Environmental Modelling and Software 150 (2022) 105359

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft

MarineTools.temporal: A Python package to simulate Earth and environmental time series

M. Cobos [a,b,*], P. Otiñar [b], P. Magaña [a,b], A. Lira-Loarca [c], A. Baquerizo [a,b]

[a] Department of Structural Mechanics and Hydraulic Engineering, University of Granada, Edificio Politécnico, Campus de Fuentenueva, 18071, Granada, Spain
[b] Andalusian Institute for Earth System Research, Avda. del Mediterráneo s/n, Granada, Spain
[c] Department of Civil, Chemical and Environmental Engineering, University of Genoa, Via Montallegro 1, 16145, Genoa, Italy

ARTICLE INFO

Keywords:
Time expansion of parameters
Non-stationary probability models
Stochastic characterization
Environmental modelling

ABSTRACT

The assessment of the uncertainty about the evolution of complex processes usually requires different realizations consisting of multivariate temporal signals of environmental data. However, it is common to have only one observational set. MarineTools.temporal is an open-source Python package for the non-stationary parametric statistical analysis of vector random processes suitable for environmental and Earth modelling. It takes a single timeseries of observations and allows the simulation of many time series with the same probabilistic behavior. The software generalizes the use of piecewise and compound distributions with any number of arbitrary continuous distributions. The code contains, among others, multi-model negative log-likely functions, wrapped-normal distributions, and generalized Fourier timeseries expansion. Its programming philosophy significantly improves the computing time and makes it compatible with future extensions of scipy.stats. We apply it to the analysis of freshwater river discharge, water currents, and the simulation of ensemble projections of sea waves, to show its capabilities.

1. Introduction

In environmental and engineering sciences, the understanding and quantification of the evolution of processes are highly related to the capacity to measure the variables involved and the ability to build up models capable to reproduce their interrelationships (Moges et al., 2021). Some processes can be modeled as systems that respond to a forcing, usually represented by one or several input time series. If the driving agent has a random character, as with climate forced processes, it may be interesting to stochastically characterize the response. However, for a given measured or projected time series, the model just provides a single realization of the output. Most processes are complex enough so that their statistical nature cannot be directly inferred from the joint distributions of the input variables. Under such circumstances, simulation techniques can be used to obtain a large number of input realizations whose responses constitute a sample of the output. From that sample, it is possible to assess the uncertainty that the response inherits from the forcing (see e.g. Baquerizo and Losada (2008); Meier et al. (2019)). The study of this type of simulation techniques is nowadays a hot topic in climate driven processes (Refsgaard et al., 2007) and climate change projections (Kundzewicz et al., 2018; Ellis et al., 2021; Uhe et al., 2021). One of approaches consists in the stochastic characterization of a vector random process (VRP) coupling several probability models (PMs) which parameters are assumed non-stationary (NS), with a vector autoregressive model (VAR), which characterize the multivariate and temporal dependency [see Solari and Van Gelder (2011); Solari and Losada (2011) among others]. The theoretical applicability of the later approach has been illustrated in works where specific subsets of probability models were analyzed independently for different purposes, among others: (i) the observed wave climate variability in the preceding century and expected changes in projections under a climate change scenario (Lira-Loarca et al., 2021); (ii) the optimal design and management of an oscillating water column system (Jalón et al., 2016; López-Ruiz et al., 2018), (iii) the planning of maintenance strategies of coastal structures (Lira-Loarca et al., 2020), (iv) the analysis of monthly Wolf sunspot number over a 22 year period (Cobos et al., 2022), and (v) the simulation of estuarine water conditions for the management of the estuary (Cobos, 2020).

We follow a novel approach in *MarineTools.temporal*, a package included in *MarineTools* which is a Python framework that integrates

* Corresponding author. Department of Structural Mechanics and Hydraulic Engineering, University of Granada, Edificio Politécnico, Campus de Fuentenueva, 18071, Granada, Spain.
E-mail address: mcobosb@ugr.es (M. Cobos).

https://doi.org/10.1016/j.envsoft.2022.105359
Received 5 August 2021; Received in revised form 25 January 2022; Accepted 8 February 2022
Available online 24 February 2022
1364-8152/

software that can be used in the search for solutions to real engineering and marine problems. This "temporal" package is, to the best of the authors' knowledge, the first open-source and free available general tool aimed at providing users with a friendly, general code to statistically characterize vector random processes (VRP) and to obtain realizations of them. It generalizes the above-mentioned approaches for VRPs whose components are NS and have piecewise or compound distributions defined by means of any number of arbitrary continuous distributions. Its programming philosophy, widely used in Python, significantly improves the computing time and allows the use of any probability distributions included in future extensions of scipy.stats, the most widely used Python statistical package. It is implemented in Python - an interpreted, high-level, object-oriented programming language widely used in the scientific community - and it makes the most of the Python packages ecosystem. Among the existing Python packages, it uses Numpy, which is the fundamental package for scientific computing in Python (Harris et al., 2020), SciPy, which offers a wide range of optimization and statistics routines (Virtanen et al., 2020), Matplotlib (Hunter, 2007), that includes routines to obtain high-quality graphics, and Pandas (McKinney et al., 2010) to analyze and manipulate data. In addition to third-party packages, *MarineTools.temporal* also contains, among others, novel routines for the required negative log-likely functions, the generalized Fourier time series expansion, the use of wrapped-normal distributions, the multi model ensemble approach for climate projections and a whole set of plotting functions for the non-stationary representation.

These examples use time series commonly used in coastal engineering (sea waves significant wave height, peak period and mean direction), oceanography (sea surface temperature and currents at open sea), hydrology (river discharge), and astronomy (sunspots number). Their choice is aimed at representing all the capabilities, including (i) the analysis of circular variables, (ii) the treatment of times series with persistent low values and (iii) the analysis of multi-model climate projections. In addition, four more examples that enlarge even more the applicability of the software are included in the repository.

The paper is divided into five sections. After the introduction, the theoretical basis for fitting the parameters of the NS probability models (PMs), the description of the temporal and multivariate dependence, the time series simulation and the analysis of multi-model climate data are briefly presented in Section 2. Some applications to illustrate the use of *MarineTools.temporal* are presented in Section 3. The discussion about the algorithms, some hints about its use and expected upgrades is given in Section 4, and finally, Section 5 concludes the study.

2. Theoretical background and definition of the parameters

This tool is aimed at (1) characterizing a vector random process, $\overrightarrow{X} = (X_1(t), \ldots, X_i(t), \ldots, X_N(t))$, that can be uni- or multivariate, where t belongs to a set of index, on the basis of an observation of it at N_o discrete points, $\overrightarrow{x^o}(t_j) = (x_1^o(t_j), \ldots, x_i^o(t_j), \ldots, x_N^o(t_j))$ for $j = 1, \ldots, N_o$, and (2) simulating other random realizations that have the same joint probabilistic behavior than the first one. For the sake of simplicity, from now on we will speak about temporal time series and assume that the RPs are measured at equally spaced instants.

The characterization includes (1a) the fit of the marginal NS distribution functions of each random variable X_i and (1b) the description of the dependence of the value of each component at time t_j with previous values of all the components through a vectorial autoregressive model.

The tool can also deal with multi-model time series coming from different combinations of global and regional climate models (GCM-RCM) through the use of compound marginal variables and an ensemble averaged VAR model as proposed by Lira-Loarca et al. (2021).

The time series simulation is based on those results. It first generates a stationary multivariate time series of non-exceedance probabilities and then it recovers the values of each variable by using the inverse of the non-stationary cumulative distribution function.

Fig. 1 shows the flow chart of the algorithm. The cascade analysis consists of several steps being the initial inputs the observed or hindcasted data, $\overrightarrow{x^o}(t_j), j = 1, \ldots, N_o$ and the dictionary that contain the key parameters that provide the probability structure of the RPs, the information regarding the VAR model and the simulation. The tool has a simple syntax based on dictionaries (see Table 1) and it helps users to supply the input information, particularly for step (1a), which requires some knowledge about the phenomena under study.

At the end of each step the results of the analysis are saved to a file whose name and some information regarding the modelling process, are appended to the input dictionary.

In the following, a brief description of the methodology implemented in *MarineTools.temporal* is presented and, at the same time, the options and values of the parameters that represent the relation between mathematical and statistical methods used in the tool are commented. The interested reader is referred to Cobos et al. (2022); Lira-Loarca et al. (2021, 2020); Egüen et al. (2016); Solari and Losada (2011); Solari and Van Gelder (2011), among others, where several examples of its application can be found.

The description, values and examples of the key variables for the analysis and simulation tool are defined in Table 1. The identifiers 'MF', 'TD', 'TS', and 'EA' given in the 1st column stand for the marginal fit, the temporal dependence analysis, the time series simulation, and the ensemble average multimodel analysis of climate projections, respectively.

2.1. Fit to marginal probability models (MF options)

Each random component X_i is fitted to a marginal distribution $F_{X_i}(x_i)$ with the Maximum Likelihood method.

