

# Timescape: A Novel Spatiotemporal Modeling Tool

Marco Ciolfi <sup>\*,†</sup> , Francesca Chiocchini <sup>†</sup> , Rocco Pace , Giuseppe Russo and Marco Lauteri 

Institute of Research on Terrestrial Ecosystems of the National Italian Research Council (CNR-IRET),  
Via Marconi 2, 05010 Porano, Italy; francesca.chiocchini@cnr.it (F.C.); rocco.pace@iret.cnr.it (R.P.);  
giuseppe.russo@iret.cnr.it (G.R.); marco.lauteri@cnr.it (M.L.)

\* Correspondence: marco.ciolfi@cnr.it

† These authors contributed equally to this work.

**Abstract:** We developed a novel approach in the field of spatiotemporal modeling, based on the spatialisation of time, the Timescape algorithm. It is especially aimed at sparsely distributed datasets in ecological research, whose spatial and temporal variability is strongly entangled. The algorithm is based on the definition of a spatiotemporal distance that incorporates a causality constraint and that is capable of accommodating the seasonal behavior of the modeled variable as well. The actual modeling is conducted exploiting any established spatial interpolation technique, substituting the ordinary spatial distance with our Timescape distance, thus sorting, from the same input set of observations, those causally related to each estimated value at a given site and time. The notion of causality is expressed topologically and it has to be tuned for each particular case. The Timescape algorithm originates from the field of stable isotopes spatial modeling (isoscapes), but in principle it can be used to model any real scalar random field distribution.

**Keywords:** spatiotemporal modeling; ecological modeling; sparse data; minkowskian geometry; time series analysis; spatial statistics; isoscapes

**MSC:** 92-04; 92-08; 92D40; 92F05



**Citation:** Ciolfi, M.; Chiocchini, F.;

Pace, R.; Russo, G.; Lauteri, M.

Timescape: A Novel Spatiotemporal  
Modeling Tool. *Earth* **2022**, *3*, 259–286.

[https://doi.org/10.3390/  
earth3010017](https://doi.org/10.3390/earth3010017)

Academic Editor: Steven R.  
Fassnacht

Received: 23 December 2021

Accepted: 15 February 2022

Published: 17 February 2022

**Publisher's Note:** MDPI stays neutral  
with regard to jurisdictional claims in  
published maps and institutional affil-  
iations.



**Copyright:** © 2022 by the authors.  
Licensee MDPI, Basel, Switzerland.  
This article is an open access article  
distributed under the terms and  
conditions of the Creative Commons  
Attribution (CC BY) license ([https://  
creativecommons.org/licenses/by/  
4.0/](https://creativecommons.org/licenses/by/4.0/)).

## 1. Introduction

One of the major issues in ecological modeling is the associated occurrence of spatial and temporal variability of the observations underlying the models [1]. Often one has to merge old-fashioned and modern measurements in a coherent mixture, especially when following the evolution of some phenomenon over time. Concerning both the purely spatial and purely temporal dependence of the variables, we have a rich modeling toolbox at hand, ranging from time series analysis [2,3] to spatial statistics [4–6]. However, when one cannot decide whether to favour spatial or temporal variability, the toolbox offers fewer options, commanding greater statistical intricacy [7–10], see for example the moving average processes [11] or the covariance-based methods [12] implemented in R gstat package [13,14]. In some cases we could have a dynamical model at hand [15], so the modeling is generally conducted via partial differential equations integration, as is often the case with population dynamics [16], stochastic differential equations, and Gaussian random fields [17,18]. Other trending approaches include neural networks and deep learning techniques [19,20] and space–time Bayesian modeling [21]. Generalized linear models [22] and regression trees [23] are also used in spatial and temporal ecological modeling. Furthermore, one is often confronted with the problem of model scaling [24], though the computational limitations are less and less important.

We chose to treat our observations as a random scalar field through standard geostatistical techniques, without introducing an explicit dynamic model or any kind of machine learning. Spatial statistics demands that the distribution of our observations obeys more or less strict constraints, depending on the kind of modeling we would like to perform [4,25]. On the other hand, the distribution of observations in ecology is often sparse, both in

time and space, especially when merging new data, collected according to a robust spatial pattern, with old observations, typically collected just at spot sites or along transects: data non-stationary and a spatially skewed distribution of the observations is often encountered [26], so the modeling should be conducted, as a matter of principle, also with sub-optimal datasets, maybe at the expense of accuracy.

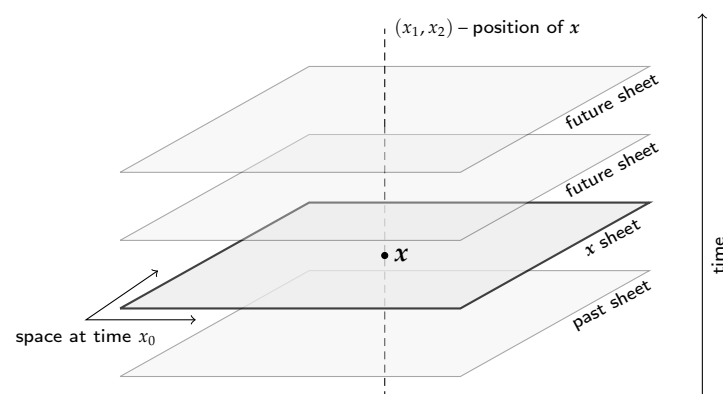
In order to deal with entangled spatial and temporal variability, we propose a novel spatiotemporal interpolation technique—the Timescape algorithm—that estimates the spatiotemporal distribution of a given variable (technically, a real random field) from a set of observations. The key idea is borrowed from the machinery of relativistic physics [27,28], which incorporates a notion of causality rooted into its very topological structure, the Minkowski spacetime [29]. Time is treated as new spatial dimension [30], via the multiplication by a constant with the dimensions of a velocity, and we also forbid the instantaneous propagation in space of the field values by filtering the possible sources when estimating the random field.

In a nutshell, the Timescape algorithm works at a topological level, introducing a spatiotemporal distance that mimics the Minkowskian structure, then exploiting any ordinary spatial interpolation tool for actually estimating the field values. Our spatiotemporal distance can be tuned by a couple of parameters, the aforementioned velocity and a causal strictness parameter, plus a form factor related to seasonality. The Timescape distance is exposed in Sections 2.1 and 2.4 and the actual implementation is discussed at length in [31].

On practical grounds, we sought for a modeling tool that is: (1) not too choosy about the distribution of the input data, (2) capable of accommodating seasonal variability, (3) suitable for both both projected and geographical coordinates, (4) capable of running on affordable hardware in decent times, (5) distributable as open source. The resulting software implementations are a Python module [31] and two Java applications, one for projected [32] and one for geographical coordinates [33]. The software is exposed in Section 2.5. A few case studies are discussed in Section 3.

## 2. Methods

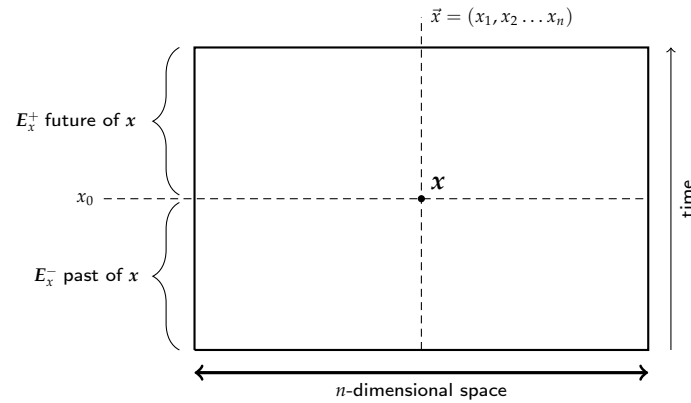
A Timescape model is a discrete representation of a scalar field on a space–time continuum with a temporal dimension and any number of spatial dimensions (Figure 1), i.e., a lattice of spatiotemporal cells (here and henceforth the target) whose values are interpolated from a sparse set of observed field values (the source). A Timescape can be regarded as a stack of sheets, each of which is an ordinary discrete spatial model, indexed by the time, as well as a sheaf of time series, identified by their spatial location.



**Figure 1.** The whole space–time centered about the event  $x$  of coordinates  $(x_0, x_1, x_2)$ :  $x$  is located at  $(x_1, x_2)$  in a bi-dimensional space at time  $x_0$ :  $x = (x_0, x_1, x_2)$ .

To define a spatio-temporal distance, we mimic the structure of the Minkowskian relativistic metric [27,28,34]. According to the conventions of relativistic physics, we call event any space–time element  $x = (x_0, x_1, x_2 \dots x_n)$ , where  $x_0$  is the time and  $x_1, x_2 \dots x_n$  are the spatial coordinates.

In the following, we will represent the time and only a single spatial component, regardless of the actual number of dimensions (generally two, in both geographical and whatever projected coordinates)—see Figure 2.



**Figure 2.** A limited portion of the space–time centered about the event  $x = (x_0, x_1, x_2 \dots x_n)$ .

The Minkowskian space–time incorporates a notion of causality due to its topology [29]. In order to implement this geometric realization of causality, we must impose an equivalent causal structure on our space–time.

2.1. Space–Time Distances

Each event  $x$  can be split in its time component  $x_0$  and space component  $\vec{x} = (x_1, x_2 \dots x_n)$ , i.e.,  $x = (x_0, \vec{x})$ . Now define a spatial distance  $D_s(x, y)$  between the events  $x$  and  $y$  using any given ordinary metric function  $d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}^+$ , where  $\mathcal{D}$  is the spatial domain of interest (a subset of  $\mathbb{R}^n$  or any suitable manifold, e.g., the sphere  $S^2$  of geographical coordinates). A metric  $d(\cdot)$  must satisfy the the following conditions: identity:  $d(\vec{x}, \vec{y}) = 0 \iff \vec{x} \equiv \vec{y}$ , non-negativity:  $d(\vec{x}, \vec{y}) \geq 0$ , symmetry:  $d(\vec{x}, \vec{y}) = d(\vec{y}, \vec{x})$  and subadditivity:  $d(\vec{x}, \vec{y}) \leq d(\vec{x}, \vec{z}) + d(\vec{z}, \vec{y}) \quad \forall \vec{z} \in \mathcal{D}$ , also known as triangle inequality. The latter, in particular, gives the metric its distinctive character of nearness measurement and it is often the key property for the convergence of the interpolation algorithms. We define the spatial distance simply as

$$D_s(x, y) = d(\vec{x}, \vec{y}) \tag{1}$$

The metrics of practical interest in ecological modeling are the ordinary Euclidean distance in  $\mathbb{R}^n$  for projected coordinates

$$D_s(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \tag{2}$$

and the geodesic distance in  $S^2$  for geographic coordinates

$$D_s(x, y) = R \arccos \left[ \sin x_2 \sin y_2 + \cos x_2 \cos y_2 \cos (x_1 - y_1) \right] \tag{3}$$

where  $R$  is the radius of the Earth,  $x_1, y_1$  the longitudes and  $x_2, y_2$  the latitudes of the events  $x$  and  $y$  [35,36].

Since the temporal coordinates live in (a subset of)  $\mathbb{R}$ , we could naïvely take  $|x_0 - y_0|$  as the temporal distance between  $x$  and  $y$ , but the dimensions of  $D_s(x, y)$  (length) and  $|x_0 - y_0|$  (duration) are not coherent. In order to integrate space and time, we introduce the first of the Timescape parameters: a time-to-space conversion factor  $c$ , i.e., a velocity  $\in \mathbb{R}^+$  associated with each Timescape model. This velocity can have or not have a physical

interpretation as a propagation/diffusion coefficient, depending on the model. It has to be tuned to an optimal value, see Section 2.4. Thus, we define the temporal distance

$$D_t(\mathbf{x}, \mathbf{y}) = c |x_0 - y_0| \tag{4}$$

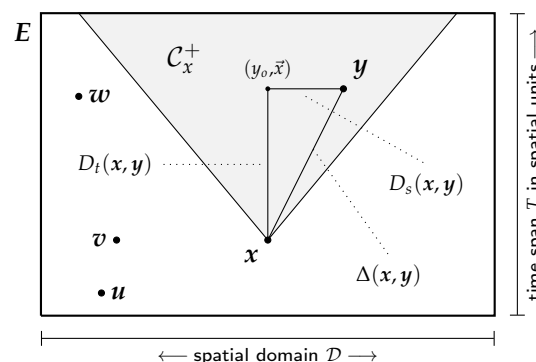
Now we combine the spatial and temporal distances via the Pythagorean theorem, thus assuring the fulfillment of identity, non-negativity, symmetry, and subadditivity, taking  $\sqrt{D_t^2(\mathbf{x}, \mathbf{y}) + D_s^2(\mathbf{x}, \mathbf{y})}$ . This function is a well-behaved metric but lacks a causal structure: the point is that the symmetry property commands no distinction between  $x$  and  $y$ , whatever  $x_0$  and  $y_0$ , so we renounce the symmetry constraint, resorting to what is technically a distance in a pseudo-metric space [37,38]. The distinctive “measure of closeness” character of a metric is given by its subadditivity so, respecting the latter constraint, we impose a directionality to our distance, restricting to the subset  $x_0 \leq y_0$ , thus introducing the asymmetric function  $D(\mathbf{x}, \mathbf{y})$  from  $x$  to  $y$ :

$$D(\mathbf{x}, \mathbf{y}) = \begin{cases} \sqrt{D_t^2(\mathbf{x}, \mathbf{y}) + D_s^2(\mathbf{x}, \mathbf{y})} & \text{if } x_0 \leq y_0 \\ \infty & \text{otherwise} \end{cases} \tag{5}$$

so that if  $D(\mathbf{x}, \mathbf{y})$  is finite  $D(\mathbf{y}, \mathbf{x}) = \infty$  (the reverse is not true:  $D(\mathbf{x}, \mathbf{y}) = \infty \not\Rightarrow D(\mathbf{y}, \mathbf{x})$  is finite. The distances can be both  $\infty$ , e.g., whenever  $x_0 = y_0$  and  $D_s(\mathbf{x}, \mathbf{y}) \neq 0$ ). In a context of spatial interpolation, what (5) is saying is that something happening at  $x$  might be a cause of  $y$  but  $y$  cannot influence  $x$  (i.e., no time traveling in the past). We can be more strict, requiring that  $x_0 < y_0$  is not enough for a source in  $x$  to propagate up to  $y$ , i.e., we forbid instantaneous actions-at-a-distance, introducing the second Timescape parameter, the causal parameter  $k$ , positing

$$\Delta(\mathbf{x}, \mathbf{y}) = \begin{cases} D(\mathbf{x}, \mathbf{y}) & \text{if } D_s(\mathbf{x}, \mathbf{y}) \leq k D_t(\mathbf{x}, \mathbf{y}) \\ \infty & \text{otherwise} \end{cases} \tag{6}$$

where  $k \geq 0$ . Said  $E = T \times \mathcal{D}$  the space of all the events, where  $T \subseteq \mathbb{R}$  contains all the events’ times, the  $\Delta(\mathbf{x}, \mathbf{y})$  in (6) coincides with  $D(\mathbf{x}, \mathbf{y})$  in (5) only in a conical subset whose tip is  $x$  (Figure 3). We call this subset  $\mathcal{C}_x^+$  the future causal cone of  $x$ . It is the locus of the possible outcomes of something happening at  $x$ . Formally,  $\mathcal{C}_x^+ = \{\mathbf{y} \in E \mid \Delta(\mathbf{x}, \mathbf{y}) < \infty\}$ .



