Using small samples to estimate neutral component size and robustness in the genotype-phenotype map of RNA secondary structure

Marcel Weiß[1,2*] and Sebastian E. Ahnert[1,2]
[1] Theory of Condensed Matter Group, Cavendish Laboratory, University of Cambridge, Cambridge, UK
[2] Sainsbury Laboratory, University of Cambridge, Cambridge, UK

(Dated: April 11, 2020)

In genotype-phenotype (GP) maps, the genotypes that map to the same phenotype are usually not randomly distributed across the space of genotypes, but instead are predominantly connected through one-point mutations, forming network components, that are commonly referred to as neutral components (NCs). Due to their impact on evolutionary processes, the characteristics of these NCs, like their size or robustness have been studied extensively. Here, we introduce a framework that allows the estimation of NC size and robustness in the GP map of RNA secondary structure. The advantage of this framework is that it only requires small samples of genotypes and their local environment, which also allows experimental realisations. We verify our framework by applying it to the exhaustively analysable GP map of RNA sequence length $L = 15$, and benchmark it against an existing method by applying it to longer, naturally occurring functional non-coding RNA sequences. Although it is specific to the RNA secondary structure GP map in the first place, our framework can likely be transferred and adapted to other sequence-to-structure GP maps.

I. INTRODUCTION

Genotype-phenotype (GP) maps are of fundamental importance to biological evolution and their properties have been studied extensively in a range of biological contexts. Examples include the mapping between RNA sequences and their secondary structure [1–5], the GP maps of transcription factor binding sites [6], gene regulatory [7, 8], and metabolic [9] networks, as well as more abstract systems, like the GP maps of the HP lattice model of protein folding [5, 10–12], the Polyomino model [13, 14] of protein quaternary structure, and even digital evolving organisms [15].

A GP map can be seen as a network, in which the nodes correspond to genotypes and the edges to one-point mutations between the genotypes [16, 17]. Genotypes that map to the same phenotype form a 'subnetwork' that is commonly referred to a neutral network (NN) or neutral set, as a 'neutral' one-point mutation does not change the phenotype. A NN can be fully connected, or consist of several disjoint connected components, which are commonly referred to as neutral components (NCs).

Over the years, studies have revealed that GP maps share several properties [5, 14, 18, 19] – which are believed to be universal for most GP maps – and that these properties have a strong impact on evolutionary processes [20, 21]. Most of these properties are related to global characteristics of a GP map, like the size, robustness and evolvability of phenotypes (NNs) and NCs. The size of a phenotype (or NC) refers to the number of genotypes that map to this phenotype (or that are part of the NC). For phenotypes, the size is often normalised by the total number of genotypes and referred to as the phenotype frequency. The robustness of a phenotype (or NC) refers to the fraction of one-point mutations of a genotype that are neutral, averaged over all genotypes mapping to this phenotype (or that are part of the NC) [3]. The evolvability of a phenotype (or NC) refers to the number of distinct alternative phenotypes that are accessible through one-point mutations from any genotype mapping to this phenotype (or that is part of the NC) [3]. The properties that are consistently observed across several different GP maps are: (a) a highly skewed distribution of the phenotype (or NC) sizes, with a small number of very large sizes, and many small ones [1, 3–5, 11, 12, 14, 22]; (b) a linear scaling of the robustness with the logarithm of the size [4, 14, 18, 22]; (c) a positive correlation between the robustness and evolvability [3, 6, 14, 23]. All of these properties have been found at both the phenotype and NC level. Recently, simple analytical models have been used to show that these properties are mainly caused by the organisation of genotype sequences into constrained and unconstrained parts, together with non-local effects of mutations [24–26].

