Populations of genetic circuits are unable to find the fittest solution in a multilevel genotype-phenotype map
Supplementary Material

Pablo Catalán[1,2,*], Susanna Manrubia[1,3] and José A. Cuesta[1,2,4,5]

[1]Grupo Interdisciplinar de Sistemas Complejos (GISC), Madrid, Spain; [2]Dept. de Matemáticas, Universidad Carlos III de Madrid, Leganés, Madrid, Spain; [3]Programa de Biología de Sistemas, Centro Nacional de Biotecnología (CSIC), Madrid, Spain; [4]Instituto de Biocomputación y Física de Sistemas Complejos (BIFI) Universidad de Zaragoza, Spain; [5]UC3M-Santander Big Data Institute (IBiDat), Universidad Carlos III de Madrid, Getafe, Madrid, Spain.

[*]To whom correspondence should be addressed: pablocatalanfdez@gmail.com (PC)

Contents

1 Supplementary Text

t$_{oy}$LIFE was originally presented in [1]. We give here its main details, with slight modifications in the definition of the model, as presented in [2].

1.1 Building blocks: genes, proteins, metabolites

The basic building blocks of t$_{oy}$LIFE are toyNucleotides (toyN), toyAminoacids (toyA), and toySugars (toyS). Each block comes in two flavors: hydrophobic (H) or polar (P). Random polymers of basic blocks constitute toyGenes (formed by 20 toyN units), toyProteins (chains of 16 toyA units), and toyMetabolites (sequences of toyS units of arbitrary length). These elements of t$_{oy}$LIFE are defined on two-dimensional space (Supplementary Figure S1).
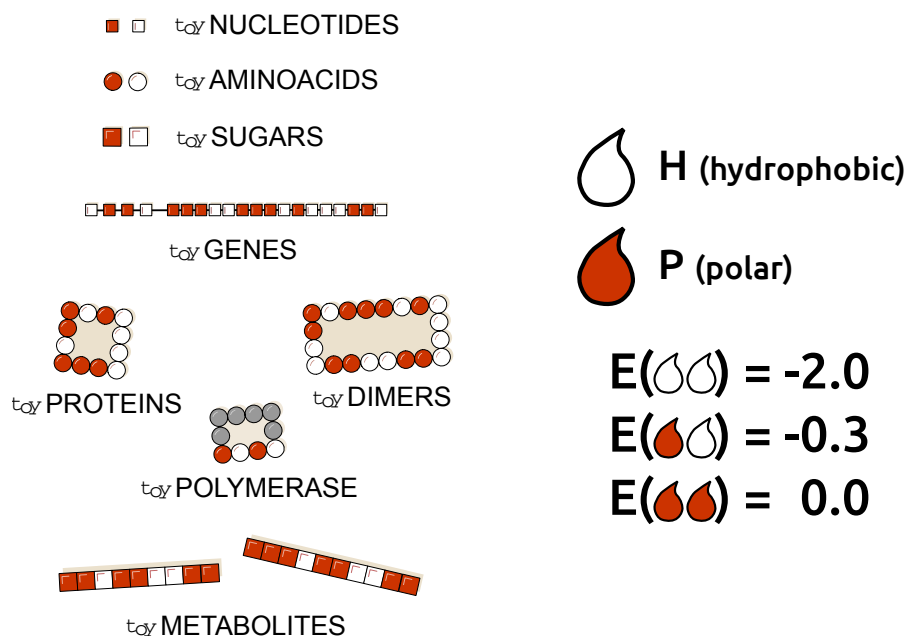
Supplementary Figure S1: **Building blocks and interactions defining t$_{oy}$LIFE.** The three basic building blocks of t$_{oy}$LIFE are toyNucleotides, toyAminoacids, and toySugars. They can be hydrophobic (H, white) or polar (P, red), and their random polymers constitute toyGenes, toyProteins, and toyMetabolites. The toyPolymerase is a special polymer that will have specific regulatory functions. These polymers will interact between each other following an extension of the HP model (see text), for which we have chosen the interaction energies $E_{HH} = -2$, $E_{HP} = -0.3$ and $E_{PP} = 0$ [3].

toyGenes

toyGenes are composed of a 4–toyN promoter region followed by a 16-toyN coding region. There are 2^4 different promoters and 2^{16} coding regions, leading to $2^{20} \approx 10^6$ toyGenes. An ensemble of toyGenes forms a genotype. If the toyGene is expressed, it will produce a chain of 16 toyA that represents a toyProtein. Translation follows a straightforward rule: H (P) toyN translate into H (P) toyA. Point mutations in toyLIFE are easy to implement: they are changes in one of the nucleotides in one of the genes in the genotype. If the sequence has a H toyN in that position, then a mutation will change it to a P toyN, and vice versa.

toyProteins

toyProteins correspond to the minimum energy, maximally compact folded structure of the 16 toyA chain arising from a translated toyGene. Their folded configuration is calculated through the hydrophobic-polar (HP) protein lattice model [3,4].

We only consider maximally compact structures. That is, every toyProtein must fold on a 4×4 lattice, following a self-avoiding walk (SAW) on it. After accounting for symmetries —rotations and
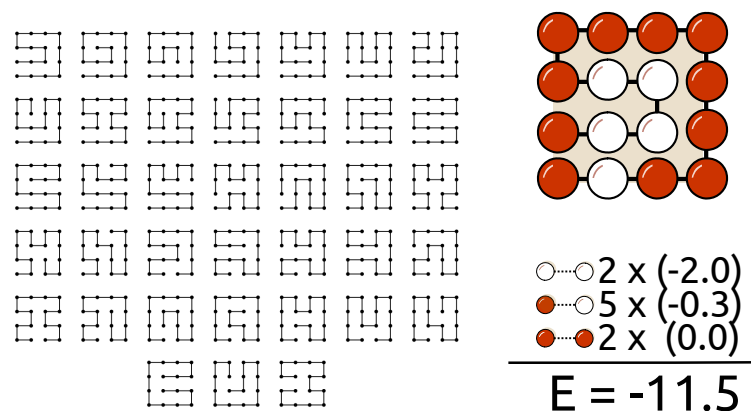
Supplementary Figure S2: **Protein folding in toyLIFE.** toyProteins fold on a 4×4 lattice, following a self-avoiding walk (SAW). Discarding for symmetries, there are 38 SAWs (left). For each binary sequence of length 16, we fold it into every SAW and compute its folding energy, following the HP model. For instance, we fold the sequence PHPPPPPPPPPHHHHP into one of the SAWs and compute its folding energy (right). There are two HH contacts, five HP contacts and two PP contacts —we only take into account contacts between non-adjacent toyAminoacids. Summing all this contacts with their corresponding energies, we obtain a folding energy of -11.5. Repeating this process for every SAW, we obtain the minimum free energy structure.

