Mapping the complexity of ecological models

Fabio Boschetti
CSIRO, Australia

Abstract

We propose to define the complexity of an ecological model as the statistical complexity of the output it produces. This allows for a direct comparison between data and model complexity. Working with univariate time series, we show that this measure 'blindly' discriminates among the different dynamical behaviours a model can exhibit. We then search a model parameter space in order to segment it into areas of different dynamical behaviour and calculate the maximum complexity a model can generate. Given a time series, and the problem of choosing among a number of ecological models to study it, we suggest that models whose maximum complexity is lower that the time series complexity should be disregarded because unable to reconstruct some of the structures contained in the data. Similar reasoning could be used to disregard models' subdomains as well as areas of unnecessary high complexity. We suggest that model complexity so defined better captures the difficulty faced by a user in managing and understanding the behaviour of an ecological model than measures based on a model 'size'.

Introduction

The increasing complexity of ecological models is a growing concern in the modelling community. Ecological models are used to integrate process knowledge from different parts of the system, and in doing so allow us to test system understanding and generate hypotheses about how the system will respond to particular actions via virtual experiments. However, as we strive to make our models more 'realistic', the more parameters and processes we include. With increased model complexity we are less able to manage and understand model behaviour. As a result, the ability of a model to simulate complex dynamics is no more an absolute value in itself, rather a relative one: we need enough complexity to realistically model a process, but not so much that we ourselves can not handle. From a practitioner's perspective, this can be rephrased as: "how complex a model do I need to use in order to study *this* problem with *this* data set?". In this work we propose some steps that begin to address this issue.

Clearly, an answer to the above question requires a definition and a measure of complexity. Importantly, it also requires the measure to be equally applicable to the model *and* to the data, since some sort of comparison is necessary. Often in the modelling community (both inside and outside ecological studies) complexity is seen as somehow related to a model architecture, that is, there is a notion of some sort of monotonic relation between complexity and model 'size', where size accounts roughly for dimensionality, connectiveness, number of interacting processes, etc. It is indeed reasonable to expect that an extra factor/dimension may *potentially* increase the effective space available to the model's state space trajectory. Similarly, it is reasonable to expect that an extra link between model components may *potentially* increase the level of feedback loops in the dynamics. Nevertheless, the relation

between complexity and model 'size' needs to be considered carefully since we may expect a model to behave very differently in different areas of its parameter space; this clearly defies the idea of relating complexity to model 'size' as well as to assign a single measure of complexity to a model.

These considerations lead us to focus on a view of complexity which is more related to a model's dynamical properties, rather than its architecture. Ideally, we would like to develop a tool which answers the following two questions:

- what is the maximal dynamical complexity a given model can generate?
- what kind of different dynamical behaviours can a given model generate?

To help clarify and set upfront the thread of this work, let us suppose we did have such tool and describe how we would use it. The scenario we consider is one in which we measured a time series *T* of a component of an ecological process *E* and we need to choose among three different models *M1*, *M2* and *M3,* which model *E* at different levels of sophistication/realism. We seek the best compromise between complexity and manageability in order to answer the practitioner's question "how can I check if this model is appropriate to study *this* problem with *this* data set?". We envisage the following approach:

1) we calculate the maximum complexity *M1, M2* and *M3* can generate, (call it $C_{M1}^{\max}$, $C_{M2}^{\max}$ and $C_{M3}^{\max}$);
2) we calculate the complexity of the time series (C_T);
3) suppose $C_{M1}^{\max} < C_T$; then we can deduce that there are some structures in the time series *T* which model *M1* is not able to reproduce. This does not necessarily refer to specific values in *T*, as much as to some of its dynamical properties; we thus disregard model *M1*;
4) suppose $C_T < C_{M2}^{\max} << C_{M3}^{\max}$; then both models *M2* and *M3* are able to reproduce the dynamics in the time series. However, *M3* seems to be unnecessarily complex, since it is much more complex that *M2*, whose maximum complexity is already sufficient to analyse *T*. The extra complexity in *M3* does not seem necessary for this modelling exercise, and, depending on our purpose, we may or may not decide to disregard *M3*;
5) further, suppose model *M2* behaves differently in different areas of its parameter space, with $C_{M2}^{a} < C_{M2}^{b} < C_{M2}^{c} ..... < C_{M2}^{f}$ where *a, b, c ...f* are different domains. Finally, suppose $C_{M2}^{a} < C_{M2}^{b} < C_T < C_{M2}^{c} ..... < C_{M2}^{f}$. Then we may limit our analysis to the areas *c..f* since the dynamical properties of subdomains *a* and *b* do not allow us to capture all the structures in *T*.
6) we thus have been able to restrict our analysis to one (or two) model, and, within this model, to a subdomain of its entire parameter space.

Our approach to develop the tool to enable such analysis is the following:

- first, among the many different measures of complexity available in the literature, we adopt the *statistical complexity* defined in Crutchfield and Young (1994), which is commonly applied to time series analysis. Then, we define the complexity of a model as the complexity of the time series it can generate. In the first part of the paper we give a rationale for a) choosing this particular measure of complexity, b) for associating model complexity to time

series complexity and c) for using this idea as a measure of the 'difficulty' a user may encounter in employing and managing a specific model.
- Second, we show that this measure is able to detect areas in the model parameter space with different dynamical behaviours. Also, we show that this can be achieved in a sort of 'black box' approach, in which we do not need to specify what feature of the dynamics we wish to detect. We test the potential of this method against a number of analytical results.
- Finally, we search the model parameters space in order to establish the maximum statistical complexity a model can generate. As a by-product of this search, we visualise the extensive sampling of the parameter space in order to roughly partition it into areas of different dynamical behaviour.

We conclude with a careful discussion of the limitations of the current method and with a sketch for future developments.

This approach combines the use of several algorithms and tools, a detailed description of which would not only result in a very long paper but also obscure the overall thread of the method. Consequently, we make extensive use of appendices to briefly describe some of the algorithms while we refer the reader to the related literature for more details.

Statistical complexity

In the information theory literature the concept of complexity is closely related to predictability and in particular to the amount of information required (difficulty) to achieve optimal prediction. One of the first and most popular attempts to characterise this idea is Kolmogorov's algorithmic complexity (also called Kolmogorov-Chaitin complexity, see Li and Vitányi, 1997 and Chaitin, 1969). Given a time series, this is defined as the length (in bits of information) of the minimal program which can reproduce the time series. According to this definition, a fully periodic time series has low complexity since very short program (which stores 1 period and outputs it indefinitely) can reproduce the entire time series exactly. Departures from periodic behaviour towards randomness would require programs of increasing length and consequently display increasing algorithmic complexity.

In relation to our work it is important to notice the following: first, a time series and a model (minimal program) which can reconstruct it are used interchangeably in the definition of complexity. On this very idea we base our definition of ecological model complexity. Second, according to Kolmogorov's definition, a fully random time series has maximum complexity, since the only program which can reproduce it is a program which stores and outputs the time series itself (a random time series is, by definition, not predictable and consequently not compressible). This somehow contradicts our intuition about complexity, which is usually seen as something in between order and randomness. Finally, this definition does not come with a tool for its computation, since we can never ensure the model of a time series is of minimal length (see Chaitin, 1982 for a formal proof).

To circumvent some of these problems, Crutchfield and Young (1994) propose that complexity is characterised by the amount of information needed to perform *useful* "statistical" prediction. In other words, they seek to achieve a prediction which

captures the statistical properties of the time series, rather that the exact time series itself. As in the case of Kolmogorov's complexity, little information is needed to capture the statistical properties of a simple periodic function. Unlike Kolmogorov's definition though, very little information is needed to statistically reproduce a random time series. Since the time series is random, and it can not be predicted, no amount of memory (effort) can help improving our predictive ability, i.e., an 'optimal' prediction can be performed with zero memory (there is no point in storing the outcomes of roulette draws to bet on the next draw). So we have a definition of complexity which captures our intuition that very simple, as well as fully random time series, have low complexity and that processes in between ('at the edge of chaos') have high complexity. The main strength of this definition is that it also comes with a procedure to calculate it numerically. This results from an algorithm (Causal State Splitting Reconstruction, CSSR, Shalizi et al 2004) which can *provably* reconstruct the *minimal* model able to capture the statistical properties of the time series (Shalizi et al, 2003). The approach is summarized in the following:

1) take a symbolized time series (that is a time series whose values are restricted to a finite alphabet (in the Discussions we will address the implications of this limitation);
2) run the CSSR algorithm to reconstruct the causal states of the process and their transitions (usually called an ε-machine); this represents the minimal model able to reproduce the time series statistically;
3) calculate the entropy of the causal states, which measures the uncertainty in predicting the next state of the system, given the information on its past behaviour and can be seen as a measure of the amount of memory in the system (in bits) which does a *useful* job in prediction; this entropy is the statistical complexity of the time series (or, equivalently, of the minimal model which reconstructs it).