Quantifying the exact size, robustness or evolvability of a phenotype or NC requires the exhaustive enumeration of all respective genotypes and their one-point mutational neighbourhood. In most cases, this is an unfeasible task due to the extraordinary size and complexity of most GP maps. This is a particular challenge for experimental studies, which usually can only analyse small regions of genotype space. As a result, there have been efforts to estimate the characteristics from samples of the GP map. *Jörg et al.* [22] introduced an algorithm and software that allows NN size and robustness estimations of RNA secondary structure phenotypes. It uses an RNA secondary structure prediction software and is based on a 'Nested Monte Carlo approach'. Drawbacks of this approach are that it requires a large sample of genotypes and a significant number of computational steps and that it does not allow an estimation of NC characteristics. *Aguirre et al.* [4] considered a simple analytical formula to estimate

[*] mw636@cam.ac.uk

NC sizes in the RNA secondary structure GP map in order to derive the scaling of the NC robustness with the logarithm of the NC size. Recently, *García-Martín et al.* [27] verified another method to estimate NC sizes for several computationally tractable GP maps. Both of the latter methods are based on the exhaustive enumeration of the full NCs, and not designed for estimates to be made from small samples in the first place.

In this article, we introduce a framework to estimate NC sizes and robustness values that does not require the exhaustive enumeration of full NCs. Instead, it is designed to allow estimations from small samples of genotypes from the NCs of interest, thereby facilitating an experimental applicability. We apply the framework to the RNA secondary structure GP map, but it is likely transferable and adaptable to other sequence-to-structure GP maps.

Due to its computational tractability, the GP map of RNA secondary structure is one of the most widely studied GP maps. Readily available prediction software can be used to predict the secondary structures (phenotypes) of RNA sequences (genotypes), thereby allowing an exhaustive enumeration of the GP map for short sequence lengths. Throughout the analysis in this article, we use the so-called *ViennaRNA* package [28–30] as prediction software. In detail, we use version 2.4.9, default parameters, the Python implementation and the function *RNA.fold*. For a given sequence, this function returns the secondary structure with minimum free energy, which we consider as the phenotype. Figure 1(A) shows examples of genotypes and phenotypes for the GP map of sequence length $L = 12$, i.e. sequences consisting of twelve nucleotides. For the RNA secondary structure GP map, it has been shown that NN fragmentation into NCs is common at least for short sequence lengths. *Schaper et al.* [23] showed that a NN 'typically fragments into at least 2^n NCs, often of similar size', where n is the number of base pairs in the respective phenotype structure. This is explained by the fact that out of the six possible nucleotide combinations for a base pair, only three are connected through one-point mutations, respectively: CG↔UG↔UA and GC↔GU↔AU, leading to potentially two disconnected sequence sets for each base pair [23]. In Figure 1(B), an example NC is shown.

This article is structured as follows. In the first part, we introduce the theory of our framework. In the second part, we verify it by applying it to the NCs of the exhaustively analysable GP map of sequence length $L = 15$. Finally, we consider naturally occurring functional non-coding RNA sequences taken from the fRNA database [31, 32] and use these to benchmark our framework against an existing method.

II. THEORY

In our framework, the starting point for estimations is a sample of S genotypes from the NC of interest. Later, we will introduce and validate different sampling methods to generate such a sample. Using a (small) sample limits the information that can be used for the estimations, as we are mostly restricted to the local properties of the sample genotypes. We base our NC size and robustness estimations on the neutral mutations per sequence site, in particular their sample average and sample standard deviation. These quantities can be determined from measurements of the one-point mutational neighbourhoods of the sample genotypes experimentally or computationally.

