Predicting phenotype transition probabilities via conditional algorithmic probability approximations

Kamaludin Dingle,[1,2,3,*] Javor K. Novev,[1] Sebastian E. Ahnert,[1] and Ard A. Louis[4]

[1] *Department of Chemical Engineering and Biotechnology, Cambridge University, UK*
[2] *Department of Computing and Mathematical Sciences, California Institute of Technology, USA*
[3] *Centre for Applied Mathematics and Bioinformatics (CAMB), Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, Kuwait*
[4] *Rudolf Peierls Centre for Theoretical Physics, Department of Physics, Oxford University, UK*

(Dated: September 21, 2022)

Unravelling the structure of genotype-phenotype (GP) maps is an important problem in biology. Recently, arguments inspired by algorithmic information theory (AIT) and Kolmogorov complexity have been invoked to uncover *simplicity bias* in GP maps, an exponentially decaying upper bound in phenotype probability with increasing phenotype descriptional complexity. This means that phenotypes with very many genotypes assigned via the GP map must be simple, while complex phenotypes must have few genotypes assigned. Here we use similar arguments to bound the probability $P(x \to y)$ that phenotype x, upon random genetic mutation, transitions to phenotype y. The bound is $P(x \to y) \lesssim 2^{-a\tilde{K}(y|x)-b}$, where $\tilde{K}(y|x)$ is the estimated conditional complexity of y given x, quantifying how much extra information is required to make y given access to x. This upper bound is related to the conditional form of algorithmic probability from AIT. We demonstrate the practical applicability of our derived bound by predicting phenotype transition probabilities (and other related quantities) in simulations of RNA and protein secondary structures. Our work contributes to a general mathematical understanding of GP maps, and also facilitates the prediction of transition probabilities directly from examining phenotype themselves, without utilising detailed knowledge of the GP map.

Keywords: Genotype-phenotype maps; evolution; complexity; algorithmic probability

I. INTRODUCTION

An important challenge within theoretical biology is understanding the structure of *genotype-phenotype (GP) maps*, which dictate how gene sequences are translated into different biological forms, functions and traits, known as phenotypes. Elucidating GP map structure is essential to a proper understanding of evolution [1] because while random mutations occur at the genetic level, the effects of mutations occur at the level of the phenotype and therefore depend on GP map structure.

Several common properties of GP maps have been identified [2, 3], such as a strong bias in terms of how many genotypes are assigned to each phenotype [4–7] and high degrees of robustness to genetic mutations [8–11]. Significantly, GP map structure has been shown to strongly influence the trajectories and outcomes of evolution: Computer simulations of the evolution of RNA secondary structures (SS) [12, 13], protein complexes [6, 14], genetic circuits [15], among others [16], have shown that even in the presence of natural selection the bias arising from GP map structure can influence and even dominate outcomes. More significantly, for naturally occurring RNA shapes [9, 12, 17–22] and protein quaternary structures [23], the frequency in nature of different molecular shapes can be predicted from GP map biases. The way genotypes are associated to phenotypes via the GP map is also known to limit and constrain evolutionary accessibility of phenotypes [24–26].

Despite the importance of GP maps and the observed common properties across a variety of example maps, the theoretical underpinnings for these observations are not well developed. A recent approach to mathematically studying GP map structure is to use arguments inspired by *algorithmic information theory* [27–29] (AIT), a field of computer science that studies the information content and complexity of discrete patterns, structures, and shapes. Based on these arguments, in refs. [23, 30, 31] it was shown that the estimated information content, or *Kolmogorov complexity*, of a phenotype shape is closely connected to the probability that the shape appears on random sampling of genotypes. Moreover, high-probability phenotype shapes were shown to be simple, and more complex shapes were exponentially less probable, leading to the discovery of *simplicity bias* in GP maps. Interestingly, this complexity approach has enabled predicting the frequency with which biomolecule shapes appear in databases of natural biomolecules [23, 32]. More broadly, many studies have shown that employing AIT as a theoretical framework combined with estimates of Kolmogorov complexity can be fruitful natural sciences. Example applications include in thermodynamics [33–35], understanding the regularity of the laws of physics [36], entropy estimation [37, 38], classifying biological structures [39], evolution theory [40, 41], networks [42, 43], in addition to data analysis [44–46] and time series analysis [47, 48], among others [49].

Here, we extend the earlier work on simplicity bias in GP maps by utilising information complexity arguments to predict phenotype transition probabilities: We use AIT as a theoretical framework to derive probability-complexity relations in order to make predictions about the probability $P(x \to y)$ that phenotype y appears upon the introduction of a single random genetic point mutation in a genotype coding for the phenotype x. We show that $P(x \to y)$ is fundamentally related to the conditional complexity of y given x, which measures how much information is required to produce

* Correspondence: dinglek@caltech.edu

phenotype y given access to x. We derive an upper bound equation for this prediction and test it computationally on models of RNA and protein secondary structure, finding high quantitative accuracy.

II. NULL MODELS FOR TRANSITION PROBABILITIES

A. Problem set-up

We will focus on GP maps which have some large (finite) number N_g of discrete genotype sequences, a large number of N_p of possible phenotypes, each of which is designable (i.e. has at least one genotype). Further, we will assume that there are many more genotypes than phenotypes, such that $1 \ll N_p \ll N_g$, which is a common property of well-studied GP maps [3]. The GP map will be denoted f. The phenotypes are assumed to be some kind of discrete pattern, sequence, or shape, or at least one that could be discretised. For example, a protein quaternary structure can be represented as a discrete graph of nodes [32], and a continuous chemical concentration-time curve can also be discretised in a number of ways [30]. We assume the map is deterministic, such that each genotype maps consistently to only one phenotype.

As an example of such a GP map, we can take the RNA sequence-to-secondary structure map for which a genotype of length L featuring four nucleotides (A, U, C, and G) has $N_g = 4^L$ possible sequences. Also, $N_p \sim 1.8^L$ [20] so that $1 \ll N_p \ll N_g$ even for modest L, and the phenotypes can be represented as discrete sequences because RNA SS are commonly given in dot-bracket form, consisting of a string of L symbols defining the bonding pattern of the molecule. In contrast to this RNA example, we are not considering GP maps with only few phenotypes such as whether a patient does or does not have cancer, for which there are only two possible phenotypes, and 'has cancer' is not a discrete pattern or shape.

We will write $P(x)$ for the probability that phenotype x appears when uniformly randomly sampling a genotype out of the full collection of N_g genotypes. $P(x)$ will be called the *global frequency* of x. Although the average probability will be $1/N_p$, possibly for some phenotypes $P(x) \gg 1/N_p$ due to bias, and also possibly $P(x) \ll 1/N_p$ for some phenotypes. The *neutral set* (NS) of x is the set of genotypes that map onto x. If we pick one random genotype g from the NS of x, and make a random single point mutation such that we have a new genotype g', we will call the resulting phenotype y. It is possible that phenotypes x and y are the same or different, because g' may possibly be in the NS of x.

If $x = y$, then we designate the mutation as neutral. We will define $P(x \to y)$ as the transition probability that a randomly selected genotype from the NS of x, upon a single random mutation, yields the phenotype y. Note that we will still use the word 'transition' even if $x = y$. A phenotype is called *robust* to mutations if the probability that $y = x$ is high, i.e. the phenotype typically remains unchanged after a random mutation. The high robustness of phenotypes to genetic mutations is essential to life, and evolution (at least as we know it) would not be able to proceed without it [8, 50]. The origin of high robustness has been seen as something of a mystery, however [50]. Additionally, it has been noticed in several GP maps that the robustness scales with the logarithm of the global frequency of a phenotype, as opposed to scaling with the global frequency itself which would be expected from a random null model. The cause of this general logarithmic scaling is presently not fully explained, although an abstract model of GP maps has been used to study this property [51]. In a future study we will look in detail at robustness.

The problem of estimating $P(x \to y)$ is the main focus of the work, and in particular relating this quantity to the relative information contents of x and y.

B. Null models of transition probability

How can we estimate the transition probability $P(x \to y)$? If we have access to data recording the frequency of transitions in simulations, then we could directly estimate $P(x \to y)$ from those data by counting the number of time x transitioned to y as a fraction of all transitions starting with x. It may also be possible to arrive at an estimate of $P(x \to y)$ by examining details of the map and the particular phenotype x. However, what we are interested in here is a general method for predicting $P(x \to y)$, that does not rely on using past frequency data, or details of the map. Indeed, we are interested in general properties of GP maps which will both help to develop a theory of GP maps, and also be useful for other maps for which we neither have data nor a clear understanding of exactly how the phenotype arises from the genotypes. In a sense we are interested in *a priori* prediction for $P(x \to y)$ which is based only on the patterns in x and y. At first sight, this goal may not appear possible, because $P(x \to y)$ *will* depend on the details of the map. However, we will go on to argue in this current work that even without knowing details of the map, and without recourse to historical frequency data, non-trivial predictions for the transition probabilities can be derived. In another sense, we are interested in establishing a good null model for $P(x \to y)$, which could serve as a starting point for predictions about transitions. In this connection, we now consider some possible null models and weigh up their merits.

Perhaps the simplest null model expectation is that the transition probability is

$$P(x \to y) = 1/N_p \tag{1}$$

which corresponds to a maximum-entropy estimate, assigning a uniform probability to each possible outcome y. However, this has a limitation which is that a common property of GP maps is bias (described above), and so it seems reasonable to expect some degree of non-uniformity in $P(x \to y)$. Further, the high levels of robustness discussed above do not accord with this uniform distribution model. From here onwards, we will assume that the distribution $P(x \to y)$ over the possible values of y is strongly non-uniform (biased).

Another simple null model for $P(x \to y)$ was proposed in ref. [13]

$$P(x \to y) \approx P(y) \tag{2}$$

for $y \neq x$. This null-model prediction is correct if genotypes are randomly assigned to phenotypes, with no correlations

between genotypes or phenotypes. While the approximation in Eq. (2) is straightforward and was observed to be quite accurate on average [13], it also has limitations. Firstly, as pointed out in ref. [30] many GP maps have fixed and somewhat simple rule sets by which genotypes are assigned to phenotypes (technically, they are $O(1)$ complexity maps). Hence, these maps do not randomly assign genotypes, but assign them with a definite structure and pattern, which is likely to produce some clear patterns in genotype architecture. Secondly, it is intuitively reasonable that phenotype x will transition to some y which is similar or even the same as x. The logic being that one single point mutation represents a small change to the genotype, and consequently a small change to the phenotype appears to be a rational null assumption. Of course this assumption that GP maps are roughly 'continuous' in the mathematical sense of the word does not always hold because some (well-chosen) mutations may drastically change the phenotype, but nonetheless the assumption has intuitive appeal and may typically hold. Greenbury et al [50] have also suggested that transitions are more likely to be to similar phenotypes (with the caveat that it must be possible for enough genotypes to be sampled), arguing via genetic correlations in GP maps. Hence Eq. (2) has limitations as a null model.

In order to improve on Eq. (2), we would like to incorporate the 'similar phenotypes' notion in a formal way, which will lead to a new null model that we propose. To do this we first need to survey some pertinent theoretical background.

III. ALGORITHMIC INFORMATION THEORY

A. Kolmogorov complexity

Developed within theoretical computer science, *algorithmic information theory* [27–29] (AIT) connects computation, computability theory, and information theory. The central quantity of AIT is *Kolmogorov complexity*, $K(x)$, which measures the complexity of an individual object x as the amount of information required to describe or generate x. More formally, the Kolmogorov complexity $K_U(x)$ of a string x with respect to a universal Turing machine [52] (UTM) U, is defined [27–29] as

$$K_U(x) = \min_p \{|p| : U(p) = x\} \tag{3}$$

where p is a binary program for a prefix (optimal) UTM U, and $|p|$ indicates the length of the (halting) program p in bits. Due to the invariance theorem [49] for any two optimal UTMs U and V, $K_U(x) = K_V(x) + O(1)$ so that the complexity of x is independent of the machine, up to additive constants. Hence we conventionally drop the subscript U in $K_U(x)$, and speak of 'the' Kolmogorov complexity $K(x)$. Despite being a fairly natural quantity, $K(x)$ is uncomputable, meaning that there cannot exist a general algorithm that for any arbitrary string returns the value of $K(x)$. Informally, $K(x)$ can be defined as the length of a shortest program that produces x, or simply as the size in bits of the compressed version of x. If x contains repeating patterns like $x = 1010101010101010$ then it is easy to compress, and hence $K(x)$ will be small. On the other hand, a randomly generated bit string of length n is highly unlikely to contain any significant patterns, and hence can only be described via specifying each bit separately without any compression, so that $K(x) \approx n$ bits. $K(x)$ is also known as *descriptional complexity*, *algorithmic complexity*, and *program-size complexity*, each of which highlight the idea that $K(x)$ measures the amount of information required to describe or generate x precisely and unambiguously. Note that Shannon information and Kolmogorov complexity are related [53], but differ fundamentally in that Shannon information quantifies the information or complexity of a random source, while Kolmogorov complexity quantifies the information of individual sequences or objects. More details and technicalities can be found in standard AIT references [49, 54–56] and a book aimed at natural scientists [57].

