SERDP

FINAL REPORT

Empirical Dynamics: A New Paradigm for Understanding and Managing Species and Ecosystems in a Non-Stationary Nonlinear World

George Sugihara
Ethan Deyle
University of California, San Diego

April 2022

SERDP Project RC-2509

DISTRIBUTION STATEMENT A
This document has been cleared for public release.

This report was prepared under contract to the Department of Defense Strategic Environmental Research and Development Program (SERDP). The publication of this report does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official policy or position of the Department of Defense. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the Department of Defense.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 15-04-2022	**2. REPORT TYPE** SERDP Final Report	**3. DATES COVERED** *(From - To)* 09-28-2015 - 09/28/2021

4. TITLE AND SUBTITLE Empirical Dynamics: A New Paradigm for Understanding and Managing	**5a. CONTRACT NUMBER** W912HQ-15-C-0058
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Sugihara, George, and Deyle, Ethan, R	**5d. PROJECT NUMBER** RC-2509
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of California, San Diego 9500 Gilman Dr. La Jolla, CA 92093	**8. PERFORMING ORGANIZATION REPORT NUMBER** RC-2509
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Strategic Environmental Research and Development Program Office of the Deputy Assistant Secretary of Defense (Environment & Energy Resilience) 3500 Defense Pentagon, RM 5C646 Washington, DC 20301-3500	**10. SPONSOR/MONITOR'S ACRONYM(S)** SERDP
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) RC-2509

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project was aimed to develop *empirical dynamic modeling* (EDM) as a practical framework for studying and managing ecosystems, specifically developing technical capacity to build mechanistic models that can confidently forecast environmental futures in a dynamically changing world. To this end, the project demonstrated real solutions for understanding environmental futures in two tactical case studies. In the first case, EDM analysis untangled causal drivers of harmful algal blooms in the Southern California bight, demonstrating short-term prediction capability where none had seemed possible. The second case study developed a predictive framework for conservation management of reef areas in the U.S. Pacific Islands. This required developing new EDM methods for cases where historical time series are short. Together they have grown the predictive data science methods to address non-stationary, non-analogue futures and serve as road maps for new practitioners and applications to understand.

15. SUBJECT TERMS

Empirical Dynamic Modeling, Environmental Data Science, Ecological Forecasting, Harmful Algal Blooms, Red Tides, Physical-Biological Coupling, Coral Reef Dynamics, Resilience

16. SECURITY CLASSIFICATION OF: **None – Available to Public**			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** George Sugihara
a. REPORT UNCLASS	**b. ABSTRACT** UNCLASS	**c. THIS PAGE** UNCLASS	**None – Available to Public**	157	**19b. TELEPHONE NUMBER** *(include area code)* 858-534-5582

Front Matter

Table of Contents

List of Tables

List of Figures

List of Acronyms

EDM- Empirical Dynamic Modeling.
mEDM- Multivariate Empirical Dynamic Model.
S-map- Sequentially Weighted Local Linear Map.
CCM- Convergent Cross Mapping.
AR- Autoregressive (Model).
LEV- Largest Eigenvalue (of the Jacobian Matrix).
EWS- Early Warning Sign (of Critical Transition).
SVD- Singular Value Decomposition (of a Matrix).
SIO- Scripps Institution of Oceanography.
NOAA- National Oceanic and Atmospheric Administration.
ROMS- Regional Ocean Simulations.
NWW3- Wave Watch III Model.
HAB- Harmful Algal Bloom.
SST- Sea Surface Temperature.
SST_{upper}- Upper Limit of Sea Suface Temperature Over 2-Year Survey Intervals.
Chl- Chlorophyll-a.
L_{ot}- Thorpe Length Scale (of Turbulent Oscillations).
M_{pcyn}- Slope of the Pycnocline (Density Change with Depth).
U_{wind}- Wind in the North-South Direction.
ρ_{surf}- Surface Density.
WEF- Wave Energy Flux.
DWEF- Directed Wave Energy Flux.
$DWEF_{max}$- Maximum Directed Wave Energy Flux Over 2-Year Survey Intervals.
DV-DWEF- Directional Variance of Directed Wave Energy Flux.
RAMP- Pacific Reef Assessment and Monitoring Program.
TDS- Towed Diver Survey.
%LC- Percent Benthic Cover of Live Coral.
%StC- Percent Benthic Cover of Stressed Coral (Bleached or Diseased).
%MA- Percent Benthic Cover of Macroalgae.
%S- Percent Benthic Cover of Sand.

Keywords

Empirical Dynamic Modeling, Nonlinear Time Series Analysis, Environmental Data Science, Ecological Forecasting, Harmful Algal Blooms, Red Tides, Physical-Biological Coupling, Coral Reef Dynamics, Resilience

Acknowledgements

We wish to thank all our collaborators through the various phases of this grant, especially past and present members of the Sugihara Lab at UCSD. We would also like to thank SERDP for giving us the opportunity to grow empirical dynamic modeling.

Abstract

Introduction and Objectives

This project was aimed to develop *empirical dynamic modeling* (EDM) as a practical framework for studying and managing ecosystems. A key challenge was to address the complex reality of ecosystems that are nonlinear, nonstationary and not in equilibrium – properties of real ecosystems that are not addressed by classical models (and that may explain their poor predictive ability). Our goal was to develop capacity to build mechanistic models that can be used to confidently forecast environmental futures in a dynamically changing world. Though many of the tactical objectives of the project are technically framed, together they provide tools that have practical utility to support DoD's environmental interests and SERDP's deployment of best-available science at the cutting edge.

Technical Approach

Empirical dynamic modelling is an inductive data-driven approach for studying complex systems from time series observations. While classical approaches involve hypothesized models, EDM attempts to minimize these assumptions by allowing the data to speak for itself. Thus, instead of assuming a particular model, EDM allows the data to tell us what the underlying model should look like. EDM is based on reconstructing an attractor from time series data (https://youtu.be/fevurdpiRYg). This allows us to identify and study causal variables and interactions that are nonlinear and state-dependent and make skillful out-of-sample predictions. The abilities developed here to forecast and to explore alternative scenarios (e.g. different future environmental constraints), and to evaluate state dependent risk in terms of uncertainty (e.g., local sensitivity to dramatic change driven by dynamic instability), are essential for ecosystem management – the stated goals of this project.

Results

Central among the new high-level insights brought to the fore is the fact that *causal drivers can be completely uncorrelated with their effects*. We find that uncorrelated variables (that are therefore invisible to normal data exploration) are ubiquitous in nature and can be key elements for understanding mechanisms and for predicting and managing environmental futures — an inconvenient fact that profoundly impacts our normal study protocols and model-building efforts. To this end, the project demonstrated real solutions for understanding environmental futures in two tactical case studies. In the first case, EDM allowed us to untangle causal drivers of red tides (potentially harmful algal blooms) in the Southern California bight and demonstrate short-term prediction capability where none had seemed possible. By allowing for non-equilibrium and nonlinear dynamics, EDM analysis confirmed hypothesized drivers like nitrate and ocean temperature had predictable effects even though traditional methods had previously found them to be uncorrelated with the algal blooms. Combined with regional ocean simulations (ROMS) of future ocean conditions, the hybrid ROMS-EDM model predicts an increase in red tides over the next 3 decades. Our second case study develops a predictive framework for conservation management of reef areas that continue to be long-term DoD responsibilities in the U.S. Pacific Islands. This required developing new EDM methods for cases where historical time series are short. These methods uncovered the major environmental drivers operating at the reef sites, which in turn can identify specific areas in the U.S. Pacific Island region where conservation investments are most likely to succeed.

Benefits

Both case studies are useful SERDP targets on their own, but together they have grown the methodology and serve as road maps for new practitioners and applications. Critically, the project developed a standardized set of computational tools in R, Python, and C++ with documentation to communicate the essential foundations for new users wishing to apply EDM to their research. The benefits of this work can be quantified by how highly subscribed the resources are. The rEDM package has more than 19,000 downloads from CRAN; and pyEDM has more than 76,729 downloads through the central PyPI server in the first year of its launch. Together the code and documentation are already having a transformative effect on multiple scientific domains with well over 100 new publications citing EDM in just the first few months of 2020. It seems likely that beyond the conservation benefits of our two specific case studies, EDM should be useful to future SERDP projects ranging from hydrology to fire ecology. It is a natural tool set for environmental science.

1 Executive Summary

1.1 Introduction

The strategic aim of this study was to develop "empirical dynamic modeling" (EDM) as a practical framework for studying and managing ecosystems. A key challenge was to address the complex reality of ecosystems that are nonlinear, nonstationary and not in equilibrium – important properties of real ecosystems that are not addressed by classical models and that may explain the poor predictive ability of classical ecological models (1, 2). Our goal was to develop the capacity to build mechanistic models that can be used to confidently forecast environmental futures in a dynamically changing world.

EDM involves constructing attractors directly from time series data — attractors are geometric shapes or manifolds that embody the rules (underlying equations) governing the ecosystem (see video 1 below). Thus, EDM is inductive and data-driven, and thereby quite different from classical deductive approaches involving models based on preconceived (hypothesized) equations. It allows the data to speak for itself.

Though many of the tactical objectives of the project are technically framed, together they provide tools that have practical utility to support DoD's environmental interests while also supporting SERDP's deployment of best-available science at the cutting edge. Indeed, the 2014 SERDP call that brought this project into being, to develop a predictive data-driven framework for addressing complex natural systems, presaged the 2019 national 21st century science initiative for *Harnessing the Data Revolution* (https://www.nsf.gov/cise/harnessingdata/). As NOAA Acting Administrator, Timothy Galludet remarked about EDM in 2018: "I am especially interested in novel modeling approaches that treat the system as it actually functions, not how it theoretically should (http://deepeco.ucsd.edu/nonlinear-dynamics-research/edm/)... the EDM work is just that sort of novel approach."

High-level Strategic Results
Central among the new insights that this project brought forward is the fact that *causal drivers can be completely uncorrelated with their effects*. This inconvenient fact (the converse of Bishop Berkeley's 1712 dictum) profoundly impacts our normal study protocols and model building efforts. In this project we find that such uncorrelated (invisible) variables are ubiquitous in nature, and that beyond raising consciousness about this fact, they can be key elements for predicting and managing environmental futures. To this end we have developed general analytical tools to acknowledge and exploit this important consequence of nonlinearity and that thereby can produce skillful predictive models where this was previously not possible (3, 4). More importantly these tools can be used more generally for attribution: to measure causes to assign responsibility for environmental impacts.

Another key high-level insight is the discovery that species interactions and other biological relationships (e.g. nutrient uptake rates) are not fixed constants as classical models or statistical analysis would assume, but are non-stationary in time and even highly episodic (5). This insight has deep theoretical implications for understanding how ecological systems are structured, but also has major practical implications (developed through this project) for how ecosystems should be studied and modeled for prediction and management (not with constant coefficients), especially how to think about risk, resilience (6), and uncertainty (7).

Practical Applied Results and Benefits
At the more tactical level, the project provides practical solutions for understanding environmental futures in two case studies of DoD significance. In the first case, EDM allowed us to uncover uncorrelated causal drivers that can be used to forecast red tides in the Southern California bight — because of the lack of correlation the red tides in Southern California eluded prediction for over a century. This mechanistic predictive model, combined with regional ocean models (ROM), *predicts an increase in red tides over the next 3 decades*. This is significant as red

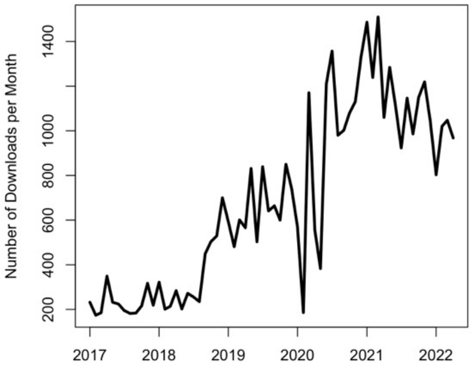
tides in Southern California have interfered with ship operation and coastal industrial operations, and toxic blooms (e.g. by the Domoic Acid producing Pseudo-nitzschia) have caused mortality and reproductive failure in marine mammal and avian conservation targets (e.g. the California Brown Pelican). Going forward, near-term forecasts could also assist the scheduling of marine exercises at Camp Pendleton.

Our second case study developed a predictive framework for conservation management of reef areas that continue to be long-term DoD responsibilities in the U.S. Pacific Islands. As we describe below, new methods were developed to apply to cases such as this one where historical time series are short. These methods were then applied to discover the major environmental drivers operating at the reef sites, which in turn allowed us to *identify areas and circumstances where conservation investments are most likely to succeed*. Both of these case studies are useful SERDP targets on their own, but together they have grown the methodology, expanding the types of systems and data EDM can be applied to, and serve as road maps for new practitioners and for applications in environmental management.

Building on the methods used to study red tides, an additional study introduced new environmental management tools based on a hybrid of simple physical models and EDM. Using the apparent irreversibility of eutrophication in iconic Lake Geneva as a high profile example (in press at PNAS) – a nonstationary, nonequilibrium system without future analogues – the hybrid approach not only leads to substantially better prediction, but also to a more actionable description of the emergent rates and processes (biogeochemical, ecological etc.) that drive water quality. Notably, the hybrid model suggests that the future impact of a moderate 3°C increase in air temperature of would be on the same order as the eutrophication of the previous century, and that best management action may no longer involve a single control lever such as reducing phosphorus inputs alone. The Lake Geneva study was a nice demonstration of the portability of the methods and their ability to address a central challenge for 21st century environmental management – forecasting nonlinear, non-stationary, non-analogue futures.

Strategic Benefits (confirmed impact)

Critically, and likely one of the most broadly significant consequences of this award, is the development of a standardized set of applied computational tools for EDM. Code packages are now publicly available in R, Python, and C++, and are quickly finding a large user base. Since initial launch ca. January 2017, the **rEDM** package has had more than **43,735 downloads** (1047 just in March 2022) from the central CRAN server (see figure below).



Redm monthly downloads

More remarkably, **pyEDM** through the central PyPI server, has had more than **331,366 downloads** since launch in August 2019 (~20,433 downloads just in March 2022). See https://pepy.tech/project/pyEDM for current download statistics. Python has a broader scientific audience than R.

Simultaneously, along with 32 peer-reviewed publications (nearly all in first-tier journals), the award allowed us to develop documentation to communicate the essential foundations for new users wishing to apply EDM to their research. Together the code and documentation are already having a transformative effect on multiple scientific domains. In the first few months of 2022, there are over 100 new publications using convergent cross mapping across ecology and diverse other fields including meteorology, social sciences, and finance. In fact, a data visualization tool **visEDM** (https://youtu.be/GLTB8d-vexc) is soon to become available and a **Stata** package ofEDM tools based on rEDM is now available for social scientists and financial engineers (https://ideas.repec.org/c/boc/bocode/s458593.html) along with a tutorial for its use (Li, Zyphur, Sugihara 2021

It is our hope that EDM will be a useful method in the toolbox of other future SERDP projects, not just the two case studies of marine environments focused on here. Conversations with SERDP researchers along with past SERDP review board members and former SERDP Program Manager, John A. Hall have identified critical needs that EDM can fill: ranging from forecasting/understanding red tides at Camp Lejeune (analogous to the SoCal study), to hydrology to fire ecology.

1.2 Objectives

The core strategic goal of the project was developing EDM to address the need for credible ecological forecasting to understand past, present, and future in non-stationary, non-equilibrium systems, and to raise consciousness of the particular complications that arise for predicting futures in such complex natural systems. This was accomplished using two case studies (described below) as exemplars with focused relevance to the SERDP mission, and an additional high-profile study to promote broader impact for the approach. In order to meet the objectives of these studies it was necessary to develop methods that broaden the scope of EDM (our global objective), thus making the technology more relevant and applicable to a wider range of ecological systems (most notably, those that lack long time-series and that would require long term monitoring programs).

Thus, our basic work scheme was to create a dialogue between specific real-world applications and general theory-methods development, and to use the case studies to: 1) produce practical advice relevant for each system, and 2) to use the strengths and constraints of the data in each study (eg. multiple long time series for multiple variables in the red tide study, and the short but spatially numerous time series of coral reefs) to catalyze new methods and insights.

1.2.1 Tactical Case Studies

1.2.1.1 Red Tides in Southern California

The immediate objective of the first case study was to develop an empirical dynamic model to forecast red tide events in Southern California — a case with substantial long-term monitoring of many potentially relevant variables. The occurrence of red tides in Southern California was an unsolved problem for over 100 years because drivers of this phenomenon were invisible to correlation-based studies. A key sub-objective was to analyze the multiple data series to try to uncover the suite of causal environmental drivers (variables that should be included in a model but that were uncorrelated with red tides). This would then allow us to construct EDM models for the short-term prediction of specific events days or weeks in advance (our first objective). This case study was a first large-scale demonstration of EDM where linear methods have failed to give insight. As an important check, the EDM models (constructed with mechanistic variables) must be validated by their ability to make skillful out-of-sample forecasts.

While short-term predictions address short-term operations decisions (e.g. training exercises), long-term prediction can help address long term planning of operations and conservation in future environments (where conditions might promote more frequent red tides). Thus, another major objective of this case study was to translate short-term predictive modeling of events into a long-term understanding of bloom behavior, to try to predict the future frequency of red tides 30 years from now. Here regional ocean models (ROMs) for the Southern California Bight are used to provide inputs to the mechanistic EDM models that have been previously validated by their ability to make out of sample predictions.

1.2.1.2 Pacific Coral Reef Conservation

The immediate objective of the second case study was to create a baseline empirical dynamic model for the benthic community dynamics of reef communities in the U.S. Pacific Islands — a case involving hundreds of islands having abundant line transect data collected over a relatively brief interval (short time series). The main applied objective was to demonstrate how an EDM framework can be constructed to identify practical benchmarks of reef status and address actions required to evaluate and restore past or future environmental impacts on islands. This includes predicting coral loss, growth, and stress as well as characterizing sensitivity to environmental drivers.

Thus, in addition to providing practical advice for a specific system of DoD interest, this case study furthers the strategic objective of expanding the practical scope of EDM by focusing on a system without traditional long time-series measurements but with high spatial power instead.

1.2.2 Expanding the scope of EDM methodology

1.2.2.1 Multivariate Data Leverage

While complexity and multi-dimensionality can be barriers to traditional analysis, we have shown that they also open up new ways of leveraging observations — allowing different sets of observations to effectively provide different kaleidoscopic views of the system. Thus, we aimed to expand EDM theory to develop ensemble models for improved forecasting and prediction both short-term, and for long-term scenarios, including multiview embedding (3) and refining ideas on random projection theory (8).

1.2.2.2 High Spatial Power Data Leverage

The core of empirical dynamic modeling was built on studies that involved long-term ecological observations. However, extensive longitudinal data may not be practical for many management applications. Indeed, it is much more common to encounter detailed but brief cross-sectional data from the intensive monitoring of many sites and species over short time periods (as in the reef island study). This motivates the objective here to extend EDM methods to accommodate high-spatial, low-temporal power data series common in ecosystem study, to make EDM more practical for treating emerging management needs.

1.2.2.3 Long-term Simulation for Multiple Plausible Futures

Empirical dynamic models can be used not just for predicting future environmental states from current conditions, but also for predicting behavior in hypothetical scenarios of management or environment (9). Previously, this capability was demonstrated over short horizons (9). A tactical objective for this project was to examine the potential for extending EDM scenario exploration under longer-term futures that are beyond the constraints of past behavior (that is, to distant no-analogue states). Furthermore, the results of extrapolation can depend heavily on the assumptions chosen such as to which variables interact with each other. We explore the sensitivity of extrapolation to changes in assumptions in models and explore avenues for adopting multivariate data leverage (see 1.2.2.1) to clarify future projections. Conceptually, we must also reconcile the need for long-term prediction with the inherent indeterminism in nonlinear systems due to stochasticity, forecast decay, and even classical dynamic chaos (10).

1.2.2.4 Non-Analogue Projection and Nonlinear EWS

Early warning signs (EWS) of critical transitions have garnered considerable interest over the last decade in giving simple metrics to predict dramatic change in nonlinear systems. However, these methods rely on assuming change is driven by a single slowly-varying external parameter, and view critical change as abrupt change between different alternative equilibria states.

1.2.3 Technology Transfer: Code & Documentation

A final transition goal of the project, and really our main challenge in terms of making EDM widely accessible, was to develop a standardized set of computational tools for public distribution. This includes providing detailed documentation and tutorials to allow new users who might wish to apply the methods to new areas of study. This should not only benefit a variety of SERDP projects but should have an impact on environmental forecasting, environmental management and science in general.

1.3 Technical Approach

1.3.1 EDM Basics

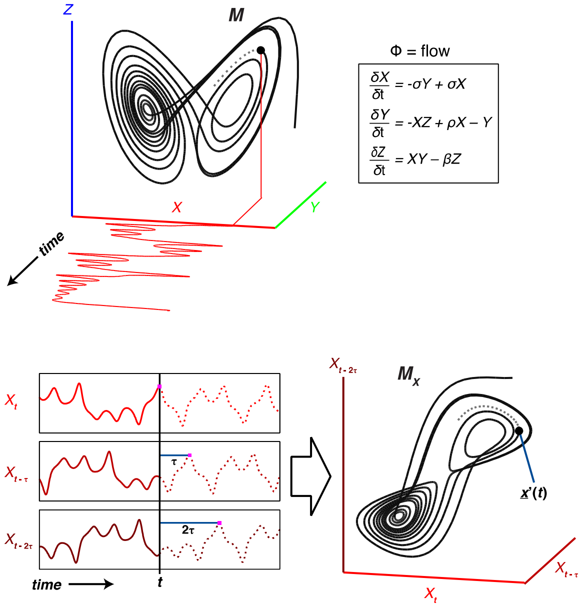

Z
M
Φ = flow
$\frac{\delta X}{\delta t} = -\sigma Y + \sigma X$
$\frac{\delta Y}{\delta t} = -XZ + \rho X - Y$
$\frac{\delta Z}{\delta t} = XY - \beta Z$
X
Y
time
X_t
$X_{t-\tau}$
$X_{t-2\tau}$
τ
2τ
time
t
$X_{t-2\tau}$
M_x
$x(t)$
X_t
$X_{t-\tau}$

Figure 1.1—Schematic of multivariate system reconstruction from a single time series. When there are missing variables or uncertainty about relationships, time lags of a single variable can be used as substitutes or proxy coordinates (bottom left, bottom right). Thus, a single time series can be used to make a univariate embedding. The geometry of the original multivariate data (the attractor above in x, y, z coordinates) can be used predictively, to identify relevant variables, and even measure changes in variable interactions through regime change.

Video 1: https://youtu.be/fevurdpiRYg
Video 2: https://youtu.be/QQwtrWBwxQg
Video 3: https://youtu.be/NrFdIz-D2yM

Empirical dynamic modelling is an inductive data-driven approach for studying complex systems from time series observations. While classical approaches involve hypothesized models, EDM attempts to minimize these assumptions by allowing the data to speak for itself. Thus, instead of assuming a particular model, EDM allows the data to tell us what the underlying model should look like — understanding nature as it “is,” rather than as we think it “should be.” The benefits of this are most clear in the context of fisheries management, where the classical model structures used to inform management fail to demonstrate real forecast skill (11). The ability to make skillful forecasts out of sample is an important validation step in the empirical dynamic approach.

EDM is based on reconstructing an attractor from time series data (https://youtu.be/fevurdpiRYg). Thus, we consider time series of different system variables together as coordinates of a single system rather than as separate objects. This allows us to identify and study causal variables and interactions that are nonlinear and state-dependent and make skillful out-of-sample predictions (12). The abilities developed here to forecast and to explore alternative scenarios (e.g. different future environmental constraints), and to evaluate state-dependent risk in terms of uncertainty (e.g., local sensitivity to dramatic change (7) driven by dynamic instability (6)), are essential for ecosystem management – the stated goals of this project.

Determining key causal variables and their interrelationships is fundamental to understanding mechanisms and for constructing predictive models. The approach we take builds on convergent cross-mapping (CCM) as a way of identifying relevant causal variables in ecosystems—for a brief introduction see (13) or the following 1-minute videos:

1.3.2 EDM without Traditional Time Series

The geometric trajectory reconstructed with traditional EDM is closely related to another concept, the vector flow. Rather than a long winding thread through system states, the vector flow can be thought of as many short arrows, pointing how the system will evolve in the next time interval (Figure 1.2). In principle, the vector flow contains the same information about interactions and ability to predict. The advantage potential advantage is that a vector flow can be recovered from systems without traditional time-series so long as there are many equivalent observations of change over short intervals.

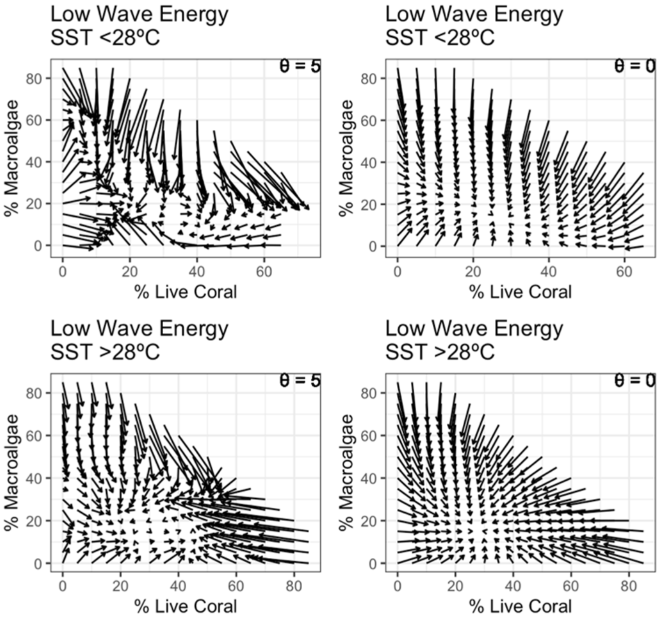


Figure 1.2— Empirical dynamics composed of many short observations. Data on the left are smoothed through a nonlinear model (θ = 5), while the panel on the right shows what data would look like if the system were linear (θ = 0). While linear analysis would suggest a single equilibrium with fixed environmental conditions, representations of the attractor dynamics show a much richer range of behavior, and point to the need for coral reef conservation to look beyond the classic idea of static multiple stable states. Moreover, these plots shown on a 2-dimensional page smooth out the causal influences occurring in higher dimensions (nearby reef patches, ocean temperatures) that are explicitly treated in the full EDM analysis.

In order for EDM to apply to these low-temporal power time series a key methodological problem we had to overcome was to find a way to determine how many active variables there are (embedding dimension (14)) and which variables to include together (causal analysis (13)). In the past, these questions have been answered with univariate EDM models using time lags.

1.3.3 Data

1.3.3.1 Case Study 1: Coastal Algal Blooms in Southern California

The core data for Case Study 1 are from a manual observational program at the Scripps Institution of Oceanography Pier in La Jolla, CA. The time series consist of chlorophyll-a and nutrient measurements at approximate half-week intervals since 1986 (with some interruption), together with physical measurements (sea temperature, salinity) measured daily.

1.3.3.2 Case Study 2: Pacific Coral Reefs

The core data for Case Study 2 are benthic cover characterizations from a visual towed diver survey (TDS) program run under the Pacific Reef Assessment and Monitoring Program (RAMP) by NOAA between 2000 and 2012 (https://inport.nmfs.noaa.gov/inport/item/35618). U.S. Pacific islands and atolls were revisited at 2-year intervals, and most islands had between 2 and 4 sequential observations. Benthic cover observations were geolocated to ~100m segments (5 minutes of diver tow), and thus the data set contains over 4,000 observations of benthic change in patches across 32 reefs over 2-year intervals. The observations include 7 islands and atolls with current or former DoD presence.

1.4 Results & Discussion

1.4.1 EDM Methods & Software Tools

A major result of this grant is the discovery that species interactions are not fixed constants, but are non-stationary in time and even highly episodic (5), as shown in Figure 1.3. This insight led to developing more informative approaches to understanding risk, resilience (6), and uncertainty (7). Tools for these approaches and others were standardized in interoperable packages in R (**rEDM**), Python (**pyEDM**), and C++ (**cppEDM**).

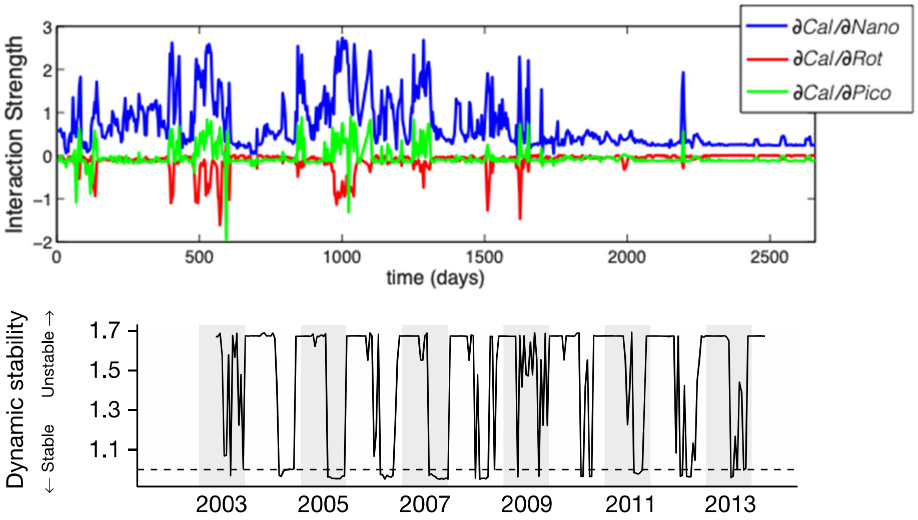
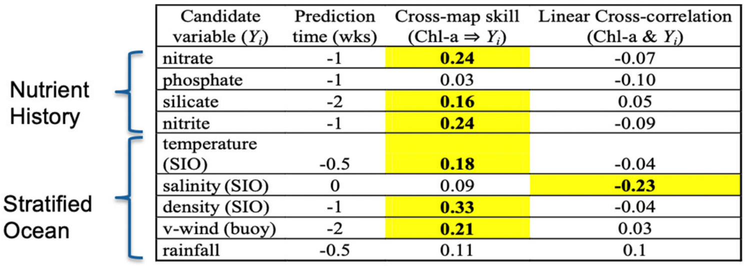


Figure 1.3—Episodic interactions and stability. (Top) Analysis of a planktonic system uncovered by EDM analysis shows that competition between grazers (red line) occurs in narrow windows of time. Reproduced from. (Bottom) These episodic interactions lead to a new view of ecosystem stability as a dynamic process Reproduced from (5) (top) and (6) (bottom).

1.4.2 Red Tide Prediction

Episodic Red Tides around Scripps are a classic example of something that no one has been able to predict. They have been thought to be regime-like, and the mechanism for the rapid transition to this state remained a mystery for over a century. Earlier attempts to understand bloom mechanisms using the Scripps Pier data were frustrated by absent or disappearing correlations between chlorophyll-a and the hypothesized drivers (see Pearson correlations reported in the right-hand column of table in Figure 0.3 below). Such so-called "mirage correlations" produce the appearance of non-stationarity and are common in nonlinear systems.

However, using an EDM nonlinear causality test, convergent cross mapping (CCM), we examined data ending in December 2010 and identified strong nonlinear causal driving in a suite of variables that relate to physical conditions and nutrient history (see Cross-map skill (reported as Pearson Correlations) column of table in Figure 0.3). Once identified, these otherwise invisible causal drivers were built into predictive EDM models that were subsequently tested on data from January 2011 to April 2012 — out of sample data that were unavailable during the model construction process. Interestingly, the multivariate models that showed best in-sample predictability (on data up to 2010) also demonstrated true out-of-sample forecast skill on data from January 2011 to April 2012 which included a large chlorophyll bloom. These core results on short-term prediction of specific red tide events were presented in McGowan et al. 2017.



	Candidate variable (Y_i)	Prediction time (wks)	Cross-map skill (Chl-a ⇒ Y_i)	Linear Cross-correlation (Chl-a & Y_i)
Nutrient History	nitrate	-1	**0.24**	-0.07
	phosphate	-1	0.03	-0.10
	silicate	-2	**0.16**	0.05
	nitrite	-1	**0.24**	-0.09
Stratified Ocean	temperature (SIO)	-0.5	**0.18**	-0.04
	salinity (SIO)	0	0.09	**-0.23**
	density (SIO)	-1	**0.33**	-0.04
	v-wind (buoy)	-2	**0.21**	0.03
	rainfall	-0.5	0.11	0.1

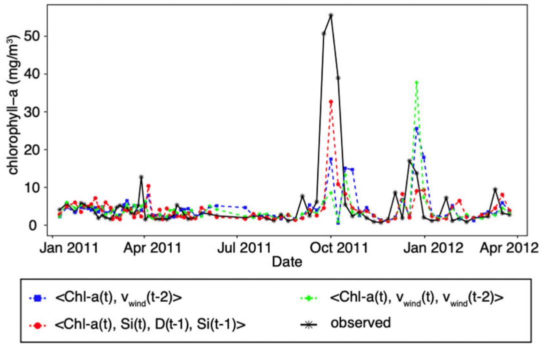


Figure 1.4—Nonlinear forecasting of coastal algal blooms in La Jolla, CA. Previous attempts to make quantitative predictions of blooms found no correlation between hypothesized drivers and chlorophyll-a (see Pearson correlation values in the right-hand column of the top table), but nonlinear causal analysis shows they are mechanistic drivers (CCM skill reported as Pearson correlations). Yellow highlighted values are statistically significant with $p < 0.05$. EDM forecasts that incorporate these variables successfully predicted novel blooms held out-of-sample in measurements made available only after the analysis was complete (bottom panel).

Building on these short-term predictions, we developed long-term scenario exploration to identify sensitivity of bloom frequency and size to non-stationary changes in physical and nutrient drivers. The analysis was structured to incorporate the best available regional ocean model (ROMs) forecasts of the future environment in SoCal, but to also acknowledge the fact that the “best available” ROMS forecast is rapidly changing and that there are multiple plausible futures. A 7-km COBALT ROMs simulation run under the “business as usual” RCP 8.5 scenario for climate change suggests temperatures will be cooler (more frequent upwelling or more influence from southward transport of cold water from the main California current) as well as increases in nutrients. Both of these environmental trends are predicted to lead to increased bloom frequency by our EDM analysis. There is also a more fundamental understanding that atmospheric warming will lead to greater stratification, and this also shows an increasing effect on bloom frequency in our analysis. Thus, although we are limited by the current state-of-art of the ROMs predictions of future physical conditions in the area, there are multiple lines of evidence all suggesting blooms will become increasingly frequent as we approach 2050.

1.4.3 Pacific Coral Reef Conservation

We successfully adapted EDM methods to the spatially-rich, temporally limited RAMP towed diver surveys to build predictive models of change in benthic coral cover. Figure 1.5 shows sequential addition of variables to identify multivariate dynamics of % coral change over 2-year sampling intervals at individual survey patches. The best forecasts are produced by integrating local coral and macroalgae cover with nearby coral cover and environmental drivers.

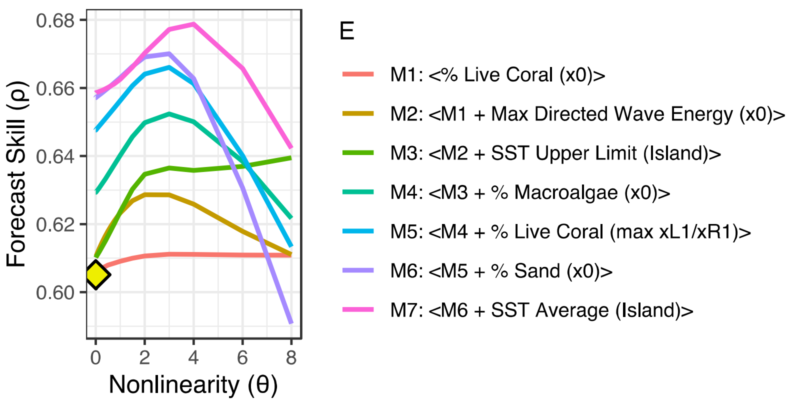


Figure 1.5—2-year prediction of % Live Coral Cover (%LC) using a greedy algorithm for selecting variables. Generally, %LC changes slowly over a 2-year period, reflected in high base-line predictability with a linear AR-1 model (yellow diamond). However, additional variance is predictable by incorporating multivariate information and allowing nonlinear interdependence between effects (θ>0). The best prediction (light purple) comes from integrating all seven listed variables.

The same procedure can produce forecasts of change in the other benthic cover variable. For example, empirical dynamics can predict occurrence of stressed coral cover two years in the future from local benthic character (%Macroalgae Cover), spatial information (nearby %Stressed Coral Cover), and environmental drivers (maximum wave energy, temperature upper limit, directional variance in wave energy). It is critical to consider these variables acting interdependently to predict stressed coral. If a linear regression is used and variables are treated as acting independently, there is no ability to predict stressed coral events.

Establishing a predictive framework for change in the benthic community then paves the way to develop dynamic benchmarks for management that describe the conservation landscape within reefs and across the 32 islands. Specifically, we apply the advances mentioned above (5–7) to quantify the stability of the benthic cover across time and space. Figure 1.6 shows how EDM estimates of the stability of the benthic dynamics and the expected growth or decline in coral across a study island to map out areas of different conservation relevance, e.g. reef segments that can be susceptible to positive intervention (quadrant IV in the figure). EDM models also provide benchmarks for sensitivity to changes in wave forcing and high temperature events.

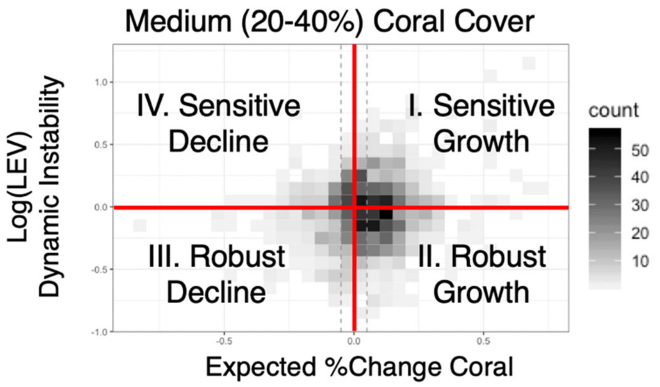


Figure 1.6— Benchmarks derived from multivariate EDM models for guiding management. The four quadrants map out the conservation landscape of an island. Areas with high instability indicate patches more susceptible to both positive intervention and negative impacts. A patch in quadrant I may grow, but this growth is sensitive to changes in the bottom and environment, meaning it could be easily derailed by human actions. On the other hand, a patch in quadrant IV is expected to decline, but could be steered to positive growth through restoration actions more easily than a patch in quadrant III. Note, each data point corresponds to a physical location on the reef.

1.5 Benefits and Implications for Future Research

This project has already produced 22 published manuscripts that advance EDM methodology and applications, with several more papers being planned. Among these are applications to understanding climate drivers of mosquito born illness (15–17) and further studies on understanding natural resource exploitation under climate variability (18, 19).

1.5.1 Data Fusion and Extending Modern Observations

During this project, a new automated profiling instrument platform was deployed just off the SIO Pier. The data from this so-called "wire walker" between 2016-09-11 and 2017-09-05 presented the opportunity to use short-term, high resolution measurements to enhance mechanistic interpretation of original study variables by explicitly considering ocean stratification (Figure 1.7A). In particular, the EDM cross-mapping relationship (13) between the wire-walker measured stratification and the manually measured surface state allowed us to reconstruct or *impute* a historical time series of stratification. This synthesized time series (Figure 1.7B) shows enhanced ability to predict historical blooms, and allows us to reconsider our long-term predictions of blooms in terms of more fundamental oceanographic change. Demonstrating this extension of EDM cross-mapping has huge relevance to environmental management through data-science, as it provides a way to reconstruct (impute) novel modern measurements over historical periods before the modern technology came into being.

Bloom prediction in the historical data in principle demonstrated good predictability of red tide events with a 1-week or greater lead time. However, the current data collection pipelines pose practical limitations to implement this for operations decisions: manually collected nutrient measurements are not processed in real time, but were key variables for prediction. A practical way forward could be to impute nutrients through cross-map (ala Figure 1.7) from other variables easier to measure in real time like pH or dissolved oxygen.

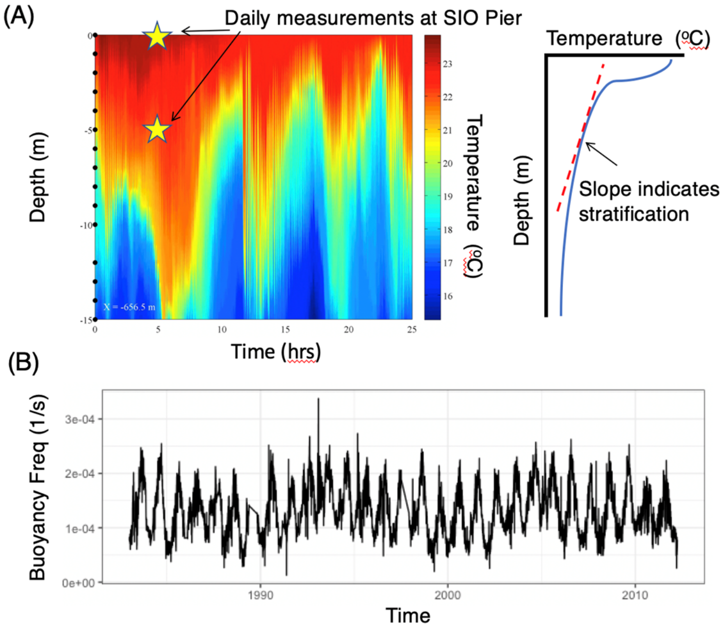


Figure 1.7— Empirical dynamic model reconstruction of ocean stratification (calculated by the buoyancy frequency) from surface pier measurements. (A) Example data from wire-walker automated profile moored 100m off-shore of SIO Pier study site (77), contextualizing the detailed ocean state measurement possible with modern automated methodology against traditional daily measurements at the SIO Pier made by hand (yellow stars). (B) Shared causal information between surface measurements and ocean stratification (13) measured over a 1-year deployment in 2018, however, allow historical reconstruction of stratification across the historical SIO Pier time series.

1.5.2 Further study of coral reefs

EDM Benchmarks for Pacific Coral Reefs derived from the RAMP Towed Diver Surveys have practical utility for informing future management. Although NOAA no longer collects these data, the historical library we have built from the TDS remains relevant. For any given site in the future, only a single new snapshot is required to seed model forecasts and update benchmarks. An additional possibility is to adapt the analysis framework from TDS data to photomosaic quadrats, a rapidly expanding "big data" technology for reef monitoring. Photomosaic quadrats offer much greater detail spatially and taxonomically, but also have backwards compatibility to TDS as the same general % cover variables can be calculated as well.

1.5.3 Lake Geneva

During the grant, there was opportunity to apply the same methodology developed for the red tide case study to a conceptually similar problem. This additional case study emerged as a case of opportunity involving a classic limnological example: predicting water quality health in Lake Geneva under various plausible climate and management future scenarios. Like the red tide event, there are extreme events that dominate the relevant ecological history, but in the lake they are changes in oxygen rather than changes in phytoplankton. Previous attempts at parametric modeling of hypoxia in Lake Geneva have had some success, as 1-dimensional physical circulation models can capture the physical mechanisms of oxygenation well. However, long-term changes (non-stationarity) in biological relationships that play in oxygen consumption, like the carbon:phosphorous ratio and carbon export fluxes, confound the validity of extrapolating the parametric models to new scenarios. We find that EDM can capture these changing relationships and explain (predict) historical variability better than a strict parametric approach, particularly recent behavior as the lake has entered new regimes of biogeochemistry.

1.5.4 New avenues

The successful adaptation of EDM methods to the Pacific Coral Reef case study demonstrates a radical advancement for EDM as a practical tool for management. Traditional EDM relies on long time-series measurements to build an understanding of system complexity and identify relevant interacting variables, thus making it poorly suited to addressing emerging management questions in ecosystems or environments that don't happen to have long-term observations associated with them. The road map set out in case study 2 shows how emerging management questions in previously un-studied systems could be addressed through EDM, so long as change (dynamics) can be measured over a limited time (months or a few years) in many equivalent systems, such as patches across a large area in a spatially explicit system like tropical coral reefs or forest fire. However, replicates do not need to have a spatial relationship, as the same general approach would work for e.g. water quality in US Army Core multi-use reservoirs across the United States (20).

2 Objectives

The purpose of this project was to investigate and further develop empirical dynamic modeling (EDM) as a practical nonparametric approach for environmental science that explicitly acknowledges the reality of natural (non-engineered) systems. Ecosystems have interconnected and interdependent pieces, meaning that causes, mechanisms, and dynamics are state dependent, non-separable and ephemerally non-stationary.