**Figure 3.** A geometrical interpretation of  $\Delta$ :  $\Delta(\mathbf{x}, \mathbf{y})$  is the hypotenuse of the triangle formed by  $D_s(\mathbf{x}, \mathbf{y})$  and  $D_t(\mathbf{x}, \mathbf{y})$  if  $\mathbf{y} \in \mathcal{C}_x^+$ . For all the other events  $\notin \mathcal{C}_x^+$  the distance is infinite. For example:  $\Delta(\mathbf{x}, \mathbf{u}) = \infty$  because  $u_0 < x_0$ ,  $\Delta(\mathbf{x}, \mathbf{v}) = \infty$  because  $v_0 = x_0$  and  $\Delta(\mathbf{x}, \mathbf{w}) = \infty$  because  $D_s(\mathbf{x}, \mathbf{w}) > k D_t(\mathbf{x}, \mathbf{w})$  even if  $w_0 > x_0$ .

Exploiting the Heaviside theta function  $\theta(t) = 1$  if  $t \geq 0$  or 0 otherwise, we also posit  $\frac{\text{finite}}{0} = \infty$ , the formal definition of the Timescape distance from  $x$  to  $y$  can be rephrased (it is important to distinguish the starting event (from  $x$ ) from the ending one (to  $y$ )). We cannot

simply say “distance *between*  $x$  and  $y$ ” since  $\Delta(x, y)$  lacks symmetry; the remnant shadow of symmetry lies in the relation  $y \in C_x^-$  if  $x \in C_y^+$ .

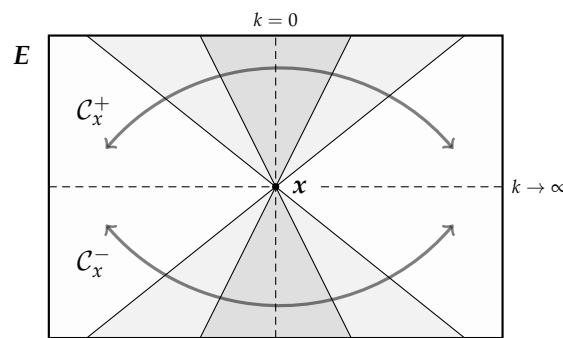
$$\Delta(x, y) = \frac{\sqrt{D_t^2(x, y) + D_s^2(x, y)}}{\theta[kD_t(x, y) - D_s(x, y)]} \tag{7}$$

Note that (7) is parameterized by  $c$  and  $k$ : each pair of parameters gives a distinct distance, and therefore a different Timescape model. In Section 2.4 we discuss in detail how to choose  $c$  and  $k$ , maximizing the accuracy of the interpolation.

2.2. Causal Structure: Topology of the Events Space

Analogously of  $C_x^+$ , we introduce the *past causal cone* of  $x$ , defined as  $C_x^- = \{y \in E \mid \Delta(y, x) < \infty\}$ , where something occurring at an event in  $C_x^-$  is a potential cause of  $x$ .

The value of  $k$  tunes the causal cones width. The bigger  $k$ , the larger the cones, the looser the causal constraint. While the smaller  $k$ , the thinner the cones, the stricter the constraint (Figure 4). As limiting cases, as  $k \rightarrow 0$ , the cones shrink to vertical lines, while as  $k \rightarrow \infty$ ,  $C_x^+$  becomes,  $E_x^+$  the whole future of  $x$  and  $C_x^-$  becomes  $E_x^-$ ;  $k \rightarrow \infty$  corresponds to an infinitely fast propagation of information in  $E$ .



**Figure 4.** Sheaves of future and past causal cones.  $x$  sits at the common tip of the cones. The cones widen as  $k$  increases. For  $k = 0$  the cones shrink to segments, and for  $k \rightarrow \infty$  the cones coincide to the whole past ( $E_x^-$ ) and future( $E_x^+$ ) of  $x$ .

The Timescape algorithm, in a nutshell, consists of topologically filtering the possible causes for each target event from a set of given source events, picking only the source events that fall into the past causal cone of the target event.

Further mathematical details can be found in the manual accompanying the Python Timescape module [31].

Accommodating Seasonal Variability

In ecological modeling, it is of paramount importance to be able to cope with periodic phenomena. We propose a refinement of the Timescape topological structure that includes seasonality by reshaping the cones by means of a multiplicative positive form factor  $\psi(t)$  that shrinks and inflates the cone width as  $\psi(t) < 1$  or  $\psi(t) > 1$ , thus modifying the time distance (4) as

$$\tilde{D}_t(x, y) = c |x_0 - y_0| \psi(|x_0 - y_0|) \tag{8}$$

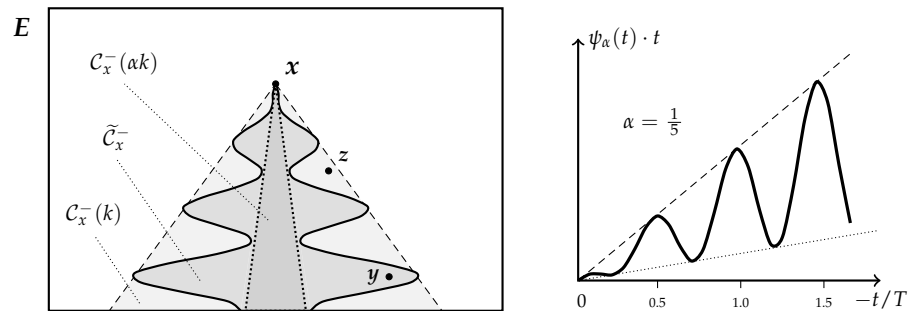
and the Timescape distance (7) accordingly:

$$\tilde{\Delta}(x, y) = \frac{\sqrt{\tilde{D}_t^2(x, y) + D_s^2(x, y)}}{\theta[k\tilde{D}_t(x, y) - D_s(x, y)]} \tag{9}$$

Figure 5 shows an example of periodic shrinking with period  $T$ , using the form factor  $\psi_\alpha$ , tuned by the parameter  $\alpha$ , defined as (the period of  $\cos^2$  is half the period of  $\cos$ , so the cosine argument’s normalization factor is  $\pi/T$ ).

$$\psi_\alpha(t) = \alpha + (1 - \alpha) \cos^2\left(\frac{\pi t}{T}\right) \tag{10}$$

with  $0 \leq \alpha \leq 1$  (the smaller  $\alpha$ , the larger the correction). The resulting “Xmas tree” past cone  $\tilde{\mathcal{C}}_x^-$  clearly shows on- and off-season peaks.

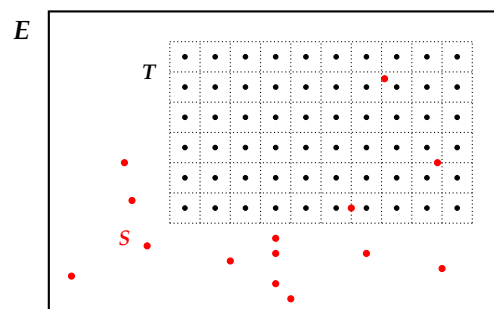


**Figure 5. (Left):** the causal cones  $\mathcal{C}_x^-$  and  $\tilde{\mathcal{C}}_x^-$  of  $x$ ; note that  $\tilde{\Delta}(y, x) < \tilde{\Delta}(z, x) = \infty$  even if  $\Delta(y, x) > \Delta(z, x)$ . **(Right):** the cross section of  $\tilde{\mathcal{C}}_x^-$ , backwards in time, in units of  $T$ . The slope of the dashed line is 1 and that of the dotted line is  $\alpha$ .

The trick operated by  $\psi_\alpha(t)$  in (10) consists of shrinking  $\mathcal{C}_x^-(k)$  the regular causal cone of aperture,  $k$ , down to  $\mathcal{C}_x^-(\alpha k)$ , i.e., a smaller cone of aperture  $\alpha k$ , during the off-season times, thus reducing the number of possible causes of  $x$  since  $\mathcal{C}_x^-(\alpha k) \subset \tilde{\mathcal{C}}_x^- \subset \mathcal{C}_x^-(k)$ . If  $\alpha = 0$ , the shrinking is so severe that  $\tilde{\mathcal{C}}_x^-$  becomes a union of saucers connected at single events on the cone axis. The  $n$ -dimensional shrunk causal cone  $\tilde{\mathcal{C}}_x^-$  is obtained by rotating the  $t \psi(t)$  function’s profile about the horizontal axis (Figure 5). In bi-dimensional models it is an ordinary solid of revolution; the rotational symmetry is required by the statistical hypothesis of spatial isotropy, while the widening along time is due to the (finite speed) propagation of the modeled field.

### 2.3. The Algorithm

A *Timescape model* consists of a set of estimated values of a scalar field  $\Phi$ , generally arranged as cells of a regular lattice (the *target T*). The field is interpolated from a set of known values (the *source S*). Each model is characterized by the choice of a spatial distance (the parameters  $c$  and  $k$ , and the form factor  $\psi$ ) and the spatial interpolation method. In general, there are no constraints on the distribution of the source events—see Figure 6.



**Figure 6.** Source (●) and target (●) events. The target events are the centers of a regular lattice of equal cells arranged as an  $(n+1)$ -dimensional box.

Some interpolation methods (e.g., Kriging) require the analysis of the variogram and are also able to evaluate the accuracy of the estimate, while some others (e.g., Inverse Dis-

tance Weighting) are less demanding but do not provide an accuracy figure. The Timescape algorithm operates a topological filtering of source events before the actual interpolation, so in principle one can choose any tool from the spatial statistical catalog.

The evaluation of a Timescape variogram deserves a note of caution. First of all, each target event  $x$  possesses its own input set  $S_x$ , the causal neighborhood of  $x$  (Figure 7), defined as

$$S_x = S \cap C_x^- = \{u \in S \mid \Delta(u, x) < \infty\} \tag{11}$$

so, as a matter of principle, each  $x \in T$  possesses its own variogram. However, a global variogram can be evaluated by the pairwise comparison of the field values within the source events  $S$ .

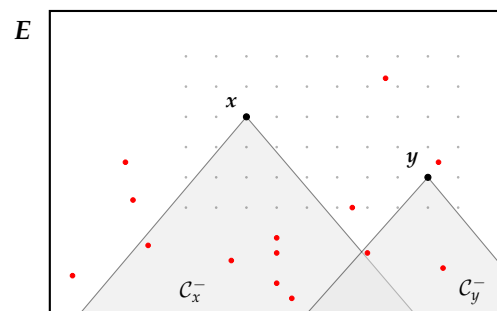


Figure 7. Two target events  $x$  and  $y$  with their input sets  $S_x$  and  $S_y$ .

To evaluate the variogram, we introduce the set  $S_\delta(h)$  of the causally connected pairs of  $S$  with a distance within the interval of width  $\delta$  about  $h$ :

$$S_\delta(h) = \left\{ (u, w) \in S \times S \mid h - \frac{\delta}{2} < \Delta(u, w) \leq h + \frac{\delta}{2} \right\} \tag{12}$$

the corresponding variogram value is

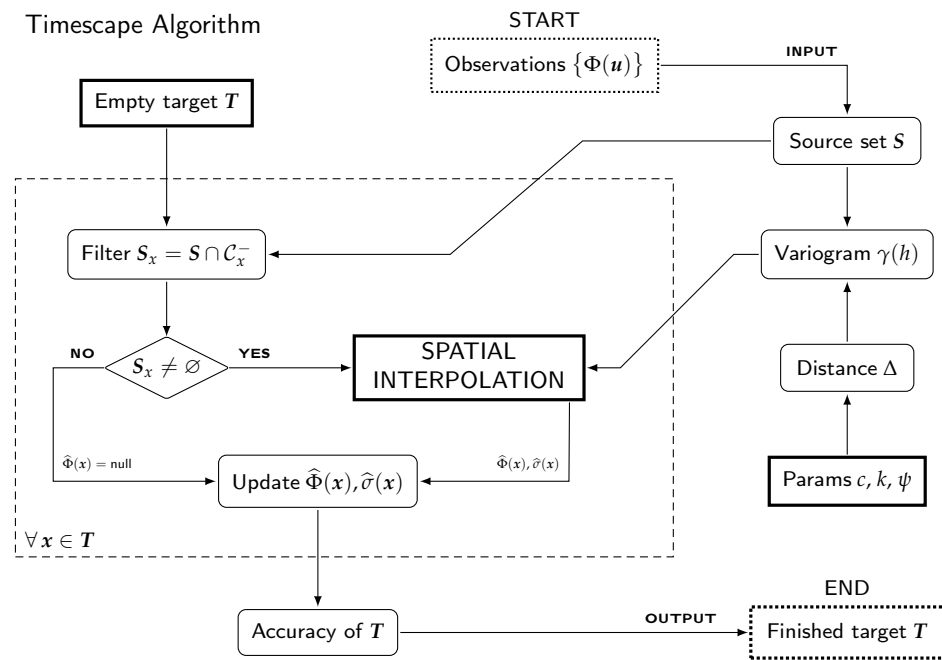
$$\gamma(h) = \frac{1}{|S_\delta(h)|} \sum_{(u,w) \in S_\delta(h)} \left( \Phi(u) - \Phi(w) \right)^2 \tag{13}$$

where  $|S_\delta(h)|$  is the number of elements of  $S_\delta(h)$ . The bin width  $\delta$  has to be chosen in order to have no overlapping bins, typically  $\delta = \frac{\max\{\Delta\}}{N}$ , where  $N$  is the optimal number of bins [39] and  $h$  is taken at regular intervals  $h_n = (n - \frac{1}{2})\delta$  counting  $n = 1 \dots N$ . There is not the usual variogram factor  $\frac{1}{2}$  in (13) since the asymmetry of  $\Delta$  in (12) prevents double-counting, in fact the intervals in (12) do not overlap but they do not contain 0. This is of no harm since a zero Timescape distance means *at the same place and the same time* and, since  $\Phi$  has to be single-valued, there cannot exist  $u, w \in S$  such that  $\Delta(u, w) = 0$  and  $\Phi(u) \neq \Phi(w)$ .

Generally speaking, for each target event  $x$ , the value of the field at  $x$  is a weighted average of the field values of the source:

$$\hat{\Phi}(x) = \sum_{u \in S_x} W(u, \Delta(u, x) \dots) \Phi(u) \tag{14}$$

the actual functional form of the weight  $W$  depends on the spatial interpolator of choice, distance being the only common input to all the interpolators. Note that if  $S_x = \emptyset$  there is no way to estimate  $\hat{\Phi}(x)$ . If this is the case, we attribute a null value to  $\hat{\Phi}(x)$ . Other than the field value  $\hat{\Phi}(x)$ , one has to calculate the accuracy of the estimate  $\hat{\sigma}(x)$ ; some interpolators provide such a figure, but some others do not. In the latter case, it is advisable to find the accuracy by jackknifing techniques upon the completion of the interpolation—see Section 2.4 for details.