B. Algorithmic probability

An important result in AIT is Levin's [58] coding theorem, establishing a fundamental connection between $K(x)$ and probability predictions. Building on Solomonoff's discovery of *algorithmic probability* [27, 59], the coding theorem loosely states that random programs are much more likely to produce simple rather than complex outputs. Mathematically, Levin's coding theorem [58] states that

$$P(x) = 2^{-K(x)+O(1)} \tag{4}$$

where $P(x)$ is the probability that (prefix optimal) UTM U generates output string x on being fed random bits as a program. Thus, high-complexity outputs have exponentially low probability, and simple outputs must have high probability. $P(x)$ is also known as the *algorithmic probability* of x.

The *conditional coding theorem* [60] states that the probability $P(y|x)$ of generating string y with UTM U given access to string x as side information is

$$P(y|x) = 2^{-K(y|x)+O(1)} \tag{5}$$

Notice that outputs with high probability here must have small conditional complexity $K(y|x)$. To have $K(y|x)$ small means that either y is simple itself, or it is similar to x. To see this, consider that if y is simple, then $K(y)$ is small, then so too is $K(y|x)$, hence $P(y|x)$ is big. Also, if y is similar to x — for example they share common subsequences — then $K(y|x)$ will be small, and $P(y|x)$ will be big.

C. Simplicity bias

Eq. (4) as well as many other AIT results cannot be straightforwardly applied to typical natural systems of interest in engineering and sciences, due to the fact that: (1) Kolmogorov complexity is uncomputable and so cannot be calculated exactly; (2) the theory is asymptotic, valid only up to $O(1)$ terms; (3) the theory is largely based on UTMs, which are seldom present in nature; and (4) the coding theorem assumes infinite purely uniform random programs, which do not exist in nature. Despite these points, many of the fundamental ideas of AIT do not depend on UTMs and uncomputable quantities, but have analogs in computable settings.

For example, the fundamental fact that most sequences are incompressible results from basic counting principles, and the fact that short programs appear with higher probability in many prefix codes holds outside of UTM settings. Further, because a UTM can simulate any computable function, we can always bound the behaviour of a computable function with known limits of the behaviour of a UTM. In addition, Kolmogorov complexity can be bounded by lossless compression algorithms. With these points in mind, AIT can be used as a theoretical framework to guide the derivation of mathematical predictions for real-world systems, which we might call AIT-inspired arguments.

Adopting the methodology of AIT-inspired arguments, Dingle et al [30] studied coding theorem-like behaviour and algorithmic probability for (computable) real-world input-output maps. This led to their observation of *simplicity bias* (SB), governed by the equation

$$P(x) \leq 2^{-a\tilde{K}(x)-b} \tag{6}$$

where $P(x)$ is the (computable) probability of observing output x on random choice of inputs, and $\tilde{K}(x)$ is the approximate Kolmogorov complexity of the output x. In words, SB means complex outputs from input-output maps have lower probabilities, and high probability outputs are simpler. The constants $a > 0$ and b can be fit with little sampling and often even predicted without recourse to sampling [30].

Examples of systems exhibiting SB are wide-ranging, and include molecular shapes such as protein structures and RNA [23], outputs from finite state machines [31], as well as models of financial market time series and ODE systems [30], among others.

The ways in which SB differs from Levin's coding theorem include that it does not assume UTMs, uses approximations of complexities, and for many outputs $P(x) \ll 2^{-K(x)}$. Hence, the abundance of low-complexity, low-probability outputs [31, 61] is a signature of SB. Further, the upper bound was intended to apply to more general notions of complexity such as the number of stems on an RNA molecule structure, not strictly Kolmogorov complexity.

A full understanding of exactly which systems will, and will not, show SB is still lacking, but the phenomenon is expected to appear in a wide class of input-output maps, under fairly general conditions. Some of these conditions were suggested in ref. [30], including (1) that the number of inputs should be much larger than the number of outputs, (2) the number of outputs should be large, and (3) that the map should be 'simple' (technically of $O(1)$ complexity) to prevent the map itself from dominating over inputs in defining output patterns. Indeed, if an arbitrarily complex map was permitted, outputs could have arbitrary complexities and probabilities, and thereby remove any connection between probability and complexity. Finally (4), because many AIT applications rely on approximations of Kolmogorov complexity via standard lossless compression algorithms [62, 63] (but see [64, 65] for a fundamentally different approach), another proposed condition is that the map should not generate pseudo-random outputs like $\pi = 3.1415...$, which standard compressors cannot handle effectively. The presence of such outputs may yield high-probability outputs which appear 'complex' hence apparently violating SB, but which are in fact simple.

IV. SIMPLICITY BIAS IN TRANSITIONS

A. Simplicity bias: conditional form

Just as for the original coding theorem, the conditional coding theorem in Eq. (5) cannot be directly applied to practical real-world systems, such as making estimates for phenotype transition probabilities. So we now derive a conditional form of the simplicity bias equation Eq. (6), which we subsequently apply to phenotype transition probabilities.

In our finite and computable setting we can proceed with the following argument: Given x and the GP map f, we can enumerate genotypes and record those that map onto x, i.e. we can generate the NS of x. Next, we make a list $\mu_1(x)$ consisting of all possible 1-point mutations of the sequences in the NS of x. Denote the number of entries in the list $\mu_1(x)$ with $|\mu_1(x)|$. The value of $|\mu_1(x)|$ is related to the size of the genotype alphabet α and the length L of the genotype; specifically $|\mu_1(x)|$ will be $(\alpha-1)L$ multiplied by the number of genotypes in the NS of x. Notice that $\mu_1(x)$ may have repeated sequences. For example, if the NS of x was made up of the binary strings [001,100] then μ_1 = [101,011,000, 000,110,101] and so $|\mu_1(x)| = 6$, even though the number of unique sequences in $\mu_1(x)$ is only 4. On the other hand, for the NS given by [001,000], we have $|\mu_1(x)| = 6$ and also 6 unique members.

Next, we enumerate all genotypes in the $\mu_1(x)$ and use the map to make a list of length $|\mu_1(x)|$ of the corresponding phenotypes. Counting the frequencies of each phenotype in this list gives a discrete (and computable) probability distribution $P(x \to y)$ for all possible phenotypes y. (Note that $P(x \to y)$ may be 0 for some or many of the possible y, meaning that it might not be possible to transition directly from phenotype x to phenotype y.) In this manner, given x and the map f we can use a Shannon-Fano code [66] to make a binary code using $\log_2(1/P(x \to y)) + O(1)$ bits to describe each y. (In fact this Shannon-Fano code will be a prefix code, meaning that no code forms the prefix of another code; see [66] for more details). So we have

$$K(y|x, f) \leq \log_2(1/P(x \to y)) + O(1) \tag{7}$$

Rearranging yields the upper bound,

$$P(x \to y) \leq 2^{-K(y|x,f)+O(1)} \tag{8}$$

B. Complexity of the GP map

The complexity of the GP map f is an important quantity, which we discuss in this section.

If genotypes are randomly assigned to phenotypes, then the map f will contain a large amount of information due to literally storing all the genotype assignments. Because there are $N_p^{N_g}$ different ways to assign N_g genotypes to N_p phenotypes (assuming for simplicity that phenotypes do not necessarily have to have at least one genotype assigned), this means that we can estimate the Kolmogorov complexity $K(f)$ of the map f if genotypes are randomly assigned to phenotypes: $K(f) \approx \log_2(N_p^{N_g}) = N_g \log_2(N_p)$, which is typically a very large number, due to N_g typically being exponential in L. To give a reference point, note that typical

phenotypes x have $K(x) \approx \log_2(N_p)$, which is far smaller. Because of the large size of $K(f)$, under the assumption of random assignment, the quantity $K(y|x, f)$ in Eq. (8) may be very different to $K(y|x)$ and hence the probability $P(x \to y)$ may have little relation to the complexities of x, y or their conditional complexity. In this case, predicting $P(x \to y)$ just from these complexities will not be feasible. More generally, if the map f is allowed to have arbitrary complexity values, then f could be chosen such that $P(x \to y)$ takes arbitrary values, and hence will be very hard to predict.

Fortunately, many GP maps are not random, but in fact have simple fixed rule-sets for determining how genotypes are assigned to phenotypes [30]. For example, there is a fixed-size algorithm for computationally predicting an RNA SS from a given nucleotide sequence. This algorithm does not depend on the size L of the sequence. The algorithm may not appear to be 'simple' in the everyday sense of word due to the various biophysical rules and constants that underlie it, but in terms of information content we can write $K(f) = O(1)$, and hence it is 'simple' in this sense.

If we restrict our attention to GP maps for which f is of fixed complexity, i.e. $K(f) = O(1)$, then this means that $K(y|x, f) \approx K(y|x)$ so that Eq. (8) becomes

$$P(x \to y) \leq 2^{-K(y|x)+O(1)} \tag{9}$$

and we see that this upper bound depends only on the phenotypes x and y. So the complexity of the map f is an important quantity which either does or does not allow predictions to be made just using conditional complexities.

Note that for many maps the complexity sits between the extreme values of a completely random assignment and an $O(1)$ complexity map. Presumably, the closer the map complexity is to $O(1)$, the stronger the connection between $P(x \to y)$ and $2^{-K(y|x)}$. This reasoning would predict that Eq. (9) applies approximately to any 'low'-complexity GP maps, not specifically $O(1)$ maps. We leave this to future work to explore.

C. Approximation of the upper bound

Because Kolmogorov complexity is uncomputable, in practice we must use approximations, such as real-world compression algorithms [49]. Following the approximation and scaling arguments of ref. [30], we can write an approximate form of Eq. (9)

$$P(x \to y) \lesssim 2^{-a\tilde{K}(y|x)-b} \tag{10}$$

which is a weaker form of the full AIT conditional coding theorem [60] given in Eq. (5). The term $\tilde{K}(y|x)$ is an approximation of the conditional Kolmogorov complexity $K(y|x)$, which we will calculate according to the complexity Lempel-Ziv [62] estimate used earlier [30, 31] and also scale the complexity values so that $0 \lesssim \tilde{K}(y|x) \lesssim \log_2(N_p)$ as described in Appendix (A 3). To estimate the conditional complexity $\tilde{K}(y|x)$, we employ the approximation (as used earlier [67]) that

$$\tilde{K}(y|x) \approx \tilde{K}(xy) - \tilde{K}(x) \tag{11}$$

where $\tilde{K}(xy)$ is the compressed length of the concatenation of strings x and y. For example, if $x =$`ABC` and $y =$`XY` then $xy =$`ABCXY`. Note that for true prefix Kolmogorov complexity the relation $K(y|x) \approx K(x, y) - K(x)$ only holds to within logarithmic terms [49], but that is close enough for our purposes.

The constant a may depend on the map, but not on the phenotype x. If the complexity $\tilde{K}(y|x)$ is scaled properly (Methods), then $a = 1$ is the default prediction. Otherwise, a might have to be fit to the data. The main requirement for scaling properly is having a reasonably accurate estimate of the number $N_y(x)$ of phenotypes y such that $P(x \to y) > 0$, i.e., the number of accessible phenotypes via a single point mutation from x (Methods). If $N_y(x)$ is known a priori or can be estimated a priori, then $N_y(x)$ can be used for a priori prediction of $P(x \to y)$. Otherwise, if random genotype sampling is employed, then simply counting the number of different y phenotypes observed in sampling is one way to estimate $N_y(x)$. Naturally, this counting method will be more accurate for larger samples, and may produce very low underestimates of $N_y(x)$ for small sample sizes.

The constant b has default value $b = 0$ [30], but can also be found by fitting to the data if necessary. Looking at the examples of simplicity bias presented in the literature to date, it appears that $b = 0$ often works very well.

It follows that in practice, provided some reasonable estimate of the number of accessible phenotypes $N_y(x)$ is known and hence complexities are scaled well, then Eq. (10) reduces further to the practically applicable relation

$$P(x \to y) \lesssim 2^{-\tilde{K}(y|x)} \tag{12}$$

allowing transition probabilities to be made just based on phenotype conditional complexities.

D. The bound is close with high probability

Based on the arguments in [31], we expect the upper bound Eq. (10) (and also Eq. (12)) to be tight for x, y pairs which are generated by random genotypes. That is, for a phenotype x generated by a random genotype, and y subsequently arising from a random mutation, we expect $P(x \to y) \approx 2^{-a\tilde{K}(y|x)-b}$ with high probability, as opposed to the right-hand side being only a loose upper bound.

On the other hand, the ubiquity of low-complexity, low-probability outputs [31, 61] suggests that for many y we may have $P(x \to y) \ll 2^{-a\tilde{K}(y|x)-b}$. Such phenotypes y are those that have low conditional complexity, yet at the same time appear with low probability due to map-specific constraints and biases. See ref. [61] for an in-depth discussion of this low-complexity, low-probability phenomenon.

