Self-assembly, modularity, and physical complexity

S. E. Ahnert,[1] I. G. Johnston,[2] T. M. A. Fink,[3,4,5] J. P. K. Doye,[6] and A. A. Louis[2]

[1] *Theory of Condensed Matter, Cavendish Laboratory, University of Cambridge, JJ Thomson Avenue, Cambridge CB3 0HE, United Kingdom*

[2] *Rudolf Peierls Centre for Theoretical Physics, University of Oxford, 1 Keble Road, Oxford OX1 3NP, United Kingdom*

[3] *Institut Curie, INSERM U900, CNRS UMR144, 26 rue d'Ulm, Paris F-75248, France*

[4] *Mines ParisTech, Fontainebleau F-77300, France*

[5] *London Institute for Mathematical Sciences, 22 South Audley Street, London W1K 2NY, United Kingdom*

[6] *Physical & Theoretical Chemistry Laboratory, Department of Chemistry, University of Oxford, South Parks Road, Oxford OX1 3QZ, United Kingdom*

(Received 12 October 2009; revised manuscript received 22 April 2010; published 27 August 2010)

We present a quantitative measure of physical complexity, based on the amount of information required to build a given physical structure through self-assembly. Our procedure can be adapted to any given geometry, and thus, to any given type of physical structure that can be divided into building blocks. We illustrate our approach using self-assembling polyominoes, and demonstrate the breadth of its potential applications by quantifying the physical complexity of molecules and protein complexes. This measure is particularly well suited for the detection of symmetry and modularity in the underlying structure, and allows for a quantitative definition of structural modularity. Furthermore we use our approach to show that symmetric and modular structures are favored in biological self-assembly, for example in protein complexes. Lastly, we also introduce the notions of joint, mutual and conditional complexity, which provide a useful quantitative measure of the difference between physical structures.

DOI: 10.1103/PhysRevE.82.026117 PACS number(s): 89.75.Fb, 87.18.−h, 89.70.−a, 33.15.−e

I. ALGORITHMIC COMPLEXITY

More than forty years ago, Kolmogorov [1] and Chaitin [2] laid the foundations of algorithmic information theory, by introducing the concept of algorithmic information content, or Kolmogorov complexity, for a given string of information [3]. This measure of complexity is defined as the length of the shortest possible program on a universal computer (or Turing machine) [4] that will output the string in question. Here, we propose a conceptually analogous measure of the complexity of any connected physical structure. Instead of a universal computer which translates a program into a string of information, we consider a general framework of self-assembly rules, which act together to create a physical object. The "program" now is our set of self-assembly building blocks and rules, the "computer" is given by the physical interactions of the self-assembling building blocks, and the "output" is the final structure. Using this approach we investigate the physical complexity of shapes in two and three dimensions, including polyominoes, molecules and protein complexes. Our work generalizes ideas first explored in [5,6], and opens them up to a wide range of applications. Furthermore, in the context of protein complexes it offers the kind of biological application of information-theoretic concepts demanded in [7].

II. SELF-ASSEMBLY KIT

There are many examples of self-assembling structures in physics, chemistry and biology [8]. These include thin films [9], micelles [10], viruses [11,12], protein complexes [13], and DNA [14–19]. Our aim is to introduce a general framework for the theoretical study of self-assembling structures. This framework can be used to study the properties of real self-assembling systems, but, more generally, it can also be used to measure the physical complexity of any construct that can be divided into building blocks, regardless of whether the structure forms through self-assembly in real life. The exact nature of the self-assembly framework depends on the underlying physical system, but it always contains two basic ingredients: a set of building blocks and a set of rules. We shall call this combination an *assembly kit* S. Each building block i has f_i interfaces, which typically are subject to geometric constraints (depending on the physical system). Attached to each interface j of a given building block i is an integer $\chi_{ij} \in [1, \ldots, c]$. The c possible values of these integers denote different interface types, and we will refer to them as *colors*, to connect with the language of combinatorics. The number of distinct colorings of the building blocks depends entirely on the geometry of the problem. The second ingredient of the assembly kit is the set of rules, which takes the form of an interaction matrix between colors. In the simplest case this matrix is binary, where 1 signifies attraction and 0 signifies no interaction at all. Many more sophisticated interaction matrices involving repulsion and a continuous spectrum of energies are easily imaginable, but here we only consider binary matrices.

For any system of self-assembling particles we need to also specify a model for the actual assembly process. A convenient choice is a model assuming a single nucleus in solution [5], which makes the assumption that each disjoint object has one fixed nucleus building block which is surrounded by a solution containing a freely moving population containing many copies of each type of building block. Each time step (i) a fixed building block, (ii) a site adjacent site to it, (iii) a rotational orientation, and (iv) a building block from the solution are chosen at random, and the new

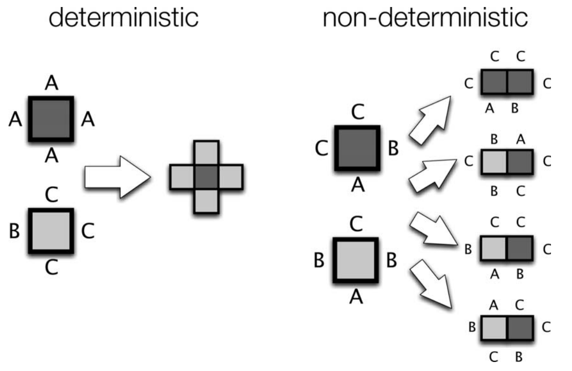


FIG. 1. An example of deterministic and nondeterministic self-assembly kits, using simple 2D lattice structures (polyominoes). In both cases, colors A and B attract each other, but C attracts neither A nor B. No color attracts itself. The kit on the left will always assemble into the cross shape while that on the right will assemble into an irregular cluster, as there are several ways in which the two blocks can attach.

building block becomes fixed to its position if the rules allow it. Note that some assembly kits always assemble into the same shape—these we call "deterministic"—while ones which contain ambiguous rules are "nondeterministic." See Fig. 1 for an example of a deterministic and a nondeterministic self-assembly kit.

III. MINIMUM KIT

Every deterministic assembly kit S_A, which always assembles into a structure A, requires a certain amount of information $I(S_A)$ to describe it in some given formal language. Our aim is to minimize this quantity, since we define the length of the description of the *minimum assembly kit* $\widetilde{S}_A$ as the *complexity* $K(A)$ of structure A:

$$K(A) = I(\widetilde{S}_A) = \min_{S_A} I(S_A),$$

in analogy to the concept of Kolmogorov complexity. Any symmetry or modularity which the structure A contains is very likely to decrease the amount of information required to describe the structure (as the same type of building block can be reused several times) and will therefore be reflected in its minimum assembly kit $\widetilde{S}_A$, and by extension in the value of $K(A)$.