The procedure is schematically depicted in Figure 1(C). Firstly, for each genotype in the sample, the number of neutral mutations (i.e. mutations that do not change the phenotype) per sequence site is measured. We label these quantities $x_{i,j}$, where $i \in \{1, \ldots, S\}$ runs over the genotypes in the sample and $j \in \{1, \ldots, L\}$ over the sites of the sequences of length L. By definition, for RNA secondary structure, $x_{i,j}$ can only take the discrete values 0, 1, 2 and 3, while 0 corresponds to a fully constrained site – likely a paired site – and 3 to a fully unconstrained site – likely an unpaired site. Secondly, the sample average $\overline{x_j}$ and sample standard deviation σ_j of the number of neutral mutations for each site j are computed as follows:

$$\overline{x_j} = \frac{1}{S} \sum_{i=1}^{S} x_{i,j} \tag{1}$$

$$\sigma_j = \sqrt{\frac{1}{S-1} \sum_{i=1}^{S} \left(x_{i,j} - \overline{x_j}\right)^2} \tag{2}$$

The division by $S-1$ instead of S in the sample standard deviation is a common correction accounting for the fact that the sample is smaller than the source it is drawn from, here the NC of interest.

A. NC size estimation

As mentioned in the introduction, NC size estimations have been considered before, however not by using samples, but by using all genotypes of the NC. The simple NC size estimation introduced by *Aguirre et al.* [4] for the RNA secondary structure GP map is based on the grouping of sites into unpaired sites and base pairs. For each of the two groups, the average number of neutral mutations is considered, averaged over all genotypes of the NC and all group members. Then, the NC size is estimated by the product of the two quantities (each plus 1) to the power of the number of group members, respectively. The NC size estimation introduced by *García-Martín et al.* [27] is not specific to a particular GP map, less coarse-grained and considers sites individually by estimating the NC size in terms of the product of the versatility of each site, which is related to the letter distribution at that site across all genotypes of the NC. In principle, this method would also work with samples but would require sample genotypes to be significantly different, and larger sample sizes to obtain meaningful letter distributions and versatility values.

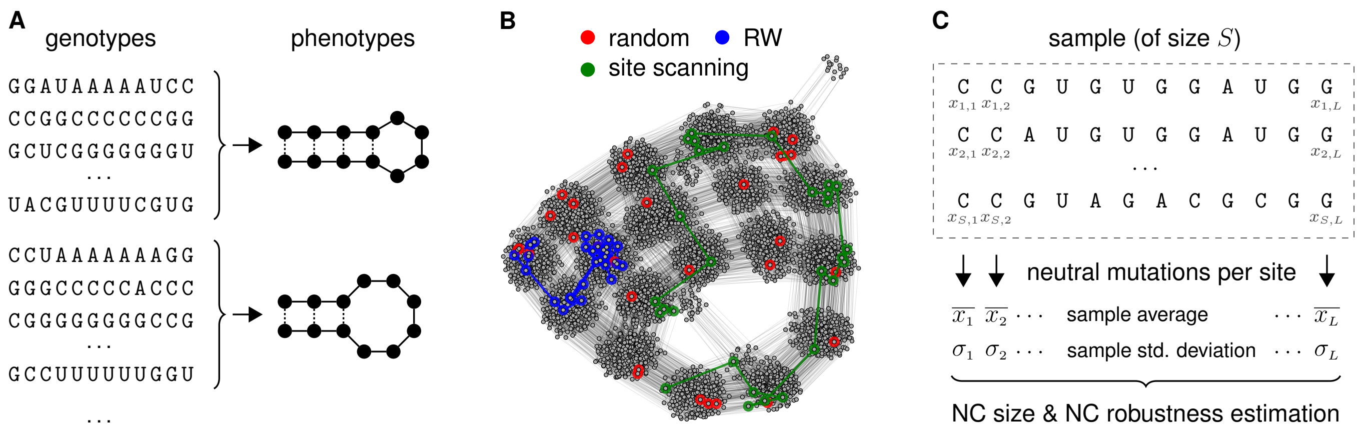
A
genotypes
phenotypes
GGAUAAAAAUCC
CCGGCCCCCCGG
GCUCGGGGGGGU
...
UACGUUUUCGUG
CCUAAAAAAAGG
GGGCCCCCACCC
CGGGGGGGGCCG
...
GCCUUUUUUGGU
...
B
random
RW
site scanning
C
sample (of size S)
C C G U G U G G A U G G
$x_{1,1}$ $x_{1,2}$ $x_{1,L}$
C C A U G U G G A U G G
$x_{2,1}$ $x_{2,2}$ $x_{2,L}$
...
C C G U A G A C G C G G
$x_{S,1}$ $x_{S,2}$ $x_{S,L}$
neutral mutations per site
$\overline{x_1}$ $\overline{x_2}$ ··· sample average ··· $\overline{x_L}$
σ_1 σ_2 ··· sample std. deviation ··· σ_L
NC size & NC robustness estimation

