot()` method visualizes each member of the ensemble time series, while the `plot_qs()` method plots only the quantile envelopes (Fig. 4). Similar to `ClimateField`, the `compare()` method is useful for a quick comparison of the ensemble median against a reference time series. The `fetch` method is supported as well. For instance, `bc09 = cfr.ProxyDatabase.fetch('BC09_NINO34')` will remotely load the NINO3.4 estimate by Bunge and Clarke (2009).

4.4 psm: proxy system modeling

The `psm` module incorporates classes for multiple popular proxy system models (PSMs; Evans et al., 2013), including a univariate linear regression model (Hakim et al., 2016; Tardif et al., 2019) (`Linear`), a bivariate linear regression model (Hakim et al., 2016; Tardif et al., 2019) (`Bilinear`), a tree-ring width model VS-Lite (Tolwinski-Ward et al., 2011, 2013; Zhu and Tolwinski-Ward, 2023) (`VSLite`), a simple lake varve thickness model (Zhu et al., 2023a) (`Lake_VarveThickness`), the ice core δ^{18}O model (`Ice_d18O`), coral δ^{18}O model (`Coral_d18O`), and coral Sr/Ca ratio model (`Coral_SrCa`) adopted from PRYSM (Dee et al., 2015), among which `Linear` and `Bilinear` are commonly used in paleoclimate data assimilation (PDA), and others are more useful to generate pseudoproxy emulations (Zhu et al., 2023a, b). A summary of available PSMs in `cfr` is listed in Table 5.

Taking the default univariate linear regression model (`Linear`) as a typical example, the essential attributes and methods are listed in Table 6. As illustrated in the notebook tutorial at https://fzhu2e.github.io/cfr/notebooks/psm-linear.html (last access: 24 April 2024) (Zhu et al., 2024), a `Linear` PSM object can be initialized with a `ProxyRecord` object, and by calling the `calibrate()` method, the PSM utilizes the proxy measurement and the nearest instrumental observation to calibrate the regression coefficients over the instrumental period, with the option to test multiple seasonality candidates, yielding an optimal regression model. Then by calling the `forward()` method, the calibrated model will forward the nearest model simulated climate and generate the pseudoproxy estimate, translating the climate signal to the proxy space. For more illustrations of the other available PSMs, please refer to https://fzhu2e.github.io/cfr/ug-psm.html (last access: 24 April 2024) (Zhu et al., 2024).

4.5 da: data assimilation

The `da` module implements the data assimilation algorithms, currently including the ensemble Kalman filter (Kalman, 1960; Evensen, 2009) used in the last millennium reanalysis (LMR; Hakim et al., 2016; Tardif et al., 2019) framework.

Table 2. The essential attributes and methods of cfr.ProxyDatabase.

Attributes	Description	Methods	Description
records	The dictionary of the records.	plot()	Plot the database on a static map.
pids	The list of proxy IDs.	plotly()	Plot the proxy database on an interactive map.
nrec	The number of the proxy records.	from_df()	Load the database from a pandas.DataFrame.
type_dict	The count of each proxy type.	to_df()	Convert the database to a pandas.DataFrame.
lats, lons	Location information in lists.	filter()	Filter the database by supported conditions.
		+	Combine multiple ProxyRecords/ProxyDatabase together.
		−	Removing multiple ProxyRecord from a ProxyDatabase.
		find_duplicates()	Find duplicated records.
		squeeze_dups()	Remove duplicated records and keep one.
		fetch()	Fetch and load the database from cloud.

Table 3. The essential attributes and methods of cfr.ClimateField.

Attributes	Description	Methods	Description
da	The data in xarray.DataArray.	plot()	Plot the climate field on a map.
		get_anom()	Calculate the anomaly field against a reference time period.
		regrid()	Regrid the field.
		crop()	Crop the field.
		annualize()	Annualize/seasonalize the field.
		geo_mean()	Calculate the geospatial mean over a region.
		compare()	Compare against a reference field.
		load_nc()	Load the field from a netCDF file.
		to_nc()	Output the field to a netCDF file.
		fetch()	Fetch and load the gridded climate field from cloud.

The class is named EnKF, and Table 7 lists its essential attributes and methods. Direct operations on this class are not recommended unless for debugging purposes. It is designed to be called internally by the ReconJob class that we introduce later for a high-level control of the workflow.

4.6 graphem: graphical expectation maximization

The graphem module implements the GraphEM algorithm (Guillot et al., 2015). The class is named GraphEM, and Table 8 lists its essential attributes and methods. Direct operations on this class are not recommended unless for debugging purposes. Similar to da, graphem is also designed to be called internally by the ReconJob class.

4.7 reconjob: reconstruction workflow management

The reconjob module provides pre-defined reconstruction workflows attached to the class ReconJob, as listed in Table 9. We will illustrate the usage of these methods in detail in the sections on PDA and GraphEM workflows.

4.8 reconres: result analysis and visualization

The reconres module focuses on postprocessing and visualization of the reconstruction results. It contains a ReconRes class that helps to load the reconstruction outputs stored in netCDF files from a given directory path and organize the data in the form of either ClimateField or EnsTS. The essential attributes and methods of ReconRes are listed in Table 10. We will illustrate the usage in detail in the sections on PDA and GraphEM workflows.

5 cfr's PDA workflow

In this section, we illustrate the cfr workflow (Fig. 5) with a reconstruction experiment taking the last millennium reanalysis (LMR; Hakim et al., 2016; Tardif et al., 2019) paleoclimate data assimilation (PDA) approach. A similar pseudoproxy reconstruction experiment can be accessed at https://fzhu2e.github.io/cfr/notebooks/pp2k-ppe-pda.html (last access: 24 April 2024) (Zhu et al., 2024).

The task here is to assimilate tropical coral records and reconstruct the boreal winter (December–February, DJF) surface temperature field, which can be used to calculate the El Niño–Southern Oscillation (ENSO) indices (NINO3.4 in this example). The "iCESM1" last millennium simulation (Brady et al., 2019) is utilized as the model prior, and the coral records from the PAGES 2k phase 2 database (PAGES 2k Consortium, 2017) are used as the observations to update the prior. NASA Goddard's Global Surface Temperature Analysis (GISTEMP) (Lenssen et al., 2019), combining land surface air temperatures primarily from the GHCN-M version 4

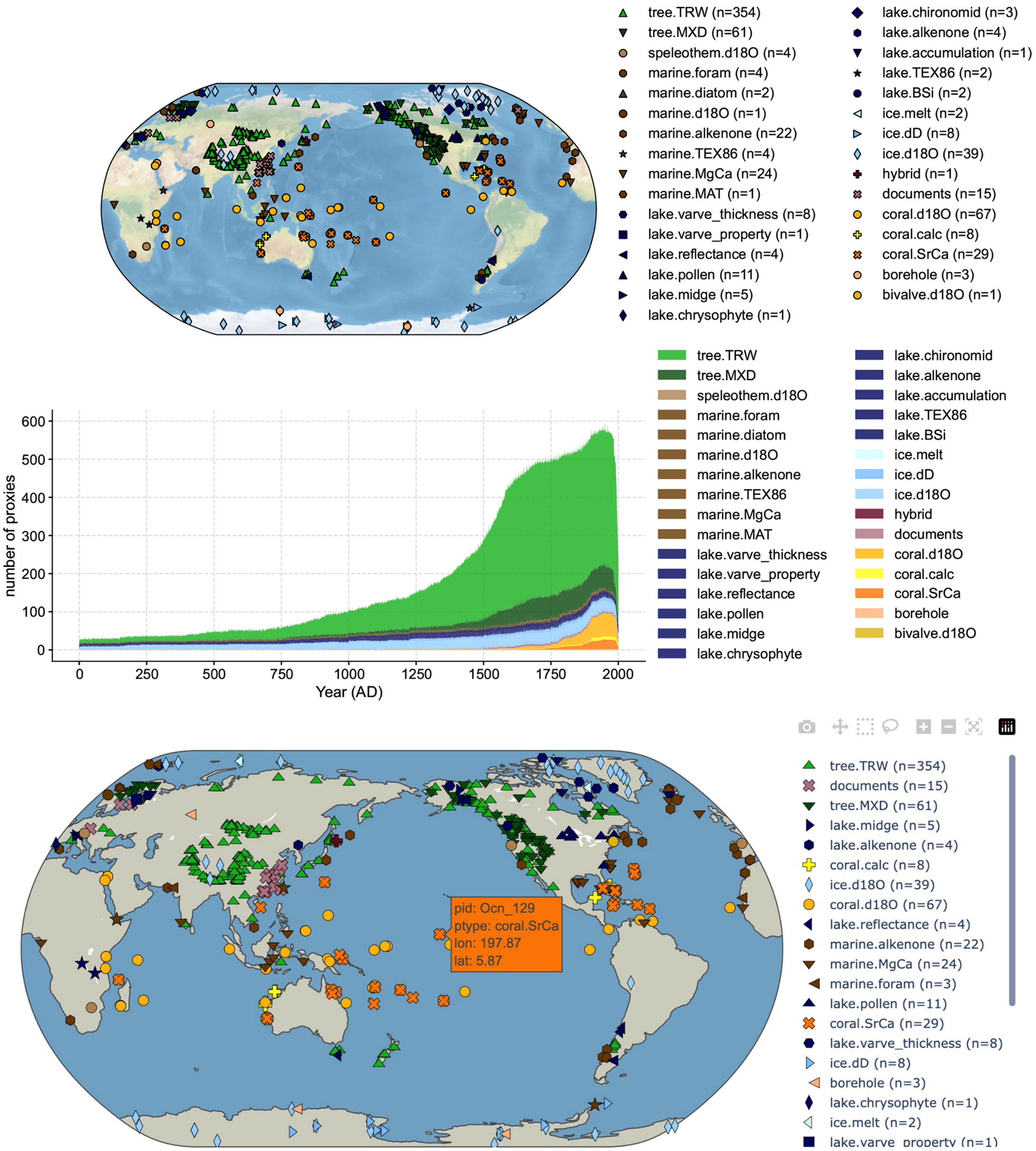

Figure 2. Visualization examples of the PAGES 2k phase 2 multiproxy database (PAGES 2k Consortium, 2017) using `cfr.ProxyDatabase.plot()` and `cfr.ProxyDatabase.plotly()`.

Table 4. The essential attributes and methods of `cfr.EnsTS`.

Attributes	Description	Methods	Description
time	The time axis.	plot()	Plot each of the ensemble time series.
value	The ensemble value in matrix.	plot_qs()	Plot the quantile envelopes.
nt	The temporal length.	compare()	Compare against a reference time series.
nEns	The ensemble size.	fetch()	Fetch and load the time series data from cloud.

Table 5. A summary of the available proxy system models (PSMs) in `cfr`.