For a deterministic minimum assembly kit, an interaction matrix D (with elements d_{ij}) between a total of c colors, of which c_s self-interact, can be rewritten as:

$$d_{ij} = [1 - (i \bmod 2)]\delta_{i(j+1)} + (i \bmod 2)\,\delta_{i(j-1)},$$

for $i \leq c - c_s$, and $d_{ij} = \delta_{ij}$ otherwise, so that one color always only interacts with one other color. With this constraint in mind, consider an assembly kit S_A consisting of N blocks and c colors, of which c_s self-interact. If we write f_i for the number of faces of block i, and c_i for the number of colors on block i, we can write the information required to store the kit as:

$$I(S_A) = \log_2(c_s + 1) + \sum_{i=1}^{b} (c_i \log_2 c + \log_2 F_i). \qquad (1)$$

The first term, $\log_2(c_s+1)$, is the amount of information required to specify the number of self-interacting colors, necessary to distinguish these from the non-self-interacting colors which may also be present in the kit. The second term consists of a sum, over all building blocks, of the information required to specify the structure of each block. This is represented by two terms. The first of these, $c_i \log_2 c$, measures the information required to describe which c_i colors out of the total of c colors appear on building block i. The second, $\log_2 F_i$, measures the information required to represent the distinct arrangement of colors on block i.

For this last term, the physical structure of the building blocks must be considered. If the interacting parts of a block are geometrically constrained to lie in some particular ordering, such as, for example, the faces of a square tile, we must work with *labeled faces*. If the interacting parts are rearrangeable, so that blocks with differently ordered faces are equivalent, we need to consider *unlabelled faces*. In this paper we will treat the structure of molecules and proteins at a level of resolution where the positions of the chemical bonds are not constrained, thus, leading to unlabelled faces.

For unlabelled faces, it is only necessary to specify the presence of colors on a block. This can be achieved with the representation

$$F_i = \prod_{j=1}^{c_i} k_j^{(i)}, \qquad (2)$$

so that $\log_2 F_i = \Sigma_{j=1}^{c_i} \log_2 k_j^i$, and where the $k_j^{(i)}$ signify the number of times color j occurs on block i. Note that this simplified picture only works under the condition that multiple connections between the same pair of building blocks are prohibited, and that connections are short range. For more complex systems in which these assumptions do not hold, we have to resort to labeled faces.

In the case of labeled faces, we must also specify the ordering of these faces, leading to the following general expression for $F_i = F(c_i, f_i)$:

$$F(c_i,f_i) = \sum_{k_1=1}^{f_i-c_i+1} \sum_{k_2=1}^{f_i-c_i+2-k_1} \cdots \sum_{k_{c_i-1}=1}^{f_i-c_i+(c_i-1)-\Sigma'} \frac{f_i!}{\prod_{m=1}^{c_i} k_m!},$$

where $\Sigma' = \Sigma_{j=1}^{c_i-2} k_j^{(i)}$, and the $k_j^{(i)}$ are defined as above.

IV. APPLICATION TO POLYOMINOES

As a simple example of a self-assembling system with labeled faces, we will consider self-assembling *polyominoes*. A polyomino (also known as a *lattice animal*) is a set of connected sites on a (typically square) lattice [20] (see Fig. 1). These connected sites are our self-assembly building blocks. Every building block has four sides (so that $f_i = 4$ for all i), which are painted with one of c colors. These colors can attract each other or not, as encoded in a $c \times c$ binary interaction matrix. Each distinct way of coloring a building block corresponds to a different building block *type*. We do not regard rotated colorings as distinct, since we can rotate building blocks in the self-assembly process. The geometry of the two-dimensional (2D) lattice gives rise to a particular set of building block colorings in the context of self-assembly. If we have c colors, the total number of such colorings is

$$N_c = (c^4 + c^2 + 2c)/4.$$

These particular colorings are also known as *necklaces*, which can be defined as equivalence classes of strings under rotation. Necklaces are discussed in more detail in Appendix C and [21]. The definition of necklaces used here assumes that the building blocks have a fixed chirality—in other words that the necklaces which the colors form on the building blocks are *fixed* [33].

For polyominoes $F_i = F(c_i) = N'_{c_i}$, where N'_{c_i} is the number of necklaces with *exactly* c_i colors, and is given by

$$N'_{c_i} = N_{c_i} - \sum_{k=1}^{c_i-1} \binom{c_i}{k} N'_k$$

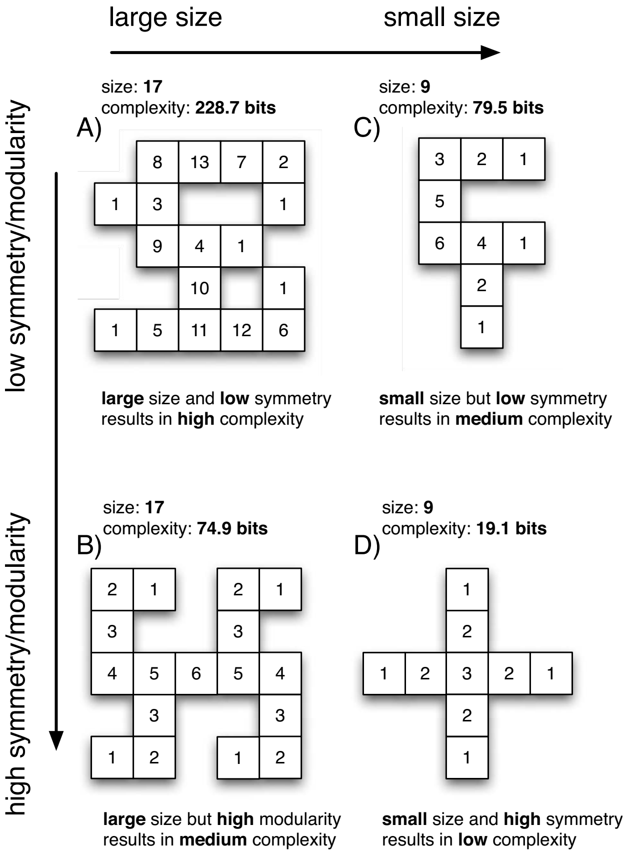
with $N'_1 = 1$. It follows that $N'_2 = 4$, $N'_3 = 9$, and $N'_4 = 6$. As before, the complexity $K(A)$ of polyomino A is the minimum of $I(S_A)$ over all possible assembly kits S_A. Note that Wang tiles [22], and the tile system described in [6] are similar to our framework for the case of polyominos, in that they use square tiles with binary attractive interactions. However, both only consider self-interacting colors, and treat rotated tiles as distinct. As a result our encoding, based on necklaces, has two advantages: First, recording the complexity of rotation-invariant colorings allows us to measure the amount of symmetry in the building block. Second, having colors which are not self-interacting means that a color interacting with one particular different color can appear on several different building blocks, encoding the multiple use of the same module without causing nondeterministic assembly.

