18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

# Detecting qualitatively different behaviours in complex models

**F. Boschetti**

*CSIRO – CMAR, Australia*
*Email: Fabio.Boschetti@csiro.au*

**Abstract:** We suggest that one of the main purposes of modelling is to explore the potential dynamical behaviours a system can display. Within this view, we aim to discriminate model behaviours which appear to be qualitatively different given a problem at hand. This approach fits nicely within a pre-cautionary approach to ecological and social problems aimed to inform policy-makers on the range of scenarios a policy may need to address.

Numerical modelling is increasingly being used to inform policy-making with examples including resource management, biodiversity conservation, global warming mitigation and economic policy. The interpretation of modelling results thus has the potential to profoundly affect our environment and millions of people.

Currently, there is an on-going discussion among modelling practitioners on what a model output represents, how it should be interpreted and what its overall scientific significance is: views cover a continuum between two extremes: one suggests that models can provide only a qualitative understanding of the modelled process and their output simply offers insight into general trends; another sees a model as a virtual laboratory in which real processes are roughly mimicked and whose outcome can be interpreted as predictions. Somewhere in between these views, a number of practitioners suggest that the purpose of modelling is to explore the potential behaviours a system can display. This is the framework we adopt in this work and we try to design an algorithm able to discriminate different model behaviours from a numerical model output.

There are three main challenges in implementing this approach: the first one is how to *define* and *discriminate* different behaviours. This is clearly problem-specific and depends not only on the purpose of the analysis but also on the kind of output a model produces. We define a number of simple measures able to detect both local and global features in the model output and we discuss how the method could be extended to qualitative model output, that is a subjective evaluation of the model output performed by an expert user.

The second challenge is how to *detect* different behaviours, which we address via a search in a high-dimensional input space. Finally, once a set of different behaviours have been found, these need to be presented to the user and if many of such behaviours have been detected some sort of *classification* and *simplification* is also needed. We employ a Self-Organised Map to allow an approximate visual representation of our results. While these tools do not provide for an 'exact' analysis of the results, we believe they allow a potential decision-maker to obtain a rough picture of the variability and the range of behaviours policies may need to address.

In general, the proposed approach should not be seen as an avenue to obtain firm problem-independent answers on a model behaviour, rather as a tool to highlight difference in model behaviours and provide their rough categorization. This information can then be used to guide a more focussed search of the model space aimed at answering problem-dependent specific questions in more details. Here we provide a proof of concept on a number of numerical models and discuss an extension to participatory modeling.

***Keywords:*** *Ecological modeling, Time series analysis, Anomaly detection, Visualisation*

# 1. THE APPROACH – DETECTING LOCAL FEATURES IN A 2D EXAMPLE

In this section I describe the numerical implementation of the approach via a working example: I employ a θ-Ricker map as a population dynamics model and I simplify the analysis so that all steps can be easily visualised in 2D.

I define the θ-Ricker map as in Ellner and Turchin (2005):

$$N_{t+1} = N_t \exp(r(1 - N_t^{\theta}) + \sigma W_t \qquad \text{Eq 1}$$

where $N_t$ is the population at time *t*, *r* is the growth rate, *W* is a normally distributed random variable with mean 0 and variance 1, and *σ* is the intensity of the additive noise.

I aim to study the behaviour of the time-series (*ts*) generated by the θ-Ricker map (*Rm* in the following) as a function of a 2D input parameter space represented by *r* and θ, ($ts = Rm(r, \theta)$).

For this application I also assume that we are not interested in the overall behaviour of *ts,* rather in some of its local features (extension to the analysis of global features is discussed in the following section). This addresses the classes of problems for which local extreme values or local trends are important.

I assume I have a signal processing method which allows to extract local features from the time-series: in this work I embed *ts* in a 2D delayed-coordinate space represented by the pairs [$ts_t$,$ts_{t-1}$] (Takens, 1981; Kantz and Schreiber, 1999) and I call these *extracted features* in the rest of the document. This description leads to a natural way to define the difference between two *extracted features* as their Euclidian distance in the embedding space.