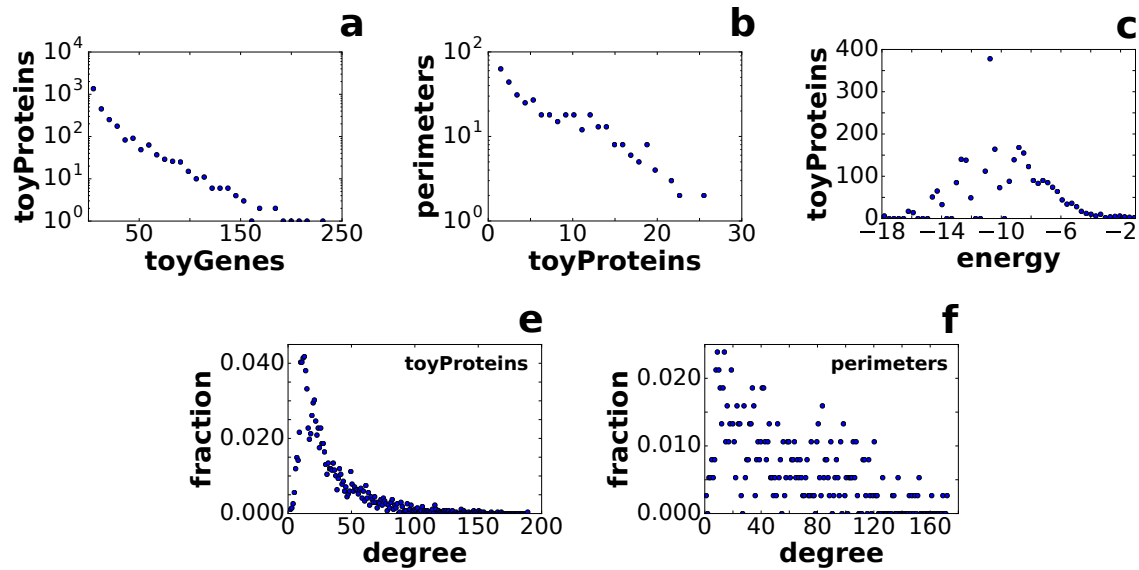reflections—, there are only 38 SAWs on that lattice (Supplementary Figure S2).

The energy of a fold is the sum of all pairwise interaction energies between toyA that are not contiguous along the sequence. Pairwise interaction energies are $E_{HH} = -2$, $E_{HP} = -0.3$ and $E_{PP} = 0$, following the conditions set in [3] that $E_{PP} > E_{HP} > E_{HH}$ (Supplementary Figure S2). toyProteins are identified by their folding energy and their perimeter. If there is more than one fold with the same minimum energy, we select the one with fewer H toyAminoacids in the perimeter. If still there is more than one fold fulfilling both conditions, we discard that protein by assuming that it is intrinsically disordered and thus non-functional. Note, however, that sometimes different folds yield the same folding energy and the same perimeter. In those cases, we do not discard the resulting toyProtein.

Out of $2^{16} = 65,536$ possible toyProteins, $12,987$ do not yield unique folds. We find $2,710$ different toyProteins with 379 different perimeters. Not all toyProteins are equally abundant: although every toyProtein is coded by 19.4 toyGenes on average, most of them are coded by only a few toyGenes. For instance, $1,364$ toyProteins —roughly half of them!— are coded by less than 10 toyGenes each. On the other hand, only 4 toyProteins are coded by more than 200 toyGenes each, the maximum being 235 toyGenes coding for the same toyProtein. The distribution is close to an exponential decay (Supplementary Figure S3**a**). The same happens with the perimeters, although with less skewness: each perimeter is mapped by 7.15 toyProteins on average, but the most abundant perimeters correspond to 26 toyProteins, and 100 are mapped by 1 or 2 toyProteins each (Supplementary Figure S3**b**).

Folding energies range from -18.0 to -0.6, with an average in -9.63. The distribution is unimodal, although very rugged (Supplementary Figure S3**c**). Note that folding energies are discrete, and that separations between them are not equal. For instance, there are 6 toyProteins that have a folding energy of -18.0, but the next energy level is -16.3, realised by 17 toyProteins, and yet the next level is -16.0, realised by 14 toyProteins. The mode of the distribution is -10.6, realised by 202 toyProteins.

We can also study the structure of the toyProtein network (Supplementary Figure S3**e**, **f**). The nodes of this network will be the $2,710$ toyProteins. toyProtein 1 and toyProtein 2 will be neighbors if there is a pair of toyGenes that express each toyProtein and whose sequence is equal but for one toyN. The weight of the edge between toyProtein1 and 2 will be the sum of such pairs of toyGenes. It is surprising that there are no self-loops in this network —there are no mutations connecting one toyProtein to itself. In other words, although there is a strong degeneracy in the mapping from toyGenes to toyProteins, there are no connected neutral networks. If we consider just the perimeters, however, the neutrality is somewhat recovered: out of the 379 perimeters, 224 of them have neutral neighbors. So there are many mutations that alter the folding energy of a toyProtein without changing the perimeter. In this

a

toyProteins (10^0–10^4) vs toyGenes (50–250)

b

perimeters (10^0–10^2) vs toyProteins (0–30)

c

toyProteins (0–400) vs energy (-18 to -2)

e

fraction (0.000–0.040) vs degree (0–200), toyProteins

f

fraction (0.000–0.020) vs degree (0–160), perimeters

Supplementary Figure S3: **Distributions of toyProteins in t$_{OY}$LIFE. (a)** Distribution of toyProtein abundances —that is, the number of toyGenes that code for them. Most toyProteins are coded by few toyGenes, but some of them are very abundant: the most abundant toyProtein is coded by 235 toyGenes. **(b)** Distribution of the perimeters associated with each toyProtein. Again, not all perimeters are equally abundant, and some of them correspond to as many as 25 toyProteins, while 100 correspond to 1 or 2 toyProteins. **(c)** Distribution of folding energies. The range of folding energies goes from -18.0 to -0.6, with a unimodal, rugged distribution. The mode is -10.6, a folding energy achieved by 202 toyProteins. **(d)** Degree distribution in the toyProtein network. Two toyProteins are connected if there are two toyGenes coding for them that have the same sequence, except for one toyN. The average degree is 32.2. **(e)** Degree distribution in the perimeter network. Two perimeters are neighbors if the toyProteins associated to them are neighbors. The average degree is 53.3.

sense, t$_{OY}$LIFE is capturing a complex detail of molecular biology: mutations appear to be neutral from one point of view —in this case, perimeter— but are rarely entirely neutral. In other words, the value of a mutation is context and environment-dependent. There are always some small changes in the molecule —in this case, folding energy— that may affect their function later down the line. Real world examples of this *cryptic* effects of mutations on molecules are everywhere [5–8]. Connections between toyProteins are scarce too: the average degree in the toyProtein network is 32.2 (with a standard deviation of 25.7), a very small number — on average, each toyProtein is connected to hardly 1% of the rest of toyProteins! (Supplementary Figure S3**e**). The maximum degree is 190. This means that mutating from one toyProtein to another is not easy in general. In terms of perimeters this is more relaxed, as the average degree in the perimeter network is 53.3 (standard deviation is 38.1), with a

maximum degree of 173. On average, every perimeter is connected to 14% of the rest of perimeters: it is a small number, but it is still higher than in the toyProtein case (Supplementary Figure S3**f**).