**Figure 8.** Timescape flowchart. Users must choose a target set  $T$ , a spatial interpolator, the distance parameters  $c$  and  $k$ , and form factor  $\psi$  (thick boxes). The accuracy can be evaluated by the spatial interpolator itself or a posteriori, depending on the chosen spatial interpolator.

Summing up, the Timescape workflow proceeds as follows (a block flowchart is given in Figure 8): firstly, one must create an empty target  $T$ , choose a distance function (Equation (7) or (9)) and a spatial interpolator, then, for each target event, the algorithm goes as detailed in Table 1.

**Table 1.** Target events’ loop (dashed box of figure 8).

1	$C_x^- \leftarrow \mathbf{u} \in S \mid \Delta(\mathbf{u}, \mathbf{x}) < \infty$	Equation (7) Equation (9) Figure 3
2	$S_x \leftarrow S \cap C_x^-$	filtering—Equation (11) Figure 7
3	$\eta(\mathbf{x}) \leftarrow  S_x $ if $S_x = \emptyset \implies \hat{\Phi}(\mathbf{x}) \leftarrow \text{null}$	Equation (15)
4	else $\hat{\Phi}(\mathbf{x}) \leftarrow \sum_{S_x} W(\mathbf{u} \dots) \Phi(\mathbf{u})$ if <b>interp.</b> allows $\hat{\sigma}(\mathbf{x}) \leftarrow \sqrt{\text{var}}$ else $\hat{\sigma}(\mathbf{x}) \leftarrow \text{null}$	estimate—Equation (14) Kriging... IDW...
5	update $\mathbf{x}$ : $[\hat{\Phi}(\mathbf{x}), \hat{\sigma}(\mathbf{x}), \eta(\mathbf{x})]$	

It is advisable to allocate all the space needed by the finished  $T$  before starting the actual interpolation, not to run short of resources during the interpolation. The Python and Java Timescape implementations do so, as a bulk of internal memory (Python) or database records (Java)—see Section 2.5.

The loop above can be parallelized in a number of ways, since all the events in  $T$  are all considered independent of each other. A useful performance figure can be calculated during the interpolation—it is the number  $\eta(\mathbf{x})$  of elements in  $S_x$ :

$$\eta(\mathbf{x}) = |S_x| \tag{15}$$

the larger  $\eta(\mathbf{x})$ , the more robust are  $\hat{\Phi}(\mathbf{x})$  and  $\hat{\sigma}(\mathbf{x})$ .

Upon the completion of the loop one has to estimate the model’s accuracy, if it has not already be done by the interpolator itself. If it is not the case, the accuracy can be estimated



by jackknifing the source set [40], removing a few random elements at a time, and repeating this process a number of times  $N$  [41]. A consistent standard deviation estimate is thus [42]:

$$\hat{\sigma}(x) = \sqrt{\frac{1}{N-1} \sum_k (\hat{\Phi}'_k(x) - \hat{\Phi}(x))^2} \tag{16}$$

where  $S'_k$  is the  $k$ th replica ( $k = 1 \dots N$ ) of  $S$  with some random elements removed,  $\hat{\Phi}(x)$  is the model's field estimate at  $x$ , and  $\hat{\Phi}'_k(x)$  is its estimate on the subset  $S'_k$ . Note that this accuracy estimate is very time-consuming. On practical grounds, (16) can be performed only on a few  $S'_k$  sub-sources, since the running time of each subset is comparable with that of the full Timescape model. Note also that one cannot remove too many events from  $S$  since it could lead to too many empty  $S'_k$  sets, i.e., too many null  $\hat{\Phi}'_k(x)$ . Further details on the procedure can be found in Python Timescape module manual [31].

The overall accuracy can be assessed by two figures: a global variance  $\sigma_{\text{model}}^2$  and the ratio of null target events over the total target size  $\eta_{\text{model}}$

$$\sigma_{\text{model}}^2 = \frac{1}{|T^*|} \sum_{x \in T^*} \hat{\sigma}^2(x) , \quad \eta_{\text{model}} = \frac{|T^*|}{|T|} \tag{17}$$

where  $T^* = \{x \in T \mid \hat{\Phi}(x) \neq \text{null}\}$  is the set of not-null elements of  $T$ . The  $\eta_{\text{model}}$  index, in particular, measures the predictive power of the model.  $\sigma_{\text{model}}$  is expressed in the same units of  $\Phi$ , while  $\eta$  is adimensional and  $0 \leq \eta \leq 1$ .

#### 2.4. Model Tuning

Given the same input set  $S$  and output  $T$ , the very character of a Timescape model is given by a number of factors: the spatial interpolator, the metric parameters  $c$ ,  $k$ , and the form factor  $\psi$ . We give some suggestions in order to wisely choose an optimal distance.

##### 2.4.1. Spatial Interpolator

The vast family of Gaussian process regression techniques (in short, Kriging) provides a wide choice of interpolators. Furthermore, such techniques provide an accuracy estimate such as a variance figure, so that there is no need to perform the post hoc jackknifing. On the other hand, some old-style deterministic interpolation techniques, like the Inverse Distance Weighting, though lacking the accuracy, are extremely lightweight and fast to perform, so they should be considered, especially when geographic coordinates are involved, with a spatial distance like (3) (As a matter of principle, one should evaluate the geodesic distances on the Earth as elliptic integrals, while (3) just measures the arc length between the locations  $\vec{x}$  and  $\vec{y}$ ). As a rule of thumb, it is advisable to pick the same spatial interpolator one would choose if all the events were located at the same time.

##### 2.4.2. Metric Parameters

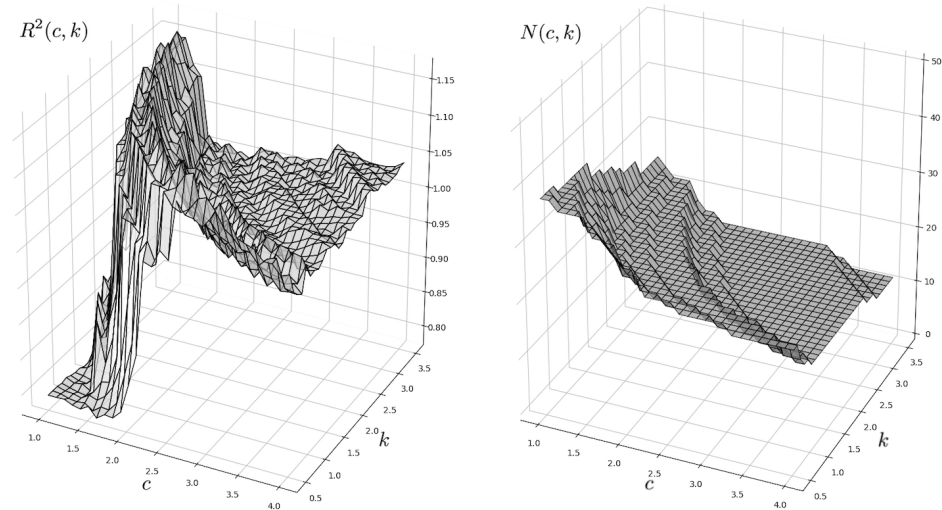
$c$  and  $k$  affect the metric of  $E$ : tuning these values is of paramount importance for a successful interpolation. Other than a trial-and error strategy, one should pick the best performing  $(c, k)$  pair. What best performing means is a compromise between fast running and accuracy [43]. We recommend a procedure based on residuals minimization by source jackknifing [40]. The idea is based on the cross-estimation of the field values of  $S$  with  $S$  itself, similar to (16) with  $S$  in place of  $T$ , for a whole ensemble of  $(c, k)$  values, picking the best performing pair  $(\bar{c}, \bar{k})$ .

Let us define  $\hat{\Phi}_{ck}(u)$  the Timescape estimate of  $\Phi(u)$ —which is known—with parameters  $(c, k)$ . The estimate is performed on the reduced source set  $S \setminus \{u\}$ . Mimicking (16), we introduce two indicators: the cumulative residuals  $R^2(c, k)$  and the non-null events ratio  $N(c, k)$ :

$$R^2(c, k) = \frac{1}{|S| - 1} \sum_{u \in S} (\hat{\Phi}_{ck}(u) - \Phi(u))^2 \tag{18}$$

$$N(c, k) = \left| \{ \mathbf{u} \in \mathcal{S} \mid \widehat{\Phi}_{ck}(\mathbf{u}) \neq \text{null} \} \right| \tag{19}$$

the best performing pair  $(\bar{c}, \bar{k})$  can be found by minimizing  $R^2(c, k)$  and maximizing  $N(c, k)$  at the same time. Figure 9 shows an example of  $R^2$  and  $N$  surfaces as functions of  $c$  and  $k$  on a regular lattice. Note that, for a given  $c$ ,  $N(c, k)$  is a decreasing function of  $k$ : this is the consequence of the fact that a larger  $k$  implies a looser causality (see Figure 4).



**Figure 9.** (Left): An example of  $R^2(c, k)$ . (Right):  $N(c, k)$  in percent units. Plots from a dedicated function of the Python package [31].

As a rule of thumb, any ensemble of  $(c, k)$  values should initially include  $k_0 = 1$ , i.e., treating space and time on equal grounds. As of  $c$ , if there is a characteristic velocity in the model (transport, diffusion, etc.) one can play around such a value, otherwise a naïve first guess could be, given  $\mathbf{u}, \mathbf{v} \in \mathcal{S}$ :

$$c_o = \left[ \max_{\mathbf{u}} (u_0) - \min_{\mathbf{u}} (u_0) \right]^{-1} \times \max_{\mathbf{u}, \mathbf{v}} \left[ d(\vec{\mathbf{u}}, \vec{\mathbf{v}}) \right] \tag{20}$$

ensuring a fair coverage of the causal cones: for Euclidean spaces, (20) makes  $\mathcal{S}$  base diameter and height equal.

### 2.4.3. Form Factor

The topology of the event space–time  $E$  is controlled by the form factor  $\psi$ . The causal cones are actual cones if  $\psi \equiv 1$  or any positive constant, which by the way can be absorbed into  $k$ . The form factor is used to modify the shape of the cones and so the topology of  $E$ , promoting or depressing the contribution of the events according to their temporal distance. We propose two cases:

**Threshold:** we can impose a threshold  $T$  as a maximum time distance requiring  $c |y_0 - z_0| \leq T$ , posing  $\psi(t) = \theta(T - t)$  whose effect is to close the cone  $\tilde{\mathcal{C}}_y^+$ . See Figure 10 for a comparison with a regular  $\mathcal{C}_x^+$ , with  $\psi(t) = 1$ .

**Periodicity:** a periodic  $\psi(t) \leq 1$  accommodates seasonality into a Timescape model shrinking the causal cone (Figure 5), thus filtering out the off-season events.

Given a period  $T$ , a straightforward form factor is  $\psi(t) = \cos^2 \omega t$ , where  $\omega = \pi/T$ . If one wants to be more or less selective, some variations on the theme are at hand (Figure 11). Higher even powers of the cosine have sharper maxima and flat minima, so the cone is more selective. On the other hand, roots of cosine give smoother maxima and sharp minima, corresponding to a looser causal constraint. The cross section of the  $\tilde{\mathcal{C}}^\pm$  cones is given by the product function  $t \psi(t)$ .

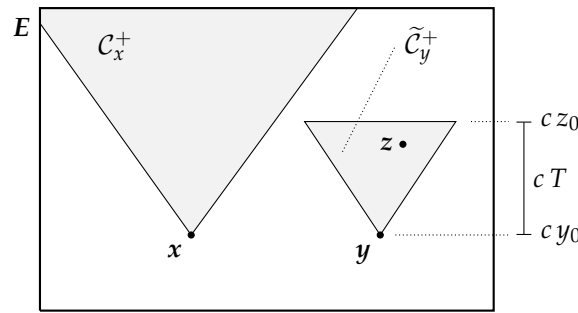


Figure 10. The effect of  $\psi(t) = \theta(T - t)$ .

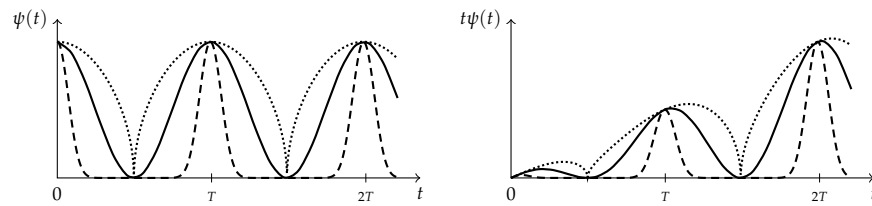


Figure 11. (Left): three periodic form factors, solid line  $\psi(t) = \cos^2 \omega t$ , dashed line (stricter)  $\psi(t) = \cos^{20} \omega t$ , dotted line (looser)  $\psi(t) = \sqrt{|\cos \omega t|}$ . (Right): the corresponding cross-sections of the causal cones.

As a rule of thumb, as a first guess take  $\psi(t) = 1$  for non-seasonal field values and  $\psi(t) = \cos^2(\frac{\pi}{T}t)$  for seasonal fields modeling, then  $\alpha$ -tempering the form with  $\psi_\alpha$  in (10), as discussed therein. The larger the area below  $\psi$ , the looser the seasonal selectivity.

### 2.5. Timescape Implementations

Two algorithm implementations have been published: a Python and two Java versions. All the software packages are released under the GNU-GPL license. Other than going deep into the mathematical subtleties of Sections 2.1 and 2.2, the manuals accompanying the software distributions treat in detail the practical implementation and describe the code. In this paper, we only sketch the key features of the software. Both implementations are limited to two spatial dimensions and provide some statistical analysis tools to explore the source dataset and the finished models.

#### 2.5.1. Python

The Timescape Python module [31] consists of a single module, exposing all the objects, methods, and functions needed to perform source data analysis, parameter tuning, and the actual model interpolation, alongside additional functions dedicated to data plotting and exporting. All the examples in Section 3 have been computed with this software.

The usage is command line-based, but the functions can be easily integrated into users' own code. The default spatial interpolator exploits the PyKriging module functions [44]; a finished model is a Python object that can be saved in binary format and can be included in a GIS workflow as a multi-layered geotiff image [45], one band per time sheet. The storage is RAM-based, so this is the limiting factor for the number of target cells.

The Python package is designed to be easily configurable, allowing users to code their interpolators. Predefined interpolation methods include Kriging (examples in Sections 3.1 and 3.2) and Inverse Distance Weighting (Sections 3.3 and 3.4). Other than the Euclidean (2) and Geodesic (3) spatial distances, square and diamond metrics are provided. The allowed form factors are  $\psi \equiv 1$  and a periodic  $\psi = \cos^2 \frac{\pi t}{T}$  (Equation (10), see for example Section 3.2). Defining other distances or form factors requires a non-trivial intervention on the code.

### 2.5.2. Java

The Timescape Java implementation consists in two stand-alone applications, one for projected coordinates, the local version [32,46], and one for geographical coordinates, the global version [33,47]. These are distributed as self-consistent jar executables.