PSM name	Class	References
Univariate linear	`cfr.psm.Linear`	Hakim et al. (2016); Tardif et al. (2019)
Multivariate linear	`cfr.psm.BiLinear`	Hakim et al. (2016); Tardif et al. (2019)
VS-Lite	`cfr.psm.VSLite`	Tolwinski-Ward et al. (2011, 2013)
Lake varve thickness	`cfr.psm.Lake_VarveThickness`	Zhu et al. (2023a)
Ice core δ^{18}O	`cfr.psm.Ice_d18O`	Dee et al. (2015)
Coral δ^{18}O	`cfr.psm.Coral_d18O`	Dee et al. (2015)
Coral Sr/Ca	`cfr.psm.Coral_SrCa`	Dee et al. (2015)

Table 6. The essential attributes and methods of the `cfr.psm.Linear` proxy system model (PSM).

Attributes	Description	Methods	Description
pobj	The `ProxyRecord` object.	calibrate()	Calibrate the PSM.
climate_required	The list of variable names of the required climate input.	forward()	Forward the calibrated model.

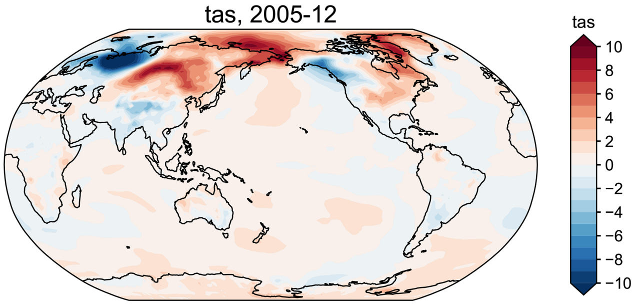

Figure 3. A visualization example of an iCESM-simulated surface temperature anomaly field using `cfr.ClimateField.plot()`.

(Menne et al., 2018) with the ERSSTv5 sea surface temperature analysis (Huang et al., 2017), is used as the calibration target for the PSMs. Finally, a spatially completed version of the near-surface air temperature and sea surface temperature analyses product HadCRUT4.6 (Morice et al., 2012) leveraging the GraphEM algorithm (Guillot et al., 2015; Vaccaro et al., 2021a) and the "BC09" NINO3.4 reanalysis (Bunge and Clarke, 2009) are used as the validation target for the posterior, i.e., the reconstruction.

5.1 Load the proxy records, model simulations, and instrumental observations

First of all, we import the `cfr` package and create a `ReconJob` object named `job`.

```
import cfr
job = cfr.ReconJob()
```

Next, we load the PAGES 2k proxy database (PAGES 2k Consortium, 2017) from the cloud and pick coral records by calling the `filter_proxydb()` method and then annualize them to be boreal winter average by calling `annualize_proxydb()`.

```
job.load_proxydb('PAGES2kv2')
job.filter_proxydb(by='ptype',
   keys=['coral'])
job.annualize_proxydb(months=[12, 1, 2],
   ptypes=['coral'])
```

Now we load the model simulated prior "iCESM1" (Brady et al., 2019) and the instrumentally observed climate data GISTEMP (Lenssen et al., 2019) by calling the `load_clim()` method.

```
job.load_clim(tag='prior', path_dict={'tas':
   'iCESM_past1000historical/tas'},
   anom_period=(1951, 1980))
job.load_clim(tag='obs', path_dict={'tas':
   'gistemp1200_GHCNv4_ERSSTv5'},
   anom_period=(1951, 1980),
   rename_dict={'tas': 'tempanomaly'})
```

Here, the `tag` argument is used to specify whether the loaded climate data will be used as prior or observation. The `rename_dict` argument is used to map variable names if the netCDF file does not name a variable as we assumed internally in `cfr`. For instance, we assume "lat" for latitude, "lon" for longitude, and "time" for the temporal dimension. The `anom_period` argument specifies against which time period we will calculate the anomaly.

Note that here we assume both the prior and observation fields being regularly gridded (lat/lon) datasets, which ensures a fast nearest-neighbor search in later processing steps. Any irregularly gridded datasets should be regridded beforehand using the `cfr.utils.regrid_field_curv_rect()` function or other third-party tools such as `scipy.interpolate.griddata` or ESMPy

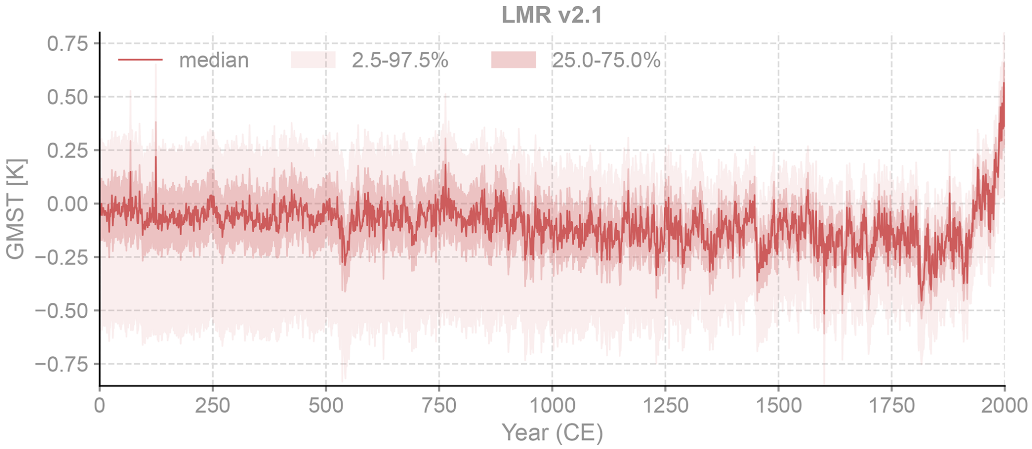

Figure 4. A visualization example of the LMR v2.1 ensemble global mean surface temperature (GMST) (Tardif et al., 2019) using `cfr.EnsTS.plot_qs()`.

Table 7. The essential attributes and methods of `cfr.da.EnKF`.

Attributes	Description	Methods	Description
prior	The dictionary of prior fields.	gen_prior_samples()	Generate prior samples.
pdb_assim	The assimilated `ProxyDatabase`.	gen_Ye()	Generate the forward estimates.
seed	The seed for randomization.	gen_Xb()	Generate the background matrix.
nens	The ensemble size.	update_yr()	Update a specific year utilizing the EnKF solver.
recon_vars	The names of the reconstructed variables.	run()	Run the EnKF solver through multiple years.

(https://earthsystemmodeling.org/esmpy, last access: 24 April 2024).

5.2 Calibrate and forward the PSMs

With the loaded proxy data, climate simulation, and instrumental observation, we are ready to calibrate the proxy system models for each proxy record by calling the `calib_psms()` method. For coral records, we use the univariate linear regression model `Linear`, as defined by the `ptype_psm_dict` argument. The `ptype_season_dict`, `calib_period`, and `ptype_clim_dict` arguments specify the season, time span (in year), and climate variables involved for calibration, respectively.

```
job.calib_psms(
    ptype_psm_dict={'coral.d18O': 'Linear',
    'coral.calc': 'Linear',
    'coral.SrCa': 'Linear'},
    ptype_season_dict={'coral.d18O':
    [12, 1, 2],  'coral.calc': [12, 1, 2],
    'coral.SrCa': [12, 1, 2]},
    ptype_clim_dict={'coral.d18O': ['tas'],
    'coral.calc': ['tas'],
    'coral.SrCa': ['tas']},
    calib_period=(1850, 2015))
```

In this example, we have specified a fixed seasonality for the whole proxy database. Users may also perform the calibration of each proxy record through a `for` loop of the proxy database, specifying individual seasonalities manually or based on the metadata if available. For records whose corresponding PSM is successfully calibrated, a "calibrated" tag is added to the record to facilitate later filtering. Proxy records with low calibration skill will be assigned a small weight during the assimilation process as per the EnKF algorithm, so there is no need to set a calibration threshold to manually filter the records. However, with the flexibility of the package, users have the freedom to easily filter the proxy database, using the `cfr.ProxyDatabase.filter()` method, based on any calibration threshold they like.

Once the PSMs are calibrated (and filtered), they can be applied by simply calling the `forward_psms()` method.

```
job.forward_psms()
```

5.3 Define the season, resolution, and domain of the reconstruction

Now we annualize the prior by calling the `annualize_clim()` method and specify on which resolution and domain we would like to conduct the reconstruction by calling the `regrid_clim()` and `crop_clim()` method.

Table 8. The essential attributes and methods of `cfr.graphem.GraphEM`.

Attributes	Description	Methods	Description
field_r	The reconstructed field.	fit()	Estimate the parameters and reconstruct the climate field.
proxy_r	The reconstructed proxy matrix.		
calib	Indices of the calibration period.		

Table 9. The essential attributes and methods of `cfr.ReconJob`.

Attributes	Description	Methods	Description
configs	The dictionary of configurations.	load_proxydb()	Load the proxy database.
proxydb	The loaded `ProxyDatabase`.	filter_proxydb()	Filter the proxy database.
prior	The dictionary of prior fields.	annualize_proxydb()	Annualize the proxy database.
obs	The dictionary of instrumental observations.	split_proxydb()	Split the proxy database.
		load_clim()	Load the simulated or observed climate data.
		annualize_clim()	Annualize the climate data.
		regrid_clim()	Regrid the climate data.
		crop_clim()	Crop the climate data.
		calib_psms()	Calibrate the PSMs for each proxy record.
		forward_psms()	Forward the PSMs for each proxy record.
		run_da()	Run the DA solver.
		run_da_mc()	Run DA with Monte Carlo iterations.
		prep_graphem()	Prepare data for the GraphEM solver.
		run_graphem()	Run the GraphEM solver.

```
job.annualize_clim(tag='prior',
    months=[12, 1, 2])
job.regrid_clim(tag='prior', nlat=42,
    nlon=63)
job.crop_clim(tag='prior', lat_min=-35,
    lat_max=35)
```

Here we annualize the prior to boreal winter and regrid the field to a global grid with 42 latitudes and 63 longitudes before cropping the domain to be within the 35° N to 35° S band where the corals are located.

5.4 Conduct the Monte Carlo iterations of the data assimilation steps

Once the pre-processing is complete, we are ready to conduct the data assimilation step. We call the `run_da_mc()` method to perform Monte Carlo iterations of the EnKF assimilation steps. The argument `save_dirpath` specifies the directory path where we store the reconstruction results, and the argument `recon_seeds` specifies the seed for randomization for each Monte Carlo iteration.

```
job.run_da_mc(save_dirpath=
    './recons/lmr-real-pages2k',
    recon_seeds=list(range(1, 11)))
```

5.5 Validate the reconstruction

Once the Monte Carlo iterations are done, we should have several netCDF files stored in the specified directory, and we can initiate a `ReconRes` object by specifying the directory path and load the results by calling the `load()` method.

```
res =
    cfr.ReconRes('./recons/lmr-real-pages2k')
res.load(['nino3.4', 'tas'])
```

Note that the variable names listed in the `load()` method are predefined. `'nino3.4'` refers to the NINO3.4 index, and the reconstructed ensemble time series will be formed as a `EnsTS` object. `'tas'` refers to the surface temperature field and the ensemble mean will be formed as a `ClimateField` object. These two objects will be organized in a dictionary named `res.recons`, and their original `xarray.DataArray` forms will be stored as `res.recons['nino3.4'].da` and `res.recons['tas'].da` for more universal purposes.