In the $\text{t}_{\text{OY}}\text{LIFE}$ universe, only the folding energy and perimeter of a toyProtein matter to characterise its interactions, so folded chains sharing these two features are indistinguishable. This is a difference with respect to the original HP model, where different inner cores defined different proteins and the composition of the perimeter was not considered as a phenotypic feature. However, subsequent versions of HP had already included additional traits [9].

The toyPolymerase (Supplementary Figure S1) is a special toyA polymer, similar to a toyProtein in many aspects, but that is not coded for by any toyGene. It has only one side, with sequence PHPH, and its folding energy is taken to be -11.0. We will discuss its function and place later on.

1.2 Extending the HP model: interactions

toyProteins interact through any of their sides with other toyProteins, with promoters of toyGenes, and with toyMetabolites (see Supplementary Figure S4**a**). When toyProteins bind to each other, they form a toyDimer, which is the only protein aggregate considered in $\text{t}_{\text{OY}}\text{LIFE}$. The two toyProteins disappear, leaving only the toyDimer. Once formed, toyDimers can also bind to promoters or toyMetabolites through any of their sides —binding to other toyProteins or toyDimers, however, is not permitted. In all cases, the interaction energy (E_{int}) is the sum of pairwise interactions for all HH, HP and PP pairs formed in the contact —these interactions follow the rules of the HP model as well. Bonds can be created only if the interaction energy between the two molecules E_{int} is lower than a threshold energy $E_{\text{thr}} = -2.6$. Note that a minimum binding energy threshold is necessary to avoid the systematic interaction of any two molecules. Low values of the threshold would lead to many possible interactions, which would increase computation times. High values would lead to very few interactions, and we would obtain a very dull model. Our choice of $E_{\text{thr}} = -2.6$ achieves a balance: the number of interactions is large enough to generate complex behaviours, as we will see later on, while at the same time keeping the universe of interactions small enough to handle computationally. If below threshold, the total energy of the resulting complex is the sum of E_{int} plus the folding energy of all toyProteins involved. The lower the total energy, the more stable the complex. When several toyProteins or toyDimers can bind to the same molecule, only the most stable complex is formed. Consistently with the assumptions for protein folding, when this rule does not determine univocally the result, no binding is produced.

As the length of toyMetabolites is usually longer than 4 toyS (the length of interacting toyProtein sites), several binding positions between a toyMetabolite and a toyProtein might share the same energy.
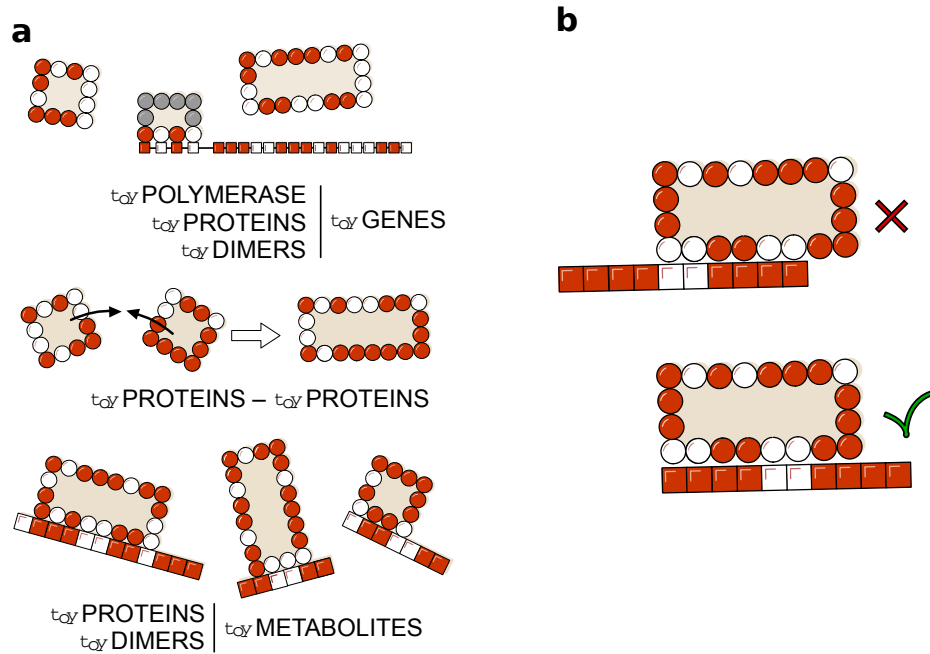
Supplementary Figure S4: **Interactions in toyLIFE. (a)** Possible interactions between pairs of toyLIFE elements. toyGenes interact through their promoter region with toyProteins (including the toyPolymerase and toyDimers); toyProteins can bind to form toyDimers, and interact with the toyPolymerase when bound to a promoter; both toyProteins and toyDimers can bind a toyMetabolite at arbitrary regions along its sequence. **(b)** When a toyDimer or toyProtein binds to a toyMetabolite with the same energy in many places, we choose the most centered binding position. If two or more binding positions have the same energy and are equally centered, then no binding occurs.

In those cases we select the sites that yield the most centered interaction (Supplementary Figure S4**b**). If ambiguity persists, no bond is formed. Also, no more than one toyProtein / toyDimer is allowed to bind to the same toyMetabolite, even if its length would permit it. toyProteins / toyDimers bound to toyMetabolites cannot bind to promoters.

Interaction rules in toyLIFE have been devised to remove any ambiguity. When more than one rule could be chosen, we opted for computational simplicity, having made sure that the general properties of the model remained unchanged. A detailed list of the specific disambiguation rules implemented in the model follows:

1. **Folding rule:** if a sequence of toyAminoacids can fold into two (or more) different configurations with the same energy and two different perimeters with the same number of H, it is considered degenerate and does not fold.

2. **One-side rule:** any interaction in which a toyProtein can bind any ligand with two (or more)

7

different sides and the same energy is discarded.

3. **Annihilation rule:** if two (or more) toyProteins can bind a ligand with the same energy, the binding does not occur. However, if a third toyProtein can bind the ligand with greater (less stable) energy than the other two, and does so uniquely, it will bind it.

4. **Identity rule:** an exception to the Annihilation rule occurs if the competing toyProteins are the same. In this case, one of them binds the ligand and the other(s) remains free.

5. **Stoichiometric rule:** an extension of the Identity rule. If two (or more) copies of the same toyProtein / toyDimer / toyMetabolite are competing for two (or more) different ligands, there will be binding if the number of copies of the toyProtein / toyDimer / toyMetabolite equals the number of ligands. For example, say that P1 binds to P2, P3 and P4 with the same energy. Then, (a) if P1, P2 and P3 are present, no complex will form; (b) if there are two copies of P1, dimers P1-P2 and P1-P3 will both form; but (c) if P4 is added, no complex will form. Conversely, if all ligands are copies as well, the Stoichiometry rule does not apply. For example, three copies of P1 and two copies of P2 will form two copies of dimer P1-P2, and one copy of P1 will remain free.