The variables that describe the angles of vector magnitudes such as the wind or wave direction, require a special analysis. The model allows to indicate that a variable is circular by including the key "circular" with the value `True`. Among others, circular PMs like von-Mises (Mardia and Jupp, 2009) included in Scipy, can be used. The wrapped-normal (Pewsey, 2000) that was implemented ad-hoc by following the philosophy of SciPy package is also available.

For some variables such as the precipitation or the fresh-water river discharge in water-scarce areas, the presence of abrupt changes in time makes difficult the fitting of multiple PMs. For those cases, it is suggested to apply a parametric and monotonic transformation to deal with Gaussian distributed data. For that purpose, the methods included in *MarineTools.temporal* are those proposed by (Box and Cox, 1964) and (Yeo and Johnson, 2000). This option can be set-up by adding a dictionary with the key "transform" with a set of keys. The first one, "make", indicates that the transformation will be done if it is equal to `True`. The "model" key indicates the selected transformation method (items "Box-cox" or "Yeo-Johnson"). The users can also indicate to the tool whether they want to plot the CDF of the normalized data (key "plot" equal to `True`).

2.1.1. Structure of the marginal CDFs

The distribution can be constructed from multiple PMs so that it can properly represent the non-exceedance probability of all the plausible values, including the tails and the body (see Appendix B.1). The distributions are defined either by a piecewise function (key "piecewise" equal to `True`, see Cobos et al. (2022)) or by a mixture or compound distribution (key "piecewise" equal to `False`). The first option also contemplates the use of a single distribution. In any of these choices the functions involved are specified in key "fun". For a piecewise distribution with more than one PM, the user has to provide (in key: "ws_ps") the initial guesses for the common endpoints percentiles of the intervals in which the real axes has been divided. If a compound distribution is selected, the values provided in "ws_ps", are the first guesses of the

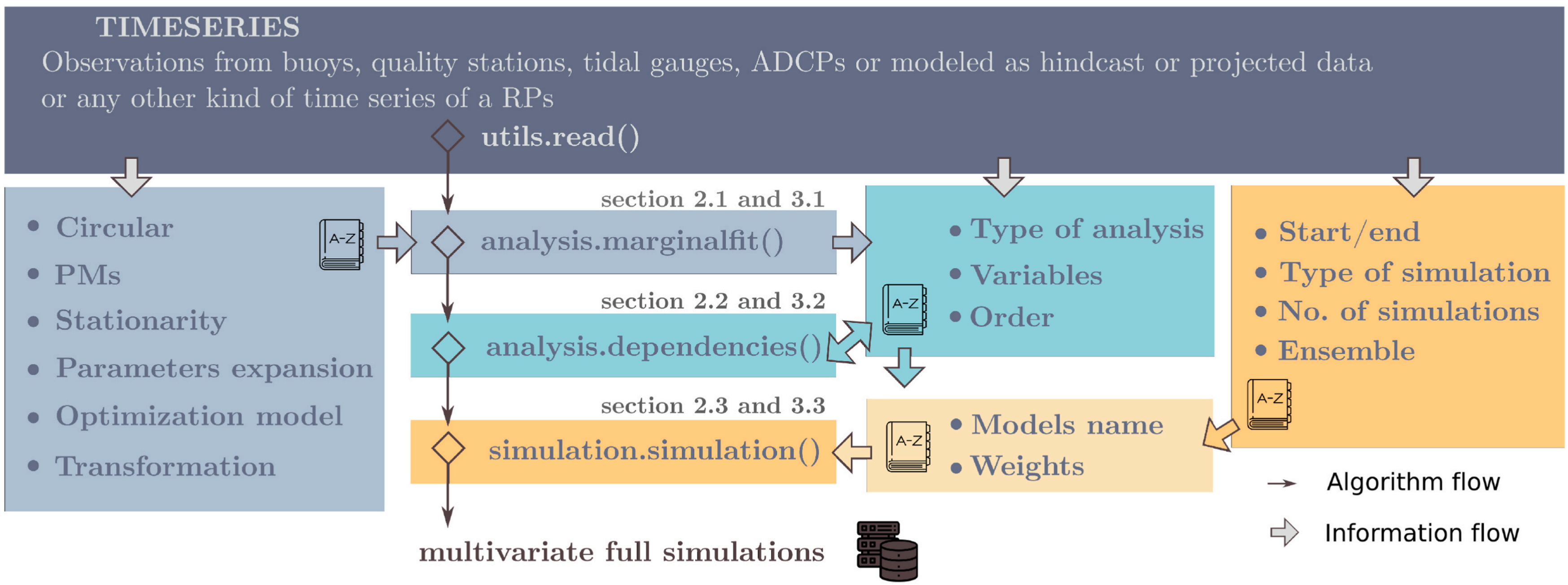

Fig. 1. Flow chart of *MarineTools.temporal*. Observations and some information about the VRPs (PMs, number of models, weights, stationarity, etc …) have to be provided. These initial guesses are collected into transverse dictionaries that focus on the different features of the tool (marginal fit, temporal dependency, simulation, and multi-model ensemble analysis). Finally, a json file (a simple format, free open and readable file with any text editor) per variable is created with the estimated parameters of the marginal fit, a json file including the multivariate and temporal dependency and several simulation files according to the number of simulations given by users.

Table 1
Description of the input dictionary to define the keys to obtain the marginal distribution of X_i, the multivariate and temporal dependency of X_i and the simulations. Asterisks in default column stand for mandatory parameters. [a] The default parameters are given in scipy.optimize.minimize.

Id.	Description	Key	Values/data type	Default	Observations	Example
	Marginal fit (see 2.1 and B.1)					
MF	**Name to identify the variable**	"var"	"X_i"	*	This name will appear in output files and in plots	"H_s"
MF	Option to deal with a **circular variable**	"circular"	`True` or `False`	`False`	Its use is recommended for variables that describe the angle of a vector field.	`True`
MF	Option to **normalize the variable** and to **plot the CDF of the normalized data**	"transform"	"make": `True` or `False`, "plot": `True` or `False`, "method": "box-cox" or "Yeo-Johnson"	`False`, `False`, `None`	It is recommended to use this option for variables that values show abrupt changes in time	{"make": `True`, "plot": `False`, "method": "box-cox"},
–	**Definition of the CDF** (see 2.1.1 and B.1)	–	–	–	–	–
MF	**Structure of the CDF**	"piecewise"	`True` or `False`	`False`	If set to `True`, the CDF is a piecewise function (eq. 1), otherwise it is a compound distribution (eq. 3).	`True`
MF	**Definition of the PMs**	"fun"	0: "function_1", 1: "function_2", …, n_F-1: "function_n_F"	*	Available functions: wrapped-normal ("wrap-norm") PM and all PMs included in SciPy package	0: "genpareto", 1: "lognormal", 2: "genextreme"
MF	Option to provide **Initial guesses** of the **percentiles** of the common endpoints or the **weights** of the PMs, for piecewise and compound distributions respectively	"ws_ps"	$[p_1 \ldots p_\alpha \ldots p_{N_l-1}]$ if "piecewise" = `True` $[\omega_1 \ldots \omega_\alpha \ldots \omega_{N_l-1}]$ if "piecewise" = `False`	List like	If multiple PMs are given, this information has to be provided	[0.2, 0.8]
–	**Use of a NS approach** (see 2.1.2 and B.2)	–	–	–	–	–
MF	Option to use a **NS approach**	"non_stat_analysis"	`True` or `False`	`True`	If it is set to `True`, the parameters are assumed to be time-dependent functions	`True`
MF	Option to provide the **number of years (N_y)** for the NS analysis	"basis_period"	Integer	1	It defines the interval $t \in [0, N_y]$ in which the NS analysis will be performed. It implicitly assumes that the signal has a periodicity of N_y. Only smaller time scales variations will be detected	22
MF	Option to specify the parameters of **basis functions for the GFS expansion** and the number of terms	"basis_function"	"method": "trigonometric", "modified", "sinusoidal", …(see Table 2), "no_terms": An integer	"trig. "	It is mandatory if "NS_analysis" is set to True. See all the available options in Table 2	{"method": "sinusoidal", "no_terms": 4}
MF	Option to choose the **optimization method** ad to set up the options	"optimization"	"method":"SLSQP", "dual_annealing", "differential_evolution", "shgo", "eps": double, "maxiter": integer, "ftol": double, "giter": double, "bounds: double	[a]	Options to run the optimization algorithm. See scipy.optimize. minimize for further information. The key "giter" stands for ensuring the convergence of the optimization algorithm by making several attempts from different initial conditions	{"method":"SLSQP", "eps":1e-7, "maxiter":1000, "ftol":1e-3, "giter":10, "bounds":0.5},
Id.	Description	Key	Values/data type	Default	Observations	Example
–	**Time dependence**(see 2.2 and B.3)	–	–	–	–	–
TD	**Names of the variables** for the uni- or multivariate time dependence analysis	"vars"	["X_1", …, "X_i", …, "X_N"]	*	–	["H_s", "T_p", "θ_m"]
TD	**Time dependence analysis model** for the uni- or multivariate analysis	"method"	"AR or "VAR"	"VAR"	–	"VAR"
TD	Maximum **order of the time dependence model**	"order"	Integer	72	The tools analyzes all the orders up to that value and selects the one with a smaller BIC value	24
TD	**Alternative analysis of storms analysis**	"events"	`True` or `False`	`False`	Storm events are defined as exceedances of the 1st variable over a certain threshold and the time dependence analysis is done between them and the concomitant values (see Lira-Loarca et al. (2020)). In this case, the marginal fit of the main variable is used for the simulation (key "mvar", see below)	`False`
TD	Main variable for **conditional analysis** (Lira-Loarca et al., 2020)	"mvar"	"X_i"	`None`	The governing variable for multivariate cases and conditional analysis	"H_s"
–	–	–	–	–	–	–