The conventional parametric approach involves hypothesized model structures and key variables, then fitting data to adhere to these notions. But parameter values that optimize an in-sample fit don't necessarily make for a predictive model. This is most clear in the context of fisheries management, where a fixed set explicit model structures are used to inform management across fisheries and countries, but fail to fundamentally demonstrate real forecast skill (11). EDM represents a paradigm shift away from current parametric models, built on the principle of empirically determines structure and variables, and uses true out-of-sample forecast skill as the rigorous measure of model merit rather than goodness-of-fit.

Thus, **the core strategic goal of the project was developing EDM to address the need for credible ecological forecasting to understand past, present, and future in non-stationary, non-equilibrium systems, and to raise consciousness of the particular complications that arise for predicting futures in these kinds of complex natural systems**. Through model tests and case study development, we have developed and demonstrated real solutions to the forecasting problem with particular attention paid to ongoing development of methods for new kinds of data sets and computational tools that are publicly available.

Two specific case studies were chosen to advance this broad strategic goal of developing non-analogue prediction that also have specific relevance to the SERDP mission. The first focused on predicting red tides and the second on predicting coral reef dynamics in the US Pacific Islands. The first is relevant to the Army Corp of Engineers, and the second included direct investigation on islands with current (Wake, Guam) and historical (Midway, Johnston, Palmyra) DoD presence.

2.1 Working Hypotheses

The EDM approach uses time-series data and non-parametric calculations to reveal the actual dynamic relationships operating among variables as they have occurred rather than as we imagine them to be. Thus it is a minimally assumptive modeling paradigm compared to parameterized approaches which assume *a priori* functional forms, as well as linear statistical modeling that assumes interactions between components occur independently. Rather, the grounding assumption of EDM is that there are deterministic rules governing the interactions between eco-system components. Stated more technically, our basic hypothesis is that **changes through time in ecological variables** (e.g. chlorophyll-a in the case of red tides, % coral cover in the case of Pacific reefs) **can be explained by low-dimensional nonlinear endogenous dynamics forced by environmental drivers**. Importantly, it is standard in applications of EDM to explicitly evaluate this hypothesis as a first step of analysis. Additionally, our working hypothesis is that **EDM models constructed on historic data (**in case of red tides**) or data from similar but non-identical systems (**in case of coral reefs**) can forecast non-analogue behavior outside the strict bounds of the original observations.**

2.2 Methodological Advances:

The specific objectives of this project involving methodological advancement all serve the broad strategic goals of developing EDM to address the need of credible ecological forecasting and broadening the applicability of EDM for environmental management under the purview of SERDP and beyond. Furthermore, tactical goals 2.2.1, 2.2.2, 2.2.3, and 2.2.4 most directly serve the strategic objective of **creating an empirical forecasting framework that can accommodate multiple plausible futures**.

2.2.1 Leverage quasi-replicates for predicting non-analogue events

Dewdrop regression is an approach with empirical dynamic modeling to leverage similarity across observed systems such that attractor dynamics from one can "co-predict" behavior of another (21). This project sought to expand on this idea of dewdrop regression specifically to predict novel extreme behavior (e.g. rapid transitions between non-equilibrium regimes) in a system based on observations of previous occurrences in other areas. Examples could include species collapses in fisheries, disease outbreaks, or bleaching in reef systems. This can be particularly applicable as climate change impacts (e.g. coral reef stresses) progress up latitudinal gradients. The working hypothesis is that **using multiple time series replicates (each showing a single transition), we can forecast these transitions in a system that has not yet undergone transition.**

2.2.2 Long-term simulation for multiple plausible futures

A tactical objective for this project was to examine the potential for extending EDM scenario exploration under longer-term futures that may go beyond the constraints of past behavior (i.e. to distant no-analogue states). This was originally stated under Task 5, and included building an understanding of how results of extrapolation can depend on the assumptions chosen such as which variables interact with each other. This objective was expanded on in Task S4 of the supplement funding, to include exploring avenues for adopting multivariate data leverage to clarify future projections. The EDM framework needs to account for stochasticity, allowing for both uncertainties in drivers and in the ecological response.

2.2.3 Multivariate Data Leverage & Multimodal Inference

This objective was articulated under the supplemental funding (Task S2). We sought to expand EDM theory to develop ensemble models for improved forecasting and prediction both short-term, and for long-term scenarios. This will build upon existing advances in EDM to leverage high-dimensional datasets (3) and will better support Tasks 5 and 7 ("Develop Theory and Methods to Extrapolate EDM to Non-Analogue Futures" & "Develop Theory and Methods to Leverage EDM using Spatial Data").

2.2.4 Non-Analogue Projection and Nonlinear EWS

We sought to develop an EDM approach for credible forecasting of longer-term ecosystem dynamics under multiple plausible future scenarios of climate and human action. Explicitly, the project was aimed to use models to explore the ability of EDM to extrapolate dynamics beyond the constraints of past behavior, and to predict attractor switching and critical thresholds. Furthermore, the supplemental funding added an explicit objective to incorporate method advances made in the early stages of the project that established an EDM approach to explicitly measuring interactions in real-time (22) using the coefficient of local linear regression with S-maps. The advancement pointed towards the potential to develop new early warnings of nonlinear ecosystem shifts better suited to non-equilibrium systems and applicable to data sets lacking long time-series. The working hypothesis was that **matrix stability metrics corresponding to the local linear regression coefficients (Jacobian matrix elements) indicate risk of critical change (e.g. collapse) in ecosystems.**

We will explore the sensitivity of extrapolation to changes in assumptions in models. We will also explore ways to validate the assumptions that go into extrapolation. For example, convergent cross-mapping (CCM) can be used to validate which variables should be included together.

2.2.5 Quantify Uncertainty

In the past, uncertainty in EDM forecasts has generally been presented through aggregate forecast skill or forecast error for a given empirical dynamic model at making predictions across a test set of data. However, uncertainty is not understood to be uniform across an attractor, and can in theory be strongly influenced by the density of similar observations, by the varying local instability or stability of trajectories (23, 24), and by the varying sensitive to stochastic (environmental) drivers. Thus, a goal of this project was to develop state dependent uncertainty measures for EDM forecasts (with simplex projection and S-maps). This advancement was essential to other pieces of the project (e.g. 2.2.2) but also for suitability of EDM to broader application in environmental management.

2.2.6 High Spatial Power Data Leverage

The core of empirical dynamic modeling was built through the study of long-term ecological observations. However, reliance on long-term monitoring is not practical for many management applications. At the same time, detailed monitoring data are often collected over short time periods in spatially explicit data sets, and these have the potential to contain as much or more empirical information on dynamics and causal relationships as single long-term studies. Moreover, spatially rich data sets (unlike long time-series) can be rapidly collected in response to acute emerging environmental management needs. Thus, we proposed to **extend EDM methods to accommodate high-spatial, low-temporal power data series** common in ecosystem study and more practical for treating acute management.

Dewdrop with spatial replicates was previously demonstrated for CCM. However, the method still relied on time-lagged vectors. However, enough time lags are needed to sufficiently resolve the multi-dimensional attractor dynamics. Even a three-component system requires at minimum three years of history at each site to take this approach.

2.2.6.1 Spatial Lags

Parallel measurements made at sites offer one way to reduce the need for taking sequential time lags. However, spatial data sets often correspond to system that have aspects of dynamics playing out in space, such as dispersal. This project will investigate using measurements of variables at spatial neighbors (“spatial lags” rather than “temporal lags”) as variables for unfolding (embedding) empirical attractors. The working hypothesis for this objective is that **observations from adjacent sites (“spatial lags”) can be used in place of univariate time lags to reconstruct system attractors for EDM analysis**, and additionally, that **spatial lags can be incorporated through the same methods as single site multivariate data (**ala Dixon et al. 1999**).**

2.2.6.2 Multivariate alternatives to univariate methods

Although multivariate EDM has existed for over 20 years(25), univariate time-lag embeddings are established for basic components of EDM analysis including assessing the number of coordinate variables needed to represent the system (14) and assessing causal relationships through attractor cross-mapping(13). A key strategic objective, then, was **to develop rigorous alternatives analyzing embedding dimension and dynamic causality using multivariate embeddings without time lag coordinates** (but possibly spatial lags).

2.2.7 Code & Documentation

A central objective and deliverable for the project was a unified coding package for basic EDM and the methodological advances made here-in, complete with documentation, to facilitate direct technology transfer to new areas of application.

2.3 Case Studies:

2.3.1 Red Tide Forecasting

The primary goal of Case Study 1 was to develop an empirical dynamic modeling framework to forecast red tide events in Southern California under the hypotheses that blooms can be understood as rapid, threshold behavior produced from nonlinear dynamics and amplification of stochastic forcing. as an initial large-scale demonstration case where linear, stationary methods have failed to give insight. Additionally, we sought to apply long term scenario exploration to generate insight into likely future red tide dynamics under climate change.

Initial red tide models (Task 2) suggest that stratification is an important predictive indicator of red tide events along the Southern California coast. Under Task S1 of the supplemental funding, we proposed to extend research into these mechanisms to better be able to adapt the results to other systems of interest to SERDP, taking advantage of additional monitoring in Southern California including novel high-resolution physical oceanographic measurements initiated by colleagues at SIO during the course of this grant.

2.3.2 Coral Reef Resilience

We will establish baselines for benthic community dynamics in Pacific Island reefs (neighboring islands without a DoD history), and use these control baselines to address actions required to evaluate and restore potential environmental impacts on islands with DoD history.

The tactical objective of the second case study was to establish a baseline empirical dynamic model for benthic community dynamics in Pacific Island reefs, and demonstrate ways this EDM framework can identify benchmarks and address actions required to evaluate and restore past or future environmental impacts on islands. This includes predicting coral loss, growth, and stress as well as characterizing sensitivity to environmental drivers. Simultaneously, this case study furthers the strategic goal of expanding the practical scope of EDM by focusing on a system without traditional long time-series measurements, but high spatial power instead.

3 Background

3.1.1 The need for a new paradigm

There is a clear and growing national and global need for analytical tools that better address the underlying complexity of natural, non-engineered systems. Engineered systems are generally created with linear causality where causes act independently of one another. That is, the channel button on a television remote has a single effect, and its effect is independent of the action of the volume button. Natural, non-engineered systems don't necessarily follow this rule. Causal associations between variables are not fixed, but change with system state. For example, marine grazers can be shown to only complete for resources when they are experiencing food limitation (5). The dynamics that result from state-dependent (nonlinear) interaction can appear non-stationary and exhibit erratic, seemingly unstable ups and downs with unpredictable and sometimes severe outcomes (e.g. fisheries collapse and unstable financial markets).

The conventional parametric approach involves "hypothesized" model structures and key variables, not empirically determined ones. These models can certainly be made "complex", in some sense, but not necessarily meaningfully so. Increasing "model complexity" often means adding terms with more and more assumptions about the causal interactions in a system that are ultimately counter-productive (26). All such models are actually hypotheses. While ideally the structure and parameters of these equations provide insight into the mechanistic relationships between variables, in reality models are hampered by hidden variables, uncertain ecological relationships, and ambiguity between inherent unpredictability and model error.

The alternative to structurally assumptive modeling has been a correlation-based framework of statistically understanding complex systems. Despite the known reality and ubiquity of nonlinear dynamics (and the costs associated with unanticipated threshold phenomena and tipping points) it is still accepted practice to apply linear statistical tools like correlation analysis based on their convenience and familiarity. The paradigm is based on stable, stationary equilibrium points that allows the system to be studied as a decomposable sum of independent parts. When applied to systems that don't match these expectations, correlation analysis can give erroneous results.

3.1.1.1 Mirage Correlation

In nonlinear systems, it is not that correlations necessarily don't exist. Variables in nonlinear systems can appear correlated for years. However, this correlation can quickly evaporate even though the dynamics have not changed in any significant way. Such transient correlation followed by apparent lack of stationarity (aka "mirage correlation", Sugihara et al. 2012), is part of the phenomenology of nonlinear systems that produces the appearance of non-stationarity (Figure 3.1). Thus, just as "correlation does not imply causation," in a nonlinear system *lack of correlation does not imply lack of causation* Therefore, for systems consisting of nonlinear webs of interacting parts, correlation, though insidiously ingrained in our thinking, is fundamentally the wrong tool for identifying relevant variables. Variables (e.g., species) may be dynamically coupled (and be perfectly cross-predictable), but show no correlation in time.

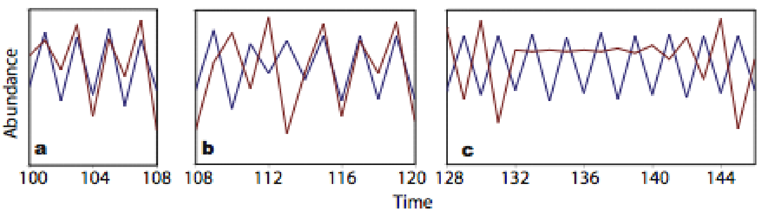


Figure 3.1—(from Sugihara et al. 2012) Correlations between variables (red and blue) can be ephemeral in nonlinear systems. In panel (a), the two variables appear correlated for 9 time steps (years). In panel (b), the correlation breaks down but returns by year 115. We may incorrectly ascribe this breakdown in correlation to a perturbation. In panel (c), and over longer time periods, there is no correlation. Although uncorrelated, the system is dynamically coupled. Both series are generated by a simple deterministically coupled 2-variable logistic difference system that remained unchanged in the simulation.

Struggles with mirage correlations continue to frustrate ecological understanding and management. In fisheries science, attempts have long been made to connect reproductive success (“recruitment”) to environmental conditions like sea surface temperature. Lacking a single precise mechanism for environmental driving (but a long list of plausible hypotheses), scientists have turned to correlation analyses. Perhaps the most high-profile case is the Pacific sardine, where environmental non-stationarity has been invoked in understanding the dramatic collapse of the fishery in the mid 1900s. However, documented correlations (27) that were then integrated into management (ostensibly to improve prediction) failed to hold up to retesting a decade later (28), throwing the management into back-and-forth argument (29). Indeed, this same story of vanishing linear effects has played out across most fisheries where environmental-recruitment correlations were put forward (30).

3.1.1.2 Rapid Transitions

Rapid transitions in state are another type of apparent non-stationarity that can arise in ecological systems due to underlying nonlinear dynamics. These include blooms, where a population or taxon rapidly undergoes exponential growth; collapses, where populations abruptly fall to low numbers and even local extinction; but also may include irreversible transitions between characteristic communities, such as suspended algae to macrophytes in temperate lakes. While gradual, monotonic change has been invoked in explanations of regime shifts in many classic cases (31), it is possible for nonlinear endogenous dynamics to produce regime shifts all on their own. Witness the two lobes of the classic Lorenz attractor or food-chain switching by a predator with context dependent prey preference (32).

Previous research with empirical dynamical modeling demonstrated in a natural system (Dixon et al. 1999) to explain spikes in damselfish larval supply dynamics in terms of physical variables related to lunar cycle, wind direction and turbulence, the latter two variables showing no apparent correlation to larval supply. In this example, when the three physical conditions aligned correctly, the effect was multiplicative and a perfect storm resulted, producing a huge spike in larval abundance. However, if any one condition was not satisfied then abundance was effectively zero. Such nonlinear state dependence predisposes ecosystems to rapid shifts in state, a major source of concern for marine resource management and environmental management more broadly. However, thresholds and regimes of real systems are difficult to reliably derive from conceptual models. The answers can depend greatly on arbitrary choices of parameter or structural form (33).

3.2 Data Science for Management of Nonstationary Futures

In the era of big data, there is enormous opportunity for a renewed ecological empiricism that is both quantitative and minimally assumptive and can address the pressing needs for a new paradigm. Rich observations can let us study nature as it is, rather than as we imagine it to be, so long as the appropriate data science is applied. The fundamental idea of EDM is to use data to study ecosystem dynamics from a geometrical perspective—the attractor manifold. This geometry can then be studied in simple, general ways that require few assumptions, then used to predict (25, 34), test causal relationships (12), and analyze environmental scenarios (35). It is front-and-center a dynamic approach that seeks to understand change rather than ignore it, and thus is especially suited to systems with complex interdependence between variables be they species, environmental drivers, or even human behavior.

The multivariate attractor can be intuitively thought of in relation to Hutchinson's *n*-dimensional niche. While Hutchinson niches might be thought of as a cloud of points in a multivariate space, the attractor is the trajectory as the system winds through the cloud due to whatever particular combination of endogenous interaction and environmental forcing. Thus, as the system changes over time, it occupies different points in the state-space, forming trajectories that comprise an *n*-dimensional geometric attractor (see Figure 4.1 in the next section), and time series of the state variables can then be viewed as sequential projections of the attractor onto the coordinate axes. Moreover, although the attractor manifold was originally conceptualized as generated from an underlying set of equations, it can also be recovered directly from observational time series data. This is trivial (yet powerful) for a system where all essential variables are known and measured, it simply requires viewing the system as a multidimensional whole rather than separate single pieces of time series.

3.2.1 Variable identification: CCM instead of correlation

Determining key causal variables and their interrelationships is fundamental for constructing interaction networks and making reliable forecasts of ecosystem change. Clive Granger made the connection between forecasts and causality in his Nobel Prize-winning work (36). However, the test he developed, *Granger causality*, is designed for systems where causes and effects are neatly separable (linear). It does not generally apply to nonlinear dynamic systems, where information about the dynamics of causal variables is embedded in the time series of response variables. However, this property of nonlinear systems then forms the basis of convergent cross mapping (CCM) as a way of detecting dynamic causal links built on the general principles of EDM. Specifically, CCM uses cross-prediction between variables to establish causation—e.g. if past sea surface temperatures can be estimated from sardine time series, then temperature must have been a causal variable, whose historical effect was recorded by sardine populations (Sugihara et al. 2012).

Empirical dynamic modeling has traditionally been applied to long time-series data. Indeed, work supported by this grant showed with fisheries data how short time series can mask underlying nonlinearity (37); it can lead to large uncertainty in causality assessment and bottom line prediction. Nevertheless, needs of environmental management generally only match up with long-term ecological observations by happy coincidence. Additionally, in truly non-analogue climate futures, causal relationships made fundamentally change, rendering causal associations learned from historical dynamics less useful. Combining short time-series from spatial replicates can be used to overcome this to some extent, as shown by Clark et al. (38). However, the approach is still limited to having time series at each site at least as long as the number of dimensions needed to unfold dynamics. In the towed-diver surveys we seek to study for Case Study 2, many sites were only revisited a single time. Generalizing convergent cross-mapping to these situations is thus critical to developing EDM as a practical tool for studying and managing non-stationary, non-equilibrium futures.

3.2.2 Stability away from equilibrium

The classic ideas of stability in ecology are built on linearization around a static equilibrium point, which reduces the dynamics of the system to a matrix of linear interaction coefficients (the Jacobian or Community matrix). For linear dynamics around an equilibrium, the stability is explained by the largest eigenvalue of the Jacobian matrix, λ_1. In discrete-time representation, if $\lambda_1 > 1$, then the system will iteratively diverge from the equilibrium, and hence the point is unstable (see left side of *Figure 3.2*). The equivalent in a continuous time model is $\lambda_1 > 0$.

This framework lead to explanations of rapid change in ecosystems as arising from multiple stable states modulated either by a slow change differentially affecting the stability of equilibria or by sudden perturbations driving the system from one basin of attraction to another. However, it has long been appreciated that persistence in ecological communities is not always so easily understood as point equilibria (39). In this case, the Jacobian matrix still controls the local convergence or divergence of dynamics. However, the Jacobian matrix evolves as the system passes through ecosystem states, since it is not just remaining in linear neighborhood of a single point. This sequential Jacobian estimation is in fact the basis of an important EDM approach, S-maps (40, 41), and thus points a way towards understanding dynamic stability and anticipating rapid change within established EDM methodology.

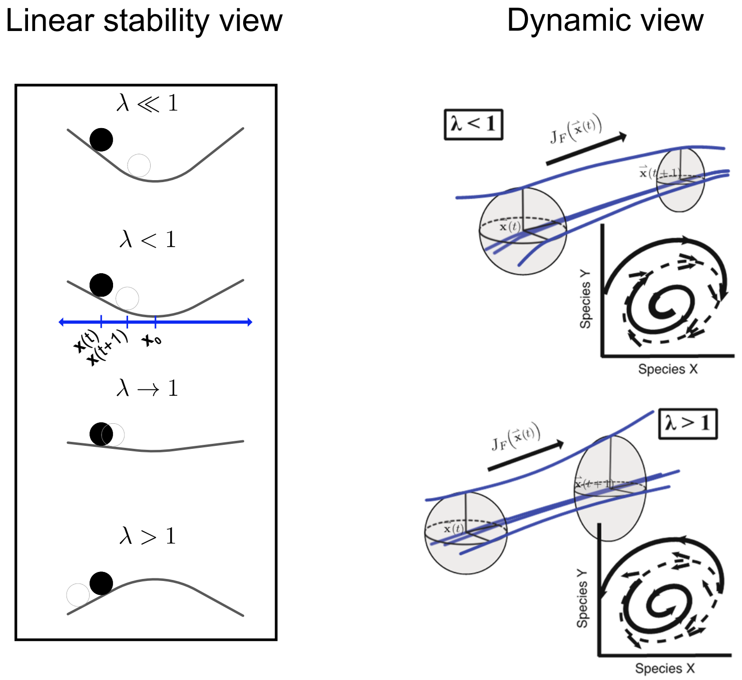


Figure 3.2— Linear and dynamic views of stability both are described by eigenvalues of the Jacobian matrix (local linear approximation). However, in a system near equilibrium a single fixed Jacobian can be defined for the

system, while in a non-equilibrium system that Jacobian will evolve through time. Both can be accommodated within EDM.

3.2.3 Early Warning Signs

Importantly, the existing early warning signs of critical transitions (EWS) are based on a conceptual framework of linear equilibria (left hand of Figure 3.2). Decreasing stability of a linear equilibria is marked by "critical slowing" of dynamics. This is illustrated on the left panel of Figure 3.3. When an equilibrium is highly stable ($\lambda << 1$), dynamics quickly return the system to the point equilibrium after perturbations, but this return time increases as the point loses stability ($\lambda \rightarrow 1$) before the system ultimately transitions to a different equilibrium. This gives rise to a number of related indicators of critical change such as rising autocorrelation and variance (right panel of Figure 3.3).

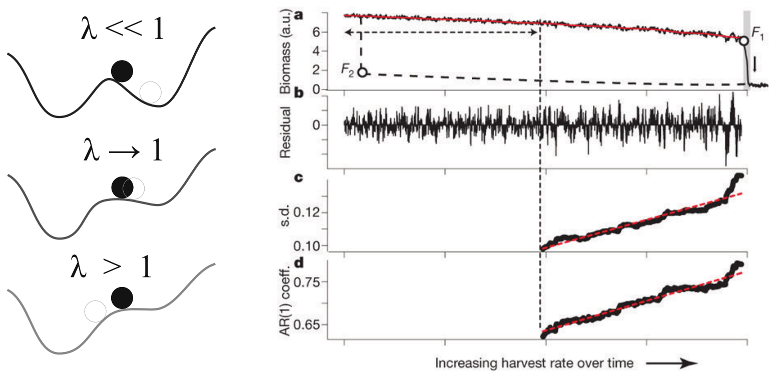


Figure 3.3— Multiple stable states and early warning signs of critical transitions. Right-side panels reproduced with permission from Schaffer et al. (42). Decreasing stability of point equilibria leads to "critical slowing" of dynamics (left), which has given rise to a number of related indicators of critical change such as rising autocorrelation and variance (right).

However, despite being a popular topic in the literature, the EWS metrics have not seen practical implementation for management. Neither rising variance, nor autocorrelation come with natural thresholds for prediction, but are rather qualitatively indications. Additionally, these are built on having long time series of a single system to notice these gradual changes over time. Finally, their justification is based on a stable equilibrium view of resilience at odds with many ecosystems. Returning to the underlying idea of Jacobian matrix stability points the way towards a more general and rigorous approach to EWS.

3.3 Case Study 1: Red Tides in Southern California

3.3.1 A Century-long Mystery with Mirage Correlations

Rapidly appearing dinoflagellate blooms off the coast of La Jolla and Southern California have been a mystery for over a century (43). A predictive understanding of these events has long eluded science, and thus this case study provides a high profile example for understanding and forecasting episodic, threshold events.

A comprehensive near-shore monitoring program at the Scripps Institution of Oceanography (SIO) pier has been ongoing for over three decades. Biweekly and daily measurements include chlorophyll-a, dissolved nutrient concentrations, water temperature, and density for much of the observational period. Additionally, rainfall, wind, and offshore temperature measurements are available from nearby stations. It is a unique opportunity, both in terms of data-breadth and temporal depth, to approach the mystery of what controls the legendary episodic algal blooms.

Past attempts to understand this system (e.g. Kim et al. 2009) based on a paradigm of stable, stationary equilibria, treated the different hypothetical causes piecewise (as a decomposable sum of independent parts), yielding little in the way of explanation or prediction. The analyses were based on linear cross correlation. In past decades, the best explanation found with linear analysis was an apparent positive correlation between temperature anomaly and chlorophyll-a (Figure 3.4). While the relationship initially appeared strong when data through 1993 were used (correlation of ρ = -0.33), it switched sign then vanished in the data collected from 1993-present (ρ = -0.02). Mirage correlation such as this is expected in nonlinear ecosystems (Sugihara et al. 2012).

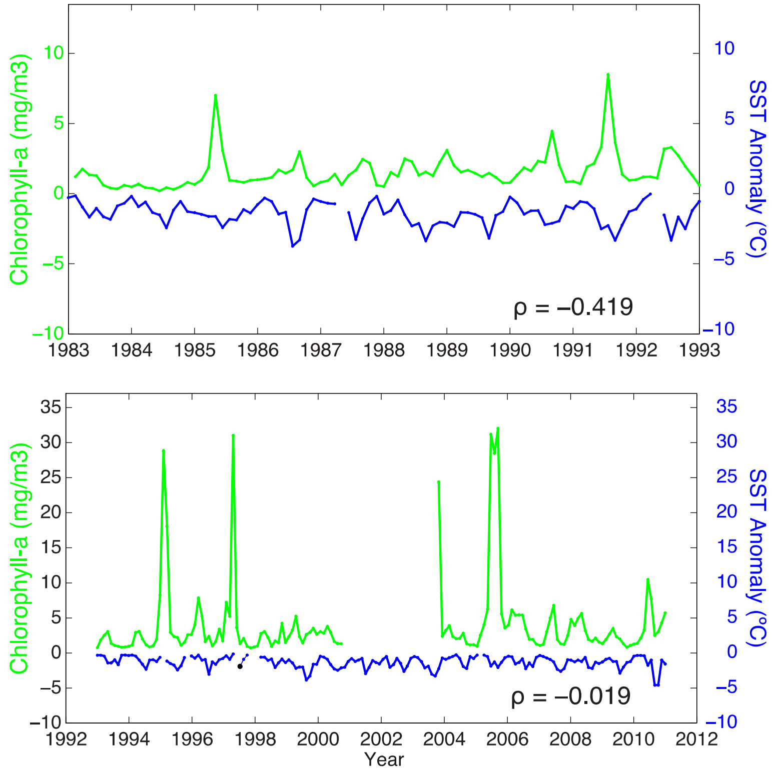


Figure 3.4— Mirage correlation frustrated earlier attempts at identifying a causal mechanism for predicting red tides in La Jolla. A sea surface temperature (SST) anomaly was constructed by differencing surface and bottom measurements at the SIO pier that had a strong negative correlation to chlorophyll blooms between 1983 and 1993 (top). However, this relationship was not immediately published and this linear relationship has since vanished (bottom). Such mirage correlations are a hallmark of nonlinear ecological dynamics.

3.3.2 Negative Impacts

In recent history, the most common dinoflagellate causing blooms in La Jolla is *Lingulodinium polyedrum*. Although *L. polyedrum* does not produce toxins directly harmful to humans (or other vertebrates), other blooms in the Southern California bight do produce toxins; these may cause detrimental health impacts during open water training/activities (e.g., reports of increased ear and sinus infections for swimmers). Even non-toxic dinoflagellate blooms produce negative impacts, however, and can be classified as harmful algal blooms (HABS). Exposure to blooms can affect equipment, requiring more maintenance to clean small boat engines, and various sensors (esp. optical) in bays/harbors for long-term monitoring. Advance notice of blooms via predictive models may allow proactive measures to be taken to limit their negative effects.

More importantly, blooms of *Pseudo-nitzschia*, which produce Domoic acid (a neurotoxin), can lead to mortality events among seabirds and marine mammals. This includes the California Least Tern, an endangered bird; over 1/3 of California Least Terns nest on land managed by the DoD (Camp Pendleton & Naval bases in Southern California). Algal blooms reduce open water foraging areas, cause large hypoxic zones, precipitate mass fish die-offs, leading to reproductive failure or even direct mortality in conservation targets. Thus, knowledge of whether red tides will increase or decrease 25 years from now would provide actionable information and advance warning.

Finally, beyond southern California, harmful algal blooms occur across several other ecosystems where the DoD has responsibilities and interests, including the Gulf of Mexico, Atlantic coast estuaries, and freshwater reservoirs maintained by the Army Corps of Engineers. Developing a robust analysis for predicting threshold, exponential response of algae in southern California, thus, can lay the foundation for future approaches in these other areas.

3.4 Case Study 2: Pacific Coral Reef Resilience

3.4.1 Ecological concerns at U.S. Pacific Reefs

Coral reefs are in crisis globally. Loss of coral cover has been tied to multiple stressors within and across reef zones, including human and natural physical disturbance, thermal stress, disease, and over-fishing of important herbivores. Coral reef status at current and former DoD sites in the Pacific differ greatly, with military presence associated with some positive characteristics (like fishing bans) but also physical disturbance due to operations and ongoing issues associated with past activities, such as the reengineering of Palmyra atoll. Understanding the factors that shape reef resilience is critical to ongoing conservation at Pacific island locations, particularly in the face of non-stationary environmental forcing under a changing climate. As reef cover continues to decline globally, it is important to avoid adverse impacts on sites that remain resilient to change, as well as identify opportunities to improve reef health where recover is possible.

Statistical analysis across the Hawaiian archipelago identifies clusters of states indicative of multiple regimes of behavior, including a “calcification regime” dominated by hard corals, a “turf algae regime”, and a mixed macroalgae and sand regime (44). All three statistical regimes exist across human and environmental gradients. These regimes do not necessarily constitute stable states, however, and occurrence of the statistical regimes seemed to hinge on variable ecosystem states, like the abundance of classes of herbivorous fish, and the dependencies suggested nonlinear relationships.

Previous quantitative analysis has also highlighted the importance of wave energy. Generally speaking, high wave stress is associated with low or absent coral cover, but the effect can depend on the community composition, as well as other stressors like temperature and sediments (45). While higher wave energy areas leave coral more vulnerable to physical damage, they are also associated with lower temperatures (due to greater circulation and flushing with deep water) and sedimentation than sheltered areas like lagoons. Critically, relationships with the environment show characteristics of nonlinear relationships, where coral communities can robustly reorganize to maintain resilience, but occasionally reach abrupt tipping points (46). Predicting these thresholds from individual physiologies and mechanistic models is a dizzying prospect, and thus is an ideal case to try to employ empirical, data-driven approaches.

3.4.2 Restoration as an emerging opportunity for data-driven science

In addition to basic science needs for identifying factors of resilience and changes to resilience over time, there is also an opportunity for empirical approaches to guide next generation restoration efforts. There are many (often creative) tools emerging for restoration include algal removal with vacuums, but these cannot be usefully deployed without a predictive framework of reef resilience and recovery. Coral out-planting has shown exciting potential in a number of cases. At Laughing Bird Caye in Belize, for example, Fragments of Hope and collaborators have increased coral cover by an estimated 50% within the National Park. Nevertheless, coral reef conservation is inherently a multi-scale endeavor, with management actions, ecosystem processes, stressors, and outcomes all happening at a variety of spatial and temporal scales. As coral declines have multiple attributable causes across systems, resilient reefs seem to require a number of characteristics. Critically, these processes and factors do not neatly decompose in a separable way. Although this is certainly widely appreciated in the abstract, the full ramifications of this reality are not often assimilated into the analytical approaches we take to management and conservation. Large scale out-planting is already in the works, including an ambitious plan by NOAA and partners to restore Seven Iconic Reefs along the Florida Keys. The more reef resilience and restoration success can be best understood as emergent properties of complex ecosystem context and dynamics, the more these efforts can hope to meet the promise of their price tag.

3.5 Other Empirical Examples

3.5.1 Cedar Creek LTER

Long-term experiments and observations at Cedar Creek have demonstrated regime shift behavior in the vegetation community which can be driven by nitrogen addition (47). The shift is characterized by loss of native C4 grasses, and dominance by two invasive species, *Poa pratensis* and *Elymus repens*. Experiments have shown these shifts display hysteresis, i.e. the dominance of invasive species can persist for decades after nitrogen additions have ceased. However, the dynamics in the plots also don’t conform to neat ideas of stable equilibria. For example, even after plots have shifted to dominance by invasives, there are large year-to-year fluctuations in biomass (Figure 3.5).

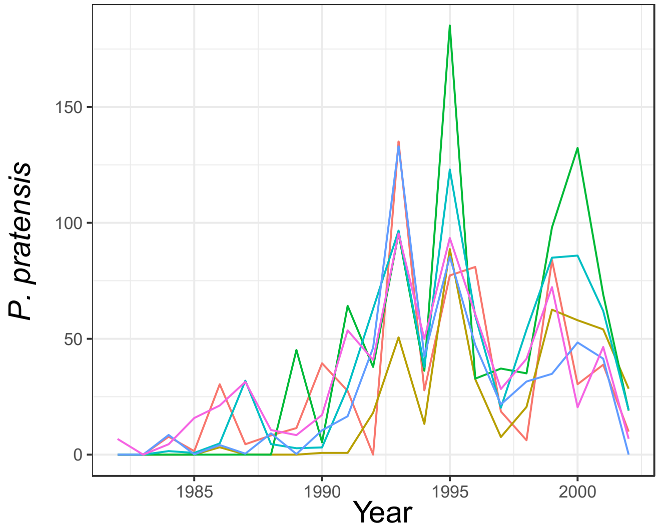


Figure 3.5— Time-series of above ground P. pratensis biomass in Cedar Creek experimental plots under moderate nitrogen enrichment treatment, 5.44 kg N ha^{-1} yr^{-1}.

4 Materials and Methods

4.1 EDM Basics

We begin by briefly describing established EDM methodologies. These methods were used extensively in the project and variously adapted in new ways and innovated upon. Several of these methods were explicit technical objectives of the project, while a few arose organically in pursuing our strategic goals. The established basics described in 4.1.1 form the foundation for the innovations described in the remaining sections.

4.1.1 Attractors: What and Why?

As discussed in 3.2, Empirical Dynamic Modeling is a data-driven approach to quantitative ecosystem study and forecasting that centers on the geometric attractor. Like equations, the geometric attractor represents the relationships between variables that cause changes through time in a system, but unlike equations, these can be recovered and studied in simple, minimally assumptive ways.

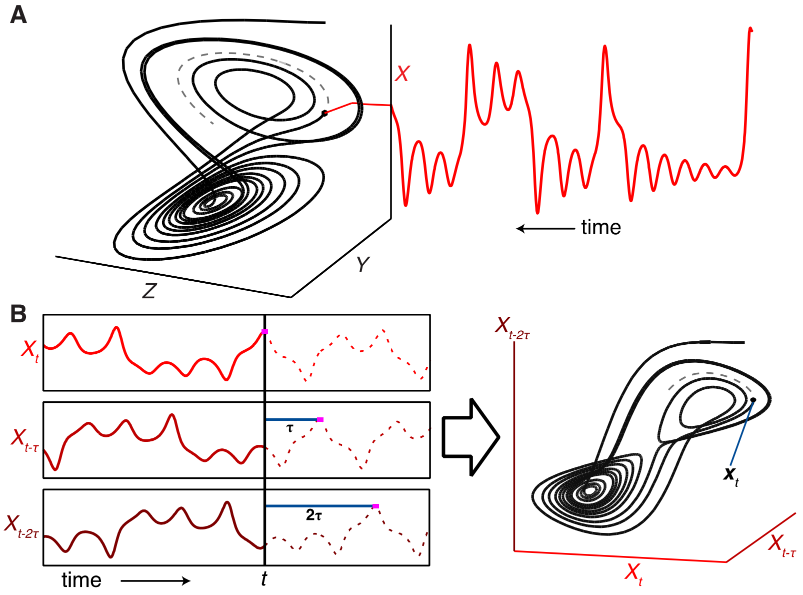


Figure 4.1—Reconstructing the system dynamics from a single time series. (A) The basic concept of EDM is that time series can be understood as a projection of the motion of the multivariate system along its attractor trajectories. Here, the canonical Lorenz attractor projected onto the x-axis yields a time series for variable x. (B) Successive lags (with time step τ) of the time series x_t are plotted as separate coordinates to form a reconstructed (and visually similar) "shadow" manifold that preserves essential mathematical properties of the original system.

Takens's theorem and extensions (48–52) formalize the idea of using time-lagged coordinates of a single variable to recover a shadow version of the underlying system attractor. In practice, the time-lag interval τ needs to be appropriate so that successive lags aren't too highly correlated, but aren't past the horizon of information decay.

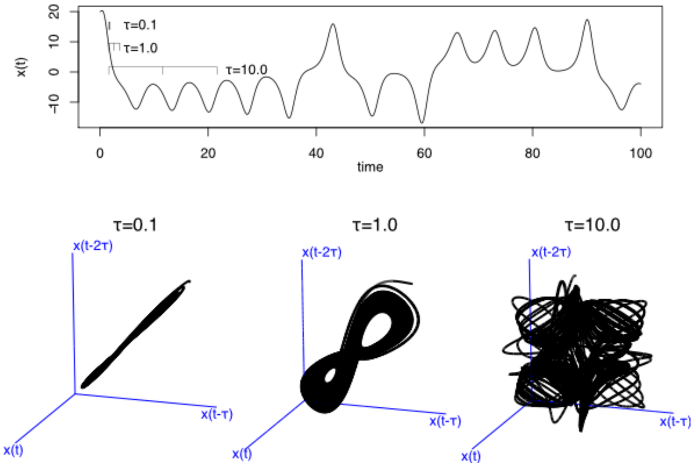


Figure 4.2— Effect of time-lag interval τ on manifold reconstruction. Using time-lag coordinates that are too highly correlated (left) or are past the information decay of the system (right) can lead to poorly resolved embeddings.

Figure 4.2 gives an illustration with the Lorenz system of how the ability to recover a predictive attractor can be affected by picking a tau that is too short or too long. If measurements are made rapidly relative to the characteristic rate of change of the system and we attempt to do a lag reconstruction with this very short τ (left-most attractor), the successive lags have almost the same information, and the attractor gets smashed up against the 1-to-1-to-1 line. On the other end of the spectrum, if τ is much too long (right-most attractor) then the successive time lags don't really contain current information and the attractor becomes a tangled mess. In the middle ground, the successive lags are sufficiently different, but still contain current information (middle attractor).

If observations are made continuously (e.g. an in situ fluorometer for measuring chlorophyll), then an appropriate τ can be picked from analyzing the autocorrelation series and identifying a lag within the decorrelation time for autocorrelation to drop below $\rho = 0.6$. More commonly, observations are made at some frequency, and this sets the minimum tau we can consider. Many observational programs are keyed off of our existing understanding of the major time scales of a species or system, e.g. we survey salmon annually.

While reconstruction can be done solely from lags of a single time series, it can also be done a mixture of different, related system variables(53). Moreover, linear combinations of variables; coordinates derived from simple rescaling and recombination of existing variables are generally also valid as embedding coordinates. This equivalence raises the point that there isn't a single "correct" set of coordinates to use to treat the system. The implications of this are discussed later in 4.4.

4.1.2 Forecasting with reconstructed attractors

4.1.2.1 Simplex Projection

Simplex projection (14) is a minimal nearest-neighbor method for forecasting dynamic systems on an attractor reconstruction. The basic principle is that points nearby on an embedded attractor will evolve similarly in time. Thus, the method involves finding nearest neighbors on the reconstructed attractor to a target point and averaging the trajectories of the neighbors to estimate the future state (*p* time steps ahead) of the target. For most ecological data, there is only a single meta-parameter that tunes the forecasts, the embedding dimension E. In the previously rare, but increasingly common case of high frequency measurement, it is also possible to have choice in the spacing of time-lags, τ. Psuedo-code for the algorithm is as follows (reproduced from (4)):

(i) Use the time series data to create a library of vectors on the reconstructed attractor. For simple univariate attractor reconstruction, these vectors will just be the $\boldsymbol{x}_i = \boldsymbol{x}(\mathrm{t_i}) = [\mathrm{X(t_i), X(t_i} -\tau), \ldots, \mathrm{X(t_i} - (E{-}1)\tau)]$, for time points $\mathrm{t_i}$.

(ii) Identify a target time point, t^*, and its corresponding vector $\boldsymbol{x}^* = \boldsymbol{x}(\mathrm{t}^*)$.

(iii) Compute the Euclidian distance between the target vector $\boldsymbol{x}^*$ and all the library vectors, $\boldsymbol{x}_i$. Recall that the Euclidian distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is $d(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = [(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_\mathrm{E} - y_\mathrm{E})^2]^{1/2}$.

(iv) Using these distances, identify the E+1 nearest neighbors (the library vectors with the shortest Euclidean distance to $\boldsymbol{x}^*$). Label these $\boldsymbol{x}_{n(1)}$ be the closest vector with corresponding time index $\mathrm{t}_{n(1)}$, $\boldsymbol{x}_{n(2)}$ with time index $\mathrm{t}_{n(2)}$, etc.

(v) Assign weights w_i to the neighbors based on their distance:

$$w_i = \exp\left(-\|\boldsymbol{x}^* - \boldsymbol{x}_{n(i)}\| / \|\boldsymbol{x}^* - \boldsymbol{x}_{n(1)}\|\right) \tag{4.1}$$

where $\boldsymbol{x}_{n(1)}$ is the nearest neighbor and $\boldsymbol{x}_{n(\mathrm{i})}$ is the i^{th} nearest neighbor.

(vi) Estimate the future value of variable X from the target point as a weighted average of the neighbors

$$\hat{X}(t^* + p) = \sum_{i=1}^{E+1} w_i \, X\left(t_{n(i)} + p\right) \Big/ \sum_{i=1}^{E+1} w_i \ . \tag{4.2}$$

(vii) Repeat (*ii*) – (*vi*) for other target points.

4.1.2.2 S-map

Sequentially-weighted local linear maps or S-maps (40) are another minimal, non-parametric approach to attractor forecasting. While simplex projection is based on an average of the reference states nearby on the attractor based on their distance (locally weighted averages), S-maps is based on a linear regression of nearby states on the attractor, also weighted by their distance. S-maps has one additional parameter to simplex projection, *θ*, which tunes the degree of local weighting. Psuedo-code for the calculations are as follows (reproduced with modification from (4)):

(i) Use the time series data to create a library of vectors on the reconstructed attractor. For simple univariate attractor reconstruction, these vectors will just be the $\boldsymbol{x}_i = \boldsymbol{x}(\mathrm{t_i}) = [\mathrm{X(t_i), X(t_i} -\tau), \ldots, \mathrm{X(t_i} - (E{-}1)\tau)]$, for time points $\mathrm{t_i}$.

(ii) Identify a target time point, t^*, and its corresponding vector $\boldsymbol{x}^* = \boldsymbol{x}(\mathrm{t}^*)$.

(iii) Compute the Euclidian distance between the target vector $\boldsymbol{x}^*$ and all the library vectors, $\boldsymbol{x}_i$. Recall that the Euclidian distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is $d(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = [(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_\mathrm{E} - y_\mathrm{E})^2]^{1/2}$.

(iv) Using these distances, calculate the elements of the $n \times E$ matrix **A** and n-dimensional vector **B** as follows:

$$A_{ij} = w(\|\boldsymbol{x}(t_i) - \boldsymbol{x}(t^*)\|) \, x_j(t_i)$$
$$B_i = w(\|\boldsymbol{x}(t_i) - \boldsymbol{x}(t^*)\|) \, X(t_i + p)$$

where

$$w(d) = \exp\left(-d\theta/\bar{d}\right)$$

and $\bar{d}$ is the average distance of all library points to the target point $\boldsymbol{x}$(t*).

(v) Solve the following linear model for **C** using truncated SVD (singular value decomposition):

$$\mathbf{B} = \mathbf{A} \cdot \mathbf{C}$$

(vi) Estimate the future value of variable *X* from the target point using the calculated (least-squares) linear model $\tilde{\mathbf{C}}$.

(vii) Repeat (*ii*) – (*vi*) for other target points.

Generally, a constant term is also included in the linear equation for step (v).

4.1.3 State-dependent uncertainty

Historically, uncertainty in EDM forecasts have been characterized by the overall prediction skill or error measured across many forecasts. This uncertainty is understood to arise from a number of interrelated phenomena, including measurement error, finite sampling, local instability, and incomplete dimensionality. In a nonlinear system, we have no reason to suspect these act uniformly through time, that is, certain predictions will have higher uncertainty than others. This is especially true in predicting extreme behavior, such as chlorophyll blooms or reef collapses. To further EDM as a practical tool for managing non-stationary futures, it is necessary to establish methods (and code) to capture this.