FIG. 1. (A) Example genotypes and phenotypes for the RNA secondary structure GP map of sequence length $L = 12$. (B) Example NC together with highlighted samples of 30 genotypes generated by uniformly random sampling, random walk (RW) sampling, and site scanning sampling. Each node represents a genotype, each edge a one-point mutation. The NC is one of the NCs of the first phenotype in (A) (NC rank: 129, size: 4663). (C) Schematic depiction of the measurements on the sample genotypes required for the NC size and robustness estimation: For each genotype of the S sample genotypes, the number of neutral mutations per sequence site ($x_{i,j}$, $i \in \{1, \ldots, S\}$, $j \in \{1, \ldots, L\}$) is measured in order to determine the respective sample averages ($\overline{x_j}$, $j \in \{1, \ldots, L\}$) and sample standard deviations (σ_j, $j \in \{1, \ldots, L\}$).

Our estimation method is similar to the approach of [27] in the sense that it also examines sites individually, but it exploits local properties of the GP map. We estimate the NC size $s_{\mathrm{NC,est}}$ from a sample as follows:

$$s_{\mathrm{NC,est}} = \prod_{j=1}^{L} \begin{cases} \max\left(1 + \overline{x_j} + \alpha \cdot \sigma_j, 4\right) & \text{if } j \text{ unpaired} \\ \max\left(1 + \overline{x_j} + \alpha \cdot \sigma_j, 2\right) & \text{if } j \text{ paired} \end{cases} \tag{3}$$

Without the term $\alpha \cdot \sigma_j$, the formula would simply assume that each sequence site j can be occupied by $1 + \overline{x_j}$ different letters (nucleotides) independent of the other sites and that the estimate of the NC size is given by the product of these factors over all sites. In fact, there can be dependencies between individual sites, for example epistatic effects, that affect the number of neutral mutations of one site depending on the occupation of other sites. This in turn means that the average of the number of neutral mutations for a site can underestimate its full versatility and so the NC size estimate. We account for this by adding the standard deviation σ_j to the average number of neutral mutations $\overline{x_j}$ for each site j with a factor $\alpha \geq 0$. We limit the factor $1 + \overline{x_j} + \alpha \cdot \sigma_j$ to a maximum of 4 if the site is unpaired, and 2 if it is paired to prevent unphysical factors. In the case of RNA secondary structure, the alphabet consists of four letters and there can only be a maximum of two different letters at a particular paired site across the sequences forming a NC. For other GP maps, these values would need to be replaced by the respective alphabet size and other constraints.

The correction factor α accounts for the amount of correlation between sites, which we assume to be dependent on the sequence length L: $\alpha = \alpha(L)$. In a first step, we validate the NC size estimation formula by applying it to the exhaustively analysable RNA secondary structure GP maps of sequence lengths $L = 12$ to $L = 16$, working with the full set of genotypes for each NC instead of samples (see Supplementary Information I). In this case, we find that the NC size estimation works very well and optimal results (defined in terms of the minimal root-mean-square deviation between estimated and true NC sizes) are achieved for $\alpha_{\mathrm{opt}} \approx 0.43$ ($L = 12$), $\alpha_{\mathrm{opt}} \approx 0.44$ ($L = 13$), $\alpha_{\mathrm{opt}} \approx 0.44$ ($L = 14$), $\alpha_{\mathrm{opt}} \approx 0.46$ ($L = 15$) and $\alpha_{\mathrm{opt}} \approx 0.47$ ($L = 16$). This shows the sequence length dependence of the correction parameter and a likely increase of correlations. Later in this article, we will address this problem in more detail and derive an analytical formula for this dependence when considering longer, naturally occurring functional non-coding RNA sequences.