(continued on next page)

Table 1 (*continued*)

Id.	Description	Key	Values/data type	Default	Observations	Example
	Simulations of new time series (see 2.3)					
TS	**Date and time** of the start/end of the period	"start" and "end"	A string with the date	*	In the standard format of a datetime (Python)	"2000/01/01 00:00:00"
TS	**Number of simulations**	"nosim"	An integer	*	The tool obtains as many as indicated realizations (time series) of the vector RP	100
TS	**Simulation of extreme events**	"events"	`True` or `False`	`None`	The simulation method described in Lira-Loarca et al. (2020)	`True`
TS	Option to deal with of **ensemble multi-models** from GCM-RCMs	"ensemble"	`True` or `False`	`False`		`True`
–	**Use of ensemble mean** (see 2.4)	–	–	–	–	–
EA	Option to use the **ensemble average for multi-model** climate projections	"make"	`True` or `False`	`False`		`False`
EA	**Regional or Global Climate Models name**	"models"	['RCA4-MPI-ESM-LR', 'RCA4-IPSL-CM5A-MR', …, 'RCA4-CNRM-CM5', 'CCLM4-8-17-MIROC5']	`None`	The main dictionary must contain the keys defined in for the marginal fit and temporal and multivariate analysis updated with the results obtained	['RCA4-MPI-ESM-LR', 'RCA4-IPSL-CM5A-MR']
EA	**Weights of GCM-RCM models**	"weights"	"equal" or a list	"equal"	–	[0.45, 0.55]

weights of the first PM's (the weight of the last PM is derived from the others so that their sum equals 1).

2.1.2. Choice of the stationary or NS character of the analysis

The analysis can be stationary, which means that the parameters of the PMs (i.e., shape, location or scale) do not vary in time. Otherwise, they are assumed to vary over a given time interval with a duration of N_y years and, therefore, the distribution function, $F_{X_i}(x_i(t);t)$ is non-stationary. The choice is done by setting the option "NS_analysis" to `False` or `True` respectively.

In that regard, if the NS analysis is chosen, the parameters are expanded into a generalized Fourier series (GFS) over the interval [0, N_y], which is truncated to N_F terms (see Appendix B.2). The basis functions are chosen by setting up "basis_function" with one of the choices provided in Table 2 and by indicating the number of terms to be retained in the series (key "no_terms" equal to an integer number). If the basis function contains periodic functions, it is recommended to perform a previous spectral or harmonic analysis of the time series in order to gain knowledge about the oscillatory components.

The tool selects by default $N_y = 1$, which allows to analyze time variations up to the yearly scale. In some time series, longer variations associated to climatic indexes oscillations can be found (Le Mouël et al., 2019). The tool offers an option to modify the basis period to N_y years, which allows the oscillations to be analyzed on a different (see Cobos et al. (2022)).

With the choices provided by the user in the input dictionary for variable X_i, the tool internally computes an initial guess of all the parameters involved in the definition of the PMs and the coefficients of the GFS and it finds the optimum values of these coefficients and the values provided in key "ws_ps". The tool also shows some informative messages and includes some guidance plots to help in the selection of the marginal fit properties. The package also contains several functions to assess the goodness of the fit of the analysis and the simulations. Some of these functions have been used to present the results of the study cases in section 3.

Table 2
Sets of basis expansion solutions available for the GFS.

Denomination	Values
Trigonometric series expansion	"trigonometric"
Modified Fourier series expansion	"modified"
Sinusoidal series expansion	"sinusoidal"
Chebyshev series expansion	"chebyschev"
Legendre series expansion	"legendre"
Laguerre series expansion	"laguerre"

2.2. Multivariate and temporal dependency of the RPs (TD options)

Once the marginal probability structure of the random variables has been obtained, the temporal multivariate (or univariate) dependency might be inferred by using autoregressive methods. In this version the models available are the uni- and multivariate autoregressive AR(q) and VAR(q) respectively that assume a linear relationship between the value of the RPs at a given time and their past q-values. Other possibilities such as ARMA, ARIMAX, DAR, EGARCH, GAS or Gaussian local level, can be easily implemented with Python functions included in PyFlux package.

The information related to the temporal dependence is provided in a dictionary where the names of the variables to be jointly analyzed are given in key "vars" and the method selected in "method". Starting from the first order, the model tests all the orders up to the maximum value, q, indicated in "order" and provides the information from the one with the smaller value of the Bayesian Information Criterion (BIC).

It is worth noting that although in the marginal fitting of every RP, the time series can contain small gaps in time, for the assessment of the temporal univariate or multivariate dependency, the observations need to have the same resolution.

Table 1 with Id. 'TD', contains a description of the possibilities that the tool offers for multivariate temporal dependency analysis, their values, syntax and examples.

2.3. The simulation process (TS options)

The simulation process is done by retrieving the information from dictionaries of the marginal fit, multivariate and temporal dependency analysis and, if selected, the multi-model ensemble average. In addition, the tool requires the starting and ending dates of the simulation period (keys "start" and "end" in the standard datetime format of Python) and the number of simulations (key "nosim"). Then it writes in different files every random realization of the univariate or multivariate RP as output. The same temporal time step than the original dataset is used in order to guarantee the same probabilistic behavior.

2.4. Characterization of ensemble average multi-model climate projections (EA options)

Climate services provide several climate projections from different combinations of GCM-RCM climate models under a certain greenhouse gas concentration trajectory associated to a Representative Concentration Pathway (RCP) adopted by the IPCC. Lira-Loarca et al. (2021) proposed to gather the information of different GCM-RCMs through the use of a compound distribution that weights the NS marginal distributions and their VAR(q) coefficients to obtain a multi-model ensemble average characterization of the multivariate random processes.

The package includes an option (key "ensemble" equal to `True`) to work with the multi model ensemble averages, using equal weights, a choice corresponding to the commonly used rule "one model-one vote" recommended by the IPCC (Pörtner et al., 2019) or the values provided in key "weights". To do so, the marginal fit analysis and the temporal multivariate dependence analysis need to be done previously for all the models as explained in Sections 2.1 and 2.2. The tool provides several files with the non-exceedance probabilities of the compound variables and the corresponding VAR coefficients.

3. Applications to Earth and environmental modelling

In the following subsections, we illustrate the capabilities of *MarineTools.temporal* with some novel applications for the analysis of environmental VRPs. More precisely, the examples include (i) the univariate PM analysis of projections of freshwater river discharge at Alcalá del Río dam (Spain), (ii) the multivariate analysis and temporal dependency of velocity currents at the Strait of Gibraltar (Atlantic Ocean). Furthermore, (iii) the simulation of new projections from marine climate data (significant wave height, peak period and incoming mean direction, wind velocity and incoming mean direction) at the Alborán Sea (at the western Mediterranean Sea) under the RCP8.5 scenario, from the ensemble multivariate multi-model information (Lira-Loarca et al., 2021). We present a detailed description of the *MarineTools.temporal* framework methods and how they operate together to capture the non-stationarity and to create simulations of some general examples. Small pieces of code will be shown next to the exemplifying dictionaries to run the analysis.

These applications and those described in Table 3 have been selected in order to illustrate a set of representative options that the tool integrates. However, many other types of VRPs can also be considered.

The documentation is available at the GitHub (gdfa-ugr) repository. It includes the code, case studies and the examples of Table 3 (included as Jupiter notebook files). This will allow interested readers to reproduce the applications with similar synthetic data sets and to facilitate its adaptation to other time series.

3.1. Freshwater river discharge projection at Alcalá del Río dam

This first example is focused on a univariate time series, denoted by $Q_d(t)$, of the projection of daily fresh-water river discharge from the Alcalá del Río dam to the Guadalquivir river estuary (37.29° N, −6.06° W), from January, 1st of 2020 to December, 31st of 2040. This series are obtained from the Hydrological Predictions from the Environment model HYPE forced with the atmospheric model REMO 2009 for the RCP 2.6 scenario (Source: Swedish Meteorological and Hydrological Institute, SMHI). Due to the strong regulation of the river at this dam, the last one in the river course before its flow into the Atlantic Ocean, the series shows low values in summer ($Q_d < 40$ m^3/s) that are almost squared in winter ($Q_d \approx 1000$ m^3/s). To deal with the high variability between seasons, a Box-Cox transformation with $\lambda = 0.0796$ parameter is used. The properties of the marginal fit are given in Table 4. A Weibull of maxima model was selected. The highly temporal variability and the clear seasonal behavior lead to the Sinusoidal temporal expansion over $N_y = 1$ year retaining $N_F = 10$ oscillatory terms (covering frequencies up to 10 yr^{-1}).

