Modular Interdependency in Complex Dynamical Systems

Richard A. Watson

Dept. of Organismic and Evolutionary Biology
Harvard University, Cambridge, MA 02138, USA.
e-mail: rwatson@oeb.harvard.edu

Abstract

Hierarchical modularity is a familiar characteristic of a large class of natural dynamical systems. A normal interpretation of modularity is that interactions between subsystems are sparse compared to interactions within subsystems and this leads some to assume that the interactions between modules are largely unimportant. In the evolution of complex systems, if a system can be decomposed into modules that are basically independent then evolving such a system is greatly aided. However, this interpretation of modularity is over-simplistic. Specifically, although modules may be sparsely connected, dynamical properties of modules may be strongly interdependent, and if functional properties depend on dynamical properties then interactions between modules may be critically important. If inter-module dependencies are significant then resolving the dependencies between modules is problematic, and simple evolutionary processes can be inadequate. However, 'compositional' evolutionary mechanisms, such as sexual recombination between diverse lineages, lateral transfer, or symbiogenesis, are better able to resolve module interdependencies. In this brief discussion paper, we overview our recent work related to modularity and compositional evolution, and in addition, describe its relation to the dynamical properties of hierarchical systems, and provide some discussion of some well-known examples.

Concepts of modularity and evolvability

Simon (1969) discusses a broad ranging set of systems – from business organizations to biological systems – that exhibit what he calls the property of being 'nearly decomposable'. Simon suggests that interactions between modules in such systems are weak and that the subsystems behave nearly independently. This fits with the common intuition that modularity is synonymous with the property that inter-module interactions are somehow less important than intra-module interactions. In fact, it is often assumed that a modular system cannot have strong significant inter-module interactions *by definition*.

However, this view that inter-module interactions are not very important is sometimes overly simplistic when applied to complex dynamical systems. This over-simplification might arise from a tendency to conflate *structural* modularity with the *functional* behaviour of the system. Specifically, it is often useful to describe a complex system using a graph that represents the interactions between components, and modularity is taken to mean that inter-module edges are sparser than intra-module edges (e.g. Ravasz et. al, 2002, Hartwell et. al, 1999). However, this interpretation of modularity is merely a structural description of the graph of interactions, and for complex dynamical systems a structural description of interconnectedness is not sufficient to determine the behavioural independence of one module from another. In general, the dynamical dependence or independence of a module from another module is related to how changes in the state of one module affect changes in the state of the other. The structural interconnectedness of modules tells us something about the likelihood of *immediate* effects between one module and another, but it is not necessarily indicative of the extent of consequent state changes over time. In principle, one module may be strongly and non-linearly sensitive to small state changes in another module despite being sparsely connected. Other important dynamical properties such as the number, location, and stability of attractors may also be affected despite sparse connections. The exact consequence of interactions between modules is dependent on the exact nature of the systems involved – but in general, it is not correct to assume that sparsely connected dynamical systems have only small effects on one another's dynamical properties.

If the functional behaviour of one module may be strongly dependent on the state of other modules then we should not allow modularity to imply that inter-module interactions are unimportant, despite the scarcity of interconnections.

The over-simplified view of modularity also presents problems when we apply the concept *hierarchically*: The notion that inter-module interactions are less important than intra-module interactions tends to imply that interdependencies between higher-level modules become less and less important, as we will discuss. Although it may well be the case that inter-module connections are sparser than intra-module connections at every level, this need not suggest that interactions between higher-level components are, in any sense, unimportant. In contrast, it seems to be the case in natural systems that interactions between large subsystems are sometimes just as important as interactions between small subsystems. For example, is inter-cellular communication weaker or less significant than intra-cellular communication in living organisms? Our general intuition that cells are meaningful entities likely comes from the property of encapsulating internal processes – but we would not generally suppose that (the remaining) inter-cellular interactions are unimportant or incidental to the functioning of the cell.

To understand these issues properly it is necessary to develop a description of modularity that accommodates the possibility of strong inter-module dependencies. To this end, the first part of this paper provides a simple modular dynamical system and examines the stability of its attractors. In particular, we show that although it exhibits clear structural modularity, the location of the most stable state configuration for one module can change completely (i.e. all sub-components change state) in response to state changes in another module. Building from this example system, and contrasting with examples from Simon, we are able to clarify and resolve the conceptual problems with hierarchical modularity.