In appendix A we give a brief summary of the CSSR algorithm, while we refer the reader to Shalizi et al (2004) for further details.

Statistical complexity of ecological models

There are examples in the ecological literature of the application of information theory measures to time series. For example, Fath et al (2003) and Mayer et al (2006) use Fisher Information to infer regime changes in dynamical behaviour. Similarly, there are pieces of work aimed at inferring relative roles of determinism and stochasticity in ecological time series (e.g. Hsieh et al, 2005, Ellner and Turchin, 2005). The Statistical Complexity we employ in this work accounts for both of the above measures and allows us to more readily assess whether models capture the *dynamic* characteristics of data and to investigate the sensitivity of model dynamical behaviour to changes in model assumptions.

The work by Crutchfield and Young (1994) and Shalizi et al (2003, 2004) is focussed on a rigorous analysis of stationary univariate stochastic time series of symbols drawn from a finite alphabet. Ecological models are approximating continuous times series of real numbers. The CSSR algorithm and associated information theory measures have not been extended to deal with time series of this kind. If CSSR and associated

measures are to be applied to these time series, a set of assumptions and steps are required to convert the time series to a string of symbols (see Appendix B).

In this section we propose a way of extending the concept of statistical complexity to ecological models by employing the CSSR algorithm. The idea is summarized in Figure 1:

1) take a point *P* in the model parameter space and run the ecological model with initial conditions and parameters defined by *P* in order to obtain a time series *T* of interest (step *a* in Figure 1);
2) employ the CSSR algorithm to reconstruct the ε-machine from the time series (i.e. the minimal model able to reproduce the time series statistically) (step *b*);
3) calculate the entropy of the ε-machine in order to define the statistical complexity of the time series/epsilon machine (step *c*); and
4) assign the value of the statistical complexity to the ecological model parameter space at point *P* (step *d*).

The rationale for this approach is illustrated in Figure 2. We can define an *informal* equivalence between the ecological model at point *P* and the ε-machine so reconstructed, since they both (statistically) reconstruct the time series *T*. Here a few considerations are appropriate. First, the ability of the CSSR algorithm to reconstruct the 'correct' ε-machine depends on the information contained in the time series. This, like most time series analysis techniques, is a data hungry process. We comment briefly on this issue below and we refer the reader to Bertello et al (2005) for more details. Second, as mentioned above, the reconstruction is 'stochastic', not exact. That is, the ε-machine can reproduce a time series with the same dynamical features rather than the exact time evolution.

Does the complexity of the ecological model so defined capture our intuition of model complexity? The statistical complexity measures the amount of information needed in order to make a useful prediction of the future time series behaviour given information about its past. This is a measure of how difficult it is to predict or model the time series. With a slight abuse of terminology, a rough analogy would be as follows. Consider the difficulty encountered by a modeller attempting to *predict/guess the model behaviour* at a particular point in the parameter space. For a very complex model, a user will find it difficult to guess how the model will behave at a certain point in the parameter space even given expert knowledge of the model itself, since the dynamical evolution of the actual time series is complex (i.e. difficult to predict stochastically). In our opinion, this is a better view of model complexity than 'size', though in some cases the two may be related.

Under this view of model complexity, it is clear that the complexity of the model may vary depending on the parameters chosen. Ideally, we would like to achieve the scenario illustrated in Figure 3. Given a set of time series produced by model runs generated from different parts of parameter space (thick lines in Figure 3), we'd like to detect regions of similar dynamical behaviour (regions *a, b* and *c* in Figure 3), using estimates of statistical complexity as our measure of similarity (estimated from ε-machines for each time series).

In order to see whether this approach is feasible we first need to answer two questions:

1) can the statistical complexity discriminate between areas of different dynamical behaviour?
2) can this discriminatory power define a rough partition into areas of different dynamical behaviour?

We address these two questions in the following sections.

Detecting different dynamical behaviours

In this section we explore whether the approach described above can be used to discriminate between the different dynamical behaviours displayed by an ecological model. Because our ultimate intent is to apply the method to very different models, we would like to achieve a 'blind' (black-box like) discrimination, without needing to specify what features of the dynamics we are interested in.

In the previous section we have established an informal equivalence between an ecological model with specific initial conditions, the time series it can generate and, via this time series, with the ε-machine reconstructed via the CSSR algorithm. This suggests that in order to detect different dynamical behaviours, we may work on either the ecological model, the time series or the ε-machine. Because of its minimality properties, it seems convenient to focus on the ε-machine. We thus say that the dynamical behaviour of the ecological model at two different locations in the parameter space is similar if the corresponding ε-machines are 'similar'. This rationale is simple: if two ε-machines are similar, the process' states and transition probabilities are similar and so are the dynamical behaviours.

We thus need a criterion to determine the similarity of two ε-machines. The most obvious approach would be to design a metric based on the ε-machine's causal states and the transition probabilities themselves. Unfortunately this is not a trivial task, since different ε-machine may have different numbers of causal states and it may be hard to establish a relation between similar states in different machines. Also equivalent causal states in the two machines may include slightly different features due to errors either in the measurements or in the symbolisation. This not only makes comparing the states difficult, but also makes particularly challenging tracking which state transition in one machine corresponds to which state in the other machine (we refer the reader to Ray (2004) for more details). Consequently, in the rest of the paper we employ the difference in statistical complexity as an approximate measure of similarity between two ε-machines.

The Test Case. We test the idea against known theoretical results. We employ an NPZ model as described in Edwards & Brindley (1999) (EB99 in the following). A brief description of the specific equations used, the list of parameters and their ranges can be found in Appendix E. Edwards & Brindley studied the dynamical behaviour of this NPZ both analytically and numerically. They aimed to analyse how the trajectories in state space vary for different values of the control parameters. In particular they showed the existence of bifurcations at locations where the orbits

change abruptly from a stable steady state to a unstable limit circle, implying an oscillatory behaviour in N, P and Z. They analyse the location Hopf and fold bifurcations in a set of 2D plots in which the effect of varying a number of parameters versus variation in the predation on Z is studied (EB99, Figure 4).

The method. Here we ask whether the statistical complexity is able to detect such bifurcations. The ε-machine of a time series with a stable fix point limit has a single state and consequently its statistical complexity (the entropy of the causal state) is zero. The ε-machine of a time series with a limit cycle, or more complex dynamics, has more states and consequently a higher statistical complexity. This suggests a computational method to detect bifurcations in cases these can not be detected analytically:

1) Choose a subset of *K* ecological model input parameters, $p_k, k = 1...K$. Vary smoothly each of the parameter within suitable ranges, $p_k = i, \min_k \le i \le \max_k$.
2) For each value of p_k calculate the statistical complexity $SC_{\varepsilon_{p_k=i}}$ where $\varepsilon_{p_k=i}$ is the ε-machine of the time series for $p_k = i$.
3) Detect where the Statistical Complexity differs from zero.

The result can be seen in Figure 4, in which we reproduce Figure 4 in EB99 and we compare each plate to the bifurcation detected via the statistical complexity. As can be seen, a good match is found for each plot.

As a further direction of enquiry, we may ask how the statistical complexity behaves inside the bifurcation areas. For several of these plots the statistical complexity within the bifurcation area is constant, which results in a binary map as seen in Figure 5. Here, we display the plot of the predation rate on Z (X axis) versus the respiration rate of Z (Y axis), which is a gray scale version of plate *e* in Figure 4 (white maps high values). The binary nature of the image is clear; the statistical complexity is equal to zero everywhere on the plot, except inside the bifurcation area, where its value is one. The 3D delayed coordinate representations of the time series corresponding to seven points on the plot are also shown, from which the difference between fixed point steady state (corresponding to zero statistical complexity) and limit cycle (corresponding to the statistical complexity value of one) is evident. The statistical complexity value of one is typical of dynamics characterised by 2 states which alternate at each step and consequently have the same probability of occurrence. As mentioned above, the statistical complexity is measured in bits of information. A statistical complexity equal to one means that we need only one bit of information to optimally predict the next state in the dynamics; in other words, if we know what symbol the time series is at the current time, we know that, due to the oscillation, at the next time step the time series will display the different symbol.