The general algorithm we use to find the minimum assembly kit $\tilde{S}$, and thus, the complexity K, for polyominoes and other structures is described in Appendix A. Note that we can also use this framework to define the joint, conditional and mutual complexities of two or more structures (see Appendix B).

Figure 2 illustrates how the complexity value K reflects symmetry and modularity present in the structure. Structures which are the same in size may differ considerably in complexity. Equally, larger structures that are more symmetric or modular may have similar complexity to smaller structures that are less symmetric or less modular.

V. APPLICATION TO MOLECULES

The self-assembly approach can be used to calculate complexity values for any physical structure. In order to demonstrate the broad range of potential applications we determine the complexity of (a) molecules and (b) protein complexes.



large size and **low** symmetry results in **high** complexity

small size but **low** symmetry results in **medium** complexity

large size but **high** modularity results in **medium** complexity

small size and **high** symmetry results in **low** complexity

FIG. 2. The complexity values of these four polyomino shapes illustrate why the self-assembly approach is an effective way of measuring symmetry and modularity without requiring prior assumptions. If two shapes are of equal size, the one with more symmetry and modularity has a lower complexity value—compare A with B, and C with D. If on the other hand, two shapes are of similar *complexity*, but of different size, the larger one will be more symmetric or modular (compare B and C).

The problem of molecular complexity has been studied extensively over the past seventy years, starting with work by Pólya [23] and Rashevsky among others [24,25], and culminating in a seminal paper by Bertz [26]. These approaches are based on Shannon entropy rather than algorithmic information theory and focus on symmetries rather than the more general concept of modularity. In molecules, we take atoms to be the building blocks and chemical bonds to be their interfaces. Simple molecules, such as those in Fig. 3, for which we are only interested in the bond connectivity, are an example of a structure in which none of the interfaces between building blocks can be regarded as redundant. This is because, unlike for polyominoes, we are not assuming any inherent geometry for the building blocks. If two atoms play the same self-assembly role but represent atoms of different atomic species, they must be differentiated. This also goes for atoms connected by different bond types. For example, in glutamine (see Fig. 3), the oxygen atom connected with a double bond is a leaf of the self-assembly tree just like any of the (implicit) hydrogen atoms, but it requires a separate building block. The two molecules in our example of Fig. 3 are the amino acid glutamine and the explosive nitroglycerine, which both consist of 20 atoms. Nitroglycerine however exhibits a much higher degree of modularity, with its three

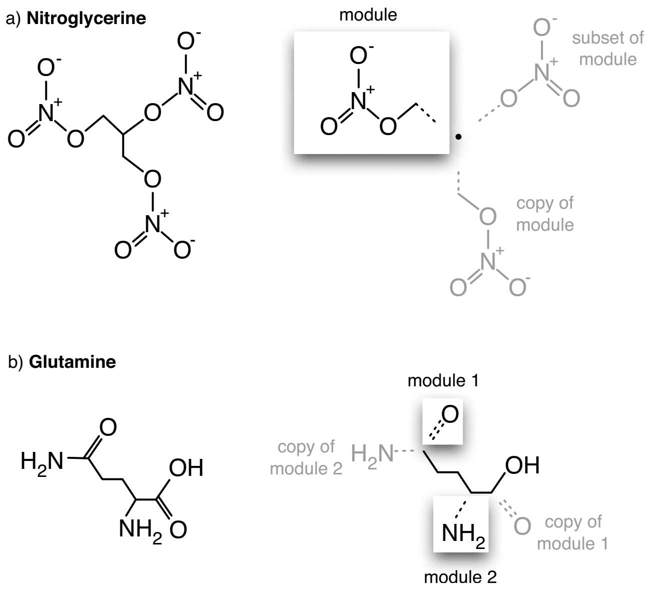


FIG. 3. Measuring the complexity of molecules—The explosive nitroglycerine (top) and the amino acid glutamine (bottom) both consist of 20 atoms, but differ greatly in complexity. The highly modular structure of nitroglycerine with its three NO_3 groups means that its complexity value K, at 52.2 bits, is little more than half that of glutamine (K=91.0 bits). Note that nitroglycerine does not have simple threefold symmetry, but a more subtle modular structure, which the self-assembly approach fully reveals. Note that we do not consider neutral colors in this structure.

NO_3 groups, and therefore has a much lower complexity of K=55.3 bits than the glutamine, for which the value is K =94.7 bits. Note that nitroglycerine does not exhibit simple threefold symmetry, but a more subtle, hierarchical modularity, in which the subset of a module reappears in another place. Such structural features would be harder to identify using traditional approaches to the measurement of molecular complexity [24–26], which rely on Shannon entropy, because these approaches do not take into account the relative complexity of different building blocks (cf. Figures 1(o) and 1(p) in [26]) and conversely can treat components of the same self-assembly module as inequivalent points (see discussion of Fig. 1(h) in [24]), thereby missing the underlying modularity.

VI. APPLICATION TO PROTEIN COMPLEXES

Protein complexes are an important class of biochemical structures, consisting of several individually formed and folded protein *subunits* bound together to produce functional cellular machinery. These subunits may include different types of protein and several copies of the same protein. The physical structure of protein complexes, as with protein themselves, is important in determining the functionality of the complex. The manner in which the subunits bond to form the final complex is known as the *quaternary structure* of the complex. The 3DComplex database [27] contains a description of the quaternary structures of thousands of protein complexes, in terms of subunit type and intersubunit bonding. If we have two proteins which play the same role in the self-assembling structure but are different proteins, we can choose to count them as two different building blocks (analogous to the aforementioned distinction between atomic species in molecules). Here, however, we are only interested in the connectivity of proteins (equivalent to the QS Topology level in the 3DComplex database), and therefore do not distinguish between different proteins.