The storage of the Timescape models is based on a Hibernate midlayer [48] that manages the connection with any compatible relational database management system (MySQL [49] scripts are included in the distribution). The user interaction is mediated by a Graphical User Interface based on standard swing widgets. The database storage is more flexible, in that each target event is just a record of a table, but there is the need for a dedicated database service.

The Java applications allow a full customization of distance and form factor via a javascript-like scripting language but, since it requires run-time interpretation, the performance is dramatically affected. It is also possible to include ancillary variables along with—or even instead of—the source field values.

The java applications come with a utility for the configuration of the database connection and other parameters. A set of tools is also provided to explore, subset, and export the finished models.

### 2.6. Coping with Binary and Count Data

The Timescape Algorithm per se is designed for the interpolation of real scalar fields, however, discrete data (binaries and counting) can be treated considering the source dataset's values as reals and applying these workarounds:

Counts: Apply ex-post (i.e., on the finished interpolated values) the floor or ceiling function (The floor  $\lfloor x \rfloor$  is the closest integer lesser than or equal to  $x$ , similarly, the ceiling  $\lceil x \rceil$  is the closest integer greater than or equal to  $x$ ) thus reducing real values to integers:  $\hat{\Phi}(x) \mapsto \lfloor \hat{\Phi}(x) \rfloor$  or  $\hat{\Phi}(x) \mapsto \lceil \hat{\Phi}(x) \rceil$ . A less crude method consists of counting the passing of a series of not equally-spaced numerical thresholds  $\{T_1 \dots T_n\}$ , mapping  $\hat{\Phi}(x) \mapsto \sum_k \theta(\hat{\Phi}(x) - T_k)$ . The latter reduces to  $\lfloor \hat{\Phi}(x) \rfloor$  when  $\{T_1 \dots T_n\} \equiv \{1 \dots n\}$ .

Binaries: we propose two distinct techniques. The first one consists simply of ex-post thresholding:  $\hat{\Phi}(x) \mapsto \theta(\hat{\Phi}(x) - T(x))$ , where  $T(x)$  is a threshold value, possibly depending on the event  $x$ ; in its simplest version, representing true/false source events' values in the  $\{0, 1\}$  set and given a single threshold  $T \in (0, 1)$ , map eventually

$$\hat{\Phi}(x) \mapsto \theta(\hat{\Phi}(x) - T) = \begin{cases} 1 \sim \text{true} & \text{if } \hat{\Phi}(x) \geq T \\ 0 \sim \text{false} & \text{otherwise} \end{cases} \quad (21)$$

The second technique consists of loosely mimicking the neural network propagation of signals [19,20,50]: only the first time sheet of target events is evaluated as usual, then the sheets  $\hat{\Phi}(x)$  are reduced to firing/true and silent/false by (21) and used as input for the next time sheet, which is in turn reduced through (21) and so on, up to the last sheet, thus terminating the algorithm with binary target values. Notice that this technique introduces fictitious source values.

The Java Timescape version (Section 2.5.2) provides a scripting language that can be exploited to implement ex-post thresholding.

## 3. Results

We present four case studies. They cover a variety of coordinate systems, time scales, and peculiarities encountered in ecological geostatistics: projected and geographical coordinates, seasonal effects, uneven distribution, and number of observations.

Three examples come from the field of stable isotopes spatial and temporal distribution, also known as isoscapes [51], an increasingly widespread tool in the complex field of spatial ecological analysis [52–54]. Isoscapes (i.e. isotopic landscapes) consist of spatial or spatiotemporal distributions of stable isotopes elemental ratios [51], evaluated after a set of suitable measurements. From a spatial point of view, isoscapes are just maps, or series

of maps at different times; from a formal point of view, isoscapes are ordinary real scalar fields, the relevant random variable being the actual isotope ratio, depicted as a function of space and, possibly, time. Isoscapes are an important tool in ecological modeling, since they can highlight spatial patterns in complex bio- and geochemical interactions, from the field of hydrology [55] to plant physiology [56,57] to animal migration [58] and to forensic science [59], naming only a few. In particular, we focus on the spatial distribution and temporal evolution of three stable isotopes from the fields of hydrology and plant physiology. The interpolated field in such isoscapes is the isotopic abundance of hydrogen ( $\delta^2\text{H}$ ), nitrogen ( $\delta^{15}\text{N}$ ) and oxygen ( $\delta^{18}\text{O}$ ). In further case studies, the interpolated field is temperature, demonstrating the potentiality of the Timescape algorithm to deal with seasonality.

The measured isotope ratios are usually expressed as differences with respect to a given reference standard, the so-called  $\delta$ -notation:

$$\delta^nX\text{‰} = 1000 \times \frac{R_{\text{sample}} - R_{\text{standard}}}{R_{\text{standard}}} \tag{22}$$

where  $R$  is the molar ratio of the heavy  $^nX$  stable isotope vs. the lighter, most abundant  $^mX$ , the factor 1000 rescales the otherwise small numbers to easily readable figures; the deltas are adimensional; for example,  $\delta^{13}\text{C}$  represents the abundance of  $^{13}\text{C}$  vs.  $^{12}\text{C}$  in a given sample. A positive  $\delta^{13}\text{C}$  indicates an enrichment in  $^{13}\text{C}$  vs. the common  $^{12}\text{C}$  with respect to the standard, while a negative  $\delta^{13}\text{C}$  denotes a  $^{13}\text{C}$  depletion.

Table 2 resumes the spatial and temporal characteristics of the source events for each model: number of events, geometry, spatial scale and temporal extension. All the examples are available as extra material, included in the Python software distribution [31] (For the sake of reference, all the examples are accompanied by R’s SpatialPointsDataFrame objects [14,60] with the corresponding projection’s CRC objects [61]).

**Table 2.** Model features.  $|S|$  is the number of source events.

Model	$ S $	Coordinates	~Scale	Time Span-Resol.
Fungi $\delta^{15}\text{N}$	62	Local Euclidean $(x, y)$	100 m	few days– 1 day
Min temperature	2521	WGS84 UTM33 N	100 km	20 years–1 month
Olive oil $\delta^{18}\text{O}$	275	European Lambert	1000 km	3 years–1 year
Precipitation $\delta^2\text{H}$	1152	Geographical $(\lambda, \phi)$	$10^4$ km	10 years–1 year

Table 3 shows the details of the target events set  $T$  consistency, the spatial interpolation tool employed (Kriging or Inverse Distance Weighting), the distance function, and the causal cone form factor.

**Table 3.** Model parameters. The target cells are ordered by time  $\times$  horizontal  $\times$  vertical.

Model	Method	Distance	Form	Target Cells $ T $
Fungi $\delta^{15}\text{N}$	Kriging	Euclidean	$\psi \equiv 1$	$64 \times 128 \times 128 = 1,048,576$
Min temperature	Kriging	Euclidean	Equation (10) $\alpha = 0$	$240 \times 25 \times 32 = 192,000$
Olive oil $\delta^{18}\text{O}$	IDW	Euclidean	$\psi \equiv 1$	$7 \times 74 \times 99 = 51,282$
Precipitation $\delta^2\text{H}$	IDW	Geodesic	$\psi \equiv 1$	$20 \times 120 \times 50 = 120,000$

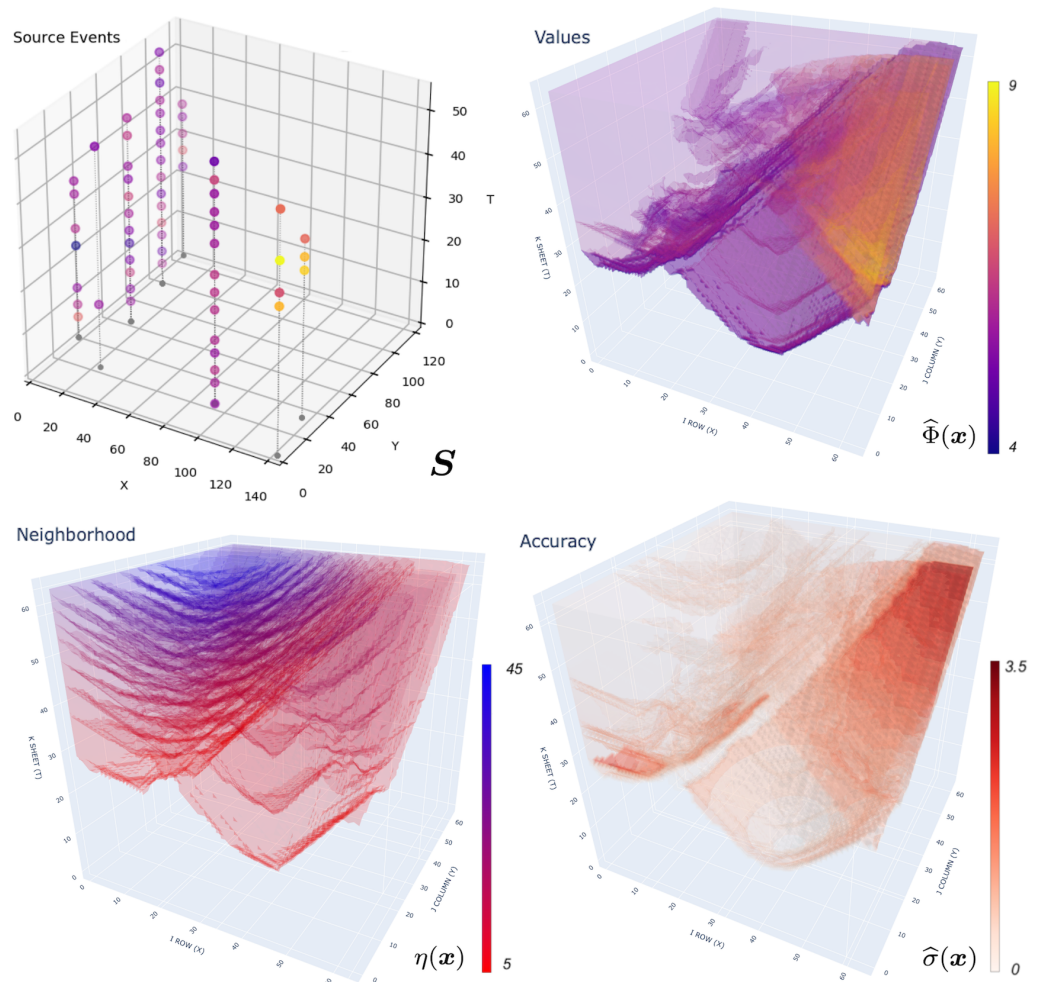
In the following, we briefly describe each case.

### 3.1. Fungi $\delta^{15}\text{N}$

This dataset comes from a study about host trees–mycorrhiza relationship conducted via geostatistical methods exploiting carbon and nitrogen stable isotopes abundances [62]. The model’s scalar field represents the hypogeous fungi’s ascomata  $\delta^{15}\text{N}$  (*Tuber aestivum* Vittad—edible black truffles). The complex tree–fungi relations include the nitrogen transport from fungi to the host tree, in particular, the  $\delta^{15}\text{N}$  difference, or fractionation, is a clue

of tree–fungi symbiosis [63–65], with a significant difference in terms of  $\delta^{15}\text{N}$ , resulting in a fungal  $^{15}\text{N}$  enrichment with respect to the host leaves [66,67]. From a spatio-temporal point of view, the collection took place within 64 days of continuous fungi–tree interactions, so it was impossible to treat spatial and temporal variability on different grounds. This study led to the development of the Timescape algorithm.

The source consists of 62 events, repeated measures at the same sites, skew-distributed both in space and time. The target is a stack of 64 time sheets counting  $128 \times 128$  spatial cells each (about one million cells).

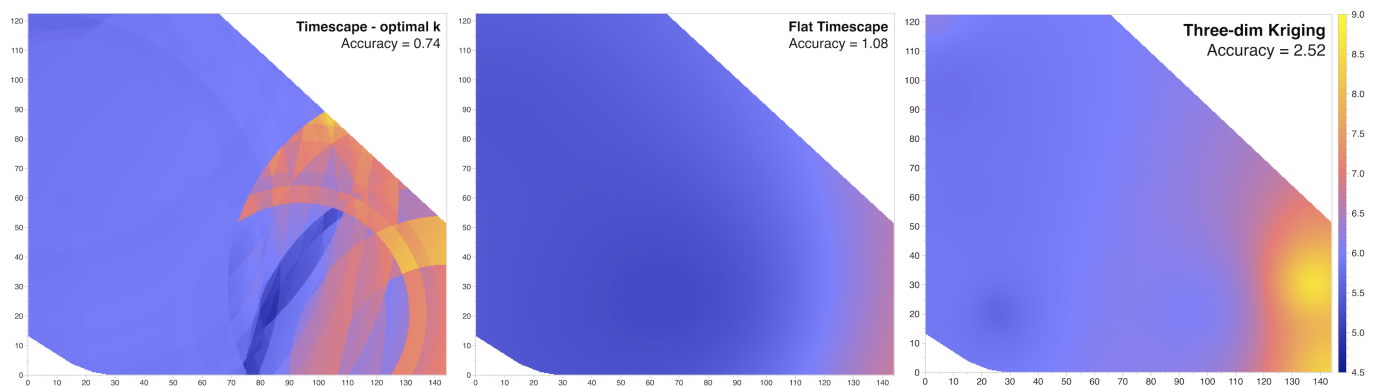


**Figure 12.** (Top left): source events  $S$ . (Top right): target field values  $\hat{\Phi}(x)$ . (Bottom left):  $\eta(x)$  i.e., events in the causal neighborhood  $S_x$ . (Bottom right): accuracy  $\hat{\sigma}(x)$ . The source coordinates are the actual ones, the target coordinates are cell indexes.

Figure 12 illustrates the typical Timescape behavior, time increasing from bottom to top: as a new observation comes in, there is a sort of perturbation in the field that pushes its value towards that of the following ones. This perturbation will eventually consolidate or fade according to the closest subsequent source events. The round shapes are due to the choice of Euclidean metric (2) as the spatial distance.

When treating cases like this one, it is important to check the accuracy of the interpolation, so it is advisable to choose a statistical interpolator like Kriging, which is capable of calculating a target event’s variance estimate. Such variance could be relatively high, but it is noteworthy that a time-unaware geostatistical interpolation would be impossible to perform altogether. As it appears from Figure 12, accuracy stabilizes about the value 1 as time goes on; the upper right corner is meaningless since, spatially, it lies out of the

convex hull of the source events, it would be a non-informative area with any ordinary geostatistical tool.



**Figure 13.** The topmost sheet of three models. **(Left):** fine-tuned Timescape, i.e., the topmost horizontal sheet of predicted values  $\hat{\Phi}(x)$  (top right of Figure 12). **(Center):** the loosest possible Timescape. **(Right):** three-dimensional Kriging.