The second part of this paper addresses the effect of modularity on evolvability. In the evolution of complex systems, modularity can have a significant impact. It is a familiar intuition that if a complex system has several subparts that are essentially independent then evolving such a system is much easier than evolving a monolithic, or non-decomposable, system. If a system is structurally *and* functionally modular, then it may well be the case that its evolution is trivial. However, it is generally assumed that to the extent that a system has strong interdependencies between modules, it is in fact not modular, and therefore difficult to evolve. We find that it is indeed the case that systems that have strong dependencies between modules can be difficult to evolve. But in fact, when systems are structurally modular, behavioural interdependencies can be resolved in certain non-mainstream evolutionary scenarios. Specifically, the normal hill-climbing, or *accretive*, conception of evolution, such as mutation and selection, or sexual recombination in converged populations, is unable to resolve inter-module dependencies, but *compositional* mechanisms, such as sexual recombination between diverse lineages in a subdivided population, lateral transfer, or symbiogenesis, are better able to resolve inter-module dependencies.

In this paper, these two parts briefly overview our recent work related to modularity and compositional evolution (Watson 2002), and describe its relation to the dynamical properties of hierarchical systems. Whereas the prior work provided a system with modular interdependency and proofs about the evolvability of such systems, in this paper we have connected these models and results with the structural and dynamic modularity of dynamical systems. We have also taken the opportunity to discuss Simon's examples of modular and hierarchical systems at some length. These provide a very valuable starting point for illustrating the concepts of interest and focussing discussion.

Part 1: Modular Interdependency

In this part of the paper we clarify the ideas introduced above with the use of a simple example dynamical system. We show that structural modularity does not imply isolation, or near independence, of the dynamical behaviour of modules. Thus we show that there is a meaningful sense in which a system may be modular and yet have strong interdependencies between modules. We use this to contrast and clarify some concepts introduced by Simon (1969).

An example modular dynamical system

Let us consider the domain of gene regulation networks as an example dynamical system. We suppose for simplicity, that each gene may be in one of two states: "high", meaning highly expressed, or otherwise, "low". Let the future state of each gene be determined by some function of the states of the genes that regulate its expression. (In general, this describes a 'Random Boolean Network', Kauffman 1993). We can use a graph to represent the connectivity of regulation activity where nodes represent genes, and edges represent regulation interactions. In general, edges may be directed, but, for our purposes here, we can reduce unnecessary complications by asserting that all interactions are bi-directional - i.e. an edge between A and B means that A regulates B, and B regulates A. It also suits our purposes to assume that all nodes have a self-recurrent connection so that a node's future state is a function of its own state at the previous time step.

To define a modular network, we modify a fully connected network such that the interactions within particular subsets of genes will be stronger than the interactions between genes in different subsets. To represent this we will simply use multiple edges between genes (since we do not really wish to distinguish between numerous and strong interactions). Figure 1, thus shows an example system that exhibits a clear, two-module structure.

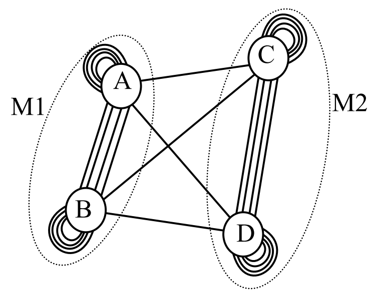

Figure 1: An example system of 4 genes, A, B, C, and D, arranged in two modules, M1 and M2, where intra-module (and self-recurrent) interactions are of strength 4, and inter-module interactions are of strength 1.

This graph describes the structural properties of our system. But as we have noted, we wish to examine its dynamical properties and the structural properties alone are insufficient to make conclusions about dynamical properties. For illustration we use a simple 'voting' style discrete-time update rule. Specifically, the probability of a gene taking a given state in the next time step will be equal to the proportion of regulating connections from genes that are themselves in that state at the current time step. i.e.

Let "high"=1 and "low"=0 then,

$$P(A_{t+1}=1)=\tfrac{1}{K_A}\sum_{i=1}^{k_A} Bi_t \;,$$

$$P(A_{t+1}=0)=1-P(A_{t+1}=1). \qquad \text{Eq.1.}$$

where A_t is the state of gene A at time t, k_A is the number of edges connecting to A, B_i is the state of the i^{th} regulating gene connected to A. This update rule is popular in models of physical dynamical systems such as Ising models.