4.1.3.1 Simplex

A straightforward way to estimate uncertainty of simplex predictions, then, is to compute the variance of the nearest neighbors.

$$var\left(\hat{X}(t^* + p)\right) = \sum_{i=1}^{E+1}\left(w_i\ \left(\hat{X}(t^* + p) - X\left(t_{n(i)} + p\right)\right)\right)^2 \Bigg/ \sum_{i=1}^{E+1} w_i{}^2\ .$$

This formulation assumes that the uncertainty follows a parametric distribution. The most convenient would be to treat the error distribution as normal, but this is not always a good assumption. For example, many ecological time series are bounded by 0, so you can easily estimate error distributions that would suggest a finite probability of a physically impossible value (<0). This can theoretically be reduced by making data transformation, but any choice of data transformation also involves assumptions and risks. For example, log-transformation is popular in many statistical analyses of ecological data to create more Gaussian distributions of values. However, log transformation can also massively inflate observation errors at low counts, and encounters particular problems when faced with 0-count inflated data. Furthermore, data transformations that create Gaussian uncertainty distributions may not be optimal for the main objective of analysis (e.g. forecast skill or causal inference).

Whether an appropriate parametric distribution and/or data transformation can be selected or not, the method is appropriate to quantify relative uncertainty between different predictions for the same system or a similar system.

4.1.3.2 S-map

Just as with simplex projection, we can look at the disagreement of the near-neighbors to assess forecast uncertainty. The variance of deviations of the reference points from the prediction is:

$$var\left(\hat{X}(t^* + p)\right) = \sum_i \left(w_i \left(\hat{X}(t^* + p) - X(t_i + p)\right)\right)^2 \Big/ \sum_i w_i{}^2.$$

Where the weights are now the S-map weights

$$w_i = w\left(d\left(\boldsymbol{x}(t^*), \boldsymbol{x}(t_i)\right)\right) = \mathrm{exp}\left(-\theta d\left(\boldsymbol{x}(t^*), \boldsymbol{x}(t_i)\right)/\bar{d}\right).$$

However, by using S-maps we are ascribing meaning to linear relationships in this neighborhood. If there are strong local linear effects of embedding variables on the forecast, these will inflate the error above. Thus, as with standard linear regression, it is wiser to use the residuals from the linear models. Since S-map is computed with truncated SVD, we can compute the residuals from the SVD matrices. Recall that if the singular-value decomposition of the regression matrix **A** is given by:

$$\mathbf{A} = \mathbf{U\Sigma V}^{\boldsymbol{T}}$$

then sum of squared residuals is given by

$$\left\|\mathbf{A} \cdot \tilde{\mathbf{C}} - \mathbf{B}\right\|_2 = \|(\mathbf{UU}^{\boldsymbol{T}} - \mathbf{I})\mathbf{B}\|_2.$$

Since S-maps weights the regression, the variance will be

$$var\left(\hat{X}(t^* + p)\right) = \|(\mathbf{UU}^{\boldsymbol{T}} - \mathbf{I})\mathbf{B}\|_2 \Big/ \sum_i w_i{}^2.$$

4.1.3.3 Jack-knife Sub-sampling

In either case, it is also possible to heuristically model forecast uncertainty by subsampling the library data used to construct the EDM model. This approach has the advantage of not requiring an assumption about the statistical distribution of error, but it is also computationally intensive to perform sufficient resampling to reproduce many state-dependent error distributions. However, this method is well suited to using EDM to simulate system dynamics as discussed in the next section where only a single random draw from the theoretical error distribution is needed.

4.2 Constructing Multivariate Models

Takens's theorem provides for reconstruction a shadow version of the system attractor from a single observation function, and allows dynamics to be treated as nonlinear and multivariate without having to explicitly identify interacting variables. However, this is largely phenomenological. Univariately modeling the empirical dyanmics allows the possibility of multivariate interactions without explicitly resolving what those are. That is to say, they can be predictive, but not necessarily interpretable.

When the system variables are known, then EDM models become much more interpretable. Even though the dynamics are modeled without parametric equations, the interactions between variables can still be quantified, for example by examining the local linear structure (Jacobian) at a given point in time

(5). However, this possibility can be difficult in practice, ecologically, because the systems were not built and thus the identity of interacting variables is often more hypothesis than fact.

4.2.1 Greedy search

The basic framework for selecting a single embedding for analysis is established for the mesocosm case study in (5). It involves first testing hypothesized drivers for interaction with convergent cross-mapping to establish a set of valid embedding variables as candidates, then sequentially replacing the phenomenological time-lag variables with interacting variables using a "greedy" approach. At each step of the algorithm the embedding is added to by a single variable that most improves prediction. Additionally, this grant developed novel approaches e.g. to multi-model inference described in later sections.

Psuedo-code for basic multivariate model selection is as follows. Given target variable Y_1, and causally interacting variables $\{Y_2, \ldots Y_N\}$ validated with CCM:

(i) Determine embedding dimension E^* using univariate simplex projection (see Simplex Projection).

(ii) Normalize all variables to have $mean(Y_i) = 0$ and $sd(Y_i) = 1$.

(iii) Compute the simplex projection forecast skill (see 4.1.2.1) of $Y_1(\mathrm{t} + \mathrm{tp})$ for the N-1 embeddings constructed with $Y_1(\mathrm{t})$, E^*-2 univariate time lag coordinates, and one of the causal interactors (bolding used for emphasis):

$[Y_1(\mathrm{t}), \boldsymbol{Y_2}(\mathrm{t}), Y_1(\mathrm{t_i} - \tau), \ldots, Y_1(\mathrm{t_i} - (E-2)\tau)]$,
$[Y_1(\mathrm{t}), \boldsymbol{Y_3}(\mathrm{t}), Y_1\,(\mathrm{t_i} - \tau), \ldots, Y_1\,(\mathrm{t_i} - (E-2)\tau)]$,
… ,
$[Y_1, \boldsymbol{Y_N}(\mathrm{t}), Y_1\,(\mathrm{t_i} - \tau), \ldots, Y_1\,(\mathrm{t_i} - (E-2)\tau)]$.

(iv) Identify Y_{i*} that gives the highest forecast skill.

(v) Repeat (ii)-(iv) but replacing a random projection coordinate with the Y_{i*}, and removing Y_i from the candidate embedding variables.

4.3 Long-term Scenario Simulation with EDM

Technically speaking, it is trivial to generate a long-term prediction with the same methods listed above. One can simply set the prediction time, p, to a very large number or do a repeated iteration of short-term forecasts, where the output of the first prediction say $x(t+1)$ is taken as the input state for the next prediction. However, just because it is technically possible does not mean it is a reliable method.

Conceptually, the chief obstacle is to reconcile a long-term prediction with the inherent indeterminism in nonlinear systems due to stochasticity, forecast decay, and possibly even classical dynamic chaos (10). If one extends the prediction time for simple dynamic models, the forecasts of simplex or S-map begin to approach the sample mean, making that approach not particularly useful as a modeling tool. Alternatively, if traditional simplex or S-map is iteratively applied to create a long-term trajectory, the simulated dynamics will not trivially fall to the mean, but they will be completely deterministic and biased towards averaging out large variances. This is particularly concerning for one of our key objectives, which is to develop EDM to simulate extreme events like red tides. This problem can be addressed by instead employing stochastic EDM forecasts, where process error is added to forecasts by drawing from the state-dependent uncertainty in forecasts described in 4.1.3. This allows EDM to reconstruct multiple plausible futures. Note, this process is intrinsically sensitive to systematic bias in the forecasts, so it is important not to introduce bias through the simulation of process error. For that reason, we simulate the process error use jack-knife sub-sampling rather than approximating a parametric error distribution.

4.4 Multi-model Approaches

Multivariate Takens theorem (54) presents a challenge and an opportunity. The result that the dynamic attractor can be reconstructed "diffeomorphically" from various combinations of lags of different variables in effect says there is no unique representation of a dynamical system with multiple interacting variables. Moreover, any combination of simple transformations like rotations or other linear combinations of observation functions are also valid embedding variables. That is, mathematically, there is an infinite number of valid ways to reconstruct a system from multiple time series.

In practice, different combinations of variables amplify and reduce uncertainty in different ecosystem states. Take the 5 species competition model from (55)as an example. This toy ecosystem oscillates between generalists and specialists, and the dominate assemblage shows two alternate regimes (Figure 4.3 left). Per Takens's theorem and its multivariate generalization, these dynamics can be recovered directly from time-series data by embedding the data in a multivariate coordinate space. Two different choices of variables are shown in (Figure 4.3 middle, right). Although both reconstructions are equally valid according to mathematical theory (52, 53), they represent the system in different ways and can contain unique information. The reconstruction in the middle panel of Figure 4.3 resolves the system state in the blue regime quite well, but only resolves the red regime in the noise free ideal case. The reconstruction in the right panel does just the opposite. One consequence is that the dynamics leading to a regime shift are better identified by the currently dominant species.

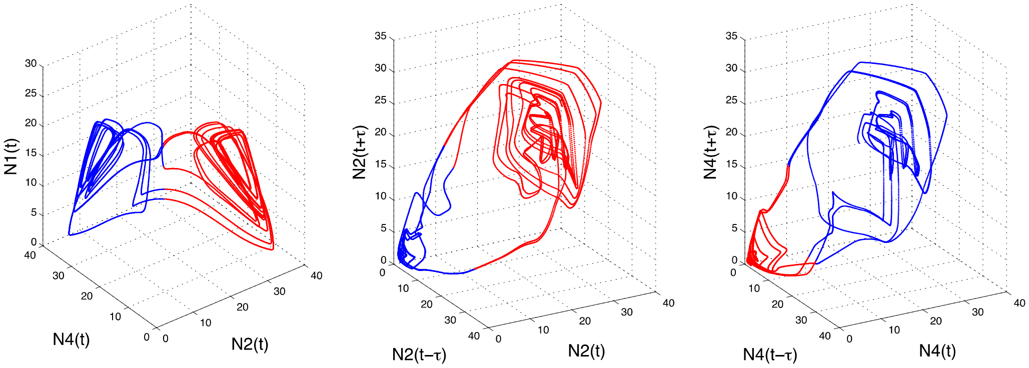


Figure 4.3— Comparison of different 3-dimensional projections of the chaotic 5-species competition model in Huisman et al. (55). Areas of the attractor where the N4 assemblage dominates are colored blue and areas where the N2 assemblage dominates are colored red. In principle either of these univariate attractors (middle, right) are 1-1 mappings with the native attractor (left) and can be used for EDM forecasting, but in practice the compressed areas of the attractor (blue for middle, red for right) have extremely high noise amplification. Note in all three cases, two additional coordinates are treated with EDM to fully unfold the system, but those additional dimensions cannot be represented graphically.

While this muddies the notion of what the “best” set of variables are to study a system, it also motivates multi-modeling approaches to multivariate empirical dynamic forecasting. Combining the predictions of multiple embeddings can smooth out forecast skill across areas of the attractor, improve forecast skill, and reduce uncertainty.

4.4.1 Multiview embedding

Multiview embedding (MVE) represents a simple but powerful approach. The details are presented in (3), but the basic principle is an extension of weighted nearest neighbor forecasting with simplex projection as described in 4.1.2.1. Instead of nearest neighbor $\boldsymbol{x}_{n(\mathrm{i})}$ getting weighting based on its distance from the forecast target $\boldsymbol{x}^* = \boldsymbol{x}(\mathrm{t}^*)$ on a single manifold as in simplex, the weighting in MVE is assigned based on how many multivariate models $\boldsymbol{x}_{n(\mathrm{i})}$ is the single nearest neighbor of $\boldsymbol{x}^*$. In practice, it is optimal for forecasting to only consider a limited number k of the top multivariate models ranked on forecast skill in a test set and a good heuristic is to take $k = \sqrt{m}$ where m is the total number of multivariate models considered.

Psuedo-code for the algorithm is as follows. Given n observational variables of the same system $\{X_1, X_2, \ldots, X_\mathrm{n}\}$, including the variable being forecast, X_i:

(i) Partition the time points of the data into an in-sample training set and an out-of-sample test set.

(ii) Construct the m possible E-dimensional variable combinations of l lags of the n observed variables.

$$m = \binom{nl}{E} - \binom{n(l-1)}{E}.$$

(iii) For each such embedding evaluate the simplex projection skill of forecasting Xi over the in-sample training set, and order these $\mathbf{M}_j$ by the measured in-sample forecast skill such that $\mathbf{M}_1$ has the highest forecast skill, $\mathbf{M}_2$ next, and so on.

(iv) Identify a target time point, t^*, in the out-of-sample test set.

(v) For each of the top k embeddings $\mathbf{M}_j$ ($j \leq k$), compute the Euclidian distance between the target vector $\boldsymbol{x}_{\mathbf{M}j}(\mathrm{t}^*)$ and all the library vectors, $\boldsymbol{x}_{\mathbf{M}j}(\mathrm{t})$.

(vi) Identify the single nearest neighbor of $\boldsymbol{x}_{\mathbf{M}j}(\mathrm{t}^*)$ and its corresponding time index, i.e. $\mathrm{t}_{\mathrm{nn}(j)}$ that corresponding to $\|\boldsymbol{x}_{\mathbf{M}j}(\mathrm{t}^*) - \boldsymbol{x}_{\mathbf{M}j}(\mathrm{t}_{nn(j)})\| < \|\boldsymbol{x}_{\mathbf{M}j}(\mathrm{t}^*) - \boldsymbol{x}_{\mathbf{M}j}(\mathrm{t})\|$ for all $\mathrm{t} \neq \mathrm{t}_{\mathrm{nn}(j)}$.

(vii) Estimate the future value of variable X from the target point as a weighted average of the single nearest neighbor in each of the top k embeddings.

$$\hat{X}_i(t^* + p) = \frac{1}{k}\sum_{i=j}^{k} X_i(t_{nn(j)} + p).$$

(viii) Repeat (*iv*) – (*vii*) for other target points in the out-of-sample test set.

The general idea is amenable to other specific implementations. For example, in (4) we adapted the principle to evaluate the binary prediction of “bloom” or “no bloom” in the red tide case study. There, the averaging in (vii) was replaced by a quorum “vote” such that a bloom was predicted if p% of the top k embeddings predicted a bloom.

4.4.2 Multivariate EDM with Randomized Embeddings

Work in neuroscience by Tajima (Tajima et al. 2015) introduced the idea of using random projection coordinates popular in branches of machine learning for empirical dynamic modeling. Tajima et al. used random linear combinations of univariate time lags to generate coordinates for EDM that were more robust to multiple time-scales than strict, sequential lags. The idea, however, generalizes to multivariate coordinates as well, and this insight provides solutions for applying EDM to high spatial, low temporal power environmental data.

If you have *N* observation functions $\{Y_1, Y_2, \ldots Y_N\}$ that embed a system (which can be any arbitrary mix of lags of different variables ala (53)), then generically any linear combination of the Y_i will also be a valid embedding coordinate. Thus, we can generate any arbitrary number of random project coordinates ξ_j

$$\xi_j = a_{1,j}Y_1 + a_{2,j}Y_2 + \cdots + a_{N,j}Y_N = \sum_{i=1}^{N} a_{i,j}Y_i$$

where the $a_{i,j}$ are random variables drawn from the same distribution. Tajima et al. (8) note that Gaussian and Unitary distributions are both reasonable choices. If we replace a coordinate Y_i with one of these random projection coordinates ξ_j, the new manifold **M'** will be diffeomorphic to the old manifold **M**, because the linear transformation between the two sets of coordinates is non-degenerate for almost all random draws of $a_{i,j}$:

$$f(M \to M') = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{1,j} & a_{2,j} & \cdots & a_{i,j} & \cdots & a_{N,j} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}.$$

In practice, it will not necessarily be true that all variables under consideration will in fact be valid embedding coordinates due to uncertain causal coupling. In this case, including non-interacting variables in the random project coordinates combine essential contributes noise. EDM forecasting methods like simplex and S-map are robust to noise up to a point, however, so the approach can be practical so long as the noise does not degrade the predictive signal to the point of non-detection.

Note that these randomized embeddings do not readily enhance predictive skill. Rather, they can instead aid inference in system identification and causality detection. In particular allow a first attempt to reconstruct trajectories before embedding dimension and causal variables are identified.

4.4.2.1 Greedy EDM model selection with random projection coordinates

Past EDM work with long time series has built multivariate models by replacing univariate lag coordinates with explicit variables. A similar approach can be used with randomized coordinates. This is motivated by two considerations. First, it is especially practical if the data do not allow for multiple time lags, such as the TDS benthic data for Case Study 2. In this sense it can be a technique to build predictive multivariate EDM models without univariate analysis. The second advantage is that for each observed variable, it is possible to generate a whole distribution of forecast skill at each stage of variable selection, and thus deciding on the single best variable at each stage can be done much more robustly because it is done with a multi-model approach.

The first step, however, is to determine a rough estimation of the necessary embedding dimension, E. This can also be done with random projection coordinates. Psuedo-code for the general algorithm is as follows. Given a target variable Y_1 and n-1 other observational variables of the same system {Y_2, Y_3, …, Y_{n-1}}, some of which may be time-lags of each other:

(i) Normalize all variables to have $mean(Y_i) = 0$ and $sd(Y_i) = 1$.

(ii) Construct E_{max} - 1 random projection coordinates from the n-1 observed variables, and renormalize the ξ_j

$$\xi_j = \sum_{i=1}^{n-1} a_{i,j} Y_i \, , (a_{i,j} \sim N(0,1))$$

(iii) Compute the simplex projection forecast skill (see 4.1.2.1) of $Y_1(t + tp)$ for the sequentially constructed embeddings that include the target X_0 and an increasing number of ξ_i: {Y_1}, {Y_1, ξ_1}, {Y_1, ξ_1, ξ_2}, …, {Y_1, ξ_1, ξ_2, …, ξ_{Emax-1}}.

(iv) Repeat ii-iii for many ensembles (e.g. 500) randomly generated coordinates ξ_j.

(v) Identify the number of coordinates E^* that maximize median forecast skill across ensemble replicates.

In practice, if there are candidate variables included in the X_i that ultimately are not causal, then adding additional random coordinates can continue to slightly improve forecast skill for ever increasing E due to incrementally increasing averaging of noise. We suggest a heuristic of identifying a lower dimensional E^* such that the forecast skill at E^* is at least 95% that of the forecast skill at E_{max}.

Once a practical embedding dimension can be found, then greedy model selection can be applied. Psuedo-code for the general algorithm is as follows (using the same designation of target Y_1 and n-1 other Y_i variables as above)::

(vi) Normalize all variables to have $mean(Y_i) = 0$ and $sd(Y_i) = 1$.

(vii) Construct E_{max} - 2 random projection coordinates from the n-1 observed variables, and renormalize the ξ_j

$$\xi_j = \sum_{i=1}^{n-1} a_{i,j} Y_i \, , (a_{i,j} \sim N(0,1))$$

(viii) Compute the simplex projection forecast skill (see 4.1.2.1) of $Y_1(t + tp)$ for the embeddings including Y_1, one candidate Y_i, and E^*-2 random projection coordinates: {Y_1, $\boldsymbol{Y_2}$, ξ_1, ξ_2, …, ξ_{Emax-1}},{Y_1, $\boldsymbol{Y_3}$, ξ_1, ξ_2, …, ξ_{Emax-1}}, …, {Y_1, $\boldsymbol{Y_n}$, ξ_1, ξ_2, …, ξ_{Emax-1}}. (Bolding used for emphasis).

(ix) Repeat (ii)-(iii) for many ensembles (e.g. 500) randomly generated coordinates ξ_j.

(x) Select Y_i that gives the highest median forecast skill.

(xi) Repeat (ii)-(v) but replacing a random projection coordinate with the Yi, and removing Yi from the candidate variables.

To increase statistical inference, the same ensemble of random projection coordinates can be used for each candidate embedding variable.

4.4.2.2 Pairwise multivariate forecast analysis

This general idea can be used in a similar mode to model selection, but instead to compare forecast skill of similar embeddings to help untangle information about environmental drivers (9, 56). The idea is that if X_1 and X_2 are both purported drivers of Y_1, then whichever gives better prediction skill when used in embeddings to predict Y_1 is a more proximate driver. This can be tricky for comparing just a single set of embeddings, however, e.g. a univariate embedding of Y_1 with just a single lag of one of the drivers, $\{X_1(t), Y_1(t), Y_1(t - \tau), \ldots, \mathrm{Y}_1(t - (E - 2)\tau)\}$ and $\{X_2(t), Y_1(t), Y_1(t - \tau), \ldots, \mathrm{Y}_1(t - (E - 2)\tau)\}$, and again cannot be readily applied to cases where using many time-lag coordinates isn't possible.

Alternatively, this can be done using random projection coordinates using a similar algorithm to that described previously for determining embedding dimension with random coordinates, but instead of comparing embedding with different numbers of randomized coordinates, we compare $\{X_1, Y_1, \xi_1, \xi_2, \ldots, \xi_{E-1}\}$ and $\{X_2, Y_1, \xi_1, \xi_2, \ldots \xi_{E-1}\}$. This procedure was applied to evaluate the reconstructed stratification observations for Case Study 1 (see 5.3.2.4).

4.5 EDM with Spatial Lags

The fundamental idea of using spatial lags for empirical dynamic modeling is that in nonlinear spatio-temporal systems, variables across space constitute different observation functions of the underlying dynamics. Thus, just like any other multivariate observation, the observation of a state variable $Y(\mathbf{x},t)$ at a nearby location, $Y(\mathbf{x} + \delta, t)$ can be used as an embedding coordinate. This idea has been demonstrated in principle with simple statistical physics models (57), but has not been practically implemented for prediction in ecological systems. In principle, one might be able to fully reconstruct dynamics from spatial lags of a single variable.

Implementation of EDM forecasting with spatial lags can be carried out with the baseline functionality of *rEDM* and *pyEDM* packages together with some necessary data pre-processing in basic R or Python to translate the spatio-temporal information of samples. This involves first generating a description of neighbor relationships between points, then second translating these neighbor relationships into data series to append to an *rEDM* "block".

4.5.1 Spatio-temporal data processing

If the spatial data conform to a densely populated rectangular grid, (e.g. many remote sensing data products), the task of translating spatial data into neighbor relationships is essentially trivial. However, this was not an option for applying spatial lags to the visual towed-diver benthic surveys. The surveys track a single depth contour on islands and atolls, and thus generally have a ring structure. There are several alternatives but the ultimate goal is to produce a graph data structure that contains the spatial and temporal relationships between measurements so that blocks with different combinations of spatial lags can be easily generated on the fly. Code for generating blocks with mixes of spatial and temporal lags is included in Appendix A, to generalize the functionality of the rEDM 0.7.4 function *make_block*().

4.5.1.1 General Approach

Gridding the data through spatial binning is one option. This is relatively trivial for geolocated measurements. However, unless the grid size is larger than the distance between individual transects, there will be substantial observation error introduced by the spatial binning. In the case of coral, we know that important dynamics happen on scales substantially smaller than the resolution of the sampling protocol, and thus did not wish to further reduce the spatial resolution. There are certainly other cases where this would be less of a concern, however, and the procedure of gridding is widely applicable. Moreover, gridding the data was able to qualitatively reproduce the spatial lag analysis.

Another option is to use graph algorithms, such as implemented through *iGraph* in R. The idea is to create a graph where each sample location is a vertex, and vertices are connected if they are separated by a set distance ± tolerance. For annular reef geometry, it is then possible to select a subgraph of this spatial neighbor graph so that each vertex has uniquely defined neighbors. This was particularly relevant for the benthic TDS, since small islands and atolls were often circumnavigated twice on the sampling day. From the spatial neighbor graph, identify the convex hull of the island, then find the shortest paths on the neighbor graph between sequential points on the convex hull.

4.5.1.2 Approach for Case Study 2

In the case of the benthic TDS studied for Case Study 2, observations are geolocated and time-stamped, but observations in different years are not explicitly connected as in a convential time series. Thus, the first step was to identify temporal relationships between observations of the same stretch of reef (within 50m tolerance) separated by 2 years. Then, to identify spatial relationships, the observational design and metadata provided a shortcut. Spatial segments in the benthic TDS correspond to 5-minute segments that stretch ~150m, with each dive lasting 50-minutes. Thus, sequential dive segments from the same dive in the TDS data automatically correspond to spatial neighbors along depth a depth contour. Then, all that remains is to connect start and end segments of different dives (same year, same island) that are within the distance tolerance, 200m ± 50m.

4.6 Static Cross-Mapping

Conventional CCM is based on univariate attractor reconstruction using time lag coordinate embeddings. CCM is a pairwise test for causality but applicable to systems that cannot be understood in a piece-meal, reductionist manner. However, the need to use time-lag information to capture the underlying multivariate state-dependence means that conventional CCM is best suited to temporally rich data sets. Combining spatial replicates can be used to overcome this to some extent, as shown by Clark et al. (38). However, the approach is still limited to having time series at each site at least as long as the number of dimensions needed to unfold dynamics. In the towed-diver surveys we seek to study for Case Study 2, many sites were only revisited a single time.

However, instead of using time-lag coordinates of a single variable to unfold the system attractor, it is also possible to recover the attractor multivariately, without taking time-lags. So long as the system is dissipative—that is, the important dynamics have lower dimensionality than the number of state variables—the system can be embedded from a subset of observables. Moreover, even if the system is not fully embedded, nearest-neighbor forecasting can still be accurate for all but a few areas of the manifold dynamics.

4.6.1 Identifying additional variables from existing causal understanding

This multivariate approach to cross-mapping is simplest to execute when many of the observed variables are already understood to interact. In that case, the manifold can be reconstructed from the known interacting variables, and additional candidate variables can be tested by evaluating cross-mapping skill from this reconstruction to the candidate. This is well suited to a common problem in ecology: detecting environmental drivers of community dynamics. In this case, data on different taxa in a community or ecosystem can be used as coordinate axes to multivariately cross-map candidate environmental drivers.

4.6.2 De novo causal testing

When there is not existing understanding of causal relationships, one must face the question of which variables to combine together to create multivariate attractors for cross-mapping. Random embeddings (see 4.4.2) provide an agnostic solution that lets multivariate cross-mapping be done in a nearly pairwise manner. If we label the pair of variables to test X_1 and X_2, and the remaining variables ($Y_1, Y_2, \ldots, Y_N$), we can construct E-1 coordinate variables out of random linear combinations of the Y_i. Then, we test multivariate cross-mapping from X_1 and these E-1 random projection variables to X_2.

4.7 Data Imputation with EDM

Throughout environmental science, it is more rule than exception that the most commonly made measurements are the ones that are easiest to make, whether or not they are most fundamental to the system. Chlorophyll is measured much more often than phytoplankton cell counts or species biomass are made, much less true rates of carbon fixation/primary production. This is a strength of EDM, because attractor reconstruction allows researchers to make the most of what they've measured, right up to using just time-lags of a single variable to stand in for the underlying, multidimensional nature of the dynamics. However, ecological monitoring is progressing at an astonishing pace. Measurements that used to require a days-worth of lab work to process can now be issued in near real time from moored automated sensors. This is especially critical in understanding non-stationary futures as relationships between commonly measured indirect variables and underlying mechanistic variables may be changing and obscuring insight.

With this in mind, we have developed a method for reconstructing historical time series of mechanistic variables from a set of new measurement, supplemented by long-term monitoring of related variables. The innovation is largely in just recognizing the potential for existing EDM techniques to be used for a new purpose. Call the core variables measured over the long-term history {Y_1, Y_2, Y_3, …} and the new measurement variables be {Z_1, Z_2, Z_3, …}. So long as there is a sufficient period of overlap between the measurements of the historical variables Y_i and the novel variables Z_i, then a multivariate model for predicting any Z_i can be made from the previously described methods (4.24.2) using the overlapping period as the training set. Then out-of-sample prediction of Z_i can be made based on the values of the predictor Y_i in the historical period.

4.8 Jacobian estimation and dynamic stability

The Jacobian matrix describes the local dynamics of a dynamical system, whether globally linear or nonlinear. This was illustrated in Figure 3.2. Multiple properties of the matrix are connected to the stability of dynamics, including the largest eigenvalue, determinant (or trace in continuous time), and the singular values. Properly speaking, the singular values are the most direct measures of the divergence of trajectories in Euclidian measure. However, singular values are not scale-invariant, and this is a challenge for environmental systems where interacting variables don't share a common scale. The largest eigenvalue is bounded by the largest singular value, and hence is a more conservative estimate of the divergence of trajectories, but is not dependent on the scales of individual coordinates (58). Thus, we focus on the largest eigenvalue of the Jacobian matrix as a practical indicator of dynamics stability and warning sign of rapid change.

The approach has three important advantages over previously popularized early warning signs (EWS) (59) especially relevant to our project. First, it can apply equally if reefs exhibit true alternative stable states or more general alternative dynamic regimes. Second, there is a clear benchmark for transitions ($\lambda=1$) that comes directly from the mathematical theory, while previous EWS relied on qualitative changes in parameters, like "rising" autocorrelation. Finally, by building a picture of attractor dynamics from spatial replicates, the approach can be applied to very short time series.

4.9 Data

4.9.1 Case Study 1:

The core data for Case Study 1 are manual measurements of chlorophyll, nutrients, water temperature, and salinity. These were supplemented by meteorological measurements. These core data are described in detail in the published results (4).

4.9.1.1 Wire-walker Automated Profiler

The "wire-walker" automated profiler uses passive wave energy to perform depth transects at a speed of approximately 20 meters per minute with a payload of automated sensors. A wire-walker was deployed in 90 meters of water directly off-shore of the SIO Pier from September 2016 – September 2017 (60), and included temperature, conductance/salinity, chlorophyll-a fluorescence, particle backscatter, and intermittent 3-dimensional water particle velocities. The sensors observe the rich variability in depth and time that are aliased by daily and weekly manual measurements. The resolution of density with depth allow for straightforward calculation of pycnocline slope. We focused on stratification averaged across the euphotic zone, so were interested in essentially $m_{pycn} = (\rho(0m) - \rho(90m))/90m$. To reduce noise contributions by the two extreme measurements, we first fit the depth data to a smoothing spline, then calculated the slope from the smooth measurements. Additional indicators of stability were computed from the Thorpe approximation including turbulent dissipation rate and length scale. Additionally, although we did not directly study the chlorophyll-a fluorescence measurements, the wire-walker observations unveil the dependence of pier observations of physics and biology alike on a strong mode-2 internal tide (~6 hour period of oscillation). This is exemplified in Figure 4.4 for a 10-day period in April 2016.

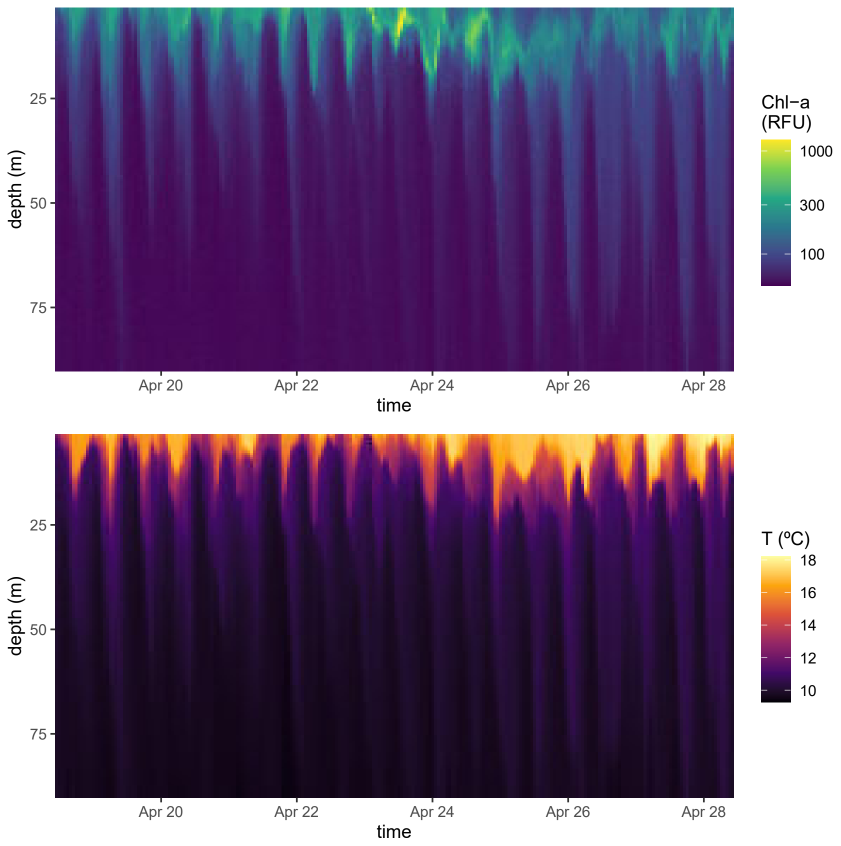


Figure 4.4— 10-day period of wire-walker measurements during April 2017 with an ephemeral high chlorophyll event. time-course of depth resolved measurement in 90 meters of water off the end of Scripps Institution of Oceanography Pier. Chlorophyll-a fluorescence is shown on *top*, water temperature on *bottom*.

4.9.2 Case Study 2:

The core data for Case Study 2 are benthic cover characterizations from a visual towed-diver survey (TDS) program run under the Pacific Reef Assessment and Monitoring Program (RAMP) by NOAA between 2000 and 2012 (https://inport.nmfs.noaa.gov/inport/item/35618). A diver visually estimates percent cover during a 5-min across several broad categories, while the boat attempts to follow the 15m depth contour of the island. Cover categories used for the entire survey program are % Live Coral, % Soft Coral, % Stressed Coral, % Macroalgae, % Coraline Algae, % Sand, and % Rubble. Further details of the standard operating procedures are described in (61).

Thirty-two U.S. Pacific islands and atolls were revisited at 2-year intervals, and most islands had between 2 and 4 sequential observations. The benthic cover observations were geolocated to ~150m segments (5 minutes of diver tow), and thus the data set contains over 4,000 observations of benthic change in patches across 32 reefs over 2-year intervals. The observations include 7 islands and atolls with current or former DoD presence.

Satellite derived physical variables were previously synthesized for study with this data set by Gove et al. (46). These include 2-year statistics on average and maximum sea surface temperature, as well as wave data. We considered the temperature measurements as predictor variables for our multivariate EDM analysis, but refined the wave calculations by Gove et al. to resolve differences in wave energy across reef segments in the same islands. We used the Wave Watch III model outputs provided by NOAA. These are split over the RAMP TDS period by the historical reanalysis (1979-2009) and modern analysis (2006-present). Both give outputs at 3-hour intervals, although the modern analysis is done at a slightly higher resolution spatial resolution (0º30’ x 0º30’). Gove et al. describe calculating wave energy flux, WEF, from the height and period, but the NWW3 model also gives the primary angle of wave direction from true north, θ_{PW}. Note that this is the angle from which the waves are coming, rather the angle of their propagation.

To determine how direct a reef segments exposure was to wave energy at any given angle, we made the rough approximation of islands as points located at their centroid. Shape files of island coasts were obtained from (http://www.soest.hawaii.edu/wessel/gshhg/) described in (62), and we used the high resolution (“h”) shape files, version 2.3.3. The coastlines were used to identify the centroids of the study islands, from which we could calculate the angle from true north of individual reef segments, relative to the island centroid. The directed wave energy at segment *k* was thus approximated by:

$$DWEF(k) = WEF(1 + \cos(\theta_{PW} - \theta_k))/2$$

If the wave energy is head-on at *k*, then DWEF(k) will be the full WEF in that grid square of the NWW3 model. If the incident direction is exactly opposite the island, DWEF(k) will be 0. The NWW3 model allowed for a directional wave flux to be calculated for 3-hr segments between 2000 and 2021. We considered the maximum of DWEF, mean of DWEF, and standard deviation of DWEF as driving variables. We also considered the angular variance of directed wave energy using the *circular* package for R (https://cran.r-project.org/web/packages/circular/circular.pdf).

4.9.3 Cedar Creek LTER

Data used for Cedar Creek stability analysis were obtained from the LTER database, then processed according to previous EDM analysis by Clark et al. (63). Above ground biomass of plant species were aggregated by guild. Our analysis focused on C3 grasses, C4 grasses, and woody shrubs based on the previous EDM analysis.

5 Results and Discussion

5.1 Software Development

Under this grant, a standardized set of applied computational tools for empirical dynamic modeling was developed. Code packages are now publicly available in R, Python, and C++, and are quickly finding a large user base. Since initial launch in March 2016, the **rEDM** package had more than **19,000** downloads (>1000 just in March 2020) from the central CRAN server; and **pyEDM** through the central PyPI server, had more than **76,729** downloads in the first year of its launch (~20,000 just in May 2020). The grant allowed us to develop documentation to communicate the essential foundations for new users wishing to apply EDM to their research. The code was also put through a formal code review, and the report from that is included in the Appendix.

5.2 Key Model Demonstrations of Novel Approaches

Model demonstrations with known structure and behavior were developed to further a number of tactical research goals of this project. Many of these are already published in the literature. These are summarized in Table 5.1. Additional key model demonstrations that are not yet published are described in more detail below.

Multimodal inference with EDM.	Ye & Sugihara, 2016 (3)
Environmental driver identification with seasonality.	Deyle et al., 2016 (56)
S-map estimation of Jacobian coefficients.	Deyle et al., 2016 (22), Cenci et al. 2020 (64)
S-map estimation of Jacobian stability.	Cenci et al. 2020 (64)
Multivariate EDM model selection.	Deyle et al. 2016 (22)
EDM Forecast uncertainty	Cenci et al. 2019 (65), Cenci et al. 2020 (64)

Table 5.1— Summary of model demonstrations conducted during grant in publications.

5.2.1 Static Cross Mapping

In molecular biology, large datasets are often cross-sectional, collected across huge batches of cells with no repeated temporal measurement. This offers an extreme context for extending EDM to low-temporal power data sets in preparation for analysis of pacific coral reefs. Thus, we used the classic Repressilator gene circuit (66) to demonstrate our innovated approach to detecting causality in the absence of time-series, static cross-mapping (see description in 4.6).

In the Repressilator, there are three genes (*a*, *b*, and *c*) whose products (*A*, *B*, and *C*) each inhibit the next gene in the cycle (Figure 5.1A). An auto-inducer *S* is also included, which provides delayed feedback. Simulating this system produces with the appropriate parameters produces chaotic oscillations, evident in the time-series of the individual genes (Figure 5.1B). One could reconstruct the attractor by doing a lag-coordinate embedding with any single gene time-series, but it is also possible to reconstruct the attractor by taking each time-series as coordinates. In fact, even though there are 7 variables in the model (3 genes, 3 protein products, and the auto inducer), the attractor is approximately

3-dimensional and can be recovered just from using the three gene time series. This is shown in Figure 5.1C for coordinates <*a*(t), *b*(t)*,* *c*(t)>.

If we take a sample of the model data at random time points, we no longer have time-series data. This is equivalent to experiments that provide a single snapshot of gene expression for each cell measured. Even without a time-sequence to the data, we can still embed each individual sample point in the multidimensional space. This is shown in (Figure 5.1D). The geometry of the manifold is still preserved, even though there is no immediate way to reconstruct the flow between points.

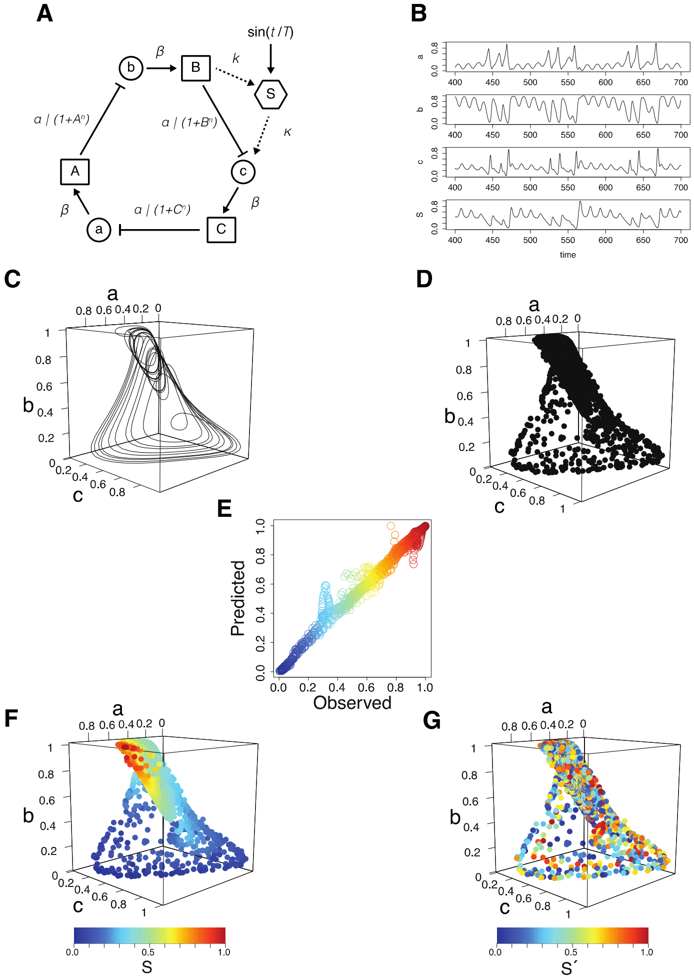


Figure 5.1— Static cross mapping demonstrated in the Repressilator gene circuit (A,B). Even when data with underlying nonlinear dynamics (C) have no temporal evolution observed (D), manifold geometry can still be used to predict causally associated variables (E,F).

With just the manifold geometry, the value of other interacting variables can still be cross-mapped from the manifold (just like in univariate CCM). Panel E shows that cross-mapping from the <*a*, *b*, *c*> manifold to *S* gives nearly perfect predictions. This is possible because the concentration of the auto-inducer *S* (shown as the point color) is well-determined by the state on the <*a*, *b*, *c*> manifold. For contrast, panel F shows the value of an unrelated variable *S'*—in this case, the concentration of auto-inducer from a separate (independent) realization of the Repressilator. *S'* shares no causal information with genes *a,* b, and *c*. Hence, the value of *S'* is not well-determined from the state on the <*a*, *b*, *c*> and cannot be cross-mapped from the static <*a*, *b*, *c*> manifold.

5.2.2 Critical Transitions and Dynamic Stability

We tested the ability of the S-map Jacobian estimates to characterize and predict threshold behavior and critical transitions on two models. The first demonstration used the seasonal phytoplankton model described by Huppert et al. (67). Chaotic dynamics modulate the size of the initial spring bloom varies strongly from year to year, as well as produce a second bloom in some but not all years. The analysis (Figure 5.2) demonstrates how small changes in nutrient concentration in spring (February-April) can lead to disproportionately large differences in primary bloom size and the presence or absence of a secondary peak.

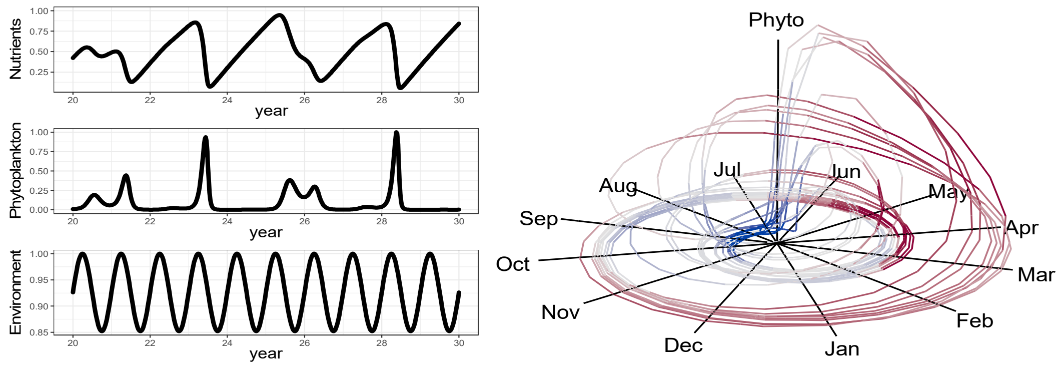


Figure 5.2— Dynamic instability in a seasonal phytoplankton model. (Left) The model produces variable season phytoplankton blooms, including intermittent secondary blooms. (Right) Dynamics are shown on an attractor in polar coordinates, where height designates phytoplankton abundance, radius denotes nutrient supply, and the polar angle indicates season. The discrete-time determinant of the Jacobian is calculated from EDM, and shows areas of dynamic instability (red) and contraction (blue). Grey sections of trajectory are neutrally stable (det(J)=1).

We also investigated the ability of EDM to predict population collapse in a fishery model. The model is a generalization of the widely used Ricker model, but includes a term for population depensation so that collapse occurs when the population is below a minimum viable threshold. In the simulation (Figure 5.3), fishing mortality gradually increases until the population is fished to the point of collapse. The largest eigenvalue of the Jacobian estimated with EDM (see section 4.8) identifies growing instability well before the "point of no return" is reached.

These demonstrations highlight the ability for EDM calculations to predict radical change in a system even without historical analogues. The calculations capture dynamic instability both in a fully deterministic system (Figure 5.2) and a system with exogenous stochastic forcing (Figure 5.3).