E. Size of the genotype alphabet

For the bound of Eq. (10) to have predictive value on point mutations, we suggest that the size of the genotype alphabet α should be small, in the sense of $\alpha \ll L$. This is not a very onerous condition, and is in fact quite naturally satisfied. For RNA, $\alpha = 4$ and even for protein sequences there are

20 normally occurring amino acids, so that $\alpha = 20$, while L in natural RNA and proteins is much larger than these small values. Nonetheless it is worth specifying for the sake of completeness.

If α is large and L is small, then even one single point mutation will significantly randomise the genotype and the correlation between the original genotype g and the mutated version g' will be much reduced, making the relationship between phenotypes x and y comparatively weak. On the other hand, if α is small and L is large (the more common case) then a single mutation will not alter the genotype strongly, so that g and g' will share a lot of information and be highly correlated. Hence we can expect x and y to be related, and follow Eq. (10).

Additionally, for the bound of Eq. (10) to have predictive value, the number of mutations should be small: With 1 or a small number of mutations, the correlations between genotypes remains higher, but for more and more mutations, the correlation becomes very weak, and the conditional complexity becomes irrelevant.

F. When is $P(y)$ a good predictor of $P(x \to y)$?

From AIT we know that almost all pairs of phenotypes x and y share almost no information, in other words $K(x, y) \approx K(x) + K(y)$, so that

$$K(y|x) \approx K(y) \tag{13}$$

From this we can infer that for almost all pairs of phenotypes x and y the conditional complexity $K(y|x)$ in Eq. (8) can be replaced with just $K(y)$, and so the equation becomes

$$P(x \to y) \lesssim 2^{-K(y)} \tag{14}$$

for almost all y.

The preceding argument suggests that for most outputs y, the phenotype x may be largely irrelevant in estimating the probabilities $P(x \to y)$. However, this statement comes with the caveat that nearly all the probability mass is likely to be associated to only a small fraction of the possible outputs, those for which $K(y|x)$ is low. In other words, with high probability $2^{-K(y)}$ may not be a good estimator of $P(x \to y)$, even if it is for most outputs y. Indeed, when $K(y|x) \approx 0$ but $K(y) \gg 0$, meaning that y is complex and very similar to x then we might expect

$$P(x \to y) \gg 2^{-K(y)+O(1)} \geq P(y) \tag{15}$$

and hence $P(y)$ is not a good approximator of $P(x \to y)$, discording with Eq. (2). Potentially these types of phenotypes with $K(y|x) \approx 0$ may absorb much of the probability mass. For complex outputs x, even though they are rare, it could be that the majority of the y for which $P(x \to y)$ is non-zero are transitions of this type.

Another important case in which conditioning on x does not have much effect on the complexity or probability is the case that x is very simple. If x is very simple, then $K(x) \approx 0$ and $K(y|x) \approx K(y)$ for any y, and so we can also predict that Eq. (14) will hold when x is very simple:

$$K(x) \approx 0 \Rightarrow K(y|x) \approx K(y) \Rightarrow P(x \to y) \lesssim 2^{-K(y)} \tag{16}$$

This case is important because in maps with simplicity bias, very simple outputs may be the most likely (and also common in biology [23]), hence this type of transition is likely to be quite common. However, for more complex x it may be that $\tilde{K}(y|x)$ and $\tilde{K}(y)$ are different with high probability.

Whether Eq. (14) holds due to the fact that almost all pairs x and y share no information, or due to the fact that x is very simple, it is interesting to further make the rough approximation that $2^{-K(y)}$ is a good estimator of $P(y)$, then this suggests that for most phenotypes y, we might expect

$$P(x \to y) \lesssim P(y) \tag{17}$$

to hold roughly, which almost recovers the null model of $P(x \to y) \approx P(y)$ described in Eq. (2)

It is noteworthy that when Greenbury et al [50] investigated transition probabilities and concluded that $P(x \to x) \approx P(y)$ is a good estimator, or $\phi_{qp} \approx f_p$ in their terminology, they chose the phenotypes of the highest or second-highest probability as the starting phenotype x. For a map with simplicity bias these high-probability phenotypes would be among the simplest ones and have $K(x) \approx 0$, which is precisely the scenario in which we expect $P(x \to y)$ to be most closely related to $P(y)$. It is therefore possible that this simple prediction will break down when the starting phenotype x has a lower probability, or similarly if x has high complexity.

G. Predicting which of $P(x \to y_i)$ or $P(x \to y_j)$ is higher

Another quantity that may be predicted using the preceding theory is the ratio of probabilities for transitioning from one phenotype to different alternative phenotypes. In this section, we describe a method for such predictions, which we test numerically below.

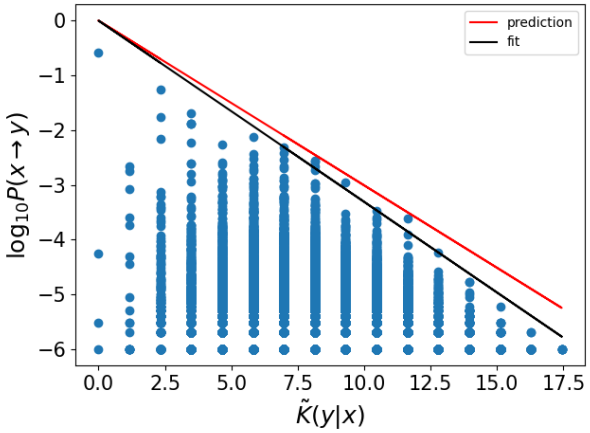
Call y_i the resulting phenotype after a single point mutation to a randomly chosen genotype in the NS of x. Call y_j the resulting phenotype after a single point mutation to another independently chosen random genotype, also in the NS of x. We can use the preceding theory to predict which of the two phenotypes y_i and y_j has a higher probability directly from complexity estimates. This is interesting because it is often valuable to know whether $P(x \to y_i) > P(x \to y_j)$ or $P(x \to y_i) < P(x \to y_j)$, rather than trying to guess the exact values of $P(x \to y_i)$ and $P(x \to y_j)$. Fortunately, the constants a and b are not required for predicting this via Eq. (10) because only the relative value of $\tilde{K}(y_i|x)$ and $\tilde{K}(y_j|x)$ determines whether $2^{-a\tilde{K}(y_i|x)-b}$ or $2^{-a\tilde{K}(y_j|x)-b}$ is larger. So even if we could not estimate a or b accurately, we could still make a prediction about which phenotype is more likely to arise through a point mutation of x. The approach we use is to predict:

$P(x \to y_i) > P(x \to y_j)$	`if`	$\tilde{K}(y_i\|x) < \tilde{K}(y_j\|x)$
$P(x \to y_i) < P(x \to y_j)$	`if`	$\tilde{K}(y_i\|x) > \tilde{K}(y_j\|x)$
$P(x \to y_i) \geq P(x \to y_j)$	`w/prob 0.5 if`	$\tilde{K}(y_i\|x) = \tilde{K}(y_j\|x)$

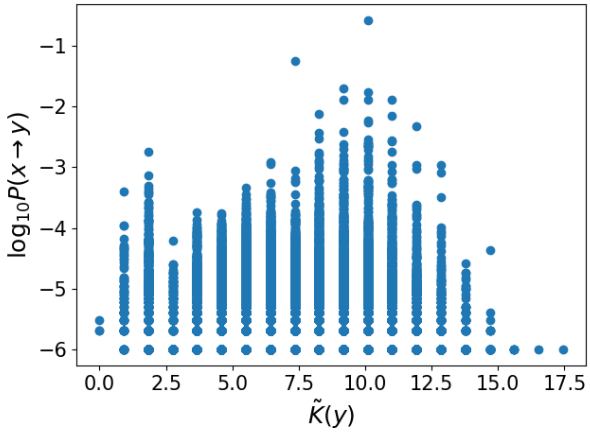
For the last condition, $\tilde{K}(y_i|x) = \tilde{K}(y_j|x)$, a random number (e.g., coin flip) is chosen and used to predict whether $P(x \to y_i) \geq P(x \to y_j)$ or not, each with 50% probability. According to this prediction protocol, if complexity had no

$\tilde{K}(y\|x)$	$P(x \to y)$	y
0.0	0.26	.(((...)))..((((.((..((....)).)))))).... *
1.2	0.0022	(((....)))..((((.((..((....)).))))))....
2.3	0.056	.....((((.((..((....)).))))))....
3.5	0.020	.(((...)))..((((.(((.....))))))....
4.7	0.0056	.(((...)))..(((.((..((....)).))))))....
5.8	0.0075	.(((...)))..(((((....((....))..))))))....
7.0	0.0048	.(((...)))((((((.((..((....)).))))))))..
8.1	0.0028	.....(((.((((.((..((....)).)))))).)))
9.3	0.0011	.....(((...((((.((..((....)).)))))).)))
10.5	0.00034	.(((.....)))(((...((....))...)))....
11.6	0.00024	.((((.....))))(((...((....))...)))....
12.8	0.000057	.(((...)))((((.(((.(.((....)).).)))))).
14.0	0.000017	.(((...)))...((..((((.(.....).))))..)).
15.1	0.0000060	.....(((.(.(..((((.....))))..).).)))..
16.3	0.0000020	.((((((.(.....).)))))((.(((....))).)).
17.4	0.0000010	.(((((.(.....).).)))..((..(((....)))..))

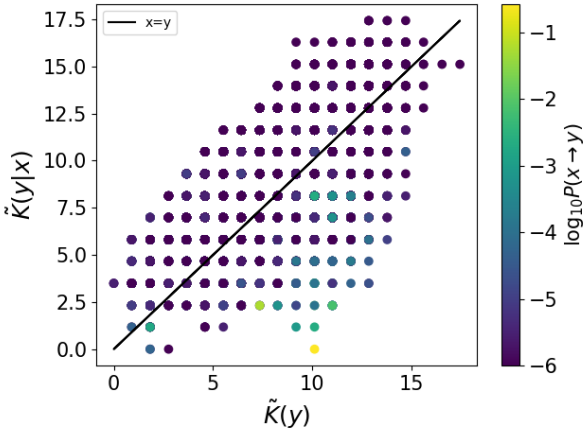
(a)



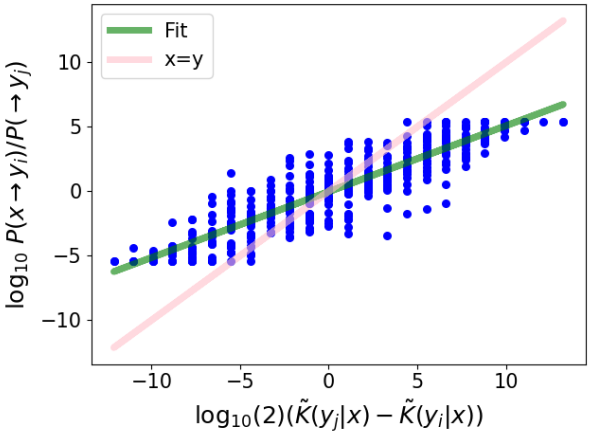
(b)



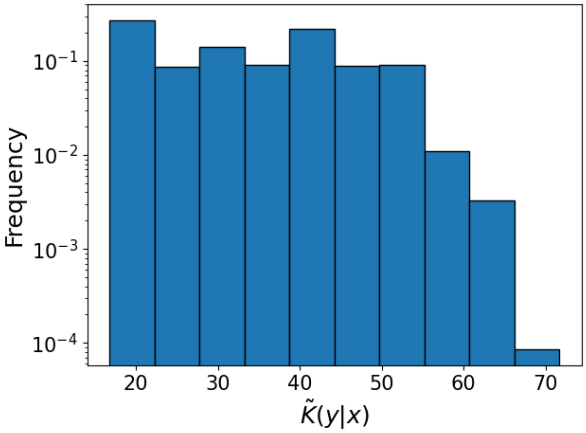
(c)



(d)



(e)



(f)

FIG. 1. **RNA secondary structure transition probabilities for a sequence of length $L = 40$ nucleotides**. The starting phenotype is $x =$.(((...))).((((.((..((....)).)))))).... and transitions result from choosing random genotypes in the NS of x, and introducing a single random mutation to each genotype. (a) Table illustrating the highest-probability SS for each conditional complexity value. The starting phenotype x is marked with an asterisk (*), and $P(x \to y)$ is just the robustness since $x = y$. (b) Transition probabilities $P(x \to y)$ decrease exponentially with increasing conditional complexity $\tilde{K}(y|x)$, upper bound of Eq. (10) depicted in black. The highest probability SS is the same phenotype as x. The predicted upper bound (red) and fitted bound (black) are close. (c) The unconditional complexity $\tilde{K}(y)$ does not predict the transition probabilities well. (d) $\tilde{K}(y|x) \approx \tilde{K}(y)$ for most y, leading to a positive linear correlation between values, cf. Eq. (13). (e) Ratios of probabilities correlate strongly with differences in conditional complexity, partially according with Eq. (18). (f) The histogram of conditional complexity values shows a roughly flat distribution (on a log scale), but with some slight bias towards simplicity.

role in modulating probabilities, then the null success rate of correctly guessing which of $P(x \to y_i)$ or $P(x \to y_j)$ is higher should be roughly 50% (if all probabilities were distinct it would be 50% exactly, but some of the probabilities are repeated).