To evaluate the reconstruction skill of the surface temperature field, we load a reference target named "HadCRUT4.6_GraphEM" (Vaccaro et al., 2021a), which is a spatially completed version of the near-surface air temperature and sea surface temperature analyses product HadCRUT4.6 (Morice et al., 2012) leveraging the GraphEM algorithm (Guillot et al., 2015; Vaccaro et al., 2021a), and we validate both the prior and the posterior (`res.recons['tas']`) by calling the `compare()` method over the 1874–2000 CE time span, which returns another `ClimateField` object, and we are able to visualize it by calling its `plot()` method.

```
target = cfr.ClimateField().
    fetch('HadCRUT4.6_GraphEM', vn='tas').
```

Table 10. The essential attributes and methods of `cfr.ReconRes`.

Attributes	Description	Methods	Description
paths	The paths of the reconstruction netCDF files.	load()	Load the reconstruction files.
recons	The dictionary of reconstruction objects.	valid()	Perform validation against given targets.
da	The dictionary of reconstructions in `xarray.DataArray`.	plot_valid()	Visualize the validation.

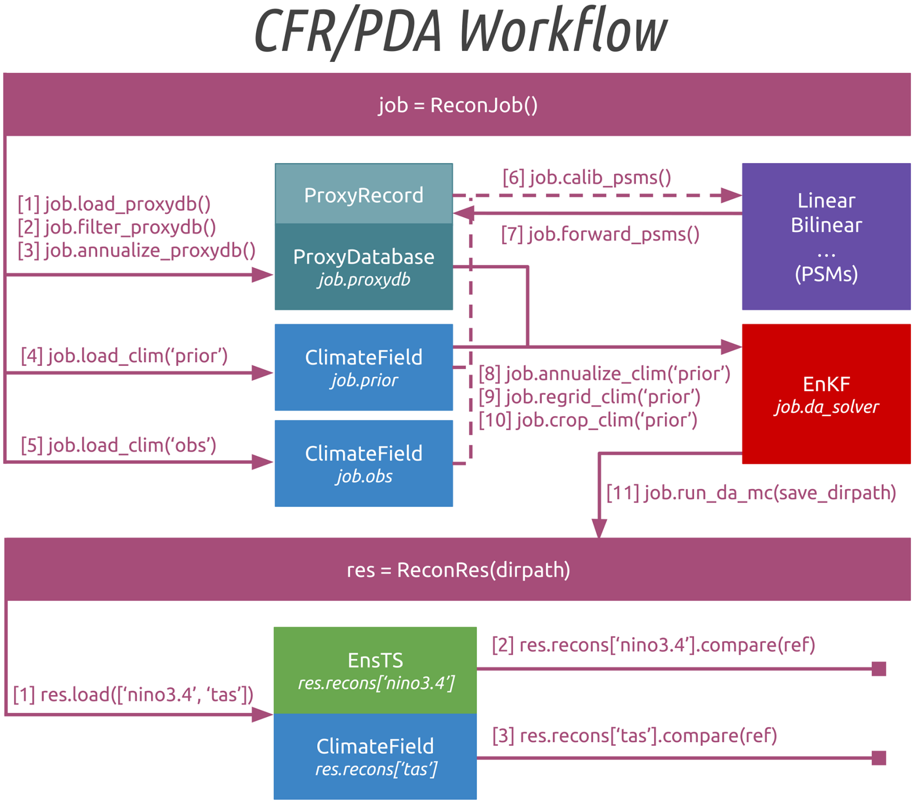

Figure 5. The `cfr` workflow for the last millennium reanalysis (LMR; Hakim et al., 2016; Tardif et al., 2019) paleoclimate data assimilation (PDA) framework. Here "nino3.4" and "tas" stand for two examples of the reconstructed variables for validation, referring to the El Niño–Southern Oscillation index series calculated as the spatial average of surface temperature within the tropical Pacific region (5° N–5° S, 170–120° W) and the surface temperature field, respectively. PSMs is for proxy system models, and EnKF is for ensemble Kalman filter.

```
    get_anom((1951, 1980))
target = target.annualize(months=[12, 1, 2]).
    crop(lat_min=-35, lat_max=35)

# validate the prior
stat = 'corr'
valid_fd = job.prior['tas'].compare(target,
    stat=stat, timespan=(1874, 2000))
fig, ax = valid_fd.plot(
    title=f'{stat}(prior, obs),
    mean={valid_fd.geo_mean().
    value[0,0]:.2f}',
    projection='PlateCarree',
    latlon_range=(-32, 32, 0, 360),
    plot_cbar=False)

# validate the reconstruction
valid_fd = res.recons['tas'].compare(target,
    stat=stat, timespan=(1874, 2000))
valid_fd.plot_kwargs.
    update({'cbar_orientation': 'horizontal',
    'cbar_pad': 0.1})

fig, ax = valid_fd.plot(
    title=f'{stat}(prior, obs),
    mean={valid_fd.geo_mean().
    value[0,0]:.2f}',
    projection='PlateCarree',
    latlon_range=(-32, 32, 0, 360),
    plot_proxydb=True,
    proxydb=job.proxydb.filter(by='tag',
    keys=['calibrated']),
    plot_proxydb_lgd=True,
    proxydb_lgd_kws={'loc': 'lower left',
    'bbox_to_anchor': (1, 0)})
```

With the above code lines, we get Fig. 6a and b. Figure 6a shows the map of Pearson correlation between the prior and the reference target ("HadCRUT4.6_GraphEM"), while panel (b) shows that between the posterior (i.e., reconstruction) and the reference target, indicating that the reconstruction is working as expected and boosting the mean correlation from 0.05 to 0.34. This is one measure of the information added by the proxies.

To evaluate the reconstruction skill of the NINO3.4 index, we load a reference target named "BC09" (Bunge and Clarke, 2009) and then validate `res.recons['nino3.4']` against BC09 by calling the `compare()` method, which returns another `EnsTS` object that can be visualized by calling its `plot_qs()` method.

```
bc09 = cfr.EnsTS().fetch('BC09_NINO34').
    annualize(months=[12, 1, 2])
fig, ax = res.recons['nino3.4'].compare(bc09,
    timespan=(1874, 2000), ref_name='BC09').
    plot_qs()
ax.set_xlim(1600, 2000)
ax.set_ylabel('NINO3.4 [K]')
```

The above code lines yield Fig. 6c, and it indicates that the reconstruction skill over the instrumental period is remarkably high, with correlation coefficient $r = 0.80$, and the coefficient of efficiency CE $= 0.57$.

6 cfr's GraphEM workflow

In this section, we illustrate the `cfr` workflow (Fig. 7) for GraphEM (Sect. 3.2) with a similar reconstruction experiment setup that we presented above for the illustration of the PDA workflow. Another similar pseudoproxy reconstruction experiment can be accessed at https://fzhu2e.github.io/cfr/notebooks/pp2k-ppe-graphem.html (last access: 24 April 2024) (Zhu et al., 2024).

6.1 Load the proxy records and instrumental observations

The `cfr` workflow for GraphEM is very similar to that for PDA. The first part is loading and preprocessing the required data, including the PAGES 2k proxy records (PAGES 2k Consortium, 2017) and the GISTEMP instrumental observation (Lenssen et al., 2019). The only difference compared to the PDA workflow is that no model prior is needed. Due to the data sparsity requirement of the GraphEM method, we set a relatively small domain around the NINO3.4 region for the reconstruction (20° N–20° S, 100° W–150° E) in this specific experiment.

```
import cfr
job = cfr.ReconJob()

# load and preprocess the proxy database
job.load_proxydb('PAGES2kv2')
job.filter_proxydb(by='ptype',
    keys=['coral'])
job.annualize_proxydb(months=[12, 1, 2],
    ptypes=['coral'])

# load the instrumental observation dataset
job.load_clim(tag='obs',
    path_dict={'tas':
    'gistemp1200_GHCNv4_ERSSTv5'},
    anom_period=(1951, 1980),
    rename_dict={'tas': 'tempanomaly'})

# define the season, resolution, and domain
  of the reconstruction
job.annualize_clim(tag='obs',
    months=[12, 1, 2])
job.regrid_clim(tag='obs', nlat=42, nlon=63)
job.crop_clim(tag='obs', lat_min=-20,
    lat_max=20, lon_min=150, lon_max=260)
```

6.2 Prepare the GraphEM solver

Compared to PDA, the preparation of the solver is much easier with GraphEM. Here, we set the calibration period to be 1901–2000 CE, while the reconstruction period is 1871–2000 CE; this reserves 1871–1900 for validation. Such dates can be easily changed with the arguments `recon_period` and `calib_period`. We also process the proxy database to be more uniform by keeping only the records that span the full reconstruction time period (`uniform_pdb=True`) to make the GraphEM algorithm more efficient.

```
job.prep_graphem(recon_period=(1871, 2000),
    calib_period=(1901, 2000),
    uniform_pdb=True)
```

6.3 Run the GraphEM solver

We run the GraphEM solver with the "hybrid" approach (Sect. 3.2) with a cutoff radius (`cutoff_radius`) of 5000 km for the neighborhood graph, 2 % target sparsity for the in-field part (`sp_FF`) and 2 % target sparsity for the climate field/proxy part (`sp_FP`) of the inverse covariance matrix for the GLASSO method. For more details about these parameters, please refer to Guillot et al. (2015).

```
job.run_graphem(save_dirpath='./recons/
    graphem-real-pages2k',
    graph_method='hybrid',
    cutoff_radius=5000, sp_FF=2, sp_FP=2)
```

6.4 Validate the reconstruction

Similar to the PDA reconstruction experiment, we validate the reconstructed surface temperature field against the "HadCRUT4.6_GraphEM" dataset (Vaccaro et al., 2021a) and the reconstructed NINO3.4 index against the "BC09" reanalysis (Bunge and Clarke, 2009) but only over the 1874–1900 CE