1.3   Regulation

Expression of toyGenes occurs through the interaction with the toyPolymerase, which is a special kind of toyProtein (see Supplementary Figure S1). The toyPolymerase only has one interacting side (with sequence PHPH) and its folding energy is fixed to value -11.0: it is more stable than more than half the toyProteins. It is always present in the system. The toyPolymerase binds to promoters or to the right side of a toyProtein / toyDimer already bound to a promoter. When the toyPolymerase binds to a promoter, translation is directly activated and the corresponding toyGene is expressed (Supplementary Figure S5**a**). However, a more stable (lower energy) binding of a toyProtein or toyDimer to a promoter precludes the binding of the toyPolymerase. This inhibits the expression of the toyGene, except if the toyPolymerase binds to the right side of the toyProtein / toyDimer, in which case the toyGene can be expressed.

The minimal interaction rules that define $t_{\odot y}$LIFE dynamics endow toyProteins with a set of possible activities not included *a priori* in the rules of the model (see Supplementary Figure S5). For example, since the 4-toyN interacting site of the toyPolymerase cannot bind to all promoter regions —because some of these interactions have $E_{int} > E_{thr}$—, translation mediated by a toyProtein or toyDimer binding might allow the expression of genes that would otherwise never be translated. These toyProteins thus
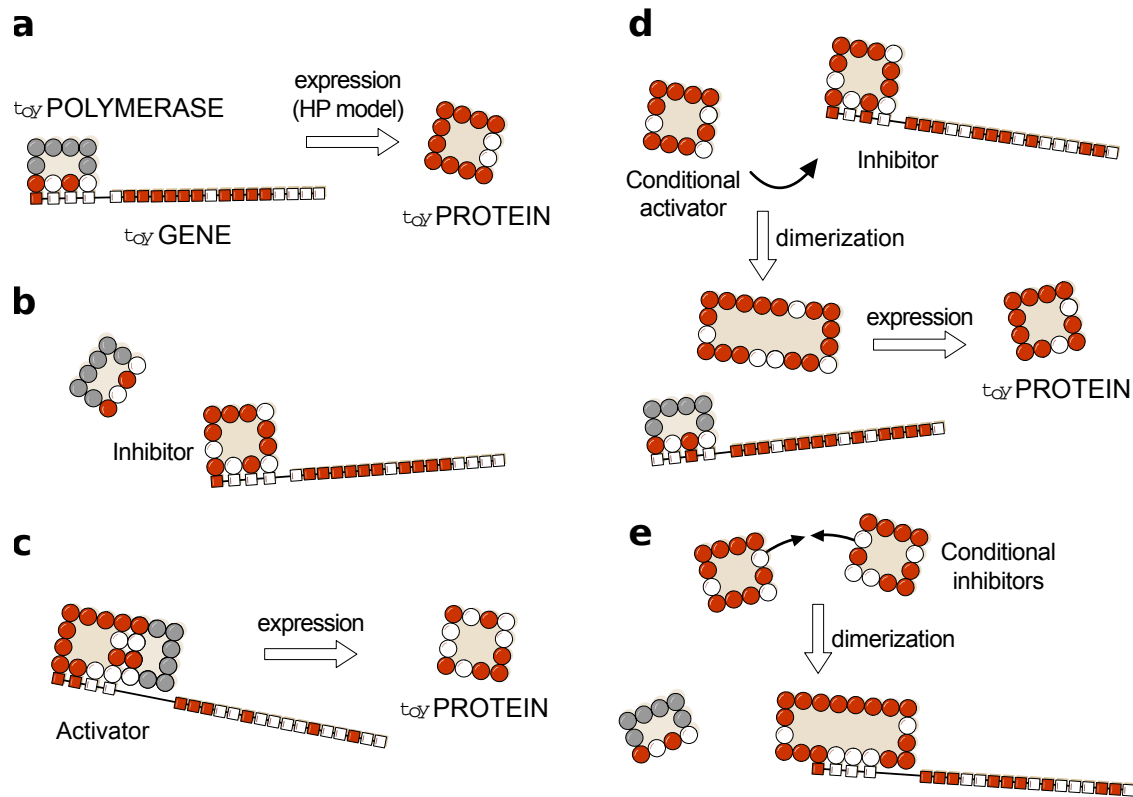
Supplementary Figure S5: **Regulatory functions in toyLIFE. (a)** A toyGene is expressed (translated) when the toyPolymerase binds to its promoter region. The sequence of Ps and Hs of the toyProtein will be exactly the same as that of the toyGene coding region. **(b)** If a toyProtein binds to the promoter region of a toyGene with a lower energy than the toyPolymerase does, it will displace the latter, and the toyGene will not be expressed. This toyProtein acts as an *inhibitor*. **(c)** The toyPolymerase does not bind to every promoter region. Thus, not all toyGenes are expressed constitutively. However, some toyProteins will be able to bind to these promoter regions. If, once bound to the promoter, they bind to the toyPolymerase with their rightmost side, the toyGene will be expressed, and these toyProteins act as *activators*. **(d)** More complex interactions —involving more elements— appear. For example, a toyProtein that forms a toyDimer with an inhibitor —preventing it from binding to the promoter— will effectively activate the expression of the toyGene. However, it does neither interact with the promoter region nor with the toyPolymerase, and its function is carried out only when the inhibitor is present. We call this kind of toyProteins *conditional activators*. **(e)** Two toyProteins can bind together to form a toyDimer that inhibits the expression of a certain toyGene. As they need each other to perform this function, we call them *conditional inhibitors*. As the number of genes increases, this kind of complex relationships can become very intricate.

act as activators (Supplementary Figure S5**c**). This process finds a counterpart in toyProteins that bind to promoter regions more stably than the toyPolymerase does, and therefore prevent gene expression — this happens if $E_{\text{int(PROT)}} + E_{\text{PROT}} < E_{\text{int(POLY)}} + E_{\text{POLY}}$. They are acting as inhibitors (Supplementary Figure S5**b**). There are two additional functions that could not be foreseen and involve a larger number of molecules. A toyProtein that forms a toyDimer with an inhibitor —preventing its binding to the promoter— effectively behaves as an activator for the expression of the toyGene. However, it interacts neither with the promoter region nor with the toyPolymerase, and its activating function only shows up when the inhibitor is present. This toyProtein thus acts as a conditional activator (Supplementary Figure S5**d**). On the other hand, two toyProteins can bind together to form a toyDimer that inhibits the expression of a particular toyGene. As the presence of both toyProteins is needed to perform this function, they behave as conditional inhibitors (Supplementary Figure S5**e**). This flexible, context-dependent behavior of toyProteins is reminiscent of phenomena observed in real cells [10], and permits the construction of complex toyGene Regulatory Networks (toyGRNs).