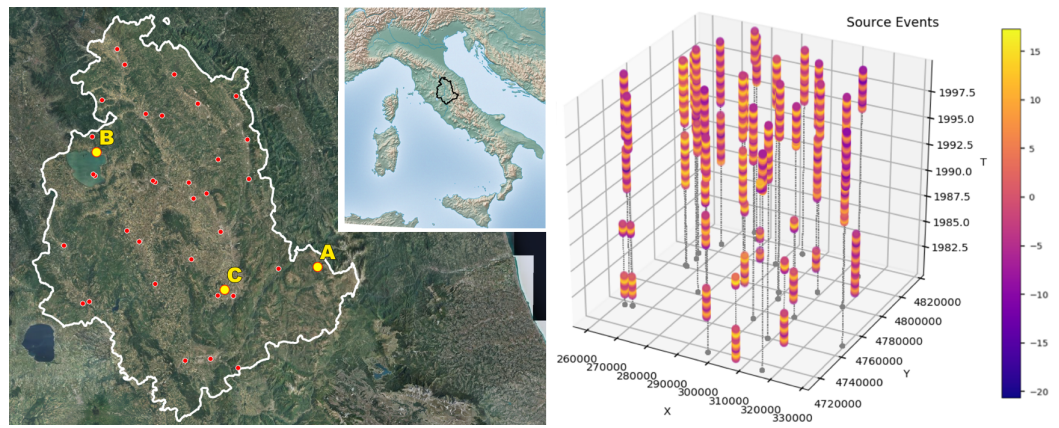
Figure 13 shows a comparison between two Timescape interpolations and a three-dimensional spatial Kriging. The leftmost graph shows the optimally tuned Timescape, and it corresponds to the upper sheet of Figure 12 top-right, predicted values. Its accuracy, estimated by the residuals with the source, is 0.74. The middle graph is from a Timescape model with identical parameters, except for  $k \rightarrow \infty$ . It is the loosest causal constraint, the model's accuracy is 1.08, significantly looser than the fine-tuned Timescape. The rightmost graph is derived from a purely spatial Kriging: the time is transformed into the third spatial dimension via the same conversion factor  $c$  of the previous models. The accuracy is only 2.52. The leftmost graph in Figure 13 clearly shows the interference of the causal cones as they propagate over time; the circular shapes are due to the choice of Euclidean metric for the spatial distance. The loose Timescape clearly shows the importance of the causal parameters  $c$  and  $k$ . In fact, the loose Timescape is not far from a purely spatial model, as the one rightmost graph of Figure 13, the only difference being that for the loose Timescape  $C_x^- = E_x^-$ , while for the three-dimensional Kriging  $C_x^- = E \forall x$ —see also the topological modifications of the Timescape distance in the discussion, Section 4.2.

### 3.2. Lowest Temperatures

This case study shows the ability of the Timescape algorithm to cope with seasonal variability. The source dataset consists in the monthly lowest temperatures recording of the Umbria region in central Italy [68,69].

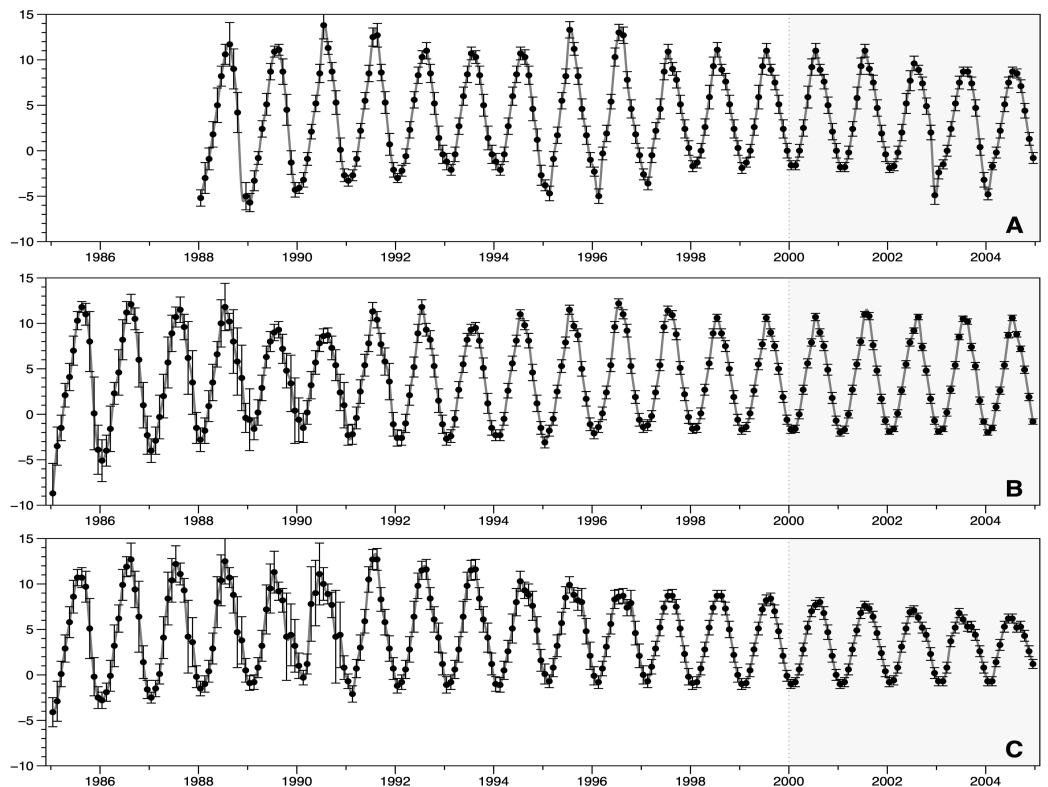
The temperature records span 20 years with monthly time resolution (1/12 year), and their spatial resolution increases over time as more stations have been added in the two decades considered. The interpolation has been conducted with a periodic causal cone ( $T = 1$  year) using universal Kriging. The great number of source events, more than 2000, added further computational load: in fact, the throughput is as low as 30 target events per second, see the following Table 4 in Section 3.5.

The source events are shown in Figure 14. Note the temporal gaps in the measurements. Additionally, new stations started logging data over time, resulting in denser events from year 1990 on, while around 1980 the collection is sparse in both space and time. This is the typical distribution skewness of surveys based on a growing measurement network.



**Figure 14.** **Left):** The weather stations network (red dots) in Umbria; A, B and C are referred to the actual sites of the time series of Figure 15. **(Right):** spatio-temporal distribution of the source events. Base map and boundaries from NaturalEarthData [70] and Geoportale Nazionale [71], map prepared with Qgis [72].

The seasonal character of the output is apparent in the time series shown in Figure 15, extracted from the Timescape at the sites marked A, B, and in Figure 14: the upper time limit is 2005 (five years later than the source data ending), so in fact we are forecasting the values (shaded areas).



**Figure 15.** Time series extracted at the sites A (top), B (middle) and C (bottom) marked in Figure 14. Time is in years and field values in °C. The shadowed area marks the extrapolated values. Time series A starts in 1988, as there were no recording stations nearby before.

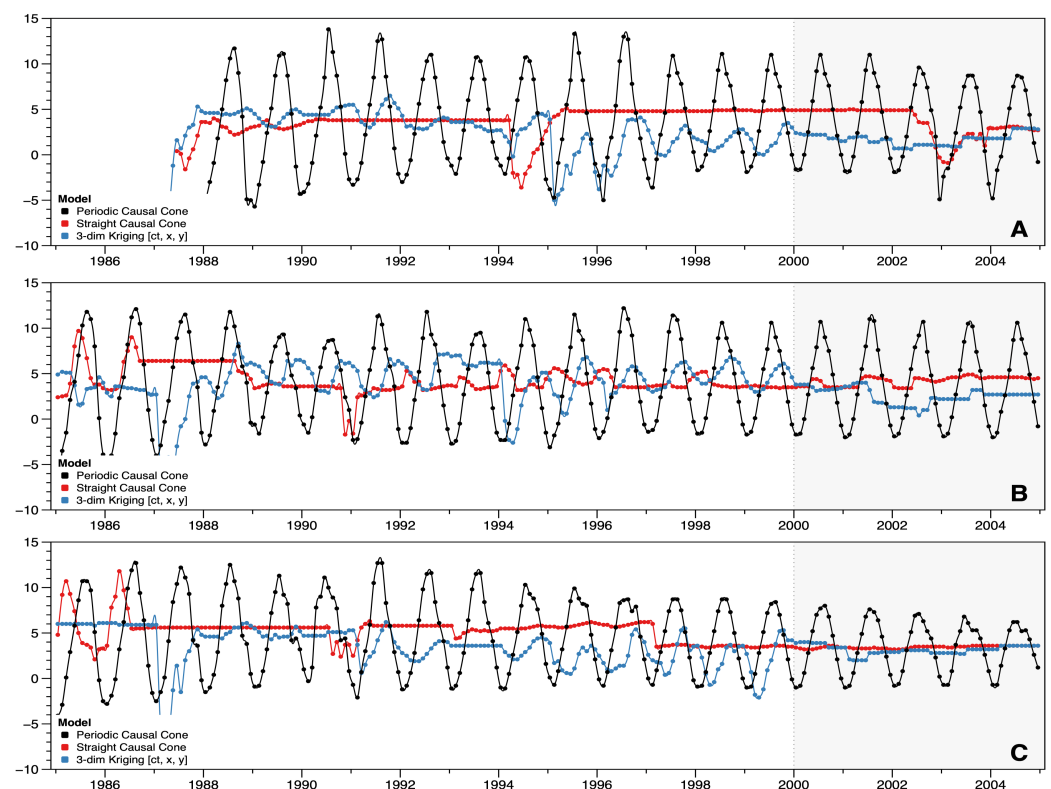
The series period remains constant (one year, of course) and the damping effect is negligible. Seasonal variability is cumbersome for Kriging [73]; in our case, the Timescape algorithm filtered out the off-season source values, thus reproducing the correct periodic behavior. The plots shown in Figure 15 have different starting times, corresponding to



the earliest meaningful time (i.e., the first non-null  $\hat{\Phi}(x)$ ) for the the given site. The time resolution of the series is one month. The B series of Figure 15 is initially influenced by outlying low values, due to the influence of a distant mountain station, which eventually becomes irrelevant, as closer measuring stations started logging data over time.

It is worth noting that the accuracy increases as more and more events are included in the target  $x$  past causal cones  $C_x^-$ . This effect is not a sort of reverse heteroskedasticity, it is just an effect of  $\eta(x)$  increasing with time (Equation (15)).

This example demonstrates how easily the Timescape distance incorporates seasonality. Figure 16 shows our model versus a “wrong” non-seasonal Timescape and a spatial (three-dimensional) Kriging. It clearly appears from the time series that seasonality is totally absent in the non-seasonal Timescape, whose other parameters are identical to the original model, and that the spatial Kriging reproduces an extremely damped seasonality. The Kriging has been performed, transforming the input times in metric coordinates (UTM—Universal Transverse Mercator projected coordinates) and multiplying time by the same  $c$  of the Timescape model, which has the physical dimensions of a velocity, see Table 2.



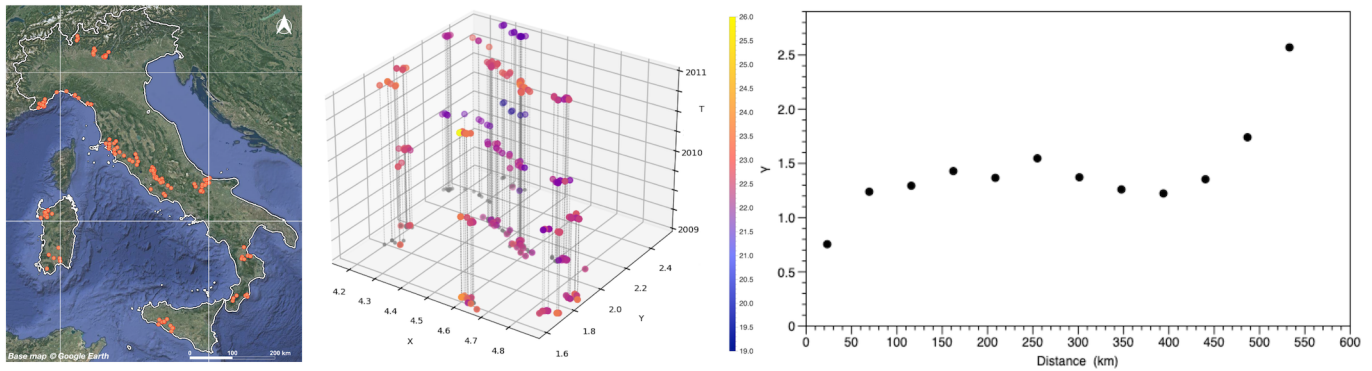
**Figure 16.** The same time series of Figure 15 at sites (A)–(C) compared with different interpolation techniques. Black: periodic Timescape, the actual case. Red: Straight cone (i.e., non-seasonal) Timescape. Blue: three-dimensional spatial Kriging. Time is in years and field values in °C, error bars not shown.

### 3.3. Extra Virgin Olive Oil $\delta^{18}\text{O}$

This dataset comes from a study on the geographic traceability of Italian Extra Virgin Olive Oil (EVOO) through isoscape analysis [74] based on the spatial distribution of Carbon [57,75,76] and Oxygen [77–79] isotopic composition of precipitation and source water [80]. The  $\delta^{18}\text{O}$  isoscapes, in particular, are related to the elevation and to the distance from the sea and correlate with the year’s precipitation [77,78]; the original results in [74] have been obtained by standard spatial statistical procedures.

The source events distribution and variogram are shown in Figure 17. Measurements have been all taken at regular time intervals (once a year, at olive harvest time), as the spa-

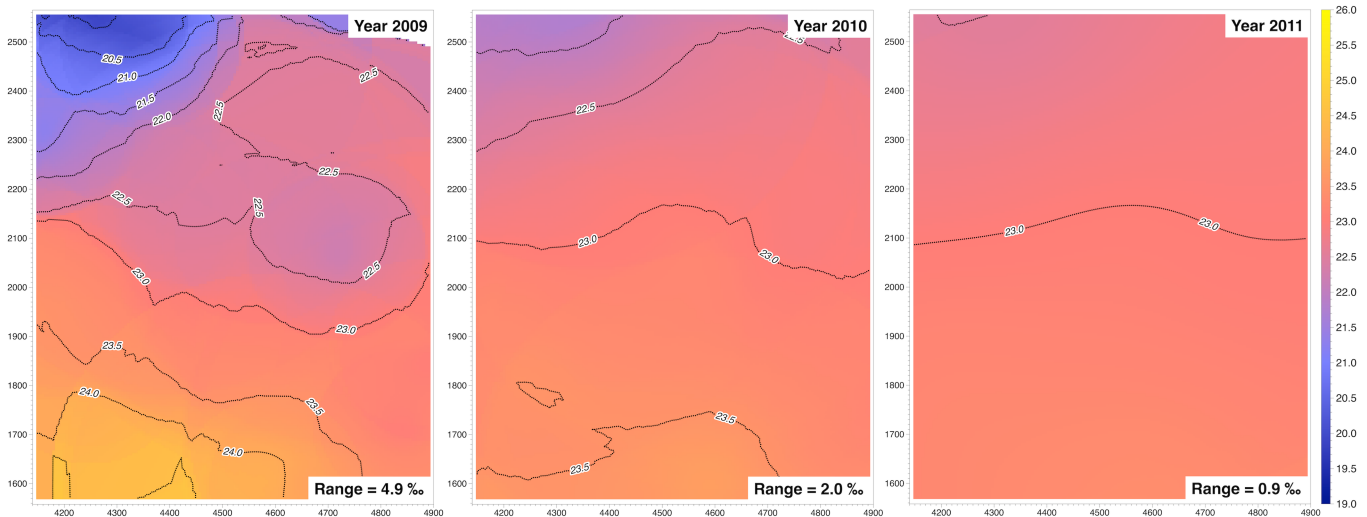
tiotemporal distribution clearly shows three distinct layers in the years 2009, 2010, and 2021. Furthermore, the EVOO's  $\delta^{18}\text{O}$  values are linked to the harvest years' precipitation [74], with nearly no correlation from one year to the other [77]. Although the variogram suggests a good source events space–time distribution, the lack of intra-year correlation makes the Timescape algorithm fail.