The next question we ask is whether other dynamical behaviours are possible in the model under study. Our approach to answering this question is the following:

4) select a ‘default’ set of input parameters and define this as ‘baseline’ behaviour; in this test, such a baseline corresponds to a predation rate on Z equal to zero and all other default values in Table 1.

5) Run the ecological model with the default parameters, generate a time series of Phytoplankton behaviour and calculate a default value for the statistical complexity. Call this SMM_0^{ε} (baseline Statistical Modelling Measure for the ε-machine).
6) Choose a subset of *K* ecological model input parameters, $p_k, k = 1...K$. Vary smoothly each of the parameter within suitable ranges, $p_k = i,\ \min_k \le i \le \max_k$.
7) For each value of p_k calculate an Anomaly Measure (*AM*) as

$$AM^{\varepsilon}(p_k = i) = SMM(\varepsilon_{p_k=i}) - SMM(\varepsilon_0) = SMM(\varepsilon_{p_k=i}) - SMM_0^{\varepsilon} = SC_{\varepsilon_{pk=i}} - SC_0 \quad (2)$$

where AM^{ε} is the anomaly measure for ecological model parameters $p_k = i$, *SC* is the statistical complexity, $\varepsilon_{p_k=i}$ are the ε-machines for $p_k = i$, and ε_0 is the ε-machine for the default setting. Clearly, *AM* measures the departure of the dynamical behaviour of the ecological model from the "default" dynamics for different input parameters.

Figure 6 shows an application of this approach to a simple case in which we study *K=2* parameters; in particular we analyse the maximum P growth rate represented as the ratio *a/b* (between 0 and 3, Y axis) versus the respiration rate of Z (X axis), which is a colour version of plate *a* in Figure 4. In this case, since the default baseline value for the statistical complexity SMM_0^{ε} is zero, we have $AM^{\varepsilon}(p_k = i) = SC_{\varepsilon_{pk=i}}$.

Unlike Figure 5, the plot in Figure 6 is not binary, rather more than 2 values of the statistical complexity are found, as displayed by the different levels of gray in the image. Even in this case, we show the 3D delayed coordinate representations of the time series corresponding to seven points on the plot. The time series corresponding to the grey areas in the plot are very similar to the limit cycles in Figure 5 and indeed have statistical complexity equal to 1. However, we can see a small area in the centre of the plot (in white) which displays higher statistical complexity. Notice that this area of high complexity is outside the rage of plate *a* in EB99 Figure 4. The 3D delayed coordinate representations corresponding to this location also indicate a limit cycle, however in this case more than 2 states are responsible for this cycle. This is due to the fact that the oscillations in the time series are not as regular as in the previous case and consequently more than 2 states are present in the ε-machines. In this case the statistical complexity is approximately 4, which means that 4 bits of information are needed to carry out an optimum prediction; knowing the binary value of the time series at a location is not enough to predict the next state.

Before proceeding with our enquiry, it is worth noticing that the 3D delayed coordinate representations in Figures 5 and 6 have a markedly different 'spread'. Steady state plots, obviously, are characterised by a single point, while statistical complexity equal to one corresponds to limit cycles of larger amplitude than those for higher statistical complexity. We may thus ask whether some simpler statistical measure, like variance of the time series for example, would be able to achieve a similar classification. The first problem we would encounter, should we decide to test a discriminator based on variance, is that this would require fine tuning; for example we would need to decide, a priori, what threshold differentiates the different limit cycles. More importantly, the statistical complexity extracts more information from a

time series than the simple variance, since it analyses the way the samples follow one another in the time series, that is their dynamical evolution, rather than a mere departure from a mean, in which time information is lost. The importance of this difference is clarified with the help of Figure 7. On the left hand side, we see the delayed coordinate plot of a steady state time series to which has been added white noise with maximum amplitude of 0.15 units. On the right hand side, we see a limit cycle time series, with statistical complexity equal to 1, to which we imposed external forcing and added white noise with maximum amplitude of 0.05 units. As a result, the 2 time series are characterised by the same variance, as shown by the spread of points in the delayed coordinate plots. However, their statistical complexity varies considerably, being zero for the left hand size time series (which is obviously random) and 5.5 for the time series on the right hand side.

To summarise this section, we can say that the AM^{ε} can detect the locations of main bifurcations in the EB99 model under different initial conditions and parameters, and that it is not necessary to specify a priori which dynamical features should be analysed in order to detect changes in dynamical behaviour. This suggests a positive answer to the first question at the end of the section "Statistical complexity of ecological models". We now turn to the second question: "Can this discriminatory power define a rough partition into areas of different dynamical behaviour?"

Complexity Map

The plots in Figures 5 and 6 represent fairly dense samplings of two 2D sections of the NPZ model parameter space. Ideally, we would like to sample the entire high dimensional space in a similar fashion. The 'curse of dimensionality' makes this approach computationally infeasible even for relatively low dimension models. In Boschetti (2004) and Boschetti et al (2002) we have explored the use of the stochastic sampling inherent in some numerical optimisation techniques in order to visualise a rough mapping of a dynamical problem parameter space. Here we propose a similar approach:

1) we cast the calculation of the maximum statistical complexity of a model *M* ($C_M^{\max}$) into a numerical optimisation problem in which we seek to maximise the measure in equation 2 (see also Appendix C).
2) We do this by using a number of stochastic search algorithms (Genetic Algorithm, Swarm Optimisation, Direct Method, see Appendix C for details) and multiple runs for each algorithm.
3) Each of the stochastic search algorithms needs to run the ecological model and CSSR iteratively (see Appendix C). For a single call *k* of the ecological modelling, we store the point P_k and the statistical complexity SC_k, for $k = 1...K$, where *K* in the total number of function calls. Notice that P_k is a vector of dimension D_p, where D_p is the dimensionality of the ecological model parameter space.
4) After all searches are completed, we combine the results. The maximum value of statistical complexity so found defines $C_M^{\max}$.

5) We also obtain a matrix *P* of dimensions $D_p * K$, which represents our sampling of the model parameter space.
6) We visualise the sampling of the D_p dimensional model space *P* via a Self Organised Map (SOM, see Kohonen, 2001). A SOM maps vectors in a high-dimension 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-dimension space. A SOM gives a rough idea of the high dimension space structure as well as of the clusters in the data (see Appendix D for details).
7) By analysing the SOM we attempt to determine, visually, whether domains of different dynamical behaviour are present in the parameter space and which parameters affect the different dynamical behaviours the most.

We use the same NPZ ecological model employed in the previous tests. We limit our parameter space to 6 dimensions by analysing the parameters marked with an asterisk in Table 1. After running a number of numerical inversions as described above and combining the results, we obtain *K=25000* samples of the parameter space. The highest value of the statistical complexity found is *3.67* bits; this is the value we assign to $C_M^{\max}$. We thus feed the *6*25000* matrix *P* to the SOM to obtain a rough map of the parameter space. Figure 8 displays the SOM u-matrix which is a measure of the average distance between grid points in the 2D SOM. It is important to understand that the X and Y axis of the SOM do not carry any physical meaning. The 2D image should merely be seen as an area over which we map the points from the original 6D space, ordering them in such a way that topological relations are maintained as well as possible (Kohonen, 2001). In the SOM u-matrix, blue maps small distances, which should be interpreted as mapping points which lay close to one another in the original 6 dimension parameter space. They thus correspond to clusters in the original data set. Red maps large distances, that is, points far away from one another in the original 6D space. These represent 'ridges' dividing clusters. Figure 8 suggests the presence of roughly 4 clusters:

1) cluster 4 is divided from the rest of the map by a main, almost vertical, ridge;
2) clusters 1 and 2, which are the main clusters found by the SOM, are characterised by a fairly large and almost flat surface;
3) and a group of clusters which, for convenience of description, we grouped as cluster 3, roughly parallel to the main ridge.

In the following we limit our discussion to these 4 main features. The inaccuracies and distortions inherent in mapping a high dimensional space to 2D, in our opinion, do not allow reliable analysis of finer details in the SOM.

The samples used to build the SOM come from our parameter space search. The purpose of the search was to find areas of high complexity values, so our search was inherently biased towards areas of high complexity. Since it is reasonable to expect that clusters correspond to densely sampled areas, we may also suspect that clusters correspond to areas of high complexity. Similarly, we may suspect that ridges correspond to areas of low complexity. This is confirmed by Figure 9. Here we can

see the statistical complexity values mapped *over* the SOM[1]. The 4 clusters discussed above are clearly characterised by high value of statistical complexity (red) and are separated by areas of low statistical complexity (blue). The areas of low statistical complexity appear more irregular and scattered, but this is most likely the result in the bias in the parameter space sample, as explained above. This bias needs to be kept in mind when we analyse all SOM maps. Indeed, the predominance of red areas in Figure 9 may erroneously suggest that the vast majority of the NPZ parameter space is characterised by high statistical complexity. This is probably incorrect: the predominance of high statistical complexity is the result of the (necessary) bias in the search. Were we able to sample the parameter space more uniformly, we would obtain a more reliable picture of the extent of high statistical complexity areas in the problem.