We have now defined the structure and the dynamics of the system. For some purposes, the structural modularity of a system may be sufficient, but if we are interested in functional modularity that may be determined by dynamical properties such as the stability and location of attractors, then we must provide further analysis. The long term (rather than immediate) effects of changes in one module on state changes in another module can be characterised by changes in the attractors of a system. In general, if the attractors of a subsystem or module are unaffected by state changes in another module then this is an important form of independence. At the opposite extreme, if we find that the attractors of a subsystem are completely different this indicates an important form of inter-module dependency.

By way of example, let us analyse our example system by examining which configurations of M1 are most stable, and how these configurations differ with state changes in M2. For these purposes it is sufficient to use a simple measure of stability: specifically, let us define the *stability* of a network, or subnetwork, as the probability that no gene in the network changes state. (This definition of stability closely parallels the *free-energy* measure of a configuration in an Ising model.[1]) The stability of a network is therefore the product of the stability for each gene in the network. For example, in a system of two genes, A and B, where the probability of A remaining in the same state is S(A), and the probability of B remaining in the same state is S(B), the stability of the AB system is simply S(A)S(B).

Clearly, since there are inter-module edges, the stability of a given configuration is sensitive to the state of genes in the other module, and the configuration which is most stable may also differ.

Let us write the states of the 4 genes in our system as A, B, C, and D; and we will write the stability of a subset of variables *x*, given the state of a subset of variables *y*, as S(*x*, *y*): e.g. the stability of M1, given the state of M2, is written as S(AB,CD). Given the intra-module symmetries in our example system, the effect of 01 and 10 are the same, so we will list only the combinations 00, 01, and 11 for each module. Then, for our example system, using Eq.1:

A B C D	S(A,BCD)	S(B,ACD)	S(AB,CD)
0 0 0 0	10/10	10/10	1
0 1 0 0	6/10	4/10	0.24
1 1 0 0	8/10	8/10	0.64
0 0 0 1	9/10	9/10	0.81
0 1 0 1	5/10	5/10	0.25
1 1 0 1	9/10	9/10	0.81
0 0 1 1	8/10	8/10	0.64
0 1 1 1	4/10	6/10	0.24
1 1 1 1	10/10	10/10	1

We immediately notice that the configuration of M1 that is most stable is strongly sensitive to the configuration of states in M2. Specifically, the most stable configuration of M1 when CD=00 is AB=00, and the most stable configuration of M1 when CD=11 is AB=11. The systems' symmetries provide corresponding stabilities for M2.

This example system is therefore sufficient to illustrate the following: 1) Although we have a system that is clearly modular in one respect, it also has strong inter-module dependencies in another respect: Specifically, it is *structurally* modular in the sense that it has stronger (more numerous) intra-module dependencies than inter-module dependencies, but it has strong *functional* inter-module dependencies in the sense that a simple dynamical property – the most stable configuration for a module – is strongly dependent on the state of other modules. 2) Following from this, if the function of a system depends on dynamical properties, then we cannot assume that module interactions are unimportant even if the structural modularity of the system is unambiguous.

From this example, one might conclude that (despite the apparent structural modularity) this system is in fact not modular in respect to the property of 'most stable configuration'. But, it should also be noted that even though the configuration of a module that is most stable is strongly dependent on the state of the other module, there is some degree of independence in this respect also. Specifically, we notice that although the most stable configuration of states may be either 00 or 11 it is always one of these and never 01 or 10 regardless of the state of the other module. This means that there is something we know about the property of interest, the most stable configuration, that is *independent* of inter-module interactions.

From these observations we arrive at a means to quantify the independence of a module, given a property of

[1] Our observations will be unsurprising for those readers familiar with Ising models and the stability of their dynamics, but working through the example in some detail is useful for providing clarity.

interest. Specifically, given some property that identifies particular configurations, like the configuration that is most stable, we can assess what we know about the identity of such states that is independent of the state of other modules. In this example, there are four possible configurations for a module, C=4. Accordingly, a meaningful definition of non-decomposability, given this property of the system, is that we know nothing about which of the four possible states is the most stable state. This will be the case when, for every configuration of M1, there is some configuration of M2 (or the remainder of the system) that would make that configuration of M1 the most stable. However, in our example system, this is not the case. There are only two configurations that could be maximally stable, C'=2; regardless of the state of the other module. In our modular system 1<C'<C. In a non-decomposable system, C'=C.