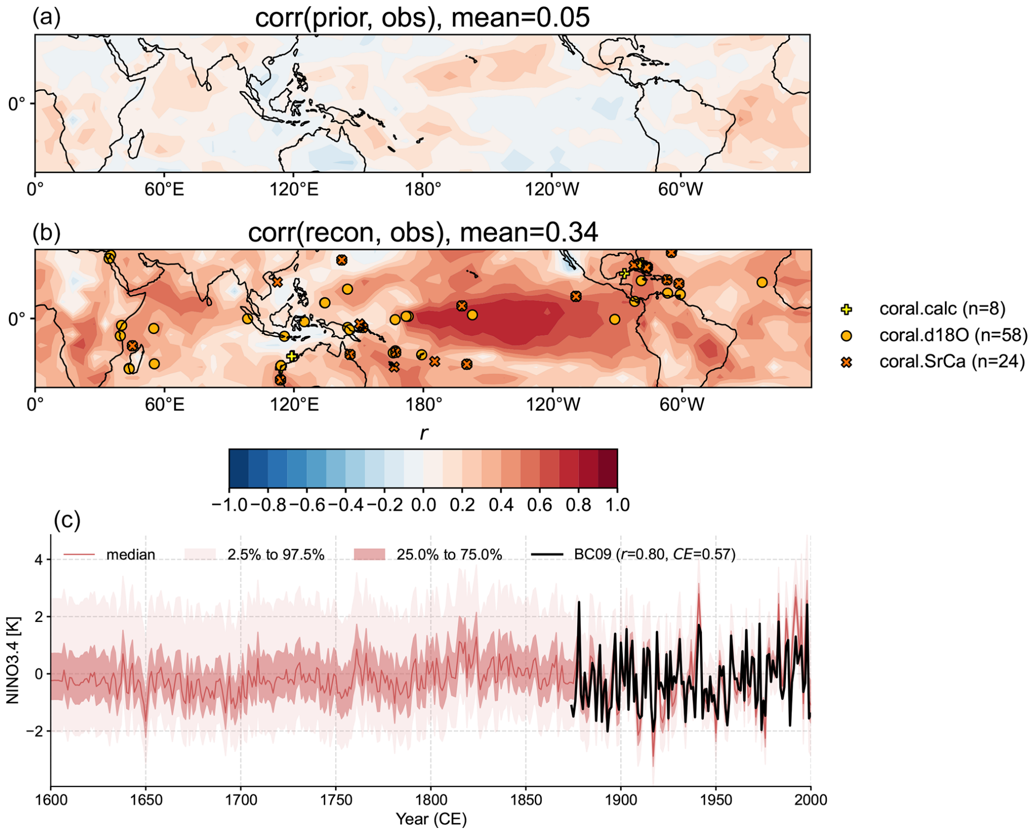

Figure 6. Validation of the reconstruction taking the paleoclimate data assimilation approach. **(a)** The correlation map between the prior field and the observation target (Vaccaro et al., 2021a). **(b)** The correlation map between the reconstruction median field and the observation target. **(c)** The correlation (r) and coefficient of efficiency (CE) between the reconstructed NINO3.4 median series and the BC09 reanalysis (Bunge and Clarke, 2009). Note that the proxy calibration is performed over 1850–2015 CE, while the validation is performed over 1874–2000 CE, indicating an overlap of calibration and validation periods. The risk of overfitting is mitigated by randomly picking model prior and proxies during each Monte Carlo iteration (Hakim et al., 2016; Tardif et al., 2019).

time span, as the reconstruction over the calibration period (1901–2000 CE) is identical to the GISTEMP observation (Fig. 8).

```
res = cfr.ReconRes('./recons/
    graphem-real-pages2k')
res.load(['nino3.4', 'tas'], verbose=True)

# validate the reconstructed surface
  temperature field
target = cfr.ClimateField().
    fetch('HadCRUT4.6_GraphEM', vn='tas').
    get_anom((1951, 1980))
target = target.annualize(months=[12, 1, 2]).
    crop(lat_min=-25, lat_max=25,
    lon_min=120, lon_max=280)

stat = 'corr'
valid_fd = res.recons['tas'].compare(target,
    stat=stat, timespan=(1874, 1900))
valid_fd.plot_kwargs.
    update({'cbar_orientation': 'horizontal',
    'cbar_pad': 0.1})
```

```
fig, ax = valid_fd.plot(
    title=f'{stat}(recon, obs),
    mean={valid_fd.geo_mean().
    value[0,0]:.2f}',
    projection='PlateCarree',
    latlon_range=(-25, 25, 0, 360),
    plot_cbar=True, plot_proxydb=True,
    proxydb=job.proxydb,
    plot_proxydb_lgd=True,
    proxydb_lgd_kws={'loc': 'lower left',
    'bbox_to_anchor': (1, 0)})

# validate the reconstructed NINO3.4
bc09 = cfr.EnsTS().fetch('BC09_NINO34').
    annualize(months=[12, 1, 2])
fig, ax = res.recons['nino3.4'].
    compare(bc09, timespan=(1874, 1900)).
    plot(label='recon')
ax.set_xlim(1870, 1900)
ax.set_ylim(-3, 4)
ax.set_ylabel('NINO3.4 [K]')
```

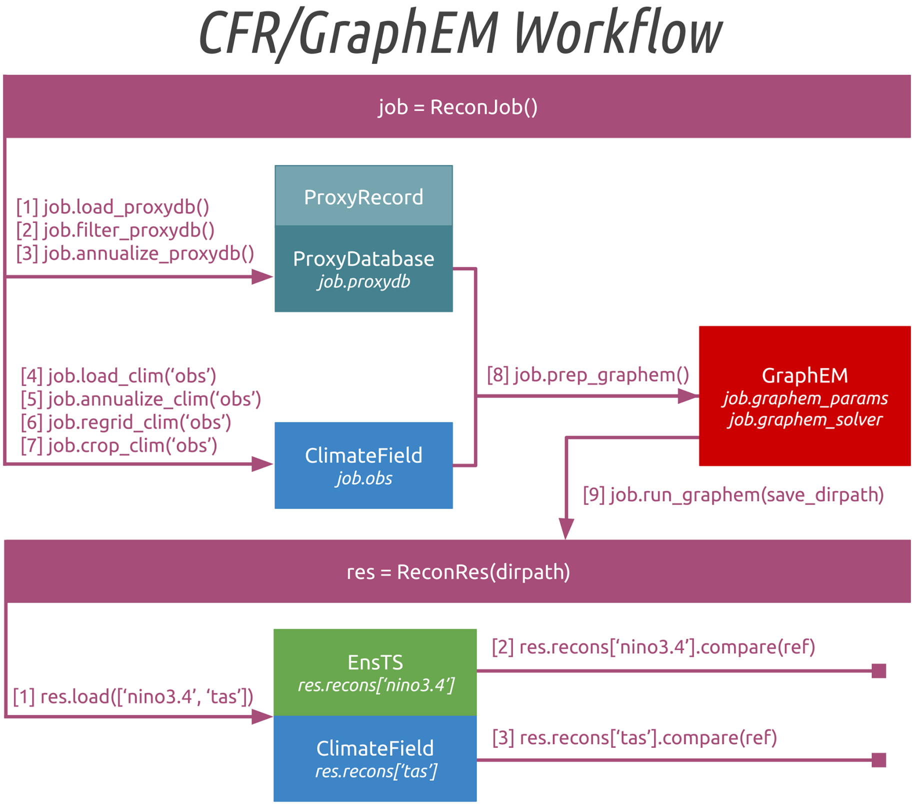

Figure 7. The `cfr` workflow for the graphical expectation–maximization (GraphEM; Guillot et al., 2015) algorithm. Here "nino3.4" and "tas" stand for two examples of the reconstructed variables for validation, referring to the El Niño–Southern Oscillation index series calculated as the spatial average of the surface temperature within the tropical Pacific region (5° N–5° S, 170–120° W) and the surface temperature field, respectively.

The result indicates that GraphEM yields a seemingly slightly better skill compared to PDA, although we note that this skill is sensitive to the parameters (`cutoff_radius`, `sp_FF`, and `sp_FP`), which in practice require some tuning work in order to be optimized. In addition, we note that the current comparison is not "apples-to-apples" due to the different domains and time spans for validation. A more systematic comparison with a consistent setup is presented in the next section.

6.5 Comparison to PDA

A main value proposition of `cfr` is to enable convenient inter-methodological comparisons of multiple reconstructions. Here, we illustrate how to conduct an "apples-to-apples" comparison of the reconstructions generated by the GraphEM and PDA approaches.

First, we create `ReconRes` objects for the two reconstructions and load the validation targets "HadCRUT4.6_GraphEM" (Vaccaro et al., 2021a) and "BC09" (Bunge and Clarke, 2009).

```
res_graphem = cfr.ReconRes('./recons/
    graphem-real-pages2k')
res_lmr =
    cfr.ReconRes('./recons/lmr-real-pages2k')

tas_HadCRUT = cfr.ClimateField().
    fetch('HadCRUT4.6_GraphEM', vn='tas').
    get_anom((1951, 1980))
tas_HadCRUT = tas_HadCRUT.
    annualize(months=[12, 1, 2])
nino34_bc09 = cfr.EnsTS().
    fetch('BC09_NINO34').
    annualize(months=[12, 1, 2])
```

Then we call the `valid()` method for the reconstructions to compute the validation statistics, including Pearson correlation and coefficient of efficiency (CE) against the same set of validation targets over the same domain (17° N–17° S, 108° W–153° E) and the same time span 1874–1900 CE.

```
res_graphem.valid(target_dict={'tas':
    tas_HadCRUT, 'nino3.4': nino34_bc09},
    timespan=(1874, 1900),
    stat=['corr', 'CE'])

res_lmr.valid(target_dict={'tas':
    tas_HadCRUT, 'nino3.4': nino34_bc09},
    timespan=(1874, 1900),
    stat=['corr', 'CE'])
```

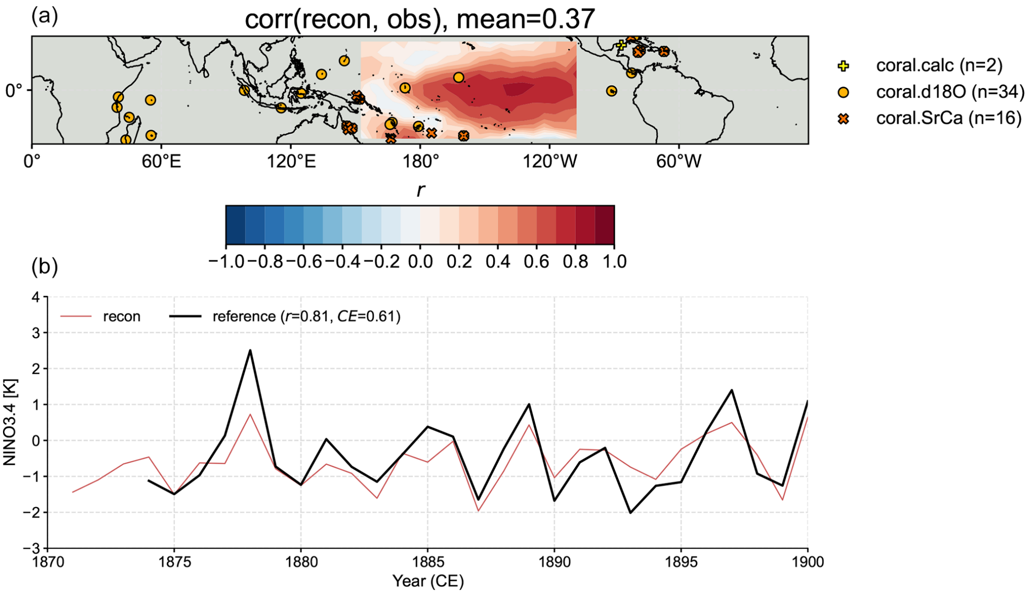

Figure 8. Validation of the reconstruction taking the graphical expectation–maximization (GraphEM) approach. **(a)** The correlation map between the reconstruction median field and the observation target (Vaccaro et al., 2021a). **(b)** The correlation (r) and coefficient of efficiency (CE) between the reconstructed NINO3.4 median series and the BC09 reanalysis (Bunge and Clarke, 2009). Note that the calibration is performed over 1901–2000 CE, while the validation is performed over 1874–1900 CE.