B. NC robustness estimation

For NCs of the RNA secondary structure GP map, it has been shown that the degrees of the nodes in a NC network follow a distribution that is single-peaked [4], implying that the same holds for the distribution of the robustness values of individual genotypes in a NC. Since the robustness of a NC is just defined by the average of the genotype robustness values, we estimate the NC robustness $r_{\mathrm{NC,est}}$ from a sample simply by averaging over the robustness values of the sample genotypes:

$$r_{\mathrm{NC,est}} = \frac{1}{S} \sum_{i=1}^{S} r_{\mathrm{g},i} = \frac{1}{3L} \sum_{j=1}^{L} \overline{x_j} \tag{4}$$

where $r_{\mathrm{g},i}$ refers to the robustness of sample genotype i, which can be calculated from the number of neutral mutations per site, resulting in a formula only using the average number of neutral mutations per site.

III. APPLICATION TO EXHAUSTIVELY ANALYSABLE NCS

We verify the framework by applying it to the NCs of the $L = 15$ RNA secondary structure GP map. This GP map and its NCs can be exhaustively analysed, allowing an exact computation of the true NC sizes and robustness values. In total, we find 8792 NCs ignoring the undefined phenotype (unbound structure). We rank them according to their size, with the largest having rank one.

A. Sampling methods

In general, we are interested in samples that broadly cover and best represent the NCs and that allow to extract as much information as possible for the estimations. Here, we compare three main sampling methods: uniformly random sampling, random walk (RW) sampling, and site scanning sampling. In the following, we explain the different methods, the detailed algorithms of which can be found in the Supplementary Information II A. In Figure 1(B), examples of samples generated using the three different methods are highlighted in the NC shown.

Uniformly random sampling simply considers a sample of genotypes from the NC of interest, chosen with equal probability. It is only applicable if the full NC is known and serves as a reference here.

RW sampling and site scanning sampling are always applicable and generate connected samples through one-point mutations. RW sampling considers steps comprising the random selection of a site and the random selection of a letter to which the letter at this site is mutated to. The mutation is done if neutral, otherwise both steps are repeated. Starting from a randomly selected starting genotype on the NC, the process is repeated until a sample of S genotypes is obtained. This sampling favours mutations of less constrained sites over more constrained sites, because those mutations are more likely to be neutral. Site scanning sampling is a novel method that is designed to overcome this effect. Starting again from a randomly selected starting genotype on the NC, we 'scan' the sequence sites periodically from left to right for neutral mutations in the following way: For the first site of the initial genotype, we perform a random mutation. If this turns out to be neutral, we retain this mutation and continue with the second site of the new genotype. If the mutation of the first site is not neutral, we randomly test all remaining potential mutations for the first site until a neutral one is found, and then continue with the second site of the new genotype. If no neutral mutation at all is found for the first site, we repeat the process with the second site of the initial genotype, and so on. Again, the process is repeated until a sample of S genotypes is obtained. The periodic scanning of sites ensures that all sites of a sequence are tested for potential neutral mutations before an individual site is tested and potentially mutated the next time.

For the GP map of RNA secondary structure, NC networks have been found to be assortative and to exhibit a community structure [4, 33], affecting dynamics on the network [33–35]. The examples in Figure 1(B) show that RW sampling tends to spread less broadly and is more constrained to individual communities of the NC network compared to site scanning sampling, which travels more across the communities.