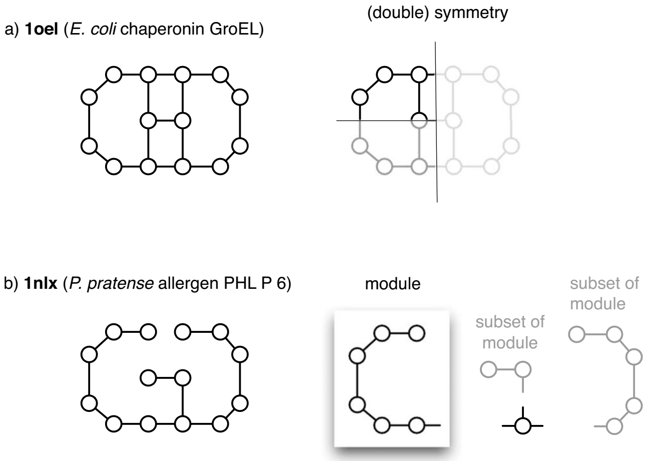

a) 1oel (E. coli chaperonin GroEL)
(double) symmetry
b) 1nlx (P. pratense allergen PHL P 6)
module
subset of module
subset of module

FIG. 4. We measure the complexity of two homomeric protein complexes, with PDB identifiers 1oel (a chaperonin, top) and 1nlx (an allergen, bottom), which have 14 proteins each. The symmetry of the chaperonin complex means that it has a significantly lower complexity value of K=31.5 bits, compared to K=50.2 bits for the allergen complex. Note that we do not consider neutral colors, and in the case of the chaperonin complex we have three self-interacting colors (c_s=3).

As an illustration of our approach, we compare the complexity values of two protein complexes in Fig. 4. These are a chaperonin complex (*E. coli* chaperonin GroEL; PDB identifier: 1oel) and an allergen complex (*P. pratense* allergen PHL P 6; PDB identifier: 1nlx). Both consist of 14 proteins, but the chaperonin complex has a lower complexity value of K=31.5 bits, reflecting the fact that it has a higher symmetry than the allergen complex for which K=50.2 bits.

More complex protein structures require more unique intersubunit bond types, compared to less complex structures which can reuse bonds and be constructed through simple repetition of subunits. As an increase in bond types corresponds biologically to the presence of more unique bonding sites on subunit proteins, more complex protein structures can be thought of as requiring more evolutionary innovation to produce, and would therefore perhaps be expected to occur less frequently in biological organisms [13,28]. This hypothesis is confirmed by Fig. 5, which shows a histogram of complexity values—normalized by the size of the protein complex, to avoid size effects—for the 15733 protein complexes in the 3DComplex database [27]. We note that this distribution closely (R^2=0.93) follows a power-law decay. However, given the complex evolutionary pressures acting on the different protein complexes, and the fact that the protein data bank (PDB) database may itself contain biases because some complexes are easier to crystallize than others, we feel it might be too early to speculate what the exact

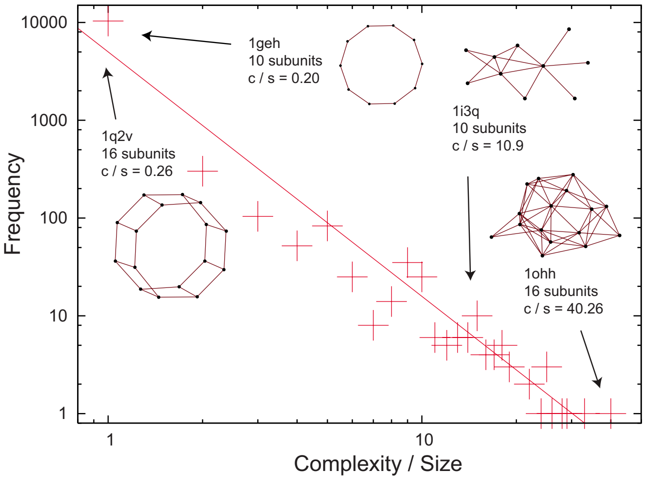


FIG. 5. (Color online) Histogram of the assembly complexity of protein quaternary structures with frequency of occurrence in the 3DComplex database. Insets illustrate two pairs of equally sized structures with high and low complexity values. 1geh, 1i3q, 1q2v, and 1ohh are the PDB identifiers of the complexes. The plot has an $R^2=0.93$ correlation with a power-law decay. Note that in this case we do not distinguish between different types of subunit.

origin of this observed power-law behavior might be.

In both of these cases—molecules and protein complexes—we use unlabelled faces, so that $F_i=\Pi_j^{c_i} k_j^{(i)}$ (see the discussion in the Minimum Kit section above). While the chemical bonds of atoms and the interfaces of proteins are in fact usually constrained, this information is not part of the structural formula of the molecule or the network of contacts between adjacent proteins in the protein complex. If this additional level of resolution is required, a more realistic self-assembly model can be constructed, based on the exact three-dimensional characteristics of the atoms or proteins, and using the $F(c_i,f_i)$ term specified above.

VII. MODULARITY

The self-assembly perspective provides an intuitive definition of the modularity of a structure: If part of the structure appears several times, it still only needs to be encoded once. This is why modularity and symmetry lead to more efficient self-assembly kits and a lower value of the complexity measure K. Formally we can define the modularity m of a structure of size z as the average number of times one of the b different building block types in the minimum assembly kit is used in the structure, which is simply

$$m=\frac{z}{b}.$$

We can furthermore define a *module* formally as a connected set of building blocks which appears more than once in a given structure. Note that modules can overlap: A subset of a module could form another module, appearing a different number of times than the whole module. The molecule in Fig. 3(a) and the protein complex in Fig. 4 illustrate such cases. The majority of protein complexes in the 3DComplex database show modularity values of two or greater (Fig. 6) with a distinct trend observable along the $m=2$ line, indicating many proteins consist of structures involving two copies of all constituent subunits.