In general, we will say that when C'<C the system is *decomposable*. But note that there is a special case of decomposability where C'=1 which means that the configuration of interest is always the same regardless of the state of the remainder of the system. In this case, the module is fully independent in the property of interest, a case which we call *separable*. In the general case where a system is decomposable but not separable, i.e. 1<C'<C, as in our example, we say the system exhibits *modular interdependency.*

This manner of quantifying the dependence or independence of one module from another, given a property of interest, is detailed in (Watson 2002). In that work, the property of interest is the configuration that maximises the fitness of the system. This understanding of modularity clearly allows meaningful decomposability when C'<C, and yet still allows the possibility of important interactions between modules when C'>1.

Example Modular Systems and Hierarchy

Simon (1969) describes nearly decomposable systems as those where "the short-run behaviour of each of the component subsystems is approximately independent of the short-run behaviour of the other components" and "in the long run the behaviour of any one of the components depends in only an aggregate way on the behaviour of the other components". This characterization, especially the latter emphasis on aggregate effects, is closely related to the notion of encapsulation that is used in the above definition of modular interdependency (see Watson 2002). But, although we suggest that this characterisation of modular systems is useful, and in fact both properties are true of the example system described above,[2] Simon gives examples of modular systems where modules are separable, i.e. C'=1. Here we detail Simon's examples for the purposes of illustrating how easy it is to conflate modularity with the notion that inter-module dependencies have no functional significance.

In discussing the expected time for a complex system to evolve, Simon uses an analogy based on finding the combination for a combination lock on a safe. Specifically, suppose a lock has 10 dials each with 100 positions – by blind trial-and-error we would expect to need on average half of 100^{10} guesses to open the lock. Simon, contrasts this with a 'defective' lock where a click can be heard when the dial is at the correct setting. In this case, we would expect to require on average about half of 100*10 guesses to open the lock, because opening the lock is merely 10 independent repetitions of finding the correct setting for each dial. Clearly, this describes a modular system where the property of interest for a dial, i.e. 'is it the correct setting to open the lock', is entirely independent of the setting of other dials and can be determined uniquely, i.e. C'=1.

To highlight the consequences of assuming separable modules, let us now take a look at what a hierarchical lock might be. Let us suppose that an 'hierarchical-super-lock' has 10 groups of 10 dials each where a group of dials makes a big click when all 10 dials are set correctly. If the individual dials are defective as before – what is the expected number of guesses to open the lock? We can see that it is merely 100*10*10. Note that this is exactly the same as would be expected for a defective 'flat-super-lock' being just a lock that has 100 dials. In other words, when each dial is independent, subsequent grouping of dials is redundant and does not change the combinatorics involved.

As an example of a completely decomposable system, the defective lock scenario is fine. But from Simon's descriptions, it is not at all clear how to modify the lock analogy to make dials 'nearly-decomposable'.[3]

The hierarchical inadequacy of the lock example is no less present in Simon's famed watchmakers' parable. One watchmaker assembles watches made of 1000 components. The other assembles watches made of 10 modules, each of which is comprised of 10 sub-modules, each of which is comprised of 10 components. The difference in the success of the two watchmakers arises in their robustness to interruptions. When the first is interrupted he has to start again from scratch and looses on average 500 assembly steps per interruption. When the second is interrupted he only has to return to the previous stable module and restart from there, loosing on average only 5 assembly steps. This, Simon claims, explains why modular complex systems,

---

2 'Short-run' behaviour can be understood by equating a low stability value with rapid changes. And 'the proportion of ones' is a sufficient 'aggregate effect' to determine the inter-module effect on stability.

3 Kauffman (1993) provides a means to describe arbitrary systems of interdependent variables, and the system we described earlier is similar in some respects to the 'NKC landscape' where subsets of variables have more intra-connections than inter-connections. However, it is not clear that the inter-module dependencies formed from random 'k-wise' interdependencies among variables across modules produces significant inter-module interactions (Watson and Pollack 1999).

through their inherent stability, are more likely to be evolved than non-modular ones.

This is all very well, and this metaphor has the advantage that there is a bit of extra work implied in assembling sub-modules into modules and modules into a complete watch (unlike the lock example where having set each of the dials correctly, there is nothing more to be done to open the lock). But what role is the hierarchy playing here? Suppose a third watchmaker assembles watches that consist of 100 sub-modules of 10 components each. In this case an interruption during the assembly of a sub-module wastes on average 5 assembly steps, and an interruption during the assembly of the 100 sub-modules into a complete watch wastes on average 50 steps. On average the wasted steps for the third watchmaker is a little more than 9 steps.[4] Therefore we compare 500 wasted steps for the non-modular scenario (watchmaker 1), with ≈9 wasted steps in modular but not hierarchical scenario (watchmaker 3), with 5 wasted steps for the hierarchically modular scenario (watchmaker 2). So we see that the additional hierarchical level in Simon's original example adds very little advantage. Why is this? Notice that in the scenario Simon describes the second watchmaker knows the correct assembly for a module (or sub-module) uniquely – i.e. again, C'=1. In other words, although there is some extra work to be done in putting modules together, which makes it seem like an example of a *nearly*-decomposable system rather than a completely-decomposable system; In fact, the problem of finding the correct assembly for each module is still entirely separable from the configuration of every other module.