Finally, we call the `plot_valid()` method to visualize the validation results (Fig. 9).

```
fig, ax = res_graphem.plot_valid(
    target_name_dict={'tas': 'HadCRUT4.6',
    'nino3.4': 'BC09'},
    recon_name_dict={'tas': 'GraphEM/tas',
    'nino3.4': 'NINO3.4 [K]'},
    valid_fd_kws=
    dict(projection='PlateCarree',
    latlon_range=(-17, 17, 153, 252),
    plot_cbar=True),
    valid_ts_kws=dict(xlim=(1870, 1900),
    ylim=(-3, 4)))

fig, ax = res_lmr.plot_valid(
    target_name_dict={'tas': 'HadCRUT4.6',
    'nino3.4': 'BC09'},
    recon_name_dict={'tas': 'PDA/tas',
    'nino3.4': 'NINO3.4 [K]'},
    valid_fd_kws=
    dict(projection='PlateCarree',
    latlon_range=(-17, 17, 153, 252),
    plot_cbar=True),
    valid_ts_kws=dict(xlim=(1870, 1900),
    ylim=(-3, 4)))
```

It can be seen that the GraphEM and PDA approaches show overall comparable skill with a consistent spatiotemporal validation setup, with the mean correlation skill being around 0.5 in the specific domain (Fig. 9a and d). PDA shows fairly uniform reconstruction skill, while GraphEM's is more spatially variable and is notably higher in the eastern equatorial Pacific. This leads to a slightly higher reconstruction skill for NINO3.4 (Fig. 9c and f), as measured by both r (0.82 vs. 0.72) and CE (0.63 vs. 0.51).

The weak and negative correlation shown in Fig. 9a is located far from proxy locales and could be due to two main causes which are not mutually exclusive.

- *Covariance modeling*. The graph used here has been minimally optimized, and it is possible that it is too sparse, which would understate the strength of teleconnections. This would result in under-constrained temperature far from proxy locales, as observed here.
- *Proxy modeling*. GraphEM implicitly applies a so-called "direct regression" framework (Brown, 1994), whereby the climate variable (here, `tas`) is modeled as a linear combination of all the proxies within the neighborhood identified by the graph. This stands in contrast to the "indirect regression" framework (Brown, 1994) used as part of PDA, where proxies are modeled as linearly dependent on local temperature. In the climate context, it has been shown that direct regression is problematic and may result in high variance estimates, especially when under-constrained by observations (Tingley and Li, 2012). This high variance would manifest as temperature estimates with poor correlation to the instrumental target.

Both the GraphEM- and PDA-based reconstructions show poor CE skill over the western Pacific region (Fig. 9b and e),

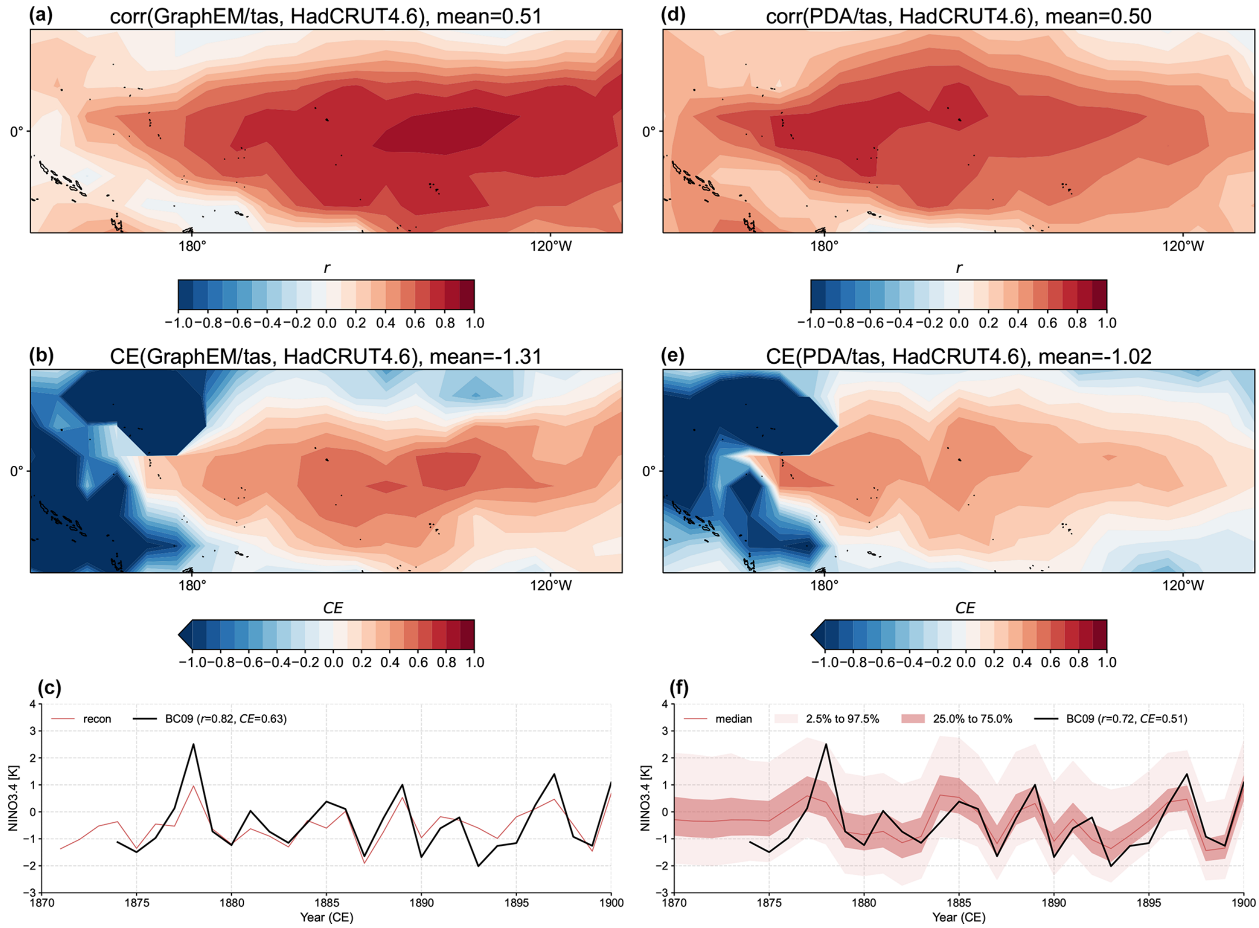

Figure 9. Comparison of the reconstructions generated by the GraphEM and PDA approaches against the same validation targets “HadCRUT4.6_GraphEM” (Vaccaro et al., 2021a) and “BC09” (Bunge and Clarke, 2009) over the same domain (17° N–17° S, 108° W–153° E) and the same time span 1874–1900 CE.

indicating large amplitude biases, which is likely due to issues in proxy modeling; here corals are considered temperature recorders, while in the western Pacific their signal is known to be largely driven by surface hydrology (e.g., Lough, 2010; Tierney et al., 2015).

By conducting more reconstruction experiments, we find that the PDA approach shows a robust ability to handle various spatiotemporal reconstruction setups, while the GraphEM approach is computationally limited to restricted reconstruction domains and time intervals so as to ensure a relatively dense data matrix, without which the EM algorithm can take a long time to converge. This computational restriction also limits the degree of graph optimization that can be performed.

It is also worth noting that the GraphEM-based reconstruction produced by `cfr` does not, at this stage, produce ensembles. In future versions, approaches such as block bootstrapping will be implemented for uncertainty quantification, following Guillot et al. (2015).

7 Summary

Climate field reconstruction provides spatial details about past climate, and the `cfr` package makes this process intuitive, modular, and efficient for Python users. In particular, `cfr` enables a sophisticated laboratory for comparisons of different climate field reconstruction methods, which is of critical importance to better estimate and interpret reconstructions, given their sensitivity to multiple error sources rooted in the climate model prior, proxy records, proxy system modeling, and the reconstruction algorithm itself. In addition, `cfr` is a handy toolbox for paleoclimate data analysis and visualization, lowering the bar for processing and visualizing proxy records, climate model output, and instrumental observations.

The current version of the package provides native support for climate field reconstruction over the Common Era, and its framework can be readily expanded for deep-time reconstructions (e.g., Tierney et al., 2020; Osman et al., 2021) through porting PSMs of deep-time proxies such as planktic foraminiferal $\delta^{18}O_c$, Mg/Ca, TEX'86, and UK'37 (e.g., Malevich et al., 2019; Tierney and Tingley, 2014, 2018; Tierney et al., 2019; King et al., 2023). As an open-source and community-based code, future versions of the package will combine contributions from the core development team and the community to support more CFR methods such as BARCAST (Tingley and Huybers, 2013), an online ensemble Kalman filter (Perkins and Hakim, 2017, 2021), particle filters (Goosse et al., 2010; Dubinkina et al., 2011; Dubinkina and Goosse, 2013; Liu et al., 2017; Shi et al., 2019; Lyu et al., 2021), and a wider array of PSMs for deep-time reconstructions, along with more postprocessing, visualization, and validation functionalities, catalyzing open- and reproducible paleoclimate research. A contributing guide is available at https://fzhu2e.github.io/cfr (last access: 24 April 2024). We hope this ethos invites more replicability and reproducibility in paleoclimate reconstructions, thereby deepening confidence in our knowledge of past climates.

Code and data availability. The `cfr` codebase is available at https://github.com/fzhu2e/cfr (Zhu and Emile-Geay, 2024) under the BSD-3 Clause license. Its documentation can be accessed at https://fzhu2e.github.io/cfr (last access: 23 April 2024). The exact version used to produce the results shown in this paper is archived on Zenodo at https://doi.org/10.5281/zenodo.10575537 (Zhu et al., 2024), which has been tested with the current mainstream Python versions 3.9, 3.10, and 3.11. Please follow the evolving installation guide at https://fzhu2e.github.io/cfr/ug-installation.html (last access: 23 April 2024) for more updated details, including the recommended Python version.