Nevertheless, a number of important conclusions can be drawn. Figure 10 shows the values of each of the 6 parameters at each location over the SOM map. For each plate, blue and red map the minimum and maximum allowed values, respectively, as per Table 1. They show how each dimension has been distorted in order to accommodate the point in 2D. They tell us what contribution each dimension gives to the clusters under analysis and thus to the NPZ model statistical complexity. Of note are the following observations:

1) parameters *k* and *alpha* (plates *a* and *f*) appear to peak roughly where the ridges are located in Figure 8 and where some of the low statistical complexity areas are located in Figure 9. It is thus reasonable to suggest that low values for *k* and *alpha* generate high statistical complexity and high values generate low statistical complexity. This seems to be party confirmed by the plots in Figure 4.
2) Parameters *r* and *s* (plates *c* and *d*) show wide range of variation in correspondence to the high statistical complexity areas. This suggests that high statistical complexity values can be obtained for very different values of these parameters. This is also confirmed by the plots in Figure 4. However, this does *not* mean that statistical complexity is insensitive to *r* and *s*. Not all values of *r* and *s* result in high statistical complexity, rather suitable combinations of these values and of values of other parameters are necessary. Notice the high correlation between areas of roughly constant values for *r* and *s* and the clusters in Figure 8.
3) Parameter *N0* (plate *e*) seems to be mostly responsible for the presence of the large pseudo perpendicular ridge in Figure 8. Indeed all high values of N0 are placed at the right hand side of the ridge (in correspondence with cluster 4) and medium and low values corresponding to the other clusters.
4) Finally, plate *b* shows that large areas of high statistical complexity can be obtained mostly for medium or medium-high values of parameter *d* and that high statistical complexity for extreme values of *d* are very rare. Notice also how its trend is roughly perpendicular[2] to the main ridge and to cluster 4 in Figure 8, which probably is the reason why cluster 4 is much smaller in size.

[1] Notice that the statistical complexity has not been used to build the SOM. Here we simple map its values over the topology reconstructed by SOM by using information on the sample of the 6D parameter space.

[2] Although the term perpendicular in a SOM needs to be treated with care because of the high distortions in the map.

Discussion

Complexity is not currently a well defined concept, either in the ecological modelling or in computer science or mathematics. Several definitions are available (extensive references can be found on line at http://cscs.umich.edu/~crshalizi/notebooks/complexity-measures.html and http://bruce.edmonds.name/combib) and different users may perceive complexity differently, depending on the problem at hand. The method we propose is based on a fairly rough sampling of a potentially very high-dimension space, with the use of models which may require heavy computation. Also, we use a set of tools/algorithms (ecological model, CSSR, search algorithms, symbolisation procedure, SOM) most of which are still at a research stage, which implies that the overall procedure can not be stronger that the weakest of these tools. It is thus clear that the approach we propose needs to be seen as a first step towards a definition of complexity of ecological models and further work is necessary to improve and assess its potential. Some of the most relevant issues are:

1) The characterisation of the complexity of an ecological model should not be seen strictly as a criterion to choose which model to use, but rather as a criterion to *disregard unsuitable models*. This difference is important. We do not suggest that matching the value of the complexity between model and data would indicate a correct fit. In the procedure we describe, both the measured and modelled data go through a chain of processes (symbolisation, CSSR, entropy calculation) and the resulting statistical complexity should be seen as the collapse of the combination of these non-linear operators (acting on high dimensional vectors) into a single (0D) number. As with other simple statistical measures (fractal dimension, Lyapunov exponent, etc) we can not expect to fully capture complex information into a single number. What we are suggesting is that a model which is *not* able to generate sufficient complexity will *not* be able to capture complex structures in the data. Conversely, a model which is able to produce enough complexity *may or may not* be able to reproduce the structures we are interested in and consequently may or may not be appropriate for studying a specific data set.
2) Models of completely different physical/ecological processes (a chaotic oscillator, human heart pulsation and fish population fluctuations) may very well have similar complexity, but clearly are not equally suited to analysing the same data set. Our analysis assumes that the user has chosen an ecological model which is realistic and suitable for studying the process at hand.
3) It could be argued that some delay-coordinate embedding technique (Takens, 1981, Kantz and Schreiber, 1999) could be used to establish the dimensionality of a time series and somehow relate this to the optimal model dimensionality. This could enable us to find a measure of complexity which is applicable to both data and model and which may appear to be more closely related to model 'size'; the technique we discussed here does belong to the delay-coordinate embedding family and can be seen as an attempt to discretise the state space in order to simplify the detection of the state transitions (for a nice discussion see Ray, 2004, p. 1118). However, no simple relation exists

between the dimensionality of a model (determined by the ecological control parameters) and the embedding dimension of a time series. It could be quite difficult, therefore, to derive a criterion that helps reduce the number of input parameters as a function of the embedding dimensions.

4) Reliably sampling a high dimensional space is beyond current computational tools. Models controlled by thousands input parameters may simply not be suited to the analysis we propose. For smaller size problems, we are less pessimistic for a number of reasons. First, the parameter space of 'real world' problems is generally fairly smooth, or at least much smoother than the perversely complex surfaces often employed for search algorithms benchmark tests (More et al, 1981, see also http://www.geatbx.com/docu/index.html). An interesting analysis of this subject can be found in (Cheeseman et al., 1991) and our fairly extensive experience seems to confirm this (Boschetti and Moresi, 2001, Wijns et al, 2003). Second, including the user in the evaluation of the model output can further simplify the solution surface (Boschetti, 2005, Takagi, personal communication). The pattern analysis capability of the human brain is able to detect and process far more detail and many more structures than can be collapsed into the single fitness value that search algorithms use, whereby providing them with more information to improve the search. Also, search algorithms can be designed in such a way as to employ higher dimension measures of fitness and so perform reverse mapping between data space and model space, which also speeds up the parameter space exploration considerably (Boschetti, 2005b). Finally, we should consider that the characterisation of a model complexity map needs to be done only once and then stored for future use. In practise, an ecological model could be set up in such a way that every time it runs (for whatever purpose), the location of the point *P* in the parameter space and the statistical complexity are stored in an ever increasing data base, thereby potentially improving our parameter space sampling and the resulting understanding of the model complexity over time.
5) A Self Organised Map is not the only available method to visualise the high dimensional complexity space. Of particular interest are mapping methods such as LLE (Roweis and Saul, 2000) and HLLE (Donoho and Grimes, 2003), which allow both for forward and inverse mapping between spaces of different dimensionality and for a more accurate projection of the 2D complexity maps back into the original parameter space. In the present paper the SOM has been used because of its better numerical stability but alternative avenues are worth exploring.
6) The SOM is basically a clustering algorithm and in principle the analysis in Figures 8-10 could be performed automatically in order to recover a partition of the parameter space into clusters. We prefer to include the user into this step; first, because the most delicate component of every clustering algorithm is the detection of the suitable number of clusters; and second because we believe that the familiarisation of the user with the complexity map is probably the single most important outcome of the overall process. In our view, the ultimate aim of a modelling exercise is the user's understanding of the problem at hand and the development of a conceptual model of the problem which can be communicated to others (the numerical algorithms simply being tools needed to simplify this process). We believe that the complexity map may be of great help to this understanding.