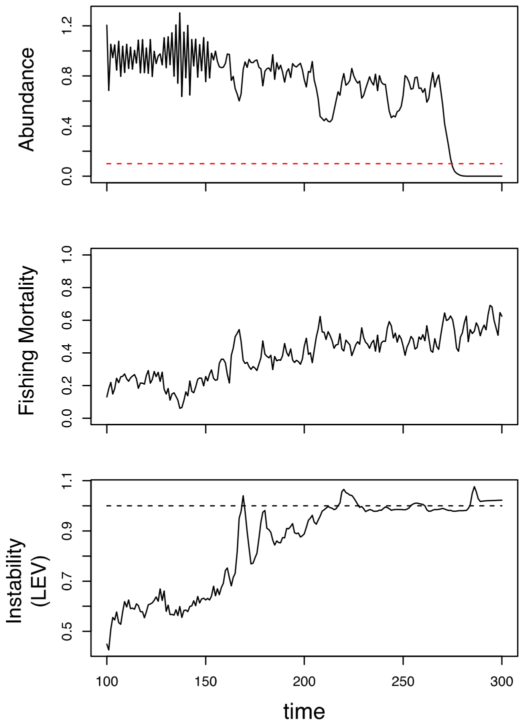


Figure 5.3— Rising instability predicts critical transition in a fishery collapse model. As fishing mortality increases (middle), the population (top) approaches the depensation point where it can no longer replenish itself. The largest eigenvalue calculated with EDM (bottom) captures the loss of stability well in advance of collapse, despite collapse having no historical analogue in the EDM analysis.

5.3 Case Study 1: Red Tides

5.3.1 Historical Analysis

Core results from the first phase of the analysis for the red tide case study were presented in McGowan et al. 2017 (4). These results are briefly discussed. The first step of results identified evidence of nonlinear dynamics in the chlorophyll-a time series (Fig. 3 in (4)), as well as statistical patterns between chlorophyll-a and purported causal variables that indicated multiple conditions for blooms that were necessary but not sufficient conditions (Fig. 4 in (4)). These patterns were not generally matched by linear correlations, however (Table 1 in (4)). Nevertheless, convergent cross mapping revealed significant driving by a range of hypothesized drivers (Table 1 in (4)), including nitrate, nitrite, silicate, water temperature, water density, and longshore wind speed.

Combining identified drivers in multivariate embeddings showed strong in-sample forecast skill. However, the ranking of possible embeddings based on skill was not strongly stationary under cross-validation tests. That is, the relative in-sample forecast skill of different embeddings did not closely match the relative out-of-sample forecast skill, although there was a clear pattern (Fig. 5 in (4)). One explanation is that blooms can occur from multiple combinations of observed drivers, which is consistent with understanding of dinoflagellate blooms globally. Indeed, different embeddings showed the ability to predict different bloom events (Fig. 8 in (4)). For that reason, we developed a multi-modal approach to produce forecasts, with a tunable sensitivity based on the relative importance of high true-positives and low false-positives (Fig. 9 in (4)).

5.3.2 Future Climate Scenarios

Iterative forecasting with S-map was developed in this grant as a tool for assessing changes in dynamic behavior (e.g. frequency or magnitude) of threshold events like red tides under non-stationary climates. In the ideal case, outputs of physical models for future conditions can be piped into EDM forecasts. However, the results are necessarily only as reasonable as the predictions of the physical model. As an alternative, non-stationarity can be simulated by making perturbations to the historical observations of drivers, such as offsetting the mean value or scaling the time series by a percentage. This provides a climate sensitivity analysis that is independent of individual physical forecasts.

5.3.2.1 Evaluation of ROMS Future Conditions

We analyzed output from NOAA-GFDL biogeochemical model COBALT developed by Curchitser and Dussin at Rutgers University that downscaled the FDL ESM2M RCP8.5 future global climate projections to a 7km x 7km grid in the California Current. The model is similar to what was used for a historical comparison of chlorophyll dynamics from 1996-2006 (68), but was fully coupled to the atmospheric model. However, the COBALT output we used was run with a fully coupled ocean and atmosphere. That is, the simulation is not forced by historical atmospheric observations. Thus, the ROMS outputs should be regarded as a simulation of realistic historical dynamics but not a reproduction of the particular historic trajectory. Therefore, a direct comparison of the exact historical and ROMS time series doesn't make sense (unlike in (68)). However, if the ROMS is producing realistic behavior then the statistics of the simulated variables should match the statistics of observations.

Figure 5.4 shows a comparison of modeled historical and future behavior of the five ROMS variables that match the SIO Pier measurements we used for empirical modeling. In all cases the changes predicted by the ROMS model from historical conditions (red) to future conditions (green) under the RCP 8.5 scenario are substantially smaller than the disagreement between the historical modeled oceanography (red) at the 7km x 7km grid-square containing the SIO pier and the historical empirical observations at the SIO the pier (blue). SST and silicate are the two variables closest to having the simulated ROMS distributions line up with the historical observations.

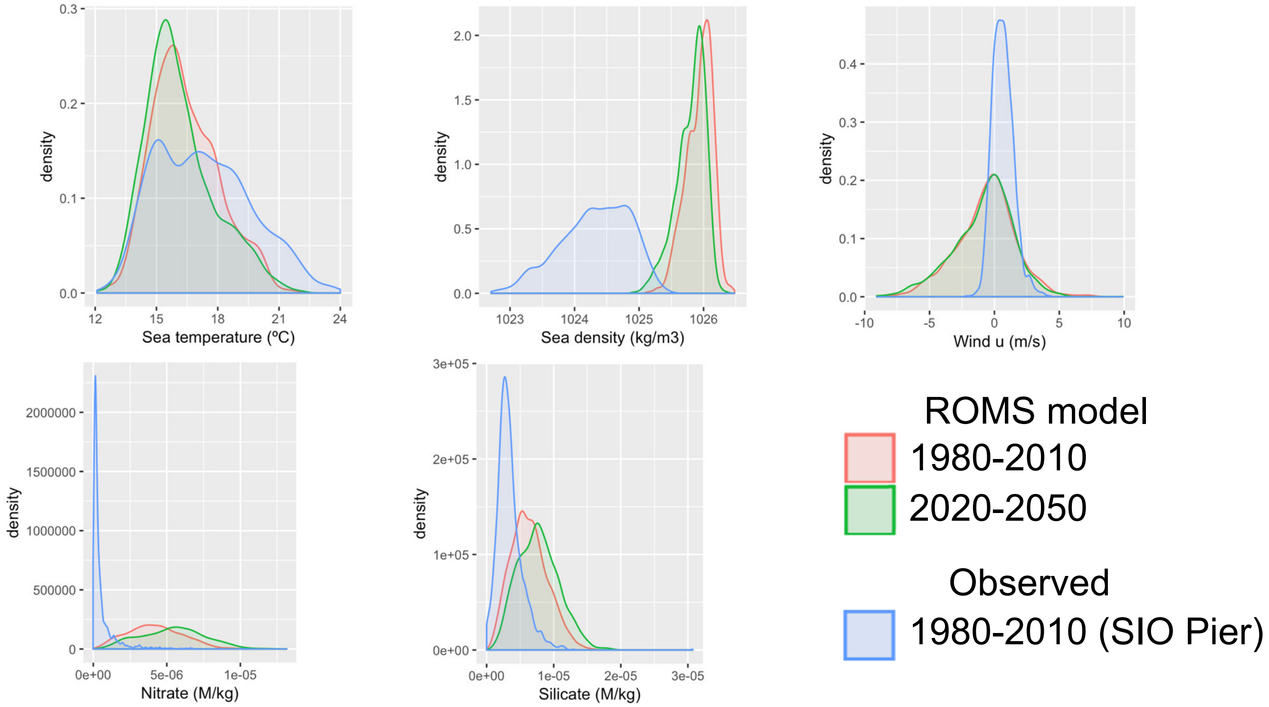


Figure 5.4— Comparison of ROMS modeled oceanography to empirical observations at the SIO pier. Density plots of the distribution of values are shown for five key environmental variables for ROMS historical (1980-2010) ROMS predictions (red), 30-year future (2020-2050) ROMS predictions under RCP 8.5 warming scenario (green), and historical (1980-2010) observations at the SIO Pier used in (McGowan et al. 2017).

The obvious cause of this extreme disagreement is a matter of scale. Although the blooms can stretch hundreds of kilometers up and down the coast, the pier measurements are made just outside the surf zone, where things can be quite different from 7km offshore; this model behavior might match much better the conditions further off shore, but do not realistically capture physics and chemistry as we measure at the pier, which means we can't reasonably simulate the inputs to our empirical models built from the pier data. This speaks to the difficulty in resolving and projecting the oceanography in near shore environments. While this ROMS simulation is quite sophisticated and the best available science at the time of the project, it doesn't resolve the environmental variability we require.

Note a slightly unexpected prediction of the Rutgers COBALT model: the sea temperature off the coast of La Jolla is simulated to be lower on average in the future (2020-2050) than historically (1980-2010) (Figure 5.4 top-left panel). This certainly goes against our expectation that average ocean temperatures will rise under the 8.5 "business as usual" climate scenario. At the same time, ocean temperature in the Southern California Bight is the product of complicated oceanography. Cold water is transported from the north by the California Current but there is complicated recirculation, upwelling, and occasional intrusion of tropical water from Baja Mexico (69), not to mention eddy-scale features and smaller that shape oceanography but occur at scales below the ROMS resolution. Increased direct influence of the California Current or upwelling cold both plausibly drive temperatures in La Jolla lower in the future.

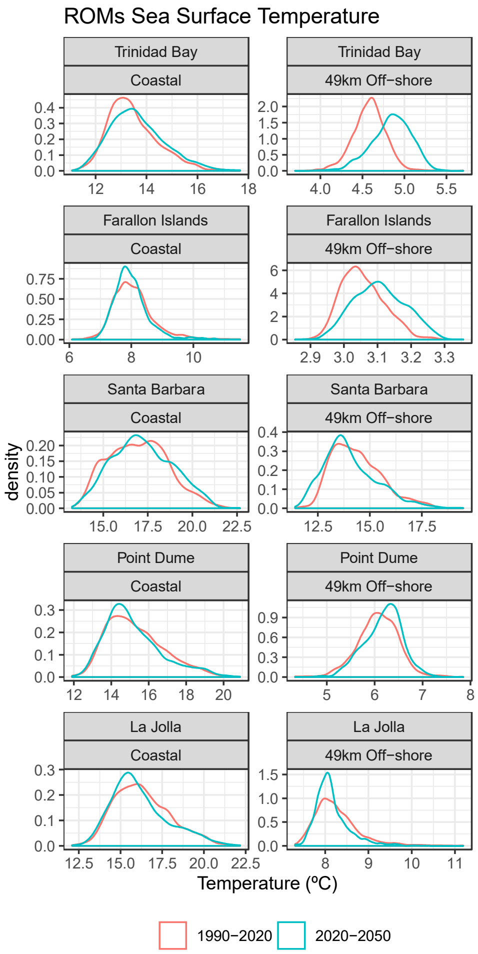


Figure 5.5—ROMs simulated sea surface temperature across coastal stations along the California coast order from North to South. Temperatures are split between simulations of historic conditions from 1990-2010 (red) and future conditions from 2020-2050 (blue).

Figure 5.5 puts the La Jolla projections in the context of temperatures at coastal stations further up the coast. At the most northern stations, Trinidad Bay and Farallon Islands, the oceanography is driven much more simply by the main California Current that sits just off the coast. Indeed, there is a clear prediction of warming water as we expect for global averaged ocean temperature. Thus, the temperature shifts simulated in the ROMs model are reasonable. Detailed analysis of the ROMs output has not yet been made available to us at the writing of this report. Until such times as our limited understanding can be clarified, our working hypotheses for our use of the simulations was that even if the absolute levels do not match near-shore observations, we can still trust the relative changes predicted by the model for La Jolla: decreasing temperature and density, but increasing nutrient supply. So, while we cannot reasonably directly drive our EDM models with the outputs (although it is technically possible), we can still use the ROMs to interpret a climate sensitivity analysis.

5.3.2.2 Long-term simulation of multiple climate scenarios

Climate sensitivity analysis was performed using long-term simulation with EDM and basic statistical perturbations to the historical time series of environmental conditions. Given our conclusions presented in McGowan et al. (4) that there was no single optimal EDM model, we repeated the long-term simulation calculations for the same top 100 embeddings used for Figures 8 and 9 in the published paper (4). Figure 5.6 and Figure 5.7 show the results for changes to the six key environmental variables, sea density, sea temperature, nitrate, nitrite, silicate, and u-directional wind. Changes in bloom frequency under sea temperature and silicate scenarios show coherence across possible multivariate models, while there is more disagreement between multivariate models for the other variables.

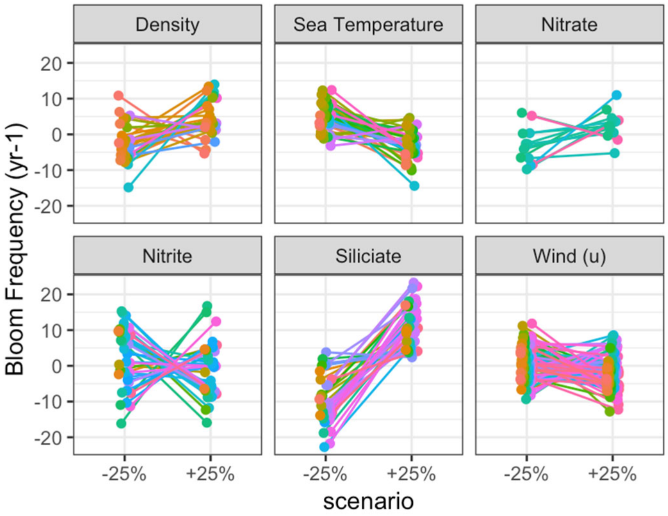


Figure 5.6— Multi-model predictions of bloom frequency under long-term simulation of future environmental scenarios. Changes in bloom frequency (chl > chl_{bloom}) are shown for hypothetical changes in each of the 6 key driving variables.

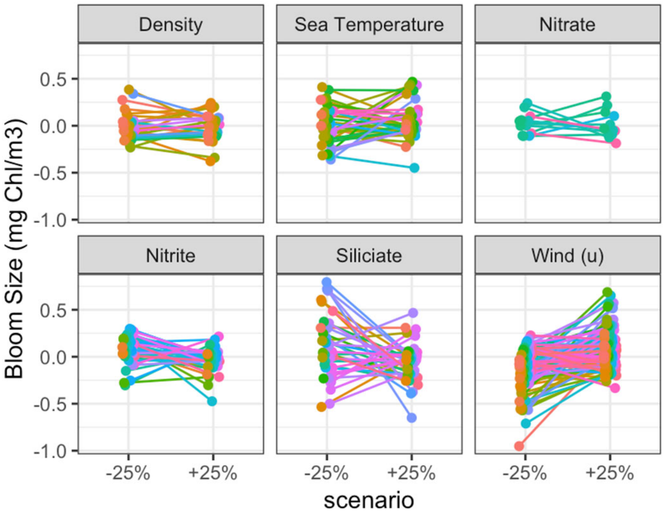


Figure 5.7— Multi-model predictions of bloom size under long-term simulation of future environmental scenarios. Changes in the magnitude of blooms, **E**(chl | chl > chl_{bloom}), are shown for hypothetical changes in the same 6 key driving variables as Figure 5.6.

5.3.2.3 Synthesis

SST and Silicate showed both the most definitive predictions of the EDM sensitivity analysis and the closest match between the available oceanographic model. Both decreasing temperature and increasing silicate showed to increase red tide frequency in the sensitivity analysis. Thus, our best prediction from synthesizing these results is that there will be an increase in red tide frequency along the San Diego coast over the next 30 years. At the same time, looking at the average size of blooms simulated under the sensitivity analysis (Figure 5.7), there is some agreement across multivariate models that increasing silicate will mean the increased bloom frequency will lead to more smaller blooms.

More generally, this ensemble approach to long-term scenario simulations shows how we can explore the sensitivity of extrapolation to changes in assumptions of models (a key piece of the tactical objective as articulated in 2.2.4). Here, the multi-modal approach should agreement across models for temperature and silicate, but diverging predictions about changes in wind and nitrogen compounds. Our understanding is that relationships between indirect drivers and direct drivers make models that utilize indirect variables for prediction more sensitive to non-stationarity. The robust forecasted outcome of change in temperature and silicate suggest these two variables are the least convolved with indirect effects and are the best to infer future behavior.

Nevertheless, our interpretation of the physical variables in McGowan et al. (4) in light of basic algal ecology is that they are indicators of a stratified or stable ocean. Thus, while the long-term scenario simulations show a uniform effect of temperature, we are motivated to clarify the simulations by clarifying the mechanistic variables.

5.3.2.4 Imputing Water Column Stability

Conceptually, surface conditions can contain information about subsurface physical structure in the ocean. If surface density is low, there can be a much greater difference between surface density and density at 90 m depth. This might appear as linear correlations, low density predicting high stratification, but the relationship need not be independent of ocean state. Thus, we investigated nonlinear multivariate prediction to see if surface conditions could predict subsurface physical structure. Initially, we focused just on wire-walker data and high frequency wind measurements so that we could maximize the high temporal resolution of the automated observing system. We focused on predicting the average slope of the pycnocline over the 90 m of depth (see 4.9.1.1) from surface density, surface sea temperature, surface salinity, and wind speed. Although surface density has a strong correlation to pycnocline slope (indicated by the red dot in Figure 5.8 at θ=0), the prediction was substantially improved by nonlinearly incorporating multiple predictors. Using a greedy search algorithm, the optimal prediction was found with surface density, salinity, and wind. Since density is computed as a function of temperature and salinity, it is unsurprising that including all three of temperature, density, and salinity did no better (and slightly worse even) than just two.

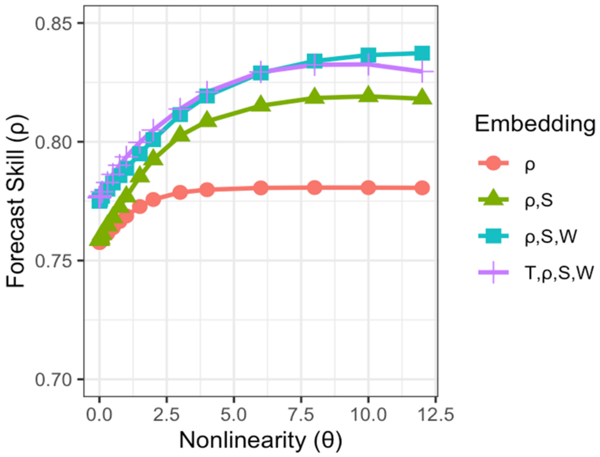


Figure 5.8— Multivariate cross-prediction of pycnocline slope from surface variables. Density, salinity, temperature, and wind were all considered as predictors.

This multivariate relationship only connected surface measurements by the wire walker to measurements across depth. To utilize this relationship for analyzing historic blooms, it was necessary to also look at the relationship between surface wire walker measurements and the pier measurements. These measurements have a different frequency, thus we used window averaging to smooth the wire walker measurements bad at ~15 min period to a daily time scale. These measurements showed very high correlation, but departed slightly from a strict 1:1 equivalence. The daily averaged wire walker measurements of temperature, for example, where systematically slightly higher than the single daily point measurement at the pier.

Thusly, subsurface structure (slope of the pycnocline) can be predicted from the historic manual measurements at the pier by first applying a linear transformation from “surface pier” to “surface wire walker”, then using the nonlinearly tuned multivariate S-map model of density, salinity, and wind speed (Figure 5.8). This allowed us to reconstruct a daily estimation of historical stratification across the initial period of historical analysis in (4). This is shown in Figure 5.9.

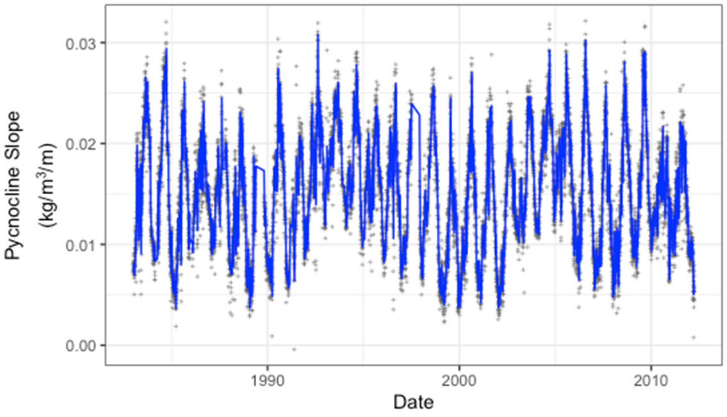


Figure 5.9— Imputed pycnocline slope for the coastal water of La Jolla reconstructed over the historical period of red tide analysis. Daily measurements are shown with crosses, and the weekly average is shown in blue.

In principle, in-so-far as the original time series variables are observation functions of the system, these imputed time series are also observation functions of the system and hence valid embedding coordinates (53). So has anything been gained? In practice, there can be two advantages: (1) an appropriate transformation of data can give improvements to forecast skill in the face of finite time series length, observation noise, and process noise (70), but also (2) the transformation can make the model more interpretable. This is really key for the newer applications of EDM to scrutinize changing, state-dependent interactions between variables.

If we go back and evaluate multivariate EDM forecasting using this imputed stratification, we see in fact that the pycnocline slope gives improved predictability relative to original surface measurement that went into the reconstruction. Combining the reconstructed pycnocline slope with random projection coordinates gives consistently better forecast skill than surface density ρ_{surf} and surface temperature SST. The random projection coordinates are constructed from the 1 and 2 week lags of physical measurements and 0, 1, and 2 week lags of the nutrient variables in (4). A reconstruction was also done for the Thorpe length scale L_{OT} that aims to capture water column stability/mixing, and not just physical stratification. While the ability to predict L_{OT} from surface quantities was lower than for pycnocline slope (Figure 5.8), the reconstructed L_{OT} still showed improved prediction over temperature (but not surface density).

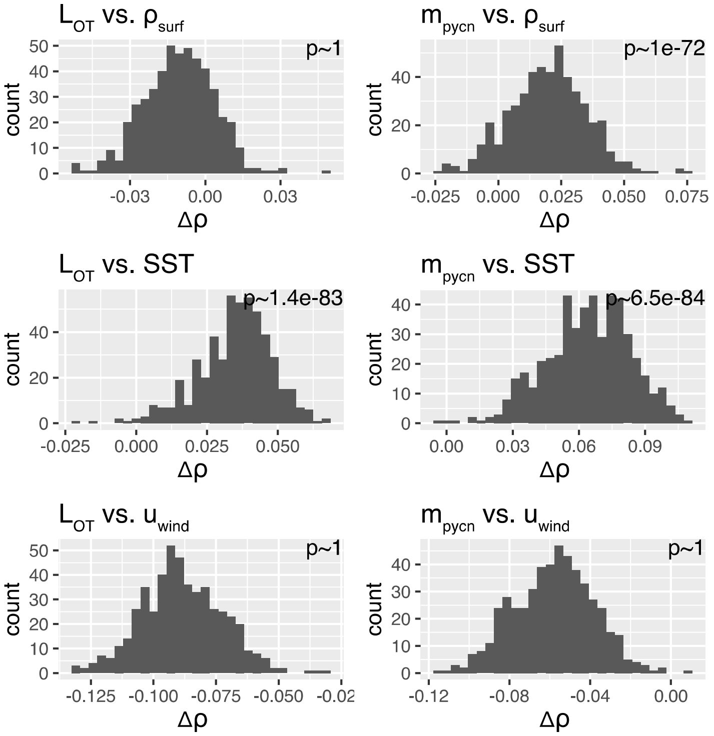


Figure 5.10— Forecast skill of random multivariate embeddings for predicting historical chl-a using EDM reconstructed stratification metrics, Thrope length scale L_{OT} and pycoline slope m_{pycn}, relative to original historical observations, surface density ρ_{surf}, sea surface temperature SST, and long-shore wind u_{wind}.

5.3.2.5 Reanalysis of future scenarios with imputed stability

While the reconstructed pycnocline slope shows the ability to incrementally improve the red tide forecast skill, the greater promise lies in making the models more interpretable. In this way, the imputed pycnocline slope also allows us to revisit the long-term scenario exploration with a variable we hypothesize has a more direct mechanistic role. The systematic effects of temperature and silicate are not qualitatively changed when we include this other variable in embeddings (Figure 5.11, green and silicate boxes). However, contrary to our expectation, simulating chlorophyll dynamics with a 25% multiplicative increase on stratification does not lead to a systematic change in bloom frequency across top embeddings (Figure 5.11, red boxes). While a decreasing in stratification (less sloped pycnocline) on average produced fewer blooms per year, when a steeper pycnocline is simulated, the EDM prediction is that bloom frequency will not change. This could indicate that either the hypothesis that increased stratification promotes blooms is wrong or the imputed water column stability is misleading us. One possibility is that dinoflagellates benefit from an intermediate level of stratification, and that a scenario of a gross increase in the baseline stratification does not actually increase bloom-favorable conditions.

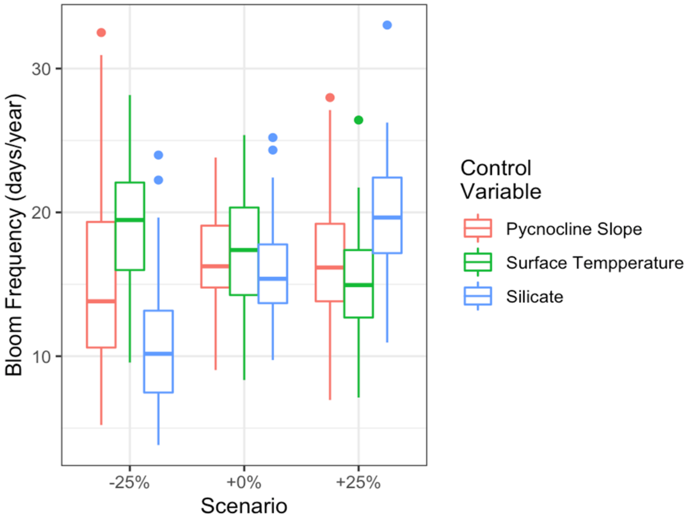


Figure 5.11— Revisited long-term scenario exploration of bloom frequency including imputed pycnocline slope. Differences in bloom frequency under hypothetical non-stationarity in pycnocline slope (red) is shown along with surface temperature (green) and silicate (blue) which both showed systematic patterns under the original analysis (Figure 5.6).

5.4 Case Study 2: Pacific Coral Reef Resilience

Lack of long time-series was the central challenge for building a predictive EDM framework for understanding the benthic cover dynamics at U.S. Pacific reefs. With traditional EDM, univariate lag-coordinate embeddings are used for two important analyses that provide the foundation for multivariate prediction and analysis (5). First, the presence of low dimensional dynamics is evaluated with simplex projection and S-map analysis. In addition to validating the foundational working assumption of EDM, the analysis also provides an estimation of embedding dimension, *E*, roughly the number of dimensions that need to be accounted. Second, convergent cross-mapping is used to establish which variables actually show evidence of causal coupling and hence can be used to account for these dimensions of variability. In this traditional approach to EDM analysis, the univariate time lags stand in for the yet-to-be-determined dimensions of dynamics.

In the case of Pacific reefs, we have a number of candidate variables for interaction, including different components of benthic cover and environmental variables, but also benthic cover at neighboring reef segments (see 4.9.2). We first implemented static cross-mapping 4.6 to test hypothesized environmental drivers of benthic dynamics at individual sample segments, then conducted a series of analysis across half of the study islands (randomly selected before all analyses) using spatial lags, multivariate, and we used random projection coordinates of these variables to stand in for undetermined dimensions of the dynamics much in the way univariate time lags are used in traditional EDM.

5.4.1 Environmental drivers of benthic dynamics

In the case of the benthic dynamics, we can reasonably hypothesize that the different % cover variables share some causal relationship. Thus, we can take the 8 core benthic cover classifications as different coordinates of a multivariate manifold, and test how well the manifold can cross-map values of hypothesized environmental drivers (4.6.1). We restrict analysis to the individual reef segments in the in-sample islands, across all study years (n = 10,280). To be conservative, we extend the standard leave-one-out cross validation to additionally exclude all measurements made from the same island in the same year from the nearest neighbor selection (4.1.3.1). Figure 5.12 shows that the maximum value of DWEF experience at a reef segment over the previous 2-years has the most direct causal association with the benthic cover. There is also evidence that directional variance of DWEF also has a causal effect, thus we entertain both DWEF$_{max}$ and DV-DWEF as embedding variables.

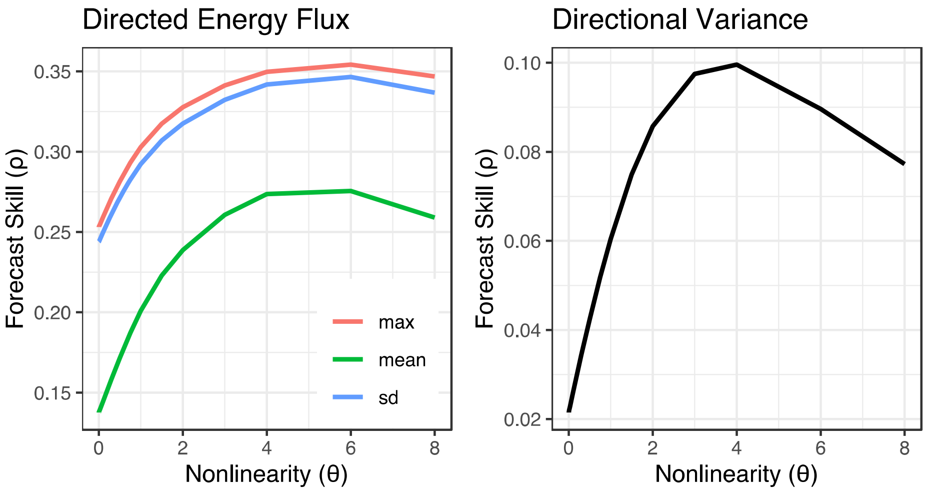


Figure 5.12— Static cross-mapping of wave forcing metrics to identify drivers of benthic cover. (Left) analysis across different statistical summaries of DWEF show that the maximum DWEF over a 2-year period has the most direct causal association. (Right) the directional variance in DWEF also shows some evidence of causal association. Thought predictability is substantially lower, the forecast skill is still highly significant (n = 10,280).

5.4.2 EMD for Forecasting % Live Coral

5.4.2.1 Spatial lags

To assess the predictive value of including spatial lags for predicting changes in % coral cover, we took a multivariate EDM approach incorporating random projection coordinates to account for our initial uncertainty in the most proximate causal variables (4.4.2). Initial analysis showed evidence that spatial lags contain predictive information for % live coral cover (Figure 5.13). However, the % live coral cover at immediate spatial neighbors is strongly correlated. Additionally, a pure "spatial lag" embedding does show significantly lower error (both mean absolute error and root-mean squared error) than the AR1 model, although the forecast skill measured with Pearson's ρ.

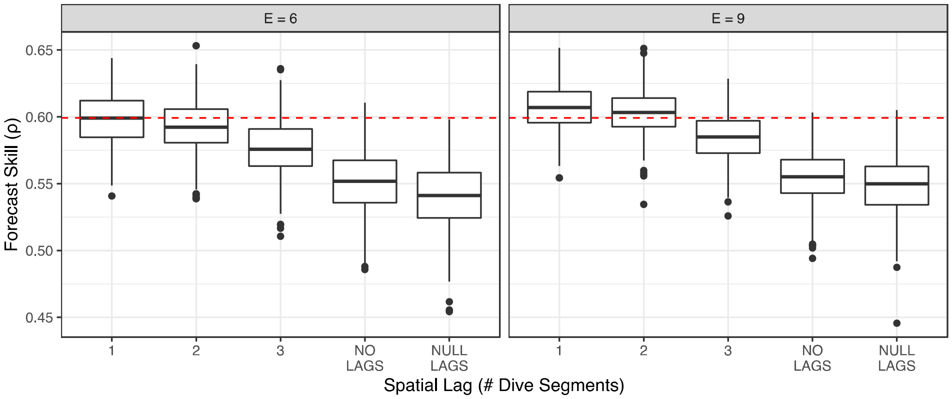


Figure 5.13— Forecast skill using spatial lags to predict 2-year change in % Live Coral Cover. Presence of spatial dynamics at 150+ meter scale is evaluated using a multivariate EDM approach with randomized lags to predict future % coral cover at sites across half of the RAMP islands. Two spatial lags of coral cover are used ("left" and "right") with tau, the spatial distance from these lags to the target point ranges from 1 - 5 dive segments (approximately 150 m on average). The embedding also includes the "un-lagged" % Coral cover (at the target location), then is filled out with random projections of multivariate data at the target location. Forecast skill of these is then compared to skill using no spatial lags ("NO LAGS") but 2 more random projections of the multivariate data, as well as null spatial lags ("NULL LAGS") created from random permutation of the data. Finally, the forecast skill of an AR(1) model (i.e. just using the inertia present in % coral cover) is shown as a red dotted line for comparison.

These indicate that spatial dynamics can potentially be incorporated as variables for forecast, but spatial dynamics alone cannot predict the 2-year dynamics. It is important to note that these results are also likely sensitive to the type and scale of the coral observations. Concurrent with this project, team member Stuart Sandin published an analysis with other colleagues (McNamara et al. 2019). The study offers additional insight into empirical dynamic modeling of nonlinear spatio-temporal dynamics of coral using similar (but not identical methods) on data at a much smaller scale. Their analysis was concerned with predicting species variation across space in 100 m^2 photomosaics taken from Palmyra Atoll, with dynamics grouped into 10 cm pixels. In empirical images taken from areas of reef recovering from disturbance, pixel by pixel forecasting of coral species showed a clear deterministic signature compared to randomized images (see Figure 1, reproduction of Figure 3 in McNamara et al. 2019). While insight into the true dynamics is indirect, as the analysis was restricted to single temporal snapshots, the study illustrates strong spatial processes operating at a scale about 1/1000th of the resolution of the visual towed-diver survey data we have studied.

Our preliminary understanding (or working hypothesis depending on perspective) is that direct spatial processes play out at smaller spatial scales (10 cm – 1m), while at larger spatial scales (100m – 1km), spatial lags may only contain indirect information about causal mechanisms acting at large scales (fish communities, environmental conditions, etc.). However, it is also possible that part of the difference between these studies is due to the scale of taxonomic aggregation. The visual towed-diver survey data only quantify total hard coral cover, while processing of photomosaic quadrats allowed McNamara et al. to resolve down to the species level. Since coral are broadcast spawners, it is possible that there are additional spatial dynamics at 100m – 1km scale at the species level we have missed due to the coarse resolution of visual survey data.

5.4.2.2 Multivariate model selection

Since there is no physical mechanism to systematically distinguish the neighbors that are to the left and right (or rather clockwise and counter-clockwise) of the target, we include these spatial lags as the maximum value of the two neighbors, $\max(\%LC(x_{L1}),\%LC(x_{R1}))$ and the minimum value of the two neighbors, $\min(\%LC(x_{L1}),\%LC(x_{R1}))$.

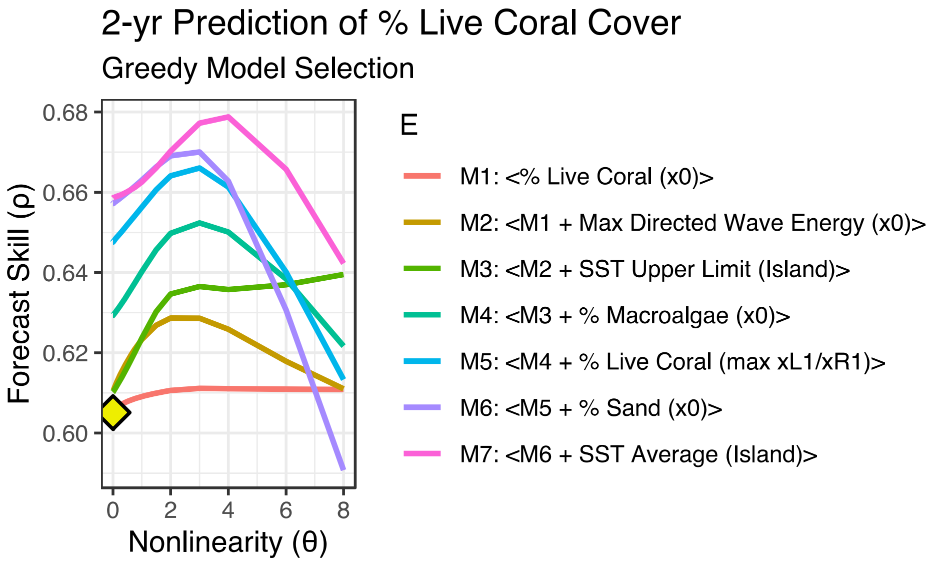


Figure 5.14— Forecast skill of multivariate embeddings for 2-year forecasts of % Live Coral Cover (%LC) selected sequentially using random projection coordinates with greedy model selection. Forecast skill is evaluated here without random coordinates, and as expected the full 7-dimensional model performs best. This model incorporates %LC, $\mathrm{DWEF}_{\mathrm{max}}$, $\mathrm{SST}_{\mathrm{upper}}$, %MA, $\mathrm{STT}_{\mathrm{mean}}$, maximum spatial neighbor %LC, and %S. The forecast is optimal for nonlinear tuning ($\theta > 0$), and shows substantial improvement on the naïve predictability due just to the serial temporal correlation in %LC (yellow diamond).

With the ability to more explicitly resolve the most proximately related variables for % live coral dynamics, we revisited the spatial lag analysis shown in Figure 5.13. The greedy model selection identified spatial lags for prediction at the 5th step, so we fixed 4 of the coordinates to % live coral, maximum directed wave energy, upper limited of island SST, and % macroalgae.

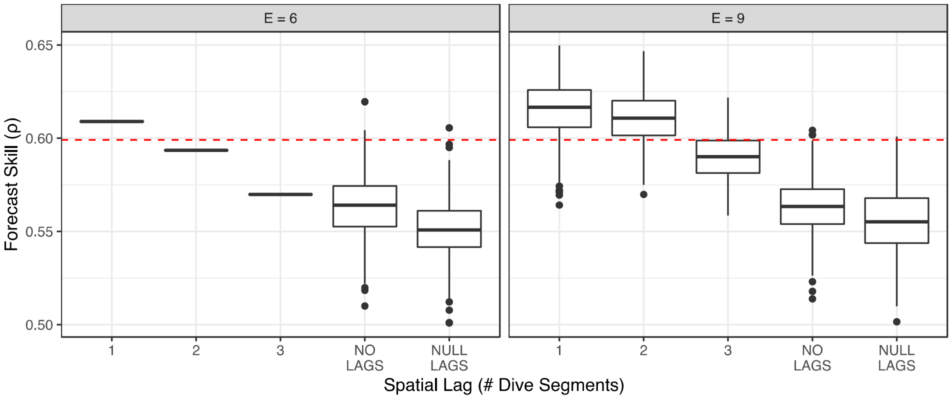


Figure 5.15— Reanalysis of forecast skill using spatial lags to predict 2-year change in % Live Coral Cover. Spatial analysis shown in Figure 5.13 was repeated, but replacing some of the random projection coordinates with explicit variables identified in greedy model selection. Forecast skill of these is then compared to skill using no spatial lags ("NO LAGS") but 2 more random projections of the multivariate data, as well as null spatial lags ("NULL LAGS") created from random permutation of the data. Finally, the forecast skill of an AR(1) model (i.e. just using the inertia present in % coral cover) is shown as a red dotted line for comparison.

5.4.3 EDM for Forecasting Other Benthic Variables

We can also investigate multivariate EDM predictions of other benthic variables. The stressed coral (%StC) category of the RAMP TDS is of particular interest, as it captures disease outbreaks and bleaching events that are of major conservation concern. Figure 5.16 shows greedy mEDM model selection applied to (A) %StC, (B) %S, and (C) %MA. Although the EDM forecast skill for predicting %StC is lower than for predicting live coral (%LC), the mEDM forecast skill is still highly significant, especially considering that the %StC shows no meaningful autocorrelation over a 2-year span and hence has less inherent statistical predictability. In all cases, information about the benthic state of spatial neighbors appears to offer some predictive skill.

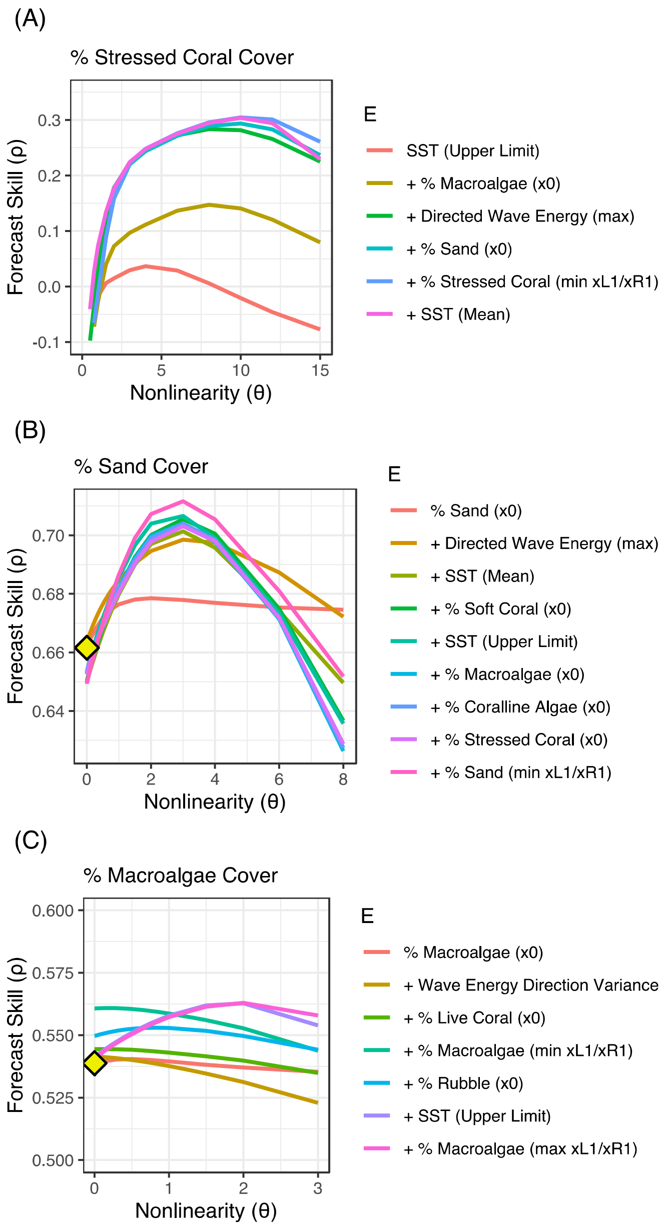


Figure 5.16— Forecast skill of multivariate embeddings for 2-year forecasts of other major % cover variables using greedy model selection and random projection coordinates. (A) Stressed Coral shows substantially lower predictability, but has no intrinsic inertia. Thus, observed forecast skill under nonlinearly tuned S-map models is noteworthy. (B) Sand Cover shows similar inertia (autocorrelation) to %LC, but nonlinearly tuned models incorporating environment and benthic cover show additional forecast skill. (C) Macroalgae predictability is dominated by the autocorrelation in %MA between 2-year observations.

5.4.4 Characterizing conservation landscape

The multivariate modeling results above can be translated into a number of metrics to describe conservation state and potential of reef segments. For example, the direct output of the forecasting model in Figure 5.16A and attached uncertainty (4.1.3.2) characterize relative risk of disease and bleaching. Further insight is possible by digging into the local linear coefficients of these models, however. For calculating stability metrics (see 4.8) it is necessary to have a single multivariate embedding that can predict all biological variables of interest. Furthermore, the issue of spatial interaction complicates the Jacobian interpretation. Based on the greedy model selection process, we observe that all %LC, %S, and %MA can all be predicted near the maximum observed predictability by the five-dimensional embedding [%LC, %S, %MA, $DWEF_{max}$, SST_{upper}]. Thus, we proceed with a detailed analysis of the Jacobian matrices estimated for this model across time and space, noting that it may overlook factors associated with the spatial processes uncovered above (5.4.2.1).

5.4.4.1 Stability

Analyzing the Jacobian matrices of the three closely associated benthic variables, %LC, %S, and %MA, the top panel of Figure 5.17 shows largely stable dynamics. Roughly 80% of plots are estimated to have LEV < 1, including most segments with currently high live coral cover (%LC). The other 20% show unstable dynamics. This picture changes, however, if bleaching and disease are considered by including the % Stressed Coral in analysis, shown in the bottom panel of Figure 5.17. The episodic and dramatic nature of disease and blenching events drives considerable instability in segments with high %LC. Note that the key environmental drivers $DWEF_{max}$ and SST_{upper} are included in the model but do not directly contribute to the calculation of the largest eigenvalue. The environmental sensitivity is considered a separate effect.