H. Predicting $P(x \to y_i)/P(x \to y_j)$ ratios

Beyond predicting which has higher probability, we can also try to predict the value of the ratio of probabilities of transitioning from one phenotype to different alternative phenotypes. The ratio is also related to how confident we are in predicting which of the probabilities $P(x \to y_i)$ and $P(x \to y_j)$ is higher: a higher ratio means more confidence in the prediction.

Because y_i and y_j are both randomly generated, we expect both $P(x \to y_i)$ and $P(x \to y_j)$ to be close to the bound of Eq. (10), with high probability [30, 31]. Therefore we can use an approximate equality assumption to predict the ratio as follows:

$$\frac{P(x \to y_i)}{P(x \to y_j)} \approx \frac{2^{-a\tilde{K}(y_i|x)-b}}{2^{-a\tilde{K}(y_j|x)-b}} = 2^{-a\tilde{K}(y_i|x)}/2^{-a\tilde{K}(y_j|x)}$$
$$\Rightarrow \log_{10}\frac{P(x \to y_i)}{P(x \to y_j)} \approx a\log_{10}(2)(\tilde{K}(y_j|x) - \tilde{K}(y_i|x)) \qquad (18)$$

notice that b is irrelevant, so that even if b is not known, the prediction is unaffected. Recalling from Section (IV C) that we set $a = 1$ yields $\log_{10}(2)(\tilde{K}(y_j|x) - \tilde{K}(y_i|x))$ as the predictor for the $\log_{10}$ ratio of the probabilities. If the scaling is not done correctly, then $a \neq 1$ and therefore the predictor will not be as accurate, but instead off by a constant factor. In this case, we still have a relative measure of how confident we are about the prediction, where larger values of $\log_{10}(2)(\tilde{K}(y_j|x) - \tilde{K}(y_i|x))$ are associated with higher-confidence predictions.

I. Distribution of conditional complexities

Eq. (10) states that higher-probability phenotypes must have low conditional complexity, so in this sense we have derived a 'conditional simplicity bias'. A related but different question is, upon choosing a random genotype in the NS of x, and introducing a random mutation, what conditional complexity value is most likely? More generally, what kind of distribution should we expect for $K(y|x)$? This is not a trivial question, because on the one hand, the upper bound on transition probabilities decays exponentially with increasing $K(y|x)$, which would suggest that lower complexities are more likely. On the other hand, from AIT we expect the number of patterns with higher complexity to grow exponentially with increasing complexity, and hence increase with $K(y|x)$, which would suggest that higher complexities are more likely. However, how the actual number of conditional complexities grows for a specific GP map system may not reflect the AIT expectation exactly.

In ref. [23] (see also [48]) it was argued that to a first approximation, these two exponential trends should cancel each other out, leading to a 'flat' complexity distribution. Based on these arguments, we suggest that perhaps a flat distribution will also be seen for $K(y|x)$. Predicting the distribution of complexities is somewhat difficult, due to the fact that for a 'flat' distribution the two exponential trends must precisely cancel out, while exponential trends can easily magnify small errors in approximation. The distribution of complexities should be investigated in future work.

V. EMPIRICAL PHENOTYPE TRANSITION PROBABILITIES

A. Main predictions

Before analysing some example GP maps, we recap the main predictions and equations discussed in this work:
(a) High transition probabilities $P(x \to y)$ will be associated to either phenotypes which are similar to x, or are very simple.
(b) Transition probabilities will conform to the upper bound $P(x \to y) \lesssim 2^{-a\tilde{K}(y|x)-b}$ as in Eq. (10), with $a = 1$ and $b = 0$ as default expectations.
(c) Some phenotypes will have probabilities close to the upper bound, while many may be low-complexity, low-probability outputs far below the bound.
(d) For most phenotypes y, $K(y|x) \approx K(y)$ and especially if $K(x)$ is low, but $K(y|x) \ll K(y)$ for some y.
(e) We can predict which of two phenotypes is more likely to arise just by comparing their conditional complexities, betting on the simpler one having higher probability.
(f) We can predict the actual ratio of these two probabilities.

In the following sections, we computationally test the applicability of these predictions to RNA and protein SS.

B. Computational experiment: RNA secondary structure

RNA are important and versatile biomolecules that act as information carriers, enzymes, and structural components, among other roles. Similarly to DNA, RNA is comprised of a sequence of nucleotides, which can contain four possible nucleobases, A, U, C, and G. RNA molecules typically fold up into well-defined SS, which denote bonding patterns in the molecule. The SS shape is determined by the underlying sequence, but at the same time the sequence-to-SS map is highly redundant with many different sequences adopting the same SS. RNA SS has been very well studied in biophysics and computational biology because it is a biologically relevant system, but at the same time fast and accurate computational algorithms exist for predicting SS from sequences. Here we use the popular Vienna RNA [68] package for predicting SS, which utilizes a thermodynamics-based algorithm.

To test our predictions, we randomly generated an RNA sequence of length $L = 40$ nucleotides, and computationally predicted its SS `.(((...)))..((((.((..((....)).)))))).....`, which we will denote by x. We chose $L = 40$ because computational efficiency requires a comparatively short sequence, but on the other hand complexity-probability connections are more pronounced for longer RNA [30]. In order to estimate

$P(x \to y)$, we need to generate a large sample of sequences within the neutral space of x, that is, different sequences that each have x as their SS. To generate the sequences, we used the site-scanning method of [69]. Afterwards, we introduced a single random point mutation for each neutral sequence, and recorded the resulting SS. More details are given in the Methods.

Figure 1(a) shows the highest-probability SS for each conditional complexity value found in the dataset generated in this way, as well as the estimated probability of transitioning from the starting phenotype x to all of these alternative phenotypes. In Figure 1(b) we see that, as expected, there is strong bias in transition probabilities with a decay in the upper bound to $P(x \to y)$. The black line is a fit to the data ($a = 1.1$ and $b = 0$) added to highlight the upper bound. Figure 1(b) also shows the predicted upper bound (red line) based on $a = 1$ and $b = 0$; this prediction is impressive, given that it is based on just the output complexities themselves. Note that there are several low probability structures for which our measure $\tilde{K}(x|y)$ gives zero, even though they are slightly different than the reference structure. This effect is most likely caused by two effects. Firstly, the lack of very fine resolution in our approximate complexity measure, and second, by our neglecting the O(1) terms. Panel (c) presents the same data as Figure 1(b), except that the horizontal axis is $\tilde{K}(y)$ instead of $\tilde{K}(y|x)$, and it is apparent that $\tilde{K}(y)$ does not provide a good predictor of the probabilities. This demonstrates that the conditional complexities are needed, not just the complexity $\tilde{K}(y)$. Figure 1(d) is a scatter plot of $\tilde{K}(y)$ vs $\tilde{K}(y|x)$ and, as expected, there is a linear (but noisy) correlation between these two quantities (Pearson $r = 0.64$, p-value$< 10^{-6}$). It is also interesting to see that, as expected, a small number of phenotypes have conditional complexities $\tilde{K}(y|x)$ much lower than the unconditional complexity $\tilde{K}(y)$, and those tend to be the higher-probability outputs (as can be seen from the colouring of the datapoints by log probability).

Turning to the prediction of which of two phenotypes has higher probability, we find that with probability-weighted sampling the accuracy is a striking 86%, and with uniform sampling, the rate is still impressive at 79%. (Recall that probability-weighted sampling refers to when genotypes are uniformly randomly sampled, and hence phenotypes appear with frequencies according to their probabilities. Uniform sampling refers to when each phenotype is sampled with equal probability.) Extending this, Figure 1(e) depicts not just predictions of which is higher, but of the ratios themselves. Although the fit does not match the $x = y$ prediction line, there is nonetheless a strong correlation (Pearson $r = 0.91$, p-value$< 10^{-6}$) between the ratios of the probabilities and the differences in complexities. This means that although the slope is not very well predicted, the expected close connection between the complexity values and the probabilities holds. Recall that the inaccuracy in the slope predictions for Figures 1(b) and (e) results primarily from a lack of precision in estimating the value of a, which depends on knowing the number of possible phenotypes y such that $P(x \to y) > 0$. The fact that the actual slope is flatter than the predicted one is presumably due to the following: The value of $\log_{10}(2)(\tilde{K}(y_j|x) - \tilde{K}(y_i|x))$ will be large when y_j is (conditionally) complex and y_i is simple. Therefore y_j is unlikely to be far from the upper bound, while very simple phenotypes can be very far from the upper bound, cf. the low-complexity, low-probability phenomenon [31, 61]. Hence the value of $P(x \to y_i)/P(x \to y_j)$ is likely to be an underestimate, rather than an overestimate, which will tend to make the slope flatter.

Finally, Figure 1(f) presents a histogram of conditional complexities, showing that their distribution is roughly flat on a log scale as tentatively predicted, but it also exhibits some mild bias towards lower-complexity phenotypes.

In Appendix B we numerically study the impact of the complexity of the starting phenotype x, showing that the mean value in the difference between $\tilde{K}(y|x)$ and $\tilde{K}(y)$ is small for simple x as expected, and grows for more complex x. In the Appendix, Figure 4 shows an additional example RNA plot, depicting results similar to those in Figure 1.

C. Computational experiment: protein secondary structure

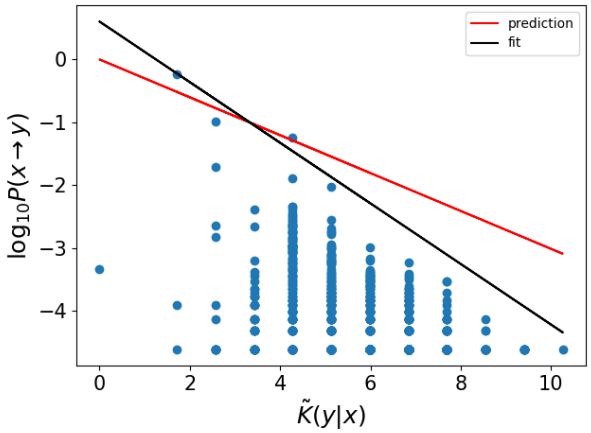
Proteins are biomolecules that form the fundamental building blocks of organisms. A protein is comprised of one or more macromolecular chains that in living organisms typically contain 20 types of amino acid residues. Similarly to RNA, a protein will fold into a particular spatial structure, which is determined by the specific amino acid sequence. There is redundancy in that many different sequences can have the same fold, see e.g. [71]. The overall 3D arrangement of a protein's polypeptide chain in space is known as its tertiary structure, while protein SS refers to the local conformation of the polypeptide backbone. SS is key to protein folding [72] and the genotype-to-phenotype map between primary and secondary structure has not received much attention in the literature to date. At the level of detail we are concerned with, determining a protein's SS is equivalent to specifying whether each amino acid residue in the chain is involved in a coil (C), sheet (E), or helix (H) structure. Hence, the SS of a protein of length L is also a sequence of length L, but with only three possible letters (C, E, H) at each site.

Predicting the full 3D structure of a protein was until recently an open problem, but it is now feasible via machine learning algorithms such as AlphaFold [73]. However, it remains very computationally taxing and potentially unreliable for sequences not related to the natural sequences used to train the underlying machine learning algorithm. In contrast, the machine learning-based Porter 5 algorithm provides accurate and relatively fast predictions of protein SS [74]. Predicting the structure of mutants remains challenging both for algorithms such as AlphaFold [75] and for SS ones like Porter 5. Here we make the implicit assumption that Porter 5 captures the large-scale properties of the mapping between protein primary and secondary structure sufficiently accurately that our computational study grants insight into the system. Such insight is particularly valuable considering that it is impractical to survey a similar number of mutants experimentally.