The lock and watchmakers' examples provide quantitative 'problem-solving' counting arguments that give straightforward analogies for evolvability, and the advantage of decomposition is clear. But, as we have highlighted, the modules in these examples are separable and that makes the combinatorics involved in subsequent hierarchical levels insignificant. These examples also do not incorporate dynamical properties (like short-run and long-run behaviour) that Simon uses to characterise nearly-decomposable systems. In a third example, Simon describes an example system which is dynamical and is intended to be nearly-decomposable not completely decomposable. This is a system of heat exchange in a building that is subdivided into rooms and cubicles. Heat exchange between cubicles within a room is more rapid than heat exchange between cubicles in different rooms. This system exhibits the properties that Simon lists for nearly-decomposable systems: the short-run dynamics of heat exchange in a room is approximately independent of the heat exchange dynamics in other rooms, and the long run dynamics of heat exchange in a room depends (approximately) only on the mean temperature of cubicles in other rooms.

Unfortunately, Simon does not describe what the 'problem' is in this system – i.e. if this is an analogy for a system that might be evolved, we do not know how the temperature configurations relate to fitnesses, or how changes in state relate to combinatorics that reflect evolvability. The natural physics of heat exchange is that all temperatures will reach the mean temperature of the building regardless of any modularity, or otherwise, in the heat exchange parameters. In other words, the temperature configuration space has only one attractor – mean temperature – and whatever the starting configuration of temperatures it is always possible for the mean temperature of a cubicle to move gradually to a more stable state without moving through a less stable state. This is an important distinction form our own example system where there are two stable states, 00 and 11, for a module and these are local optima, isolated from one-another by the less stable states 01 and 10. The absence of this property in the heat exchange example means that there are no 'difficult' dependencies between modules in the task of maximising stability. More specifically, there are no local optima in the 'stability surface', or 'energy function' of the system.

So whereas the first two examples give clear combinatorial arguments but address separable systems, the third example provides a consistently hierarchical structure in its dynamics but lacks a mapping to combinatorial arguments of evolvability. We have discussed these three example systems from Simon in some detail. Our purpose here is not to suggest that Simon is unusually misguided – on the contrary – we address these examples to indicate that even though Simon's characterisation of modularity and hierarchy is extremely useful, and his examples are very well known, it is easy to overlook aspects of these examples that are lacking. We suggest that these shortcomings derive from common confusions about the relationship between structural modularity, which is easy to describe, and significant inter-module dependencies, which are often omitted unnecessarily.

These examples illustrate that when modules are separable it is difficult to make sense of hierarchical modularity. In all Simon's examples there are no important dependencies between modules. In the first two, the position for a dial, or the way to assemble a module of the watch, are unambiguously identifiable and not sensitive in any way to the configuration of other dials or partial watch assemblies. In the heat exchange example, the temperature that maximises stability is sensitive to the temperature of other modules, but it is not difficult to find this temperature since there are no local optima in the energy surface.

To illustrate the distinction between the modular systems that Simon described above and modular systems with strong inter-module dependencies, let us describe small modifications of these examples that introduce strong inter-module dependencies.

[4] There are 100*10 assembly steps for the sub-modules and 100 assembly steps to assemble these into a complete watch: giving a total of 1100 assembly steps for the 'flat modular watch' process. Interruptions waste 5 steps with probability 1000/1100, and waste 50 steps with probability 100/1100. The average number of wasted steps is therefore (10/11*5+1/11*50)=9.09.