The following code shows the Python dictionary with the input information required for the marginal fit analysis in the present example.

Table 3
Description of other datasets and main parameters used for the analysis. All these examples are located in a folder of the GitHub repository (https://github.com/gdfa-ugr).

Data Information	basis period	basis function	noterms	fun (ps ws)
Historic daily river discharges in the Guadalquivir river estuary at the last regulation point (Navarro et al., 2019), the Alcalá del Río dam (37.29° N, −6.06° W), (Source: Andalusian Water Agency, Junta de Andalucía). The regulation of this dam is aimed not only at controlling floods but also at fulfilling several water management strategies. The time series varies from very low values (usually in summer $Q < 40$ m^3/s) to those that are almost squared in winter ($Q \approx 1000$ m^3/s) with sporadic sudden changes. The analysis deals with this high variability by setting-up a Box-Cox transformation.	1	Sinusoidal	20	Weibull of maxima
Monthly Wolf sunspot number, available from 1749 (Source: WDC-SILSO, Royal Observatory of Belgium, Brussels). The signal contains the well-known 11 years Schwabe cycle, the 22 years and smaller time scales variations described in Usoskin and Mursula (2003). The analysis focus on the detection of these cycles from 22 years down to the seasonal scale.	22	Modified	44	Lognorm and Norm ($p_1 = 0.85$)
The monthly mean sea surface temperature at 38° N, 0° (in a location next to Cape Palos, Mediterranean Sea) that covers the period 01/15/1 854 - 05/15/2 021 (source: NOAA). The analysis was conducted to reproduce large scale oscillations as in the case of sunspots.	20	Trigonometric	40	Gaussian
The multivariate time series of the wind field hindcasted at 10 m above the mean sea level at the SIMAR point 1052 048 located at 37° N, 7° W in the Gulf of Cádiz (Source: Puertos del Estado, Spain). The simulations obtained were compared to the hindcasted data in terms of bivariate density function, autocorrelation and sojourns above and below some reference levels, showing a fairly well agreement.	1	Sinusoidal	Four terms for wind magnitude and 12 for wind direction	Generalized Pareto for the tails and Lognorm for the body of the wind magnitude ($p_1 = 0.05$; $p_2 = 0.96$) Two truncated normal for wind direction ($p_1 = 0.5$)

```
params["Qd"] = {
    "var": "Qd",
    "fun": {
        "0": "weibull_max"
    },
    "basis_function": {"method": "sinusoidal",
        "no_terms": 20
    },
    "transform": {
        "make": True,
        "plot": False,
        "method": "box-cox"
    }
}
```

With the previous dictionary (params), the dataset as a pandas DataFrame (data), and importing the package analysis as
`from marinetools.temporal import analysis`
the marginal fit function is invoked just coding
`analysis.marginalfit(data, params).`
The results will be saved in a file in a new folder called "marginalfit".
Fig. 2 shows the empirical and theoretical (fitted) NS-CDF. The empirical NS-CDF was computed by using a window size of 14 days which is large enough to obtain representative values of the empirical percentiles but not so long that lower significant variations are neglected. In the authors' experience for climatic variables with this temporal resolution, a window length of 14 days is appropriate for time series longer than 20 years. As it is observed in Fig. 2, the NS-PM adequately reproduces the non-stationary behavior during the year. The theoretical PM captures the overall seasonal behavior during almost all the year for all the percentiles, with a marked valley during summer and peaks in the previous and subsequent seasons. Only some deviations from the peaky behavior are observed during February–March and at the end of November for the highest represented percentile, 0.99, which is slightly underestimated.

3.2. Currents at the Strait of Gibraltar

The second example analyzes the multivariate time series of the water current field (mean current velocity, U, and mean incident current direction, θ_U) hindcasted at 0.5058 m below the mean sea level at a point located in 35.9166° N, 5.5° W at the Strait of Gibraltar (data provided by Marine Copernicus System). The hindcast time series has ≈27 years duration, with data that spans from 1993/01/01 to 2019/12/31 with a daily temporal cadence. The IBI (Iberian Biscay Irish) ocean Reanalysis system provides 3D ocean fields (product identifier "IBI_MULTIYEAR_PHY_005_002"). The IBI model numerical core is based on the NEMO v3.6 ocean general circulation model run at 1/12° horizontal resolution.

The marginal fits were carried out with the characteristics given in Table 4. Briefly, the univariate analysis of U was carried out using a Gaussian PM. The water current incident direction was fitted using a Weibull of maxima PM. For both variables the trigonometric Fourier expansion was performed over a basis period of one year ($N_y = 1$) with eight oscillatory components ($N_F = 8$). Fig. 3.a shows the empirical and theoretical models of the NS-CDF for U. As it is observed, the Gaussian model fairly reproduces the temporal variation of the probability distribution function. Fig. 3.b shows the same analysis with θ_U. A marked eastwards flow (around 270°) is observed, showing that the mean currents at that location of the Strait of Gibraltar and at the selected depth flow into the Mediterranean Sea, which is in accordance with observations of the water masses exchanges between the Atlantic Ocean and the Mediterranean sea (see e.g., Sverdrup and Fleming (1942)). A Weibull of maxima reproduced also fairly well the probability structure in time, slightly overestimating the water current directions at the highest percentiles.

Once the parameters of the marginal distributions were obtained, the multivariate and temporal analysis were carried out. The maximum q-order analyzed of the VAR model was 72. These properties are set-up in the dictionary as:

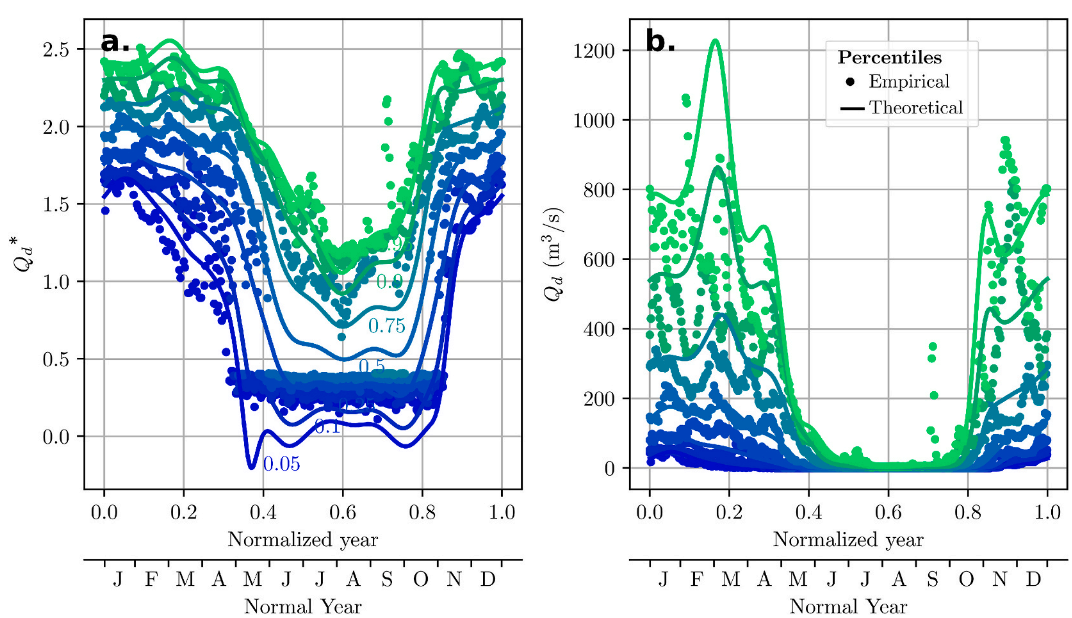

Fig. 2. Non-Stationary Cumulative Distribution Function for (a) normalized fresh-water river discharge using Box-Cox method ("transform":{"plot": True}), and (b) fresh-water river discharge ("transform":{"plot": False}). Dots represent the empirical NS-CDF from observations. Lines stand for the theoretical NS-CDF. Two horizontal axes are represented showing the normalized and natural year.

```
params["TD"]: {
        "vars": ["U", "DirU"],
        "order": 72,
    }
```

The following code is required to load the json file with the parameters from the preceding marginal fits:

```
from marinetools.auxiliar import read
params["U"] = read.rjson("filename_U")
params["DirU"] = read.rjson("filename_DirU")
```

Given the dictionaries with the results from the marginal fit and the properties of the temporal and multivariate analysis, the VAR model is applied by coding:

```
df_dt = analysis.dependencies(data, params)
```

The results are saved to a file in a new folder called “dependency”.

Among all the orders analyzed, the one with the minimum BIC was $q = 6$. As it is observed in Fig. 4, the pattern of the joint density function of hindcast data U and θ_U (panel a) is well reproduced by one of the random simulations (panel b). The simulation slightly reduces the probability of the modal bump (yellow pixels at 275° and 0.4 m/s approx.).