The LEV was also calculated for these models using the regularized S-map as studied in (65, 71). Using either LASSO or Ridge Regression penalties on coefficient magnitudes, the LEV are highly correlated (Spearman ρ>0.90 in both cases), although the standard (non-regularized) S-map estimates of LEV are systematically larger than Ridge Regression. This is unsurprising, given that individual coefficients should be smaller, and thus the standard S-map estimates of LEV can be considered the more sensitive, the Ridge regression estimates the more conservative.

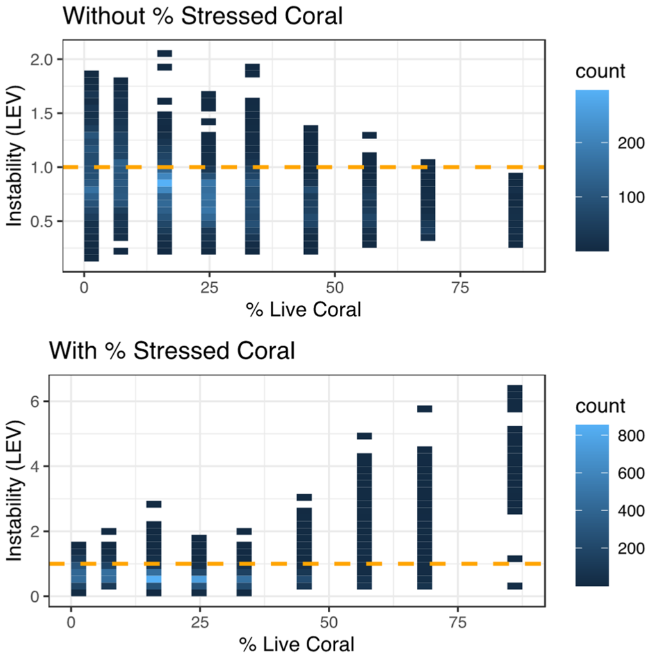


Figure 5.17—Dynamic stability of reef segments as a function of current % Live Coral. (Top) The largest eigenvalue of the Jacobian matrix is estimated via S-map for the 5-dimensional multivariate EDM model containing [%LC(t), %S(t),%MA(t), $DWEF_{max}$(t), SST_{upper}(t)]. (Bottom) Results are shown for equivalent

calculations on the EDM model [%LC(t), %S(t), %MA(t), %StC(t), $DWEF_{max}$(t), SST_{upper}(t)] that has % Stressed Coral as an additional dimension.

5.4.4.2 Environmental sensitivity

Since the multivariate S-map models contain $DWEF_{max}$ and SST_{upper}, we can also investigate the patterns in environmental sensitivity across reef segments. Broad patterns are shown in Figure 5.18. Reefs generally show negative relationships with both SST_{upper} and $DWEF_{max}$, and this is particularly pronounced for reefs with high coral cover (at least %40 Live Coral). However, this sensitivity shows considerable variance, and near-zero estimates of these sensitivities can be understood as an indicator of resilience to environmental change.

There are also reef segments that show positive dependence with SST_{upper} and $DWEF_{max}$. Figure 5.19 shows the sensitivity of %LC to SST_{upper} as a function of SST_{upper}. Unsurprisingly, there is a general pattern consistent with a dome-shaped temperature sensitivity. For islands with lower temperatures, there is a positive response of %LC to the SST upper limit, and at islands with higher temperatures, the response becomes strongly negative.

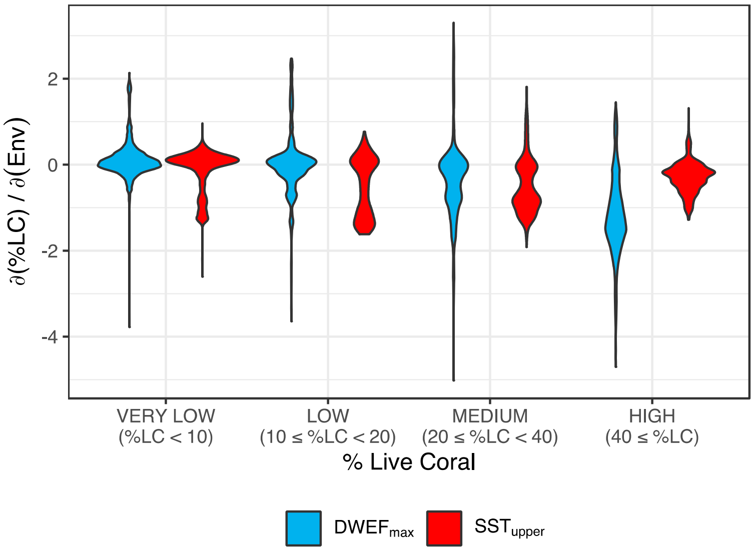


Figure 5.18— Environmental sensitivity of % Live Coral across all in-sample islands. High coral cover sites generally show a strong negative effect of maximum wave energy ($DWEF_{max}$), as well as a negative effect of SST upper limit (SST_{upper}).

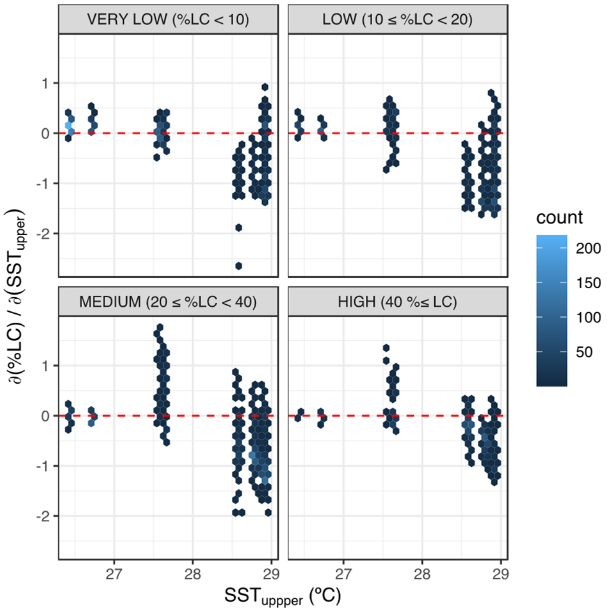


Figure 5.19— Effect of SST_{upper} on %LC across the range of SST_{upper}.

5.4.4.3 Growth potential

Dynamic instability is not necessarily a counter indicator for conservation. Figure 5.20 shows estimated stability in relation to the predicted change in coral cover. Across islands, there are many high coral cover sites (%LC ≥ 40) that show stable declines. That is, the LEV indicates stability, log(LEV)<0, and the EDM forecast predicts that %LC + %StC will decline over the next two years. However, there are also sites that show decline, but with unstable dynamics. This characterizes sites where the right intervention might "turn the tide." Sites with high coral cover, but

Sites with low coral cover that show unstable growth, log(LEV) > 0 and expected %change in coral cover > 0, are areas where growth could take hold. Sites with these characteristics are especially interesting to consider in light of recent advances in coral out-planting for enhancing recovery. In fact, the S-map model coefficients contain an even more proximate indicator of out-planting potential, the self-regulation term of the Jacobian for %LC. If this coefficient, ∂%LC(t+2)/∂%LC(t), is larger than 1, it indicates a small coral addition would grow in time. Figure 5.21 shows the estimated growth of a coral addition in relation to the LEV. Few sites predicted by this model predict growth, but these are indeed areas with unstable dynamics.

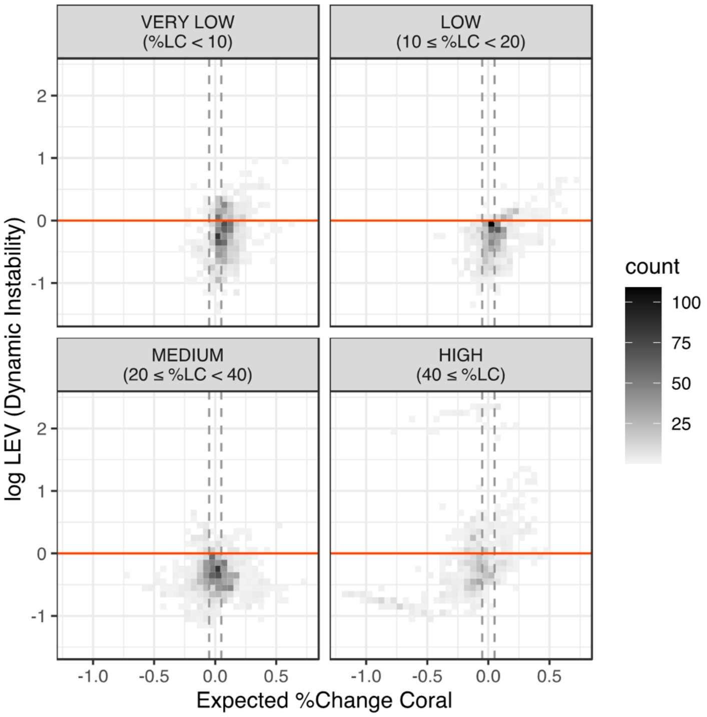


Figure 5.20— Dynamic instability related to expected change coral cover. Here the S-map estimation of LEV is shown against the predicted change in total coral cover (%LC + %StC). Sites are sorted by the level of live coral cover.

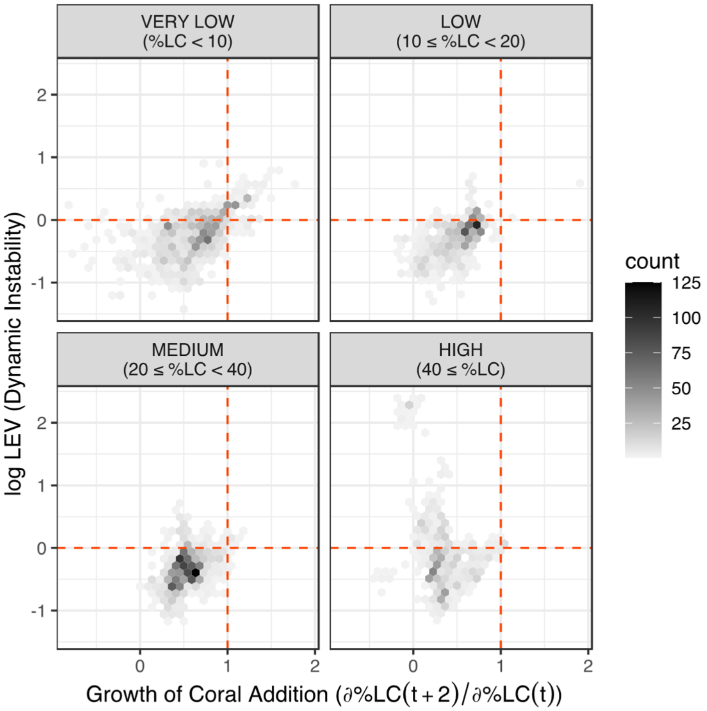


Figure 5.21— Dynamic instability related to the inferred growth of a coral addition. The inferred growth of a coral addition is predicted by the self-regulation term of the S-map Jacobian, , ∂%LC(t+2)/∂%LC(t).

5.4.4.4 Spatially resolved metrics

Since each prediction of the multivariate EDM model is resolved to a particular reef location, the above calculations can all be projected onto maps of individual islands. Figure 5.22 shows Johnston Atoll as an example. The long-term prevailing direction of wave forcing is from the Northeast. These areas of reef have mostly low coral cover, but show stability and are not sensitive to changes in wave energy. Stability is lower and environmental sensitivity appears much higher on the opposite side of the land mass to the West.

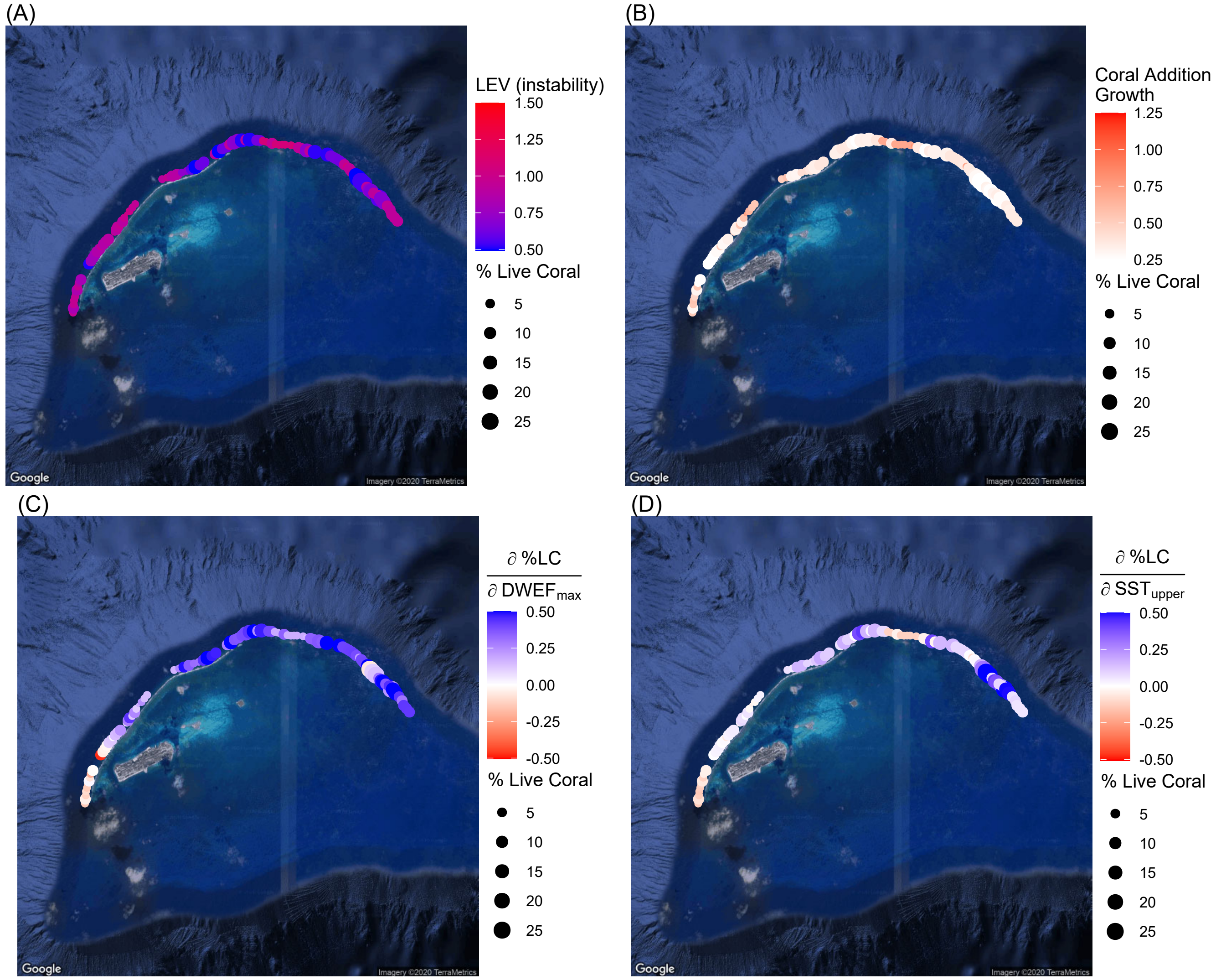


Figure 5.22— EDM metrics for identifying reef status at Johnston Atoll evaluated on the 2010 observations. Panels show (A) the dynamic instability measured by the largest eigenvalue of the S-map Jacobian, (B) the estimate fractional growth rate of a small coral addition, (C) the sensitivity of %LC to DWEF$_{max}$, and (D) the sensitivity of %LC to SST$_{upper}$. Note that substantial shallow coral cover exists in the interior lagoon of Johnston Atoll that was not reached by the RAMP TDS which followed the 15m depth contour.

5.4.5 Out-of-sample prediction

The multivariate EDM model for predicting the 2-year dynamics of % Live Coral Cover was applied to the 16 Pacific Islands held out-of-sample for all above analysis. These are summarized in Table 5.2. Although the model is not able to predict dynamics at some islands beyond the autocorrelation in %LC, 11 of the islands show out-of-sample forecast skill that exceeds the autocorrelation. Notably, the two strongest predictions are seen at Lisianski and Pearl & Hermes, which are both in the Hawaiian archipelago and have islands nearby that were analyzed in the in-sample portion (Kure, Midway). However, the general pattern of which islands were well predicted and which were not do not seem readily explained by human population, reef area, or island group.

Island	**# pred**	ρ	**mae**
Agrihan	54	0.606 (0.676)	7.87 (7.55)
Alamagan	46	**0.390** (0.203)	**8.84** (9.66)
Asuncion	56	**0.565** (0.469)	**7.51** (8.48)
Farallon de Pajaros	62	**0.328** (0.321)	**5.33** (8.73)
Guam	145	**0.529** (0.525)	**8.8** (11.2)
Guguan	28	**0.534** (0.496)	**7.88** (9.57)
Laysan	20	**0.478** (0.418)	**5.61** (9.21)
Lisianski	71	**0.748** (0.625)	**9.63** (11.6)
Maro	54	0.671 (0.701)	13.2 (12.7)
Maug	142	**0.627** (0.608)	11 (11)
Pearl & Hermes	64	**0.668** (0.593)	**5.98** (7.85)
Rose	412	0.365 (0.403)	9.03 (8.47)
Sarigan	58	0.388 (0.416)	8.61 (8.61)
Swains	293	**0.144** (0.137)	16.7 (15.2)
Tinian	52	0.574 (0.767)	**6.97** (7.65)
Tutuila	285	**0.383** (0.308)	8.68 (7.65)

Table 5.2— Out-of-sample forecast skill for 2-year ahead predictions of %LC across 16 islands. Forecast skill by Pearson's ρ and mean absolute error (mae) are listed, with grey numbers in parenthesis indicating the corresponding metric for the AR1 model of %LC.

5.5 Key Empirical Demonstrations

In the course of the grant, two empirical systems arose for demonstrating EDM advancements made under the grant. These systems served as more meaningful demonstrations than toy models. Additionally, they demonstrate the broad applicability of EDM beyond the marine realm.

5.5.1 Cedar Creek

We applied S-map analysis to long-term observations at Cedar Creek to demonstrate quantitative treatment of previous described regime shift (47). We focused on a 5-dimensional multivariate EDM model (embedding) based on strong interactions found in the Cedar Creek data with convergent cross-mapping: [Nitrate(t), Richness(t), C3(t), C4(t), W(t)], where C3 is above-ground biomass of C3 grasses, C4 is above-ground biomass of C4 grasses, and W is above-ground biomass of woody shrubs. The largest eigenvalue calculated for this multivariate S-map model identifies rising instability of plots with nitrogen additions (Figure 5.23), and the LEV reaching 1.0 can identify impeding switch from native C4 to invasive C3 community composition in many cases.

While instability is strongly associated with the enrichment regime, plots do not behave identically. At moderate levels of nitrogen addition, some plots destabilized and some do not (see for example the panel for +5.44 kg N ha^{-1} yr^{-1} in Figure 5.23). Looking at the S-map estimate of LEV as a function of Plot Richness (Figure 5.24) we can see that within nitrogen treatments, higher plot richness is one associated with greater stability (lower LEV), recapitulating the common-held notion in ecology that diversity begets stability.

Finally, previous study done after the initial 8-year experiment studied here identified the total nitrogen addition as a driver of hysteresis behavior. Analysis with S-map confirms this notion (Figure 5.25), but without the need for cessation experiments, and based solely on the first 8-years of experiments.

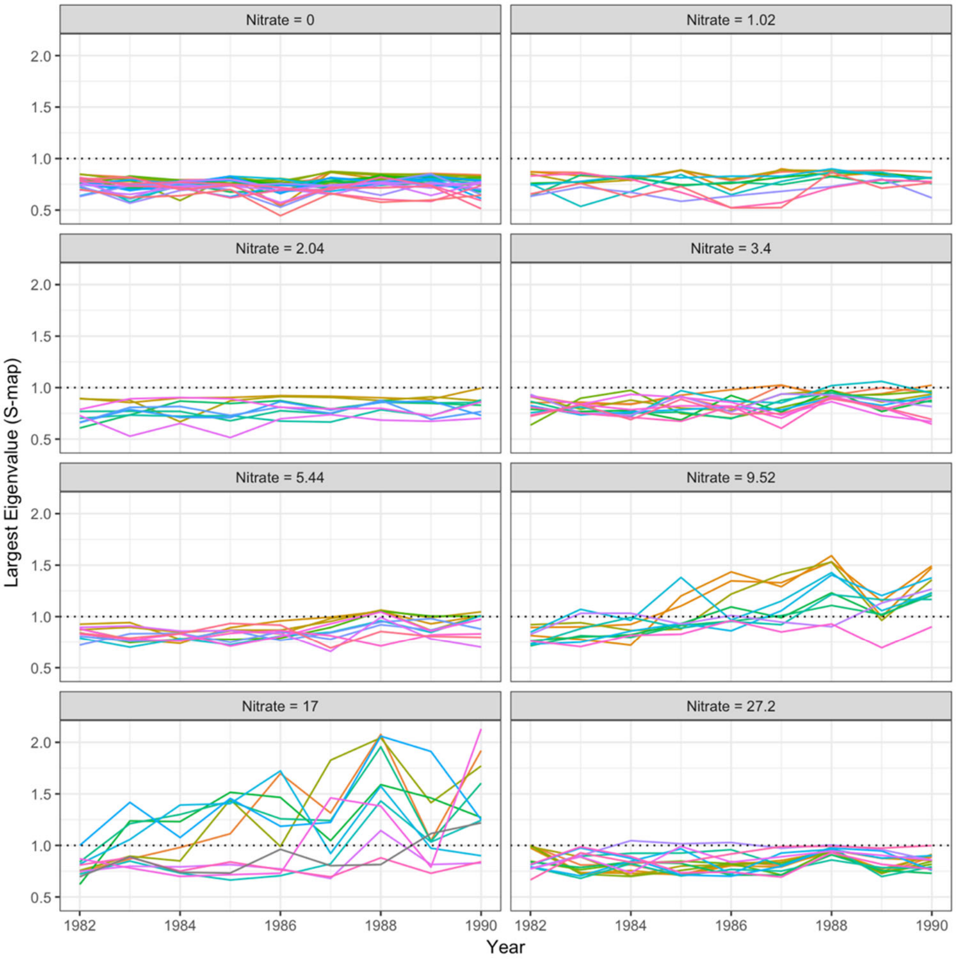


Figure 5.23— EDM estimated instability of Cedar Creek patches under various nitrogen treatments (0.0 - 27.2 kg N ha^{-1} yr^{-1}). LEV > 1 indicates unstable dynamics, while LEV < 1 indicates stable dynamics.

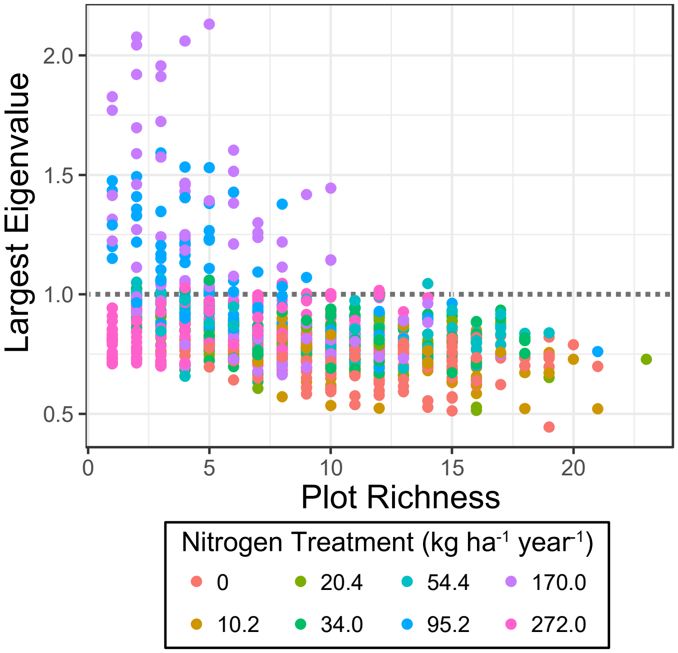


Figure 5.24—Instability of vegetation dynamics (largest eigenvalue of the S-map estimated Jacobian) shown across all plots as a function of plot richness. High richness is associated with more stable dynamics (LEV < 1), recapitulating the common held notion in ecology that diversity buffers against instability.

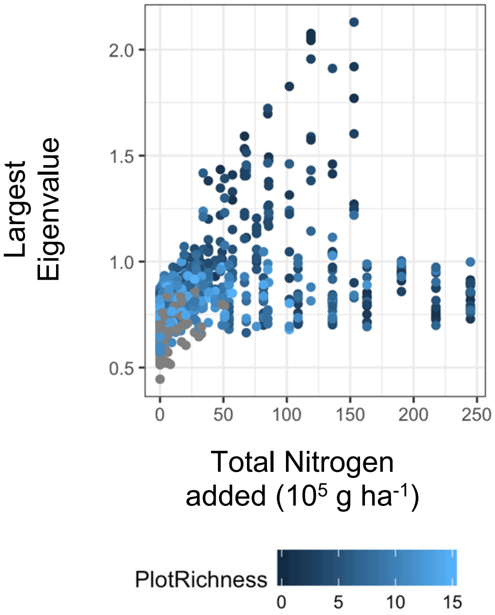


Figure 5.25—Plot instability in relation to total nitrogen added. Dynamics of the vegetation assemblage are stable in the absence of nitrogen additions. However, total additions of nitrogen of as little as 25 x 10^5 g ha^{-1} can lead to unstable dynamics in plots with low richness (darker blue circles). At extreme levels of total additions, dynamics become stable again but these plots are dominated by a low richness community of invasive C3 grasses.

5.5.2 Lake Geneva

Water quality in Lake Geneva emerged as an opportunity to apply the same methodology developed for the red tide case study to a conceptually similar problem in a classic system with significant potential to produce a high visibility study for EDM. It was presented to us by colleagues from EAWAG (Zurich) as a problem with large amounts of excellent monitoring data, a long history of modelling attempts and a mystery as to why it has not returned to it's former state. Despite aggressive measures to reduce phosphorous loading to pre-20th century levels, the overall water quality of Lake Geneva, measured by dissoved oxygen at depth (DO_B) and chlorophyll (CHL), have not returned to their expected pre-20th century levels. To better understand this mystery and provide actionable management advice going forward (in the context of 21st century climate change), a hybrid approach was developed. This involved using classical equation-based physics models with unrealistic fixed constants for the changing ecology and complex chemistry, in combnination with EDM models to accommodate this complexity. Aside from resolving the mystery, the key concern was to make credible predictions of water quality health in Lake Geneva under various plausible climate and management future scenarios.

Like the red tide event, there are extreme events that dominate the relevant ecological history, but in the lake, these extreme events are changes in oxygen rather than changes in phytoplankton. Previous attempts at parametric modeling of hypoxia in Lake Geneva have had some success (72) with 1-dimensional physical circulation models being able to faithfully capture the physical mechanisms of oxygenation reasonably well. However, long-term changes (non-stationarity) in biological relationships that affect oxygen consumption – like the carbon:phosphorous ratio and carbon export fluxes – confound the validity of extrapolating the simple parametric models to new scenarios where, for example, in the real world temperature changes and nutrient dynamics will interact.

These changes are directly quantifiable by EDM, as shown in Figure 5.26. Consequently, EDM can explain (predict) historical variability better than a strict parametric approach, especially as the lake has entered new regimes of biogeochemistry. Because parametric models can capture the (relatively) simple physical mixing that is a key driver, a hybrid modeling approach emerged as the best predictive framework, where large remixing events are predicted by the parametric physics model, and the effects of complex biogeochemistry in the inter-mixing intervals (that can span multiple years) are accounted for with EDM. This enables long-term simulations to reproduce behavior with a great fidelity.

The paper demonstrates the hybrid approach not only leads to substantially better prediction (Figure 5.26), but also to a more actionable description of the emergent rates and processes (biogeochemical, ecological etc.) that drive water quality. Notably, the hybrid model suggests that the impact of moderate air temperature increase ($\Delta T_{air} = 3°C$; a lower bound in recent CH2018 scenario (RCP8.5) for the western part of Switzerland) on water quality would be on the same order as the eutrophication of the previous century, and more significantly that the best management action may no longer involve a single control lever such as reducing phosphorus inputs alone.

We believe that this hybrid approach represents a template that that sets the bar for the next generation of environmental management tools – tools that pass the validation test of out-of-sample forecasting, that accommodate nonlinearity and nonstationarity, and that have the flexibility to accommodate non-analogue futures. Full details of the Lake Geneva example are contained in the manuscript (now in press at PNAS) attached in the Appendix.

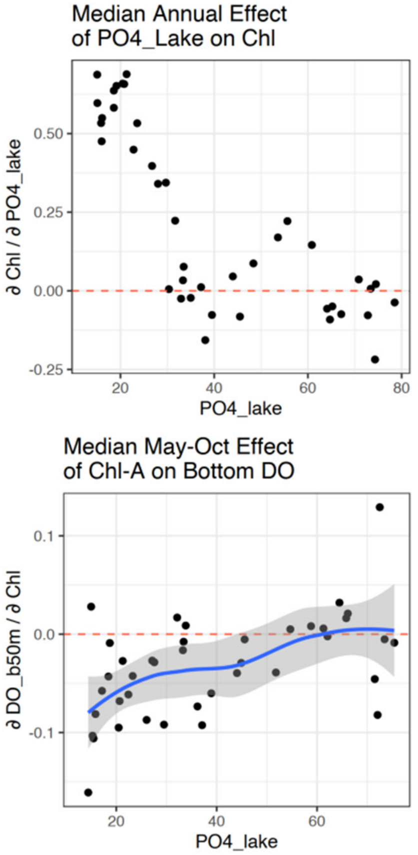


Figure 5.26— S-map estimations of first order partial derivative quantify the state-dependent effect of (A) $PO4_T$ on Chl and (B) Chl on DO_{deep} under changing levels of $PO4_T$. Positive (negative) values indicate that an increase in $PO4_T$ lead to an increase (decrease) in (A) Chl or (B) DO_{deep}. At high levels of $PO4_T$, there is little evidence of phosphorous limitation as the effect of $PO4_T$ is near 0 until $PO4_T$ drops below 40 µg/L. However, a drop in Chl does not necessarily translate into improved water quality, since the effect of Chl on DO_{deep} also strongly depends on state. Importantly, the relationships quantified here can provide a way to dynamically parameterize constituents in equation-based models.

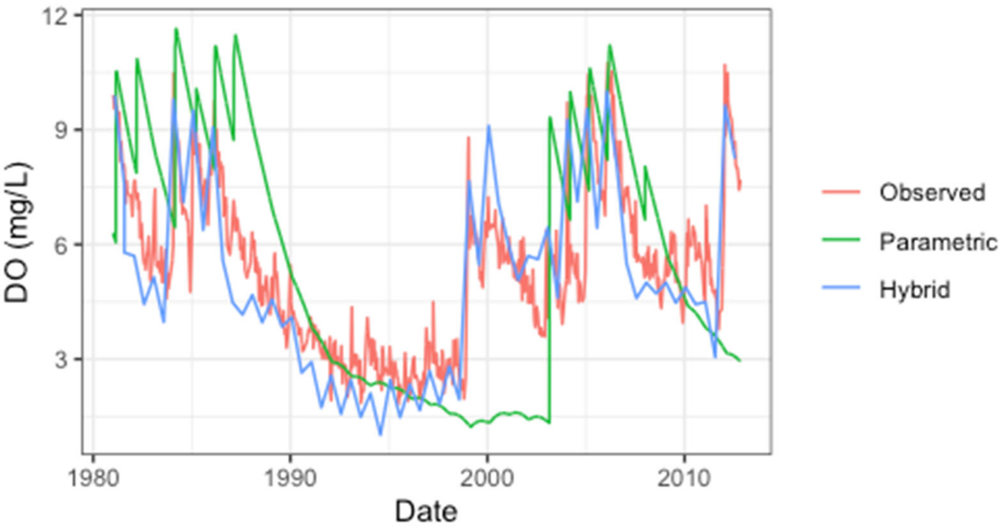


Figure 5.27—Comparison of long-term predictions of dissolved oxygen (DO) in Lake Geneva using the previously reported parametric model of Schwaffel et al. (72) and the newly adapted parametric/EDM hybrid model. Both models are initialized at the start of the time course, then run iteratively with climate and biogeochemical inputs.

6 Conclusions and Implications for Future Research

The project case studies, in synergy with projects of opportunity, have provided templates for nonlinear data analysis that is able to skillfully model nonlinear systems even when faced with significant data challenges. Moreover, the EDM approach requires demonstrated out-of-sample forecast skill (validation is built in) and we have shown that it has flexibility to credibly address nonequilibrium, non-analogue futures, that can recommend and support management actions.

Critically, the project successfully developed a standardized set of applied computational tools for empirical dynamic modeling. Code packages are now publicly available in R, Python, and C++, and as noted in the summary are quickly finding a large user base. The **rEDM** package has been downloaded more than 43,000 **times** from the central CRAN server since its launch in 2016; and more strikingly, **pyEDM**, has been downloaded more than **330,000** since launch in Fall 2021. Simultaneously, the grant allowed us to develop documentation to communicate the essential foundations for new users wishing to apply EDM to their research. Together the code and documentation are already having a transformative effect on multiple scientific domains. In the first months of 2022, there are over 130 new publications using convergent cross mapping across ecology and diverse other fields. In addition, to impact a broader audience, a Stata package of EDM tools based on rEDM is now available for social scientists and financial engineers (https://researchprofiles.canberra.edu.au/en/publications/edm-stata-module-to-implement-empirical-dynamic-modeling) along with a user tutorial.

It is our hope that EDM will be in the toolbox of other future SERDP projects, not just the two case studies of marine environments focused on here. Conversations with SERDP researchers along with past SERDP review board members and former SERDP Program Manager, John A. Hall have identified critical needs that EDM can fill: ranging from forecasting/understanding red tides at Camp Lejeune (analogous to the SoCal study herein), to hydrology to fire ecology. General thoughts about how this might take shape follow below.

6.1 EDM & SERDP

6.1.1 Reduced Sets of Variables

A key, broad contribution of this project to environmental management is demonstrating empirical dynamic modeling (EDM) for identifying reduced sets of variables. Convergent cross-mapping is a general criterion for measuring causal interactions in nonlinear systems. Thus, it allows us to explicitly identify the most important driver variables from a larger set. For example, CCM readily identifies that rainfall, despite being linked to red tides in other areas of the world (e.g. Florida) is not an important variable in the blooms off Southern California. Moreover, successful demonstration of static cross mapping on modeled (Figure 5.1) and empirical coral reef data (Figure 5.12) highlight how long time series aren't strictly necessary for the general framework to be successful.

6.1.2 Fire ecology applications

Based on conversations at an annual project review meeting with SERDP review board member, Kevin Hiers, we conducted preliminary EDM analysis of fuel moisture data (provided by Hiers) at Tall Timbers Research Station (Tallahassee, FL). Univariate analysis with simplex projection and S-map finds evidence of low dimensional, nonlinear dynamics in both midnight and noon fuel moisture data ().

This suggests a promising avenue for future work directed at clarifying meteorological drivers and forecasting long-term futures affecting wild fire frequency.

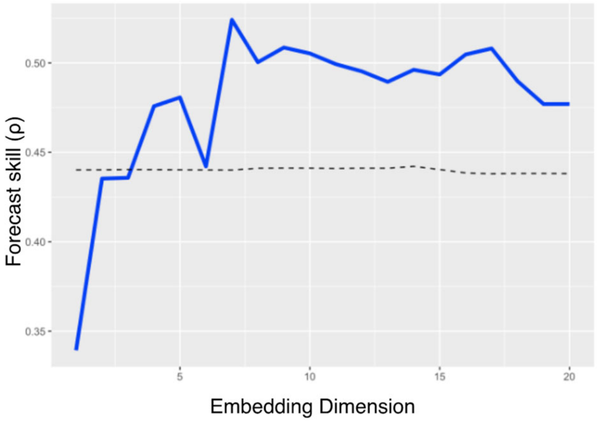


Figure 6.1— Simplex projection forecasts of midnight fuel moisture data from Tall Timbers Research Station. Univariate lag-coordinate embeddings with 6-lags show predictable dynamics (Pearson rho = 0.54) beyond the inherent predictability of the time series due to autocorrelation (dotted black line).

6.1.3 EDM for Rapid Assessment of Non-stationary Environmental Change

The successful adaptation of EDM methods to the Pacific Coral Reef case study demonstrates a major advance for EDM as a practical tool for management. Traditional EDM relies on long time-series measurements (long monitoring studies) to build an understanding of system complexity and identify relevant interacting variables, thus making it poorly suited to addressing emerging management questions in ecosystems or environments that don't have long-term observations associated with them. The road map set out in case study 2 shows how emerging management questions in previously un-studied systems could be addressed through EDM, so long as change (dynamics) can be measured over a limited time (months or a few years) in many equivalent systems, such as patches across a large area in a spatially explicit system like tropical coral reefs or forest fire. Note, however, that replicates do not need to have a spatial relationship, as the same general approach would work for e.g. water quality in US Army Core multi-use reservoirs across the United States (20).

6.2 Red Tides in Southern California

Based on combining the qualitative predictions of the ROMs model and the environmental sensitivity analyses of chlorophyll bloom frequency we performed with EDM, it is likely that red tide frequency will occur in southern California under a "business as usual" climate future (RCP 8.5). The qualitative ROMs predictions includes a prediction that the water offshore of La Jolla will on average be cooler over the next 3 decades than it was over the past 3 decades. This is perhaps counterintuitive on first examination, but is completely plausible given the complex oceanography of the region; recirculation of water coming from the North in the California Current, occasionally intrusion of water up from Baja Mexico, upwelling, and more. (Again, the ROMs model combined with our EDM mechanistic predictive model, predicts an increase in red tides over the next 3 decades… but uncertainty is not with EDM but with the ROMs general circulation model). However, our analysis has been structured so that these conclusions can be reevaluated if and when new climate downscaling products (sub ROMs) become available for the Southern California Bight.

6.2.1 Feasibility of Real Time Forecasting

The time horizon for skillful prediction in the historical data in principle allows for good predictability of red tide events with a 1 week or greater lead time. However, the current data collection pipelines pose practical limitations. The chief obstacle is that nutrient measurements are not analyzed immediately in real time, but involve processing time lags often greater than a month (though this could be nearly instantaneous with automated assays). Nevertheless, including nutrient time series was critical to achieving good prediction skill (4). Automated nutrient measurement technologies have improved greatly and have already been deployed elsewhere in the Southern California Bight (73). Another practical way forward could be building on the success of using EDM to impute water column stability. If it is possible to find proxys for nutrients – other available time series that are not mechanistic drivers but share causal associations with nutrient dynamics (such as pH or dissolved oxygen) – then the need for real-time nutrient measurements could be circumvented.

6.2.2 Predicting futures

Predictions about the future of dinoflagellate algal blooms in southern California are ultimately tentative due to a combination of ambiguity about the future climate and limits of forecast skill with the EDM model predictions. EDM framework has been designed to accommodate “multiple plausible futures”, but the practical insight is still fundamentally limited by how much we can constrain the envelope of “plausibility”.

6.2.2.1 Matching biological futures to climatic futures

At the heart of this problem is a mis-match between oceanographic measurement and modeling. Our best, long-term measurements are made from the part of the ocean easiest to observe—the coast, and yet it is also the part of the ocean that is hardest to model physically. The SIO Pier in particular is a fascinating place of study, as open ocean physics get beamed up onto the surf zone through a deep submarine canyon while getting convolved with a highly unique internal tide pattern. While much has gone into studying these physics, they are not integrated into the modeling structures like ROMS. Yet currently these are the best models for tracking down the outcomes of atmospheric warming on open ocean physics, albeit at a scale not perfectly matched to the biology of red tides..

Empirical data imputation is a possible solution to this problem, although our first attempt closed the gap of disagreement between pier variables and simulated ROMS behavior. Imputed stability is by no means the only variable we can hope to draw out of the shared causal analysis between manual pier measurements and automated profiler measurements. As shown in previous sections (Figure 4.4), the wire-walker can resolve the intense internal tide oscillations that occur at the pier and continually reshape chlorophyll distribution with depth. Another possibility would be to seek relationships between shore station measurements and measurements further off-shore. Are the physical dynamics produced by the ROMS 7 km off-shore more realistic? The moored profiler was deployed in close proximity to the SIO Pier, but similar analysis could be repeated for buoys, ARGO floats, and automated gliders.

At the end of the day, we question whether ROMS simulations should be considered credible despite being the “best available science” for long-term futures of near-shore ocean dynamics under climate scenarios. Despite the complexity and sophistication of ROMS models, we find no compelling evidence that the major simulated changes are more reliable than those reasoned by experts of local and regional oceanography based on simple arguments. Thus, while we feel confident in how the EDM model works and what it tells us, we are less confident about the scenario played out by ROMs that ultimately drives the EDM model. This is why we suggest exploring this issue and other possible solutions such as data imputation.

6.3 Coral Reefs Conservation

The most extensive global mass bleaching lasted 3 years – from 2014 to 2017 – and affected at least 75% of reefs around the globe. Nearly 30% estimated to have experienced mortality-level stress and 50% suffered at least two individual events (74). The list of anthropogenic threats to coral reefs includes much more than just thermal stress, however, and seems to be growing: local pollution, fishing pressure, acidification, direct physical destruction by boats, increasing cyclone intensity, and sea level rise (75). Runaway climate change could well eventually doom them entirely, but until this is a foregone conclusion conservation must focus on identifying and protecting pockets of reef that have the best chance at sustaining diversity of corals and associated organisms over the coming decades.

6.3.1 Decision making

We have demonstrated EDM's potential to identify differential impacts of these stressors, including high wave energy events associated with storms and maximum sea surface temperature. This framework can be refined and updated for more directed decision support. Metrics like those shown for Johnston Atoll in Figure 5.22 can help distinguish areas that are a priority for protection, from those that are lost causes. Reef areas with stable, high coral cover, and low sensitivity to temperature are the most critical to protect from direct disturbance (e.g. caused by marine vessel operations), as well as areas with lower coral cover that show positive prospects for recovery.

Although NOAA no longer collects these data, the historical library we have built from the TDS remains relevant. For any given site in the future, **only a single new snapshot is required to seed model forecasts and update benchmarks.**

6.3.2 Active Restoration

Active coral reef restoration through coral out-planting has seen received considerable attention and investment. In 2019 NOAA announced *Mission: Iconic Reefs* to restore 7 iconic reef sites within Florida Keys National Marine Sanctuary through an ambitious program of out-planting and coordinated research activities. The >90% decline of the once spectacular reefs in the Florida Keys has been tied to pretty much all known global factors: direct physical destruction from anchors, boat hulls, and storms; stress from changing ocean temperature and pH; and ecosystem stressors from eutrophication to over-fishing to disease. Yet in terms of conservation, the coral reefs of Florida are something of a paradox. The environment in Florida superficially looks favorable to reef persistence, but the reefs are in far worse shape than other places, such as areas of Kiribati with greater disturbance. What is behind this poor resilience? Understanding the landscape of reef resilience is critical to smart and effective management. The complex nexus of factors presents a dizzying landscape to tackle with traditional single or two-factor field experiments. Thus, **the data-driven approach building on what was developed here allows exploration of the dimensions of variability that will shape out-planting success and give**

practical insight to managers for improving probability of success for projects like the Seven Iconic Reefs. The most critical, concrete step forward for an EDM approach to restoration would be demonstrating prediction of past/ongoing restoration outcomes in Florida, Hawaii, or elsewhere.

6.3.3 Resolving scale-dependence

In results and discussion, we highlighted contrasting evidence of spatial dynamics between our analysis of coarse scale (150m spatial scale, no taxonomic resolution within hard corals) and an empirical dynamics analysis of photomosaic quadrat data from Palmyra Atoll by McNamara et al. (76). Photomosaic quadrat data are being replicated under a common protocol at many reefs around the globe now under the 100 Island Challenge and many other cooperating efforts. Most importantly, they are being done sequentially in many places, so that these data can, in fact, directly resolve dynamic change. Thus, future EDM analysis could repeat many of our analysis to include additional islands from around the globe, and thus have a much higher chance to resolve spatial and species-specific dynamics. If this is the case, the reasonably good predictability of our coarse scale data is only a lower bound on what is possible with better and more data in hand.