For the lock example, suppose that there were two (or more) positions of each dial that produced a click, only one of which was the correct position for opening the lock. In this case, we must search combinations of dial positions to open the lock but the number of combinations is reduced from 100^{10} to 2^{10}. In this case C'=2 which is much less than C=100 but greater than 1 in Simon's description – so the modified system is decomposable but not separable. In the watchmaker's example, we can similarly suppose that for each part (component, sub-module, and module) there are two (or more) stable states (or ways to assemble them) so that the work required at every hierarchical level is a search in combinations of parts. In this case the work required for the non-modular case is 2^{1000}. The work required for the modular but not hierarchical case is $100*2^{10}+2^{100}$. And the work required for the modular multi-level hierarchical case is $100*2^{10}+10*2^{10}+2^{10}$. Thus, because the modules are not separable in this revised scenario, the extra work to complete assemblies at every level of the hierarchy is exponential in the number of parts at that level, so the extra level of modularity provided in the multi-level case provides a significant advantage. Finally, for the heat exchange example, we may suppose that room temperatures are discretised into two or more distinct temperatures (like the "high" and "low" states of our own example or an Ising model). Then it will not necessarily be the case that changes in cubicle temperatures that increase stability are always available. In the discrete system, the combinatorics involved in finding highly stable states is also clear.

By defining a concept of modularity that accommodates strong and significant inter-module dependencies, as used in these modified examples, we both allow a more general notion of modularity, and also permit the construction of hierarchically modular systems where modules are important at all levels. In contrast, when modules are separable, additional hierarchical levels are redundant.

Part 2: Evolvability of systems with modular interdependency

In this second part of the paper we discuss the impact of strong inter-module dependencies on the evolvability of complex systems. As we mentioned already, it is a familiar wisdom that if a system can be decomposed into modules that are basically independent then evolving such a system is easy. This intuition aligns directly with the reasoning that Simon provides in the lock or watchmaker examples. However, it is also 'common wisdom' that when module interdependencies are strong and significant the system is not in fact decomposable and therefore not evolvable. But, in the first part of this paper we have shown that this view is too simplistic. Specifically, a system may be decomposable yet have strong and significant inter-module dependencies. The question addressed in this part of the paper is what impact the possibility of decomposable but not separable systems has on evolvability.

First we discuss the evolvability of systems with modular interdependency under normal mechanisms of evolution that we term 'accretive': That is, adaptive mechanisms that accumulate random changes in genetic material. Then we discuss the evolvability of systems with modular interdependency under some alternate mechanisms we call 'compositional': That is, adaptive mechanisms that involve the exchange of pre-adapted sets or subsets of genetic material between different lineages. We will find that the evolvability of systems under these different classes of mechanisms is quite different.

For these studies we utilise a system with modular interdependency based on our example system described above. To describe a fitness function using the properties of this dynamical system we can simply equate fitness with the stability of a state configuration. As we see in our modification of the watchmakers' example, when modules are not separable, additional hierarchical levels of decomposition are significant. Accordingly, we use a system with several hierarchical levels as defined below. This function, known as 'Hierarchical Equality' (or 'Hierarchical-if-and-only-if'), was first defined in Watson & Pollack 1998. To maintain alignment with previous work, the fitness measure used here sums the size of all modules that have internally equal states (rather than taking a product of module stabilities), but the functions' modularity is structurally the same as that described in our example modular system above. Specifically, it has the same local optima, and the configuration that is most stable for a module in the earlier system is the configuration that is most fit in this function. $g(s_1,\ldots,s_N)$ gives the fitness of a system with states $s_1,\ldots,s_N$.

$$g(s_1,\ldots,s_N)=\begin{cases}1 & ,if \quad N=1\\ Nf(\mathrm{S}_1,\ldots,\mathrm{S}_k)+\sum_{i=1}^{k} g(\mathrm{S}^i) & ,otherwise\end{cases}$$

where S_i is the i^{th} variable of the configuration, S^i is the i^{th} disjoint sub-partition of the variables, e.g. for equal sized sub-modules, $S^i=(s_{1+k(i-1)},\ldots,s_{ki})$, $\boldsymbol{f}$ is the fitness contribution function, defined below. $N=k^H$ where $H\in Z^+$ is the number of hierarchical levels in the system or subsystem, and k is the number of sub-modules per module.

$\boldsymbol{f}(p_1,\ldots p_k)=1$ if ($\exists s \forall i$: $p_i=s$), and 0 otherwise,

where $s\in\{0,1\}$.

This function can be used to describe modular systems with any number of sub-modules per module, and any alphabet of symbols rather than binary. However, it is sufficient to use two equal-sized sub-modules per module and binary states.