3.3. Climate projections using ensemble average multi-model NS distributions at the Alborán Sea

Finally, a multivariate analysis for wave and wind climate projections at the Alborán Sea (3.608° W - 36.66° N) is presented. From that purpose, we used hourly data from the RCP 8.5 scenario and projections of the following Global and Regional Climate Models (GCM-RCM) CCLM4-8-17-MIROC5, RCA4-CNRM-CM5, RCA4-EC-EARTH, RCA4-HadGEM2-ES, RCA4-IPSL-CM5A-MR and RCA4-MPI-ESM-LR (IH, 2019). The characteristics of these GCM-RCM are described in Pérez et al. (2017). It comprises the significant wave height (H_s), the peak wave period (T_p), the mean incident wave direction (θ_m), the wind velocity (V_w), and the mean wind incoming direction (θ_w) during an interval that spans from 2025/02/01 to 2046/01/01 (Source: IH Cantabria). The ensemble mean properties were computed using equal weights, a choice that represents the rule of ‘one model-one-vote’ recommended by the IPCC (Pörtner et al., 2019).

The options selected for the analysis and the simulation of five realizations of 20 years are shown next:

```
params["TD"]= {"vars": ["Hs", "Tp", "DirM", "Vv", "Dmv"]
    "order": 72}

params["EA"]:  {"make": True,
     "models": ['RCA4-MPI-ESM-LR',
                'RCA4-IPSL-CM5A-MR',
                'RCA4-HadGEM2-ES',
                'RCA4-CNRM-CM5',
                'CCLM4-8-17-MIROC5',
                'CCLM4-8-17-CanESM2',
                'RCA4-EC-EARTH'],
     "weights": "equal"}

params["TS"]:  {"start": "2026/02/01 00:00:00",
    "end":   "2046/01/01 00:00:00",
    "nosim": 5,
    "ensemble": True
    }
```

With the previous dictionary (`params`) of the average multi-model GCM-RCM and importing the simulation package with

```
from marinetools.temporal import simulation
```

the simulations are achieved by coding

```
simulation.simulation(params).
```

The simulated time series are saved into files in a new folder called “simulations”.

For each GCM-RCM, (i) the variables were fitted using combinations of PMs as Fig. 5 shows which properties are summarized in Table 4, (ii) the temporal dependency was computed for q-orders that range from 26

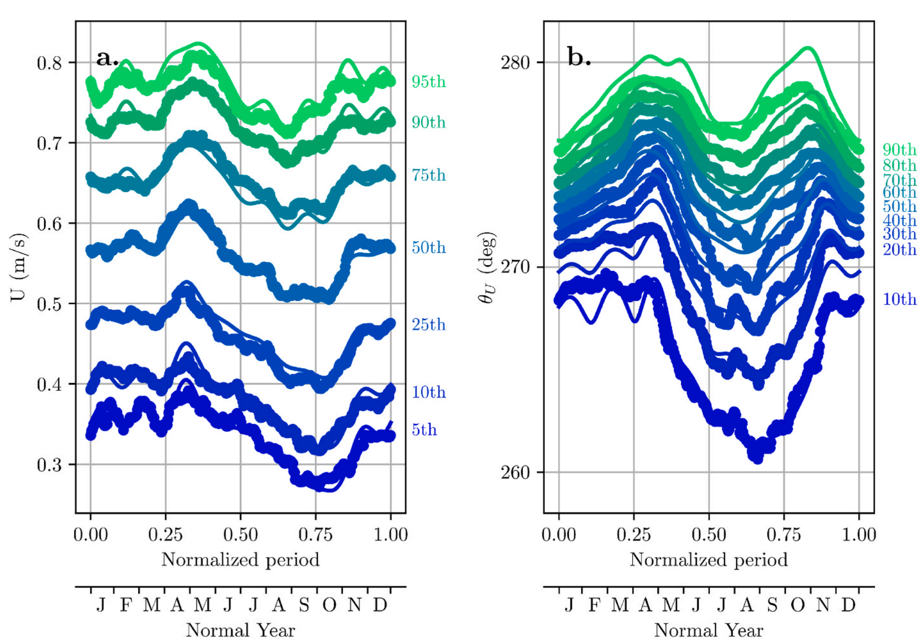

Fig. 3. Non-Stationary Cumulative Distribution Function for (a) water currents magnitude, and (b) water current mean incoming direction. Dots represent the empirical NS-CDF from observations. Lines stand for the theoretical NS-CDF. Two horizontal axes are represented showing the normalized and natural year.

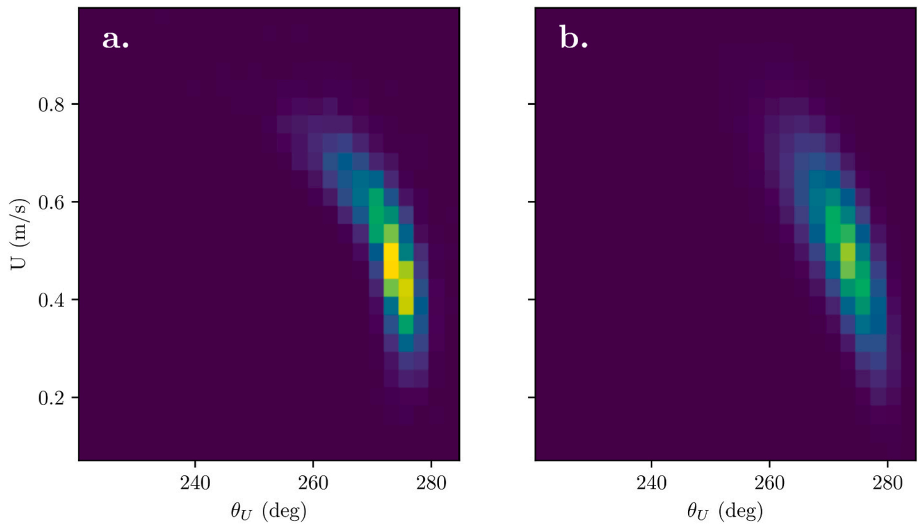

Fig. 4. Comparison between the joint probability distribution of U and θ_U for *a)* hindcast data and, *b)* one random simulation.

to 38 h. The one corresponding to 26 h had the lowest BIC, which means that H_s-T_p-θ_m-V_v-θ_w are strongly related with the previous 26 sea-states. This value is consistent with the one obtained by Lira-Loarca et al. (2021) using hindcast data and projections from the MeteOcean group of the University of Genoa.

In the following, the probabilistic characterization of one of the 20 years simulations and the obtained ensemble mean multi-model are compared. Panels a and b of Fig. 6 show the joint CDFs for the simulated and ensemble averaged projections for H_s - T_p, and H_s - V_v, respectively. It can be observed that the degree of agreement between the simulation and the hindcasted data is generally good, which demonstrate that the simulation maintains the joint dependency. Moreover, the R^2 coefficients between data of the joint probability density functions (PDF) of H_s-T_p, H_s-θ_m, H_s-V_v and, V_v-θ_w of the simulation and the multi-model ensemble were for all cases greater than 0.97.

Fig. 7 shows the wind roses at Alborán sea for ensemble mean data projections and one random simulation. The tool enables the creation of several subplots and the chance to include into them any plot function available in the graphics package Matplotlib.

A proper way to verify the temporal dependency of the simulation and the input data is by computing the autocorrelation function (Papoulis and Saunders, 1989). Fig. 8 depicts the autocorrelation of H_s, T_p, θ_m, V_v and θ_v for the simulation (solid lines). The shadow areas delimit the values obtained for the autocorrelation functions for all the RCMs. As it is observed, the simulation and observations show a similar variability with values ranging between 0.7 and 1. The wind field shows a smaller autocorrelation than the RCMs during 5 - ≈ 20 h lag. Also, the autocorrelation of θ_m is underestimated for all the lags. This difficulty of reproducing directional variables, already pointed out by (Monbet et al., 2007), is clearly seen in this dataset that correspond to the Gulf of Cádiz,