Literature Cited

1. S. B. Munch, A. Giron-Nava, G. Sugihara, Nonlinear dynamics and noise in fisheries recruitment: A global meta-analysis. *Fish Fish.* (2018) https:/doi.org/10.1111/faf.12304.
2. A. Giron-Nava, *et al.*, Circularity in fisheries data weakens real world prediction. *Sci. Rep.* **10**, 1–6 (2020).
3. H. Ye, G. Sugihara, Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. *Science (80-.).* **353**, 922–925 (2016).
4. J. A. McGowan, *et al.*, Predicting coastal algal blooms in southern California. *Ecology* **98**, 1419–1433 (2017).
5. E. R. Deyle, R. M. May, S. B. Munch, G. Sugihara, Tracking and forecasting ecosystem interactions in real time. *Proc. R. Soc. B Biol. Sci.* **283** (2016).
6. M. Ushio, *et al.*, Fluctuating interaction network and time-varying stability of a natural fish community. *Nature* **554** (2018).
7. S. Cenci, L. P. Medeiros, G. Sugihara, S. Saavedra, Assessing the predictability of nonlinear dynamics under smooth parameter changes. *J. R. Soc. Interface* **17** (2020).
8. S. Tajima, T. Yanagawa, N. Fujii, T. Toyoizumi, Untangling Brain-Wide Dynamics in Consciousness by Cross-Embedding. *PLOS Comput Biol* **11**, e1004537 (2015).
9. E. R. Deyle, *et al.*, Predicting climate effects on Pacific sardine. *Proc. Natl. Acad. Sci.* **110**, 6430–6435 (2013).
10. S. M. Glaser, *et al.*, Complex dynamics may limit prediction in marine fisheries. *Fish Fish.* **15**, 616–633 (2014).
11. C. T. Perretti, S. B. Munch, G. Sugihara, Model-free forecasting outperforms the correct mechanistic model for simulated and experimental data. *Proc. Natl. Acad. Sci.* **110**, 5253–5257 (2013).
12. G. Sugihara, *et al.*, Detecting causality in complex ecosystems. *Science (80-.).* **338** (2012).

13. G. Sugihara, *et al.*, Detecting causality in complex ecosystems. *Science (80-.).* **338**, 496–500 (2012).

14. G. Sugihara, R. M. May, Nonlinear forecasting as a way of distinguishing chaos from measurement error in time-series. *Nature* **344**, 734–741 (1990).

15. A. Telschow, *et al.*, Infections of Wolbachia may destabilize mosquito population dynamics. *J. Theor. Biol.* **428**, 98–105 (2017).

16. M. Rypdal, G. Sugihara, Inter-outbreak stability reflects the size of the susceptible pool and forecasts magnitudes of seasonal epidemics. *Nat. Commun.* **10**, 1–8 (2019).

17. N. Nova, *et al.*, Empirical dynamic modeling reveals ecological drivers of dengue dynamics. *bioRxiv*, 2019.12.20.883363 (2019).

18. E. Deyle, A. M. Schueller, H. Ye, G. M. Pao, G. Sugihara, Ecosystem-based forecasts of recruitment in two menhaden species. *Fish Fish.* **19**, 769–781 (2018).

19. C. Sguotti, *et al.*, Non-linearity in stock–recruitment relationships of Atlantic cod: insights from a multi-model approach. *ICES J. Mar. Sci.* (2019) https:/doi.org/10.1093/icesjms/fsz113.

20. B. Baker, *et al.*, Status and Challenges for USACE Reservoirs: A Product of the National Portfolio Assessment for Water Supply Reallocations (2016).

21. C. C. Hsieh, C. Anderson, G. Sugihara, Extending nonlinear analysis to short ecological time series. *Am. Nat.* **171**, 71–80 (2008).

22. E. R. Deyle, R. M. May, S. B. Munch, G. Sugihara, Tracking and forecasting ecosystem interactions in real time. *Proc. R. Soc. B Biol. Sci.* **283**, 20152258 (2016).

23. H. D. I. Abarbanel, R. Brown, M. B. Kennel, Local Lyapunov Exponents Computed From Observed Data. *J. Nonlinear Sci.* **2**, 343–365 (1992).

24. B. A. Bailey, Local Lyapunov exponents: predictability depends on where you are. *Nonlinear Dyn. Econ.* (1996).

25. P. A. Dixon, M. J. Milicich, G. Sugihara, Episodic fluctuations in larval supply. *Science (80-.).* **283**, 1528–1530 (1999).

26. R. M. May, S. A. Levin, G. Sugihara, Complex systems: Ecology for bankers. *Nature* **451**, 893–895 (2008).

27. L. D. Jacobson, A. D. MacCall, Stock-recruitment models for Pacific sardine (Sardinops sagax). *Can. J. Fish. Aquat. Sci.* **52**, 566–577 (1995).

28. S. McClatchie, R. Goericke, G. Auad, K. Hill, Re-assessment of the stock–recruit and temperature–recruit relationships for Pacific sardine (Sardinops sagax). *Can. J. Fish. Aquat. Sci.* **67**, 1782–1790 (2010).

29. M. Lindegren, J. Checkley David M, Temperature dependence of Pacific sardine (Sardinops sagax) recruitment in the California Current Ecosystem revisited and revised. *Can. J. Fish. Aquat. Sci.* **70**, 245–252 (2013).

30. R. A. Myers, When do environment–recruitment correlations work? *Rev. Fish Biol. Fish.* **8**, 285–305 (1998).

31. M. Scheffer, S. Carpenter, J. A. Foley, C. Folke, B. Walker, Catastrophic shifts in ecosystems. *Nature* **413**, 591–596 (2001).

32. D. M. Post, M. E. Conners, D. S. Goldberg, Prey preference by a top predator and the stability of linked food chains. *Ecology* **81**, 8–14 (2000).

33. S. N. Wood, Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* **466**, 1102–1104 (2010).

34. G. Sugihara, R. M. May, Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature* **344**, 734–741 (1990).

35. E. R. Deyle, *et al.*, Predicting climate effects on Pacific sardine. *Proc. Natl. Acad. Sci.* **110**, 6430–6435 (2013).

36. C. W. J. Granger, Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica* **37**, 424 (1969).

37. C. C. Giron-Nava James Johnson, A.F., Dannecker, D., Kolody, B., Lee, A., Nagarkar, M., Pao, G.M., Ye, H., Johns, D.G. and Sugihara, G.,, A, Quantitative argument for long-term ecological monitoring. *Mar. Ecol. Prog. Ser.* **572**, 269–274 (2017).

38. A. T. Clark, *et al.*, Spatial convergent cross mapping to detect causal relationships from short time series. *Ecology* **96**, 1174–1181 (2015).

39. C. S. Holling, Resilience and Stability. *Annu. Rev. Ecol. Syst.* **4**, 1–23 (1973).

40. G. Sugihara, Nonlinear forecasting for the classification of natural time series. *Philos. Trans. R. Soc. London. Ser. A, Phys. Sci.* **348**, 477–495 (1994).

41. P. Dixon, M. J. Milicich, G. Sugihara, "Noise and nonlinearity in an ecological system BT - Nonlinear dynamics and statistics" in *Nonlinear Dynamics and Statistics*, A. I. Mees, Ed. (Birkhhauser, 2001), pp. 339–364.

42. M. Scheffer, *et al.*, Early-warning signals for critical transitions. *Nature* **461**, 53–59 (2009).

43. H. B. Torrey, An Unusual Occurrence of Dinoflagellata on the California Coast. *Am. Nat.* **36**, 187–192 (1902).

44. J.-B. Jouffray, *et al.*, Identifying multiple coral reef regimes and their drivers across the Hawaiian archipelago. *Philos. Trans. R. Soc. B Biol. Sci.* **370**, 20130268–20130268 (2014).

45. G. J. Williams, *et al.*, Benthic communities at two remote pacific coral reefs: Effects of reef habitat, depth, and wave energy gradients on spatial patterns. *PeerJ* **2013**, e81 (2013).

46. J. Gove, *et al.*, Coral reef benthic regimes exhibit non-linear threshold responses to natural physical drivers. *Mar. Ecol. Prog. Ser.* **522**, 33–48 (2015).

47. F. Isbell, D. Tilman, S. Polasky, S. Binder, P. Hawthorne, Low biodiversity state persists two decades after cessation of nutrient enrichment. *Ecol. Lett.* **16**, 454–460 (2013).

48. F. Takens, "Detecting strange attractors in turbulence" in *Dynamical Systems and Turbulence, Warwick 1980*, D. A. Rand, L. S. Young, Eds. (Lector Notes in Mathematics, 1981), pp. 366–381.

49. J. Stark, D. S. Broomhead, M. E. Davies, J. Huke, Takens embedding theorems for forced and stochastic systems. *Nonlinear Anal. Theory, Methods Appl.* **30**, 5303–5314 (1997).

50. J. Stark, Delay embeddings for forced systems. I. Deterministic forcing. *J. Nonlinear Sci.* **9.3**, 255–332 (1999).

51. J. Stark, D. S. Broomhead, M. E. Davies, J. Huke, Delay Embeddings for Forced Systems. II. Stochastic Forcing. *J. Nonlinear Sci.* **13**, 519–577 (2003).

52. T. Sauer, J. Yorke, M. Casdagli, Embedology. *J. Stat. Phys.* **65**, 579–616 (1991).

53. E. R. Deyle, G. Sugihara, Generalized theorems for nonlinear state space reconstruction. *PLoS One* **6**, e18295 (2011).

54. E. R. Deyle, G. Sugihara, Generalized theorems for nonlinear state space reconstruction. *PLoS One* **6** (2011).

55. J. Huisman, F. J. Weissing, Fundamental unpredictability in multispecies competition. *Am. Nat.* **157**, 488–494 (2001).

56. E. R. Deyle, M. C. Maher, R. D. Hernandez, S. Basu, G. Sugihara, Global environmental drivers of influenza. *Proc. Natl. Acad. Sci.* **113**, 13081–13086 (2016).

57. S. Ørstavik, J. Stark, Reconstruction and cross-prediction in coupled map lattices using spatio-temporal embedding techniques. *Phys. Lett. A* **247**, 145–160 (1998).

58. R. A. Horn, C. R. Johnson, *Topics in Matrix Analysis* (1991) https:/doi.org/10.1017/cbo9780511840371.

59. M. Scheffer, *et al.*, Early-warning signals for critical transitions. *Nature* **461**, 53–59 (2009).

60. A. Lucas, R. Pinkel, M. Alford, Ocean Wave Energy for Long Endurance, Broad Bandwidth Ocean Monitoring. *Oceanography* **30**, 126–127 (2017).

61. K. Lino, *et al.*, "Ecosystem Sciences Division standard operating procedures; data collection for towed-diver benthic and fish surveys." (2018).

62. P. Wessel, W. H. F. Smith, A global, self-consistent, hierarchical, high-resolution shoreline database. *J. Geophys. Res. Solid Earth* **101**, 8741–8743 (1996).

63. A. T. Clark, *et al.*, Spatial convergent cross mapping to detect causal relationships from short time series. *Ecology* **96**, 1174–1181 (2015).

64. S. Cenci, L. P. Medeiros, G. Sugihara, S. Saavedra, Assessing the predictability of nonlinear dynamics under smooth parameter changes. *J. R. Soc. Interface* **17**, 20190627 (2020).

65. S. Cenci, G. Sugihara, S. Saavedra, Regularized S-map for inference and forecasting with noisy ecological time series. *Methods Ecol. Evol.* **10**, 650–660 (2019).

66. M. B. Elowitz, S. Leibler, A synthetic oscillatory network of transcriptional regulators. *Nature* **403**, 335–338 (2012).

67. A. Huppert, B. Blasius, R. Olinky, L. Stone, A model for seasonal phytoplankton blooms. *J. Theor. Biol.* **236**, 276–290 (2005).

68. N. Van Oostende, *et al.*, Simulating the ocean's chlorophyll dynamic range from coastal upwelling to oligotrophy. *Prog. Oceanogr.* **168**, 232–247 (2018).

69. A. W. Mantyla, S. J. Bograd, E. L. Venrick, Patterns and controls of chlorophyll-a and primary productivity cycles in the Southern California Bight. *J. Mar. Syst.* **73**, 48–60 (2008).

70. M. Casdagli, S. Eubank, J. D. Farmer, J. F. Gibson, State space reconstruction in the presence of noise. *Phys. D Nonlinear Phenom.* **51**, 52–98 (1991).

71. S. Cenci, S. Saavedra, Non-parametric estimation of the structural stability of non-equilibrium community dynamics. *Nat. Ecol. Evol.* **3**, 912–918 (2019).

72. R. Schwefel, A. Gaudard, A. Wüest, D. Bouffard, Effects of climate change on deepwater oxygen

and winter mixing in a deep lake (Lake Geneva): Comparing observational findings and modeling. *Water Resour. Res.* **52**, 8811–8826 (2016).

73. T. Martz, *et al.*, Dynamic variability of biogeochemical ratios in the Southern California Current System. *Geophys. Res. Lett.* **41**, 2496–2501 (2014).

74. R. Abernethy, *et al.*, State of the climate in 2017. *Bull. Am. Meteorol. Soc.* **99**, Si-S310 (2018).

75. C. W. Cacciapaglia, R. van Woesik, Reduced carbon emissions and fishing pressure are both necessary for equatorial coral reefs to keep up with rising seas. *Ecography (Cop.).* **43**, 789–800 (2020).

76. D. E. McNamara, N. Cortale, C. Edwards, Y. Eynaud, S. A. Sandin, Insights into coral reef benthic dynamics from nonlinear spatial forecasting. *J. R. Soc. Interface* **16** (2019).

77. G. Sinnett, F. Feddersen, D. Lucas, G. Pawlak, E. Terrill, “Non-Linear Internal Waves Pulse Cold Water Into the Shallow Inner-Shelf and Surfzone” (2016).

7 Appendices

7.1 Appendix A: Supporting Data

No primary data were collected under this project. However, we include key pieces of code for implementing novel EDM approaches here not immediately accomplished with the published rEDM tools.

7.1.1 help_functions_data.R

This script contains several functions to process spatial data from geolocated observations into neighbor graphs for spatial EDM, (see 4.5.1).

7.1.2 help_functions_greedy_EDM.R

This script contains functions to conduct a greedy multivariate EDM model selection as shown in Figure 5.14 using randomized projection coordinates (see 4.4.2.1).

7.1.3 help_functions_Jacobian_analysis.R

This script contains functions to merge S-map coefficients across interacting variables and compute Jacobian metrics of stability (see 4.8).

7.2 Appendix B: Publications

7.2.1 Journal Articles

Cenci, S., Medeiros, L. P., Sugihara, G., & Saavedra, S. (2020). Assessing the predictability of nonlinear dynamics under smooth parameter changes. *Journal of the Royal Society Interface*, *17*(162), 20190627.

Giron-Nava, A., Munch, S. B., Johnson, A. F., Deyle, E., James, C. C., Saberski, E., ... & Sugihara, G. (2020). Circularity in fisheries data weakens real world prediction. *Scientific reports*, *10*(1), 1-6.

Cenci, S., Sugihara, G., & Saavedra, S. (2019). Regularized S-map for inference and forecasting with noisy ecological time series. *Methods in Ecology and Evolution*, *10*(5), 650-660.

Lee, S. W., Yon, D. K., James, C. C., Lee, S., Koh, H. Y., Sheen, Y. H., ... & Sugihara, G. (2019). Short-term effects of multiple outdoor environmental factors on risk of asthma exacerbations: Age-stratified time-series analysis. *Journal of Allergy and Clinical Immunology*, *144*(6), 1542-1550.

Runge, J., Bathiany, S., Bollt, E., Camps-Valls, G., Coumou, D., Deyle, E., ... & van Nes, E. H. (2019). Inferring causation from time series in Earth system sciences. *Nature communications*, *10*(1), 1-13.

Rypdal, M., & Sugihara, G. (2019). Inter-outbreak stability reflects the size of the susceptible pool and forecasts magnitudes of seasonal epidemics. *Nature communications*, *10*(1), 1-8.

Sguotti, C., Otto, S. A., Cormon, X., Werner, K. M., Deyle, E., Sugihara, G., & Möllmann, C. (2019). Non-linearity in stock–recruitment relationships of Atlantic cod: insights from a multi-model approach. *ICES Journal of Marine Science*.

Deyle, E., Schueller, A. M., Ye, H., Pao, G. M., & Sugihara, G. (2018). Ecosystem-based forecasts of recruitment in two menhaden species. *Fish and Fisheries*, *19*(5), 769-781.

Jajcay, N., Kravtsov, S., Sugihara, G., Tsonis, A. A., & Paluš, M. (2018). Synchronization and causality across time scales in El Niño Southern Oscillation. *npj Climate and Atmospheric Science*, *1*(1), 1-8.

Munch, S. B., Giron-Nava, A., & Sugihara, G. (2018). Nonlinear dynamics and noise in fisheries recruitment: A global meta-analysis. *Fish and Fisheries*, *19*(6), 964-973.

Sugihara, G., Criddle, K. R., McQuown, M., Giron-Nava, A., Deyle, E., James, C., ... & Ye, H. (2018). Comprehensive incentives for reducing Chinook salmon bycatch in the Bering Sea walleye Pollock fishery: Individual tradable encounter credits. *Regional Studies in Marine Science*, *22*, 70-81.

Ushio, M., Hsieh, C. H., Masuda, R., Deyle, E. R., Ye, H., Chang, C. W., ... & Kondoh, M. (2018). Fluctuating interaction network and time-varying stability of a natural fish community. *Nature*, *554*(7692), 360-363.

Dakos, V., Glaser, S. M., Hsieh, C. H., & Sugihara, G. (2017). Elevated nonlinearity as an indicator of shifts in the dynamics of populations under stress. *Journal of The Royal Society Interface*, *14*(128), 20160845.

Giron-Nava, A., James, C. C., Johnson, A. F., Dannecker, D., Kolody, B., Lee, A., ... & Sugihara, G. (2017). Quantitative argument for long-term ecological monitoring. *Marine Ecology Progress Series*, *572*, 269-274.

McGowan, J. A., Deyle, E. R., Ye, H., Carter, M. L., Perretti, C. T., Seger, K. D., ... & Sugihara, G. (2017). Predicting coastal algal blooms in southern California. *Ecology*, *98*(5), 1419-1433.

Storch, L. S., Glaser, S. M., Ye, H., & Rosenberg, A. A. (2017). Stock assessment and end-to-end ecosystem models alter dynamics of fisheries data. *PloS one*, *12*(2).

Sugihara, G., Deyle, E. R., & Ye, H. (2017). Reply to Baskerville and Cobey: Misconceptions about causation with synchrony and seasonal drivers. *Proceedings of the National Academy of Sciences*, *114*(12), E2272-E2274.

Telschow, A., Grziwotz, F., Crain, P., Miki, T., Mains, J. W., Sugihara, G., ... & Hsieh, C. H. (2017). Infections of Wolbachia may destabilize mosquito population dynamics. *Journal of theoretical biology*, *428*, 98-105.

Deyle, E. R., Maher, M. C., Hernandez, R. D., Basu, S., & Sugihara, G. (2016). Global environmental drivers of influenza. *Proceedings of the National Academy of Sciences*, *113*(46), 13081-13086.

Deyle, E. R., May, R. M., Munch, S. B., & Sugihara, G. (2016). Tracking and forecasting ecosystem interactions in real time. *Proceedings of the Royal Society B: Biological Sciences*, *283*(1822), 20152258.

Ye, H., & Sugihara, G. (2016). Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. *Science*, *353*(6302), 922-925.

Ye, H., Deyle, E. R., Gilarranz, L. J., & Sugihara, G. (2015). Distinguishing time-delayed causal interactions using convergent cross mapping. *Scientific reports*, *5*, 14750.

7.2.2 Conference Abstracts

Sugihara, G., Deyle, E., and May, R.M. "Equation-free mathematics shows ecological interactions are highly episodic". 2018 ESA Annual Meeting, New Orleans, LA, August 5-10, 2018.

Deyle, E., Sugihara, G., Ushio, M., and Hsieh, C.-H. "Dynamic stability: Empirical measurements and their insights". 2018 ESA Annual Meeting, New Orleans, LA, August 5-10, 2018.

James, C.C., Giron-Nava, A., Johnson, A.F., Dannecker, D.… Sugihara, G. "Do food webs capture important interactions: A survey of ecosystems". 2018 ESA Annual Meeting, New Orleans, LA, August 5-10, 2018.

Piao, L., Ye, H., Fu, Z., Pao, G.M., Tsonis, A.A., and Sugihara, G. "Causal factors contributing to global sea level change on different timescales". EGU General Assembly 2018, Vienna, Austria, April 8-13, 2018.

Ecosystem-Based Forecasts of Menhaden Recruitment Using Empirical Dynamic Modeling
E Deyle, AM Schueller, H Ye, G Sugihara. American Fisheries Society 147th Annual Meeting, Tampa, FL, August 21-24, 2017.

7.2.3 Other Publications

Tsonis, A. A., Deyle, E. R., Ye, H., & Sugihara, G. (2018). Convergent cross mapping: theory and an example. In *Advances in Nonlinear Geosciences* (pp. 587-600). Springer, Cham.

7.3 Appendix C: Other

7.3.1 rEDM User Guide

7.3.2 pyEDM User Guide

7.3.3 rEDM Code Review

Appendix C: Other

cppEDM Analysis Recommendations

Preliminary Final Report

March 1, 2020

Customer: Dr. George Sugihara, UCSD

Provider: Runtime Computing Solutions, Inc
Anthony Skjellum PhD
tony@runtimecomputing.com
https://www.runtimecomputing.com

Overview

cppEDM (C++ Library of EDM tools) is an open-source library of C++ code that provides Empirical Dynamic Modeling (EDM) support. It is authored by the Sugihara Laboratory and colleagues at UCSD. This project's purpose is to evaluate the code quality, design and coding processes, and offer recommendations that will enhance the software in the short, medium, and long terms. This code is developed in the open as well. This report describes version v1.3.1 of the code.

DevOPS/GitOPS

1. Use of Git (Development Processes)

Git is used, but it is used in an effective way, but it is a bit too simplistic. There is a master branch, an a single, special purpose branch:

```
  exclusion_matrix a455f0f [origin/exclusion_matrix] Merge branch 'exclusion_matrix' of
http://github.com/sugiharalab/cppEDM into exclusion_matrix
* master           fd21fe7 [origin/master] v1.3.1 Disjoint lib & pred.
```

There are a small number of developers, but this project is developed on a public Git repo.

The ‘develop’ branch should be added, and the master and develop branch should be protected. That will require ownership rights to accept pull request (PR). It appears anyone can update the source at present. Protecting it is easy using Github.

Reference 1 below describes an effective branching model for Git (a process now known as git-flow), which is to make master super stable, the develop branch have longer term work and pre-releases, and feature branches branched off develop for focused additions. This model should be used. Then, when a certain set of features checked into develop reach the point to meet the requirements for a major, minor, or maintenance release (resp 1.0.0, 1.1.0, 1.1.1 type number), then develop should be merged into master, that hash should be tagged using annotated Git tags. At present, tags are not used at all; rather, each check-in for a version has its version number as part of the comment. That means that the developers need to know what version they are working on when they make contributions, rather than making contributions through feature branches that total to a set of new capabilities and corrections that comprise a given release. This change will be easy, but extremely beneficial.

The developer count is low, the check-in rate is low, and the 2+ developers are not developing on this code every day. For these reasons, mechanisms that benefit large teams with high development rates also make sense for small teams like this one because they codify what’s going on and help the developers keep track after some periods of inactivity of the development.

It is recommended also that a *feat-featurename* type branch name be used for shorter-term branches that implement a feature. Features should be documented in the Issues tab on Github to give documentation, background, and quality (compliance to specification) info. This can all be light but is needed. When feature branches are checked into develop, then terminate. Feature branches can also apply to bug fixes equally well.

The Git history (git graph mode) shows the last five months of development check-ins:

```
* fd21fe7 - (7 days ago) v1.3.1 Disjoint lib & pred. - SoftwareLiteracy (HEAD
-> master, origin/master, origin/HEAD)
* fa71166 - (3 weeks ago) v1.3.0 call delete[] on array. - SoftwareLiteracy
* 696b07a - (4 weeks ago) v1.3.0 Docs - SoftwareLiteracy
* ac5f6ea - (4 weeks ago) v1.3.0 Doc update SMap. - SoftwareLiteracy
* 441e599 - (4 weeks ago) v1.3.0 SMap solver function pointer. -
SoftwareLiteracy
* 4916ae2 - (6 weeks ago) v1.2.1 Update docs. - SoftwareLiteracy
* dfad6c4 - (6 weeks ago) v1.2.1 Add doc/PackageBuild.pdf, update etc/Notes. -
SoftwareLiteracy
* b8ba283 - (6 weeks ago) v1.2.1 Add -Wreorder compiler flag for CRAN checks. -
SoftwareLiteracy
* c21f3cb - (7 weeks ago) v1.2.1 E = 0 check for shift.  DistanceMax query
instead of fixed macro. - SoftwareLiteracy
* 1dd0fc6 - (7 weeks ago) v1.2.1 Add CCM sample = 0 check. - SoftwareLiteracy
* 0e16a7e - (8 weeks ago) v1.2.1 Fix library bound check. - SoftwareLiteracy
* 3107e32 - (10 weeks ago) v1.2.0 Add positive tau. Default tau = -1. Set
default knn if embedded = true. - SoftwareLiteracy
* d467bdb - (10 weeks ago) v1.2.0 Add positive tau. Default tau = -1. Set
default knn if embedded = true. - SoftwareLiteracy
* b86c0c9 - (3 months ago) v1.1.0 CCM neighbors fix, parameter check. Add
%Y-%m-%d %H:%M:%S time format. - SoftwareLiteracy
* b84c4ce - (3 months ago) v1.0.1 Error estimates excluded end data. -
SoftwareLiteracy
* 7e28c54 - (3 months ago) v1.0.1 Change embedded E data columns error to
warning. - SoftwareLiteracy
* 641e73c - (3 months ago) v1.0.1 Init unrecognized_fmt - SoftwareLiteracy
* 9ceaf15 - (3 months ago) v1.0.1 Fix extension of datetime for Tp exceeding
data. - SoftwareLiteracy
* b86dd9b - (3 months ago) v1.0.1 Sync with pyEDM - SoftwareLiteracy
* cb01531 - (3 months ago) v1.0.1 Fix lib and pred index check. -
SoftwareLiteracy
* 30b8dea - (4 months ago) v1.0.1 lib and pred start index check. -
SoftwareLiteracy
* e2e29f0 - (4 months ago) v1.0.1 lib and pred start index check. -
SoftwareLiteracy
* f9baa67 - (4 months ago) v1.0.1 Add knn to PredictNonlinear() args. -
SoftwareLiteracy
* 7538f11 - (4 months ago) v1.0.1 Update OnlyDigits() - SoftwareLiteracy
* 8fe6391 - (4 months ago) v1.0.1 LAPACK Notes in etc/ - SoftwareLiteracy
* a775074 - (4 months ago) v1.0.1 fix windows makefile. - SoftwareLiteracy
* 884b33a - (4 months ago) v1.0.1 Doc update. - SoftwareLiteracy
* ed04145 - (4 months ago) v1.0.1 CCM data copy instead of reference. -
SoftwareLiteracy
* 4a84deb - (4 months ago) v1.0.1: SMap matrix construction. - SoftwareLiteracy
* 04764e1 - (5 months ago) v0.1.10: CCM: Don't DeletePartialDataRow() prior to
CrossMap(). - SoftwareLiteracy
* dd3fff6 - (5 months ago) v0.1.10 CCM() min libSize check. - SoftwareLiteracy
* 2ea3962 - (5 months ago) v0.1.10: SMap Lapack_SVD() kludge for MSVC/pyEDM. -
SoftwareLiteracy
* b29d8ec - (5 months ago) v0.1.10: Excise Eigen. Use LAPACK dgelss() for
s-map. - SoftwareLiteracy
* 87c5481 - (5 months ago) v0.1.10: Excise Eigen.  Use LAPACK dgelss for s-map.
- SoftwareLiteracy
* 94f185b - (5 months ago) v0.1.10: Excised Eigen. Use LAPACK dgelss() for
s-map SVD. - SoftwareLiteracy
```

All this work is directly on the master, so master is evolving with each check-in, rather than with a stable release cadence.

2. Other Aspects of the Development Process

Makefile recommendations

1) Make should work from the cppEDM main directory, even if this is a supervisory makefile that recurses into a Makefile into cppEDM and/or test directories, depending on what's requested.
2) 'make clean' (executed in the src directory) does not remove the lib/libEDM.a file
3) The Makefile is missing some potentially useful features (see Appendix B).

3. Review: Requirements Documents

To be added.

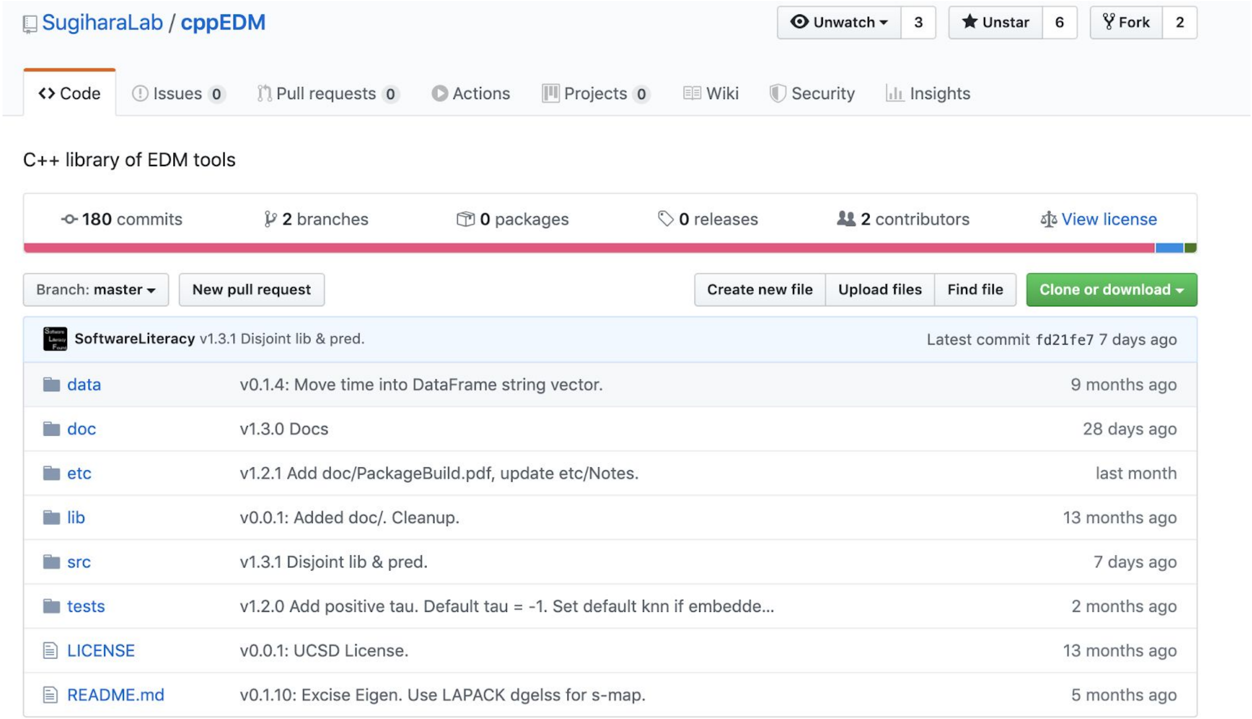
4. Review: Other Documentation

To be added.

Project and Code Structure, Complexity and Use of C++

1. Project Structure

The code is structured as follows in terms of layout in the directory structure (after the project has been built, to show where side effects occur):



C++ library of EDM tools

180 commits · 2 branches · 0 packages · 0 releases · 2 contributors · View license

SoftwareLiteracy v1.3.1 Disjoint lib & pred. Latest commit fd21fe7 7 days ago

data	v0.1.4: Move time into DataFrame string vector.	9 months ago
doc	v1.3.0 Docs	28 days ago
etc	v1.2.1 Add doc/PackageBuild.pdf, update etc/Notes.	last month
lib	v0.0.1: Added doc/. Cleanup.	13 months ago
src	v1.3.1 Disjoint lib & pred.	7 days ago
tests	v1.2.0 Add positive tau. Default tau = -1. Set default knn if embedde...	2 months ago
LICENSE	v0.0.1: UCSD License.	13 months ago
README.md	v0.1.10: Excise Eigen. Use LAPACK dgelss for s-map.	5 months ago

Directory-wise, this expands to:

```
cppEDM/:
src   tests   etc   doc   lib   data   README.md   LICENSE

cppEDM/src:
libEDM.a     CCM.o Neighbors.o   Parameter.o     Common.o        SMap.cc
Neighbors.cc  Embed.cc         Common.h      AuxFunc.cc     Version.h  Embed.h
SMap.o      Eval.o  Interface.o   DateTimeUtil.o   makefile        Parameter.h
Multiview.cc  DateTimeUtil.cc  Common.cc  build.02feb20    Neighbors.h
DateTime.h
```

```
Multiview.o  Simplex.o   Embed.o      AuxFunc.o       Simplex.cc  Parameter.cc
Eval.cc          DataFrame.h      CCM.cc    makefile.windows  Interface.cc
AuxFunc.h

cppEDM/tests:
makefile  data    SimplexTest.cc    SMapTest.cc  MultiviewTest.cc  CCMTest.cc
run   TestCommonTest.cc  TestCommon.h  TestCommon.cc   DateTimeTest.cc

cppEDM/tests/data:
Multiview_combos_valid.csv     Smplx_embd_block_3sp_pyEDM.csv
Smplx_E3_block_3sp_pyEDM.csv   Smap_circle_pyEDM.csv   CCM_anch_sst_pyEDM.csv
CCM_anch_sst_cppEDM_valid.csv  Smplx_S12CD_E3_pyEDM.csv
Smap_embd_block_3sp_pyEDM.csv  Multiview_pred_valid.csv

cppEDM/etc:
Test.cc  Notes    libstdc++_Notes.txt  PlotTest.R

cppEDM/doc:
cppEDM.pdf  PackageBuild.pdf

cppEDM/lib:
libEDM.a

cppEDM/data:
sardine_anchovy_sst.csv  circle_noise.csv  circle.csv  block_3sp.csv
TentMap_rEDM.csv  TentMapNoise_rEDM.csv  S12CD-S333-SumFlow_1980-2005.csv
LorenzData1000.csv
```

This project structure us usual and not surprising. It is fine as is, with one suggestion offered, as follows: The source directory could be further subdivided into C++ source files and header files, such as src/src (or similar) and src/include ; this is usual. It is helpful to segregate the header files from the main source files, but is not required.

The use of the .cc and .h extensions for file naming follows accepted conventions as well.

The requirement for compilers to support C++ 11 is reasonable and modest; use of additional C++14, 17, and 20 features in the future may be useful and should not be avoided. Widespread GCC/G++, CLANG, and ICC support for modern C++ is not an impediment to adoption.

2. Code Structure / Use of C++

The project structure already indicates that the library is composed from C++ sources (.cc) files supported by header files (.h) all in the src directory. This section considers design structural issues of the source itself.

I. Class structure

1) Inheritance canonical form requires

```
virtual ~Parameters();
```

[example] of the declaration of the destructors in each class. This is a best practice, in case the project ever wants to do inheritance off a given class.

2) If you ever want to inherit from these objects, use 'protected' explicitly, instead of private (implicitly)

The keyword private appears only once in the entire project, under DataFrame.h

Explicitly naming what's private is good practice too, even though its the default for C++ classes.

II. Namespaces

1. EDM should be the namespace for everything in this library.

They are used, but ad hoc, now:

```
AuxFunc.cc:namespace EDM_AuxFunc {
CCM.cc:namespace EDM_CCM {
Eval.cc:namespace EDM_Eval {
Multiview.cc:namespace EDM_Multiview {
Neighbors.cc:namespace EDM_Neighbors {
```

They are used sparingly for quite specific purposes; this should change to EDM::Neighbors, for instance. All published APIs, global constants, and singleton objects (if any), should live in the EDM namespace.

2. `using namespace std;` in each source file would allow the use of the STL and simplify coding (eliminates std::).

III. Use of float vs double; there are few scalar parameters are in float; deciding when to use float vs. double should be clarified in the design. In fact, these appear to be “dangerous” to be float when the rest of the library is double, given that they deal with tolerances; recommendation is to change these to double:

```
Parameter.h:    float      theta;              // S Map localization
Parameter.h:    float      SVDSignificance;    // SVD singular value
cutoff
Parameter.h:    float      TikhonovAlpha;      // Initial alpha
parameter
Parameter.h:    float      ElasticNetAlpha;    // Initial alpha
parameter
Parameter.h:        float      theta       = 0,
Parameter.h:        float      svdSig       = 1E-5,
Parameter.h:        float      tikhonov     = 0,
Parameter.h:        float      elasticNet   = 0.1,
Parameter.cc:    float      theta,
Parameter.cc:    float      svdSig,
Parameter.cc:    float      tikhonov,
Parameter.cc:    float      elasticNet,
```

IV. Variable naming; there is no set convention; some camelCase, some N_xxx, some all lowercase. Not a big deal, you might want to set a standard.

V. Use of by const and/or reference parameter passing (no change to correctness, only change to performance for the better implied):

1. Example:

```
   void PrintIndices( std::vector<size_t> library,
                      std::vector<size_t> prediction );
```

These should be
a) passed like this,
b) and passed as const if they are in parameters:

Parameter.h:

```
class Parameter {

...
    void PrintIndices( const std::vector<size_t> &library,
                       const std::vector<size_t> &prediction );

};
```

const references are much faster for large objects, and won't change the syntax of the function.

Neighbors.h:

```
Neighbors FindNeighbors( DataFrame<double> dataFrame,
                         Parameters        parameters );
this could better be:
Neighbors &FindNeighbors( const DataFrame<double> &dataFrame,
                          const Parameters        &parameters );
```

while these use const refs:

```
void PrintDataFrameIn( const DataFrame<double> &dataFrame,
                       const Parameters        &parameters );


void PrintNeighborsOut( const Neighbors &neighbors );

double Distance( const std::valarray<double> &v1,
                 const std::valarray<double> &v2,
                 DistanceMetric metric );
```

Embed.h

There are these functions:

```
DataFrame< double > Embed ()
```

These functions would better be declared as:

```
DataFrame< double >& Embed ()
```

This function

```
DataFrame< double > MakeBlock ( DataFrame< double >      dataFrame,
                                int                      E,
                                int                      tau,
                                std::vector<std::string> columnNames,
                                bool                     verbose );
```

would better be done as:

```
DataFrame< double >& MakeBlock ( const DataFrame< double > &dataFrame,
                                 int                       E,
                                 int                       tau,
                                 std::vector<std::string> &columnNames,
                                 bool                      verbose );
```

DateTime.h:

Similar opportunities to use `const &` as input arguments, and `std::string &` instead of `std::string` in output.

DataFrame.h:

Similar opportunities to use `const &` as input arguments, and `std::string &` instead of `std::string` in output.

Example (R-value and L-value accessors); the L-value accessor is correct

```
    std::valarray<T>  Elements() const { return elements; }
    std::valarray<T> &Elements()       { return elements; }
```

The R-value accessor could also be done as follows:

```
    const std::valarray<T>  &Elements() const { return elements; }
```

Otherwise, the whole valarray is copied.

Example program to show const ref return as R-value:

```
using namespace std;

template <typename T>
class testValarray
{
public:

  testValarray(int _N) : N(_N) {a = new valarray<T>(N);}
  virtual ~testValarray() {delete a;}

  valarray<T> &Elements() {return *a;}
  const valarray<T> &Elements() const {return *a;}

  typedef T classType;

protected:

  int N;
  valarray<T> *a;

};

int main(void)
{
   testValarray<double>       alpha(10);
   const testValarray<double> beta(10);

   valarray<double> gamma = alpha.Elements();
   valarray<double> delta = beta.Elements();

   return 0;
}
```

Common.h:

```
// forward declaration for SMap solver
std::valarray < double > SVD( DataFrame    < double > A,
                              std::valarray< double > B );


    MultiviewValues( DataFrame< double >        combo_rho,
                     DataFrame< double >        predictions,
                     std::vector< std::string > combo_rho_table ):
```

should also use `&`'s and `const &`'s.

This declaration is fine, but it is the C-style of passing function pointer:

```
SMapValues SMap( std::string pathIn          = "./data/",
...
                 std::valarray<double> (*solver)(DataFrame < double
>,
                                             std::valarray < double
>) = &SVD,
...
};
```

As a matter of principle, using Functor (function objects) is a desirable alternative to consider in future, for instance, in an object-oriented C++ refactoring that uses the full power of C++11 … C++20.

Please see also:

https://www.geeksforgeeks.org/functors-in-cpp/

This is not urgent, but design-wise, it is more of the C++ way of doing things.

AuxFunc.h:

Similar recommendations for use of pass by reference for everything R/W, and `const &` for things that are in-arguments.

Memory leaks / static & dynamic testing:

Cppcheck:

Checking SMap.cc ...

```
SMap.cc:471:9: error: Memory leak: iwork [memleak]
        throw std::runtime_error( "Lapack_SVD(): dgelss failed on
query.\n" );
        ^
SMap.cc:489:9: error: Memory leak: iwork [memleak]
        throw std::runtime_error( "Lapack_SVD(): dgelss failed.\n" );
        ^
```

These appear real; a workaround will be recommended in a pull request. It requires a catch and rethrow to delete local heap temporaries.

Valgrind testing:

TO BE DONE STILL ON EXAMPLE TEST PROGRAM.

3. Code Complexity

No concerns here; more comments may be made in the revised final report.

Overall Recommendations

1. Use Git with more features in key ways:
    a. The stable master model should be adopted [See Ref #1].
    b. The master and develop branches (develop will come soon) should be protected.
    c. All check-ins should be code reviewed through a pull-request
    d. Git annotated tags should be used to mark releases (e.g., 1.4.0)
    e. The use of Git with the project should be documented briefly on the Wiki
2. The Makefile should be enhanced incrementally as described above. Potentially, as more compiler options are selected GCC, CLANG/LLVM, and ICC Makefiles will be useful, but right now that would be overkill.
3. The Makefile should (eventually) provide a DLL (.so) file option. In large systems, and with many processes going, it is market standard to offer a .so variant, not just a static library. For now, this should be low priority until an important customer complains.
4. Documentation on the requirements for the library should be retroactively created by interviewing the developers further and creating practical enumeration of functional and

non-functional expectations for the library. This will aid with further development and as developers change over time.

5. A library namespace should be used for the cppEDM source C++ code (i.e., EDM::).
6. The published APIs (doc/cppEDM.pdf) should be declared stable with a defined commitment (or lack thereof) to retaining backward compatibility defined (that's a policy decision). Examples policies for API stability follow: permanently, per release, subject to a period of deprecation then deletion or change, or no promises. Rules for breaking backward compatibility by the developers should be written down in the documentation processes for code reviews and acceptance of pull requests.
7. Known tested compilers/OSes should be documented. For instance, the use of GCC/G++ 9.1 and 8.3 (on x86-64) posed no issues for the project to build.
8. Dependencies on third-party libraries other deserve baseline documentation. For instance, the use of LAPACK-type libraries is mentioned specifically but there is more than one API-compliant form of LAPACK besides the NETLIB version that can be superior (e.g., BLIS from UT Austin). If there should be known dependencies on minimum version numbers, these should be captured. The author believes this is unlikely at present but might become more important in the future.
9. Interactions with multicore modes of operation and concurrency both at the cppEDM level and at the LAPACK/BLAS levels is unclear and deserves additional study, clarification, and potentially additional, user-level tuning mechanisms.
10. Some type of information should be provided on how third parties can submit pull requests on forks or clones of the library for consideration including a) a developer agreement, b) a process for accepting/reviewing/rejecting such third-party contributions.
11. This code has only been run on x86-64 systems; there is no reason to believe it would not work on Power9, 64-bit ARM, and even 32-bit platforms with some careful considerations (although that might not seem useful to the authors, given their use cases). 64-bit processors appear to be no problem whatsoever. Testing on the IBM Linux for Power9 would be an easy but valuable exercise but is evidently not urgent.
12. The code structure is fine and there are no observed problems. For the size of the code, development team, and development cadence, this structure works fine and is commonly found elsewhere in mathematical software.

Conclusion

cppEDM is high-quality scientific software that can benefit from small and incremental additional investments in DevOPS practices and documentation. Overall it is sound and has the potential to continue to be developed and enhanced for the foreseeable future.   No major concerns were noted but lots of opportunities to enhance what is already strong and quality were noted.

References

1. Useful Git model: https://nvie.com/posts/a-successful-git-branching-model/
   This model is used widely in industry; for instance, it is widely accepted in Silicon Valley companies as a useful guide for developers. It has been a standard practice for about 10 years.