7) At present the CSSR algorithm works on time series with fairly strict requirements: a) it must be symbolised b) the symbolisation should involve a fairly small alphabet (coarse discretisation) for computational reasons, c) it must be sampled at regular time intervals, d) it needs to be fully sampled (missing data need to be processed prior to the analysis). Also (as for all time series analysis tools) long time series are needed (the length depends on the data at hand, but, roughly, in the order of thousands of points). This currently imposes limits on the applicability of the method. However, time series analysis is a field in rapid development and we can envisage further improvements in the near future. Further work to address the specific requirements of ecological data may also be needed. Among these, we can envisage the parallel use of different data sets. This, in principle, is already possible, by employing some clever symbolisation schemes, but improvement in this area would also be beneficial.
8) A natural question is whether this technique could be extended to spatial data. The definition of statistical complexity we use, as well as the method employed in the CSSR algorithm, are based on the concept of *causal* states and causality implicitly involves the existence of a time progression (here the word causal is taken from automata theory and we circumvent deeper philosophical implications). Shalizi and Shalizi (2003a,b) show how the statistical complexity of a time series of 2D spatial data can be calculated (basically a 3D data set in which one dimension is time). For static 2D data, some conceptual modifications are required as discussed in Feldman and Crutchfield (2003) where a possible approach is proposed.
9) Is a measure of information (in bits) representative of ecological complexity? Would other measures, somehow related to model 'size', be more 'ecologically' intuitive? These are reasonable questions to ask, especially since our ultimate aim (as discussed above) is to involve the user in the process and we envisage the complexity maps in Figures 8-10 as a tool helping the user to better understand a model. We avoid overarching philosophical views according to which information is Nature's fundamental currency and that information processing is what life is about. However, we do subscribe to the view that predicting environmental behaviour, and consequently building a model of it, is an essential feature of all living agents (for a nice discussion see Crutchfield, 1994). A model's complexity defined in terms of information processing is thus somehow related to the way a living being deals with the very same processes that the model is intended to mimic and is thus less abstract than may seem at first. For a related discussion of the role of complexity in evolving agents see Adami (2002).
10) Irrespective of the rationale of the previous point, it is important to notice that *effectively* the measures we employ in the present paper are relative, not absolute. For a pragmatic application of the method, it matters not whether the statistical complexity accurately measures the amount of memory required to make a optimal prediction, rather it matters how the time series complexity compares to the model complexity, and how different is the complexity of the model in different areas of the parameter space. It is this difference, rather than the exact value of the statical complexity, which we believe has relevance in the study of ecological model complexity.

Conclusions

We have outlined an approach to mapping the complexity of the behaviour of ecological models. We have applied this approach to a simple, well-known ecological model for which there are published results, against which we have compared our calculations. For such a system, conventional bifurcation analysis techniques are sufficient to map the range of dynamical behaviours of the model (eg. Edwards & Brindley used LOCBIF and Auto, and other packages include MATCONT, Dhooge et al 2003) while the method we presented has the potential to deal with more difficult modelling scenarios. In testing this approach on the NPZ model, we have confirmed that the technique is capable of mapping the dynamic behaviour possible in the model.

Our approach is different from conventional bifurcation analysis in the following ways:

1. there's no need to prescribe criteria for distinguishing dynamic regimes;
2. the estimated values of statistical complexity can be compared with values estimated from different models or from data;
3. it holds the promise of being applicable to systems influenced by both randomness and deterministic dynamics; and
4. the techniques making up the approach can be applied to observations, as well as model output.

Ecologists have long wrestled with the question of how to interpret and model variability in their time series. A key question is "what are the relative roles of randomness, external forcing and nonlinear internal deterministic interactions?" Statistical complexity is a powerful measure that offers the hope of bringing further insights to this and related questions. We argue that if it could be applied to ecological time series from observations and ecosystem models, we'd have the opportunity to better judge model-data consistency, to more effectively explore sensitivity of model dynamic behaviour to underlying assumptions and to be better able to detect possible regime shifts in observed ecological time series.

References

Adami, C., 2002, What is complexity? Bioessays 24:1085-94.

Bertello G., Arduin P, Boschetti. F., and Weatherley D., 2005, "First Experiments in the application of Computational Mechanics to the analysis of seismic time series", Geophysical Journal International (submitted).

Bollt, E.M., Stanford, T., Lai, Y.C. and Zyczkowski, K., 2001, What Symbolic Dynamics Do We Get with a Misplaced Partition? On the Validity of Threshold Crossings Analysis of Chaotic Time-Series. *Physica D* **154**: 259-286.

Boschetti, F., M. Dentith, R. List, Inversion of seismic refraction data using Genetic Algorithms, 1996, Geophysics, 1715-1727.

Boschetti, F., 2005, Controlling and investigating Cellular Automata behaviour via interactive inversion and visualization of search space, New Generation Computing,

Special Issues on Intertactive Evolutionary Computation, Vol.23, No.2, February 2005.

Boschetti F., 2005, A Local Linear Embedding Module For Evolutionary Computation Optimisation, Journal of Euristics, (submitted).

Boschetti F. and Moresi L., 2001, Interactive Inversion in Geosciences, Geophysics, 64, 1226-1235.

Boschetti, F., Wijns, C. and Moresi, L., 2002, "Effective exploration and visualisation of geological parameter space", Geochemistry, Geophysics, Geosystems (G Cubed), 4(10), 1086).

Chaitin, G., 1969, On The Length Of Programs For Computing Finite Binary Sequences: Statistical Considerations", Journal of the ACM 16, pp. 145-159.

Chaitin, G., 1982, Gödel's theorem and information, Internat. J. Theoret. Phys. 22, 941–954.

P. Cheeseman, R. Kanefsky, and W. Taylor, 1991, Where the really hard problems are, IJCAI, 163--169.

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

Crutchfield, J. P.. 1994, The Calculi of Emergence: Computation, Dynamics, and Induction, Physica D 75, 11-54.

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

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

Donoho, D. L., Grimes, C. E., 2003, Hessian eigenmaps: locally linear embedding techniques for highdimensional data. Proceedings of the National Academy of Arts and Sciences, 100, 5591–5596.

Dhooge A., Govaerts W., Kuznetsov Y.A., 2003, MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs. *ACM Transactions on mathematical software* 29 (2): 141-164

Edwards, A.M. & Brindley, J., 1999, Zooplankton mortality and the dynamical behaviour of plankton population models.
Bulletin of Mathematical Biology, 61(2):303-339

Ellner, S.P. and Turchin, P. (2005) When can noise induce chaos and why does it matter: a critique. Oikos 111 (3): 620-631

Fath, B. D.; Cabezas, H., and Pawlowski, C. W. (2003). Regime Changes in Ecological Systems: an Information Theory Approach. *Journal of Theoretical Biology*. 222(4):517-530.

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

Gershenfeld, N.; Schoner, B., and Metois, E. (1999) Cluster-Weighted Modelling for Time-Series Analysis. *Nature.* 397(6717):329-332.

Hsieh, C. H.; Glaser, S. M.; Lucas, A. J., and Sugihara, G. (2005) Distinguishing Random Environmental Fluctuations From Ecological Catastrophes for the North Pacific Ocean. *Nature.* 435(7040):336-340.

Jones, D.R., C.D. Perttunen, and B.E. Stuckman, 1993, "Lipschitzian Optimization Without the Lipschitz Constant". Journal of Optimization Theory and Application, 79(1):157-181.

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

Kennel, M.B. and Buhl, M., 2003, Estimating good discrete partitions from observed data: symbolic false nearest neighbors. *Physical Review Letters* **91**: 084102.

Kohonen, T., 2001, Self-organizing maps: 3rd Edition, New York, Springer, Series in Information Sciences, v. 30, 501 p.

Li, M., and Vitányi, P., 1997, An Introduction to Kolmogorov Complexity and Its Applications, Springer.

Mayer, A. L.; Pawlowski, C. W., and Cabezas, H. (2006). Fisher Information and Dynamic Regime Changes in Ecological Systems. *Ecological Modelling.* 195(1-2):72-82.

Moré, J.J., Garbow, B.S. and Hillstrom, K.E., 1981, Testing Unconstrained Optimization Software, ACM Trans. Math. Software 7, 17-41.

Mouser C., and Dunn S., 2004, Comparing Genetic Algorithms and Particle Swarm Optimisation for an Inverse Problem Exercise, The 12th Biennial Computational Techniques and Applications Conference, Melbourne, Australia (submitted).

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

Roweis, S., Saul, L., 2000, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, 290, 2323--2326.

Shalizi, C. R. and Crutchfield, J. P., 2001, Computational Mechanics: Pattern and Prediction, Structure and Simplicity. *Journal of Statistical Physics* **104**: 819--881.

Shalizi, C. R. and Shalizi, K. L., 2003, Quantifying Self-Organization in Cyclic Cellular Automata, in Noise in Complex Systems and Stochastic Dynamics, Lutz Schimansky-Geier and Derek Abbott and Alexander Neiman and Christian Van den Broeck, Proceedings of SPIE, vol 5114, Bellingham, Washington.

Shalizi, C. R. and Shalizi, K. L., 2003, Optimal Nonlinear Prediction of Random Fields on Networks, Discrete Mathematics and Theoretical Computer Science, vol AB(DMCS), 11-30.