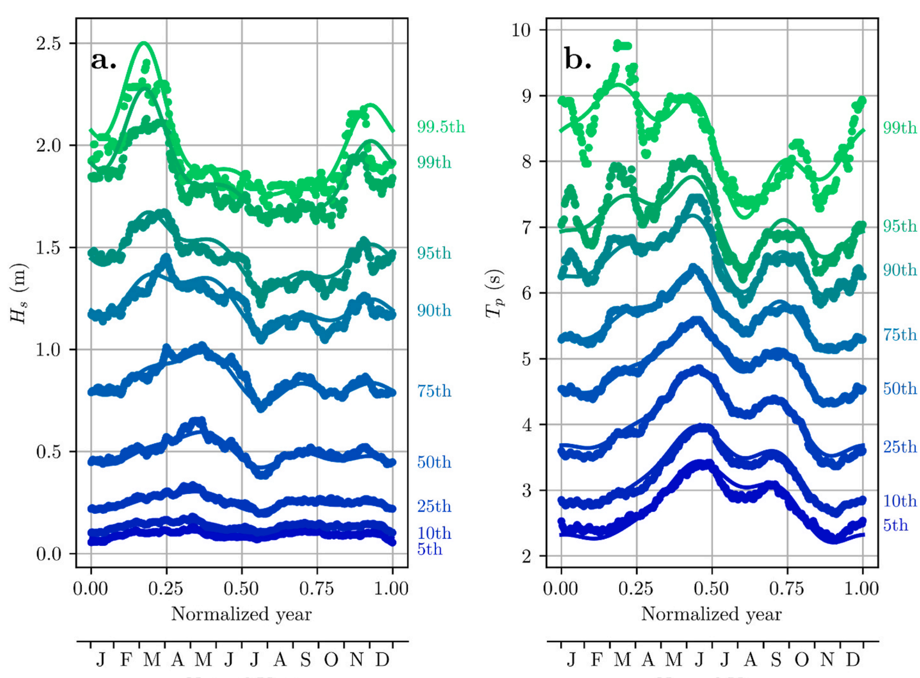

Fig. 5. Non-Stationary Cumulative Distribution Function for (a) significant wave height for RCA4-IPSL-CM5A-MR RCM, and (b) peak period for RCA4-HadGEM2-ES GCM-RCM. Dots represent the empirical NS-CDF from observations. Lines stand for the theoretical NS-CDF. Two horizontal axes are represented showing the normalized and natural year.

a place where abrupt wind changes are experienced. It should be highlighted, however, that the difference is lower than 1%, showing autocorrelation values larger than 0.91 in all cases. The results point out that the multivariate and the temporal dependency is also well reproduced.

4. Discussion

The goal of *MarineTools.temporal* is the search for the optimum values for any combination of PMs and several basis functions. Actually, most of the computational time is spent on the estimation of the optimal parameters of each marginal NS distribution. Therefore, while it is not critic, an appropriate selection of the PMs will certainly reduce the computational time. A suitable combination of PMs will usually ensure the convergence of the optimization process. However, in some cases, a slight modification of the weights or percentiles of the common endpoints should be required. The tool incorporates several sanity checks to provide a user-friendly experience working with NS-PMs.

The choice of the basis function is also relevant for the optimization, as some basis functions give a better fit with a considerable smaller amount of terms. This is the case of the sinusoidal functions that usually requires about the half of terms in the series than the Trigonometric of the Modified Fourier expansion.

The input time series for the marginal fits can have small gaps since to the search for the optimum value of the NLLF does not require to have a regularly observed time series. However, for the temporal dependence analysis, concomitant time series observed with a fixed time step are required.

The model has been coded to deal with any basis period. The examples included for the analysis are done with one year and multiples of it because at temperate latitudes the yearly periodicity is predominant in climate data. When long time series are available (as it is the case for the Sunspot number), many years can be considered to capture longer time scale variations. It is worth noting that when the basis period is larger than 1 year, the initial date and time of the simulations have to be selected carefully in order to avoid an artificial shift of the oscillatory variability.

MarineTools.temporal has just included one of the temporal dependency models AR and VAR. However, there are many other possibilities in (PyFlux for Python) to characterize these relationships such as ARMA, ARIMAX, DAR, EGARCH, GAS, Gaussian local level. For independent variables, the simulations will show no correlation in multivariate analysis. So, it is recommended in those cases an independently simulation of each variable.

The framework *MarineTools.temporal* gives a step forward in the need of bridge the gap between the current software development and the coastal and marine engineering practice (Magaña et al., 2020), including advances not only in multivariate analysis which approach to marine design is not now only available for academic use as Jonathan and Ewans (2013) indicated.

The tool has been applied to the examples given in section 3 and Table 3 which include timeseries coming from coastal engineering (sea wave height, peak period and mean direction), oceanography (sea surface temperature and currents at open sea) and hydrology (river discharge) as proof of evidence that continuous RVPs can be adequately analyzed and simulated. The tool was also applied to analyze and simulate raining patterns (results not shown).

Additionally, the use of new simulations generated within the MarineTools framework might reveal behaviours that emerge from the intrinsic nature of the vector random process analyzed and derived processes, which cannot be previously capture and analyze by using one realization, which is paramount for structure design and environmental planning.

5. Conclusions

MarineTools is an open-source project hosted on GitHub. The temporal package is dedicated to the analysis of stationary and NS RPs and the simulation of Earth and environmental data with the same probabilistic behaviour. All scripts discussed in the present paper and the synthetic data files are in the repository. Further examples, given as

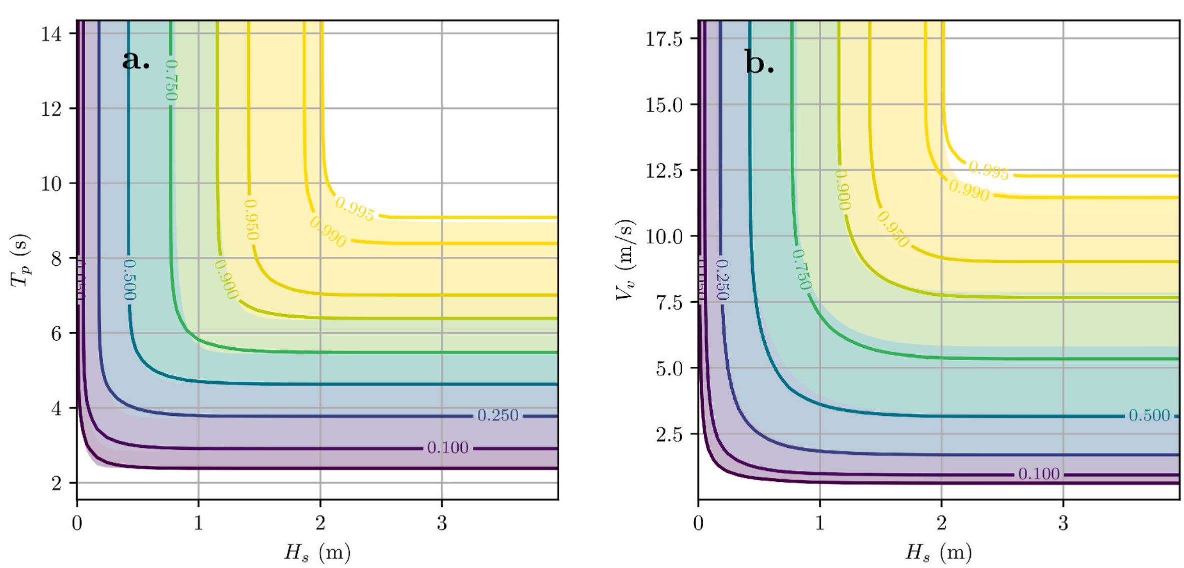

Fig. 6. Comparison between joint cumulative distribution functions of: (a) H_s and T_p, and (b) H_s and θ_m. Contour lines represent the iso-probability lines for one random simulation while the filled areas at the background showed the iso-probability areas for the ensemble data from GCM-RCMs.

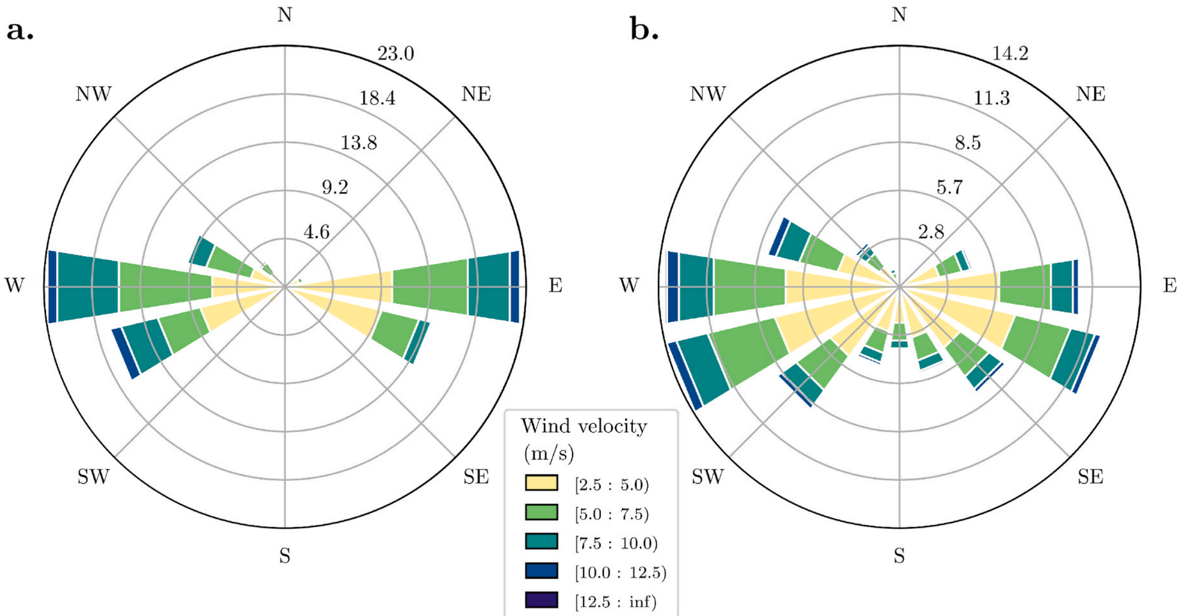

Fig. 7. Wind roses at the Alboran sea for (a) ensemble mean data from projections, and (b) one random simulation.