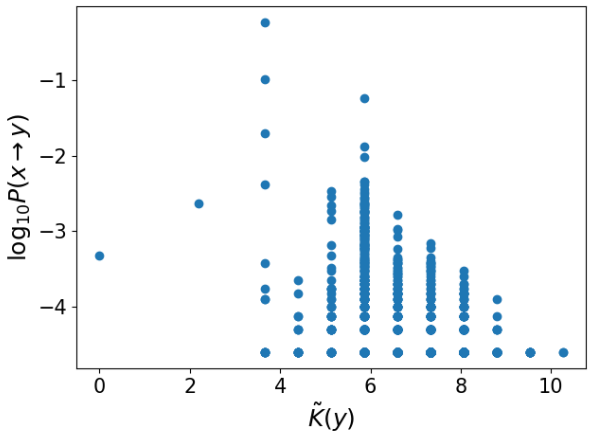
Like with the RNA example, choosing the length L of the protein under study is a balance between computational cost and accuracy: If longer proteins are used, then the number of possible SS grows exponentially, making it hard

$\tilde{K}(y\|x)$	$P(x \to y)$	y
0.0	0.00047	CCCCCCCCCCCCCCCCCCCC
1.7	0.58	EEECCCCCCCCCCCCCCCCC*
2.6	0.10	EEEECCCCCCCCCCCCCCCC
3.4	0.0042	EEEEECCCCCCCCCCCCCCC
4.3	0.058	EEECCCCCCCCCCCCCCCCE
5.1	0.0096	EEEECCCCCCCCCCCCCCCE
6.0	0.0010	EEEECCCCCEECCCCCCCCC
6.8	0.00059	EEECCCCCCCCHHCCCCCCE
7.7	0.00030	EEECCCCCCCHHHCCCCCCE
8.5	0.000074	EEECCCCCCEEHHHCCCCEE
9.4	0.000025	EEECCCCCCEEHECCCHCCC
10.3	0.000025	EEECCCCCHHCCHHEHCCCC

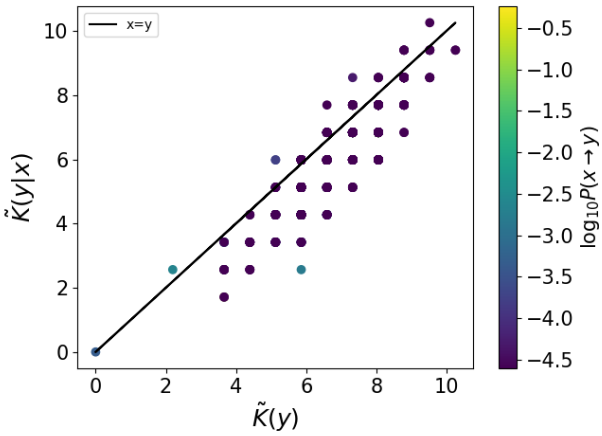
(a)



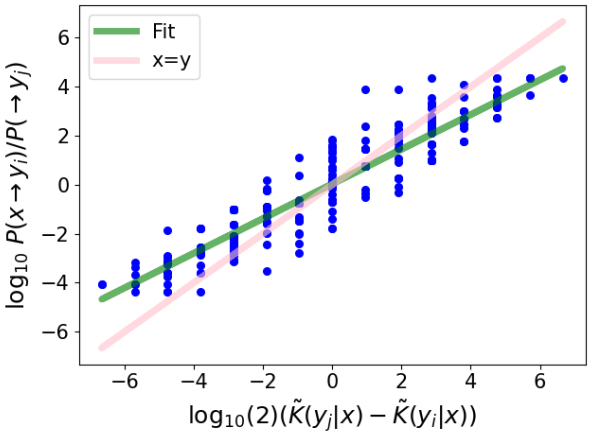
(b)



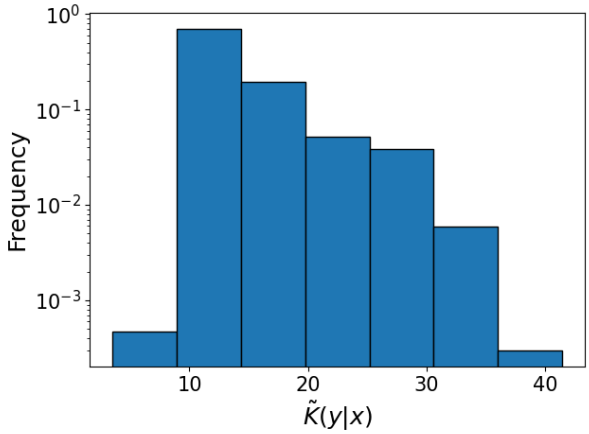
(c)



(d)



(e)



(f)

FIG. 2. **Protein secondary structure transition probabilities for CDRH3 from the heavy chain of a human monoclonal SARS-CoV antibody (PDB ID: 6WS6 [70]), length** $L = 20$**,** $x =$EEECCCCCCCCCCCCCCCCC. (a) Table illustrating the highest probability SS for each conditional complexity value. The starting phenotype x is marked with an asterisk (*), and $P(x \to y)$ is just the robustness since $x = y$. (b) Transition probabilities $P(x \to y)$ decrease exponentially with increasing conditional complexity $\tilde{K}(y|x)$. The phenotype with the lowest conditional complexity is CCCCCCCCCCCCCCCCCCCC. The phenotype with the highest probability is $y = x$. The predicted upper bound (red) is close to the fitted bound (black, cf. Eq. (10)). (c) The unconditional complexity $\tilde{K}(y)$ predicts the transition probabilities nearly as well as $\tilde{K}(y|x)$, presumably because x has low complexity (see Appendix B). (d) $\tilde{K}(y|x) \approx \tilde{K}(y)$ for most y, leading to a positive linear correlation between values. (e) Ratios of probabilities correlate strongly with differences in conditional complexity. (f) The histogram of conditional complexity values shows a bias towards simpler proteins with lower complexity values.

to get accurate estimates of probabilities as the latter require the same SS to appear multiple times. On the other hand, very short proteins are less biologically relevant, and the complexity measures and theory are expected to work worse for them than for longer sequences. Here we balance these considerations by studying a complementarity-determining region (CDR) with $L = 20$, specifically CDRH3 from the heavy chain of a human monoclonal SARS-CoV antibody (Protein Data Bank [PDB] [76] ID: 6WS6 [70]). Antibody complementarity-determining (hypervariable) regions are critical in the recognition of antigens and extreme sequence variation in them allows antibodies to bind to a nearly limitless range of antigens. The particular antibody under study potently neutralises the SARS-CoV-2 virus [77]. The CDRH3 region is especially important for antigen recognition [78, 79], and its conformation is not restricted to a small set of canonical structures, unlike those of the other CDRs [78–80]. Note also that because Porter 5 is a machine learning-based algorithm, it is only expected to be accurate for sequences similar to natural sequences on which it was trained. Hence the need to use a naturally occurring protein, rather than a completely random sequence, for which Porter 5 may yield inaccurate SS predictions. The SS predicted by Porter 5 for the chosen CDRH3 is $x =$`EEECCCCCCCCCCCCCCCCCC`, and this SS was highly accurate (85%) when compared to experimentally derived SS, `EEECCCCCCCCCCCCCCCEEE`. We used site scanning [69] to generate a large number of different sequences within the neutral space of x. We then generated all mutant sequences reachable via a point mutation in DNA for a subset of the sample of neutral genotypes, and used Porter 5 to predict their SS (Methods).

Figure 2(a) shows the most frequent SS for each conditional complexity value found in the dataset generated via site scanning, as well as the estimated probability of transitioning from the starting phenotype x to all of these alternative ones. In Figure 2(b) we see a strong bias in transition probabilities with a linear decay in the upper bound to $P(x \to y)$, as predicted. The black line is a fit to the data ($a = 1.6$ and $b = -2$), added to highlight the upper bound. The panel also shows the predicted upper bound based on $a = 1$ and $b = 0$ (red line); this prediction is useful, but not as accurate as for RNA $L = 40$. Panel (c) presents the same data as Figure 2(a), except that $\tilde{K}(y)$ is on the horizontal axis instead of $\tilde{K}(y|x)$. Interestingly, $\tilde{K}(y)$ is quite similar to $\tilde{K}(y|x)$, which is probably due to the fact that the original starting phenotype x is very simple (see Appendix B for more on this). The scatter plot of $\tilde{K}(y)$ vs $\tilde{K}(y|x)$ in Figure 2(d) shows that, as expected, there is a linear correlation between these two quantities (Pearson $r = 0.87$, p-value$< 10^{-6}$). It is interesting that the correlation is stronger than for the RNA example, and this again can be rationalised by the fact that x is very simple. There do not appear to be any phenotypes for which $\tilde{K}(y|x)$ is much smaller than $\tilde{K}(y)$. With probability-weighted sampling, the accuracy of predicting which of two phenotypes has higher probability is very high at 79%, and with uniform sampling is 77%. Figure 2(e) shows predictions for the ratio of the probabilities for transitioning to different phenotypes versus the difference between their estimated complexity in log scale. The latter displays a strong correlation (Pearson $r = 0.93$, p-value$< 10^{-6}$). Finally, Figure 2(f) presents a histogram of conditional complexities, showing a stronger bias towards lower-complexity phenotypes than the data for RNA. In the Appendix, Figure 5 shows highly accurate predictions for another protein SS example.

VI. DISCUSSION

We have studied the problem of predicting the probability of transition between phenotypes x and y, $P(x \to y)$, from the perspective of algorithmic information theory (AIT), specifically algorithmic probability. We derived an upper bound on $P(x \to y)$ which depends on the conditional complexity of phenotype y given x. The derivations were motivated by the observation that the assignment of genotypes to phenotypes is highly structured, and the expectation that the constraints of information theory should therefore have predictive value in genotype-to-phenotype maps. Upon testing our various predictions on RNA and protein secondary structure examples, we found good quantitative agreement between the theory and the simulations.

The benefit of developing this theoretical approach is that it allows predictions to be made about transition probabilities for GP maps in which only phenotypes are observed and little or no knowledge about the map is available. This approach is also relevant to uncovering general common properties of GP maps, which are important for the advancement of the currently underdeveloped field of theoretical biology [81].

In this study we restricted our attention to one aspect of the structure of the GP, without including evolutionary dynamic effects resulting from mutation rates, strength of selection, population size or others that would be relevant in biology. Therefore we leave these to future work. It is interesting, however, that studies of natural RNA shapes [21, 22] and the shapes of protein complexes [23] have shown that GP map biases alone can be very good predictors of natural biological shape frequencies (see also refs. [41, 82–84] for more on different types of biases and evolutionary outcomes). Therefore, it may be that the transition probability biases discussed here, resulting from conditional complexity constraints, are strong enough that their stamp is still observable even in natural data. We suggest that an interesting follow-on study to ours would be to test this with natural bioinformatic databases.

It is also noteworthy that other researchers have empirically observed a tendency for genetic mutations to favour simpler morphologies, specifically in teeth [85], embryo [86], and leaf shapes [87]. We believe that our results help rationalise these observations within a general theoretical framework.

Robustness to genetic mutations is an important property for organisms [8], but a general explanation for the high levels of robustness observed in GP maps has been lacking [50]. Our information theory perspective here relates to this question because we have seen that transitions to phenotypes which are similar to the starting phenotype tend to have high probability, and of course a phenotype is most similar to itself. We explore this in detail in a forthcoming study.

Other authors have used information theory for non-UTMs to derive some results which are related to the ones we derived here. For example Calude et al [88] developed bounds

for finite-state machines (the simplest computing devices), and moreover Merhav et al [89] have derived similar bounds to ours directly in terms of Lempel-Ziv complexity. It would be interesting to see if these calculations for non-UTMs could be extended to GP maps.

While the upper bound appears to work well in the simulations presented here, a main weakness in our predictions is that many phenotypes that have low conditional complexities $\tilde{K}(y|x)$ also have low probabilities. Because these phenotypes fall far below the upper bound, their precise probabilities are not well predicted by the theory. These low-complexity, low-probability patterns have been described as having low absolute information content, but due to map-specific biases they are 'hard' for the map to make, and hence have low probability [31]. The origins and nature of these types of patterns have been recently studied [61], but a full understanding of them and knowledge regarding how to improve probability predictions of these have not yet been achieved. Despite the challenge of low-complexity, low-probability patterns, we were still able to make high-accuracy upper bounds on phenotype probabilities as well as high-accuracy ($\sim 80\%$) predictions about which of two phenotypes is more likely, just using complexity values.

The quantitatively accurate predictions we describe here motivate further investigation of the use of AIT-inspired predictions in biology, evolution, and other natural sciences.

Acknowledgments: KD acknowledges financial support from the Gulf University for Science and Technology award of Seed Grant (grant number 263301) and a Summer Faculty Fellowship. This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (`www.csd3.cam.ac.uk`), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council (`www.dirac.ac.uk`).

Data availability: The data sets generated during and analysed during the current study are available from the corresponding author(s) on request.

Author contributions: Conceived the study: KD, JN, SA, AL. Analytic calculations: KD. Simulations and data analysis: KD, JN. Wrote the paper: KD, JN, SA, AL.

[1] Andreas Wagner. *Arrival of the fittest: solving evolution's greatest puzzle*. Penguin, 2014.

[2] Sebastian E Ahnert. Structural properties of genotype–phenotype maps. *Journal of The Royal Society Interface*, 14(132):20170275, 2017.

[3] Susanna Manrubia, José A Cuesta, Jacobo Aguirre, Sebastian E Ahnert, Lee Altenberg, Alejandro V Cano, Pablo Catalán, Ramon Diaz-Uriarte, Santiago F Elena, Juan Antonio García-Martín, et al. From genotypes to organisms: State-of-the-art and perspectives of a cornerstone in evolutionary dynamics. *Physics of Life Reviews*, 38:55–106, 2021.

[4] P. Schuster, W. Fontana, P.F. Stadler, and I.L. Hofacker. From sequences to shapes and back: A case study in RNA secondary structures. *Proceedings: Biological Sciences*, 255(1344):279–284, 1994.