I can now describe and visualise the procedure used to extract features of interest. This is divided into three stages: in the first stage I attempt to describe the most likely model behaviour, where likelihood is understood, albeit loosely, in a Bayesian sense: these are the features we expect to be present in the output of the model given our knowledge and expectation on the input parameter. In the rest of the document I call this 'standard' model behaviour. In the second stage I actively look for model behaviours which are as different as possible from the standard one and I define the behaviours I find as 'anomalous'. Once again this should be understood within a Bayesian framework: these behaviours are anomalous only in relation of our previous knowledge of the model. Finally, in the third stage of the procedure I update our knowledge of the system behaviour by including the anomalous features into the set of standard ones and iterate, thereby adaptively expanding the scope of the search into the model behaviour space as well as redefining the understanding of the model and consequently the understanding of what represents standard and anomalous behaviour.

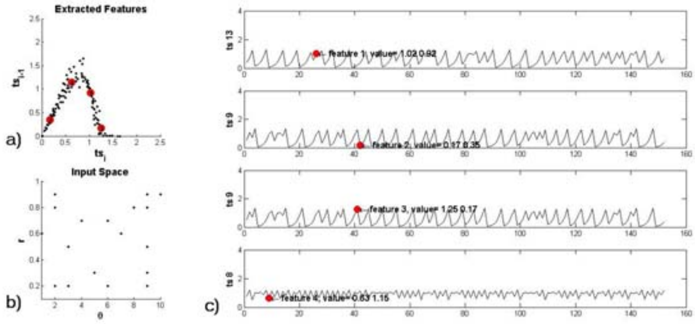

**Figure 1.** Detection of 'standard' features in the θ-Ricker map; the red dots show the representatives selected by the clustering algorithm. a) 20 random points from the *θ-r* input parameter space; b) 2D delayed-coordinate feature space with all features extracted from the 20 time series; c) time series from which the 4 representatives have been selected, and their locations along the time series themselves.

## 1.1. Defining *'standard'* system behaviour

Problem-specific knowledge and experience may inform us on expected model behaviours and these can be represented as a collection of time-series generated by the model under expected input parameters. If this information is not available, we can run the model a number of times under random input parameters chosen within expected ranges and collect these time series. From these time-series I obtain a collection of *extracted features* as described above. In the case of the θ-Ricker map this results in a set of points in the 2D embedded space as can be see in Figure 1.

Figure 1b shows the input parameter space of dimension *θ, r* containing 20 points used to generate 20 random time-series. From the time-series I obtain the 2D *extracted features* displayed in Figure 1a which shows the feature space of dimensions $ts_t, ts_{t-1}$. The dark points show all features contained in the initial 20 time series corresponding to the 20 input parameters in Figure 1b.

In order to summarise the information contained in Figure 1a, I group the points into a number of clusters (for the choice of the number of clusters see the Discussion section). I carry out the clustering via the VSH algorithm (Frey and Delbert, 2007) and the cluster centres are displayed in Figure 1a as 4 large red dots. In our approach these cluster centres represent the 'standard' behaviour displayed by the *Rm* model according to the information so far collected. Figure 1c shows the time series from which the extracted features have been extracted and the exact location of the cluster centres along these time series. The 4 extracted features seem to capture the structures contained in the time-series, which cover both average values (extracted features 1 and 4) and 'extreme' values (features 2 and 3) like peaks, valleys and sudden jumps.

To be consistent all extracted features in Figure 1b should be defined as 'standard'; to do so, I take the maximum intra cluster distance and draw a circle around each cluster centre with radius equal to such distance. Obviously, by construction, all points found so far fall inside one of the circles or, equivalently, so far we have no information about the existence of any features outside the circles. I then define as 'anomalous' features which fall outside such circles and I devise a procedure to search for those features.