1.4 Metabolism

When a toyDimer is bound to a toyMetabolite, another toyProtein can interact with this complex and break it. This reaction will take place if the toyProtein can bind to one of the subunits of the toyDimer and the resulting complex has less total energy than the toyDimer. As with the rest of interactions, the catabolic reaction will only take place if this binding is unambiguous. As a result of this reaction, the toyDimer will be broken in two: one of the pieces will be bound to the toyProtein (forming a new toyDimer), and the other one will remain free. The toyMetabolite will break accordingly: the part of it that was bound to the first subunit will stay with it, and the other part will stay with the second subunit. Note that the toyMetabolite need not be broken symmetrically: this will depend on how the
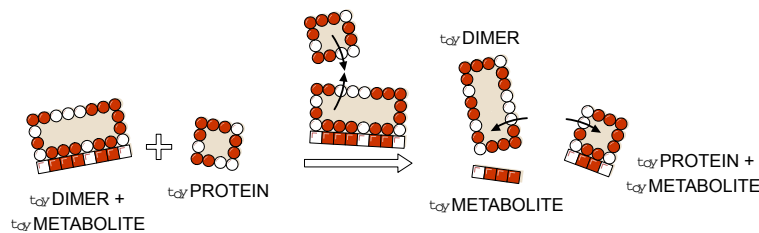
Supplementary Figure S6: **Metabolism in toyLIFE.** A toyDimer is bound to a toyMetabolite when a new toyProtein comes in. If the new toyProtein binds to one of the two units of the toyDimer, forming a new toyDimer energetically more stable than the old one, the two toyProteins will unbind and break the toyMetabolite up into two pieces. We say that the toyMetabolite has been catabolised.

toyDimer binds to it (Supplementary Figure S6).

1.5 Dynamics in t𝛼yLIFE

The dynamics of the model proceeds in discrete time steps and variable molecular concentrations are not taken into account. A step-by-step description of t𝛼yLIFE dynamics is summarised in Supplementary Figure S7. There is an initial set of molecules which results from the previous time step: toyProteins
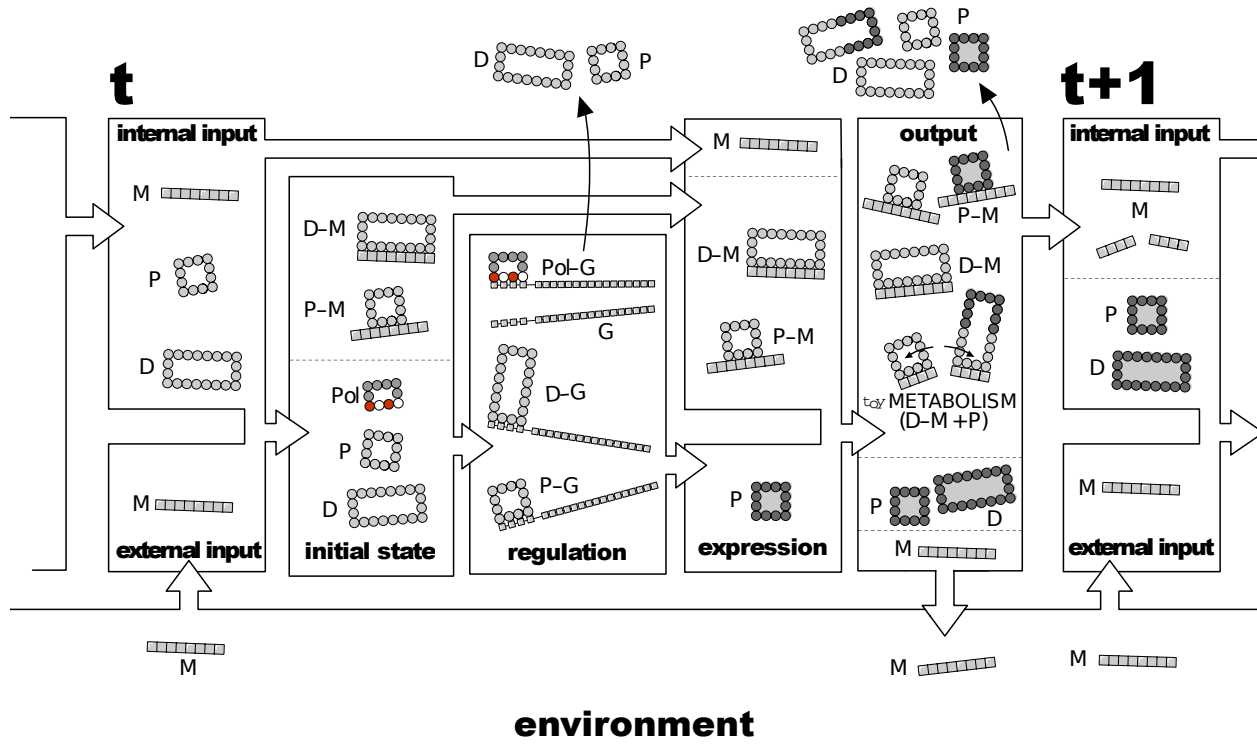
Supplementary Figure S7: **Dynamics of t𝛼yLIFE.** Input molecules at time step t are toyProteins (Ps) (including toyDimers (Ds)) and toyMetabolites, either produced as output at time step $t-1$ or environmentally supplied (all toyMetabolites denoted Ms). Ps and Ds interact with Ms to produce complexes P-M and D-M. Next, the remaining Ps and Ds and the toyPolymerase (Pol) interact with toyGenes (G) at the regulation phase. The most stable complexes with promoters are formed (Pol-G, P-G and D-G), activating or inhibiting toyGenes. P-Ms and D-Ms do not participate in regulation. Ps and Ds not in complexes are eliminated and new Ps (dark grey) are formed. These Ps interact with all molecules present and form Ds, new P-M and D-M complexes, and catabolise old D-M complexes. At the end of this phase, all Ms not bound to Ps or Ds are returned to the environment, and all Ps and Ds in P-M and D-M complexes unbind and are degraded. The remaining molecules (Ms just released from complexes, as well as all free Ps and Ds) go to the input set of time step $t+1$.