All datasets leveraged in the examples illustrated in this study can be automatically fetched from the cloud with the `cfr` remote-data-loading feature, without the hassle of manually searching and downloading, including the following:

- The PAGES 2k phase 2 global multiproxy database (PAGES 2k Consortium, 2017) hosted on the National Center for Environmental Information's World Data Service for Paleoclimatology (https://www.ncei.noaa.gov/access/paleo-search/study/21171, PAGES2k Consortium, 2024). A reorganized copy for remote-data-loading purposes hosted on Zenodo (https://doi.org/10.5281/zenodo.8367746, Zhu, 2023) and GitHub (https://github.com/fzhu2e/cfr-data/raw/main/pages2kv2.json, last access: 23 April 2024).
- The "pseudoPAGES2k" pseudoproxy dataset (Zhu et al., 2023a) hosted on Zenodo (https://doi.org/10.5281/zenodo.8173256) (Zhu et al., 2023b) and GitHub (https://github.com/fzhu2e/paper-pseudoPAGES2k, last access: 23 April 2024).
- The "iCESM1" last millennium simulation (Brady et al., 2019) hosted on a data server at the University of Washington by Rorbert Tardif (https://atmos.washington.edu/~rtardif/LMR/prior, Tardif, 2024).
- NASA Goddard's Global Surface Temperature Analysis (GISTEMP) (Lenssen et al., 2019), combining land surface air temperatures primarily from the GHCN-M version 4 (Menne et al., 2018) with the ERSSTv5 sea surface temperature analysis (Huang et al., 2017), hosted on NASA's website (https://data.giss.nasa.gov/pub/gistemp/gistemp1200_GHCNv4_ERSSTv5.nc.gz, GISTEMP Team, 2024).
- The spatially completed version of the near-surface air temperature and sea surface temperature analyses product HadCRUT4.6 (Morice et al., 2012) leveraging the GraphEM algorithm (Vaccaro et al., 2021a) hosted on Zenodo (https://doi.org/10.5281/zenodo.4601616, Vaccaro et al., 2021b) and GitHub (https://github.com/fzhu2e/cfr-data/raw/main/HadCRUT4.6_GraphEM_median.nc, last access: 23 April 2024).
- The "BC09" NINO3.4 reanalysis (Bunge and Clarke, 2009) hosted on Zenodo (https://doi.org/10.5281/zenodo.8367746 (Zhu, 2023) and GitHub (https://github.com/fzhu2e/cfr-data/raw/main/BC09_NINO34.csv, last access: 23 April 2024) for the remote-data-loading purpose.

Author contributions. FZ and JEG designed the research and drafted the paper. All authors reviewed and revised the paper and contributed to the development of the package.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. We thank Sylvia Dee and Suz Tolwinski-Ward for the development of the specific proxy system models that have been adapted in our package. We thank the NSF NCAR modeling group for publishing isotope-enabled climate simulations that have been used as the prior in our paleoclimate data assimilation experiments. We also thank both reviewers for their perceptive comments that have helped improve the quality of our paper.

Financial support. The authors acknowledge support from the National Science Foundation (grant no. 1948822 to Feng Zhu, Julien Emile-Geay, and Deborah Khider; grant nos. 2202777 and 2303530 to Feng Zhu), the Climate Program Office of the National Oceanographic and Atmospheric Administration (grant no. NA18OAR4310426 to Julien Emile-Geay; grant no. NA18OAR4310422 to Gregory J. Hakim), the Office of Naval Research (grant no. N00014-21-12437 to Deborah Khider), and the Simons Foundation Collaboration Grants for Mathematicians (grant no. 526851 to Dominique Guillot).

Review statement. This paper was edited by Yuefei Zeng and reviewed by Feng Shi and one anonymous referee.

References

Acevedo, W., Fallah, B., Reich, S., and Cubasch, U.: Assimilation of pseudo-tree-ring-width observations into an atmospheric general circulation model, Clim. Past, 13, 545–557, https://doi.org/10.5194/cp-13-545-2017, 2017.

Annan, J. D. and Hargreaves, J. C.: A new global reconstruction of temperature changes at the Last Glacial Maximum, Clim. Past, 9, 367–376, https://doi.org/10.5194/cp-9-367-2013, 2013.

Annan, J. D., Hargreaves, J. C., and Mauritsen, T.: A new global surface temperature reconstruction for the Last Glacial Maximum, Clim. Past, 18, 1883–1896, https://doi.org/10.5194/cp-18-1883-2022, 2022.

Bova, S., Rosenthal, Y., Liu, Z., Godad, S. P., and Yan, M.: Seasonal origin of the thermal maxima at the Holocene and the last interglacial, Nature, 589, 548–553, https://doi.org/10.1038/s41586-020-03155-x, 2021.

Brady, E., Stevenson, S., Bailey, D., Liu, Z., Noone, D., Nusbaumer, J., Otto-Bliesner, B. L., Tabor, C., Tomas, R., Wong, T., Zhang, J., and Zhu, J.: The Connected Isotopic Water Cycle in the Community Earth System Model Version 1, J. Adv. Model. Earth Sy., 11, 2547–2566, https://doi.org/10.1029/2019MS001663, 2019.

Brown, P. J.: Measurement, Regression, and Calibration, Oxford Statistical Science Series, vol. 12, Oxford University Press, USA, ISBN 9780198522454, 1994.

Bunge, L. and Clarke, A. J.: A Verified Estimation of the El Niño Index Niño-3.4 since 1877, J. Climate, 22, 3979–3992, https://doi.org/10.1175/2009JCLI2724.1, 2009.

Büntgen, U., Allen, K., Anchukaitis, K. J., Arseneault, D., Boucher, É., Bräuning, A., Chatterjee, S., Cherubini, P., Churakova (Sidorova), O. V., Corona, C., Gennaretti, F., Grießinger, J., Guillet, S., Guiot, J., Gunnarson, B., Helama, S., Hochreuther, P., Hughes, M. K., Huybers, P., Kirdyanov, A. V., Krusic, P. J., Ludescher, J., Meier, W. J. H., Myglan, V. S., Nicolussi, K., Oppenheimer, C., Reinig, F., Salzer, M. W., Seftigen, K., Stine, A. R., Stoffel, M., St. George, S., Tejedor, E., Trevino, A., Trouet, V., Wang, J., Wilson, R., Yang, B., Xu, G., and Esper, J.: The influence of decision-making in tree ring-based climate reconstructions, Nat. Commun., 12, 3411, https://doi.org/10.1038/s41467-021-23627-6, 2021.

Bürger, G., Fast, I., and Cubasch, U.: Climate reconstruction by regression – 32 variations on a theme, Tellus A, 58, 227–235, https://doi.org/10.1111/j.1600-0870.2006.00164.x, 2006.

Christiansen, B. and Ljungqvist, F. C.: Challenges and Perspectives for Large-Scale Temperature Reconstructions of the Past Two Millennia, Rev. Geophys., 55, 40–96, https://doi.org/10.1002/2016RG000521, 2017.

Cook, E. R., Woodhouse, C. A., Eakin, C. M., Meko, D. M., and Stahle, D. W.: Long-Term Aridity Changes in the Western United States, Science, 306, 1015–1018, https://doi.org/10.1126/science.1102586, 2004.

Cook, E. R., Anchukaitis, K. J., Buckley, B. M., D'Arrigo, R. D., Jacoby, G. C., and Wright, W. E.: Asian Monsoon Failure and Megadrought During the Last Millennium, Science, 328, 486–489, https://doi.org/10.1126/science.1185188, 2010.

Dee, S., Emile-Geay, J., Evans, M. N., Allam, A., Steig, E. J., and Thompson, D.: PRYSM: An open-source framework for PRoxY System Modeling, with applications to oxygen-isotope systems, J. Adv. Model. Earth Sy., 7, 1220–1247, https://doi.org/10.1002/2015MS000447, 2015.

Dempster, A. P., Laird, N. M., and Rubin, D. B.: Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion), J. Roy. Stat. Soc. B, 39, 1–38, 1977.

Dirren, S. and Hakim, G. J.: Toward the assimilation of time-averaged observations, Geophys. Res. Lett., 32, L04804, https://doi.org/10.1029/2004GL021444, 2005.

Dubinkina, S. and Goosse, H.: An assessment of particle filtering methods and nudging for climate state reconstructions, Clim. Past, 9, 1141–1152, https://doi.org/10.5194/cp-9-1141-2013, 2013.

Dubinkina, S., Goosse, H., Sallaz-Damaz, Y., Crespin, E., and Crucifix, M.: Testing a particle filter to reconstruct climate changes over the past centuries, Int. J. Bifurcat. Chaos, 21, 3611–3618, https://doi.org/10.1142/S0218127411030763, 2011.

Emile-Geay, J., Cobb, K., Mann, M., and Wittenberg, A. T.: Estimating Central Equatorial Pacific SST variability over the Past Millennium. Part 1: Methodology and Validation, J. Climate, 26, 2302–2328, https://doi.org/10.1175/JCLI-D-11-00510.1, 2013a.

Emile-Geay, J., Cobb, K., Mann, M., and Wittenberg, A. T.: Estimating Central Equatorial Pacific SST variability over the Past Millennium. Part 2: Reconstructions and Implications, J. Climate, 26, 2329–2352, https://doi.org/10.1175/JCLI-D-11-00511.1, 2013b.

Evans, M. N., Kaplan, A., and Cane, M. A.: Pacific sea surface temperature field reconstruction from coral δ18O data using reduced space objective analysis, Paleoceanography, 17, 7-1–7-13, https://doi.org/10.1029/2000PA000590, 2002.

Evans, M. N., Tolwinski-Ward, S. E., Thompson, D. M., and Anchukaitis, K. J.: Applications of proxy system modeling in high resolution paleoclimatology, Quaternary Sci. Rev., 76, 16–28, https://doi.org/10.1016/j.quascirev.2013.05.024, 2013.

Evensen, G.: Data Assimilation: The Ensemble Kalman Filter, 2nd Edn., Springer, ISBN 978-3-642-03710-8, 2009.

Fang, M., Li, X., Chen, H. W., and Chen, D.: Arctic Amplification Modulated by Atlantic Multidecadal Oscillation and Greenhouse Forcing on Multidecadal to Century Scales, Nat. Commun., 13, 1865, https://doi.org/10.1038/s41467-022-29523-x, 2022.

Franke, J., Brönnimann, S., Bhend, J., and Brugnara, Y.: A Monthly Global Paleo-Reanalysis of the Atmosphere from 1600 to 2005 for Studying Past Climatic Variations, Scientific Data, 4, 170076, https://doi.org/10.1038/sdata.2017.76, 2017.

Friedman, J., Hastie, T., and Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso, Biostat, 9, 432–441, https://doi.org/10.1093/biostatistics/kxm045, 2008.

Gebhardt, C., Kühl, N., Hense, A., and Litt, T.: Reconstruction of Quaternary Temperature Fields by Dynamically Consistent Smoothing, Clim. Dynam., 30, 421–437, https://doi.org/10.1007/s00382-007-0299-9, 2008.

GISTEMP Team: The GISS Surface Temperature Analysis (GISTEMP), version 4, GISTEMP Team [data set], https://data.giss.nasa.gov/pub/gistemp/gistemp1200_GHCNv4_ERSSTv5.nc.gz, last access: 23 April 2024.

Goosse, H., Renssen, H., Timmermann, A., Bradley, R. S., and Mann, M. E.: Using paleoclimate proxy-data to select optimal realisations in an ensemble of simulations of the climate of the past millennium, Clim. Dynam., 27, 165–184, https://doi.org/10.1007/s00382-006-0128-6, 2006.

Goosse, H., Crespin, E., de Montety, A., Mann, M. E., Renssen, H., and Timmermann, A.: Reconstructing Surface Temperature Changes over the Past 600 Years Using Climate Model Simulations with Data Assimilation, J. Geophys. Res.-Atmos., 115, D09108, https://doi.org/10.1029/2009JD012737, 2010.