For illustration, we list all 4-bit strings with their fitnesses below:

g(0000) = 12	g(1000) = 6
g(0001) = 6	g(1001) = 4
g(0010) = 6	g(1010) = 4
g(0011) = 8	g(1011) = 6
g(0100) = 6	g(1100) = 8
g(0101) = 4	g(1101) = 6
g(0110) = 4	g(1110) = 6
g(0111) = 6	g(1111) = 12

Examination of these fitness values shows that 0011 and 1100 are local optima separated from both 1111 and 0000, which are the global optima, by Hamming distance 2. Thus for the left subset of genes, the most fit configuration is either 00 or 11, but which of these maximises fitness is dependent on the state of the other two variables - i.e. 1<C'<C. As this system is scaled-up through successive hierarchical levels to 8 variables, and 16 variables, etc., the number of local optima increases and the distance of each local optimum to the closest point of higher fitness also scales-up. However, as with our example system, the most fit configuration is always one of a small number of possibilities (C'=2), but which of these is most fit is dependent on the configuration of the rest of the system. These properties, as before, mean that the system is decomposable but the modules are not separable.

Evolvability under normal (accretive) evolution

We can get some understanding of the evolvability of this kind of system by examining a particular cross-section through the fitness landscape – Figure 2.

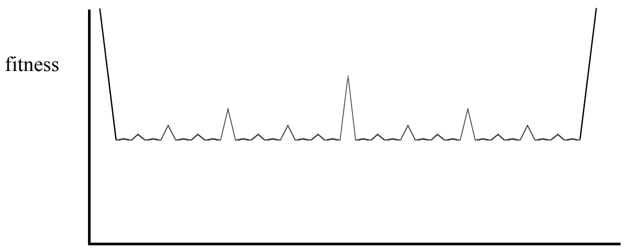

Figure 2: A cross-section through the fitness landscape corresponding to the stability function or energy surface of the modular dynamical system.

This curve runs through points from one global optimum to the other making single point-mutation changes (see Watson 2002). The separation of peaks on this curve accurately reflects the Hamming distance from a point of this fitness to the nearest point of higher fitness.

This curve shows that the landscape is highly rugged and has many local optima creating broad fitness saddles, and includes 'irreducibly complex' adaptations that cannot be reached by a succession of gradually changing proto-systems (Watson 2002). All these characteristics are usually associated with evolutionary difficulty, and accordingly, it should not be surprising that simple optimisation processes are unable to find high-fitness points in this landscape. In fact, all mutation-based methods cannot be guaranteed to succeed in time less than exponential in N (the number of genes) regardless of the mutation rate used. So a mutation hill-climber, an evolutionary algorithm using only mutation, and simulated annealing all fail to resolve the interdependencies between modules. In short, accretive variation mechanisms that cannot manipulate modules as wholes cannot resolve the interdependencies between modules to escape from sub-optimal configurations of the system.

An evolutionary process using sexual recombination and fitness proportionate selection is also unable to resolve inter-module dependencies because the population quickly converges around the best configuration found thus far and this prevents useful variation from crossover.

These observations are detailed in Watson 2002 and we provide simple proofs that the time to resolve inter-module dependencies is exponential in the size of the system, N, for accretive mechanisms. This seems to confirm the commonplace intuition that if modules have strong interdependencies then they are not evolvable and may as well be understood as being non-decomposable. However, in the next section we argue otherwise.

Evolvability under compositional evolution

In recent work, (Watson 2002), we introduced the term 'compositional evolution' to describe evolutionary processes using mechanisms that combine together systems or subsystems of genetic material that have been semi-independently pre-adapted in parallel in different lineages. Examples in nature include:

- Normal mechanisms of sexual recombination (under particular conditions of population diversity and genetic linkage);
- Mechanisms of interspecific combination such as horizontal gene transfer (Doolittle 2000), or 'symbiotic encapsulation' including endosymbiosis (Margulis 1970) and other mechanisms that encapsulate a group of simple entities into a complex entity at a higher level of organisation, as exhibited in several of the major transitions in evolution (Maynard Smith and Szathmary 1995, Michod 1999).

Compositional evolution stands in contrast to the normal 'accretive' view of evolutionary processes involving the accumulation of random variations – i.e. where the new genetic material introduced by variations has not been pre-adapted elsewhere as a set. Assuming that large random variations are less likely to be fitness positive than small random variations, the 'accretive' view supports the familiar assumption of gradual evolutionary change, i.e. 'successive slight modifications' (Darwin 1859).