**Figure 17.** (Left): source events spatial distribution. (Center): source events spatiotemporal distribution, notice the layered time structure, time is expressed in years, space distances in thousands of kilometers. (Right): the spatiotemporal variogram (13), distances in kilometers.

The Timescape interpolation of the same data gives results that are comparable with [74] for the first year only (Figure 18, left), while the subsequent years (center and right) do not match well with the source events. Note in particular how the extreme values are averaged out in 2010 and 2011, completely washing out the output variance.

This example shows clearly that the Timescape algorithm is not a universal tool. In this case, each year's data are self-contained and independent of each other, so mixing three years of observations is not correct.



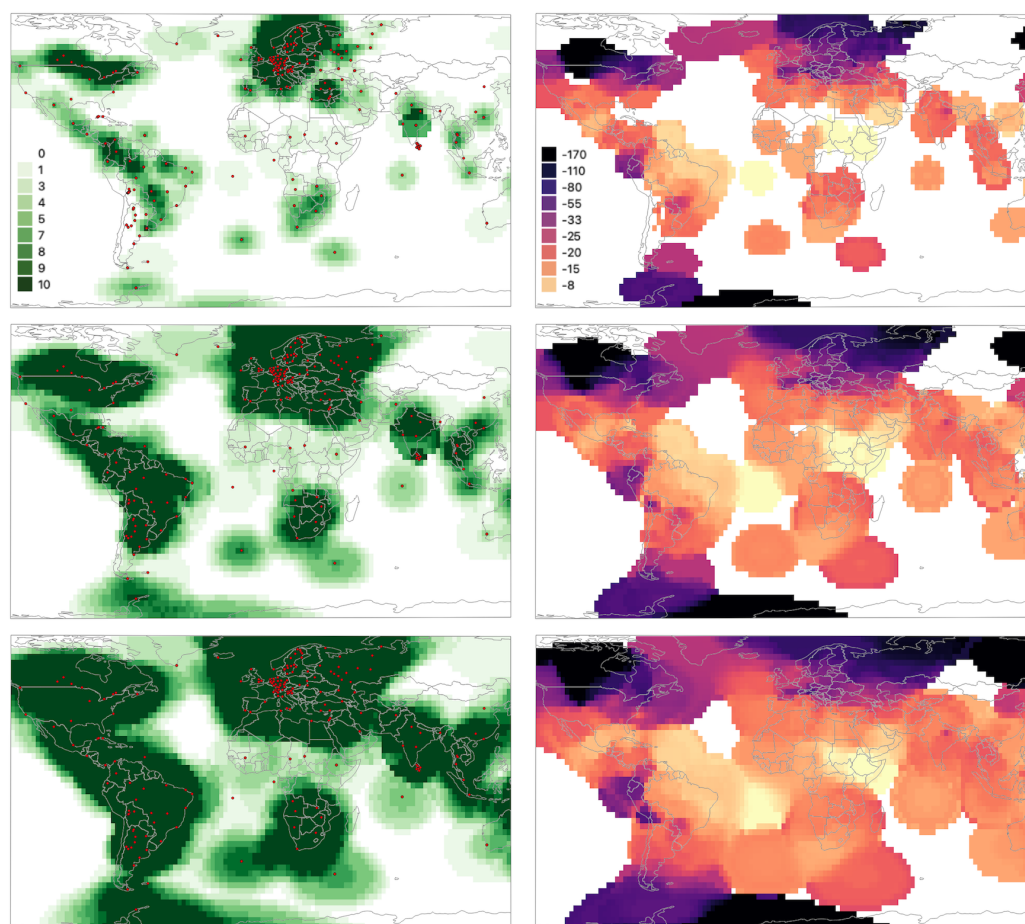
**Figure 18.** 2009, 2010, and 2011  $\delta^{18}\text{O}$  isoscapes as sheets of the model. Plots prepared with Data-Graph [81] from exported geotiff images.

The computation per se, however, is straightforward: the high events throughput (more than one thousand per second, see Table 4) is due to the relatively scarce number of source events and, mostly, to the computational speed of the IDW. In this case, the weight Function (14) is simply  $W(u, x) = \Delta(u, x)^{-1}$ . It is worth noting that the same model interpolation by Kriging gave rise to a significant number of errors, about 35% of the target events, in spite of the relatively large width of the casual cone ( $k = 5$ ). This is due to the fact that each year's  $\delta^{18}\text{O}$  integrates that particular year's water intake of the olive tree,

and it is unrelated to other years' values, resulting in different values with the same spatial coordinates, which is particularly disturbing for Kriging [4]—the high number of target event computational errors is an alarm bell, signaling a possible inconsistency of the source events' values or a badly chosen interpolation tool, as it is in this case.

### 3.4. Precipitation $\delta^2H$

This example is based on the renowned Global Network of Isotopes in Precipitation (GNIP) dataset [82,83] from the International Atomic Energy Agency [84]. The collection of isotopic abundance data started in the early 1960s and the station's network has grown dramatically over time. For historical and technical reasons, the spatial coverage is uneven (Figure 19), with the earlier, denser coverage in central Europe.



**Figure 19.** Three sheets of the Timescape model, time increasing top to bottom (year 1981, 1985, and 1989). **(Left):** Causal neighborhood  $\eta(x)$ , source events' sites are shown as red dots. **(Right):** field values  $\hat{\Phi}(x)$ , note the typical bull's eye shapes due to the IDW algorithm.

The interpolation algorithm chosen is a smooth version of the Inverse Distance Weighting method, with a bell-shaped weight function

$$W(u, x) = \frac{1}{\Delta(u, x)^2 + m^2} \tag{23}$$

where  $m^2$  is a positive inertial term that prevents the overweighting of the closest source events. In our case  $m^2 = 2$ . Moreover, a maximum number of ten neighbors has been imposed. Nonetheless, the computation is CPU-intensive since the geodesic metric (3) involves slow trigonometric calculations. Limiting the causal neighborhood requires  $S_x$  to be ordered by distance, then trimmed retaining only the closest events to  $x$ ; this is

based on the heuristic observation that only the closest elements contribute significantly to the weight functions. Both Python and Java Timescape implementations allow this kind of trimming.

This example shows how the interpolation results are affected by the addition of new sampling sites. The dates covered are from 1975 to 1984 (included). This is of course a very crude model, based only on the GNIP  $\delta^2\text{H}$  observations, and further improvements could benefit from a regionalised regression based on climatic ancillary data [78,85]. An acknowledged source of Hydrogen isoscapes is the web-based application IsoMAP [86], which also provides insightful information about precipitation hydrogen and oxygen isoscapes in general [79].

### 3.5. Performance Analysis

The interpolation throughput has been estimated in terms of evaluated target events per second, and typical values range from a few tens to more than one thousand, depending on three factors: the number of source events  $|S|$ , the spatial interpolator (see the method column in Table 3, Kriging is slower than IDW), the distance function (geodesic is much slower than Euclidean), and the causal cone form factor (periodic is slower than straight). The cone shape and width also influence the number of null target events. Table 4 resumes the modeling performance of the Python Timescape version on three quite standard hw/sw configurations, not claiming exhaustiveness:

- configuration 1: laptop–Intel i7 @ 2.60 GHz, 4 cores, 16 GByte RAM, SSD storage, Python 3.8 on Windows 10 OS.
- configuration 2: laptop–Intel i7 @ 2.30 GHz, 4 cores, 8 GByte RAM, SSD storage, Python 3.7 on Ubuntu Linux 19.10 OS.
- configuration 3: virtual machine–Oracle Virtualbox on MacPro host, Intel Xeon-E5 @ 3.50 GHz–4 dedicated cores, 16 GByte reserved RAM out of 64, SSD storage, Python 3.8 on Xubuntu Linux 20.04-LTS OS.

**Table 4.** Model performances, measured in target events per second.  $\eta_{\text{model}}$  (17) is expressed in percent, running times in minute seconds.

Model	$\eta_{\text{model}}$	Configuration 1		Configuration 2		Configuration 3	
		Time	evt/s	Time	evt/s	Time	evt/s
Fungi $\delta^{15}\text{N}$	55%	87'32''	200	125'2''	140	75'33''	231
Min temperature	73%	115'49''	28	118'6''	27	78'20''	41
Olive oil $\delta^{18}\text{O}$	81%	39''	1315	47''	1091	35''	1478
Precipitation $\delta^2\text{H}$	62%	15'42''	157	16'4''	124	11'21''	176

The lowest throughput is associated with the temperature model (Section 3.2). In fact, this model associates a large source set, a periodic form factor, and a Kriging spatial interpolator, each factor adding its computational load.

A fair estimate of the python Timescape throughput can be set roughly about a few hundreds events per second with a standard hardware configuration. A conservative estimate of 100 evt/s is a good starting point for a new model.

We give no figures for the Java applications, since their time performance depends essentially on the database connection speed.

## 4. Discussion

The outcome of a Timescape model is based on three elements: the source set distribution, the spatiotemporal geometry (i.e., the shape of the causal cone and the parameters  $c$  and  $k$ ), and the chosen spatial interpolation method. Picking a spatial interpolator is a matter of personal taste and computational performance; it is worth stressing that the Timescape distance is independent of (logically, it precedes the) spatial interpolation itself,

so two models based on the same source with the same topology but different interpolators should not be much different to two ordinary spatial models.

As for the scale of modeling, using geographical coordinates adds further complexity due to the geodesic distance (3). If it is possible, i.e. the patches must be statistically “rich enough” and the Timescape distance parameters have to be the same across the patches (a necessary hypothesis is that far apart patches are independent of each other), it is advisable to divide the whole target set into patches which are tractable with projected coordinates, then merge the results, as could be done with any ordinary spatial interpolator (Geometrically, one builds an atlas of compatible charts. Special care is required since the spherical coordinates are periodic and singular at the poles).

We can explore further topological diversions modifying the structure of distance; Section 4.2 introduces a few suggestions. Altering the definition of the causal cones has a deep impact on causality, so the results can be unexpected even if the mathematical procedure itself is sound. A possible scenario could be inferring the past from the future (properly, retrodiction), exchanging the role of past end future cones  $\mathcal{C}_x^-$  and  $\mathcal{C}_x^+$ .

#### 4.1. Measurements, Stationarity, Accuracy

As it is the case with any spatial and spatiotemporal modeling tool, the statistical distribution of the input dataset is of the utmost importance for a reliable modeling strategy. Non-stationary data are often encountered in ecological modeling, as well as spatially uneven and skew-distributed measurements. The Timescape distance can help to mitigate these problems since the topological filtering operated by (7) and (9), reducing the number of input data events for each target output event, in some cases has a stabilizing effect on the variances. This is not a general rule, though: every dataset behaves distinctly. Seasonality, in particular, is well-managed by (9) the Xmas tree causal cone, since removing the off-season events from the past causal cones stabilizes the mean value and reduces the variance (see Figure 5).

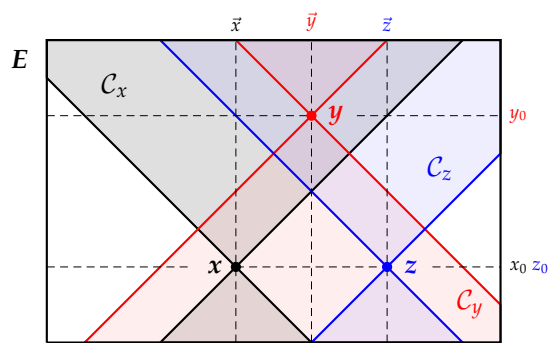
Accuracy assessment is the most delicate phase of Timescape modeling. On the one hand, statistical interpolators provide an accuracy figure  $\hat{\sigma}(x)$  along with the estimated field  $\hat{\Phi}(x)$  at running time, but it is also possible to apply jackknifing (16) on the finished target for every interpolator. The Python Timescape module [31] offers a set of functions (see Figure 9) for an initial guess of the model’s accuracy.

#### 4.2. Timescape Topology Modification

The distance (7) can be modified in a more general way than just applying a form factor that is only a function of time (9). As a matter of principle, the past causal cone  $\mathcal{C}_x^-$  of  $x$  can have any shape, provided that it is a subset of the past of  $x$ :  $\mathcal{C}_x^- \subseteq E_x^-$ , where  $E_x^- = \{y \in E \mid y_0 \leq x_0\}$ —see Figure 2. As a limiting case,  $\mathcal{C}_x^-$  could be the whole  $x$ ’s past  $E_x^-$ . Any such cone shape would retain the property  $\Delta(x, y) < \infty \implies \Delta(y, x) = \infty$  but not necessarily  $y \in \mathcal{C}_x^- \iff x \in \mathcal{C}_y^+$ .

Another modification of (7) consists of including retrodiction; just glue  $\mathcal{C}_x^-$  and  $\mathcal{C}_x^+$  by the common tip  $x$  as a double cone  $\mathcal{C}_x$ . This restores symmetry in the metric function ( $x \in \mathcal{C}_y \iff y \in \mathcal{C}_x$ ), but at the price of subadditivity, that only holds if, given  $x, y, z \in E$ , the three events are mutually causally connected, i.e.,  $x \in \mathcal{C}_y, y \in \mathcal{C}_z$  and  $z \in \mathcal{C}_x$  (the converse is guaranteed by symmetry). Otherwise symmetry is violated, as clarified in Figure 20. Furthermore, allowing retrodiction requires a factor one-half in front of variogram (13) to prevent double-counting.

Giving up causality altogether, the topological filtering can still be performed, with any  $x$ -neighborhood in lieu of  $\mathcal{C}_x$  double cone, but it would hardly be of any use in ecological modeling. It is worth mentioning that taking the whole  $E$  as  $\mathcal{C}_x$  for all  $x$ , though respecting both symmetry and subadditivity just means performing a standard spatial interpolation in one more dimension, with no topological filtering at all: this is not a Timescape model anymore.



**Figure 20.** Symmetric double causal cones. Note that  $\Delta(x, y) < \infty$  and  $\Delta(y, z) < \infty$ , nonetheless  $\Delta(x, z) = \infty$  because  $x_0 = z_0 \wedge \bar{x} \neq \bar{z} \implies x \notin C_z$ .

Other than geometrically, the distance (7) can be modified according to a set of external factors, i.e., exploiting the knowledge of one or more ancillary variables. For example, the form factor-modified time distance  $\tilde{D}_t$  (8) could be a function of some weather-related variables other than the time

$$\tilde{D}_t(x, y) = c |x_0 - y_0| \psi(x_0, y_0) \Psi(a_1(x) \dots a_n(x), a_1(y) \dots a_n(y)) \tag{24}$$

where the  $a_k$  are the observations of ancillary variables at the  $x$  and  $y$  events. Note that  $\psi(x_0, y_0)$  is more general than  $\psi(|x_0 - y_0|)$  in (8).