[5] E. Borenstein and D.C. Krakauer. An end to endless forms: epistasis, phenotype distribution bias, and nonuniform evolution. *PLoS Comput Biol*, 4(10):e1000202, 2008.

[6] I.G. Johnston, S.A. Ahnert, J.P.K. Doye, and A.A. Louis. Evolutionary dynamics in a simple model of self-assembly. *Physical Review E*, 83:066105, 2011.

[7] Kamaludin Dingle. *Probabilistic bias in genotype-phenotype maps*. PhD thesis, University of Oxford, 2014.

[8] A. Wagner. *Robustness and evolvability in living systems*. Princeton University Press Princeton, NJ:, 2005.

[9] T. Jorg, O.C. Martin, and A. Wagner. Neutral network sizes of biological RNA molecules can be computed and are not atypically small. *BMC bioinformatics*, 9(1):464, 2008.

[10] J.A. Draghi, T.L. Parsons, G.P. Wagner, and J.B. Plotkin. Mutational robustness can facilitate adaptation. *Nature*, 463(7279):353–355, 2010.

[11] Jacobo Aguirre, Javier M Buldú, Michael Stich, and Susanna C Manrubia. Topological structure of the space of phenotypes: the case of rna neutral networks. *PloS one*, 6(10):e26324, 2011.

[12] M.C. Cowperthwaite, E.P. Economo, W.R. Harcombe, E.L. Miller, and L.A. Meyers. The ascent of the abundant: how mutational networks constrain evolution. *PLoS computational biology*, 4(7):e1000110, 2008.

[13] Steffen Schaper and Ard A Louis. The arrival of the frequent: how bias in genotype-phenotype maps can steer populations to local optima. *PloS one*, 9(2):e86635, 2014.

[14] Sam F Greenbury, Iain G Johnston, Ard A Louis, and Sebastian E Ahnert. A tractable genotype–phenotype map modelling the self-assembly of protein quaternary structure. *Journal of The Royal Society Interface*, 11(95):20140249, 2014.

[15] Pablo Catalán, Susanna Manrubia, and José A Cuesta. Populations of genetic circuits are unable to find the fittest solution in a multilevel genotype–phenotype map. *Journal of the Royal Society Interface*, 17(167):20190843, 2020.

[16] S. Psujek and R.D. Beer. Developmental bias in evolution: evolutionary accessibility of phenotypes in a model evo-devo system. *Evolution & Development*, 10(3):375–390, 2008.

[17] Walter Fontana, Danielle AM Konings, Peter F Stadler, and Peter Schuster. Statistics of rna secondary structures. *Biopolymers: Original Research on Biomolecules*, 33(9):1389–1404, 1993.

[18] Erik A Schultes, Alexander Spasic, Udayan Mohanty, and David P Bartel. Compact and ordered collapse of randomly generated rna sequences. *Nature structural & molecular biology*, 12(12):1130–1136, 2005.

[19] Sandra Smit, Michael Yarus, and Rob Knight. Natural selection is not required to explain universal compositional patterns in rrna secondary structure categories. *RNA*, 12(1):1–14, 2006.

[20] Kamaludin Dingle, Steffen Schaper, and Ard A Louis. The structure of the genotype–phenotype map strongly constrains the evolution of non-coding RNA. *Interface focus*, 5(6):20150053, 2015.

[21] Kamaludin Dingle, Fatme Ghaddar, Petr Šulc, and Ard A

Louis. Phenotype bias determines how natural rna structures occupy the morphospace of all possible shapes. *Molecular biology and evolution*, 39(1):msab280, 2022.

[22] Fatme Ghaddar and Kamaludin Dingle. Random and natural non-coding rna have similar structural motif patterns but can be distinguished by bulge, loop, and bond counts. *bioRxiv*, 2022.

[23] Iain G Johnston, Kamaludin Dingle, Sam F Greenbury, Chico Q Camargo, Jonathan PK Doye, Sebastian E Ahnert, and Ard A Louis. Symmetry and simplicity spontaneously emerge from the algorithmic nature of evolution. *Proceedings of the National Academy of Sciences*, 119(11):e2113883119, 2022.

[24] Jasper Franke, Alexander Klözer, J Arjan GM de Visser, and Joachim Krug. Evolutionary accessibility of mutational pathways. *PLoS Comput Biol*, 7(8):e1002134, 2011.

[25] L. Tan, S. Serene, H.X. Chao, and J. Gore. Hidden randomness between fitness landscapes limits reverse evolution. *Physical Review Letters*, 106(19):198102, 2011.

[26] Isaac Salazar-Ciudad and Miquel Marín-Riera. Adaptive dynamics under development-based genotype–phenotype maps. *Nature*, 497(7449):361–364, 2013.

[27] R. J. Solomonoff. A preliminary report on a general theory of inductive inference (revision of report v-131). *Contract AF*, 49(639):376, 1960.

[28] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of information transmission*, 1(1):1–7, 1965.

[29] Gregory J Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM (JACM)*, 22(3):329–340, 1975.

[30] Kamaludin Dingle, Chico Q Camargo, and Ard A Louis. Input–output maps are strongly biased towards simple outputs. *Nature communications*, 9(1):761, 2018.

[31] Kamaludin Dingle, Guillermo Valle Pérez, and Ard A Louis. Generic predictions of output probability based on complexities of inputs and outputs. *Scientific reports*, 10(1):1–9, 2020.

[32] SE Ahnert, IG Johnston, TMA Fink, JPK Doye, and AA Louis. Self-assembly, modularity, and physical complexity. *Physical Review E*, 82(2):026117, 2010.

[33] C.H. Bennett. The thermodynamics of computation – a review. *International Journal of Theoretical Physics*, 21(12):905–940, 1982.

[34] Artemy Kolchinsky and David H Wolpert. Thermodynamic costs of turing machines. *Physical Review Research*, 2(3):033312, 2020.

[35] W.H. Zurek. Algorithmic randomness and physical entropy. *Physical Review A*, 40(8):4731, 1989.

[36] Markus P Mueller. Law without law: from observer states to physics via algorithmic information theory. *Quantum*, 4:301, 2020.

[37] Ram Avinery, Micha Kornreich, and Roy Beck. Universal and accessible entropy estimation using a compression algorithm. *Physical review letters*, 123(17):178102, 2019.

[38] Stefano Martiniani, Paul M Chaikin, and Dov Levine. Quantifying hidden order out of equilibrium. *Physical Review X*, 9(1):011031, 2019.

[39] P. Ferragina, R. Giancarlo, V. Greco, G. Manzini, and G. Valiente. Compression-based classification of biological sequences and structures via the universal similarity metric: experimental assessment. *BMC bioinformatics*, 8(1):252, 2007.

[40] Alyssa Adams, Hector Zenil, Paul CW Davies, and Sara Imari Walker. Formal definitions of unbounded evolution and innovation reveal universal mechanisms for open-ended evolution in dynamical systems. *Scientific reports*, 7(1):1–15, 2017.

[41] Kamaludin Dingle. Fitness, optima, and simplicity. *Preprints*, 2022080402, 2022.

[42] Hector Zenil, Fernando Soler-Toscano, Kamaludin Dingle, and Ard A Louis. Correlation of automorphism group size and topological properties with program-size complexity evaluations of graphs and complex networks. *Physica A: Statistical Mechanics and its Applications*, 404:341–358, 2014.

[43] Hector Zenil, Narsis A Kiani, and Jesper Tegnér. A review of graph and network complexity from an algorithmic information perspective. *Entropy*, 20(8):551, 2018.

[44] Paul MB Vitányi. Similarity and denoising. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2013.

[45] R. Cilibrasi and P.M.B. Vitányi. Clustering by compression. *Information Theory, IEEE Transactions on*, 51(4):1523–1545, 2005.

[46] Hector Zenil, Narsis A Kiani, Allan A Zea, and Jesper Tegnér. Causal deconvolution by algorithmic generative models. *Nature Machine Intelligence*, 1(1):58–66, 2019.

[47] H. Zenil and J.P. Delahaye. An algorithmic information theoretic approach to the behaviour of financial markets. *Journal of Economic Surveys*, 25(3):431–463, 2011.

[48] Kamaludin Dingle, Rafiq Kamal, and Boumediene Hamzi. A note on a priori forecasting and simplicity bias in time series. *arXiv preprint arXiv:2203.05391*, 2022.

[49] M. Li and P.M.B. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag New York Inc, 2008.

[50] Sam F Greenbury, Steffen Schaper, Sebastian E Ahnert, and Ard A Louis. Genetic correlations greatly increase mutational robustness and can both reduce and enhance evolvability. *PLoS computational biology*, 12(3):e1004773, 2016.

[51] Marcel Weiß and Sebastian E Ahnert. Phenotypes can be robust and evolvable if mutations have non-local effects on sequence constraints. *Journal of The Royal Society Interface*, 15(138):20170618, 2018.

[52] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.

[53] Peter Grunwald and Paul Vitányi. Shannon information and Kolmogorov complexity. *arXiv preprint cs/0410002*, 2004.

[54] C.S. Calude. *Information and randomness: An algorithmic perspective*. Springer, 2002.

[55] P. Gács. *Lecture notes on descriptional complexity and randomness*. Boston University, Graduate School of Arts and Sciences, Computer Science Department, 1988.

[56] Alexander Shen, Vladimir A Uspensky, and Nikolay Vereshchagin. *Kolmogorov complexity and algorithmic randomness*, volume 220. American Mathematical Society, 2022.

[57] Sean D Devine. *Algorithmic Information Theory for Physicists and Natural Scientists*. IOP Publishing, 2020.

[58] L.A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.

[59] Ray J Solomonoff. The discovery of algorithmic probability. *Journal of Computer and System Sciences*, 55(1):73–88, 1997.

[60] Paul MB Vitányi. Conditional kolmogorov complexity and universal probability. *Theoretical Computer Science*, 501:93–100, 2013.

[61] Mohamed Alaskandarani and Kamaludin Dingle. Low complexity, low probability patterns and consequences for algorithmic probability applications. *arXiv preprint arXiv:2207.12251*, 2022.

[62] A. Lempel and J. Ziv. On the complexity of finite sequences. *Information Theory, IEEE Transactions on*, 22(1):75–81,

1976.

[63] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.

[64] J.P. Delahaye and H. Zenil. Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of algorithmic randomness. *Appl. Math. Comput.*, 219:63–77, 2012.

[65] Fernando Soler-Toscano, Hector Zenil, Jean-Paul Delahaye, and Nicolas Gauvrit. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PloS one*, 9(5):e96223, 2014.

[66] TM Cover and J.A. Thomas. *Elements of information theory*. John Wiley and Sons, 2006.

[67] Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul MB Vitányi. The similarity metric. *IEEE transactions on Information Theory*, 50(12):3250–3264, 2004.

[68] Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. Viennarna package 2.0. *Algorithms for molecular biology*, 6(1):26, 2011.

[69] Marcel Weiß and Sebastian E Ahnert. Using small samples to estimate neutral component size and robustness in the genotype–phenotype map of rna secondary structure. *Journal of the Royal Society Interface*, 17(166):20190784, 2020.

[70] D. Pinto, D., Park, Y.J., Beltramello, M., Walls, A.C., Tortorici, M.A., Bianchi, S., Jaconi, S., Culap, K., Zatta, F., Marco, A.D., Peter, A., Guarino, B., Spreafico, R., Cameroni, E., Case, J.B., Chen, R.E., Havenar-Daughton, C., Snell, G., Telenti, A., Vi. Structural and functional analysis of a potent sarbecovirus neutralizing antibody, DOI: 10.2210/pdb6WS6/pdb, 2020.

[71] Cyrus Chothia. One thousand families for the molecular biologist. *Nature*, 357(6379):543–544, 1992.

[72] Yong Yun Ji and You Quan Li. The role of secondary structure in protein structure selection. *Eur. Phys. J. E*, 32(1):103–107, 2010.

[73] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[74] Mirko Torrisi, Manaz Kaleel, and Gianluca Pollastri. Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction. *Scientific reports*, 9(1):1–12, 2019.

[75] Gwen R. Buel and Kylie J. Walters. Can AlphaFold2 predict the impact of missense mutations on structure? *Nat. Struct. Mol. Biol.*, 29(1):1–2, jan 2022.

[76] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[77] Dora Pinto, Young Jun Park, Martina Beltramello, Alexandra C. Walls, M. Alejandra Tortorici, Siro Bianchi, Stefano Jaconi, Katja Culap, Fabrizia Zatta, Anna De Marco, Alessia Peter, Barbara Guarino, Roberto Spreafico, Elisabetta Cameroni, James Brett Case, Rita E. Chen, Colin Havenar-Daughton, Gyorgy Snell, Amalio Telenti, Herbert W. Virgin, Antonio Lanzavecchia, Michael S. Diamond, Katja Fink, David Veesler, and Davide Corti. Cross-neutralization of SARS-CoV-2 by a human monoclonal SARS-CoV antibody. *Nature*, 583(7815):290–295, 2020.