Appendices

A. Recommended Git macros (put in ~user/.gitconfig):

```
[alias]
    graph = !"git lg"
    lg = !"git lg1"
    lg1 = !"git lg1-specific"
    lg2 = !"git lg2-specific --all"
    lg3 = !"git lg3-specific --all"

    lg1-specific = log --graph --abbrev-commit --decorate
--format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset)
%C(white)%s%C(reset) %C(dim white)- %an%C(r
eset)%C(bold yellow)%d%C(reset)'
    lg2-specific = log --graph --abbrev-commit --decorate
--format=format:'%C(bold blue)%h%C(reset) - %C(bold cyan)%aD%C(reset) %C(bold
green)(%ar)%C(reset)%C(bold yellow)%d%
C(reset)%n''          %C(white)%s%C(reset) %C(dim white)- %an%C(reset)'
    lg3-specific = log --graph --abbrev-commit --decorate
--format=format:'%C(bold blue)%h%C(reset) - %C(bold cyan)%aD%C(reset) %C(bold
green)(%ar)%C(reset) %C(bold cyan)(com
mitted: %cD)%C(reset) %C(bold yellow)%d%C(reset)%n''
%C(white)%s%C(reset)%n''          %C(dim white)- %an <%ae> %C(reset) %C(dim
white)(committer: %cn <%ce>)%C(reset
)’
```

In particular, this enables that command “git graph”

B. Recommended makefile changes

A pull request for this modified makefile will be provided. Here it is for reference:

```
.PHONY: all clean distclean depend

CC  = g++
HEADERS = AuxFunc.h  Common.h  DataFrame.h  DateTime.h  Embed.h  Neighbors.h
Parameter.h  Version.h
SRCS    = Common.cc AuxFunc.cc DateTimeUtil.cc Parameter.cc Embed.cc
Interface.cc\
          Neighbors.cc Simplex.cc Eval.cc CCM.cc Multiview.cc SMap.cc
OBJ     = $(SRCS:%.cc=%.o)

LIB = libEDM.a

CFLAGS += -std=c++11 -DCCM_THREADED -DMULTIVIEW_VALUES_OVERLOAD -O3 -Wreorder #
-g -DDEBUG -DDEBUG_ALL
# optional for heavier testing:
CFLAGS += -Wpedantic -Wall -Wextra
#
LFLAGS = -L./ -lstdc++ -lEDM -lpthread # -llapacke -llapack -lblas

all:    $(LIB)
        cp $(LIB) ../lib/

clean:
        rm -f $(OBJ) $(LIB)

distclean:
        rm -f $(OBJ) $(LIB) ../lib/$(LIB) *~ *.bak *.csv

$(LIB): $(OBJ)
        ar -rcs $(LIB) $(OBJ)

%.o : %.cc
        $(CC) $(CFLAGS) -c $<

depend:
        @echo ${SRCS}
        makedepend -Y $(SRCS) -w160
# DO NOT DELETE

Common.o: Common.h DataFrame.h
AuxFunc.o: AuxFunc.h Common.h DataFrame.h Neighbors.h Parameter.h Version.h
Embed.h DateTime.h
```

```
DateTimeUtil.o: DateTime.h
Parameter.o: Parameter.h Common.h DataFrame.h Version.h
Embed.o: Embed.h Common.h DataFrame.h Parameter.h Version.h
Interface.o: Common.h DataFrame.h
Neighbors.o: Neighbors.h Common.h DataFrame.h Parameter.h Version.h
Simplex.o: Common.h DataFrame.h Parameter.h Version.h Neighbors.h Embed.h
AuxFunc.h
Eval.o: Common.h DataFrame.h
CCM.o: Common.h DataFrame.h Embed.h Parameter.h Version.h AuxFunc.h Neighbors.h
Multiview.o: Common.h DataFrame.h AuxFunc.h Neighbors.h Parameter.h Version.h
Embed.h
SMap.o: Common.h DataFrame.h Parameter.h Version.h Embed.h Neighbors.h
AuxFunc.h
```

pyEDM Version 1.15.3 November 31, 2023

pyEDM is a Python package interface to the cppEDM C++ library of empirical dynamic modeling (EDM) algorithms. It returns Pandas DataFrame objects, or Python dictionaries of Pandas DataFrames. pyEDM is hosted on the Python Package Index (PyPI) at pypi/pyEDM. A Jupyter notebook GUI is available at jpyEDM.

Table of Contents

University of California at San Diego
Scripps Institution of Oceanography
Sugihara Lab

Joseph Park, Cameron Smith

Introduction

pyEDM is a Python interface to the C++ library cppEDM. Input and output objects are based on Pandas DataFrame objects. Core algorithms are listed in Table 1.

Algorithm	API Interface	Reference
Simplex projection	`Simplex()`	Sugihara and May (1990)
Sequential Locally Weighted Global Linear Maps (S-map)	`SMap()`	Sugihara (1994)
Predictions from multivariate embeddings	`Simplex(), SMap()`	Dixon et. al. (1999)
Convergent cross mapping	`CCM()`	Sugihara et. al. (2012)
Multiview embedding	`Multiview()`	Ye and Sugihara (2016)

Convenience functions to prepare and evaluate data are listed in Table 2.

Function	Purpose	Parameter Range
`Embed()`	Timeseries delay dimensional embedding	User defined
`MakeBlock()`	Timeseries delay dimensional embedding	User defined
`EmbedDimension()`	Evaluate prediction skill vs. embedding dimension	E = [1, 10]
`PredictInterval()`	Evaluate prediction skill vs. forecast interval	Tp = [1, 10]
`PredictNonlinear()`	Evaluate prediction skill vs. SMap nonlinear localisation	θ = 0.01, 0.1, 0.3, 0.5, 0.75, 1, 1.5, 2, 3, 4, 5, 6, 7, 8, 9
`ComputeError()`	Pearson ρ, RMSE, MAE	
`Examples()`	Example function calls and plots	

Installation

There are two methods to install pyEDM:
1) Python Package Index and pip, which is only supported for certain OSX and Windows platforms
2) Download, build and install package.

Python Package Index (PyPI)

Certain Mac OSX and Windows platforms are supported with prebuilt binary distributions and can be installed using the Python pip module. The module is located at pypi.org/project/pyEDM/.

Installation can be executed as:

```
python -m pip install pyEDM --trusted-host pypi.org --trusted-host
files.pythonhosted.org pyEDM
```

Manual Compilation

Unfortunately, we do not have the resources to provide pre-built binary distributions for all computer platforms. In this case the user is required to first build the cppEDM library on their machine, and then install the Python package using pip. On OSX and Linux this requires gcc and the LAPACK library. On Windows, mingw from MSYS2 can be used.

OSX and Linux

1) Download pyEDM: git clone https://github.com/SugiharaLab/pyEDM

2) Build cppEDM library:

```
cd pyEDM/cppEDM/src
make
```

3) Build and install package:

```
cd ../..
python -m pip install . --user --trusted-host pypi.org
```

Windows

We do not have resources to maintain windows build support. These suggestions may be useful.

Requires mingw installation. MSYS2 provides a mingw package.
1) Download pyEDM: git clone https://github.com/SugiharaLab/pyEDM
2) Build cppEDM library: `cd pyEDM\cppEDM\src; make`
3) Build and install package in `pyEDM\` : `python -m pip install . --user`

Usage

```
>>> import pyEDM
>>> pyEDM.Examples()
```

See the Examples section below.

All data input files are assumed to be in .csv format or Pandas DataFrame.

> **The files are required to have a single line header with column names.**
>
> **It is expected the first column is a vector of times or time indices.** This can be disabled with the `noTime = True` parameter.

Parameters

API parameter names and purpose are listed in Table 3.

Parameter	Type	Default	Purpose
`pathIn`	`string`	`"./"`	Input data file path
`dataFile`	`string`	`""`	Data file name
`dataFrame`	`Pandas DataFrame`	`None`	Input DataFrame
`pathOut`	`string`	`"./"`	Output file path
`predictFile`	`string`	`""`	Prediction output file
`smapCoefFile`	`string`	`""`	SMap coefficient output file
`smapSVFile`	`string`	`""`	SMap singular values output file
`lib`	`string`	`""`	library start : stop row indices
`pred`	`string`	`""`	prediction start : stop row indices
`D`	`int`	`0`	Multiview state-space dimension
`E`	`int`	`0`	Embedding dimension
`Tp`	`int`	`0 or 1`	Prediction interval
`knn`	`int`	`0`	Number nearest neighbors
`tau`	`int`	`-1`	Embedding delay
`theta`	`float`	`0`	SMap localisation
`exclusionRadius`	`int`	`0`	Prediction vector exclusion radius
`columns`	`string`	`"" or []`	Column names or indices for library
`target`	`string`	`""`	Target library column name or index
`embedded`	`bool`	`False`	Is data an embedding?
`const_pred`	`bool`	`False`	Include non projected forecast data
`ignoreNan`	`bool`	`True`	SMap detect and remove nan from lib
`verbose`	`bool`	`False`	Echo messages
`validLib`	`[ bool ]`	`[]`	Conditional Embedding (CE)
`noTime`	`bool`	`False`	Do not require first column to be time
`generateSteps`	`int`	`0`	Simplex, SMap feedback prediction
`generateLibrary`	`bool`	`False`	Increment EDM library with feedback
`parameterList`	`bool`	`False`	Return Parameters map
`smapFile`	`string`	`""`	SMap coefficient output file
`solver`	`sklearn.linear_model`	`None`	SMap solver
`multiview`	`int`	`0`	Number of ensembles, 0 = sqrt(N)
`trainLib`	`bool`	`True`	Multiview use lib as training library
`excludeTarget`	`bool`	`True`	Multiview exclude target from combos
`libSizes`	`string or [ int ]`	`""`	CCM library sizes
`sample`	`int`	`0`	CCM number of random samples
`random`	`bool`	`True`	CCM use random samples?
`replacement`	`bool`	`False`	CCM sample with replacement?
`includeData`	`bool`	`False`	CCM include all projections in return
`seed`	`unsigned`	`0`	CCM RNG seed, 0 = random seed
`method`	`string`	`"ebisusaki"`	SurrogateData method
`alpha`	`float`	`range / 5`	SurrogateData seasonal noise std dev
`smooth`	`float`	`0.8`	SurrogateData seasonal spline smooth

Application Programming Interface (API)

Embed

Create a data block of Takens (1981) time-delay embedding from each of the columns in the csv file or dataFrame. The `columns` parameter can be a list of column names, or a list of column indices. If `columns` is a list of indices, then column names are created as V1, V2...

Note: The returned DataFrame will have `tau*(E-1)` fewer rows than the input data from the removal of partial vectors as a result of the embedding.

Note: The returned DataFrame will not have the time column.

```
//------------------------------------------------------------------
//
//------------------------------------------------------------------
DataFrame Embed ( path      = "./",
                  dataFile  = "",
                  dataFrame = None,
                  E         = 0,
                  tau       = -1,
                  columns   = "",
                  verbose   = False )


//------------------------------------------------------------------
//
//------------------------------------------------------------------
DataFrame MakeBlock ( dataFrame,
                      E             = 0,
                      tau           = -1,
                      columnNames   = "",
                      deletePartial = False )
```

Simplex

Simplex projection of the input data file or DataFrame. If `parameterList = False`, (default) the returned object is a `DataFrame` with 3 columns : "Time", "Observations", "Predictions". nan values are inserted where there is no observation or prediction. If `parameterList = True`, a dictionary with keys: `predictions, parameters` is returned with the respective dictionary values the predictions `DataFrame` and parameter dictionary.

See the Parameters table for parameter definitions.

Parameters

`lib` and `pred` specify [start stop] row indices of the input data for the library and predictions.

If `embedded` is `False` the data columns are embedded to dimension `E` with delay `tau`. If `embedded` is `True` the data columns are assumed to be a multivariable data block.

If `knn` is not specified, and `embedded` is `False`, it is set equal to `E+1`. If `embedded` is `True, knn` is set equal to the number of `columns + 1`.

`exclusionRadius` defines the number of library rows excluded from the state-space library with respect to a temporal "radius" from the prediction state. If `exclusionRadius = 1,` library state-space points from observation time series that are within ±1 sequential observation row of the prediction state are not included in the library. Note that units of the radius are time series rows, not time values.

`validLib` implements conditional embedding (CE). It is a boolean vector the same length as the number of time series rows. A `False` entry means that the state-space vector derived from the corresponding time series row will not be included in the state-space library.

If `columns` is a string and column names have whitespace, delimit the columns with "," or, place the column names in a list.

If `parameterList = True`, then `parameters` is populated and returned in a dictionary.

If `generateSteps` > `0`, then `Simplex` operates in feedback generative mode. The values of `pred` are over-riden to start at the end of the data. At each step one prediction is made, added to the `columns` data, a new time-delay embedded is created, and the cycle repeated for `generateSteps`. Feedback generation only operates on a univariate time series that is time-delay embedded. The `columns` and `target` variables must be the same. If `generateLibrary` is `false` the state-space library is not expanded as predictions are generated, it is static. If `generateLibrary` is `true` the state-space library has the generated prediction added to the library at each step.

If `noTime = False`, the first column of the input DataFrame or .csv file must be an index or time column. If `noTime = True` an index or time column is not required.

```
//----------------------------------------------------------------
//
//----------------------------------------------------------------
DataFrame Simplex( pathIn          = "./",
                   dataFile        = "",
                   dataFrame       = None,
                   pathOut         = "./",
                   predictFile     = "",
                   lib             = "",
                   pred            = "",
                   E               = 0,
                   Tp              = 1,
                   knn             = 0,
                   tau             = -1,
                   exclusionRadius = 0,
                   columns         = "",
                   target          = "",
                   embedded        = False,
                   verbose         = False
                   const_pred      = False,
                   showPlot        = False,
                   validLib        = [],
                   generateSteps   = 0,
                   generateLibrary = False,
                   parameterList   = False,
                   noTime          = False )
```

SMap

SMap projection of the input data file or DataFrame. See the Parameters table for parameter definitions.

`SMap()` returns a `dict` with three DataFrames:

```
dict { predictions    : DataFrame,
       coefficients   : DataFrame,
       singularValues : DataFrame
}
```

The `predictions` `DataFrame` has 3 columns "Time", "Observations", "Predictions". nan values are inserted where there is no observation or prediction. If `predictFile` is provided the `predictions` will be written to it in csv format.

The `coefficients` `DataFrame` will have `E+2` columns. The first column is the "Time" vector, the remaining `E+1` columns are the SMap SVD fit coefficients. The first column "C0" is the bias term, following coefficients are ∂ target / ∂ columns[i] .

The `singularValues` `DataFrame` holds SVD singular values if the default LAPACK solver, or scikit.learn LinearRegression solver is used. The scikit.learn LinearRegression does not return a sigular value for the intercept (bias) term.

If `parameterList = True`, a dictionary with `parameters` is added to the returned object.

Parameters

`lib` and `pred` specify [start, stop] row indices of the input data for the library and predictions.

If `embedded` is `False` the data columns are embedded to dimension `E` with delay `tau`. If `embedded` is `True` the data columns are assumed to be a multivariable data block. If `smapFile` is provided the `coefficients` will be written to it in csv format.

If `columns` is a string and column names have whitespace, delimit the columns with “,” or, place the column names in a list.

If `parameterList = True`, a dictionary with `parameters` is added to the returned object.

If a multivariate data set is used (number of `columns > 1`) it must use `embedded = true` with `E` equal to the number of `columns`. This prevents the function from internally time-delay embedding the multiple columns to dimension E. If the internal time-delay embedding is performed, then state-space columns will not correspond to the intended dimensions in the matrix inversion, coefficient assignment, and prediction. In the multivariate case, the user should first prepare the embedding (using `Embed()` for time-delay embedding if desired), then pass this embedding to `SMap` with appropriately specified `columns`, `E`, and `embedded = true`.

If knn is not specified, it is set equal to the library size. If knn is specified, it must be greater than E+1.

exclusionRadius defines the number of library rows excluded from the state-space library with respect to a temporal "radius" from the prediction state. If exclusionRadius = 1, library state-space points from observation time series that are within ±1 sequential observation row of the prediction state are not included in the library. Note that units of the radius are time series rows, not time values.

If noTime = False, the first column of the input DataFrame or .csv file must be an index or time column. If noTime = True an index or time column is not required.

ignoreNan automatically redefines the library to avoid nan observations and associated state vectors. If ignoreNan is false the library is not changed. The user can manually specify library row segements to ignore nan values.

validLib implements conditional embedding (CE). It is a boolean vector the same length as the number of time series rows. A false entry means that the state-space vector derived from the corresponding time series row will not be included in the state-space library.

If generateSteps > 0, then SMap operates in feedback generative mode. The values of pred are over-riden to start at the end of the data. At each step one prediction is made, added to the columns data, a new time-delay embedded is created, and the cycle repeated for generateSteps. Feedback generation only operates on a univariate time series that is time-delay embedded. The columns and target variables must be the same. If generateLibrary is false the state-space library is not expanded as predictions are generated, it is static. If generateLibrary is true the state-space library has the generated prediction added to the library at each step.

The default solver for the SMap coefficient matrix is the LAPACK SVD function dgelss(). If the default solver is used, SMap singular values are returned. This can be replaced with a user-instantianted class object from the python sklearn.linear_model: Linear Models. Supported solvers include: LinearRegression, Ridge, Lasso, ElasticNet, RidgeCV, LassoCV, ElasticNetCV. See the pyEDM/tests/smapSolverTest.py script for examples.

```
//----------------------------------------------------------------
//
//----------------------------------------------------------------
dict SMap( pathIn          = "./",
           dataFile        = "",
           dataFrame       = None,
           pathOut         = "./",
           predictFile     = "",
           lib             = "",
           pred            = "",
           E               = 0,
           Tp              = 1,
           knn             = 0,
           tau             = -1,
           theta           = 0,
           exclusionRadius = 0,
           columns         = "",
           target          = "",
           smapCoefFile    = "",
           smapSVFile      = "",
           solver          = None,
           embedded        = False,
           verbose         = False,
           const_pred      = False,
           showPlot        = False,
           validLib        = [],
           ignoreNan       = True,
           generateSteps   = 0,
           generateLibrary = False,
           parameterList   = False,
           noTime          = False )
```

CCM

Convergent cross mapping of `columns` against `target` via Simplex. Normally, one `column` and one `target` are specified. The `column` time series is time-delay embedded to dimension E, cross mapped with the `target` time series. The target time series is then embedded to E and cross mapped against the column as the "target" time series, not an embedding.

If there are multiple `columns` and `embedded` is false, each column is time-delay embedded to dimension `E` creating an N-columns * E dimensional "mixed" embedding. If `embedded` is true, no time-delay embedding is done, creating a multivariate embedding of the speficied columns. The same logic applies if multiple `target` are specified for the "reverse" mapping. If `embedded` is false, each target is time-delay embedded to dimension `E` creating an N-target * E dimensional "mixed" embedding cross mapped to the first `column` as the cross map target. If `embedded` is true, no time-delay embedding is done, creating a multivariate embedding of the speficied target(s).

Cross mappings are performed between `column : target`, and, `target : column` in separate threads. If multiprocessing is applied, halve the number of proccessors.

See the Parameters table for parameter definitions.

If `includeData` is False, the returned `DataFrame` has 3 columns. The first column is "LibSize", the second and third columns are Pearson correlation coefficients for "column : target" and "target : column" cross mapping. If `includeData` is True, a `dict` is returned with the `LibMeans` `DataFrame` , and a `DataFrames` of prediction statistics for all predictions in the ensembles. If `includeData` is `True`, and `parameterList = True`, then `parameters` dictionary is added to the return object.

Parameters

`The libSizes` parameter is a string of whitespace or comma separated library sizes. If the string has 3 values, and, if the third value is less than the second value, then the three values are interpreted as a sequence generator specifying "start stop increment" row values, i.e. "10 80 10" will evaluate library sizes from 10 to 80 in increments of 10.

If `random` is `true`, `sample` observations are radomly selected from the subset of each library size. If `seed=0`, then a random seed is generated for the random number generator. Otherwise, `seed` is used to initialise the random number generator.

If `random` is false, `sample` is ignored and contiguous library rows up to the current library size are used.

If `noTime = False`, the first column of the input DataFrame or .csv file must be an index or time column. If `noTime = True` an index or time column is not required.

Note: Cross mappings are performed between `column : target`, and `target : column`. The default is to do this in separate threads. Threading can be disabled in the makefile by removing `-DCCM_THREADED`.

Note: The entire prediction vector is used in the Simplex prediction at each library subset size.

```
//----------------------------------------------------------------
//
//----------------------------------------------------------------
DataFrame or dict CCM( pathIn          = "./",
                       dataFile        = "",
                       dataFrame       = None,
                       pathOut         = "./",
                       predictFile     = "",
                       E               = 0,
                       Tp              = 0,
                       knn             = 0,
                       tau             = -1,
                       exclusionRadius = 0,
                       columns         = "",
                       target          = "",
                       libSizes        = "",
                       sample          = 0,
                       random          = True,
                       replacement     = False,
                       seed            = 0,     // seed=0: use RNG
                       embedded        = False,
                       includeData     = False,
                       parameterList   = False,
                       verbose         = False,
                       showPlot        = False,
                       noTime          = False );
```

Multiview

Multiview embedding and forecasting of the input data file or DataFrame. See the Parameters table for parameter definitions.

`Multiview()` returns a `dict`:

```
dict { View        : DataFrame,
       Predictions : DataFrame,
     [ parameters  : dict ]
}
```

The `Predictions` `DataFrame` has 3 columns "Time", "Observations", "Predictions". nan values are inserted where there is no observation or prediction. If `predictFile` is provided the `Predictions` will be written to it in csv format.

The `View` `DataFrame` will have `2*D+3` columns. The first `D` columns are the the column indices in the input data `DataFrame` that are embedded and applied to Simplex prediction. The following three columns are "rho", "MAE", "RMSE" corresponding to the prediction Pearson correlation, maximum absolute error and root mean square error. The final `D` columns are the column names of the input embedding.

Parameters

`D` represents the number of variables to combine for each assessment, if not specified, it is the number of columns.

`E` is the embedding dimension of each variable. If `E = 1`, no time delay embedding is done.

`multiview` is the number of top-ranked D-dimensional predictions to "average" for the final prediction. Corresponds to parameter k in Ye & Sugihara with default k = sqrt(C) where C is the number of combinations C(n,D) available from the embedding columns taken D at-a-time.

`trainLib` specifies whether projections used to rank the column combinations are done in-sample (pred = lib, the default), or, using the lib and pred specified as input options (`trainLib false).`

`lib` and `pred` specify [start, stop] row indices of the input data for the library and predictions.

If `knn` is not specified, it is set equal to `D+1`.

If `columns` is a string and column names have whitespace, delimit the columns with “,” or, place the column names in a list.

If `parameterList = True`, a dictionary with `parameters` is added to the returned object.

`numThreads` defines the number of worker threads.

If `noTime = False`, the first column of the input DataFrame or .csv file must be an index or time column. If `noTime = True` an index or time column is not required.

```
//------------------------------------------------------------------
//
//------------------------------------------------------------------
dict Multiview( pathIn          = "./",
                dataFile        = "",
                dataFrame       = None,
                pathOut         = "./",
                predictFile     = "",
                lib             = "",
                pred            = "",
                D               = 0,
                E               = 1,
                Tp              = 1,
                knn             = 0,
                tau             = -1,
                columns         = "",
                target          = "",
                multiview       = 0,
                exclusionRadius = 0,
                trainLib        = True,
                excludeTarget   = False,
                parameterList   = False,
                verbose         = False,
                numThreads      = 4,
                showPlot        = False,
                noTime          = True )
```

EmbedDimension

Evaluate Simplex prediction skill for embedding dimensions from 1 to 10. The returned `DataFrame` has columns "E" and "rho". See the Parameters table for parameter definitions.

Note: `numThreads` defines the number of worker threads for the 10 embeddings. The maximum number of threads is 10.

If `noTime = False`, the first column of the input DataFrame or .csv file must be an index or time column. If `noTime = True` an index or time column is not required.

If `columns` is a string and column names have whitespace, delimit the columns with "," or, place the column names in a list.

```
//----------------------------------------------------------------
//
//----------------------------------------------------------------
DataFrame EmbedDimension( pathIn          = "./",
                          dataFile        = "",
                          dataFrame       = None,
                          pathOut         = "./",
                          predictFile     = "",
                          lib             = "",
                          pred            = "",
                          maxE            = 10,
                          Tp              = 1,
                          tau             = -1,
                          exclusionRadius = 0,
                          columns         = "",
                          target          = "",
                          embedded        = False,
                          verbose         = False,
                          validLib        = [],
                          numThreads      = 4,
                          showPlot        = True,
                          noTime          = False )
```

PredictInterval

Evaluate Simplex prediction skill for forecast intervals from 1 to 10. The returned `DataFrame` has columns "Tp" and "rho". See the Parameters table for parameter definitions.

Note: `numThreads` defines the number of worker threads for the 10 prediction interval forecasts. The maximum number of threads is 10.

If `noTime = False`, the first column of the input DataFrame or .csv file must be an index or time column. If `noTime = True` an index or time column is not required.

If `columns` is a string and column names have whitespace, delimit the columns with "," or, place the column names in a list.

```
//---------------------------------------------------------------------
// Overload 1: Explicit data file path/name
//---------------------------------------------------------------------
DataFrame PredictInterval( pathIn          = "./",
                           dataFile        = "",
                           dataFrame       = None,
                           pathOut         = "./",
                           predictFile     = "",
                           lib             = "",
                           pred            = "",
                           maxTp           = 10,
                           E               = 0,
                           tau             = -1,
                           exclusionRadius = 0,
                           columns         = "",
                           target          = "",
                           embedded        = False,
                           verbose         = False,
                           validLib        = [],
                           numThreads      = 4,
                           showPlot        = True,
                           noTime          = False );
```

PredictNonlinear

Evaluate SMap prediction skill for localisation parameter θ (default from 0.01 to 9). The returned `DataFrame` has columns "theta" and "rho". See the Parameters table for parameter definitions.

If `knn` is not specified, it is set equal to the library size. If `knn` is specified, it must be greater than `E`.

Note: `numThreads` defines the number of worker threads for the θ value forecasts.

If `noTime = False`, the first column of the input DataFrame or .csv file must be an index or time column. If `noTime = True` an index or time column is not required.

If `columns` is a string and column names have whitespace, delimit the columns with "," or, place the column names in a list.

```
//---------------------------------------------------------------
//
//---------------------------------------------------------------
DataFrame PredictNonlinear( pathIn          = "./",
                            dataFile        = "",
                            dataFrame       = None,
                            pathOut         = "./",
                            predictFile     = "",
                            lib             = "",
                            pred            = "",
                            theta           = "",
                            E               = 0,
                            knn             = 0,
                            Tp              = 1,
                            tau             = -1,
                            exclusionRadius = 0,
                            columns         = "",
                            target          = "",
                            embedded        = False,
                            verbose         = False,
                            validLib        = [],
                            ignoreNan       = true,
                            numThreads      = 4,
                            showPlot        = True,
                            noTime          = False );
```

ComputeError

Compute Pearson correlation coefficient, maximum absolute error (MAE) and root mean square error (RMSE) between two vectors.

`ComputeError()` returns a dict:

```
dict { rho  :  double,
       RMSE :  double,
       MAE  :  double
}

//----------------------------------------------------------------
//----------------------------------------------------------------
dict ComputeError( obsIn, predIn )
```

SurrogateData

Generate surrogate data using one of three methods.

1) random_shuffle :

Sample the data with a uniform distribution.

2) ebisuzaki :

Journal of Climate. A Method to Estimate the Statistical Significance of a Correlation When the Data Are Serially Correlated.
https://doi.org/10.1175/1520-0442(1997)010<2147:AMTETS>2.0.CO;2

Presumes data are serially correlated with low pass coherence. It is: "resampling in the frequency domain. This procedure will not preserve the distribution of values but rather the power spectrum (periodogram). The advantage of preserving the power spectrum is that resampled series retains the same autocorrelation as the original series."

3) seasonal :

Presume a smoothing spline represents the seasonal trend. The smooth parameter can range from 0 to 1. See scipy.interpolate.UnivariateSpline parameter `s`.

Each surrogate is a summation of the trend, resampled residuals, and possibly additive Gaussian noise. Default noise has a standard deviation (alpha) that is the data range / 5.

Note: It is presumed the first column of the dataFrame is a time vector. It is set as the first column of the returned DataFrame.

```
//------------------------------------------------------------------
//
//------------------------------------------------------------------
DataFrame SurrogateData( dataFrame     = None,
                         column        = None,
                         method        = 'ebisuzaki',
                         numSurrogates = 10,
                         alpha         = None,
                         smooth        = 0.8,
                         outputFile    = None )
```

Application Notes

All data input files are assumed to be in .csv format, or Pandas DataFrame.

> **The files are required to have a single line header with column names.**
> **It is expected the first column be a vector of times or time indices.** This can be disabled by setting the parameter `noTime = True`.

`SMap()` should be called with `DataFrame` that have columns explicity corresponding to dimensions E. This means that if a multivariate data set is used, it should Not be called with an embedding from `Embed()` since `Embed()` will add lagged coordinates for each variable. These extra columns will then not correspond to the intended dimensions in the matrix inversion and prediction reconstruction. In this case, use the `embedded` parameter set to `true` so that the columns selected correspond to the proper dimension.

Examples

```
    from pyEDM import *

    df = EmbedDimension( dataFrame = sampleData["TentMap"],
                         lib = "1 100", pred = "201 500",
                         columns = "TentMap", showPlot = True )

    df = PredictInterval( dataFrame = sampleData["TentMap"],
                          lib = "1 100", pred = "201 500",  E = 2,
                          columns = "TentMap", showPlot = True )

    df = PredictNonlinear( dataFrame = sampleData["TentMapNoise"],
                           lib = "1 100", pred = "201 500", E = 2,
                           columns = "TentMap", showPlot = True )

    df = Simplex( dataFrame = sampleData["block_3sp"],
                  lib = "1 99", pred = "100 198", E = 3,
                  columns = "x_t y_t z_t", target = "x_t",
                  embedded = True, showPlot = True )

    df = Simplex( dataFrame = sampleData["block_3sp"],
                  lib = "1 99", pred = "100 195", E = 3,
                  columns = "x_t", target = "x_t", showPlot = True )

    M = Multiview( dataFrame = sampleData["block_3sp"],
                   lib = "1 100", pred = "101 198", E = 3,
                   columns = "x_t y_t z_t", target = "x_t", showPlot = True )

    S = SMap( dataFrame = sampleData["circle"],
              lib = "1 100", pred = "101 198", E = 2, theta = 4,
              columns = "x y", target = "x", embedded = True, showPlot = True )

    df = CCM( dataFrame = sampleData["sardine_anchovy_sst"],
              E = 3, Tp = 0, columns = "anchovy", target = "np_sst",
              libSizes = "5 75 5", sample = 100, showPlot = True )
```

References

Dixon, P. A., M. Milicich, and G. Sugihara, 1999. Episodic fluctuations in larval supply. Science 283:1528–1530.

Sugihara G. and May R. 1990. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. Nature, 344:734–741.

Sugihara G. 1994. Nonlinear forecasting for the classification of natural time series. Philosophical Transactions: Physical Sciences and Engineering, 348 (1688) : 477–495.

Sugihara G., May R., Ye H., Hsieh C., Deyle E., Fogarty M., Munch S., 2012. Detecting Causality in Complex Ecosystems. Science 338:496-500.

Takens, F. Detecting strange attractors in turbulence. Lect. Notes Math. 898, 366–381 (1981).

Ye H., and G. Sugihara, 2016. Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. Science 353:922–925.

rEDM User Guide

Empirical Dynamic Modeling

George Sugihara Joseph Park Ethan Deyle Erik Saberski
Cameron Smith Hao Ye

2023-10-19

Abstract

Empirical dynamic modeling (EDM) is an emerging non-parametric framework for modeling nonlinear dynamic systems. EDM is based on the mathematical theory of reconstructing attractor manifolds from time series data (Takens 1981). The **rEDM** package collects several EDM methods, including simplex projection (Sugihara and May 1990), S-map (Sugihara 1994), multivariate embeddings (Dixon, Milicich, and Sugihara 1999), convergent cross mapping (Sugihara et al. 2012), and multiview embedding (Ye and Sugihara 2016). Here, we introduce the basic underlying theory, and describe the functionality of **rEDM** using examples from both model simulations and real data.

Installation

The **rEDM** package can be obtained in two main ways. The standard version of the package can be obtained through CRAN (the Comprehensive R Archive Network): https://cran.r-project.org/package=rEDM:

```
install.packages("rEDM")
```

Also available on GitHub at SugiharaLab, and can be installed using R **devtools**.

```
devtools::install_github("SugiharaLab/rEDM")
```

Introduction

Many scientific fields use models as approximations of reality and for various purposes, for example, testing hypotheses regarding mechanisms or processes, explaining past observations, and predicting future outcomes. In many cases these models are based on hypothesized parametric equations; however, explicit equations can be impractical when the underlying mechanisms are unknown or are too complex to be characterized with existing datasets. Empirical models, which infer patterns and associations from the data (instead of using hypothesized equations), represent an alternative and highly flexible approach. Here, we review the theoretical background for empirical dynamic modeling (EDM) and the functionality of the **rEDM** package, which are intended for nonlinear dynamic systems that can prove problematic for traditional modeling approaches.

The basic goal underlying EDM is to reconstruct the behavior of dynamic systems using time series data. This approach is based on mathematical theory developed initially by (Takens 1981), and expanded by others (Sauer, Yorke, and Casdagli 1991; Casdagli et al. 1991; Deyle and Sugihara 2011). Because these methods operate with minimal assumptions, they are particularly suitable for studying systems that exhibit non-equilibrium dynamics and nonlinear state-dependent behavior (i.e. where interactions change over time and as a function of the system state).

Empirical Dynamic Modeling

Time Series as Observations of a Dynamic System

The essential concept is that time series can be viewed as projections of the behavior of a dynamic system. First, the system state can be described as a point in a high-dimensional space. The axes of this space can be thought of as fundamental state variables; in an ecosystem, these variables might correspond to population abundances, resources, or environmental conditions. Second, the system state changes through time following a set of deterministic rules. In other words, the behavior of the system is not completely stochastic.

Consequently, it is possible to project the system state onto one of the coordinate axes and obtain the value of the corresponding state variable. Sequential projections over time will thus produce a time series for that variable. For example, in figure 1 the states of the canonical Lorenz Attractor (Lorenz 1963) are projected onto the x-axis, creating a time series of variable x.

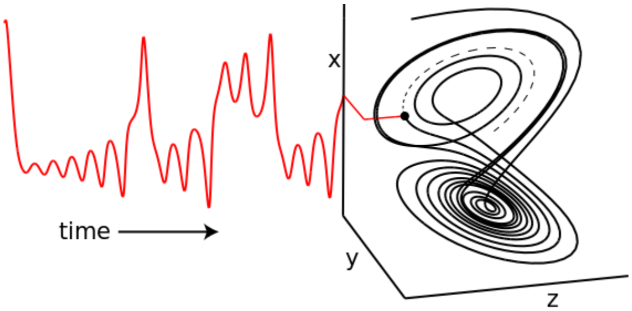


Figure 1: Time Series Projection from the Lorenz Attractor.

Although different time series observed from a system can represent independent state variables, in general, each time series is an *observation function* of the system state that may convolve several different state variables.

Attractor Reconstruction / Takens' Theorem

The goal of EDM is to reconstruct the system dynamics from time series data. As seen above, a time series can be thought of as sequential projections of the motion on an attractor; in other words, information about the behavior is encoded in the temporal ordering of the time series. Takens' Theorem (Takens 1981) states that mathematically valid and property preserving reconstructions of the attractor can be created using lags of a single time series, then substituting those lagged time series for unknown or unobserved variables. In other words, instead of representing the system state using a complete set of state variables, we can instead use an `E`-dimensional lagged-coordinate embedding:

$$\vec{x}_t = \langle x_t, x_{t-\tau}, \dots, x_{t-(E-1)\tau} \rangle$$

If sufficient lags are used, the reconstruction preserves essential mathematical properties of the original system: reconstructed states will map one-to-one to actual system states, and nearby points in the reconstruction will correspond to similar system states. Figure 2 shows a reconstruction of the Lorenz attractor where the reconstructed system state is comprised of 3 lags of variable x. Here, the visual similarity between the reconstruction and the original Lorenz Attractor is quite clear.

As a consequence of the fact that dynamical properties of the original system can be recovered from a single time series, there are multiple applications. For example, empirical models can be used for forecasting (Sugihara and May 1990), to understand nonlinear behavior (Sugihara 1994), or to uncover mechanism (Dixon, Milicich, and Sugihara 1999). Moreover, recent work describes how EDM can be used to identify

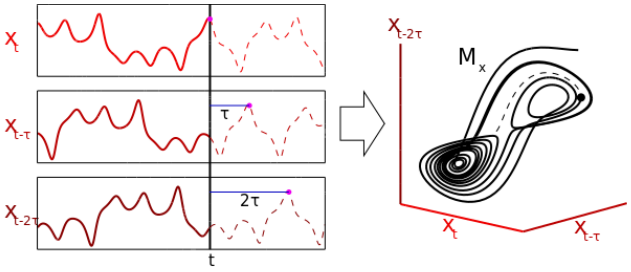


Figure 2: Attractor Reconstruction from 3 Lagged Coordinates

causal interactions, by testing whether two time series are observed from the same system (Sugihara et al. 2012). In the next section, we demonstrate how **rEDM** can be used to accomplish these various tasks.

Demonstration of EDM

Nearest Neighbor Forecasting using Simplex Projection

As mentioned previously, the reconstruction will map one-to-one to the original attractor manifold if enough lags are used (i.e. if the reconstruction has a sufficiently large embedding dimension). If the embedding dimension is too small, then reconstructed states can overlap and appear to be the same even though they actually correspond to different states. These "singularities" will result in poor forecast performance because the system behavior cannot be uniquely determined in the reconstruction. As a consequence, we can use prediction skill as an indicator for identifying the optimal embedding dimension. In the following example we demonstrate the `Simplex()` projection nearest neighbor forecasting method (Sugihara and May 1990), and its' extension `EmbedDimension()` that automates evaluation of an optimal embedding dimension.

Example

In this example, time series come from a simulation of the tent map that exhibits chaotic behavior. The tent map is a discrete-time dynamic system, where a sequence, x_t, on the interval $[0, 1]$ is iterated according to:

$$x_{t+1} = \begin{cases} 2x_t & x_t < \frac{1}{2} \\ 2(1 - x_t) & x_t \geq \frac{1}{2} \end{cases}$$

In **rEDM**, a sample time series of first-differenced values can be found in dataset `TentMap`.

We begin by loading the **rEDM** package and examining the `TentMap` data:

```
library(rEDM)
str(TentMap)
```

```
## 'data.frame':    999 obs. of  2 variables:
##  $ Time   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ TentMap: num  -0.0992 -0.6013 0.7998 -0.7944 0.798 ...
```

We can see that the data consists of a data.frame with two columns: `Time` and `TentMap`. All rEDM input data files or data.frames are assumed to have a time vector in the first column. Data files are expected to be in .csv format with the first line a header of column names, data.frames are also expected to have column names.

The `Simplex` function has 5 required parameters:

1.	`columns`	`TentMap`	name of column(s) of embedding library

2.	`target`	`TentMap`	name of column for prediction
3.	`lib`	`"1 100"`	start stop indices of embedding library
4.	`pred`	`"201 500"`	start stop indices of predictions
5.	`E`	`3`	embedding dimension

`columns` specifies the timeseries vector(s) that form the library, `target` is the column on which predictions will be made. `lib` defines row indices of the "training" portion of data, `pred` corresponds to row indices of the "test" portion, and `E` defines the embedding dimension.

Note that if any overlap in the lib and pred is found, it will enable leave-one-out cross-validation. If `verbose = TRUE`, a warning message will be raised.

In this univariate case, we specify the "TentMap" column of the data frame for both `columns` and `target`, and select the first 100 points (indices 1 to 100) in the time series to constitute the "library set", and a separate 300 point span (indices 201 to 500) as the "prediction set".

Default parameters of `knn` (k-nearest neighbors) and `Tp` (time-to-prediction) are assumed. The default `knn = 0` sets the number of nearest neighbors to `E + 1`, and default `Tp` is 1 timestep (observation row). With these parameters we demonstrate the `Simplex()` function:

```r
simplex_out <- Simplex(dataFrame = TentMap, lib = "1 100", pred = "201 500", columns = "TentMap",
    target = "TentMap", E = 3)
simplex_out[c(1:2, 300:301), ]
```

```
##     Time Observations Predictions Pred_Variance
## 1    201         0.94         NaN           NaN
## 2    202         0.11       0.077       0.00015
## 300  500        -1.09      -1.084       0.06366
## 301  501         0.91       0.873       0.00255
```

Note that the returned data.frame has 1 `NaN` as the first `Predictions` point since `Tp = 1`, and, the last `Observations` will likewise be `NaN` with the time vector adjusted to accommodate `Tp` rows beyond the data as needed.

Computation of Pearson correlation, MAE and RMSE errors between the forecast `Observations` and `Predictions` can be performed with the `ComputeError()` function.

```r
ComputeError(simplex_out$Observations, simplex_out$Predictions)
```

```
## $MAE
## [1] 0.14
##
## $rho
## [1] 0.94
##
## $RMSE
## [1] 0.23
```

Optimal embedding dimension

As noted earlier, identification of the optimal embedding dimension to best "unfold" the dynamics can be assessed with simplex prediction skill. **rEDM** provides the `EmbedDimension()` function to automate this task. `EmbedDimension()` parallelises function calls to `Simplex()`, which automatically sets values of `E` from 1 to `maxE`=10. Continuing with the previous example, we invoke `EmbedDimension()`:

```r
rho_E <- EmbedDimension(dataFrame = TentMap, lib = "1 100", pred = "201 500", columns = "TentMap",
    target = "TentMap")
```

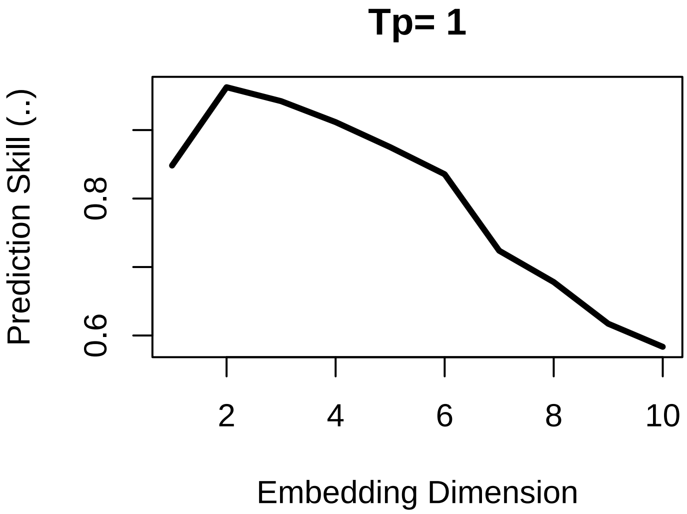


Figure 3: TentMap data prediction skill vs. embedding dimension.

The output is a data.frame with columns `E` and `rho` detailing the embedding dimension and Pearson correlation coefficient between the simplex projected forecast at `Tp = 1` timesteps ahead, and the observed data over the `pred` indices. Here, we observe that forecast skill peaks at `E = 2`, indicating that the dynamics of our data are unfolded best in 2 dimensions. *Note that this optimal value does not have to correspond to the dimensionality of the original system.* The forecast skill will be affected by factors such as observational noise, process error, and time series length, and so it is more useful to think of the embedding dimension as a practical measure that is dependent on properties of the data.

Prediction Decay

An important property of many natural systems is that nearby trajectories eventually diverge over time (i.e. “deterministic chaos” – the “butterfly effect”). In essence, this means that while short-term prediction is often possible, information about the predictive state of the system is diluted over time, hindering long-term forecasting. We can demonstrate this effect by examining how prediction skill changes as we increase the `Tp` argument, the “time to prediction”, defining the number of time steps into the future at which forecasts are made. **rEDM** provides the `PredictInterval()` function to automate this task.

Example

Using the same data with the `PredictInterval()` function, we supply the embedding dimension parameter with the value determined previously (`E = 2`):

```
rho_Tp <- PredictInterval(dataFrame = TentMap, lib = "1 100", pred = "201 500", target = "TentMap",
    columns = "TentMap", E = 2)
```

As above, the returned object is a data.frame with forecast skill `rho` and time to prediction `Tp`. As expected (because the parameters chosen for the tent map fall in the region for chaotic behavior), the decline in forecast skill (`rho` → 0) as the forecast interval `Tp` increases, indicates that the system may be chaotic.

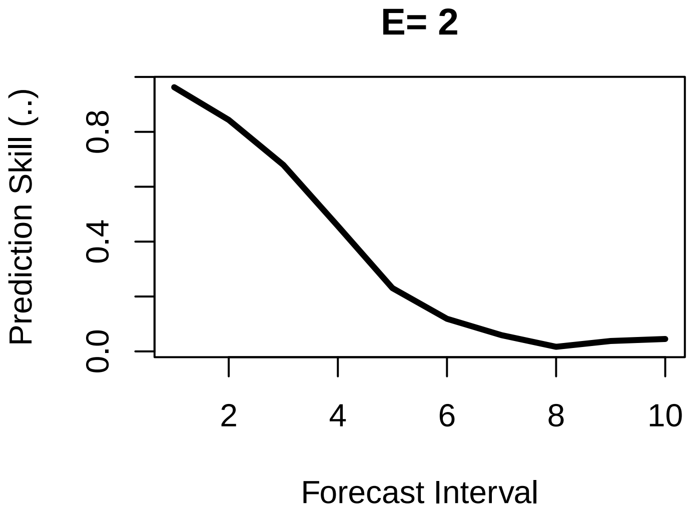


Figure 4: Tent map first differences simplex prediction skill as a function of forecast interval.