The Java applications [32,33] allow the mentioned modifications of distance function and the inclusion of ancillary variables, via a scripting language. The Python module [31] can only use symmetrical cones in the event’s past, allowing at most  $C_x^- = E_x^-$  as a limiting case, letting  $k \rightarrow \infty$ . Practically, a large enough  $k$  suffices, say  $k = \max D_s(u, x) / \min D_t(u, x)$ , where  $u \in S, x \in T$  belong to the source and target sets, respectively.

### 4.3. Timescape vs. Minkowskian Geometry

Since the seminal work of Minkowski [87], the causal structure of relativistic space–time has been encoded in the so-called light cone. Causally connected events lie within each other’s light cone, while no signal can reach two events outside the cones. Furthermore, the speed of light  $c$  acts as the obvious conversion factor between spatial and temporal coordinates. Therefore, the Minkowskian space–time is naturally equipped with a topological structure that incorporates causality straightforwardly.

The Timescape geometry purposely introduces a light cone like structure in the ordinary, non-relativistic, space–time. The role of the speed of light is shared between our  $c$  and  $k$  parameters, and the causal connection has to be introduced somehow forcefully, but the resulting topology is almost identical.

As far as topology is concerned, the Timescape and Minkowskian spaces act similarly, but the metric behavior is completely different. Using our symbols, the Timescape  $D_T$  and Minkowskian  $D_M$  squared distances are

$$\begin{aligned} D_T^2(x, y) &= D_t^2(x, y) + D_s^2(x, y) \\ D_M^2(x, y) &= D_t^2(x, y) - D_s^2(x, y) \end{aligned} \tag{25}$$

the minus sign of  $D_M$  in (25) encodes the light cone structure (The sign of  $D_M^2(x, y)$  defines, for each event  $x$ , three different space–time regions: the so-called time-like one, corresponding to the interior of our causal cones, the space-like region of causally unreachable events, and their common boundary, the locus of light-like events  $y$  whose  $D_M(x, y) = 0$ , reserved for light (or other massless particles). This is not the kind of space one expects to find in ecological modeling:  $D_T(x, y) = 0$  means that  $x$  and  $y$  coincide in space and time, unlike  $D_M(x, y) = 0$  that includes all the light-like events’ pairs. Furthermore,  $D_M$  is not even a distance since  $D_M^2$  is not positive definite and, even when  $D_M^2 \geq 0$ , it does not behave as a

distance; on the boundary of the light cone  $D_t = D_s \implies D_M = 0$ , which perfectly describes the light in the relativistic arena, but it is far from what an ordinary distance is expected to do. On the other hand, the plus of  $D_T$  in (25) ensures subadditivity, i.e.,  $D_T$  restricted to the causal cone is a proper distance. In fact, it is just a Pythagorean sum of squared sides (Figure 3). Summarizing, our Timescape distance mimics the Minkowskian topological structure, not its metric. The trick is operated by the  $\theta(kD_t - D_s)$  in the denominator of (7) that makes  $\Delta$  singular. The topological correspondence being: Minkowski space-like interval  $\rightarrow$  Timescape  $\Delta = \infty$ , time-like  $\rightarrow \Delta < \infty$ , light-like  $\rightarrow \Delta < \infty$  also. There is nothing special about the events laying on the boundary of the causal cones. Coping with infinite values is not a problem with most modern structured programming language, however, if singular distances were a concern, one could define a proximity measure  $\Delta^{-1}(x, y)$  from  $x$  to  $y$  inverting (7) as

$$\Delta^{-1}(x, y) = \frac{\theta[kD_t(x, y) - D_s(x, y)]}{\sqrt{D_t^2(x, y) + D_s^2(x, y)}} \tag{26}$$

the denominator of (26) never vanishes since, given two events  $x \neq y$ , at least  $D_t(x, y) > 0$  or  $D_s(x, y) > 0$ . This trick is based on the fact that spatial interpolators are defined or can be rephrased in terms of inverse distances. The published Timescape implementations use the distance (7), since both Python and Java cope well with infinities.

#### 4.4. A Universal Tool?

The Timescape algorithm can be exploited, in principle, with any set of observed variables varying in space and time. The Timescape algorithm should not be intended as a universal tool, however. In Table 5, we sketch a decision tree to help decide whether it should be chosen or not, depending on the distribution of input observations. In particular, it has to not be chosen whenever there is a clear temporal or spatial pattern in the source data that allows the disentanglement of spatial and temporal variability by subsetting the input set—this is the case, for example, for the Olive oil  $\delta^{18}\text{O}$  case study, Section 3.3.

**Table 5.** Observations-driven decision tree. ODE/PDE = Ordinary/Partial Differential Equations.

Field Dependence	Timescape	Alternatives	References
temporal only	useless	time series analysis ODE and PDE	[2,3]
prev. temporal	sharp cones $k \rightarrow 0$	dynamical systems machine learning	[16,17,88] [89]
entangled S + T	optimal suitability	stochastic PDE covariance-based Bayesian modeling neural networks	[17,18] [12] [21] [19,20]
prev. spatial	broad cones $k \rightarrow \infty$	gen. lin. models regression trees	[22] [23]
spatial only	useless	geostatistics	[4,5]

The subject of spatiotemporal modeling offers some interesting alternatives to our proposed algorithm. Table 5 is a decision tree, showing when we suggest applying the Timescape algorithm, along with some possible alternatives. The decision should be based not only on the actual measurements, but also on the very nature of the scalar field being interpolated. A few references are suggested. Figure 21 synthetically resumes the decision tree detailed in Table 5. The list of alternatives is far from complete and, of course, there is a certain grade of arbitrariness about how entangled the spatial and temporal field dependencies are.

In the following, we describe the cases singled out in Table 5. The list of alternative methods is far from exhaustive. The use cases are classified following the kind of spatial and temporal dependence of the modeled field [25].

**Purely temporal:** when the input dataset can be clearly partitioned as a bunch of time series, the algorithm can be applied, but it is unnecessarily complicated and time-consuming; in this case, the Timescape result is at best comparable with ordinary one-dimensional time series [2,3]. ODE techniques can also be considered [16].

**Prevailing temporal:** the Timescape algorithm can be advantageously applied. The minimum temperatures example is an example of this kind of distribution. In such cases, it is advisable to use sharp (i.e., very selective, small  $k$ ) causal cones to dampen the influence of nearby points. Many alternative tools exist from the field of dynamical systems as exact and stochastic differential equations [16,17,88].

**Entangled spatial and temporal:** this is Timescape's ideal playground; the fungi  $\delta^{15}\text{N}$  is a typical case of equal importance of time and space dependence of the measured values. The Timescape parameters must be carefully tuned (Section 2.4) to optimize the results. Alternative tools range from stochastic fields [17] to Bayesian modeling [21] and covariance-based methods [12,26]. The choice is often a matter of personal taste and confidence with the selected tool. It is also worth mentioning machine learning [50,89] and neural networks [19,20], from the field of Artificial Intelligence. In order to select the best performing tool, one should ideally have a control set against which they check the results [53]. Unfortunately, this is not always the case with sub-optimal datasets, not to mention non-stationarity, heteroskedasticity, etc. To deal with such datasets, the Timescape accuracy (17) can be estimated from the input set itself. Other methods may offer different options [43].

**Prevailing spatial:** in this case, the Timescape algorithm can advantageously be applied. The  $\delta^2\text{H}$  case study is an example of the kind. In such cases, it is advisable to use broad (i.e., less selective, large  $k$ ) causal cones to promote the influence of nearby points. Broad causal cones can easily increase the computational load. Useful alternative tools include generalized linear models [22] and boosted regression trees [23]. Sometimes ordinary geostatistics can do a good job, including the time as a third spatial coordinate, without causal constraints [5].

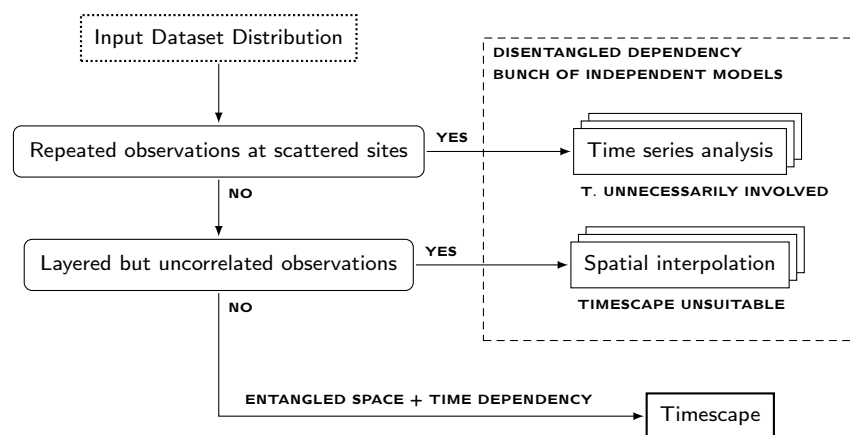
**Purely spatial:** if the measurements do not have an explicit time dependence, the Timescape algorithm is useless, of course. When the input data consists of repeated but uncorrelated measurements at different times (as the  $\delta^{18}\text{O}$  example demonstrates), the Timescape algorithm is even worse than a purely spatial model. Instead, each set of equal-time measurements should be treated independently [4,5].

In a broad sense, Artificial Intelligence techniques can be suitable alternatives in ecological spatiotemporal modeling as well as in many other fields of research [88,89], especially when variables interaction is to be considered; the Timescape algorithm, as it is implemented as of now [31–33], cannot cope with multiple variables, but its very core, the Timescape distance (7), can be exploited in a more general multi-variable context.

Summarizing, the Timescape algorithm is not a universal tool, but it has a remarkably wide range of applicability—the more entangled the space–time dependence, the better (Figure 21). Furthermore, the topological filtering (Section 2.2) operated by (7) can easily accommodate the periodic behavior of the modeled fields.

Binary variables and counts, although not strictly real fields, can be treated with the workarounds described in Section 2.6 as subsets of real random variables. The input values are simply treated as real.





**Figure 21.** An input-driven decision tree. The extreme cases (purely spatial and purely temporal dependencies, see Table 5) suggest that the Timescape algorithm is either unsuitable or far too involved, and the entangled cases can be advantageously treated.

## 5. Conclusions

The Timescape algorithm is a modeling tool aimed at the interpolation of spatiotemporal field distributions from recorded observations presenting an entangled variability in space and time. The spatial and temporal records of the modeled field are embedded into a space–time inspired by the causal topology of the Minkowskian space, the well-known arena of relativistic physics. This is achieved by applying the distance Function (7) for filtering the causally connected events before submitting the actual field estimate to an ordinary spatial interpolator. The choice of the interpolator is up to the user, as well as a couple of parameters that tune the distance to fit the particular input data. Seasonality, often encountered in ecological modeling, can be dealt with by modifying the distance Function (7) by means of a form factor (9).

The Timescape algorithm has been already implemented as a Python module and as two Java stand-alone GUI-based applications. The software is distributed under the GNU-GPL open license. The software is meant to be used with georeferenced input data and can be included in a GIS workflow, eventually exporting geotiff files.

Our modeling tool can cope with seasonality by playing with the shape of the causal cone without subsetting the input set season-by-season. Moreover, the algorithm can be applied with a variety of interpolation methods. Ideally, any spatial interpolator would do, the choice being driven by the input data and the user’s taste.

**Author Contributions:** Conceptualization M.C., F.C., G.R. and M.L.; Methodology M.C. and F.C.; Formal analysis M.C., F.C. and R.P.; Data curation M.C., F.C., R.P. and G.R., Visualization M.C. and G.R.; Software development M.C. and F.C.; Software testing R.P. and G.R.; Funding acquisition M.L. and M.C.; Manuscript curation M.C., F.C., R.P., G.R. and M.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Italian National Research Council grants IBAF003/2013/POR, IBAF/02/2017/TR, CNR STM-0051519 and the European grant COST STSM-TD1209-17435. The authors are also in debt for the fundamental scientific orientation gained within REALMed project, funded by ARIMNet2 (2017–2020), EU-FP 7th grant agreement no. 618127.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Timescape is published under the GNU-GPL3 license. The source code and the case studies’ data are available to download. The Python package along with the sample data is available at [31] and the Java implementations are available at [32,33].