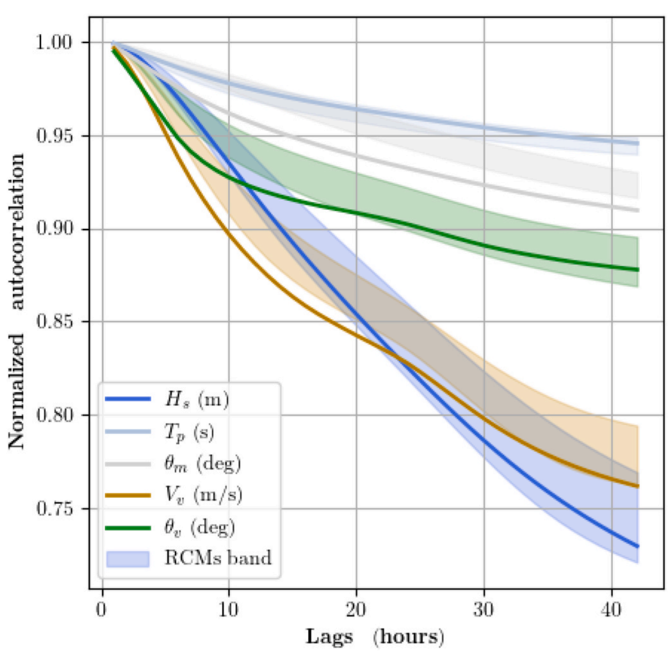

Fig. 8. Positive part of the autocorrelation for the simulation (solid lines) compared to the band between the minimum and maximum auto-correlation of the RCMs for the sea and wind fields at the Alborán Sea.

Jupyter notebooks, covering the full use of *MarineTools.temporal* are also available on GitHub.

The present paper shows the options included in *MarineTools.temporal* together with the examples: marginal fit of freshwater river discharge from a dam, multivariate and temporal dependency of water currents at the ocean, and multivariate simulations of sea and wind climate states. It is demonstrated how the optimization of the NLLF for NS-PMs succeeds for several combinations of the PMs included in *scipy.stats* and the wrap normal. Three main scripts which include the pre-processing steps, marginal fit, multivariate and temporal analysis, ensemble average multi-model GCM-RCM projections and, finally, the simulation of Earth and environmental time series are presented in detail. The use of *MarineTools.temporal* offers a very general framework which can be successfully applied to a wide variety of environmental and engineering problems.

Software availability

- Name of software: Marinetools
- Developer: Environmental Fluid Dynamics Group (University of Granada)
- Contact information: mcobosb@ugr.es
- Year first available: 2021
- Software required: specified at the GitHub repository
- Program language: Python
- Program size: 58.6 KB
- Cost: Open-source tools released under the GNU General Public License v3.0
- Repository:

CRediT authors contribution statement

M. Cobos[1]: Conceptualization, Methodology, Software, Writing - Original draft preparation. **P. Otiñar**[1]: Writing draft preparation. **P. Magaña**[1]: Data curation, Software. **A. Lira-Loarca**[2]: Data curation, Software, Methodology. **A. Baquerizo**[1]: Conceptualization, Methodology, Writing - Original draft preparation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was performed within the framework of the following projects: (1) AQUACLEW, which is part of ERA4CS, an ERA-NET initiative by JPI Climate, and funded by FORMAS (SE), DLR (DE), BMWFW (AT), IFD (DK), MINECO (ES), ANR (FR) with co-funding by the European Commission [Grant 690462] and (2) Flooding and erosion works in coastal areas of Andalusia under a climate change scenario, funded by the Ministry of Agriculture, Livestock, Fisheries and Sustainable Development of the Junta de Andalucía [Contrat No. CONTR 2018 66984]. Part of this study has been conducted using E.U. Copernicus Marine Service Information. Funding for open access charge: Universidad de Granada / CBUA.

Appendix A

Some useful information about the requirements, installation and computing times can be found in the present appendix.

The model was developed in a virtual environment which packages can be installed from a requirements file (GitHub). The working area can be easily and quickly raising up just using those packages.

The computational time mainly depends on the number of parameters to be optimized which at the same time depends on the number of GFS or polynomial time expansion and the number of PMs. So, more than three PMs are not recommended due to the computational time increases exponentially.

Table 4
Description of models and parameters for the analysis of the present paper. The computational time was obtained with a user personal computer (Intel(R) Core(TM) i7-9700F CPU @ 3.00 GHz and 8 cores). In those cases, the weights represented a threshold of the transition between PMs and not the weight of the PM. The optimization method more frequently used by authors is SLSQP due to the high efficiency, however, in some cases a high-cost computational method will be required for ensuring convergency. [a] means that a reduction of the number of parameters involved is applied following Solari and Losada (2011). Column names are described in section 2.

Var	Section	circular	transform	GFS	noterms	ps_ws	fun	Computational time (s)
Q_d	3.1	False	Yes	Sinusoidal	20	–	Weibull of maxima	11.75
U	3.2	False	No	Trigonometric	8	–	Norm	21.54
θ_U	3.2	True	No	Trigonometric	8	–	Weibull of maxima	14.25
H_s	3.3	False	No	Trigonometric	4	0.05 - 0.85	Generalized Pareto + Lognorm + Generalized Pareto[a]	1854.24
T_p	3.3	False	No	Trigonometric	4	0.05 - 0.85	Generalized Pareto + Norm + Generalized Pareto[a]	1766.18
θ_m	3.3	True	No	Trigonometric	4	0.49/0.61 - 0.39/0.51	Truncated Norm + Truncated Norm	652.87
V_v	3.3	False	No	Trigonometric	4	0.05 - 0.85	Generalized Pareto + Gamma + Generalized Pareto[a]	2759.81
θ_v	3.3	True	No	Trigonometric	4	0.42/0.48–0.52/0.58	Truncated Norm + Truncated Norm	714.97

The use of *MarineTools.temporal* requires a basic understanding of the Python programming language, as well as of functional code design. In particular, guidelines for an easy installation of the required packages of *MarineTools.temporal* will be found in GitHub repository. Some information about the properties of PMs and methods using in the examples of section 3 are described in Table 4.

Appendix B

In this section, the main formulas of the methodology are given.

B.1. Marginal distribution

The marginal distribution may have any of the following structures:

B.1.1. Piecewise distributions

The random variable X probability density function $f_X(x)$ is expressed as a piecewise function defined over a partition of the real axes into N_y subintervals: $(u_{\alpha-1}, u_\alpha]$ for $\alpha = 2, \ldots, N_I - 1$, $I_1 = (-\infty, u_1]$ and $I_{N_I} = (u_{N_I-1}, +\infty)$:

$$f_X(x) = \begin{cases} k_1 f_1(x) & x \leq u_1 \\ k_2 f_2(x) & u_1 < x \leq u_2 \\ \ldots & \ldots \\ k_\alpha f_\alpha(x) & u_{\alpha-1} < x \leq u_\alpha \\ \ldots & \ldots \\ k_{N_I} f_{N_I}(x) & u_{N_I-1} \leq x \end{cases} \quad (1)$$

In eq. 1 f_α denotes the probability density function (PDF) of the model selected for I_α. The values of the coefficients are:

$$k_\alpha = \frac{a_1}{b_1} \cdots \frac{a_{\alpha-1}}{b_{\alpha-1}} \left[c_1 + \sum_{\alpha=2}^{N_I-1} c_\alpha \frac{a_1}{b_1} \frac{a_2}{b_2} \cdots \frac{a_{\alpha-1}}{b_{\alpha-1}} \right]^{-1} \quad (2)$$

where $a_\alpha = f_\alpha(u_\alpha)$, $b_\alpha = f_{\alpha+1}(u_\alpha)$ and $c_\alpha = F_\alpha(u_\alpha) - F_{\alpha-1}(u_{\alpha-1})$, provided that b_α and the denominator in eq. (2) are both different from zero.

The corresponding dictionary must include the names to identify the N_I PMs selected (key: "fun") and the initial guesses of the values of the percentiles of the common endpoints in key "ws_ps", that is of, $p_\alpha = F^{-1}(u_\alpha)$ for $\alpha = 1, \ldots, N_I - 1$.

B.1.2. Compound distribution

The PDF of the random variable X is expressed as a weighted sum of N_I different PDFs:

$$f_X(x) = \sum_{\alpha=1}^{N_F} \omega_\alpha f_\alpha(x) \quad (3)$$

where $w_1 + \cdots + w_\alpha + \cdots + w_I = 1$. The dictionary of X must include the names to identify the N_I PMs selected (key: "fun") and the initial guesses of values of the percentiles of the $N_I - 1$ common endpoints in key "ws_ps", that is, ω_α for $\alpha = 1, \ldots, N_I - 1$.