[78] Ian A. Wilson and Robyn L Stanfield. Antibody-antigen interactions: new structures and new conformational changes. *Curr. Opin. Struct. Biol.*, 4(6):857–867, jan 1994.

[79] Brian D. Weitzner, Roland L. Dunbrack, and Jeffrey J. Gray. The Origin of CDR H3 Structural Diversity. *Structure*, 23(2):302–311, feb 2015.

[80] R. J. Pantazes and C. D. Maranas. OptCDR: A general computational method for the design of antibody complementarity determining regions for targeted epitope binding. *Protein Eng. Des. Sel.*, 23(11):849–858, 2010.

[81] David C Krakauer, James P Collins, Douglas Erwin, Jessica C Flack, Walter Fontana, Manfred D Laubichler, Sonja J Prohaska, Geoffrey B West, and Peter F Stadler. The challenges and scope of theoretical biology. *Journal of theoretical biology*, 276(1):269–276, 2011.

[82] L.Y. Yampolsky and A. Stoltzfus. Bias in the introduction of variation as an orienting factor in evolution. *Evolution & Development*, 3(2):73–83, 2001.

[83] Alejandro V Cano, Hana Rozhoňová, Arlin Stoltzfus, David M McCandlish, and Joshua L Payne. Mutation bias shapes the spectrum of adaptive substitutions. *Proceedings of the National Academy of Sciences*, 119(7):e2119720119, 2022.

[84] Isaac Salazar-Ciudad. Why call it developmental bias when it is just development? *Biology Direct*, 16:1–13, 2021.

[85] Enni Harjunmaa, Aki Kallonen, Maria Voutilainen, Keijo Hämäläinen, Marja L. Mikkola, and Jukka Jernvall. On the difficulty of increasing dental complexity. *Nature*, 483(7389):324–327, 2012.

[86] Pascal F Hagolani, Roland Zimm, Renske Vroomans, and Isaac Salazar-Ciudad. On the evolution and development of morphological complexity: A view from gene regulatory networks. *PLoS computational biology*, 17(2):e1008570, 2021.

[87] R. Geeta, L.M. Davalos, A. Levy, L. Bohs, M. Lavin, K. Mummenhoff, N. Sinha, and M.F. Wojciechowski. Keeping it simple: flowering plants tend to retain, and revert to, simple leaves. *New Phytologist*, 2011.

[88] Cristian S Calude, Kai Salomaa, and Tania K Roblot. Finite state complexity. *Theoretical Computer Science*, 412(41):5668–5677, 2011.

[89] Neri Merhav and Asaf Cohen. Universal randomized guessing with application to asynchronous decentralized brute–force attacks. *IEEE Transactions on Information Theory*, 66(1):114–129, 2019.

[90] P.D. Pancera, M., McLellan, J.S., Kwong. Crystal Structure of PG16 Fab in Complex with V1V2 Region from HIV-1 strain ZM109, DOI: 10.2210/pdb4DQO/pdb, 2013.

[91] Marie Pancera, Syed Shahzad-ul hussan, Nicole A Doria-rose, Jason S Mclellan, Robert T Bailer, Kaifan Dai, Sandra Loesgen, Mark K Louder, Ryan P Staupe, Yongping Yang, Baoshan Zhang, Robert Parks, Joshua Eudailey, Krissey E Lloyd, Julie Blinn, S Munir Alam, Barton F Haynes, Mohammed N Amin, Lai-xi Wang, Dennis R Burton, Wayne C Koff, Gary J Nabel, John R Mascola, Carole A Bewley, and Peter D Kwong. Structural basis for diverse N-glycan recognition by HIV-1 – neutralizing V1 – V2 – directed antibody PG16. *Nat. Publ. Gr.*, 20(7), 2013.

[92] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. SAbDab : the structural antibody database. *Nucleic Acids Res.*, 42(November 2013):1140–1146, 2014.

Appendix A: Methods

1. RNA

We used a (pseudo) random number generator to generate nucleotide sequences of length $L = 40$. We predicted the RNA secondary structures (SS) with the RNA Vienna folding package [68], which uses a thermodynamics-based algorithm, downloaded from `https://anaconda.org/bioconda/viennarna`. In order to estimate transition probabilities $P(x \to y)$, a sample of sequences from the neutral space (NS) of the phenotype x are required. Generating this sample by storing random sequences and saving only those which fold to phenotype x would be a possibility in principle, but it is highly computationally taxing. Hence we used site scanning [69], a method for sampling from the NS of a given phenotype. The site-scanning method introduces a random point mutation in the first site of the initial genotype; if this mutation is not neutral, it is reverted and a different random mutation is introduced at the same site. This process is repeated until a neutral mutant is identified or all possible point mutations at the first site have been exhausted, whereupon the algorithm moves on to the next site; when the algorithm reaches the last site, it restarts from the first one, and the process continues until the desired number of neutral mutants has been derived.

For the RNA figure in the main text, we used 10^7 'steps' in the site-scanning algorithm, and saved only a randomly chosen fraction of 0.1 of the sequences - a subsample in the terminology of [69] - in order to ensure low correlations between sequences in the sample. For the RNA figure in the appendix, we used 10^6 'steps'. For each of the saved sequences in the sample, we introduced a single random point mutation, and used the resulting mutated sequences to predict the resulting RNA SS. The frequency of each SS was used to define $P(x \to y)$.

2. Proteins

We predicted protein SS with the fast version of the machine learning-based tool Porter 5, which makes use of protein alignments performed via HHblits [74]. For each studied protein, we used the site-scanning technique as described in the preceding section with minor modifications as follows. 1) At each site in a given protein sequence, we introduced point mutations that are accessible via a point mutation in the DNA sequence following the standard genetic code. In doing so, we considered all such mutations equally likely without weighting for the number of codons that code for a particular amino acid or for codon bias; moreover, we disregarded mutations that change a codon encoding an amino acid to a stop codon. 2) Instead of modifying the entire protein, we only looked at mutations in the H3 hypervariable region of the heavy chains of two antibody proteins, and if the algorithm reached a dead end (i.e., a sequence with no neutral point mutants), we restarted it from a random genotype that had been derived earlier in the run. 3) When generating neutral genotypes during the site-scanning walk, we discarded genotypes that had previously been encountered in the run, instead considering ones that feature alternative amino acid residues at the site under consideration or, if all such residues have been exhausted, moving on to the next site.

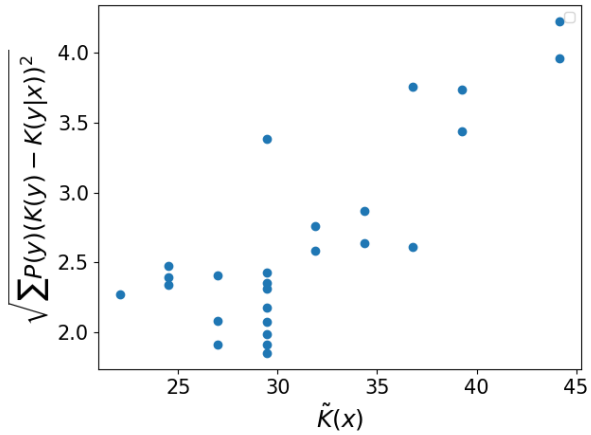
We generated a sample of 4000 neutral genotypes, from which we then generated the complete point-mutational neighborhood of a random subsample of 200 sequences, corresponding to a fraction of 0.05 of the entire sample. Figure 2 is based on the predictions of the SS of all 40402 unique sequences of the neutral mutants from the subsample and their point mutational neighbors, with CDRH3 of the human SARS-CoV antibody with PDB entry 6WS6 [70, 77] as the starting point. The SS predictions for the CDRH3 are from running the Porter 5 algorithm on the full sequence of the antibody's heavy chain. For the wild-type H3 region of 6WS6, this results in an accuracy of 85% when compared with the experimental SS data in PDB. The primary structure for the region under study (using single-letter amino acid codes) is `ARDYTRGAWFGESLIGGFDN` [77], the secondary structure as recorded in the PDB entry - `EEECCCCCCCCCCCCCCEEE`, and the fast Porter 5 prediction based on the full sequence of the heavy chain - `EEECCCCCCCCCCCCCCCCC`.

For the CDRH3 of HIV-1-neutralizing antibody PG16 (PDB entry: 4DQO [90, 91]), we applied the same approach, except that we used the fast Porter 5 predictions for just the H3 region, which was much less computationally intensive but resulted in only 36% accuracy in predicting the SS of the wild type. The primary structure of the wild-type CDRH3 is `AREAGGPIWHDDVKYYDFNDGYYNYHYMDV` according to the IMGT definition as recorded in the structural antibody database SAbDab [92], the SS predicted by Porter 5 is `CCCCCCCCCCCCCEEEECCCCEEEEEECCC`, and the SS from PDB is `EEEEEECEEECCEEEEHHHHHCEEEEEEEE` [90]. The site-scanning run and the subsequent exploration of the point-mutational neighborhoods of neutral genotypes from the subsample generated 107330 mutant sequences. The data for this protein is presented in Figure 5.

3. Complexity

To approximate the Kolmogorov complexity we followed the methodology of ref. [30], in which the function

$$C_{LZ}(x) = \begin{cases} \log_2(n), & x = 0^n \text{ or } 1^n \\ \log_2(n)[N_w(x_1...x_n) + N_w(x_n...x_1)]/2, & \text{otherwise} \end{cases} \tag{A1}$$



(a)

FIG. 3. **The conditional complexity $\tilde{K}(y|x)$ is typically more different to $\tilde{K}(y)$ for more complex starting phenotypes** x. Using RNA $L = 30$ SS, we measure the average difference between $\tilde{K}(y|x)$ and $\tilde{K}(y)$. As expected from the discussion around Eq. (16), for simpler x the difference is typically small, while for more complex x the difference increases. The linear correlation has coefficient r= 0.81 with p-value$< 10^{-6}$.

was defined where N_w forms the basis for the 1976 Lempel and Ziv complexity measure [62]. Here the simplest strings 0^n and 1^n are separated out because $N_w(x)$ assigns complexity $K = 1$ to the string 0 or 1, but complexity 2 to 0^n or 1^n for $n \geq 2$, whereas the true Kolmogorov complexity of such a trivial string actually scales as $\log_2(n)$ for typical n, because one only needs to encode n. The minimum possible value is $K(x)\approx 0$ for a simple set, and so, e.g. for binary strings of length n we can expect $0 \lesssim K(x) \lesssim n$ bits. Because for a random string of length n the value $C_{LZ}(x)$ is often much larger than n, especially for short strings, we scale the complexity

$$\tilde{K}(x) = \log_2(M) \cdot \frac{C_{LZ}(x) - \min_x(C_{LZ})}{\max_x(C_{LZ}) - \min_x(C_{LZ})} \tag{A2}$$

where M is the maximum possible number of phenotypes in the system, and the minimum and maximum complexities are over all strings x which the map can generate. $\tilde{K}(x)$ is the approximation to Kolmogorov complexity that we use throughout. This scaling results in $0 \leq \tilde{K}(x) \leq n$ which is the desirable range of values.

Appendix B: Effect of the complexity of the starting phenotype x

In the discussion around Eq. (16), we suggested that for a very simple starting phenotype x, the conditional and unconditional complexities will likely be similar, i.e. $\tilde{K}(y|x) \approx \tilde{K}(y)$. For complex x, we suggested that potentially $\tilde{K}(y|x) \ll \tilde{K}(y)$ for some y, especially those that are similar to x. Even though $\tilde{K}(y|x)$ and $\tilde{K}(y)$ may differ substantially only for a small fraction of possible y, because of the exponential dependence of $P(x \to y)$ on the value of $\tilde{K}(y|x)$, the small fraction of SS may potentially account for a large fraction of the probability $P(x \to y)$. As a brief investigation into the effect of the complexity of x on the similarity of $\tilde{K}(y|x)$ and $\tilde{K}(y)$, in Figure 3 we show a plot of the root-mean-square difference between these values, i.e. $\sqrt{\sum_y P(y)(\tilde{K}(y) - \tilde{K}(y|x))^2}$, vs the complexity of x. The Figure shows clearly that for lower complexity x, the difference is small, as expected. Interestingly, the difference grows linearly with increasing complexity of x (linear correlation r= 0.81 with p-value$< 10^{-6}$). To generate these data, we used 25 RNA SS of length $L = 30$, with 20 000 steps in the site-scanning method saving a fraction of 0.1 of the sequences, i.e. the same methodology as described above in Section A 1.

Appendix C: Supplemental Figures

Here we show some extra figures, each of which shows qualitatively the same patterns as in those from the main text.