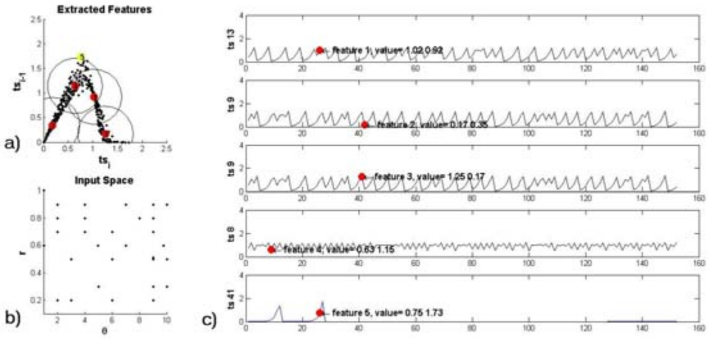

**Figure 2.** Detection of 'anomalous' features in the θ-Ricker map; a) point 5 (yellow dot) has been detected outside the range defined by the

### 1.2. Detecting '*anomalous*' system behaviour

According to the definition in the previous section, searching for anomalous system behaviours coincides with searching for features in the *extracted feature* space which lie outside the circles in Figure 2a. We can carry out this search via a numerical optimisation method in which the cost function is the distance between one feature and the closest cluster centre (the cost function needs to be maximised). Since I expect that more than one such feature may exist, a natural choice is to use a population-base search algorithm and in this work I use a Genetic Algorithm (GA, see Davis, 1991) with a population size of 16 individuals (details of the specific Genetic algorithm used in this work can be found in Boschetti *et al.*, (1996).

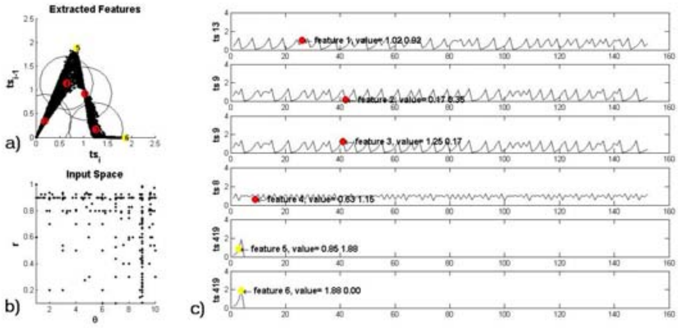

**Figure 3.** Final result of the analysis of the θ-Ricker map; a) a further 'anomalous' point (6) has been found while point 5 has been moved further away from the 4 'standard' points; b) the final sampling of the parameter space is not uniformly spaced rather focussed on extreme values of *θ* and *r*; c) locations of the detected features along the time series they belong to..

After two iterations, the GA finds an anomalous feature, that is a feature not included within the ranges of the cluster centres as defined above. This feature can be seen as a yellow dot both in the extracted feature space in Figure 2a and in the corresponding time-series in Figure 2c. It represents an extreme value of the time series which had not been seen previously.

The presence of this new feature affects the optimisation process: first, it adds a new feature to the cluster centres and consequently it changes the shape of the solution surface. Second, I want the newly discovered anomalous features to represent a new class of behaviours: if in the subsequent search another feature is found in the proximity of this anomalous feature, it will be disregarded if it is closer to a cluster centre than the anomalous features and it will replace it otherwise.

The optimisation then proceeds in this new mode. Figure 3a shows the result after 20 iterations. Two items are worth noticing: first, a new anomalous feature has been detected (dot 6 in Figure 3a and in the time series in Figure 3b). Second, the anomalous feature 5 has been replaced by a new feature, similar to the previous feature 5 but further away from the original 'standard' behaviours (that is more 'extreme') than the previous.

These two new extracted features add information to our previous knowledge of the system behaviour by informing us that larger values in the population dynamics are possible than previously thought. This information may have relevance for decision-making: if the modelled species was a pest, this result would warn us of a potential bigger threat than previously expected.