From an adaptationist perspective, the important characteristic of compositional mechanisms is that they allow the potential for complex entities to be assembled from a number of simpler entities evolved in parallel. For example, the eukaryote cell (and accordingly all plants and animals) originated from the union of more than one prokaryote cell (Margulis 1970). Variation acting in the space of possible assemblies of extant entities is clearly a different variation space from random modifications in genetic material whether small or large. This compositional variation can provide better evolvability than accretive variation when pre-adapted genetic material 'relocated' from one lineage to another, has a better chance of producing a fitness positive change than does random genetic material. Several factors influence the likelihood of this: not least, the availability of a variation mechanism that manipulates appropriate (non-arbitrary) subsets of genetic material – for sexual recombination this places requirements on the ordering of genes on the chromosome (Watson 2002).

Another requirement, of interest to us here, is the contextual (in)dependence of the fitness benefit of a set or subset of genetic material – in other words, the modularity of the genetic interactions. This latter condition is met by systems that are strongly decomposable even if they are not separable, and accordingly, our previous work has been able to show that there are conditions under which compositional mechanisms can resolve the interdependencies between modules. In compositional processes, different lineages can maintain different high-fitness configurations to a module, and then by exchanging subsets of genetic material between lineages, search combinations of different module configurations to resolve interdependencies between them. Proofs are available to show that the expected time for sexual recombination, or an abstract symbiogenesis model, to resolve module interdependencies is, under certain assumptions, polynomial in the size of the system (Watson 2002).

Conclusions

In the first part of this paper we discussed different types of modularity and different examples of modular systems. We showed that structural modularity does not necessarily imply independence of dynamical behaviour. From this we conclude that important functional dependencies may exist between modules. We used Simon's examples to show that it is easy to assume modular systems have no important dependencies between modules, but we are also able to show that this need not be the case. We also argued that modular systems without significant module interactions result in degenerate forms of hierarchies where successive levels of modularity have little or no effect. However, modular systems that are decomposable but not separable, which we call systems with modular interdependency, can form hierarchical systems where all levels of organisation are significant.

In the second part of this paper we discussed the evolvability of systems with significant inter-module dependencies. As might be expected, such systems can be problematic for evolution. However, these systems are evolvable under certain evolutionary scenarios which we call compositional evolution.

These observations and results cause us to think differently about the adaptive potential of non-mainstream evolutionary mechanisms - especially if biological systems are hierarchically modular (Ravasz et. al 2002). More generally, these observations assist us in clarifying concepts of modularity, structural modularity and functional modularity, and accompanying assumptions, especially with respect to hierarchical dynamical systems.

Acknowledgements

Thanks to Alex Platt, Josh Cherry, Daniel Weinreich and the reviewers for helpful discussion and comments.

References

Darwin, C, 1859, *On the Origin of Species by Means of Natural Selection*, John Murray, London.

Doolittle, WF, 2000, "Uprooting the Tree of Life", *Scientific American* 282: pp.90-95.

Hartwell, L, Hopfield, J, Leibler, S, Murray, A, 1999, "From Molecular to Modular Cell Biology", *Nature 402* (supp): p. C47-C52.

Kauffman, SA, 1993, *The Origins of Order*, Oxford University Press.

Margulis, L, 1970, *Origin of eukaryotic cells*. Yale University Press, New Haven.

Maynard Smith, JM & Szathmary, E, 1995 *The Major Transitions in Evolution*, WH Freeman.

Michod, RE, 1999, *Darwinian Dynamics, Evolutionary Transitions in Fitness and Individuality*. Princeton Univ. Press.

Ravasz, E, Somera, AL, Mongru, DA, Oltvai, ZN and Barabasi, A-L, 2002, "Hierarchical Organization of Modularity in Metabolic Networks", *Science*, *297*:1551-1555.

Simon, HA, 1969, *The Sciences of the Artificial,* Cambridge, MA. MIT Press.

Watson, RA, 2002, *Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis*, PhD Dissertation, May 2002, Computer Science Dept., Brandeis University, MA, USA.

Watson, RA, Hornby, GS, & Pollack, JB, 1998, "Modeling Building block Interdependency", *Procs. of Parallel Problem Solving From Nature V*, Eds. Eiben, Back, Schoenauer, Schweffel: Springer. pp.97-106.

Watson, RA, & Pollack, JB, 1999, "Hierarchically Consistent Test Problems for Genetic Algorithms", In Angeline, Michalewicz, Schoenauer, Yao and Zalzala, eds., *Procs. of 1999 Congress on Evolutionary Computation, CEC-99*. IEEE Press, pp.1406-1413.

Watson, RA, & Pollack, JB, 2002, "A Computational Model of Symbiotic Composition in Evolutionary Transitions", *BioSystems*, to appear.