Shalizi, C. R. and Shalizi, K. L., 2004, Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences. In “Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference", = "Arlington, Virginia, Max Chickering and Joseph Halpern, AUAI Press.p.504-511 (http://arxiv.org/abs/cs.LG/0406011) ,

Shalizi, C. R. and Shalizi, K. L., 2003, Optimal Nonlinear Prediction of Random Fields on Networks, Discrete Mathematics and Theoretical Computer Science, vol AB(DMCS), 11-30.

F. Takens, 1981, Detecting strange attractors in turbulence. Lecture Notes in Mathematics, 366-381

Wijns, C., T. Poulet, F. Boschetti, C. Dyt, C.M Griffiths, 2003, Interactive inverse methodology applied to stratigraphic forward modelling, in: A. Curtis and R. Wood (Eds.), Geological Prior Information, Geol. Soc. of London Special Publication.

Wolpert, David H. and William G. Macready, 1997, No Free Lunch Theorems for Optimization, IEEE Transactions on Evolutionary Computation, 1, 67-82.

Appendix A

Causal-State Splitting Reconstruction (CSSR) algorithm. Here we summarise the CSSR algorithm. More details about the algorithm implementation, together with examples of simple binary processes can be found in Shalizi et al (2004a), while a theoretical analysis, containing proof of several theorems related to the minimum properties of the reconstruction can be found in Shalizi and Crutchfield (2001).

Let's suppose we want to analyse a sequence of N discrete values $S_i, i = 1...N$, where S_i can take any of k values in an alphabet A, representing measurements taken at discrete time steps from a stochastic process. At any time i, we can divide the series S into two 'half'-series, $\overleftarrow{S}$ and $\vec{S}$, where $\overleftarrow{S} = ..S_{i-2}S_{i-1}S_i$, stepping backward in time, represents the 'past' and $\vec{S} = S_{i+1}S_{i+2}S_{i+3}...$, progressing forward in time, represents the 'future'. Following the same notation as in Shalizi et al (2004a), we call $\overleftarrow{S}^L$ and $\vec{S}^L$ histories of length L symbols in the past and in the future, respectively. Also, we call s (and s^L) specific instances of histories belonging to S. Now, let's suppose we scan the series S, looking for occurrences of the history $\overleftarrow{s}$, and we store the symbol $\vec{S}^1$ seen as 'future' in each instance. We can calculate $P(\vec{S}^1|\overleftarrow{s})$, that is, the probabilities of occurrence of any of the k symbols in the alphabet A, given the history s, and we call the vector containing these probabilities the *morph* of $\overleftarrow{s}$. We can then define a *causal state* as the collection of all histories $\overleftarrow{s}$ with the same morph (i.e., histories which share the same probabilistic future). More formally, histories $\overleftarrow{s}_1$ and $\overleftarrow{s}_2$ belong to the same *causal state* if $P(\vec{S}^1|\overleftarrow{s}_1) = P(\vec{S}^1|\overleftarrow{s}_2)$.

Given the above definition, the purpose of the CSSR algorithm is to reconstruct the set of the *causal states* of the process and the transition probabilities between the causal states. Following the nomenclature used in Shalizi et al (2004a), the combination of causal states and their transition probabilities is called a ε-machine.

The CSSR algorithm can be divided into a number of steps:

1) we start from the null hypothesis that the process is independent and identically distributed. In this case each of the k symbols $a \in A$ is equally likely at each time step and only one causal state is necessary to model the process: the morph of the state is the k-length vector of components $1/k$.
2) we select a maximum history length *max_L* for our analysis. This is the length of the longest history with which we scan the series *S.* For histories of length = 1… *max_L,* we scan the series S, storing both the histories found and their futures. Given an history $\overleftarrow{s}$, its morph is trivially obtained by calculating $P(a|\overleftarrow{s}) = v(a,\overleftarrow{s})/v(\overleftarrow{s})$, for each $a \in A$, where $v(\overleftarrow{s})$ is the number of occurrences of the history $\overleftarrow{s}$ and $v(a,\overleftarrow{s})$ is the number of occurrences of the symbol a given the history $\overleftarrow{s}$.

3) We group histories with similar morphs into the same causal states. This involves three steps: a) first, we need a measure for morph similarity. Real time series are characterised by both the presence of noise and by finite data extent. Consequently we need to relax the requirement of exactly matching morphs $P(\vec{S}^1|\overleftarrow{s}_1) = P(\vec{S}^1|\overleftarrow{s}_2)$ to an approximation $P(\vec{S}^1|\overleftarrow{s}_1) \approx P(\vec{S}^1|\overleftarrow{s}_2)$. In particular we accept $\left|P(\vec{S}^1|\overleftarrow{s}_1) - P(\vec{S}^1|\overleftarrow{s}_2)\right| < \varepsilon$, where ε is a user defined parameter; b) Second, we define the morph for a state as the average of the morph of all histories in that state; c) finally, in order to ensure the reconstruction of a minimum number of states, new states are created only when a history is found which can not match any existent causal state. That is, for each history, we look for an existent state with similar morph and we create a new state only when we can not find any. After these steps, we have a collection of states, grouping all histories found in the time series *S* according to the similarity between their morphs.

4) As a last step, we want to make sure that transitions between states, on a given symbol, are unique. That is, we want to make sure that, given any *history* in a state, and a next symbol $a \in A$, the next *state* is uniquely determined. Notice the difference between the occurrence of the *next symbol*, which is stochastic and measured by the morph, and the transition to the *next state*, *given a next symbol*, which we want to be deterministic. In order to do this, for each state, we store the next state transitions for each history, that is, we store into what state a history goes after seeing a certain symbol. This is also represented by a vector of length *k*, containing, as elements, the next state on each symbol. If a state has two histories whose next state transition vectors are different, we split the state and create a new one.

Once the ε-machine is reconstructed, we can use an approach proposed by Crutchfield and Young (1989) and define as *statistical complexity* of the process the entropy of the ε-machine itself.

Appendix B

Symbolisation. As we have seen, the CSSR algorithm requires symbolized data, that is, each datum has to take one of *k* values in an finite alphabet *A*. However, many ecological measurements, ideally, represent values on a continuous range. Even accounting for finite resolution, the number for values allowed by most ecological models defies the concept of a limited alphabet (most symbolic time series analysis applies to binary series). A means to discretise the real valued measurements is thus needed. This requires two decisions: first, how many symbols to use, and, second, how to assign symbols to numerical ranges in the data. No standard method is available in the literature to tackle either problem. For a nice review of symbolisation methods and their application we refer the reader to Daw et al, (2003). By far, the most widespread approach in the literature is to use a binary alphabet. The simplest avenue to assign the symbol to each datum (and also the most widely used) is to ensure that the *k* symbols occur evenly in the symbolised time series. For a binary discretisation, this amounts to choosing the median on the data as the separation criterion and to bin the data accordingly. However Bollt et al (2001) have shown that non optimal symbolisations can b obtained as result of this approach. A more

sophisticated approach has been proposed in Kennel and Buhl (2003), whereby consistency in the delayed coordinate representations of the original and the symbolised series is sought. In this work we have tested both Kennel and Buhl (2003) approach and the simple histogram discretisation, with no noticeable difference in outcome. Because Kennel and Buhl's method involve a considerable computational effort (it performs a stochastic search in a non-linear high dimensional space), the histogram discretisation was used in all the tests discussed below.

Appendix C

Numerical optimisation and Search Algorithms. Our aim for searching a model parameter space is twofold: first we would like to determine the maximum statistical complexity a model can generate; second, we would like to sample the parameter space as uniformly as possible in order to detect areas of different dynamical behaviour. These two different aims are usually defined as exploration and exploitation in the numerical optimisation literature, where *exploration* refers to coarsely sampling ever new areas of the parameter space to ensure no crucial features are missed, and *exploitation* refers to finely sample the areas where the current best solution lies in order to further improve it. Local search algorithms focus only on the exploitation and are likely to succeed only in problems with single global solution. Non linear problems, like the ones we address, often are characterised by multiple local minima and global searches with balance exploration and exploitation are needed. In this work we employed three search algorithms:

1) a real coded Genetic Algorithm (GA) (see Davis, 1991 for general introduction to real coded GAs and Boschetti et al, 1996 for the specific GA implementation used in this work),which performs a stochastic sampling of the parameter space by mimicking the behaviour of biological evolution;
2) A Particle Swarm Optimisation algorithms (PSO), (see Mouser and Dunn, 2004, for details about the specific PSO implementation used here) which also performs a stochastic sampling of the parameter space, this time by mimicking the strategies employed by insect colonies to search for food; and
3) The Direct method (Jones et al, 1993); which performs a deterministic sampling by subdividing the parameter space into hypercubes of ever decreasing sizes depending on the local characteristic of the solution surface.