Alternative sampling approaches would be a depth- or breadth-first search of the NC networks. While a depth-first search would be quite similar to RW sampling for small samples for the considered NC networks due to the high node degrees, a breadth-first search would lead to an even more local exploration for small samples.

For RW sampling and site scanning sampling, we additionally consider a random subsampling step. This is performed by randomly selecting S_r genotypes uniformly from the generated sample of size S, and then only using these S_r genotypes for the NC size and robustness estimation. This aims to reduce correlations between the genotypes in the initially generated connected sample. It also reduces the number of genotypes for which the one-point mutational neighbourhood needs to be measured in order to determine the number of neutral mutations per site, required for the estimations. More details on random subsampling, and how we measure the number of neutral mutations per site at this stage can be found in the Supplementary Information II A.

For each of the considered NCs, and for each of the sampling methods, we generate $N_S = 100$ independent samples, respectively, for a range of genotype sample sizes S and a fixed random subsample size S_r. For more details, see the Supplementary Information II A.

B. Results

Figure 2 shows the estimation results for nine example NCs. Details of these NCs can be found in the Supplementary Information III. For the NC size estimations, eq. (3) together with the found optimal value of $\alpha_{\mathrm{opt}} \approx 0.46$ for $L = 15$ (see Supplementary Information I) is used. We find that there is an overall satisfactory agreement between the estimations and true values, though there are significant differences between the different sampling methods, sample sizes and random subsampling, as well as individual NCs.

In terms of the sampling methods, overall, uniformly random sampling tends to lead to more accurate estimations and less spread than site scanning sampling, which itself tends to lead to better estimations than RW sampling. For larger sample sizes, the differences tend to be less significant. Of particular relevance is the observation that the estimations from a random subsample from a bigger full sample in the case of RW sampling and site scanning sampling tend to be better than those from a full sample of the same size (see Figure 2(iv) compared to (i)).