Identifying Nonlinearity

One concern is that time series may show predictability even if they are purely stochastic, they behave similarly to autocorrelated red noise. Fortunately, we can distinguish between red noise and nonlinear deterministic behavior by using S-maps as described in (Sugihara 1994).

In contrast to the nearest-neighbor interpolation of simplex projection, the S-map forecasting method (Sugihara 1994) fits local linear maps to describe the dynamics. In addition to the standard set of parameters for a lagged-coordinate reconstruction as in simplex, S-maps contain a nonlinear localisation parameter, θ, that determines the degree to which points are weighted when fitting the local linear map. For example, when $\theta = 0$, all points are equally weighted, such that the local linear map is identical for different points in the reconstructed state-space. As such, the S-map will be identical to a global linear map (i.e. an autoregressive model). When values of θ are greater than 0, nearby points in the state space receive larger weight, and the local linear map can vary in state-space to accommodate nonlinear behavior.

Consequently, if the time series are sampled from autoregressive red noise, then the linear model ($\theta = 0$) should produce better forecasts, because the global linear map (which will, in effect, be fitted to more data points) will reduce the effects of observation error compared to local linear maps. In contrast, if forecast skill increases for $\theta > 0$, then the results are suggestive of nonlinear dynamics wherein better forecasts are achieved when the local linear map can change depending on the location in state-space: it is a better description of state-dependent behavior.

Example

The `PredictNonlinear()` function provides an evaluation of S-map forecast skill as a function of the localisation parameter `theta`. If unspecified, `theta` values will range from 0.01 to 9.

Typically, when using `S-map` to test for nonlinear behavior, we want to use all available points in the reconstruction of the local linear map, not just `knn` nearest neighbors as in simplex projection. With all points available, S-map uses the `theta` parameter to control the weighting assigned to individual points, thereby localising the dynamics to capture nonlinear behavior. When `knn = 0`, the default, `SMap()` will use all available points.

Here we use an embedding dimension of `E = 2` and the same parameters as in the previous examples, however, we specify the “TentMapNoise” data that adds Gaussian noise to the TentMap data as one would normally encounter with noisy observational data.

```
rho_theta <- PredictNonlinear(dataFrame = TentMapNoise, lib = "1 100", pred = "201 500",
    target = "TentMap", columns = "TentMap", E = 2)
```

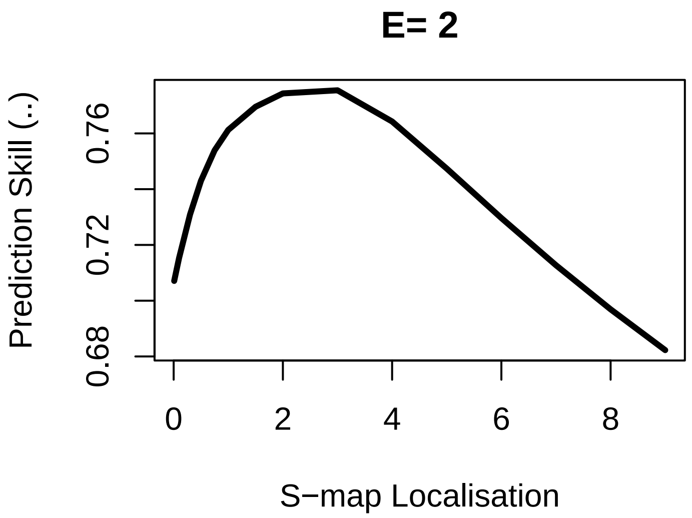


Figure 5: Tent map first differences S-map prediction skill as a function of S-map localisation parameter.

The result is a data.frame with columns `Theta` and `rho`. Here, we see that forecast skill substantially improves as `theta` increases, indicating the presence of nonlinear dynamics. We also observe a degradation in forecast skill at high values of `theta` as the local linear map overfits to insufficient nearest neighbors.

`Simplex()` and `SMap()` functions

The functions `EmbedDimension()`, `PredictInterval()` and `PredictNonlinear()` are multithreaded wrapper functions for the `Simplex()` and `SMap()` algorithms. `EmbedDimension()` and `PredictInterval()` parallelise calls to `Simplex()` to evaluate forecast skill as a function of embedding dimension and prediction interval respectively. `PredictNonlinear()` parallelises calls to `SMap()` to assess predictive skill as a function of the nearest neighbor localisation parameter. However, one can equivalently call the underlying `Simplex()` and `SMap()` functions directly.

`Simplex()`

For example, evaluation of the simplex prediction at an optimal embedding dimension of `E = 2` can be performed as:

```
tentMapPredict <- Simplex(dataFrame = TentMap, lib = "1 100", pred = "201 500", target = "TentMap",
    columns = "TentMap", E = 2)

ComputeError(tentMapPredict$Observations, tentMapPredict$Predictions)$rho
```

```
## [1] 0.96
```

`SMap()`

An individual S-map evaluation corresponding to the optimal `PredictNonlinear()` result from above is:

```
smap = SMap(dataFrame = TentMapNoise, lib = "1 100", pred = "201 500", target = "TentMap",
    columns = "TentMap", E = 2, theta = 3)
```

`SMap()` returns a named list with `predictions` and `coefficients` data.frames, with `NaN` inserted appropriately where no predictions or observations are available.

```
head(cbind(smap$predictions, smap$coefficients), 2)
```

```
##   Time Observations Predictions Pred_Variance Time   C0 TentMap/ TentMap(t-0)
## 1  201         0.99         NaN           NaN  201  NaN                  NaN
## 2  202         0.16       -0.14          0.28  202 -1.1                -0.33
##   TentMap/ TentMap(t-1)
## 1                  NaN
## 2                 -1.1
```

```
tail(cbind(smap$predictions, smap$coefficients), 2)
```

```
##     Time Observations Predictions Pred_Variance Time    C0
## 300  500         -1.3       -0.63          0.36  500 -0.33
## 301  501          1.1        0.92          0.41  501  0.24
##     TentMap/ TentMap(t-0) TentMap/ TentMap(t-1)
## 300                 -0.78                 -0.16
## 301                 -0.52                 -0.03
```

Generalized Takens Theorem

A practical reality is that sampled observations of complex dynamics are usually composed of finite, noisy data. Additionally, the presence of stochastic, non-deterministic drivers means that "multivariate" reconstructions can often be a better description than "univariate" reconstructions. This means that in addition to creating an attractor from lags of one time series, it can be advantageous to combine different time series to create the phase-space embedding, provided they are all observed from the same system (Sauer, Yorke, and Casdagli 1991; Deyle and Sugihara 2011).

In **rEDM**, the `Simplex()` and `SMap()` functions allow multivariate reconstructions from any set of observation vectors. A multivariate reconstruction is defined by specifying which columns to use as coordinates in the `columns` argument, and which column is to be forecast in the `target` argument. By default, **rEDM** will create Takens time-delay embeddings from univariate or multivariate data, however, this can be prevented by setting the `embedded` parameter `TRUE`. In this case, the input data are assumed to already constitute a valid multidimensional embedding, and no time-delay embedding is performed.

Example

We begin by examining an example dataset from a coupled 3-species model system.

```
head(block_3sp, 3)
```

```
##   time   x_t x_t-1 x_t-2   y_t y_t-1 y_t-2  z_t z_t-1 z_t-2
## 1    3 -1.92  1.24 -0.74 -0.11  1.49 -1.27  1.5 -0.48 -1.86
## 2    4 -0.96 -1.92  1.24 -1.11 -0.11  1.49 -1.5  1.54 -0.48
## 3    5  1.33 -0.96 -1.92  2.39 -1.11 -0.11 -1.1 -1.49  1.54
```

Here, `block_3sp` is a 10-column data.frame with 9 data columns. This data has already been time-delay embedded to dimension `E = 3` with a time delay of `tau = -1`. We use simplex forecasting based on a multivariate embedding of the three data vectors `x_t x_t-1 z_t` with `embedded = TRUE` and `E = 3`:

```r
smplx_3species = Simplex(dataFrame = block_3sp, lib = "1 100", pred = "101 190",
    E = 3, columns = "x_t x_t-1 z_t", target = "x_t", embedded = TRUE)
```

A plot of the predictions vs. observations can be examined with:

```r
err = ComputeError(smplx_3species$Observations, smplx_3species$Predictions)
plot(smplx_3species$Observations, smplx_3species$Predictions, pch = 19, cex = 0.5,
    xlab = "Observations", ylab = "Predictions", main = "3 Species x_t")
abline(a = 0, b = 1, lty = 2, col = "blue")
text(-1, 1, paste(capture.output(cbind(err)), collapse = "\n"))
```

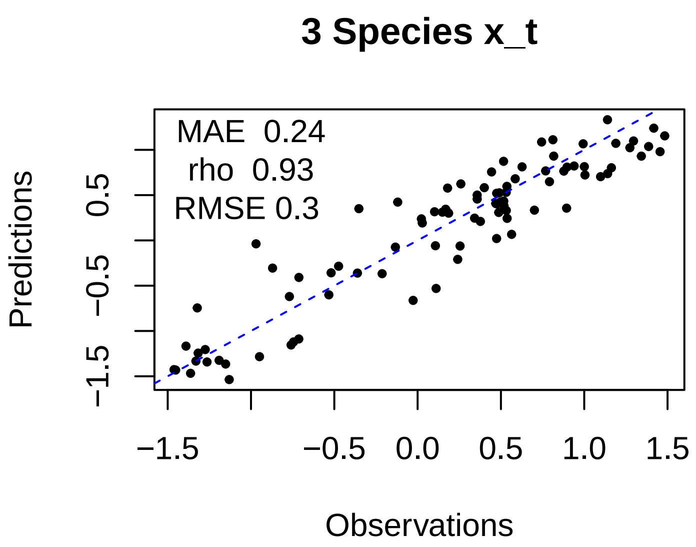


Figure 6: Scatter plot of simplex forecast of x_t vs. observations.

S-map Coefficients

As described in (Deyle et al. 2016), S-map coefficients from the appropriate multivariate embedding can be interpreted as dynamic, time-varying interaction strengths. We demonstrate this with a chaotic timeseries described in (Lorenz 1996), defined for N variables k=1, … N, as

$$\frac{dx_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - X_k + F$$

The `Lorenz5D` data.frame contains a N=5 dimensional system with F=8 from (Lorenz 1996). Here, we use `SMap()` to compute a 4-dimensional forecast at Tp=1:

```r
smap_Lorenz <- SMap(dataFrame = Lorenz5D, lib = "1 500", pred = "601 900", E = 4,
    theta = 3, columns = "V1 V2 V3 V4", target = "V1", embedded = TRUE)
```

As noted earlier, `SMap()` returns a named list with two data frames, `predictions` and `coefficients`:

```r
head(cbind(smap_Lorenz$predictions, smap_Lorenz$coefficients[, 2:6]), 3)
```

```
##    Time Observations Predictions Pred_Variance      C0  V1/ V1  V1/ V2
## 1 40.00        3.485         NaN           NaN     NaN     NaN     NaN
## 2 40.05        4.214       4.123         8.197 -0.4830  0.9914  0.1313
## 3 40.10        4.849       4.744         8.947 -0.5544  0.9890  0.1530
##      V1/ V3  V1/ V4
## 1        NaN     NaN
## 2 -0.011692 0.04222
## 3 -0.006055 0.02284
```

Here, we plot the time series for the observed (blue) and predicted (red) values of `V1` in the top panel; and the inferred interactions (S-map coefficients) for the influence of `V4`, `V3` and `V2` on future values of `V1` in the lower panels.

```
predictions = smap_Lorenz$predictions
coefficients = smap_Lorenz$coefficients
Time = predictions$Time

plot(Time, predictions$Observations, type = "l", col = "blue", ylab = "V1", xlab = "",
    lwd = 2, cex.lab = 1.3, cex.axis = 1.3)
lines(Time, predictions$Predictions, lwd = 2, col = "red")
legend("topright", legend = c("observed", "predicted"), fill = c("blue", "red"),
    bty = "n", cex = 1.3)

plot(Time, coefficients[, 6], type = "l", col = "brown", ylab = paste(" ", "V4/",
    " ", "V1", sep = ""), xlab = "", lwd = 2, cex.lab = 1.3, cex.axis = 1.3)
plot(Time, coefficients[, 5], type = "l", col = "darkgreen", ylab = paste(" ",
    "V3/", " ", "V1", sep = ""), xlab = "", lwd = 2, cex.lab = 1.3, cex.axis = 1.3)
plot(Time, coefficients[, 4], type = "l", col = "blue", ylab = paste(" ", "V2/",
    " ", "V1", sep = ""), xlab = "", lwd = 2, cex.lab = 1.3, cex.axis = 1.3)
```

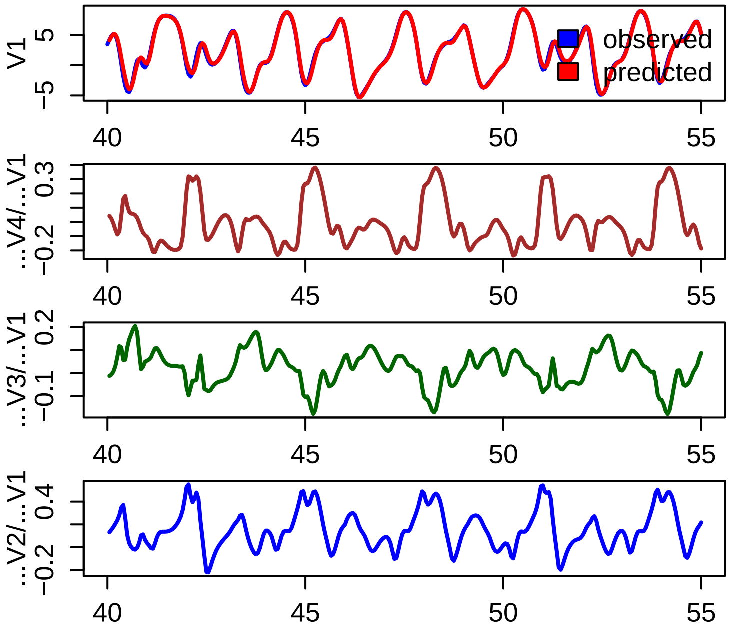


Figure 7: S-map prediction and coefficients of Lorenz'96 5-D system.

Multiview Embedding

The generality of Takens Theorem means that in situations with multivariate time series, there can often be many different, valid attractor reconstructions. As described in (Ye and Sugihara 2016), combining these different models can result in improved forecasts.

Here, we demonstrate this idea using the `Multiview()` function with the 3-species data used above. `Multiview()` operates by constructing all possible embeddings of dimension `E` with lag up to `E-1`. These embeddings are ranked by forecast skill (`rho`) over the `lib` portion of the data. The individual forecasts for the top `multiview` embeddings are then averaged together. If `multiview` is not specified it is set to sqrt(C) where C is the number of E-dimensional combinations created from all data vectors.

```
Mview = Multiview(dataFrame = block_3sp, lib = "1 100", pred = "101 190", E = 3,
    columns = "x_t y_t z_t", target = "x_t")
```

`Multiview()` returns a named list with two data.frames: `View`, and `Predictions`. `View` lists the various combinations of data embedding vectors used for the forecasts along with their prediction statistics. `Predictions` returns the final averaged multiview projections.

```
Mview$View[which(Mview$View$rho > 0.91), ]
```

```
##   Col_1 Col_2 Col_3    rho    MAE   RMSE   name_1   name_2   name_3
## 1     1     2     7 0.9320 0.2391 0.2996 x_t(t-0) x_t(t-1) z_t(t-0)
## 3     1     2     3 0.9395 0.2221 0.2815 x_t(t-0) x_t(t-1) x_t(t-2)
## 7     1     2     8 0.9215 0.2484 0.3202 x_t(t-0) x_t(t-1) z_t(t-1)
```

Causality Inference and Cross Mapping

One of the corollaries to the Generalized Takens Theorem is that it should be possible to cross predict or cross map between variables that are observed from the same system. Consider two variables, x and y that interact in a dynamic system. Then the univariate reconstructions based on x or y alone should uniquely identify the system state and and thus the corresponding value of the other variable.

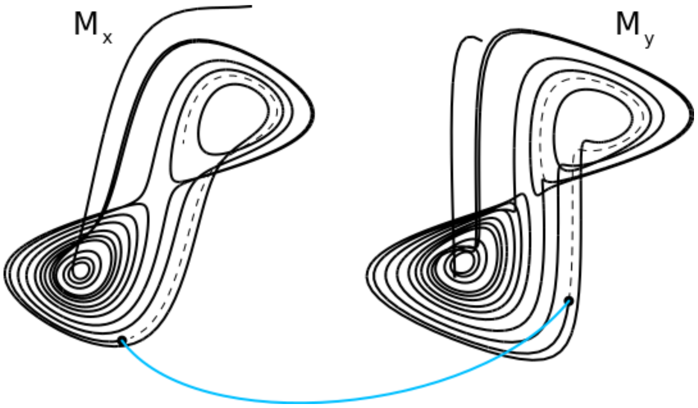


Figure 8: Cross Mapping Between Reconstructions of the Lorenz Attractor

In the case of unidirectional causality, e.g. x causes y, the causal variable (x) leaves a signature on the affected variable (y). Consequently, the reconstructed states based on y can be used to cross predict the values of x (because the reconstruction based on y must be complete, it must include information about the value of x). Note that this cross prediction is in the *opposite* direction of the causal effect. At the same time, cross prediction from x to y will fail, because the time series of x behaves independently of y, so a univariate reconstruction using only lags of x is necessarily incomplete.

Although x has incomplete information for predicting y, it does affect the values of y, and therefore will likely to have nonzero predictive skill. However, this cross mapping will be limited to the statistical association between x and y and will generally not improve as longer time series are used for reconstruction. In contrast, the cross prediction of x from y will generally improve. This convergence is therefore a crucial property for inferring causality. For practical reasons, the sensitivity of detecting causality this way is improved if, instead of predicting the future value of another variable, we estimate the concurrent value of another variable. We refer to this modified method as cross mapping, because we are not "predicting" the future.

For a more detailed description of using cross mapping to infer causation, see (Sugihara et al. 2012).

Convergent Cross Mapping (CCM)

In **rEDM**, convergent cross mapping is implemented as the `CCM()` function, which provides a wrapper to compute cross map skill for different subsamples of the data. In the following example, we reproduce the analysis from (Sugihara et al. 2012) to identify causality between anchovy landings in California and Newport Pier sea-surface temperature. For this example, a previously identified value of `3` for the embedding dimension will be used.

To quantify convergence, we compute the cross map skill over many random subsamples of the time series. The `libSizes` argument specifies the size of the library set, and `sample` specifies the number of subsamples generated at each library size. `random` and `replacement` specify how the subsamples will be generated. The default is random sampling without replacement.

```
cmap <- CCM(dataFrame = sardine_anchovy_sst, E = 3, Tp = 0, columns = "anchovy",
    target = "np_sst", libSizes = "10 70 5", sample = 100, showPlot = TRUE)
```

The output is a data.frame with statistics for each model run (in this case, 100 models at each library size) as a function of library size. Recalling that cross mapping indicates causal influence in the reverse

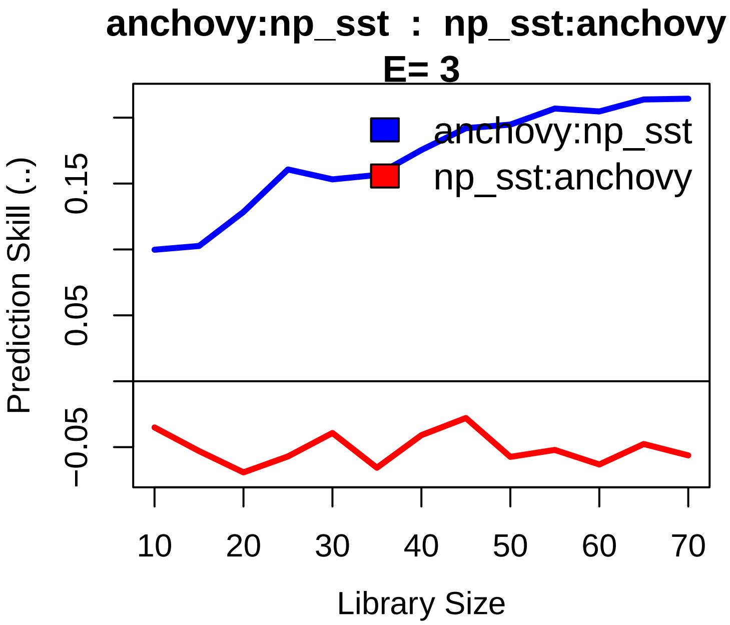


Figure 9: Convergent cross mapping of Newport sea surface temperature with anchovy landings.

direction (target to source), we see that the cross mapping `anchovy:np_sst` converges at a positive value of ρ, indicating that Newport sea surface temperature influences anchovy landings. Because average cross map skill less than 0 means there is no prediction skill, (predictions should not be anticorrelated with observations), we infer from the `np_sst:anchovy` cross mapping that anchovy landings do not effect sea surface temperatures.

Real Data Example

Apple-Blossom Thrips

In this example, we use EDM to re-examine the classic apple-blossom thrips (*Thrips imaginis*) time series from the Wait Institute in Australia (Davidson and Andrewartha 1948a, 1948b). Seasonal outbreaks of *Thrips imaginis* were observed to vary greatly in magnitude from year to year, but large outbreaks tended to coincide across large spatial domains. This lead to the hypothesis that regional-scale climatic factors were responsible for controlling the size of the seasonal outbreaks (what might now be called the Moran effect (Moran 1953)).

```
head(Thrips, 2)
```

```
##   Year Month Thrips_imaginis maxT_degC Rain_mm Season
## 1 1932     4             4.5      19.2   140.1 -0.500
## 2 1932     5            23.4      19.1    53.7 -0.866
```

The data column `Thrips_imaginis` contains counts of *Thrips imaginis* obtained from the Global Population Dynamics Database (GPDD) (NERC Centre for Population Biology 2010). `maxT_degC` is the mean maximum daily temperature (°C) taken over each month and `Rain_mm` is the monthly rainfall (mm), both from the Waite Institute. The final column `Season` is a simple annual sinusoid that peaks in December (the Austral summer) to emulate an indicator of season.

First, we plot the data. Note that all the time-series variables, particularly the mean maximum daily temperature, show marked seasonality.

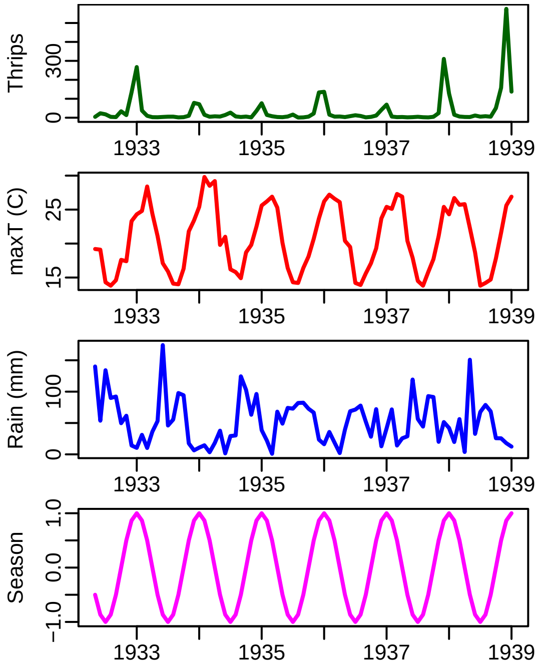


Figure 10: Thrips abundance and environmental variables.

Univariate Analysis

We first examine the dependence of simplex predictability on the embedding dimension.

```
rho_E <- EmbedDimension(dataFrame = Thrips, columns = "Thrips_imaginis", target = "Thrips_imaginis",
    lib = "1 72", pred = "1 72", showPlot = TRUE)
```

While there is an initial peak in the simplex prediction at `E = 3`, the global maximum is at `E = 8`. This suggests that both `E = 3` and `E = 8` are practical embedding dimensions, although `E = 8` is preferrable with a higher predictive skill.

To test for nonlinearity we use the S-map `PredictNonlinear()` function.

```
E = 8
rho_theta_e3 = PredictNonlinear(dataFrame = Thrips, columns = "Thrips_imaginis",
    target = "Thrips_imaginis", lib = "1 73", pred = "1 73", E = E)
```

The S-map results demonstrate clear nonlinearity in the Thrips time series, as nonlinear models `theta > 0` give substantially better predictions than the linear model `theta = 0`. This suggests that *Thrips*, despite the strong seasonal dynamics, do not simply track the environment passively, but have some intrinsic dynamics. To look more closely at the issue of seasonal drivers, however, we turn to convergent cross-mapping (CCM).

Seasonal Drivers

Recall that there is a two-part criterion for CCM to be a rigorous test of causality:

1. The cross map prediction skill is statistically significant when using the full time series as the library.
2. Cross map prediction demonstrates convergence, i.e. prediction skill increases as more of the time series is used for the library and the reconstructed attractor becomes more dense.

For an initial summary, we first compute the cross map skill (measured with Pearsons ρ) for each variable pair. Note that `CCM()` computes the cross map in both "directions".

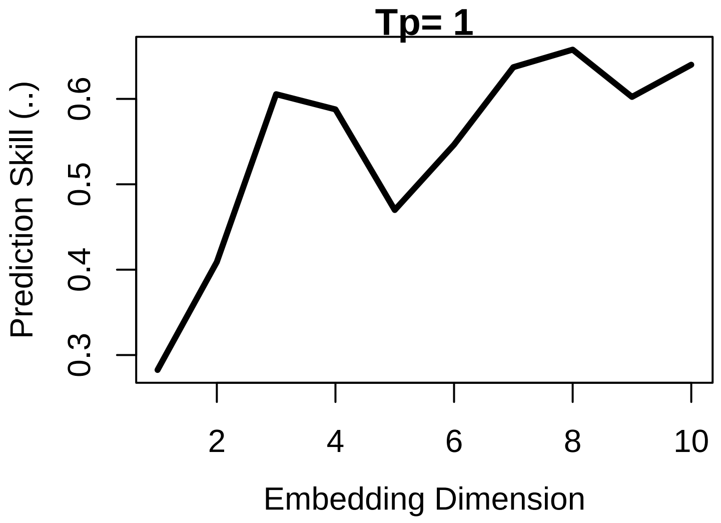


Figure 11: Simplex embedding dimension for Thrips abundance.

```
vars = colnames(Thrips[3:6])
var_pairs = combn(vars, 2)  # Combinations of vars, 2 at a time
libSize = paste(NROW(Thrips) - E, NROW(Thrips) - E, 10, collapse = " ")
ccm_matrix = array(NA, dim = c(length(vars), length(vars)), dimnames = list(vars,
    vars))

for (i in 1:ncol(var_pairs)) {
    ccm_out = CCM(dataFrame = Thrips, columns = var_pairs[1, i], target = var_pairs[2,
        i], libSizes = libSize, Tp = 0, E = E, sample = 100)

    outVars = names(ccm_out)

    var_out = unlist(strsplit(outVars[2], ":"))
    ccm_matrix[var_out[2], var_out[1]] = ccm_out[1, 2]

    var_out = unlist(strsplit(outVars[3], ":"))
    ccm_matrix[var_out[2], var_out[1]] = ccm_out[1, 3]
}
```

We note that ccm_matrix rows are the second of the CCM() returned variables, while columns are the first variable. As outlined earlier, influences are quantified from the target to the columns variable so that here, rows are considered the ’target, influencing variable, and columns thecolumns‘ influenced variable.

For comparison we also compute the lagged cross-correlation, allowing lags of up to ±6 months.

```
corr_matrix <- array(NA, dim = c(length(vars), length(vars)), dimnames = list(vars,
    vars))

for (ccm_from in vars) {
    for (ccm_to in vars[vars != ccm_from]) {
        ccf_out <- ccf(Thrips[, ccm_from], Thrips[, ccm_to], type = "correlation",
            lag.max = 6, plot = FALSE)$acf
        corr_matrix[ccm_from, ccm_to] <- max(abs(ccf_out))
```

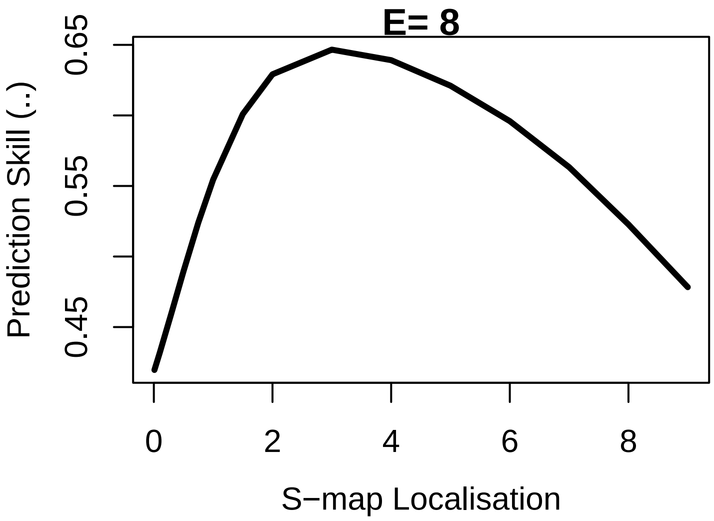


Figure 12: SMap localisation parameter for Thrips abundance.

```
    }
}
```

We compare the two matrices.

```
ccm_matrix
```

```
##                  Thrips_imaginis maxT_degC Rain_mm Season
## Thrips_imaginis               NA    0.6046  0.4254 0.5617
## maxT_degC                 0.9227        NA  0.8214 0.9625
## Rain_mm                   0.5118    0.4624      NA 0.3933
## Season                    0.9544    0.9918  0.7773     NA
```

```
corr_matrix
```

```
##                  Thrips_imaginis maxT_degC Rain_mm Season
## Thrips_imaginis               NA    0.4490  0.2668 0.4488
## maxT_degC                 0.4490        NA  0.5949 0.9453
## Rain_mm                   0.2668    0.5949      NA 0.5333
## Season                    0.4488    0.9453  0.5333     NA
```

We can see that the cross map strengths are not symmetric. In general CCM(X1 : X2) != CCM(X2 : X1). We also notice that the cross map and correlation between temperature and the seasonal indicator are high, with the cross map results suggesting that the seasonal variable can almost perfectly recover the temperature, $\rho = 0.9918$. This makes interpretation more complicated, because we have to consider the possibility that cross mapping is simply identifying the shared seasonality between two time series. In other words, cross mapping between temperature and any variable with a seasonal cycle, might suggest an interaction even if there is no actual causal mechanism.

Convergent Cross-Mapping With this in mind, we examine convergence in cross-map predictability, i.e. we compute `rho` as a function of library size `L`. The magnitude of the cross-correlation between *Thrips* and the cross mapped variable is shown as a black dashed line for comparison.

```
thrips_xmap_maxT <- CCM(dataFrame = Thrips, E = E, Tp = 0, columns = "Thrips_imaginis",
    target = "maxT_degC", libSizes = "13 73 3", sample = 300, showPlot = FALSE)
CCMPlot(thrips_xmap_maxT, E)
```

```
## [1] "Thrips_imaginis:maxT_degC  :  maxT_degC:Thrips_imaginis \nE= 8"
##  [1] 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73
```

```
abline(h = corr_matrix["Thrips_imaginis", "maxT_degC"], col = "black", lty = 2)

thrips_xmap_Rain <- CCM(dataFrame = Thrips, E = E, Tp = 0, columns = "Thrips_imaginis",
    target = "Rain_mm", libSizes = "13 73 3", sample = 300, showPlot = FALSE)
CCMPlot(thrips_xmap_Rain, E)
```

```
## [1] "Thrips_imaginis:Rain_mm  :  Rain_mm:Thrips_imaginis \nE= 8"
##  [1] 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73
```

```
abline(h = corr_matrix["Thrips_imaginis", "Rain_mm"], col = "black", lty = 2)

thrips_xmap_Season <- CCM(dataFrame = Thrips, E = E, Tp = 0, columns = "Thrips_imaginis",
    target = "Season", libSizes = "13 73 3", sample = 300, showPlot = FALSE)
CCMPlot(thrips_xmap_Season, E)
```

```
## [1] "Thrips_imaginis:Season  :  Season:Thrips_imaginis \nE= 8"
##  [1] 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73
```

```
abline(h = corr_matrix["Thrips_imaginis", "Season"], col = "black", lty = 2)
```

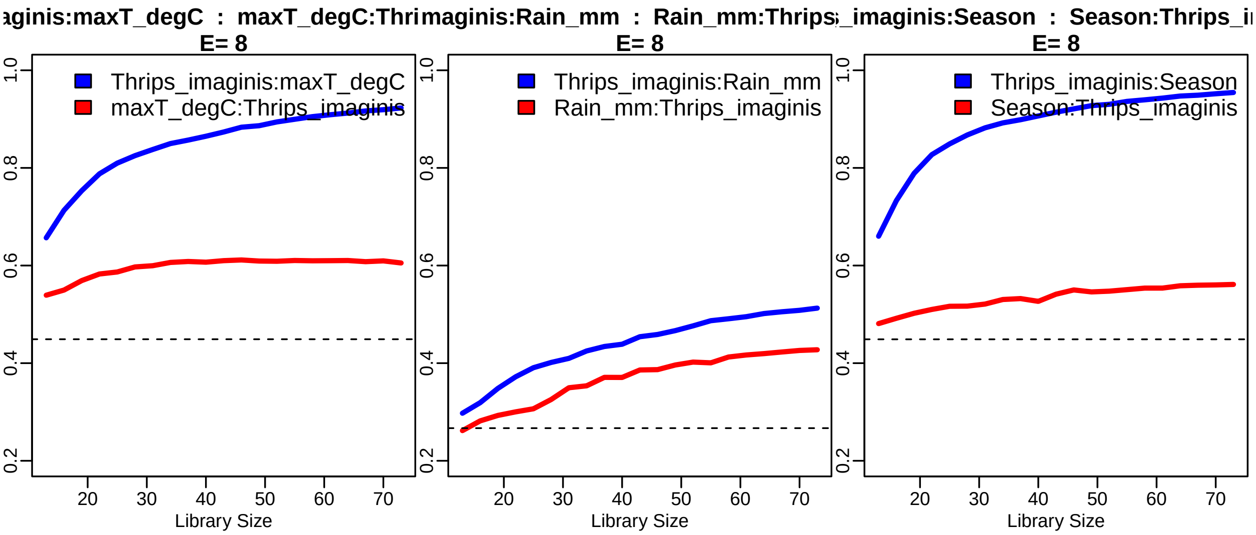


Figure 13: Thrips cross mapped to climatic variables. Vertical axis is cross map prediction skill (rho).

The results show evidence of convergence for *Thrips* cross mapping to temperature and season variables, with the ρ at maximum library size `L` significantly exceeding linear correlation. The rain variable does not indicate a substantially different cross map interaction, appearing confounded as to causal influence.

In addition, we are still left with the conundrum that temperature and to a lesser extent, rainfall, are easily predicted from the seasonal cycle, and so we cannot immediately ignore the possibility that the cross map results are an artifact of shared seasonal forcing.

To reframe, we wish to reject the null hypothesis that the level of cross mapping we obtain for `maxT_degC` and

Rain_mm can be solely explained by shared seasonality. This hypothesis can be tested using randomization tests based on surrogate data. The idea here is to generate surrogate time series with the same level of shared seasonality. Cross mapping between the real time series and these surrogates thus generates a null distribution for ρ, against which the actual cross map ρ value can be compared.

```r
# Create matrix with temperature and rain surrogates (1000 time series vectors)
surr_maxT = SurrogateData(Thrips$maxT_degC, method = "seasonal", T_period = 12, num_surr = 1000,
    alpha = 3)
surr_rain = SurrogateData(Thrips$Rain_mm, method = "seasonal", T_period = 12, num_surr = 1000,
    alpha = 3)

# Rain cannot be negative
surr_rain = apply(surr_rain, 2, function(x) {
    i = which(x < 0)
    x[i] = 0
    x
})

# data.frame to hold CCM rho values between Thrips abundance and variable
rho_surr <- data.frame(maxT = numeric(1000), Rain = numeric(1000))

# data.frames with time, Thrips, and 1000 surrogate climate variables for CCM()
maxT_data = as.data.frame(cbind(seq(1:nrow(Thrips)), Thrips$Thrips_imaginis, surr_maxT))
names(maxT_data) = c("time", "Thrips_imaginis", paste("T", as.character(seq(1, 1000)),
    sep = ""))

rain_data = as.data.frame(cbind(seq(1:nrow(Thrips)), Thrips$Thrips_imaginis, surr_rain))
names(rain_data) = c("time", "Thrips_imaginis", paste("R", as.character(seq(1, 1000)),
    sep = ""))

# Cross mapping
for (i in 1:1000) {
    targetCol = paste("T", i, sep = "")  # as in maxT_data

    ccm_out = CCM(dataFrame = maxT_data, E = E, Tp = 0, columns = "Thrips_imaginis",
        target = targetCol, libSizes = "73 73 5", sample = 1)

    col = paste("Thrips_imaginis", ":", targetCol, sep = "")

    rho_surr$maxT[i] = ccm_out[1, col]
}

for (i in 1:1000) {
    targetCol = paste("R", i, sep = "")  # as in rain_data

    ccm_out = CCM(dataFrame = rain_data, E = E, Tp = 0, columns = "Thrips_imaginis",
        target = targetCol, libSizes = "73 73 5", sample = 1)

    col = paste("Thrips_imaginis", ":", targetCol, sep = "")

    rho_surr$Rain[i] = ccm_out[1, col]
}
```

Seasonal Surrogate Test We now have a null distribution, and can estimate a *p*-value for rejecting the null hypothesis of mutual seasonality.

```
1 - ecdf(rho_surr$maxT)(ccm_matrix["maxT_degC", "Thrips_imaginis"])
```

```
## [1] 0
```

```
1 - ecdf(rho_surr$Rain)(ccm_matrix["Rain_mm", "Thrips_imaginis"])
```

```
## [1] 0.039
```

In the case of temperature, the CCM influence we estimated (0.9227) is higher than the linear correlation (0.449), and is highly significant in relation to a surrogate null distribution. Regarding rainfall, the CCM influence (0.5118) is higher than the linear correlate (0.2668), but not significant at the 95th percentile of the surrogate null distribution. We note that the original Thrips data collections were at a much higher frequency than those available through the GPDD, and that monthly accumulated rainfall may be inadequate to resolve lifecycle influences on a species with a lifecycle of approximately one month. With more highly resolved data, it may well be possible to establish significance.

Package Core

The **rEDM** package is implemented as a wrapper to the `cppEDM` library. All EDM algorithms are executed in the core `cppEDM` library and interfaced through the `Rcpp` package.

Data Input

Data can be input as an R `data.frame`, or read from a `.csv` file. In either case, the first column must define a time vector, and all columns are expected to be named. The time vector can be a string encoding a Date, or Datetime format. All subsequent columns are expected to be numeric.

S-Map coefficients and embedded data

`SMap()` should be called with a DataFrame that has columns explicity corresponding to dimensions `E`. This means that if a multivariate data set is used, it should not be called with an embedding from `Embed()` since `Embed()` will add lagged coordinates for each variable. These extra columns will then not correspond to the intended dimensions in the matrix inversion and prediction reconstruction and subsequent S-map coefficients. In this case, use the `embedded = TRUE` parameter with the multivariate data so that columns selected correspond to the proper dimension.

Parameters

Since **rEDM** is a wrapper for the `cppEDM` library, parameters largely correspond to function parameters of `cppEDM`. Primary parameters are tabulated here.

Parameter	Description
pathIn	Filesystem path to input 'dataFile'. CSV format.
dataFile	CSV format data file name. The first column must be a timeindex or time values. The first row must be column names.
dataFrame	Input data.frame. The first column must be a time index or time values. The columns must be named.
pathOut	Filesystem path for 'predictFile' containing output predictions.

Parameter	Description
predictFile	Observation and Prediction output file name. CSV format.
smapCoefFile	Output file containing S-map coefficients.
lib	String with start and stop indices of input data rows used to create the library of observations. A single contiguous range is supported.
pred	String with start and stop indices of input data rows used for predictions. A single contiguous range is supported.
D	Multiview dimension.
E	Embedding dimension.
Tp	Prediction horizon (number of time column rows).
knn	Number of nearest neighbors. If knn=0; knn is set to E+1 for Simplex(); set to number of data rows for SMap().
tau	Lag of time delay embedding specified as number of time column rows.
theta	In Smap: S-Map neighbor localisation exponent. Single numeric.
theta	In PredictNonlinear: A whitespace delimeted string with values of S-map localisation parameters to be evaluated.
exclusionRadius	Excludes vectors from the search space of nearest neighbors if their relative time index is within exclusionRadius.
columns	String of whitespace separated column name(s) in the input data used to create the library.
target	String of column name in the input data used for prediction.
embedded	Logical specifying if the input data are embedded.
validLib	Conditional embedding. Boolean vector identifying time series rows to use in state-space library.
noTIme	Default False. Set True to not require first column of data to be time.
ignoreNan	SMap: default True. Redefine lib to ignore nan in data and embedding.
generateSteps	Generative feedback predictions in Simplex or SMap.
parameterList	Add parameter dictionary to return objects in Simplex; SMap; CCM; Multiview.
libSizes	String of 3 whitespace separated integer values specifying the intial library size; the final library size; and the library size increment for CCM.

Parameter	Description
sample	Integer specifying the number of random samples to draw at each library size evaluation for CCM.
random	Logical to specify random (‘TRUE’) or sequential library sampling in CCM.
includeData	Logical to return all CCM projection data frames.
seed	Integer specifying the random sampler seed in CCM. If ‘seed=0’ a random seed is generated.
multiview	Number of multiview ensembles to average for the final prediction estimate in Multiview.
trainLib	Use in-sample (lib=pred) prediction for multiview ranking.
excludeTarget	Exclude target variable from multiviews.
maxE	Maximum value of E to evalulate in EmbedDimension.
maxTp	Maximum value of Tp to evalulate in PredictInterval.
numThreads	Number of parallel threads for computation in EmbedDimension; PredictInterval and PredictNonlinear.
verbose	Logical to produce additional console reporting.
const_pred	Logical to add a *constant predictor* column to the output. The constant predictor is X(t+1) = X(t).
showPlot	Logical to plot results.

Acknowledgements

rEDM is an evolving implementation of EDM code. Previous versions were developed by George Sugihara, Alan Trombla, Richard Penner, Victor Wong, Martin Casdagli, Jerome Cartagena, Mohsen Azarbayejani, Ava Pierce, Jennifer Trezzo, and Hao Ye.

References

Casdagli, Eubank, Farmer, and Gibson. 1991. “State Space Reconstruction in the Presence of Noise.” *Physica D: Nonlinear Phenomena* 51 (1-3): 52–98.

Davidson, and Andrewartha. 1948a. “Annual Trends in a Natural Population of *Thrips Imaginis* (Thysanoptera).” *Journal of Animal Ecology* 17: 193–99.

———. 1948b. “The Influence of Rainfall, Evaporation and Atmospheric Temperature on Fluctuations in the Size of a Natural Population of *Thrips Imaginis* (Thysanoptera).” *Journal of Animal Ecology* 17: 200–222.

Deyle, May, Munch, and Sugihara. 2016. “Tracking and Forecasting Ecosystem Interactions in Real Time.” *Proceedings of the Royal Society of London B* 283.

Deyle, and Sugihara. 2011. “Generalized Theorems for Nonlinear State Space Reconstruction.” *PLoS ONE* 6: e18295.

Dixon, Milicich, and Sugihara. 1999. “Episodic Fluctuations in Larval Supply.” *Science* 283: 1528–30.

Lorenz. 1963. “Deterministic Nonperiodic Flow.” *Journal of the Atmospheric Sciences* 20 (2): 130–41.

———. 1996. “Predictability – a Problem Partly Solved.” *ECMWF Seminar on Predictability* I.

Moran. 1953. “The Statistical Analysis of the Canadian Lynx Cycle Ii. Synchronization and Meteorology.” *Australian Journal of Zoology*, 291–98.

NERC Centre for Population Biology, Imperial College. 2010. “The Global Population Dynamics Database Version 2.”

Sauer, Yorke, and Casdagli. 1991. “Embedology.” *Journal of Statistical Physics* 65 (3-4): 579–616.

Sugihara. 1994. “Nonlinear Forecasting for the Classification of Natural Time Series.” *Philosophical Transactions: Physical Sciences and Engineering* 348 (1688): 477–95.

Sugihara, and May. 1990. “Nonlinear Forecasting as a Way of Distinguishing Chaos from Measurement Error in Time Series.” *Nature* 344: 734–41.

Sugihara, May, Ye, Hsieh, Deyle, Fogarty, and Munch. 2012. “Detecting Causality in Complex Ecosystems.” *Science* 338: 496–500.

Takens. 1981. “Detecting Strange Attractors in Turbulence.” *Dynamical Systems and Turbulence, Lecture Notes in Mathematics* 898: 366–81.

Ye, and Sugihara. 2016. “Information Leverage in Interconnected Ecosystems: Overcoming the Curse of Dimensionality.” *Science* 353 (6302): 922–25.