Guillot, D., Rajaratnam, B., and Emile-Geay, J.: Statistical paleoclimate reconstructions via Markov random fields, Ann. Appl. Stat., 9, 324–352, https://doi.org/10.1214/14-AOAS794, 2015.

Hakim, G. J., Emile-Geay, J., Steig, E. J., Noone, D., Anderson, D. M., Tardif, R., Steiger, N., and Perkins, W. A.: The Last Millennium Climate Reanalysis Project: Framework and First Results, J. Geophys. Res.-Atmos., 121, 6745–6764, https://doi.org/10.1002/2016JD024751, 2016.

Hoyer, S. and Hamman, J.: xarray: N-D labeled arrays and datasets in Python, Journal of Open Research Software, 5, 10, https://doi.org/10.5334/jors.148, 2017.

Huang, B., Thorne, P. W., Banzon, V. F., Boyer, T., Chepurin, G., Lawrimore, J. H., Menne, M. J., Smith, T. M., Vose, R. S., and Zhang, H.-M.: Extended Reconstructed Sea Surface Temperature, Version 5 (ERSSTv5): Upgrades, Validations, and Intercomparisons, J. Climate, 30, 8179–8205, https://doi.org/10.1175/JCLI-D-16-0836.1, 2017.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Comput. Sci. Eng., 9, 90–95, https://doi.org/10.1109/MCSE.2007.55, 2007.

Jones, J. and Widmann, M.: Reconstructing Large-scale Variability from Palaeoclimatic Evidence by Means of Data Assimilation Through Upscaling and Nudging (DATUN), in: The KIHZ project: Towards a Synthesis of Holocene Proxy Data and Climate Models, edited by: Fischer, H., Kumke, T., Lohmann, G., Flösser, G., Miller, H., von Storch, H., and Negendank, J., Springer, Heidelberg, Berlin, New York, 171–193, ISBN 978-3-642-05826-4, 2004.

Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems, J. Fluid. Eng.-T. ASME, 82, 35–45, https://doi.org/10.1115/1.3662552, 1960.

Khider, D., Emile-Geay, J., Zhu, F., James, A., Landers, J., Ratnakar, V., and Gil, Y.: Pyleoclim: Paleoclimate Timeseries Analysis and Visualization With Python, Paleoceanography and Paleoclimatology, 37, e2022PA004509, https://doi.org/10.1029/2022PA004509, 2022.

King, J., Tierney, J., Osman, M., Judd, E. J., and Anchukaitis, K. J.: DASH: a MATLAB toolbox for paleoclimate data assimilation, Geosci. Model Dev., 16, 5653–5683, https://doi.org/10.5194/gmd-16-5653-2023, 2023.

King, J. M., Anchukaitis, K. J., Tierney, J. E., Hakim, G. J., Emile-Geay, J., Zhu, F., and Wilson, R.: A data assimilation approach to last millennium temperature field reconstruction using a limited high-sensitivity proxy network, J. Climate, 1, 1–64, https://doi.org/10.1175/JCLI-D-20-0661.1, 2021.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C.: Jupyter Notebooks – a publishing format for reproducible computational workflows, in: Positioning and Power in Academic Publishing: Players, Agents and Agendas, edited by: Loizides, F. and Schmidt, B., IOS Press, 87–90, https://doi.org/10.3233/978-1-61499-649-1-87, 2016.

Lauritzen, S. L.: Graphical Models, Clarendon Press, Oxford, ISBN 9780198522195, 1996.

Lenssen, N. J. L., Schmidt, G. A., Hansen, J. E., Menne, M. J., Persin, A., Ruedy, R., and Zyss, D.: Improvements in the GISTEMP Uncertainty Model, J. Geophys. Res.-Atmos., 124, 6307–6326, https://doi.org/10.1029/2018JD029522, 2019.

Liu, H., Liu, Z., and Lu, F.: A Systematic Comparison of Particle Filter and EnKF in Assimilating Time-Averaged Observations, J. Geophys. Res.-Atmos., 122, 13155–13173, https://doi.org/10.1002/2017JD026798, 2017.

Lough, J. M.: Climate records from corals, WIRes Clim. Change, 1, 318–331, https://doi.org/10.1002/wcc.39, 2010.

Lyu, Z., Goosse, H., Dalaiden, Q., Klein, F., Shi, F., Wagner, S., and Braconnot, P.: Spatial Patterns of Multi–Centennial Surface Air Temperature Trends in Antarctica over 1–1000 CE: Insights from Ice Core Records and Modeling, Quaternary Sci. Rev., 271, 107205, https://doi.org/10.1016/j.quascirev.2021.107205, 2021.

Malevich, S. B., Vetter, L., and Tierney, J. E.: Global Core Top Calibration of δ^{18}O in Planktic Foraminifera to Sea Surface Temperature, Paleoceanography and Paleoclimatology, 34, 1292–1315, https://doi.org/10.1029/2019PA003576, 2019.

Mann, M. E., Bradley, R. S., and Hughes, M. K.: Global-scale temperature patterns and climate forcing over the past six centuries, Nature, 392, 779–787, https://doi.org/10.1038/33859, 1998.

Mann, M. E., Bradley, R. S., and Hughes, M. K.: Northern hemisphere temperatures during the past millennium: Inferences, uncertainties, and limitations, Geophys. Res. Lett., 26, 759–762, https://doi.org/10.1029/1999GL900070, 1999.

Mann, M. E., Rutherford, S., Wahl, E., and Ammann, C.: Robustness of proxy-based climate field reconstruction methods, J. Geophys. Res.-Atmos., 112, D12109, https://doi.org/10.1029/2006JD008272, 2007.

Mann, M. E., Zhang, Z., Hughes, M. K., Bradley, R. S., Miller, S. K., Rutherford, S., and Ni, F.: Proxy-based reconstructions of hemispheric and global surface temperature variations over the past two millennia, P. Natl. Acad. Sci. USA, 105, 13252–13257, https://doi.org/10.1073/pnas.0805721105, 2008.

Mann, M. E., Zhang, Z., Rutherford, S., Bradley, R. S., Hughes, M. K., Shindell, D., Ammann, C., Faluvegi, G., and Ni, F.: Global Signatures and Dynamical Origins of the Little Ice Age and Medieval Climate Anomaly, Science, 326, 1256–1260, https://doi.org/10.1126/science.1177303, 2009.

Marcott, S. A., Shakun, J. D., Clark, P. U., and Mix, A. C.: A Reconstruction of Regional and Global Temperature for the Past 11 300 Years, Science, 339, 1198–1201, https://doi.org/10.1126/science.1228026, 2013.

Masson-Delmotte, V., Schulz, M., Abe-Ouchi, A., Beer, J., Ganopolski, A., Rouco, J. G., Jansen, E., Lambeck, K., Luterbacher, J., Naish, T., Osborn, T., Otto-Bliesner, B., Quinn, T., Ramesh, R., Rojas, M., Shao, X., and Timmermann, A.: Information from Paleoclimate Archives, in: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by: Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S., Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P., Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 383–464, https://doi.org/10.1017/CBO9781107415324.013, 2013.

McKinney, W.: Data Structures for Statistical Computing in Python, in: Proceedings of the 9th Python in Science Conference, edited by: van der Walt, S. and Millman, J., 51–56, https://doi.org/10.25080/Majora-92bf1922-00a, 2010.

Menne, M. J., Williams, C. N., Gleason, B. E., Rennie, J. J., and Lawrimore, J. H.: The Global Historical Climatology Network Monthly Temperature Dataset, Version 4, J. Climate, 31, 9835–9854, https://doi.org/10.1175/JCLI-D-18-0094.1, 2018.

Met Office: Cartopy: a cartographic python library with a Matplotlib interface, Exeter, Devon, Met Office [code], https://scitools.org.uk/cartopy (last access: 23 April 2024), 2010.

Morice, C. P., Kennedy, J. J., Rayner, N. A., and Jones, P. D.: Quantifying Uncertainties in Global and Regional Temperature Change Using an Ensemble of Observational Estimates: The HadCRUT4 Data Set, J. Geophys. Res.-Atmos., 117, D08101, https://doi.org/10.1029/2011JD017187, 2012.

Neukom, R., Barboza, L. A., Erb, M. P., Shi, F., Emile-Geay, J., Evans, M. N., Franke, J., Kaufman, D. S., Lücke, L., Rehfeld, K., Schurer, A., Zhu, F., Brönnimann, S., Hakim, G. J., Henley, B. J., Ljungqvist, F. C., McKay, N., Valler, V., and von Gunten, L.: Consistent multidecadal variability in global temperature reconstructions and simulations over the Common Era, Nat. Geosci., 12, 643–649, https://doi.org/10.1038/s41561-019-0400-0, 2019a.

Neukom, R., Steiger, N., Gómez-Navarro, J. J., Wang, J., and Werner, J. P.: No evidence for globally coherent warm and cold periods over the preindustrial Common Era, Nature, 571, 550–554, https://doi.org/10.1038/s41586-019-1401-2, 2019b.

Osman, M. B., Tierney, J. E., Zhu, J., Tardif, R., Hakim, G. J., King, J., and Poulsen, C. J.: Globally resolved surface temperatures since the Last Glacial Maximum, Nature, 599, 239–244, https://doi.org/10.1038/s41586-021-03984-4, 2021.

Otto-Bliesner, B. L., Brady, E. C., Fasullo, J., Jahn, A., Landrum, L., Stevenson, S., Rosenbloom, N., Mai, A., and Strand, G.: Climate Variability and Change since 850 CE: An Ensemble Approach with the Community Earth System Model, B. Am. Meteorol. Soc., 97, 735–754, https://doi.org/10.1175/BAMS-D-14-00233.1, 2016.

PAGES 2k Consortium: A global multiproxy database for temperature reconstructions of the Common Era, Scientific Data, 4, 170088 EP, https://doi.org/10.1038/sdata.2017.88, 2017.

PAGES2k Consortium: PAGES2k Global 2,000 Year Multiproxy Database, PAGES2k Consortium [data set], https://doi.org/10.25921/ycr3-7588, last access: 23 April 2024.

Perkins, W. A. and Hakim, G. J.: Reconstructing paleoclimate fields using online data assimilation with a linear inverse model, Clim. Past, 13, 421–436, https://doi.org/10.5194/cp-13-421-2017, 2017.

Perkins, W. A. and Hakim, G. J.: Coupled Atmosphere–Ocean Reconstruction of the Last Millennium Using Online Data Assimilation, Paleoceanography and Paleoclimatology, 36, e2020PA003959, https://doi.org/10.1029/2020PA003959, 2021.

Plotly Technologies Inc.: Collaborative data science, https://plot.ly (last access: 23 April 2024), 2015.

Rew, R. and Davis, G.: NetCDF: An Interface for Scientific Data Access, IEEE Comput. Graph., 10, 76–82, https://doi.org/10.1109/38.56302, 1990.

Ridgwell, A., Hargreaves, J. C., Edwards, N. R., Annan, J. D., Lenton, T. M., Marsh, R., Yool, A., and Watson, A.: Marine geochemical data assimilation in an efficient Earth System Model of global biogeochemical cycling, Biogeosciences, 4, 87–104, https://doi.org/10.5194/bg-4-87-2007, 2007.