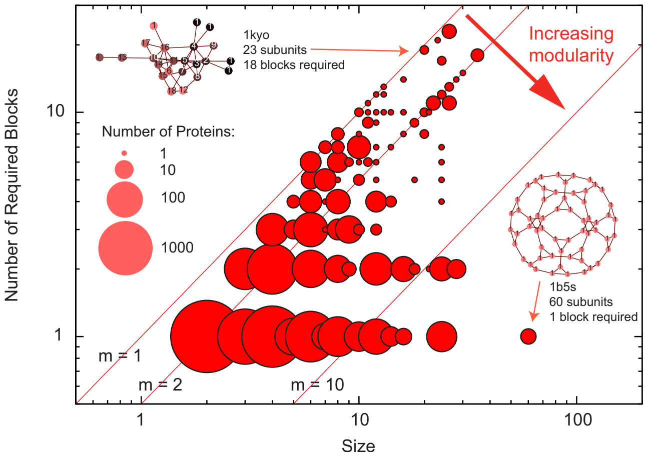


FIG. 6. (Color online) The position of the 15 733 protein complexes from [27] in the space of b (number of building block types) and z (size of the complex). Many protein complexes are highly modular, and this is true across a wide range of sizes. In this plot complexes of equal modularity $m=z/b$ lie on a diagonal line with positive gradient. The lines are shown for $m=1$, 2, and 10. The sizes of the circles show how many complexes lie at a given position (z,b). The insets show two examples (with PDB identifiers 1kyo and 1b5s), with high and low modularities.

To further illustrate how the complexity K and the modularity m measure the physical complexity of protein complexes, we consider two of the outliers in the complexity and modularity histograms, the high-complexity 1ohh (Fig. 5) and high-modularity 1b5s (Fig. 6). 1ohh consists of two copies of bovine F_1-ATPase (itself a protein complex) in complex with its regulatory protein IF_1 [29]. The regulatory protein binds simultaneously to both copies of the main complex, but slightly asymmetrically, leading to asymmetric interactions being recorded in the 3DComplex database. This asymmetry results in extra information being required to describe the combined quaternary structure, and the observed high-complexity value. 1b5s is a multienzyme complex consisting of multiple copies of dihydrolipoyl acetyletransferase (E2p) [30]. The E2p protein has the potential to occupy quasiequivalent positions, as also seen in virus structures [31], and is also observed to form cubic complexes. The highly modular, dodecahedral structure exhibited in 1b5s is an efficient way of grouping many copies of an active protein in a geometry that facilitates enzymatic activity: the large windows in the structure allow passage of the substrate and product into the inner cavity. The structure of the protein subunits allows this structure to be realized with just one building block type, resulting in high modularity.

VIII. DISCUSSION

Here we discuss several subtleties and extensions of the self-assembly framework, and its relationship to the formal theory of algorithmic complexity.

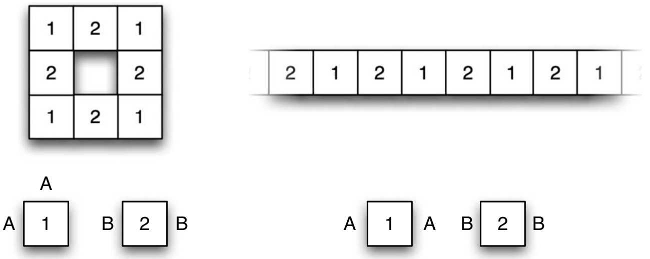


FIG. 7. A simple example of a steric effect. The two blocks 1 and 2 have colors A and B on their interfaces. These colors attract each other. All other faces are neutral. Certain arrangements of colors will lead to self-delimiting structures purely because of the geometry of the building blocks. The complexity of such structures can be taken to be the same as that of an infinite chain consisting of the same sequence of blocks, but only if each loop structure inside a bigger structure has a distinct (set of) species of building blocks.

A. Steric effects

For structures that contain loop structures formed by repeating units, it is possible to exploit steric effects in order to reduce the size of the assembly kit below the minimum size of the algorithm described in Appendix A (which explicitly excludes such effects in its definition). An example of a steric effect would be a polyomino that is self-limiting in a deterministic way, purely because of the geometric constraints of the building blocks. As long as each distinct type of loop structure is formed by building blocks of a distinct species (or set of species), the amount of information required to describe this structure can be taken to be the same as that required to describe an infinite chain consisting of the same elements. A simple example is given in Fig. 7. The crucial assumption which has to hold for this simplification to work is that the geometry of the loop is specified by the species (and, by extension, the geometry) of the building block. For proteins as building blocks of protein complexes, this is a very reasonable assumption. In the case of molecules it would furthermore be possible to simplify the self-assembly kit by introducing building blocks representing common small loop structures, such as carbon rings.

B. Multiple nuclei

In principle one could consider beginning the self-assembly with multiple nuclei in place. Multiple nuclei may, through steric hindrance or modular repetition, be used to achieve certain structures in a more efficient way, using fewer building blocks than a single nucleus would require. This reduction in complexity may however be countered in practical applications by the difficulty of achieving the required precise relative displacements of nucleus particles. It is because of these reasons that we have concentrated on a single nucleus model, as the positioning of multiple nuclei makes it much more difficult to construct a general measure of complexity.

Within the single nucleus category, we further distinguish between structure with a *specified nucleus block* and those with *general nucleus blocks*. The former case encompasses those assembly kits which are guaranteed to produce a given output structure if and only if a specified block is used as the nucleus (in other words, this block is placed on the substrate before other blocks are introduced to the system). General-nucleus assembly kits by contrast will form the same output structure regardless of which block is placed first. See Fig. 8 for an illustration how specifying a nucleus can reduce the complexity of a assembly kit.

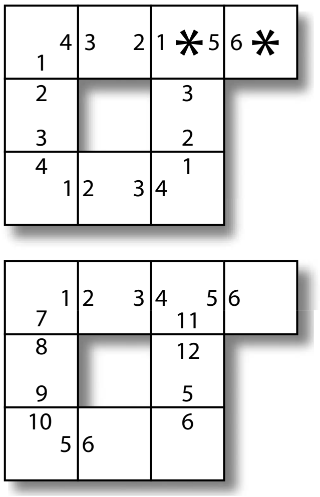


FIG. 8. Illustration of nuclei placement. (Top:) If we specify either of the two starred blocks as nuclei, deterministic bonding will result. However, if any other block is used as the nucleus, bonding will be non-deterministic, as both the {1,0,0,4} and {1,0,5,0} blocks can join the open ‘2’ faces that will form. This self-assembly kit has a complexity of $K=42.4$ bits. (Bottom:) A general nucleus system to produce the same structure, illustrating the required increase in complexity ($K=98.1$ bits).