As a final consideration, the plot in Figure 3c illustrates the sampling of the input parameter space carried out by the Genetic Algorithm; as can be seen the sampling is skewed, the search has guided the GA towards areas of the parameter space were anomalous behaviours are likely to be found, which in this case correspond to high values of both $\theta$ and $r$.

This was a simple test case since the full range of behaviours of the θ-Ricker map could be easily discovered without the need of the specific state space exploration I described. The purpose of this test case was merely to illustrate the proposed approach. More challenging test cases are then discussed in the rest of the document.

## 2. DETECTING GLOBAL FEATURES IN HIGHER DIMENSIONAL MODELS

In this second example I employ a more complicated model, the NPZ model from Edwards and Brindley, (1999). This model is higher dimensional, which poses challenges in the visualisation of the results. Also, unlike the previous example, here I focus on the global structure of the output time series, rather than on local features; in particular, for global structure I refer to the *ensemble* of features present in a time series and develop a measure to evaluate how different two time series are based on these ensembles. Similarly to Boschetti (2008), the final aim is to determine how many different behaviours the NPZ model may display and roughly partition the input state space based on these output behaviours. To study the behaviour of the NPZ model I focus on the time series of phytoplankton biomass (Taken's theorem (1981) guarantees that, provided I embed the time series in a sufficiently high dimensional space, the study of the phytoplankton biomass provides information about the overall behaviour of the NPZ system).

As in the previous analysis of the θ-Ricker map, I embed the time series into a delayed-coordinate space. In this case I choose an embedding dimension of 7, which allows us to analyse longer structures and higher derivatives. Unlike the previous example however, I do not analyse the extracted features in isolation, rather I consider the ensemble of all the extracted features from an individual time series, which corresponds to a set of points in the embedded space. In order to carry out the algorithm as described in the previous section I thus need a measure of the difference between ensembles of extracted features rather than between individual extracted features.

As before, I define this difference as a distance in delayed-coordinate space. This difference can be defined in many different ways: in Boschetti (2008) I used the difference between the statistical complexity (Crutchfield and Young, 1989; Shalizi and Shalizi, 2004) of the time series, which is calculated from the ensemble. Many other information-theoretic measures could similarly be employed (Kantz and Schreiber, 1999; Schreiber, 2000; Ray, 2004). Most of these measures require the time series to be symbolised (that is discretised), either before or after embedding. As discussed in Daw *et al.*, (2003) and Kennel and Buhl (2003) the symbolisation imposes an arbitrary distortion on the time series which I try to avoid in this work; consequently, here I define a distance which is based solely on the geometrical location of the extracted features in embedded space. Given two ensembles *e1* and *e2* of extracted features, the distance from ensemble *e1* to ensemble *e2* is defined as the mean distance from each feature in *e2* to the closest among the features in the *e1*; to clarify:

1) for each extracted feature $f2_i$ belonging to ensemble *e2*
2) look for the feature $f1_j$ in ensemble *e1* with shortest Euclidean distance $| f1_i , f2_j |$, call it $d_{i,j}$
3) the distance from ensemble *e1* to ensemble *e2* is then defined as $D_{e1 \to e2} = mean(d_{i,j})$

In general, this distance is not symmetric, that is $D_{e1\rightarrow e2} \neq D_{e2\rightarrow e1}$: the distance from the 'more complex' time series (which generates the ensemble with more diverse extracted features) tends to be shorter that the distance in the opposite direction. This measure can be interpreted as the amount of distortion we need to impose to ensemble *e1* in order to reconstruct ensemble *e2*. This lack of symmetry is not a concern in our application since the clustering algorithm I adopt is able to account for it (Frey and Delbert, 2007).

Given this distance, I can now proceed with the algorithm as described above. I run the NPZ model 40 times with random input from a 6 dimensional parameter space (the chosen input parameters are described in Error! Reference source not found.). From these runs the clustering algorithm selected 4 time series summarising the 'standard' behaviour of the model. I then run the GA for 260 further model run (13 iterations of a population of 20 individuals) looking for time series as different as possible from the currently stored time series. The algorithm found an additional 8 time series which I define as 'anomalous'. To summarise, I sampled 300 points from the input parameters space, stored 300 time series and from them extracted 12 which characterise the main behaviours arising from the model.