All the above methods are today commonly used in the optimisation of different numerical problems and have been widely tested on real world problems. The rationale for employing three different algorithms is that no algorithm is known to outperform all other algorithms on general problems (for an extreme discussion of this issue see Wolpert and Macready, 1997). Here we ran each algorithm several times independently. For each run we store the points where we sampled the parameter space as well as the statistical complexity. We then combined all this information into a single data file.

Appendix D

Self Organised Map (SOM). A self-organised map is a transformation of high-dimensional (nD) data into a lower-dimensional (usually 2D) plot. It is a classification

algorithm which separates all the input data into clusters according to similarity. Topology is preserved, *i.e.* two points lying close to one another in the higher dimensional space also do so in the 2D space. All SOM visualization is based on the u-matrix (Figure 8), which is composed of two different types of nodes: *data nodes* and *distance nodes*. The data nodes represent the high dimensional data points. Adjacent data nodes reflect points in nD space which are similar. The distance nodes connect the data nodes, and give an indication of the relative distance between them. The SOM algorithm assigns the input data vectors to particular data nodes; the distance nodes are then coloured or shaded according to the magnitude of the distance between adjacent nodes. Data nodes are also shaded, according to the average of the surrounding distance nodes, to produce a more continuous map. All parameter values are usually normalised before any calculations are made.

SOM has been extensively employed in recent years in both scientific and engineering applications in order to visualise high dimensional data and highlight data structure and clustering. Its full potential can probably be best appreciated after acknowledging the difficulties inherent in the visualisation of high dimensional data. The SOM plots we show in this work have been obtained with the use of the Matlab™ SOM Toolbox, written by Juha Vesanto. More details about SOM, as well as the specific SOM implementation used in this work, can be obtained at http://www.cis.hut.fi/projects/somtoolbox. We also refer the reader to a SOM web tutorial at http://scitec.uwichill.edu.bb/cmp/online/p21h/lecture11/lect11a.htm.

Appendix E

NPZ model. Here we briefly describe the nutrient-phytoplankton-zooplankton (NPZ) model used in this study. For more details we refer the reader to Edwards and Brindley (1999). The specific equations used are:

$$\begin{aligned}
\frac{dN}{dt} &= -\frac{N}{e+N}\frac{a}{b+cP}P + rP + \frac{\beta\lambda P^2}{\mu^2+P^2}Z + \gamma qZ + k(N_0 - N) \\
\frac{dP}{dt} &= \frac{N}{e+N}\frac{a}{b+cP}P - rP - \frac{\lambda P^2}{\mu^2+P^2}Z - (s+k)P \qquad \text{(F1)} \\
\frac{dZ}{dt} &= \frac{\alpha\lambda P^2}{\mu^2+P^2}Z - qZ
\end{aligned}$$

where N, P and Z are nutrient, phytoplankton and zooplankton respectively, with units of gCm^{-3}. The model parameters, units and ranges are described in Table 1. The parameter ranges have been selected by Edwards and Brindley after extensive literature review (see Edwards and Brindley, 1996, pp 351-353).

Table 1. The parameters, units, default values and ranges used in the NPZ model. The asterisks refer to the 6 parameters used in the 6D parameter space search.

Parameter	Symbol	Default value	Reported range
a/b gives maximum P growth rate	a	0.2 m^{-1} day^{-1}	0.07–0.28
Light attenuation by water	b	0.2 m^{-1}	0.04–0.2
P self-shading coefficient	c	0.4 m^2 (g C)$^{-1}$	0.3–1.2
Half-saturation constant for N uptake	e	0.03 g C m^{-3}	0.02–0.15
Cross-thermocline exchange rate*	k	0.05 day^{-1}	0.0008–0.13

Higher predation on *Z**	q	0.075 day^{-1}	0.015–0.150
P respiration rate*	r	0.15 day^{-1}	0.05–0.15
P sinking loss rate*	s	0.04 day^{-1}	0.032–0.08
N concentration below mixed layer*	N_0	0.6 g C m^{-3}	0.1–2.0
Z growth efficiency*	$\acute{a}$	0.25	0.2–0.5
Z excretion fraction	β	0.33	0.33–0.8
Regeneration of *Z* predation excretion	γ	0.5	0.5–0.9
Maximum *Z* grazing rate	λ	0.6 day^{-1}	0.6–1.4
Z grazing half-saturation coefficient	μ	0.035 g C m^{-3}	0.02–0.1

Figures

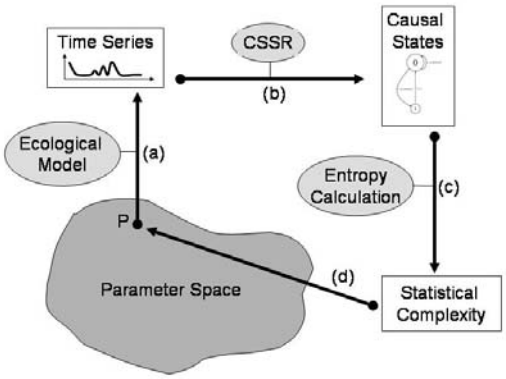

Figure 1. Graphic representation of the method we employ to define the statistical complexity of an ecological model at a point P in the parameter space. (a) We generate a time series by running the ecological model with initial conditions and parameters defined by P; (b) we reconstruct the ε-machine from the time series via the CSSR algorithm; (c) we calculate the statistical complexity of the ε-machine; (d) we define this as the statistical complexity of the ecological model at point P.

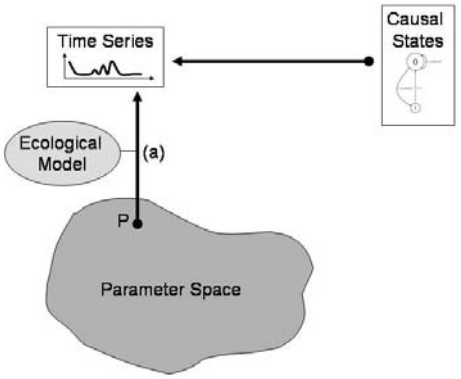

Figure 2. Visual representation of the informal equivalence between the ecological model at point P and the ε-machine reconstructed from the time series it generates. Both the ε-machine and the model at point P are able to (statistically) reconstruct the time series.

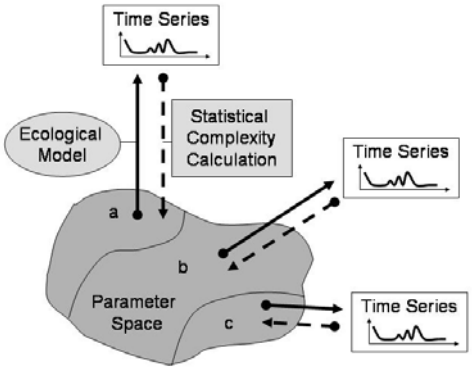

Figure 3. Schematic representation of the method we employ to discriminate between areas in the parameter space characterised by different dynamical behaviour. The parameter space is sampled extensively and a value of statistical complexity is assigned to each sample point via the method described in Figure 1. A clustering algorithm is then used to segment the space is areas of similar dynamical properties.