Which of these classes to employ in a study depends on the motivating context of the self-assembling system under consideration. If modeling assembly in a diffusion-dominated environment, for example, the order in which interacting particles meet cannot be specified, so the general-nucleus model is more appropriate. In a controlled environment where a nucleus can be placed to initiate assembly, the single-nucleus model is applicable. The two cases correspond to different “languages” being used to measure complexity, and so care must be taken in comparative studies to only compare numerical complexity values from within one class.

C. Kolmogorov complexity

Our approach to measuring physical complexity is motivated by the concept of Kolmogorov complexity. It is however important to note that while Kolmogorov complexity itself is uncomputable due to the halting problem [3], our minimum is not. This is because the runtime of a finite computer program with finite output can be infinite, while the assembly time of a finite shape is always finite [5]. It is possible to define the actual Kolmogorov complexity of a shape [6], but this is uncomputable. Our computable complexity measure $K(A)$ forms a bound on this unattainable quantity, and is dependent on the way in which we encode

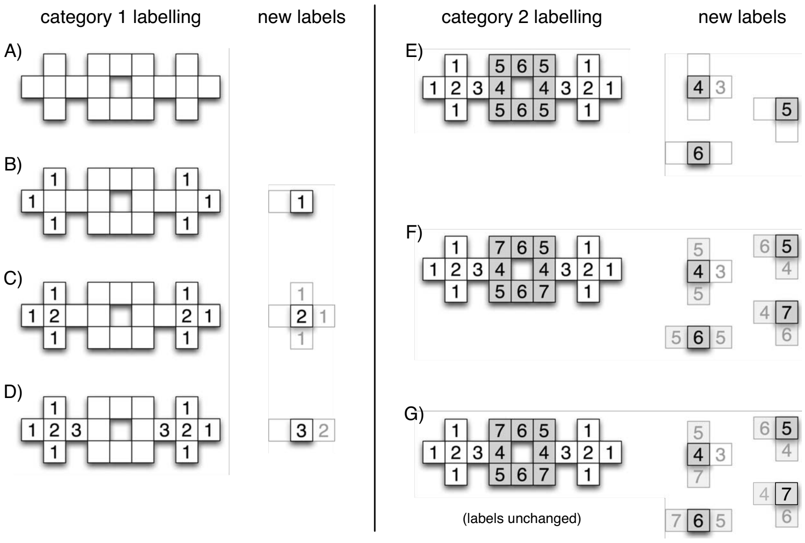


FIG. 9. An illustration of the crucial steps 5b to 5j of the algorithm for minimizing the assembly kit size, in this case for a polyomino. In every iteration of category 1 labelings (left), all unlabelled nodes with exactly one unlabelled neighbor are given labels which distinguish them according to their topologically distinct neighborhoods of unlabelled and labeled tiles. This procedure is repeated until no more blocks can be labeled in this way. The remaining blocks are given category 2 labelings (right) which are applied simultaneously, with each label distinguishing the topological neighborhoods of the tiles in the previous iteration. Note that in the last iteration the labelings have stabilized, and only the interfaces of the building block types are updated. For structures in which edges can be redundant, this operation can be performed for all spanning subgraphs of the structure's connectivity graph, which further reduces the complexity. (In polyominoes, edges can be redundant, but there are no spanning subgraphs in the above example.)

the description of the assembly kit. It therefore is useful for the analysis, classification and comparison of physical structures, as long as we use a consistent encoding.

IX. CONCLUSION

We present a general approach for measuring the physical complexity of any connected structure, using the language of self-assembly. This approach is capable of detecting symmetry and modularity in a given structure, because these features significantly decrease the size of the required self-assembly instruction set. It therefore provides a powerful tool for automated classification and categorization of physical structures, and could be applied to large-scale databases of molecules and crystal structures. In addition, the connection between self-assembly and complexity is an argument for the ubiquity of modular and symmetric features in biological systems: Since many such systems self-assemble, evolving sets of self-assembly instructions are likely to yield symmetric and modular structures, as the instructions for these are more efficient to evolve.

ACKNOWLEDGMENTS

S.E.A. was supported by The Royal Society, U.K. I.G.J. and A. A. L. were supported by te EPRSC. T.M.A.F. was supported by the Defense Advanced Research Projects Agency (DARPA), Grant No. HR0011–09–1-0055.

APPENDIX A: GENERAL ALGORITHM FOR MINIMIZING THE ASSEMBLY KIT

Below we describe a general algorithm for minimizing the assembly kit size for a connected physical structure without relying on steric effects. Taking these into account can minimize the assembly kit even further, but their computation is highly dependent on the geometry of the system and in most cases nontrivial (see Discussion). Note also that in some structures, such as polyominoes, some edges of the contact graph (meaning the graph connecting neighboring building blocks of the structure) can be redundant in the context of the assembly process. Whether contact graph edges in general can be redundant or not depends on the nature of the structure and the assumptions connected to the self-assembly of that structure (see Discussion). Similarly, when interfaces are defined by geometry, as for the four sides of a polyomino building block, it makes sense to introduce a neutral color ($\nu=1$ below). In systems with a varying number of interfaces on the building blocks, neutral colors are usually not required ($\nu=0$).

To minimize the assembly kit we take the following steps:

(1) Divide the structure into building blocks (usually a natural division). The number of building blocks is the *size* of the structure, denoted z.

(2) Determine the equivalence of these units in terms of any additional criteria (e.g., types of atoms, proteins). This categorization is the *species* of building block.

(3) Establish a contact graph for the units (in some cases, such as molecules, this may require setting a distance cutoff). The contact graph can be represented as a $z \times z$ adjacency matrix a_{ij}, which is 1 if units i and j are in contact, and 0 otherwise.

(4) *If edges can be redundant:* Consider the space of all spanning subgraphs of this graph.