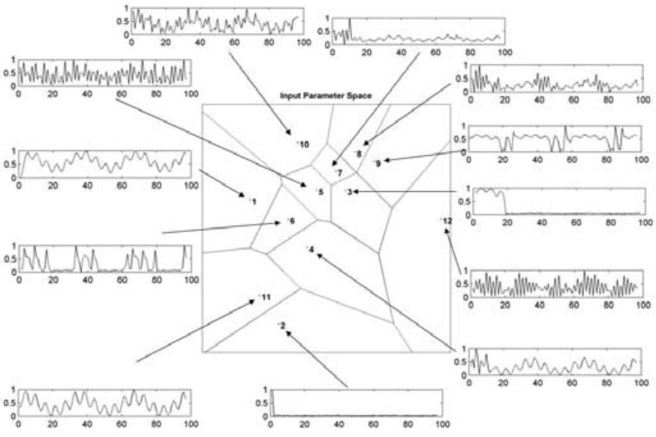

**Figure 4.** 12 different behaviours displayed by the NPZ model and their location on the SOM 2D representation of the model input parameter space

In this test case both the input parameter and the feature space are high dimensional and consequently it is difficult to visualise the results. To circumvent the problem, I employ a self-organised map (SOM), (Kohonen, 2001). A SOM maps vectors in a high-dimensional space into a lower dimensional space (2D in our case) by respecting the vector neighbourhood topology, that is, by plotting along side points which are close in the original high-dimensional space. The result can be seen in Figure 4. The central plate shows the SOM U-Matrix (Kohonen, 2001): this does not have a specific physical interpretation, which is why the axis re not labelled, rather it should be understood as the display of an acceptable arrangement of the 300 points sampled in the input parameter space such that their distance in the 2D plate is as close as possible to their distance in the original 6D parameter space. Over the U-Matrix I plotted the 12 points in the input parameter space which generated the 12 time series detected by the search process; I interpret these 12 points as representative of the behaviour of the NPZ model and partition the U-Matrix by assigning the remaining points to one of these representatives according to their ensemble distance.

In Figure 4 I plotted the time series corresponding to each of the 12 selected points, which shows the different behaviours we may expect from the NPZ model. In many applications it is also of interest to know what parameters are responsible for these different behaviours. Since I have sampled 300 values from the input parameters space I may employ this sampling to extract some rough information from them. I attempt to do this visually in Error! Reference source not found.. Following a method tested in several other studies involving searches in high dimensional spaces (Boschetti et al., 2003; Wijns et al., 2003a; Wijns et al., 2003b; Boschetti, 2005; 2008), the value for each of the 6 input parameters is mapped over the U-Matrix and interpolated; this results in the 6 plates in Error! Reference source not found. which give us an approximate visual description of how each parameter varies in the domain characterised by different behaviours. Obviously, these plates are the result of a process prone to error: first points are arranged from 6D to 2D, then an interpolation is performed in the resulting 2D space which adds to the distortion; we thus can not expect an exact outcome from this analysis, rather only a rough impression which may lead us to insights for further analysis. This also will be further discussed below.

Error! Reference source not found. suggests that the Predation on P (plate b) seems to be responsible for the partition between the behaviours in domains 1, 10 and 11 versus the rest of the domain. Similarly, plate e seems to suggest the role of concentration of N on the behaviour of domains 2 and 11. Domains 3, 5 and 7