11

(including toyDimers and the toyPolymerase) and toyMetabolites, either endogenous or provided by the environment. These molecules first interact between them to form possible complexes (see Section 1.2) and are then presented to a collection of toyGenes that is kept constant along subsequent iterations. Regulation takes place, mediated by a competition for binding the promoters of toyGenes, possibly causing their activation and leading to the formation of new toyProteins. Binding to promoters is decided in sequence. Starting with any of them (the order is irrelevant), it is checked whether any of the toyProteins / toyDimers (including the toyPolymerase) available bind to the promoter —remember that complexes bound to toyMetabolites are not available for regulation—, and then whether the toyPolymerase can subsequently bind to the complex and express the accompanying coding region. If it does, the toyGene is marked as active and the toyProtein / toyDimer is released. Then a second promoter is chosen and the process repeated, until all promoters have been evaluated. toyGenes are only expressed after all of them have been marked as either active or inactive. Each expressed toyGene produces one single toyProtein molecule. There can be more units of the same toyProtein, but only if multiple copies of the same toyGene are present.

toyProteins / toyDimers not bound to any toyMetabolite are eliminated in this phase. Thus, only the newly expressed toyProteins and the complexes involving toyMetabolites in the input set remain. All these molecules interact yet again, and here is where catabolism can occur. Catabolism happens when, once a toyMetabolite-toyDimer complex is formed, an additional toyProtein binds to one of the units of the toyDimer with an energy that is lower than that of the initial toyDimer. In this case, the latter disassembles in favor of the new toyDimer, and in the process the toyMetabolite is broken, as already mentioned in Section 1.4 and Supplementary Figure S6. The two pieces of the broken toyMetabolites will contribute to the input set at the next time step, as will free toyProteins / toyDimers. However, toyProteins / toyDimers bound to toyMetabolites disappear in this phase —they are degraded—, and only the toyMetabolites are kept as input to the next time step. Unbound toyMetabolites are returned to the environment. This way, the interaction with the environment happens twice in each time step: at the beginning and at the end of the cycle.

2 Supplementary Figures
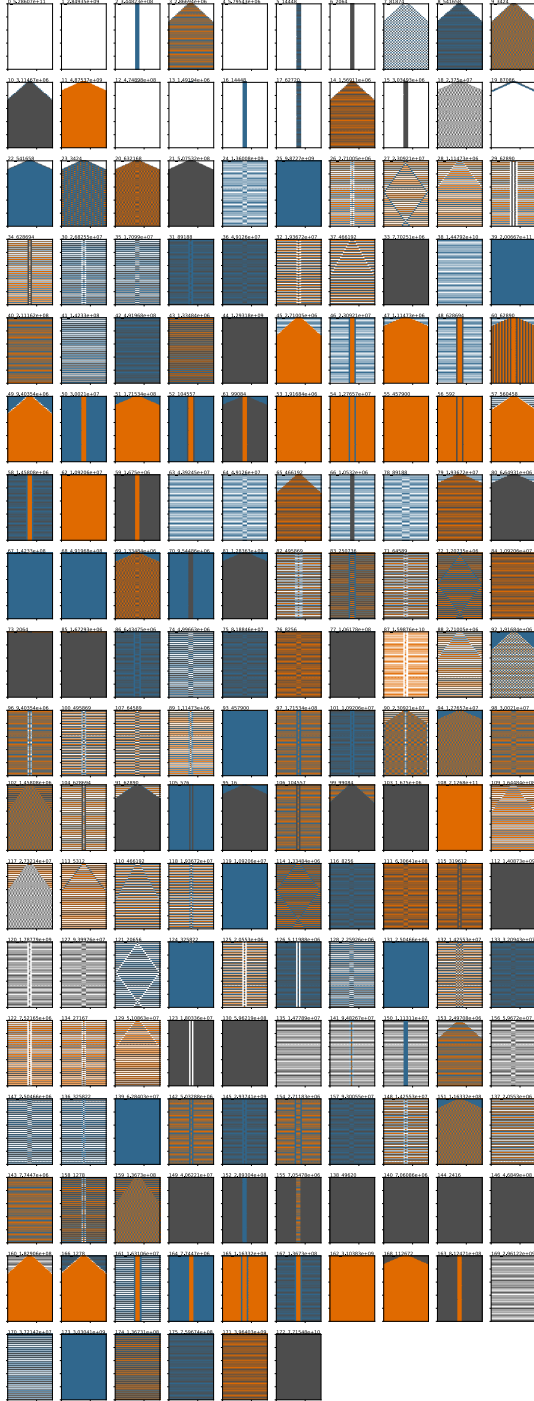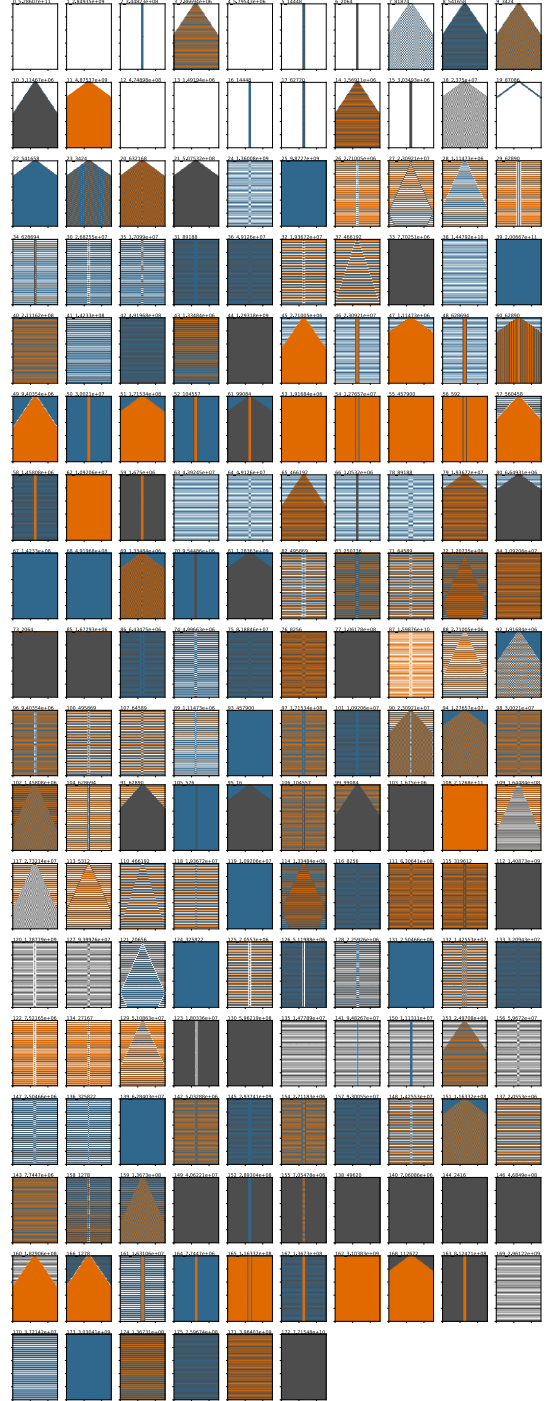
Supplementary Figure S8: **The same patterns are observed as we increase tissue size. a)** All patterns generated by toyLIFE genotypes when the tissue size is set to be 31 cells. The two numbers above each pattern represent the pattern's id and its abundance in genotype space. **b)** Same but with 51-cell tissues. The patterns are exactly the same, with the same abundances in genotype space.
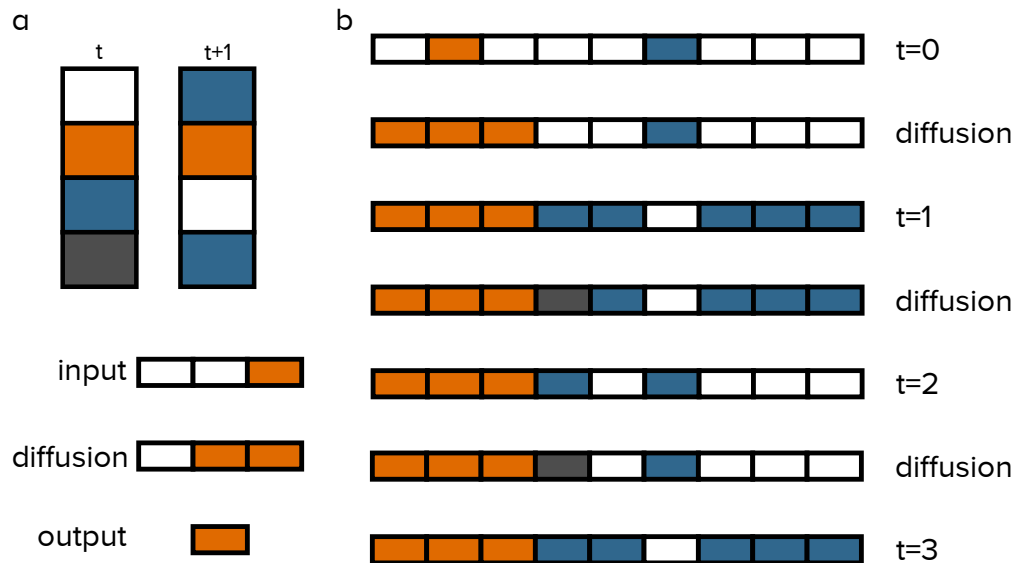
Supplementary Figure S9: **Obtaining the cellular automata rules from the truth table.** With the truth table in Figure 1b, we show how to construct the cellular automaton. As before, white means the cell is empty, blue that protein B is present, orange that protein A is present, and grey that both (or the dimer) are. **a** Suppose we want to compute the update rule for the triplet white-white-orange, i.e. the expression state of a white cell surrounded by white and orange. There is an intermediate diffusion step where protein A propagates to the central cell, and because protein A promotes its own expression, in time step $t+1$ the central cell will express protein A too. We repeat this process with each of the $4^3 = 64$ triplets. **b** As an example of how this rule works, we start with a 9-cell long tissue with one cell expressing protein A and another expressing protein B. In the diffusion step, protein A propagates to the adjoining cells. Thus, in time step $t=1$ three cells express protein A, while every remaining cell (except the one expressing protein B at $t=0$) is blue. In the next step, protein A again propagates to adjoining cells: note that now one of the cells is grey. This actually prevents the further propagation of protein A and this cell will always express protein B. The remaining tissue alternates between blue and white, as seen in Figure 1d. In practice, this process is not computed every time: the update rules for each cellular automaton are found following the procedure in **a** and then used directly to compute the evolution of the tissue.