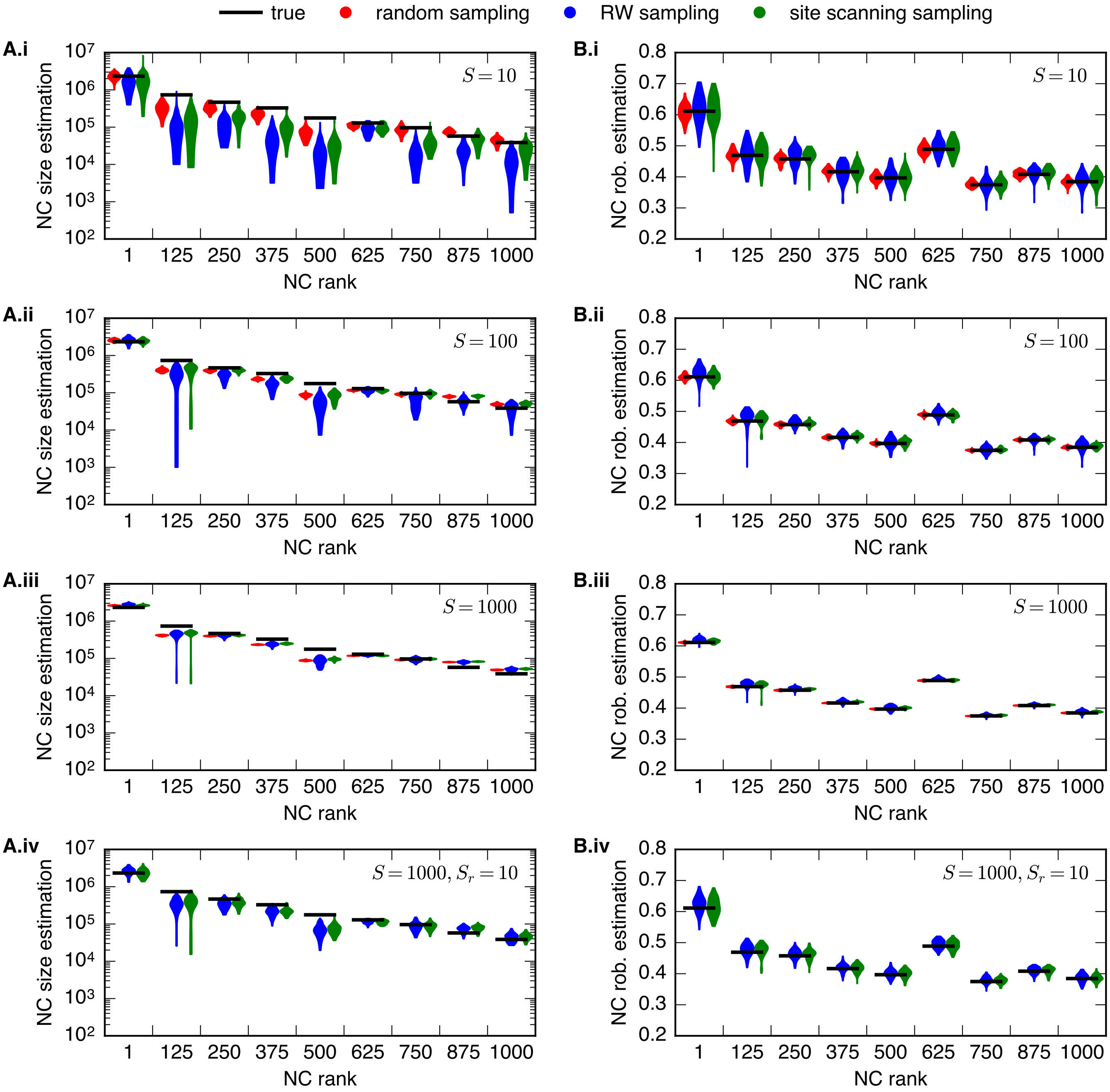

FIG. 2. (A) NC size and (B) NC robustness estimation results for nine example NCs of the $L = 15$ RNA secondary structure GP map considering uniformly random sampling, random walk (RW) sampling, and site scanning sampling and a sample size of (i) $S = 10$, (ii) $S = 100$, (iii) $S = 1000$, and (iv) $S = 1000$ reduced to a random subsample of size $S_r = 10$. Details of the respective NCs can be found in the Supplementary Information III. For the NC size estimations, eq. (3) together with the found optimal value of $\alpha_{\text{opt}} \approx 0.46$ for $L = 15$ (see Supplementary Information I) is used. The violin plots indicate the spread for 100 independent samples, respectively. Overall, there is a satisfactory agreement between the estimations and true values, while the accuracy tends to increase both with increasing sample size and from RW sampling over site scanning sampling to uniformly random sampling. Further, random subsampling (iv) tends to lead to better estimations than simple samples of the same size (i).

The NC robustness estimations, in all cases, are centred around the true values with narrowing spreads for larger sample sizes. This is due to the similarity of our NC robustness estimation method (average over sample) and the way NC robustness is defined (average over NC).

In contrast, the NC size estimations tend to be slightly too small for small sample sizes and also differ from the true values for the larger sample sizes. The systematic error for small sample sizes is likely due to the dependence of our NC size estimation formula on sample size dependent quantities like the average and the standard deviation of the number of neutral mutations per site that do not fully reflect the NC for small sample sizes. For a more detailed explanation see Supplementary Information IV. The error for large sample sizes is because our NC size estimation method never leads to fully exact estimations even if the sample covers the full NC (see Supplementary Information I).

The differences between individual NCs are likely the result of differences in the size, the spread of genotype diversity, and the topology of the NCs, which will affect the effectiveness of our estimation framework as well as the accessibility of different parts of the NC to the sampling approaches, and therefore the accuracy of the estimations.