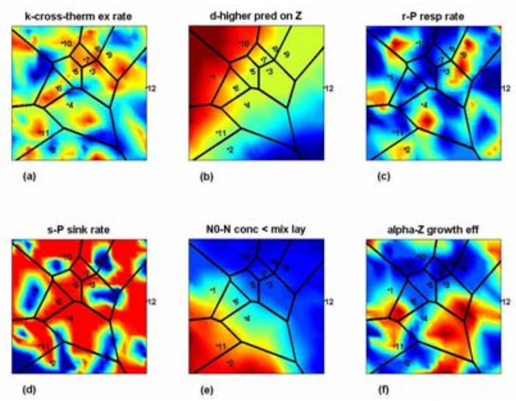

**Figure 5.** Values of each of the 6 NPZ model input parameters interpolated over the SOM 2D representation on the input parameter space

seems to share fairly similar values in plates a, b and e, suggesting that their different behaviour may be due to the value of the parameters in the other plates, particularly c and f. Similarly, the high value in plate c seems to be responsible of the behaviour in domain 9. Because of the distortions in the generation of these maps these insights should not be taken as conclusive, rather may suggest directions for further enquire.

## 3. EXTENSION TO PARTICIPATORY MODELLING

In participatory modeling we may be less interested in the precise numerical output of a model, but it may still be on great interest to inform stake holders of the possible behaviours a system can display; the rationale may include designing policies in order to achieve, avoid or respond to certain outcomes. In this framework, modeling can be used to demonstrate how certain system behaviours may arise and also to reveal the potential for behaviours which may be unexpected or never before encountered.

This could be achieved by involving human expertise directly in the analysis, interpretation and processing of the model output. As discussed in the introduction, our approach consists of three main components: the *discrimination* of novel behaviors, the *search* in the parameter space for new behaviors and the final *visualization* of the results. In a participatory modeling setting, the second and third components (search and visualization) can be carried out with no modification as described above. Human intervention could be used in the first component (the discrimination of behaviours). Let’s assume the modeling output consists of an image or an animation. By employing Interactive Inversion (see Boschetti et al, 2008, for applications to ecological modeling and references within for the general approach) the stake holders, together with the modelers, could perform a visual analysis of the model output and discriminate between standard and anomalous behaviors via joint discussion based of the accumulated experience on the system at hand. This discrimination could then feed back to the Genetic Algorithm which performs the search in the model *input* space as described above. Basically the analysis of the model *output* space would be replaced by the intervention of the expert users (stake holders and modelers), while the rest of the approach would proceed as described above. The Genetic Algorithm would still be able to direct its search to promising areas of the parameter space (area which are likely to provide new anomalous behaviors) which the users would then further evaluate. Should this approach be followed, a crucial role would be played by the user interface in order to facilitate the users’ intervention, discrimination and feed-back of this information to the numerical algorithm. We can envisage a user interface enabling users to drag and drop model results into different classes according to behaviours judged ‘different’ according to the requirements of the problem at hand. These classes would then represent the clusters in the procedure described above in this paper, and the overall approach could proceed by processing these human-selected classes. This is the direction I intend to follow in future research.

## 4. DISCUSSION

The approach here described employs a number of algorithms. Most of them have been proposed in the literature to address problems for which a ‘best practice’ is not yet agreed on. These include the clustering algorithm, the global search algorithm and the dimensionality reduction algorithm. The same applies to a number of heuristics used: the choice of the number of initial clusters, the proper embedding dimension, the stopping criteria for each algorithm and so on. Obviously, each of these components could be changed and by so doing the outcome of the method could be affected; more testing is necessary to assess the robustness of the method in different applications.

When the model is not deterministic, a given set of input parameters may result in different output values or time series. This can complicate all steps of the analysis: the definition of the clusters, the search in the model parameter space as well the visualisation of the results and the use of error bars to account for this may be beneficial, since they would provide a manager with an idea of how reliable the expectation of a certain result might be. So far, I have applied this approach only to model output in 1D representing time series of single values. Extensions to 2D and 3D are in principle possible (Shalizi et al., 2004); they involve much heavier computation but they are useful only provided the output consists of sequences of profiles or images ordered in time (animations). Should the output not have this time component, issues may arise in implementing the algorithms meaningfully and care is needed (Feldman and Crutchfield, 2003). In particular, the problem of how to analyse static images is closely related to current applications in image processing, image recognition and artificial intelligence aimed at classifying large images data bases or recognise specific events requiring human attention, for example in surveillance.