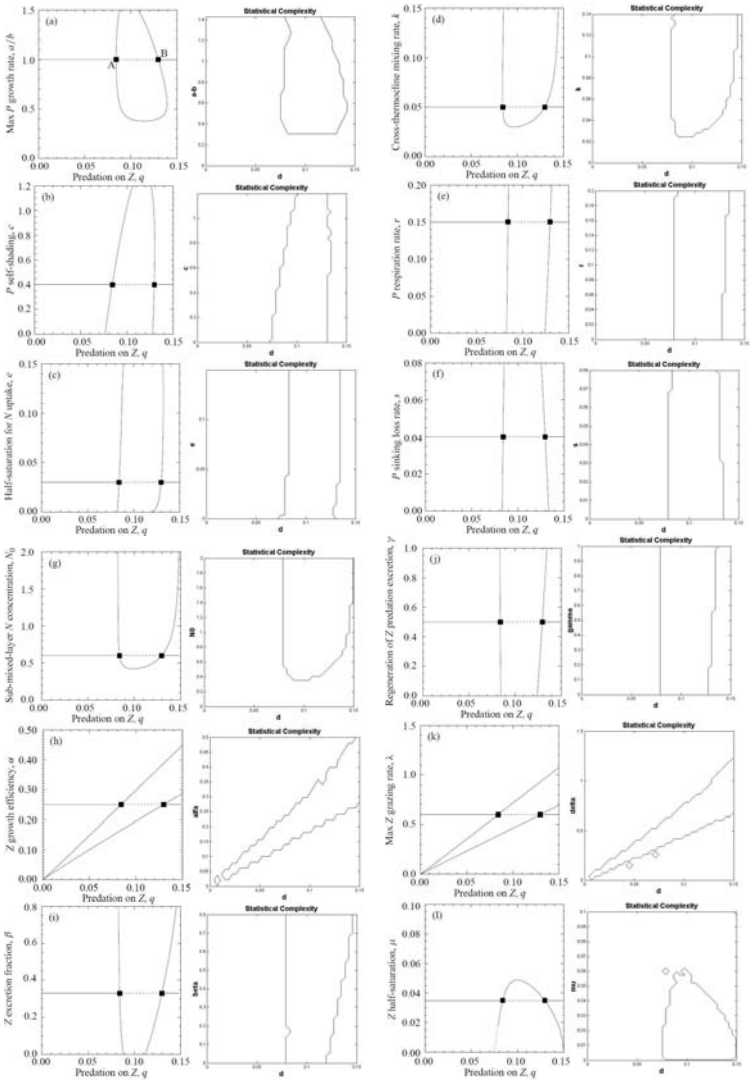

Figure 4. Reproduction of the entire EB99 Figure 4. In all plots, the X axis represents parameter *d* (predation on Z, in the range between 0 and 1.5). In different plots the Y axis represents the other model parameters listed in Table 1. The bifurcations detected by the statistical complexity (right hand side of each plate) coincide with those estimated analytically in EB99 (left hand side of each plate).

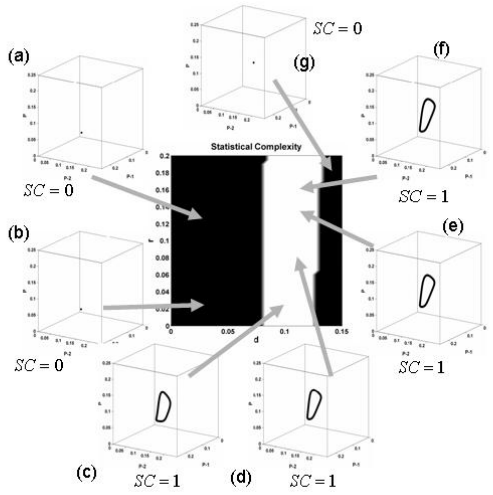

Figure 5. Statistical complexity as a function of the predation rate on Z (X axis) versus the respiration rate of Z (Y axis); white maps high values. The statistical complexity is equal to zero everywhere on the plot, except inside the bifurcation area, where its value is one. For different locations on the map we show the 3D delayed coordinate plots of the corresponding time series. Notice the different dynamics in the delayed coordinate plots for areas of different statistical complexity.

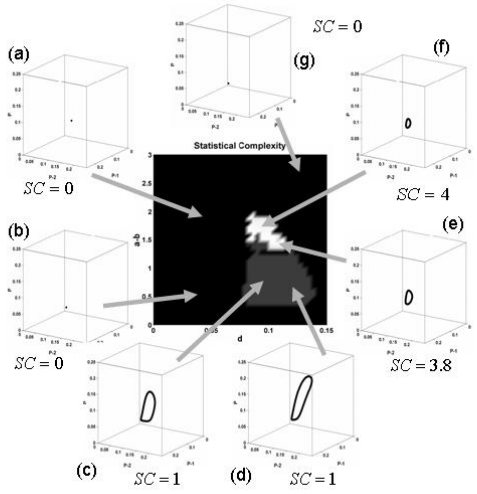

Figure 6. Statistical complexity as a function of the predation rate on Z (X axis) versus the maximum P growth rate represented as the ratio *a/b* (Y axis); white maps high values. More than 2 values of the statistical complexity are found, as displayed by the different levels of gray in the image. For different locations on the map we show the 3D delayed coordinate plots of the corresponding time series. Notice the

different dynamics in the delayed coordinate plots for areas of different statistical complexity.

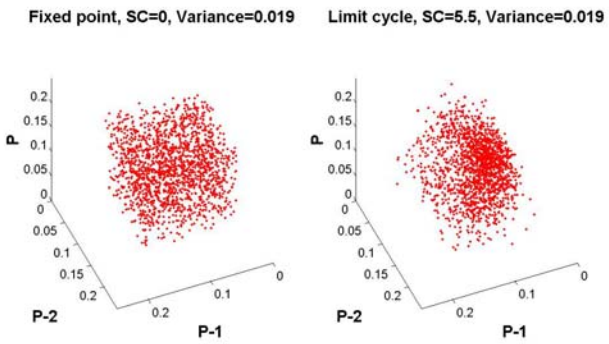

Figure 7. Left: delayed coordinate plot of a steady state time series to which white noise with maximum amplitude of 0.15 units has been added. Right: limit cycle time series, with statistical complexity equal to 1, to which we imposed external forcing and added white noise with maximum amplitude of 0.05 units. The 2 time series have the same variance but their statistical complexity differs, being zero for the left hand time series and 5.5 for the time series on the right hand side.

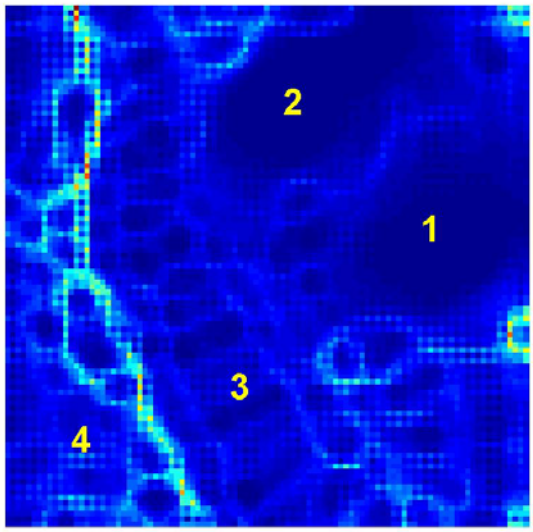

Figure 8. Self Organised Map U-Matrix representing the topology of the sampling of the parameter space as obtained by our search algorithms. Blue maps small distances, corresponding to clusters in the original data set. Red maps large distances, that is ridges separating the clusters.

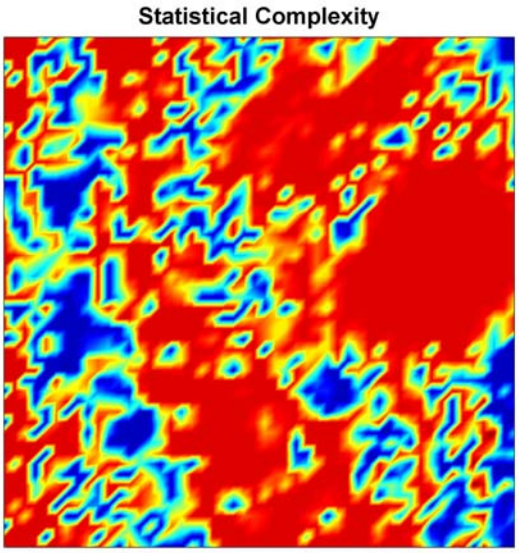

Figure 9. The statistical complexity mapped *over* the SOM. Red and blue map high and low statistical complexity, respectively. The 4 clusters in Figure 8 are clearly characterised by high values of statistical complexity.

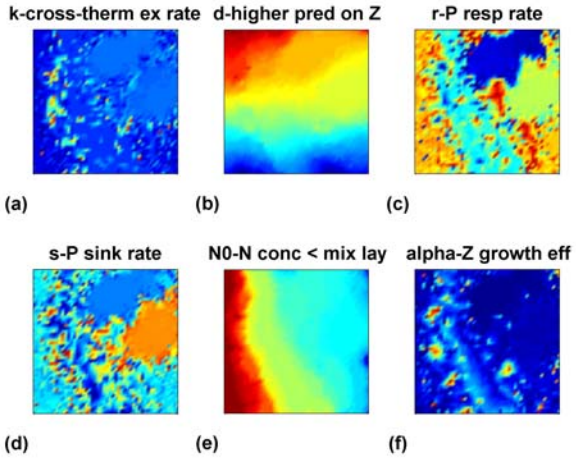

Figure 10. Values of the 6 parameters in the SOM map. Each dimension has been distorted in order to accommodate the points in 2D. These plates can be used to discriminate the contribution each dimension gives to the clusters in Figure 8.