Rutherford, S., Mann, M. E., Osborn, T. J., Bradley, R. S., Briffa, K. R., Hughes, M. K., and Jones, P. D.: Proxy-Based Northern Hemisphere Surface Temperature Reconstructions: Sensitivity to Method, Predictor Network, Target Season, and Target Domain, J. Climate, 18, 2308–2329, 2005.

Schneider, T.: Analysis of Incomplete Climate Data: Estimation of Mean Values and Covariance Matrices and Imputation of Missing Values, J. Climate, 14, 853–871, https://doi.org/10.1175/1520-0442(2001)014<0853:AOICDE>2.0.CO;2, 2001.

Seabold, S. and Perktold, J.: statsmodels: Econometric and statistical modeling with python, in: 9th Python in Science Conference, https://doi.org/10.25080/Majora-92bf1922-011, 2010.

Shakun, J. D., Clark, P. U., He, F., Marcott, S. A., Mix, A. C., Liu, Z., Otto-Bliesner, B., Schmittner, A., and Bard, E.: Global Warming Preceded by Increasing Carbon Dioxide Concentrations during the Last Deglaciation, Nature, 484, 49–54, https://doi.org/10.1038/nature10915, 2012.

Shi, F., Zhao, S., Guo, Z., Goosse, H., and Yin, Q.: Multi-proxy reconstructions of May–September precipitation field in China over the past 500 years, Clim. Past, 13, 1919–1938, https://doi.org/10.5194/cp-13-1919-2017, 2017.

Shi, F., Goosse, H., Klein, F., Zhao, S., Liu, T., and Guo, Z.: Monopole Mode of Precipitation in East Asia Modulated by the South China Sea Over the Last Four Centuries, Geophys. Res. Lett., 46, 14713–14722, https://doi.org/10.1029/2019GL085320, 2019.

Smerdon, J. E. and Pollack, H. N.: Reconstructing Earth's surface temperature over the past 2000 years: the science behind the headlines, WIREs Clim. Change, 7, 746–771, https://doi.org/10.1002/wcc.418, 2016.

Steiger, N. and Hakim, G.: Multi-timescale data assimilation for atmosphere–ocean state estimates, Clim. Past, 12, 1375–1388, https://doi.org/10.5194/cp-12-1375-2016, 2016.

Steiger, N. J., Hakim, G. J., Steig, E. J., Battisti, D. S., and Roe, G. H.: Assimilation of Time-Averaged Pseudoproxies for Climate Reconstruction, J. Climate, 27, 426–441, https://doi.org/10.1175/JCLI-D-12-00693.1, 2014.

Steiger, N. J., Smerdon, J. E., Cook, E. R., and Cook, B. I.: A Reconstruction of Global Hydroclimate and Dynamical Variables over the Common Era, Scientific Data, 5, 1–15, https://doi.org/10.1038/sdata.2018.86, 2018.

Tardif, R.: The post-processed "iCESM1" last millennium simulation, https://atmos.washington.edu/~rtardif/LMR/prior, last access: 23 April 2024.

Tardif, R., Hakim, G. J., Perkins, W. A., Horlick, K. A., Erb, M. P., Emile-Geay, J., Anderson, D. M., Steig, E. J., and Noone, D.: Last Millennium Reanalysis with an expanded proxy database and seasonal proxy modeling, Clim. Past, 15, 1251–1273, https://doi.org/10.5194/cp-15-1251-2019, 2019.

Tierney, J. E. and Tingley, M. P.: A Bayesian, Spatially-Varying Calibration Model for the TEX86 Proxy, Geochim. Cosmochim. Ac., 127, 83–106, https://doi.org/10.1016/j.gca.2013.11.026, 2014.

Tierney, J. E. and Tingley, M. P.: BAYSPLINE: A New Calibration for the Alkenone Paleothermometer, Paleoceanography and Paleoclimatology, 33, 281–301, https://doi.org/10.1002/2017PA003201, 2018.

Tierney, J. E., Abram, N. J., Anchukaitis, K. J., Evans, M. N., Giry, C., Kilbourne, K. H., Saenger, C. P., Wu, H. C., and Zinke, J.: Tropical sea surface temperatures for the past four centuries reconstructed from coral archives, Paleoceanography, 30, 226–252, https://doi.org/10.1002/2014PA002717, 2015.

Tierney, J. E., Malevich, S. B., Gray, W., Vetter, L., and Thirumalai, K.: Bayesian Calibration of the Mg/Ca Paleothermometer in Planktic Foraminifera, Paleoceanography and Paleoclimatology, 34, 2005–2030, https://doi.org/10.1029/2019PA003744, 2019.

Tierney, J. E., Zhu, J., King, J., Malevich, S. B., Hakim, G. J., and Poulsen, C. J.: Glacial Cooling and Climate Sensitivity Revisited, Nature, 584, 569–573, https://doi.org/10.1038/s41586-020-2617-x, 2020.

Tingley, M. P. and Huybers, P.: Recent temperature extremes at high northern latitudes unprecedented in the past 600 years, Nature, 496, 201–205, https://doi.org/10.1038/nature11969, 2013.

Tingley, M. P. and Huybers, P. J.: A Bayesian Algorithm for Reconstructing Climate Anomalies in Space and Time. Part 1: Development and Applications to Paleoclimate Reconstruction Problems, J. Climate, 23, 2759–2781, https://doi.org/10.1175/2009JCLI3015.1, 2010a.

Tingley, M. P. and Huybers, P. J.: A Bayesian Algorithm for Reconstructing Climate Anomalies in Space and Time. Part 2: Comparison with the Regularized Expectation-Maximization Algorithm, J. Climate, 23, 2782–2800, https://doi.org/10.1175/2009JCLI3016.1, 2010b.

Tingley, M. P. and Li, B.: Comments on "Reconstructing the NH Mean Temperature: Can Underestimation of Trends and Variability Be Avoided?", J. Climate, 25, 3441–3446, https://doi.org/10.1175/JCLI-D-11-00005.1, 2012.

Tingley, M. P., Craigmile, P. F., Haran, M., Li, B., Mannshardt, E., and Rajaratnam, B.: Piecing together the past: statistical insights into paleoclimatic reconstructions, Quaternary Sci. Rev., 35, 1–22, https://doi.org/10.1016/j.quascirev.2012.01.012, 2012.

Tolwinski-Ward, S. E., Evans, M. N., Hughes, M. K., and Anchukaitis, K. J.: An efficient forward model of the climate controls on interannual variation in tree-ring width, Clim. Dynam., 36, 2419–2439, https://doi.org/10.1007/s00382-010-0945-5, 2011.

Tolwinski-Ward, S. E., Anchukaitis, K. J., and Evans, M. N.: Bayesian parameter estimation and interpretation for an intermediate model of tree-ring width, Clim. Past, 9, 1481–1493, https://doi.org/10.5194/cp-9-1481-2013, 2013.

Vaccaro, A., Emile-Geay, J., Guillot, D., Verna, R., Morice, C., Kennedy, J., and Rajaratnam, B.: Climate field completion via Markov random fields – Application to the HadCRUT4.6 temperature dataset, J. Climate, 1, 1–66, https://doi.org/10.1175/JCLI-D-19-0814.1, 2021a.

Vaccaro, A. D., Emile-Geay, J., Guillot, D., Verna, R., Morice, C., Kennedy, J., and Rajaratnam, B.: GraphEM-infilled HadCRUT4.6.0.0 January 1850 – December 2017, Zenodo [data set], https://doi.org/10.5281/zenodo.4601616, 2021b.

Valler, V., Franke, J., Brugnara, Y., and Brönnimann, S.: An Updated Global Atmospheric Paleo-Reanalysis Covering the Last 400 Years, Geosci. Data J., 9, 89–107, https://doi.org/10.1002/gdj3.121, 2022.

van der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy Array: A Structure for Efficient Numerical Computation, Comput. Sci. Eng., 13, 22–30, https://doi.org/10.1109/MCSE.2011.37, 2011.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nat. Methods, 17, 261–272, https://doi.org/10.1038/s41592-019-0686-2, 2020.

Wang, J., Emile-Geay, J., Guillot, D., McKay, N. P., and Rajaratnam, B.: Fragility of reconstructed temperature patterns over the Common Era: Implications for model evaluation., Geophys. Res. Lett., 42, 7162–7170, https://doi.org/10.1002/2015GL065265, 2015.

Waskom, M. L.: seaborn: statistical data visualization, Journal of Open Source Software, 6, 3021, https://doi.org/10.21105/joss.03021, 2021.

Widmann, M., Goosse, H., van der Schrier, G., Schnur, R., and Barkmeijer, J.: Using data assimilation to study extratropical Northern Hemisphere climate over the last millennium, Clim. Past, 6, 627–644, https://doi.org/10.5194/cp-6-627-2010, 2010.

Wikle, C. K. and Berliner, L. M.: A Bayesian tutorial for data assimilation, Physica D, 230, 1–16, https://doi.org/10.1016/j.physd.2006.09.017, 2007.

Zhu, F.: cfr-data v1.0, Zenodo [data set], https://doi.org/10.5281/zenodo.8367746, 2023.

Zhu, F. and Emile-Geay, J.: The GitHub repository for cfr, GitHub [code], https://github.com/fzhu2e/cfr, last access: 23 April 2024.

Zhu, F. and Tolwinski-Ward, S.: PyVSL v0.4.0, Zenodo [code], https://doi.org/10.5281/zenodo.7967103, 2023.

Zhu, F., Emile-Geay, J., Hakim, G. J., King, J., and Anchukaitis, K. J.: Resolving the Differences in the Simulated and Reconstructed Temperature Response to Volcanism, Geophys. Res. Lett., 47, e2019GL086908, https://doi.org/10.1029/2019GL086908, 2020.

Zhu, F., Emile-Geay, J., Anchukaitis, K. J., Hakim, G. J., Wittenberg, A. T., Morales, M. S., Toohey, M., and King, J.: A re-appraisal of the ENSO response to volcanism with paleoclimate data assimilation, Nat. Commun., 13, 747, https://doi.org/10.1038/s41467-022-28210-1, 2022.

Zhu, F., Emile-Geay, J., Anchukaitis, K. J., McKay, N. P., Stevenson, S., and Meng, Z.: A Pseudoproxy Emulation of the PAGES 2k Database Using a Hierarchy of Proxy System Models, Scientific Data, 10, 624, https://doi.org/10.1038/s41597-023-02489-1, 2023a.

Zhu, F., Emile-Geay, J., Anchukaitis, K. J., McKay, N. P., Stevenson, S., and Meng, Z.: The pseudoPAGES2k Dataset v1.4, Zenodo [data set], https://doi.org/10.5281/zenodo.8173256, 2023b.

Zhu, F., Emile-Geay, J., Hakim, G. J., Guillot, D., Khider, D., Tardif, R., and Perkins, W. A.: cfr: a Python package for Climate Field Reconstruction, Zenodo [data set], https://doi.org/10.5281/zenodo.10575537, 2024.