B.2. Non-stationarity

If the distribution of X is non stationary, the parameters of the PMs models given in key "fun" can be approximated over the interval $[0, N_y]$ by a truncated generalized Fourier Series. If $a(t)$ is any of those parameters, its expression is given by:

$$a(t) = \sum_{n=1}^{N_I} a_n \varphi_n \; t \in [0, N_y], \tag{4}$$

Where a_n denotes any of the parameters and $\{\varphi_n(t)\}_n$ is the set of basis functions.

B.3. Temporal dependence

The Vector Auto-regressive, VAR(q) model is applied to the following normalized time series:

$$Z_{X_i}(t_j) = \Phi^{-1}\left[F_{X_i}\left(x_i^o(t_j); t_j\right)\right], \tag{5}$$

where Φ^{-1} is the inverse of the Gaussian cumulative distribution function with zero mean and unit standard deviation and $F_{X_i}(x_i(t); t)$ is the NS probability distribution function of X_i.

Denoting the values of the normalized series (eq. 5) at time t_j as $y_j^i = Z_{X_i}(t_j)$ and $Y_j = \left(y_j^1, \ldots, y_j^i, \ldots, y_j^N\right)^T$ where T stands for the vector transposition, the dependence in time between variables in the VAR(q) model is given by:

$$Y_j = c + A^1 Y_{j-1} + A^2 Y_{j-2} + \ldots. + A^q Y_{j-q} + e_j, \tag{6}$$

where $c = (c_1, \ldots, c^i, \ldots, c_N)^T$ contains the mean values of the variables, A^m, $m = 1, \ldots, q$ are the $N \times N$ coefficients matrices and $e_j = \left(e_j^1, \ldots, e_j^i, \ldots, e_j^N\right)^T$ is the vector with the white noise error terms. Using eq. (6) to relate data at an instant t_j to their previous q values, for $j = q+1, \ldots, N_o$, we obtain $Y = A\chi + E$, where $Y = (Y_{q+1} Y_{q+2} \ldots Y_N)$, $\chi = (\chi_{q+1} \chi_{q+2} \ldots \chi_N)$, with $\chi_j = \left(1 Y_{j-1}^T \ldots Y_{j-q}^T\right)^T$, $A = (A^1 A^2 \ldots A^q)$ and $E = (e_{q+1} e_{q+2} \ldots e_N)$.

The solution is obtained by means of minimum least square errors as $A = Y\chi^T(\chi\chi^T)^{-1}$, where $E = Y - A\chi$ and $Q = cov(E)$ is the covariance matrix of the error. A detailed description can be found e.g., in Lütkepohl (2005).

References

Baquerizo, A., Losada, M.A., 2008. Human interaction with large scale coastal morphological evolution. an assessment of the uncertainty. Coast. Eng. 55, 569–580.

Box, G.E., Cox, D.R., 1964. An analysis of transformations. J. Roy. Stat. Soc. B 26, 211–243.

Cobos, M., 2020. A Model to Study the Consequences of Human Actions in the Guadalquivir River Estuary. Tesis Univ. Granada. URL: http://hdl.handle.net/10481/65374.

Cobos, M., Otíñar, P., Magaña, P., Baquerizo, A., 2022. A Method to Characterize and Simulate Climate, Earth or Environmental Vector Random Processes. Submitted to Stochastic Environmental Research and Risk Assessment.

Egüen, M., Aguilar, C., Solari, S., Losada, M., 2016. Non-stationary rainfall and natural flows modeling at the watershed scale. J. Hydrol. 538, 767–782.

Ellis, C., Visser-Quinn, A., Aitken, G., Beevers, L., 2021. Quantifying uncertainty in the modelling process; future extreme flood event projections across the UK. Geosciences 11, 33.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

Hunter, J.D., 2007. Matplotlib: a 2d graphics environment. Comput. Sci. Eng. 9, 90–95.

IH, 2019. Elaboración de la metodología y bases de datos para la proyección de impactos de cambio climático a lo largo de la costa española.

Jalón, M.L., Baquerizo, A., Losada, M.A., 2016. Optimization at different time scales for the design and management of an oscillating water column system. Energy 95, 110–123.

Jonathan, P., Ewans, K., 2013. Statistical modelling of extreme ocean environments for marine design: a review. Ocean Eng. 62, 91–109.

Kundzewicz, Z.W., Krysanova, V., Benestad, R., Hov, Ø., Piniewski, M., Otto, I.M., 2018. Uncertainty in climate change impacts on water resources. Environ. Sci. Pol. 79, 1–8.

Le Mouël, J.L., Lopes, F., Courtillot, V., 2019. A solar signature in many climate indices. J. Geophys. Res. Atmos. 124, 2600–2619. https://doi.org/10.1029/2018JD028939.

Lira-Loarca, A., Cobos, M., Losada, M.Á., Baquerizo, A., 2020. Storm characterization and simulation for damage evolution models of maritime structures. Coast. Eng. 156, 103620.

Lira-Loarca, A.L., Cobos, M., Besio, G., Baquerizo, A., 2021. Projected wave climate temporal variability due to climate change. Stoch. Environ. Res. Risk Assess. 1–17.

López-Ruiz, A., Bergillos, R.J., Lira-Loarca, A., Ortega-Sánchez, M., 2018. A methodology for the long-term simulation and uncertainty analysis of the operational lifetime performance of wave energy converter arrays. Energy 153, 126–135.

Lütkepohl, H., 2005. New Introduction to Multiple Time Series Analysis. Springer Science & Business Media.

Magaña, P., Del-Rosal-Salido, J., Cobos, M., Lira-Loarca, A., Ortega-Sánchez, M., 2020. Approaching software engineering for marine sciences: a single development process for multiple end-user applications. J. Mar. Sci. Eng. 8, 350.

Mardia, K.V., Jupp, P.E., 2009. Directional Statistics, vol. 494. John Wiley & Sons.

McKinney, W., et al., 2010. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference, Austin, TX, pp. 51–56.

Meier, H., Edman, M., Eilola, K., Placke, M., Neumann, T., Andersson, H.C., Brunnabend, S.E., Dieterich, C., Frauen, C., Friedland, R., et al., 2019. Assessment of uncertainties in scenario simulations of biogeochemical cycles in the baltic sea. Front. Mar. Sci. 6, 46.

Moges, E., Demissie, Y., Larsen, L., Yassin, F., 2021. Sources of hydrological model uncertainties and advances in their analysis. Water 13, 28.

Monbet, V., Ailliot, P., Prevosto, M., 2007. Survey of stochastic models for wind and sea state time series. Probabilist. Eng. Mech. 22, 113–126.

Navarro, G., Ruiz, J., Cobos, M., Baquerizo, A., Díez-Minguito, M., Ortega-Sanchez, M., Losada, M., 2019. Atmospheric, Hydrodynamic and Water Quality Observations from Environmental-Quality Stations, Water Level Sensors, Acoustic Doppler Velocimeters, and Meteorological Stations Located at the Guadalquivir River Estuary (2008 - 2010). https://doi.org/10.5281/zenodo.3459610.

Papoulis, A., Saunders, H., 1989. Probability, Random Variables and Stochastic Processes.

Pewsey, A., 2000. The wrapped skew-normal distribution on the circle. Commun. Stat. Theor. Methods 29, 2459–2472.

Pörtner, H.O., Roberts, D.C., Masson-Delmotte, V., Zhai, P., Tignor, M., Poloczanska, E., Mintenbeck, K., Nicolai, M., Okem, A., Petzold, J., et al., 2019. Ipcc Special Report on the Ocean and Cryosphere in a Changing Climate, vol. 1. IPCC Intergovernmental Panel on Climate Change, Geneva, Switzerland.

Pérez, J., Menéndez, M., Losada, I., 2017. Gow2: a global wave hindcast for coastal applications. Coast. Eng. 124, 1–11.

Refsgaard, J.C., van der Sluijs, J.P., Højberg, A.L., Vanrolleghem, P.A., 2007. Uncertainty in the environmental modelling process–a framework and guidance. Environ. Model. Software 22, 1543–1556.
Solari, S., Losada, M., 2011. Non-stationary wave height climate modeling and simulation. J. Geophys. Res.: Oceans 116.
Solari, S., Van Gelder, P., 2011. On the use of vector autoregressive (var) and regime switching var models for the simulation of sea and wind state parameters. Mar. Technol. Eng. 1, 217–230.
Sverdrup, H. U., J.M.W., Fleming, R.H., 1942. Prentice-Hall.
Uhe, P., Mitchell, D., Bates, P.D., Allen, M.R., Betts, R.A., Huntingford, C., King, A.D., Sanderson, B.M., Shiogama, H., 2021. Method uncertainty is essential for reliable confidence statements of precipitation projections. J. Clim. 34, 1227–1240.
Usoskin, I., Mursula, K., 2003. Long-term solar cycle evolution: review of recent developments. Sol. Phys. 218, 319–343.
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., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2.
Yeo, I.K., Johnson, R.A., 2000. A new family of power transformations to improve normality or symmetry. Biometrika 87, 954–959.