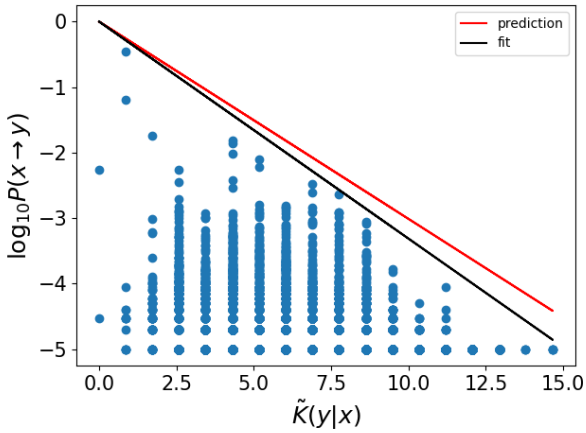
Plots for an RNA of length $L = 40$ nucleotides are shown in Figure 4. Although based on a different starting phenotype, these plots look qualitatively very similar to the RNA figure in the main text.

Figure 5 contains plots for the SS of a CDR H3 hypervariable region from the HIV-1-neutralizing antibody PG16 antibody with PDB entry 4DQO [90] and CDR H3 length $L = 30$. However, the SS prediction algorithm Porter 5 does not predict the original starting protein SS well (accuracy only 36% when compared with the PDB SS data [70]) and hence the probabilities

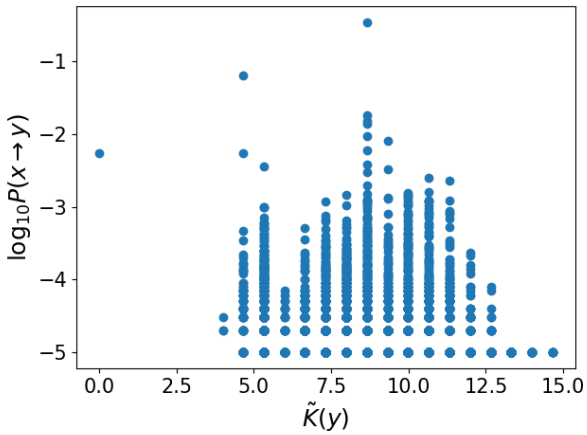
and structures underlying the figure probably do not reflect the true ones that would be obtained via in vitro mutational analysis, if this were performed.

$\tilde{K}(y\|x)$	$P(x \to y)$	y
0.0	0.055	.....
0.8	0.34	.((.....(((((.....)))))).....)).....*
1.7	0.018	(((.....(((((.....)))))).....)))....
2.6	0.0056	.....((((((.....)))))).....
3.4	0.0012	.....((.(((((.....))))))).....
4.3	0.015	.((.....((((((.....)))))).....)).....
5.2	0.0080	.((.....((((.....))))).....)).....
6.0	0.0016	.((((...(((((.....)))))).....)).)).....
6.9	0.0033	.((...(((((((.....)))))))).....)).....
7.8	0.0023	.((..((.(((((.....)))))).)).....)).....
8.6	0.00088	.((.(((.(((((.....)))))..)))...)).....
9.5	0.00017	.((..(((((((((.....)))))).....)))..))..
10.3	0.000050	.((.....(.(.(((....))).).).....)).....
11.2	0.000090	.((....(((.(.(.....).).)))....)).....
12.1	0.000010	(((((.(..(((.....((.....)).)))..)))).)))
12.9	0.000010	((((..((((.....)).)))))))((.(((...))).)).
13.8	0.000010	(((.(.....).))).(((.(((..(....)..)))))).
14.7	0.000010	((.((...)).)).(((.(((..(.....)..))).)))

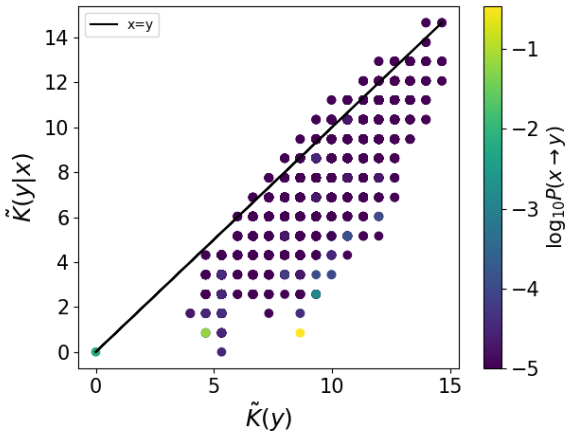
(a)



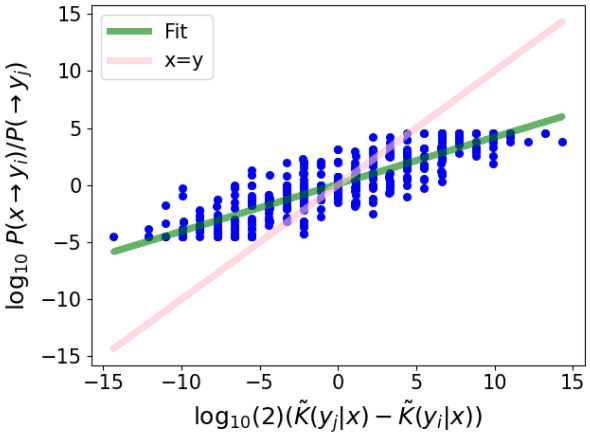
(b)



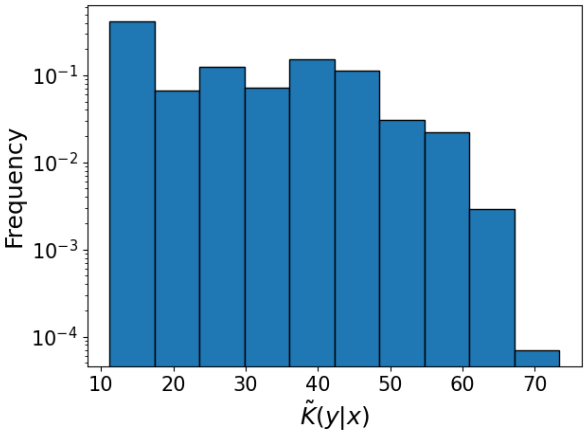
(c)



(d)



(e)

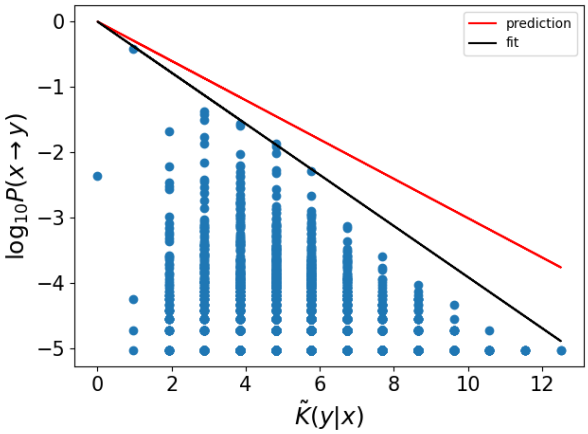


(f)

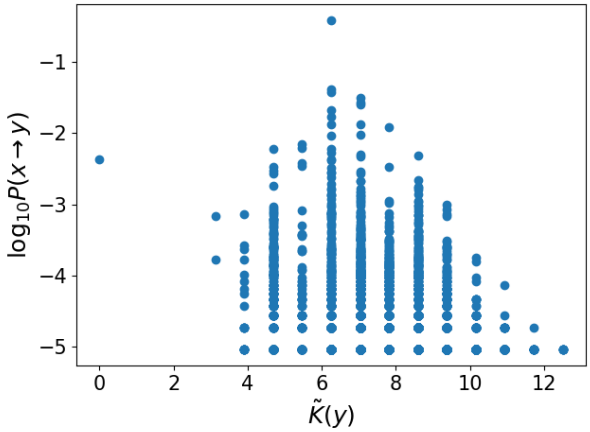
FIG. 4. **RNA secondary structure transition probabilities for a sequence of length $L = 40$ nucleotides.** $x =$ `.((.....(((((.....)))))).....)).....` (a) Table illustrating the SS with the highest probability for each complexity value. The starting phenotype x is marked with an asterisk (*). (b) Transition probabilities $P(x \to y)$ decrease exponentially with increasing conditional complexity $\tilde{K}(y|x)$, upper bound of Eq. (10) depicted in black. Predicted upper bound (red) and fitted bound (black) are close. (c) The unconditional complexity $\tilde{K}(y)$ does not predict the transition probabilities well. (d) $\tilde{K}(y|x) \approx \tilde{K}(y)$ for most y, leading to a positive linear correlation between values. (e) Ratios of probabilities correlate strongly with differences in conditional complexity. (f) The histogram of conditional complexity values shows a roughly flat distribution (on a log scale), but with some bias towards simplicity.

$\tilde{K}(y\|x)$	$P(x \to y)$	y
0.0	0.0043	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
1.0	0.38	CCCCCCCCCCCCCEEEECCCCEEEEEECCC*
1.9	0.021	CCCCCCCCCCCCCEEEECCCCEEEEECCCC
2.9	0.042	CCCCCCCCCCCCCCEEECCCCEEEEEECCC
3.8	0.027	CCCCCCCCCCCCCEEEECCCCEEEEECCC
4.8	0.014	CCCCCCCCCCCCCCEEECCCCEEEEECCC
5.8	0.0051	CCCCCCCCCCCCCCEECCCCCEEEEECCC
6.7	0.00073	CCCCCCCCCCCCCEEEECCCCEEEHEECCC
7.7	0.00025	CCCCCCCCCCCCCCEEECCCCEEEHEECCC
8.7	0.000093	CCCCCCCHHHHHHHHEECCCCEEEEECCC
9.6	0.000047	CCCCCCCCCCCCCCEECCCCEEEHEHCCC
10.6	0.000019	CCCCCCCCCCCCCEEEHHCCCEEHHEHCCC
11.5	0.0000093	CCCCCHHHHCCCCHEHCCCCCEEEHECCC
12.5	0.0000093	CCCCCHHHCCCCCCEHHHCCHHHHEEHCCC

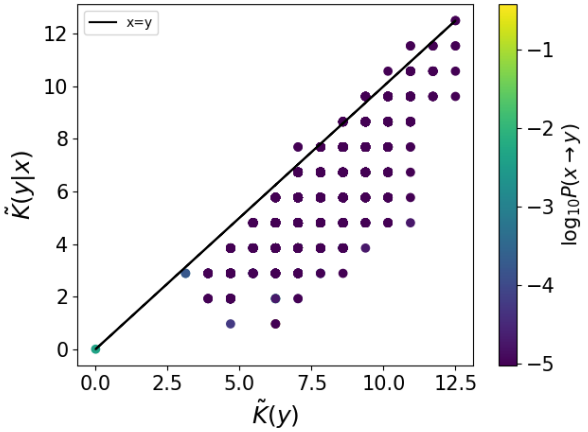
(a)



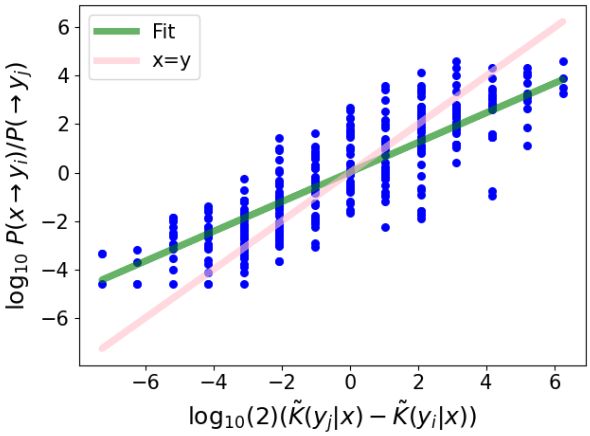
(b)



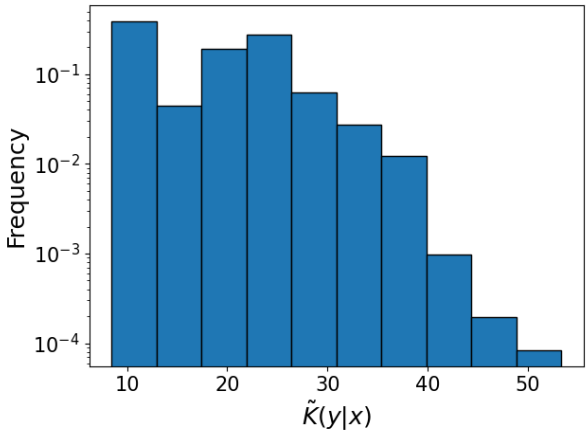
(c)



(d)



(e)



(f)

FIG. 5. **Protein secondary structure transition probabilities for the CDR H3 hypervariable region in the antibody with PDB entry 4DQO [90] and length $L = 30$ amino acids, $x =$CCCCCCCCCCCCCCEEEECCCCEEEEEECCC.** (a) Table illustrating the highest-probability SS for each conditional complexity value. The starting phenotype x is marked with an asterisk (*). (b) The predicted upper bound (red) and fitted bound (black) are similar. (c) The unconditional complexity $\tilde{K}(y)$ does not predict the transition probabilities well. (d) $\tilde{K}(y|x) \approx \tilde{K}(y)$ for most y, leading to a positive linear correlation between values. (e) Ratios of probabilities correlate strongly with differences in conditional complexity. (f) The histogram of conditional complexity values shows a bias towards simpler proteins with lower complexity values.