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Evans, M.R. Modelling ecological systems in a changing world. *Philos. Trans. R. Soc. B Biol. Sci.* **2012**, *367*, 181–190. [[CrossRef](#)] [[PubMed](#)]
2. Beran, J. *Mathematical Foundations of Time Series Analysis*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017. [[CrossRef](#)]
3. Shumway, R.H.; Stoffer, D.S. *Time Series Analysis and Its Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017. [[CrossRef](#)]
4. Cressie, N. *Statistics for Spatial Data*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1993. [[CrossRef](#)]
5. Chilès, J.P.; Delfiner, P. *Geostatistics*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012. [[CrossRef](#)]
6. Delicado, P.; Giraldo, R.; Comas, C.; Mateu, J. Statistics for spatial functional data: Some recent contributions. *Environmetrics* **2009**, *21*, 224–239. [[CrossRef](#)]
7. Cressie, N.; Wikle, C.K. *Statistics for Spatio-Temporal Data*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2015.
8. Wikle, C.K.; Zammit-Mangion, A.; Cressie, N. *Spatio-Temporal Statistics with R*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2019.
9. Kyriakidis, P.C.; Journel, A.G. Geostatistical space-time models: A review. *Math. Geol.* **1999**, *31*, 651–684. [[CrossRef](#)]
10. Le, N.D.; Zidek, J.V. *Statistical Analysis of Environmental Space-Time Processes*; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2006.
11. Epperson, B.K. Spatial and space-time correlations in ecological models. *Ecol. Model.* **2000**, *132*, 63–76. doi: 10.1016/S0304-3800(00)00305-7. [[CrossRef](#)]
12. Pebesma, E. Spacetime: Spatio-Temporal Data in R. *J. Stat. Softw.* **2012**, *51*, 1–30. [[CrossRef](#)]
13. Graeler, B.; Pebesma, E.; Heuvelink, G. Spatio-Temporal Interpolation using gstat. *R J.* **2016**, *8*, 204–218. [[CrossRef](#)]
14. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, 2018.
15. Chen, Q.; Han, R.; Ye, F.; Li, W. Spatio-temporal ecological models. *Ecol. Inform.* **2011**, *6*, 37–43. doi: 10.1016/j.ecoinf.2010.07.006. [[CrossRef](#)]
16. Holmes, E.E.; Lewis, M.A.; Banks, J.E.; Veit, R.R. Partial Differential Equations in Ecology: Spatial Interactions and Population Dynamics. *Ecology* **1994**, *75*, 17–29. [[CrossRef](#)]
17. Lindgren, F.; Rue, H.; Lindström, J. An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2011**, *73*, 423–498. [[CrossRef](#)]
18. Rue, H.; Lindgren, F.; van Niekerk, J.; Krainski, E.; Fattah, E.A. R-INLA: Approximate Bayesian Inference for Latent Gaussian Models. Available online: <https://www.r-inla.org> (accessed on 7 January 2022).
19. Christin, S.; Hervet, É.; Lecomte, N. Applications for deep learning in ecology. *Methods Ecol. Evol.* **2019**, *10*, 1632–1644. [[CrossRef](#)]
20. Brodrick, P.G.; Davies, A.B.; Asner, G.P. Uncovering Ecological Patterns with Convolutional Neural Networks. *Trends Ecol. Evol.* **2019**, *34*, 734–745. [[CrossRef](#)] [[PubMed](#)]
21. Knorr-Held, L. Bayesian modeling of inseparable space-time variation in disease risk. *Stat. Med.* **2000**, *19*, 2555–2567. doi: 10.1002/(SICI)1471-2325(200007)19:17<2555::AID-SM2555>3.0.CO;2-3. [[CrossRef](#)]
22. Bolker, B.M.; Brooks, M.E.; Clark, C.J.; Geange, S.W.; Poulsen, J.R.; Stevens, M.H.H.; White, J.S.S. Generalized linear mixed models: A practical guide for ecology and evolution. *Trends Ecol. Evol.* **2009**, *24*, 127–135. [[CrossRef](#)] [[PubMed](#)]
23. Elith, J.; Leathwick, J.R.; Hastie, T. A working guide to boosted regression trees. *J. Anim. Ecol.* **2008**, *77*, 802–813. [[CrossRef](#)]
24. Fritsch, M.; Lischke, H.; Meyer, K.M. Scaling methods in ecological modelling. *Methods Ecol. Evol.* **2020**, *11*, 1368–1378. [[CrossRef](#)]
25. Wang, J.F.; Stein, A.; Gao, B.B.; Ge, Y. A review of spatial sampling. *Spat. Stat.* **2012**, *2*, 1–14. [[CrossRef](#)]
26. Jun, M.; Stein, M.L. Nonstationary covariance models for global data. *Ann. Appl. Stat.* **2008**, *2*, 1271–1289. [[CrossRef](#)]
27. Frankel, T. *The Geometry of Physics*; Cambridge University Press: Cambridge, UK, 2009. [[CrossRef](#)]
28. Jost, J. *Geometry and Physics*; Springer: Berlin/Heidelberg, Germany, 2009. [[CrossRef](#)]
29. Ashtekar, A.; Petkov, V. (Eds.) *Springer Handbook of Spacetime*; Springer: Berlin/Heidelberg, Germany, 2014. [[CrossRef](#)]
30. Basseville, M. Distance measures for signal processing and pattern recognition. *Signal Process.* **1989**, *18*, 349–369. [[CrossRef](#)]
31. Ciolfi, M. Timescape Python Module, GNU-GPL 3 License. Available online: <https://github.com/ciolfis/timescape> (accessed on 1 December 2021).
32. Ciolfi, M. TimescapeLocal Java Application, GNU-GPL 3 License. Available online: <https://sourceforge.net/projects/timescapelocal> (accessed on 1 December 2021).
33. Ciolfi, M. TimescapeGlobal Java Application, GNU-GPL 3 License. Available online: <https://sourceforge.net/projects/timescapeglobal> (accessed on 1 December 2021).
34. Jost, J. *Riemannian Geometry and Geometric Analysis*; Springer: Berlin/Heidelberg, Germany, 2011. [[CrossRef](#)]
35. Banerjee, S. On Geodesic Distance Computations in Spatial Modeling. *Biometrics* **2005**, *61*, 617–625. [[CrossRef](#)]
36. Mamajek, E.E.; Prsa, A.; Torres, G.; Harmanec, P.; Asplund, M.; Bennett, P.D.; Capitaine, N.; Christensen-Dalsgaard, J.; Depagne, E.; Folkner, W.M.; et al. IAU 2015 Resolution B3 on Recommended Nominal Conversion Constants for Selected Solar and Planetary Properties. *arXiv* **2015**, arXiv:1510.07674.
37. Ambartzumian, R.V. A note on pseudo-metrics on the plane. *Z. Wahrscheinlichkeitstheorie Verwandte Geb.* **1976**, *37*, 145–155. [[CrossRef](#)]
38. Schroeder, V. Quasi-metric and metric spaces. *Conform. Geom. Dyn. Am. Math. Soc.* **2006**, *10*, 355–361. [[CrossRef](#)]

39. Scott, D.W. On optimal and data-based histograms. *Biometrika* **1979**, *66*, 605–610. [CrossRef]
40. Efron, B.; Stein, C. The Jackknife Estimate of Variance. *Ann. Stat.* **1981**, *9*, 586–596. [CrossRef]
41. Knight, K. *Mathematical Statistics*; Chapman & Hall CRC: Boca Raton, FL, USA, 2000.
42. McIntosh, A. The Jackknife Estimation Method. *arXiv* **2016**, arXiv:1606.00497.
43. Bennett, N.D.; Croke, B.F.; Guariso, G.; Guillaume, J.H.; Hamilton, S.H.; Jakeman, A.J.; Marsili-Libelli, S.; Newham, L.T.; Norton, J.P.; Perrin, C.; et al. Characterising performance of environmental models. *Environ. Model. Softw.* **2013**, *40*, 1–20. [CrossRef]
44. PyKriging Developers. PyKriging, 2D and 3D Python Kriging Implementation. Available online: <https://pykriging.readthedocs.io> (accessed on 1 December 2021).
45. Open Geospatial Consortium. OGC GeoTIFF Standard. Available online: <https://www.openeospatial.org/standards/geotiff> (accessed on 1 December 2021).
46. Ciolfi, M.; Chiocchini, F.; Mattioni, M.; Lauteri, M. Timescape Local spacetime interpolation tool: Projected coordinates java standalone application. *Smart ELab* **2017**, *10*, 20–39. [CrossRef]
47. Ciolfi, M.; Chiocchini, F.; Mattioni, M.; Lauteri, M. Timescape Global spacetime interpolation tool: Geographic coordinates java standalone application. *Smart ELab* **2018**, *11*, 1–51. [CrossRef]
48. Hibernate Object Relational Mapping. Available online: <http://hibernate.org> (accessed on 1 December 2021).
49. Oracle. MySQL Developer Zone. Available online: <https://dev.mysql.com> (accessed on 1 December 2021).
50. Gallant, S. Perceptron-based learning algorithms. *IEEE Trans. Neural Netw.* **1990**, *1*, 179–191. [CrossRef]
51. West, J.B.; Bowen, G.J.; Dawson, T.E.; Tu, K.P. (Eds.) *Isoscapes*; Springer: Dordrecht, The Netherlands, 2010. [CrossRef]
52. Wagner, H.H.; Fortin, M.J. Spatial Analysis of Landscapes: Concepts and Statistics. *Ecology* **2005**, *86*, 1975–1987. [CrossRef]
53. Rossi, R.E.; Mulla, D.J.; Journel, A.G.; Franz, E.H. Geostatistical Tools for Modeling and Interpreting Ecological Spatial Dependence. *Ecol. Monogr.* **1992**, *62*, 277–314. [CrossRef]
54. West, J.B.; Bowen, G.J.; Cerling, T.E.; Ehleringer, J.R. Stable isotopes as one of nature’s ecological recorders. *Trends Ecol. Evol.* **2006**, *21*, 408–414. [CrossRef] [PubMed]
55. Bowen, G.J.; Good, S.P. Incorporating water isoscapes in hydrological and water resource investigations. *WIREs Water* **2015**, *2*, 107–119. [CrossRef]
56. West, J.B.; Sobek, A.; Ehleringer, J.R. A Simplified GIS Approach to Modeling Global Leaf Water Isoscapes. *PLoS ONE* **2008**, *3*, e2447. [CrossRef] [PubMed]
57. Brugnoli, E.; Farquhar, G.D. Photosynthetic Fractionation of Carbon Isotopes. In *Photosynthesis*; Springer: Dordrecht, The Netherlands, 2000; pp. 399–434. [CrossRef]
58. Hobson, K.A.; Wassenaar L.I. (Eds.). *Tracking Animal Migration with Stable Isotopes*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2019; ISBN 9780128147238. [CrossRef]
59. Chesson, L.A.; Tipple, B.J.; Ehleringer, J.R.; Park, T.; Bartelink, E.J. Forensic applications of isotope landscapes (“isoscapes”). In *Forensic Anthropology*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2017; pp. 127–148. [CrossRef]
60. Bivand, R.S.; Pebesma, E.; Gomez-Rubio, V. *Applied Spatial Data Analysis with R*; Springer, New York, NY, USA, 2013. [CrossRef]
61. Pebesma, E.; Bivand, R. Classes and Methods for Spatial Data in R. *R News* **2005**, *5*, 1–20.
62. Ciolfi, M.; Piovesan, G. Spatiotemporal Analysis and Modeling of Ecological Processes at Ecosystem, Landscape and Bioregion Scale. Ph.D. Thesis, Università della Tuscia, Viterbo, Italy, 2016.
63. Hogberg, P.; Hogberg, M.N.; Quist, M.E.; Ekblad, A.; Nasholm, T. Nitrogen isotope fractionation during nitrogen uptake by ectomycorrhizal and non-mycorrhizal *Pinus sylvestris*. *New Phytol.* **1999**, *142*, 569–576. [CrossRef]
64. Hobbie, E.A.; Colpaert, J.V. Nitrogen availability and colonization by mycorrhizal fungi correlate with nitrogen isotope patterns in plants. *New Phytol.* **2003**, *157*, 115–126. [CrossRef]
65. Dawson, T.E.; Mambelli, S.; Plamboeck, A.H.; Templer, P.H.; Tu, K.P. Stable Isotopes in Plant Ecology. *Annu. Rev. Ecol. Syst.* **2002**, *33*, 507–559. [CrossRef]
66. Henn, M.R.; Chapela, I.H. Ecophysiology of <sup>13</sup>C and <sup>15</sup>N isotopic fractionation in forest fungi and the roots of the saprotrophic-mycorrhizal divide. *Oecologia* **2001**, *128*, 480–487. [CrossRef]
67. Govindarajulu, M.; Pfeffer, P.E.; Jin, H.; Abubaker, J.; Douds, D.D.; Allen, J.W.; Bücking, H.; Lammers, P.J.; Shachar-Hill, Y. Nitrogen transfer in the arbuscular mycorrhizal symbiosis. *Nature* **2005**, *435*, 819–823. [CrossRef]
68. Umbria Region Hydrographic Service. Available online: <https://servizioidrografico.regione.umbria.it> (accessed on 1 December 2021).
69. Annali Regione Umbria. Annals of the Umbria Hydrographic Service. Available online: <https://annali.regione.umbria.it> (accessed on 1 December 2021).
70. Natural Earth Public Domain Vector and Raster Maps. Available online: <https://www.naturalearthdata.com> (accessed on 1 December 2021).
71. Geoportale Nazionale of the Italian Ministry for the Environment. Available online: <http://www.pcn.minambiente.it> (accessed on 1 December 2021).
72. QGIS Development Team. QGIS Geographic Information System. Open Source Geospatial Foundation Project. Available online: <http://qgis.osgeo.org> (accessed on 1 December 2021).
73. Giraldo, R.; Delicado, P.; Mateu, J. Continuous Time-Varying Kriging for Spatial Prediction of Functional Data: An Environmental Application. *J. Agric. Biol. Environ. Stat.* **2010**, *15*, 66–82. [CrossRef]

74. Chiocchini, F.; Portarena, S.; Ciolfi, M.; Brugnoli, E.; Lauteri, M. Isoscapes of carbon and oxygen stable isotope compositions in tracing authenticity and geographical origin of Italian extra-virgin olive oils. *Food Chem.* **2016**, *202*, 291–301. [[CrossRef](#)] [[PubMed](#)]
75. Portarena, S.; Gavrichkova, O.; Lauteri, M.; Brugnoli, E. Authentication and traceability of Italian extra-virgin olive oils by means of stable isotopes techniques. *Food Chem.* **2014**, *164*, 12–16. [[CrossRef](#)] [[PubMed](#)]
76. Camin, F.; Larcher, R.; Nicolini, G.; Bontempo, L.; Bertoldi, D.; Perini, M.; Schlicht, C.; Schellenberg, A.; Thomas, F.; Heinrich, K.; et al. Isotopic and Elemental Data for Tracing the Origin of European Olive Oils. *J. Agric. Food Chem.* **2010**, *58*, 570–577. [[CrossRef](#)]
77. Longinelli, A.; Selmo, E. Isotopic composition of precipitation in Italy: A first overall map. *J. Hydrol.* **2003**, *270*, 75–88. [[CrossRef](#)]
78. *Isotopic Composition of Precipitation in the Mediterranean Basin in Relation to Air Circulation Patterns and Climate*; Number 1453 in TECDOC Series; International Atomic Energy Agency: Vienna, Austria, 2005.
79. Bowen, G.J.; Revenaugh, J. Interpolating the isotopic composition of modern meteoric precipitation. *Water Resour. Res.* **2003**, *39*, 1299. [[CrossRef](#)]
80. Camin, F.; Larcher, R.; Perini, M.; Bontempo, L.; Bertoldi, D.; Gagliano, G.; Nicolini, G.; Versini, G. Characterisation of authentic Italian extra-virgin olive oils by stable isotope ratios of C, O and H and mineral composition. *Food Chem.* **2010**, *118*, 901–909. [[CrossRef](#)]
81. Visual Data Tools Inc. DataGraph Scientific Graphing Tool for macOS. Available online: <https://www.visualdatatools.com/DataGraph/> (accessed on 7 January 2022).
82. The Global Network of Isotopes in Precipitation of the International Atomic Energy Agency. Available online: <https://www.iaea.org/services/networks/gnip> (accessed on 1 December 2021).
83. *Statistical Treatment of Data on Environmental Isotopes in Precipitation*; Number 331 in Technical Reports Series; International Atomic Energy Agency: Vienna, Austria, 1992.
84. IAEA. International Atomic Energy Agency. Available online: <https://www.iaea.org> (accessed on 1 December 2021).
85. Terzer, S.; Wassenaar, L.I.; Araguás-Araguás, L.J.; Aggarwal, P.K. Global isoscapes for  $\delta^{18}\text{O}$  and  $\delta^2\text{H}$  in precipitation: Improved prediction using regionalized climatic regression models. *Hydrol. Earth Syst. Sci.* **2013**, *17*, 4713–4728. [[CrossRef](#)]
86. IsoMAP—Isoscapes Modeling, Analysis and Prediction. Available online: <http://isomap.org> (accessed on 1 December 2021).
87. Minkowski, H. Das Relativitätsprinzip. *Ann. Phys.* **1915**, *352*, 927–938. [[CrossRef](#)]
88. Week, B.; Nuismer, S.L.; Harmon, L.J.; Krone, S.M. A white noise approach to evolutionary ecology. *J. Theor. Biol.* **2021**, *521*, 110660. [[CrossRef](#)]
89. Lucas, T.C.D. A translucent box: Interpretable machine learning in ecology. *Ecol. Monogr.* **2020**, *90*, e01422. [[CrossRef](#)]