A natural approach to address this class of problem may involve algorithms similar to the ones used in this work: they could be used to first detect the *interesting* features in an image or animation, and then we could proceed with the method described above, focusing *only* on these features. Algorithms like neural networks could be trained by experts for specific applications in order to detect such *interesting* features. Needless to say this would increase the complexity of the approach, but improvements in image recognition and artificial intelligence are steady and may be soon relevant to modelling applications.

## REFERENCES

Boschetti, F., (2005), Controlling and investigating cellular automaton behavior via interactive inversion and visualization of the search space. New Generation Computing, 23, 157-169.

Boschetti, F., (2008), Mapping the complexity of ecological models. Ecological Complexity, 5:37-47.

Boschetti, F., de La Tour, A., Fulton, E. and Little, R., (2008), Interactive modelling for natural resource management Submitted.

Boschetti, F., Dentith, M.C. and List, R.D., (1996), Inversion of seismic refraction data using genetic algorithms. Geophysics, 61, 1715-1727.

Boschetti, F., Wijns, C. and Moresi, L., (2003), Effective exploration and visualization of geological parameter space. Geochemistry Geophysics Geosystems, 4(10), 1086.

Crutchfield, J.P. and Young, K., (1989), Inferring Statistical Complexity. Physical Review Letters, 63, 105-108.

Davis, L., (1991), Handbook on genetic algorithms. Van Nostrand Reinhold.

Daw, C.S., Finney, C.E.A. and Tracy, E.R., 2003. A review of symbolic analysis of experimental data. Review of Scientific Instruments, 74, 916-930.

Edwards, A.M. and Brindley, J., (1999), Zooplankton mortality and the dynamical behaviour of plankton population models. Bulletin of Mathematical Biology, 61, 303-339.

Ellner, S.P. and Turchin, P., (2005), When can noise induce chaos and why does it matter: a critique. Oikos, 111, 620-631.

Feldman, D.P. and Crutchfield, J.P., (2003), Structural Information in Two-Dimensional Patterns: Entropy Convergence and Excess Entropy,. Physical Review E, 67, 051104.

Frey, B. and Delbert, D., (2007), Clustering by Passing Messages Between Data Points. Science, 319, 972–976.

Kantz, H. and Schreiber, T., (1999) Nonlinear Time Series Analysis. Cambridge University Press, Cambridge.

Kennel, M.B. and Buhl, M., (2003), Estimating good discrete partitions from observed data: symbolic false nearest neighbors. Physical Review Letters 91, 084102.084101-084102.084104.

Kohonen, T., (2001), Self-organizing maps. Springer, New York, 5001 p.

Ray, A., (2004) Symbolic dynamic analysis of complex systems for anomaly detection. Signal Processing 84, 1115—1130.

Schreiber, T., (2000), Measuring Information Transfer. Phys. Rev. Lett., 85, 461-465.

Shalizi, C., Shalizi, K. and Haslinger, R., (2004), Quantifying Self-Organization with Optimal Predictors. Physical Review Letters, 93, 118701

Shalizi, C.R. and Shalizi, K.L., (2004), Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences. Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference. AUAI Press, Arlington, Virginia.

Takens, F., (1981), Detecting strange attractors in turbulence. Dynamical systems and turbulence, Warwick pp. 366-381.

Wijns, C., Boschetti, F. and Moresi, L., (2003a), Inverse modelling in geology by interactive evolutionary computation. Journal of Structural Geology, 25, 1615-1621.

Wijns, C., Poulet, T., Boschetti, F., Griffiths, C. and Dyt, C., (2003b), Interactive Methodology Applied to Stratigraphic Inverse Modelling. In: A.C.a.R.W. (Eds.) (Editor), Geological Prior Information: Value and Quantification. Geol. Soc. of London Special Publication, London.