I) 8.216×10^9

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	0

II) 10^6

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	1

III) 1.652×10^9

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	0
1	0	1	0
1	1	1	0

IV) 3×10^6

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	0
1	0	1	0
1	1	1	1

V) 9×10^6

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	1
1	0	1	0
1	1	0	0

VI) 1.71×10^8

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	1
1	0	1	0
1	1	0	1

VII) 3.0×10^7

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	1
1	0	1	0
1	1	1	0

VIII) 0.2×10^6

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	0	1
1	0	1	0
1	1	1	1

IX) 3.536×10^9

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	0

X) 7×10^6

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

XI) 1.971×10^{11}

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	0

XII) 1.9×10^7

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

XIII) 1.42×10^8

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	1
1	0	1	0
1	1	0	0

XIV) 10^6

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	1
1	0	1	0
1	1	0	1

XV) 4.92×10^8

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	0

XVI) 1.293×10^9

A(t)	B(t)	A(t+1)	B(t+1)
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Supplementary Figure S10: **There are sixteen GRNs that generate the pattern in Figure 3b.** Truth tables for all GRNs that generate the desired pattern. The number next to the label represents how many genotypes (binary sequences of length 40) are mapped into that particular GRN. Notice the wide variation in abundances.
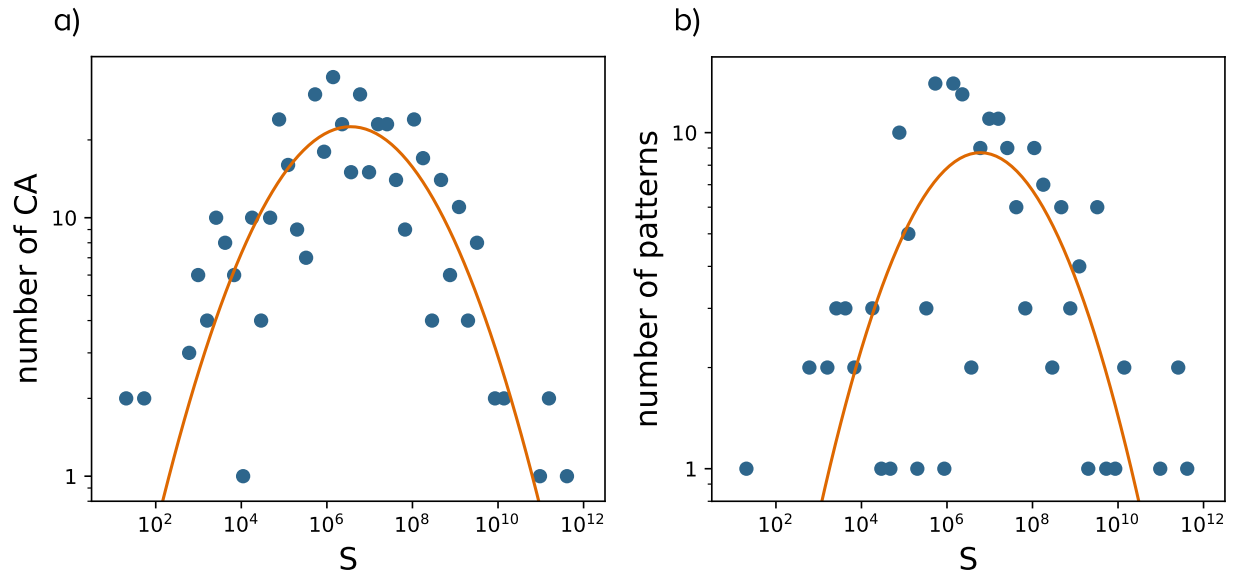
Supplementary Figure S11: **Phenotypic bias is observed in the distribution of abundances at all phenotypic levels. (a)** The distribution of abundances of cellular automata (CA) follows a log-normal law, just like the distribution of GRNs ($R^2 = 0.64$). **(b)** Likewise, the distribution of abundances of patterns can also be fitted by a log-normal distribution, although the fit is rather noisy ($R^2 = 0.42$), given that we only have 176 patterns to fit.
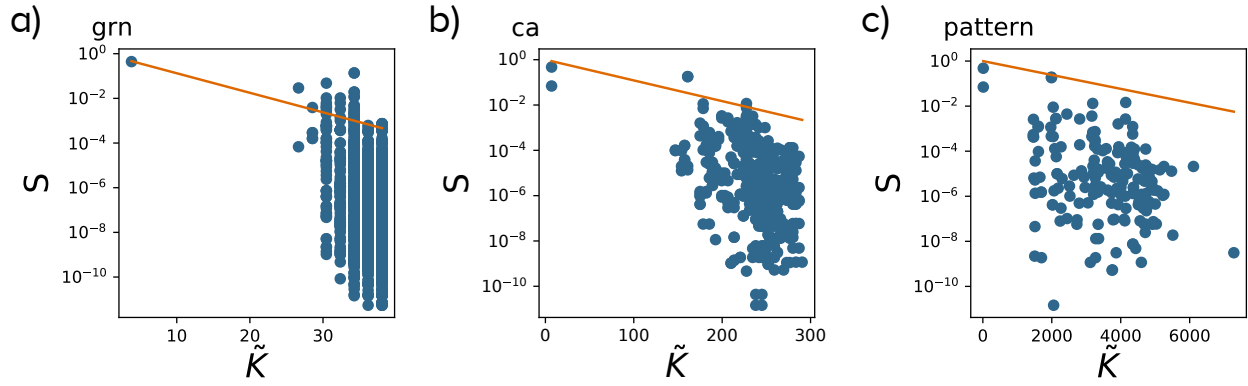
Supplementary Figure S12: **Simple phenotypes are more common in genotype space.** We approximated the algorithmic complexity ($\tilde{K}$) of GRNs (**a**), cellular automata (**b**) and patterns (**c**) following the work by Dingle *et al.* [11] (Methods), and plotted them against phenotype abundance (S). The disparity in lengths between the string representation of different phenotypic levels explains the difference in magnitude in the values of $\tilde{K}$: GRNs are represented as binary strings of length 12, cellular automata become binary strings of length 128 and patterns become binary strings of length $6,200$. Dingle *et al.* conjecture that many input-output maps have the property that simple outputs (as measured by their algorithmic complexity) should be mapped by more inputs. In our case, this would mean that simple phenotypes are more abundant in genotype space. This figure confirms this prediction for our three phenotypic levels. Lines represent the upper bound computed in [11], $S = 2^{-a\tilde{K}}$, with $a \approx \log_2 N / \max \tilde{K}$, where N is the number of phenotypes and the maximal $\tilde{K}$ is computed over all possible phenotypes (which is straightforward in our case as we know the complete maps). GRNs and cellular automata do not always lie below the upper bound. This could be explained because the results obtained by Dingle *et al.* rely on asymptotic approximations with long strings, but the strings coding these two phenotypic levels are not very long, so asymptotic approximations may fail. Another more likely possibility is that these systems have a degree of pseudo randomness to them that makes them appear complex when they are not.