(5) For the contact graph (in the case of no redundant edges) or each subgraph (if redundant edges exist):

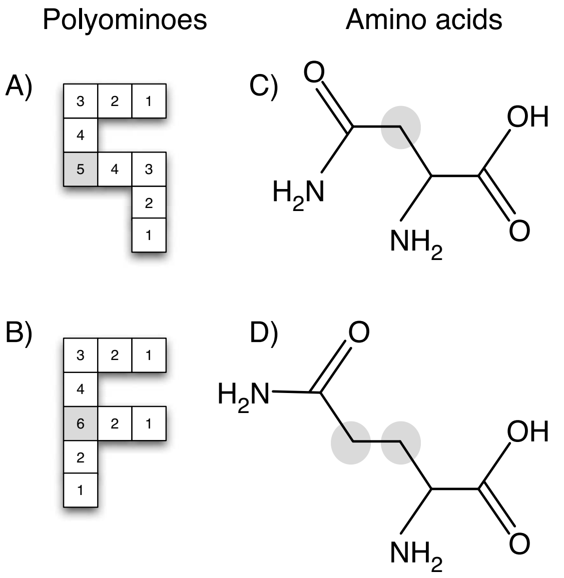


FIG. 10. POLYOMINOES (left): The two polyominoes [shown in (a) and (b)] share many building block types, with the only two unique ones being blocks 5 and 6 (marked in gray). Hence, the joint set is $\tilde{S}_{A,B}=\{1,2,3,4,5,6\}$, the mutual set is $\tilde{S}_{A:B}=\{1,2,3,4\}$ and the conditional sets are: $\tilde{S}_{A|B}=\{5\}$ and $\tilde{S}_{B|A}=\{6\}$. Building block 5 contributes $K(A|B)=2\log_2 9+2=8.4$ bits to the complexity $K'(A)$ of the A shape, while block 6 contributes $K(B|A)=4\log_2 9=12.7$ bits to $K'(B)$. It follows therefore that the joint complexity is $K(A,B)=67.4$ bits and the mutual complexity is $K(A:B)=46.4$ bits, compared to the standalone values of $K(A)=K'(A)=54.7$ bits and $K(B)=K'(B)=59.1$ bits (see Fig. 2). AMINO ACIDS (right): The two amino acid molecules asparagine (top, C) and glutamine (bottom, D) share the amino (NH_2) and carboxyl (CO_2H) groups common to all amino acids, as well as the carboxamide group ($CONH_2$). In a self-assembly framework these two structures have complexities of $K(Asn)=74.3$ bits and $K(Gln)=91$ bits. While $K'(Gln)=K(Gln)$, we have $K'(Asn)=78.0$ bits. Because the two molecules share three groups, their joint complexity is not much larger than their individual complexities, at $K(Asn,Gln)=104.0$ bits, and their mutual complexity is not much smaller, at $K(Asn:Gln)=65$ bits, than the complexities of the individual molecules. Their conditional complexities are correspondingly low, at $K(Asn|Gln)=13$ bits and $K(Gln|Asn)=26$ bits. The conditional complexities give the amount of information required to describe the building blocks (atoms) which are unique (in their self-assembly role) to the given amino acid. These atoms are marked with gray circles.

(a) Classify the nodes of the (sub)graph according to the number of connections and (depending on the geometry) the arrangement of connections.

(b) Label all nodes which are not yet labeled and which have exactly one unlabelled node among their neighbors. The new labels distinguish nodes according to their species as well as the topologically distinct label distributions among their neighbors.

(c) Repeat step 5b until all nodes are labeled or no more nodes can be labeled.

(d) All labeled nodes we define as *category 1* nodes and any remaining unlabelled nodes (i.e., nodes with at least two unlabelled neighbors) are defined as *category 2* nodes.

(e) Label all category 2 nodes simultaneously according to their neighborhoods.

(f) Repeat step 5e, using the previous labelings to distinguish neighborhoods, until labelings are stable.

(g) These final labels, for nodes in both categories, denote the building block *types*. The number of final labels, or types, is b. These can be subdivided in to b_1 category 1 building block types and b_2 category 2 building block types. The category 2 type of block i is denoted t_i.

(h) The degree of each building block type i in the contact graph (or subgraph) is the number of its interfaces f_i.

(i) For unlabelled faces, the total number of colors, including $\nu\in\{0,1\}$ neutral colors, is $c=2(b_1-1+\delta_{0b_1})+\nu+\Sigma_{i,j=1}^{b_2}\{1-\Pi_{k,l=1}^{z}[1-(a_{kl}\delta_{it_k}\delta_{jt_l})]\}$. The sum expression gives the number of different types of interfaces which occur between category 2 building block types. Heterogeneous interfaces are double counted as, unlike homogeneous interfaces, they require two colors. The number of colors c_i on building block i is equal to the number of building block types in its contact graph neighbor set. For labeled faces the sum runs over all labeled faces of the category 2 building blocks, instead of just running over b_2, and the product runs over all pairs of faces in the structure. The adjacency matrix a_{kl} becomes a matrix between the faces and the δ_{it_k} and δ_{jt_l} become indicators whether e.g., face k is an example of the labeled face i. And the number of colors c_i on building block i is now equal to the number of distinct pairs of labeled face types in the face contact graph neighbor set.

(j) Using b, c, $\{f_i\}$ and $\{c_i\}$ in Eq. (1), calculate the information I required to specify this assembly kit, and thus the complexity K of the structure.

6. *If edges can be redundant:* Minimize this quantity over all spanning subgraphs.

Figure 9 illustrates the crucial steps 5b to 5j for a polyomino.

APPENDIX B: JOINT, CONDITIONAL, AND MUTUAL COMPLEXITY

If we have two structures A and B with minimum assembly kits $\tilde{S}_A$ and $\tilde{S}_B$, then the *joint* minimum assembly kit $\tilde{S}_{A,B}$ is the minimum kit which can assemble both structures if an appropriate subset of building blocks is chosen. The amount of information required to describe this kit is the *joint complexity* $K(A,B)$ of A and B. This definition can easily be generalized to more than two structures.

Let us define $\tilde{S}'_A$ as the subset of $\tilde{S}_{A,B}$ which forms structure A, and $\tilde{S}'_B$ as the subset of $\tilde{S}_{A,B}$ which forms structure B (note that e.g., $\tilde{S}_A$ is not necessarily equal to $\tilde{S}'_A$ due to the color minimization), so that $\tilde{S}_{A,B}=\tilde{S}'_A\cup\tilde{S}'_B$. Furthermore, let us define the *conditional* minimum assembly kit $\tilde{S}_{A|B}$ as the set of building blocks we need in addition to $\tilde{S}'_B$ in order to form structure A. Then we can write:

$$\tilde{S}_{A|B}=\tilde{S}_{A,B}\setminus\tilde{S}'_B,$$

where \ denotes the set theoretic difference operation. The definition of $\tilde{S}_{B|A}$ follows accordingly. Hence we can also define a *conditional complexity* $K(A|B)$, which is the amount of information needed to describe the building blocks in $\tilde{S}_{A|B}$. Because the way we describe the assembly kit is additive in the number of building blocks, we can write

$$K(A|B) = K(A,B) - K'(B),$$

since $K'(B)$ is the information required to describe the building blocks in $\tilde{S}'_B$. The relationship between $K(B)$ and $K'(B)$ is given by

$$K'(B) = K(B) + \sum_i c_i \log_2 \frac{c_{A,B}}{c_B},$$

where $c_{A,B}$ is the total number of colors in $\tilde{S}_{A,B}$ and c_B is the total number of colors in $\tilde{S}_B$. Because of the minimization of colors, $c_{A,B}=\max(c_A,c_B)$. Hence, if $c_B \geq c_A$, then $K'(B) = K(B)$.