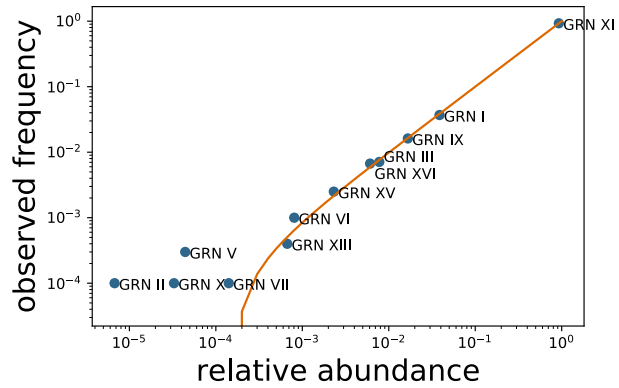
Supplementary Figure S13: **Equally fit GRNs appear as the endpoint of evolutionary simulations in proportion to their relative abundance in genotype space.** Although all sixteen GRNs are equally fit (see main text), evolutionary simulations in which populations undergo Wright-Fisher dynamics do not find every GRN with equal probability. On the contrary, those GRNs that are more abundant in genotype space appear more frequently as an endpoint of our simulations, in agreement with Refs. [12,13]. In fact, the fraction of times a given GRN is the endpoint of the simulations is almost exactly its abundance in genotype space relative to that of all sixteen GRNs. The discrepancies are a result of limited numerical sampling: we performed $10,000$ replicates of the evolution experiments. Linear fit is approximately $y = x$ ($R^2 \approx 1.0$).

References

[1] Arias, C. F., Catalán, P., Manrubia, S. & Cuesta, J. A. toyLIFE: a computational framework to study the multi-level organisation of the genotype-phenotype map. *Sci. Rep.* **4**, 7549 (2014).

[2] Catalán, P., Wagner, A., Manrubia, S. & Cuesta, J. A. Adding levels of complexity enhances both robustness and evolvability in a multi-level genotype-phenotype map. *J. Roy. Soc. Interface* **15**, 20170516 (2018).

[3] Li, H., Helling, R., Tang, C. & Wingreen, N. Emergence of preferred structures in a simple model of protein folding. *Science* **273**, 666–669 (1996).

[4] Dill, K. A. Theory for the folding and stability of globular proteins. *Biochemistry* **24**, 1501–1509 (1985).

[5] Aharoni, A. *et al.* The'evolvability'of promiscuous protein functions. *Nat. Genet.* **37**, 73–76 (2005).

[6] Amitai, G., Gupta, R. D. & Tawfik, D. S. Latent evolutionary potentials under the neutral mutational drift of an enzyme. *HFSP J.* **1**, 67–78 (2007).

[7] Khersonsky, O. & Tawfik, D. S. Enzyme promiscuity: a mechanistic and evolutionary perspective. *Ann. Rev. Biochem.* **79**, 471–505 (2010).

[8] Hayden, E. J., Ferrada, E. & Wagner, A. Cryptic genetic variation promotes rapid evolutionary adaptation in an RNA enzyme. *Nature* **474**, 92–95 (2011).

[9] Hoque, T., Chetty, M. & Sattar, A. Extended HP model for protein structure prediction. *J. Comput. Biol.* **16**, 85–103 (2009).

[10] Piatigorsky, J. *Gene Sharing and Evolution: the Diversity of Protein Functions* (Harvard University Press, 2007).

[11] Dingle, K., Camargo, C. Q. & Louis, A. A. Input–output maps are strongly biased towards simple outputs. *Nat. Comm.* **9**, 761 (2018).

[12] Schaper, S. & Louis, A. A. The arrival of the frequent: how bias in genotype-phenotype maps can steer populations to local optima. *PLoS ONE* **9**, e86635 (2014).

[13] Greenbury, S. F., Schaper, S., Ahnert, S. E. & Louis, A. A. Genetic correlations greatly increase mutational robustness and can both reduce and enhance evolvability. *PLoS Comput Biol* **12**, e1004773 (2016).