Similarly, we can define a mutual minimum assembly kit $\tilde{S}_{A:B}$, which corresponds to the intersection

$$\tilde{S}_{A:B} = \tilde{S}'_A \cap \tilde{S}'_B = \tilde{S}'_A \setminus \tilde{S}_{A|B} = \tilde{S}'_B \setminus \tilde{S}_{B|A}.$$

From this follows the *mutual complexity*

$$K(A:B) = K'(A) - K(A|B) = K'(B) - K(B|A) = K'(A) + K'(B) - K(A,B). \qquad (3)$$

In order to account for the relative sizes of the structures we compare using these measures, we can define relative versions of the above quantities. These are *relative conditional complexity*:

$$K^{rel}(A|B) = \frac{K(A|B)}{K'(B)}$$

and the *relative mutual complexity*

$$K^{rel}(A:B) = \frac{K(A:B)}{K(A,B)}.$$

Note that the latter measure resembles the Jaccard index [32]. For an illustration of joint, mutual and conditional complexity, see Fig. 10.

APPENDIX C: NECKLACES

An a-ary necklace of length n is a string of n characters taken from an alphabet of a possible characters. Cyclic rotation of the string is ignored, so that, for example, $1234 \equiv 4123$ [31]. A fixed-bond building block i, with f_i faces and c_i colors, can then be represented by a c_i-ary necklace of length f_i. The number of a-ary necklaces of length n is

$$N(n,a) = \frac{1}{n} \sum_{i=1}^{v(n)} \phi(d_i) a^{n/d_i} \qquad (4)$$

where $v(n)$ is the number of divisors of n, and d_i are these divisors listed in order, from $d_1=1$ to $d_{v(n)}=n$. $\phi(n)$ is the Euler totient function, returning the number of positive integers less than or equal to n that do not contain any factor in common with n (the number of n's relative primes). For $n=4$ and $a=c$ we get the expression for N_c in the main text.

[1] A. N. Kolmogorov, Probl. Inf. Transm. **1**, 3 (1965).
[2] G. J. Chaitin, J. ACM **13**, 547 (1966).
[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley-Interscience, New York, 1991).
[4] A. M. Turing, Proc. London Math. Soc. **s2-42**, 230 (1937).
[5] P. W. K. Rothemund and E. Winfree, *STOC '00: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing* (ACM, New York, 2000), pp. 459–468.
[6] D. Soloveichik and E. Winfree, SIAM J. Comput. **36**, 1544 (2007).
[7] C. Adami, Phys. Life. Rev. **1**, 3 (2004).
[8] G. M. Whitesides and M. Boncheva, Proc. Natl. Acad. Sci. U.S.A. **99**, 4769 (2002).
[9] G. Krausch and R. Magerle, Adv. Mater. **14**, 1579 (2002).
[10] J. Israelachvili, Langmuir **10**, 3774 (1994).
[11] H. Fraenkel-Conrat and R. C. Williams, Proc. Natl. Acad. Sci. U.S.A. **41**, 690 (1955).
[12] A. Zlotnick, J. Mol. Biol. **241**, 59 (1994).
[13] G. Villar, A. W. Wilber, A. J. Williamson, P. Thiara, J. P. K. Doye, A. A. Louis, M. N. Jochum, A. C. F. Lewis, and E. D. Levy, Phys. Rev. Lett. **102**, 118106 (2009).
[14] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, Nature (London) **394**, 539 (1998).
[15] C. Mao, T. H. LaBean, J. H. Reif, and N. C. Seeman, Nature (London) **407**, 493 (2000).
[16] A. Chworos, I. Severcan, A. Y. Koyfman, P. Weinkam, E. Oroudyev, H. G. Hansma, and L. Jaeger, Science **306**, 2068 (2004).
[17] R. P. Goodman *et al.*, Science **310**, 1661 (2005).
[18] P. W. K. Rothemund, Nature (London) **440**, 297 (2006).
[19] K. Fujibayashi, R. Hariadi, S. H. Park, E. Winfree, and S. Murata, Nano Lett. **8**, 1791 (2008).
[20] E. W. Weisstein, "Polyomino," from MathWorld—A Wolfram Web Resource. http://mathworld.wolfram.com/Polyomino.html.
[21] E. W. Weisstein, "Necklace," from MathWorld—A Wolfram Web Resource. http://mathworld.wolfram.com/Necklace.html.
[22] H. Wang, Bell Syst. Tech. J. **40**, 1 (1961).
[23] G. Pólya, Acta Math. **68**, 145 (1937).
[24] N. Rashevsky, Bull. Math. Biophys. **17**, 229 (1955).
[25] E. Trucco, Bull. Math. Biophys. **18**, 129 (1956).
[26] S. H. Bertz, J. Am. Chem. Soc. **103**, 3599 (1981).
[27] E. D. Levy, J. B. Pereira-Leal, C. Chotia, and S. A. Teichmann, PLOS Comput. Biol. **2**, e155 (2006).
[28] E. D. Levy, E. B. Erba, C. V. Robinson, and S. A. Teichmann, Nature (London) **453**, 1262 (2008).
[29] E. Cabezón, M. G. Montgomery, A. G. W. Leslie, and J. E. Walker, Nat. Struct. Mol. Biol. **10**, 744 (2003).

[30] T. Izard, A. Ævarsson, M. D. Allen, A. H. Westphal, R. N. Perham, A. de Kok, and W. G. J. Hol, Proc. Natl. Acad. Sci. U.S.A. **96**, 1240 (1999).

[31] D. Caspar and A. Klug, Cold Spring Harb Symp. Quant Biol. **27**, 1 (1962).

[32] P. Jaccard, Bull. Soc. Vaud. Sci. Nat. **37**, 547 (1901).

[33] For *free* necklaces, which represent building blocks with no fixed chirality there are $M_c = (c^4 + 2c^3 + 3c^2 + 2c)/8$ necklaces [21]. In general we will assume fixed chirality.