In a next step, we quantify the quality of the estimates by considering the root-mean-square deviation (RMSD) from the true values. For the NC size and robustness, respectively, this is:

$$\mathrm{RMSD}_s = \sqrt{\frac{1}{N_{R_{\max}}} \sum_{i=1}^{N_{R_{\max}}} \left(\frac{1}{N_S} \sum_{j=1}^{N_S} \left(\log_{10}\left(s_{\mathrm{NC,est},i,j}\right) - \log_{10}\left(s_{\mathrm{NC,true},i}\right)\right)^2 \right)} \tag{5}$$

$$\mathrm{RMSD}_r = \sqrt{\frac{1}{N_{R_{\max}}} \sum_{i=1}^{N_{R_{\max}}} \left(\frac{1}{N_S} \sum_{j=1}^{N_S} \left(r_{\mathrm{NC,est},i,j} - r_{\mathrm{NC,true},i}\right)^2 \right)} \tag{6}$$

In both cases, the first sum runs over the NCs from rank 1 to $N_{R_{\max}} = 1000$. For the quantification, we restrict ourselves to the 1000 largest NCs, because they are all significantly larger than the maximum considered sample size (NC of rank 1000 has a size of 38579), and because their sizes only span about two orders of magnitude and so the RMSD is not affected by too much variation in the considered NC sizes, though changing this threshold does not affect the qualitative result. The 1000 largest NCs cover about 94.2% of the genotype space that leads to a defined phenotype (bound structure). The second sum runs over the $N_S = 100$ independent samples considered for each NC, respectively. For the NC size estimations, we consider order of magnitude deviations. In Figure 3, these quantities are shown as a function of the sample size and with and without random subsampling.

For the NC size, in all cases, the root-mean-square deviation RMSD_s decreases with sample size. This is because the greater coverage of the NCs with increasing sample size likely leads to a greater diversity of sample genotypes, a better balancing of outliers, and therefore to more accurate estimations.

For uniformly random sampling, the RMSD_s most strongly decreases with sample size and quickly starts to saturate above a sample size of about $S = 10$. This highlights that above a certain sample size no further randomly selected genotypes are required to represent the NCs for our size estimations. For RW sampling and site scanning sampling, the decrease is less strong and saturation sets in above larger sample sizes, likely due to the slowed exploration of the NC and the less diverse connected genotype samples generated by both sampling methods. However, NC size estimations using site scanning sampling perform significantly better than those using RW sampling for sample sizes before saturation. This is likely due to the broader spread of site scanning samples over the NC network compared to RW samples, which is by design as discussed earlier.

A further observation is that the results for full samples using RW sampling or site scanning sampling are only slightly better than those achieved using random subsamples from the same full samples, highlighting that the subsamples suffice to represent the full sample from which they are selected from. Furthermore, subsamples of $S_r = 10$ genotypes taken from larger full samples, e.g. of size $S = 100$ or $S = 1000$, lead to much lower RMSD values than full samples of size $S = 10$, likely because of the greater diversity between genotypes in the subsamples.

For the NC robustness, the root-mean-square deviation RMSD_r shows a similar sample size dependence. However, in contrast to the RMSD_s of the NC size, the RMSD_r tends to zero with increasing sample size. As mentioned before, this is because our NC robustness estimate as calculated in eq. (4) approaches the NC robustness definition as the sample size approaches the NC size. If we use random subsampling, the RMSD_r tends to saturate at a value above zero, because there is a 'minimum' spread of the NC robustness estimations around the true values for significantly small sample sizes. In this case the saturation value of the RMSD_r is about the same as the RMSD_r value for pure uniformly random sampling with the same sample size.

An alternative way to quantify the quality of our estimations is by considering the average difference between the estimated and true values and by considering the standard deviation of the differences. This is shown in the Supplementary Information IV. The results confirm that over a range of small sample sizes, our NC size estimates tend to increase on average with sample size before saturating. The results also indicate that for both RW sampling and to a lesser degree site scanning sampling, our approach tends to slightly overestimate on average

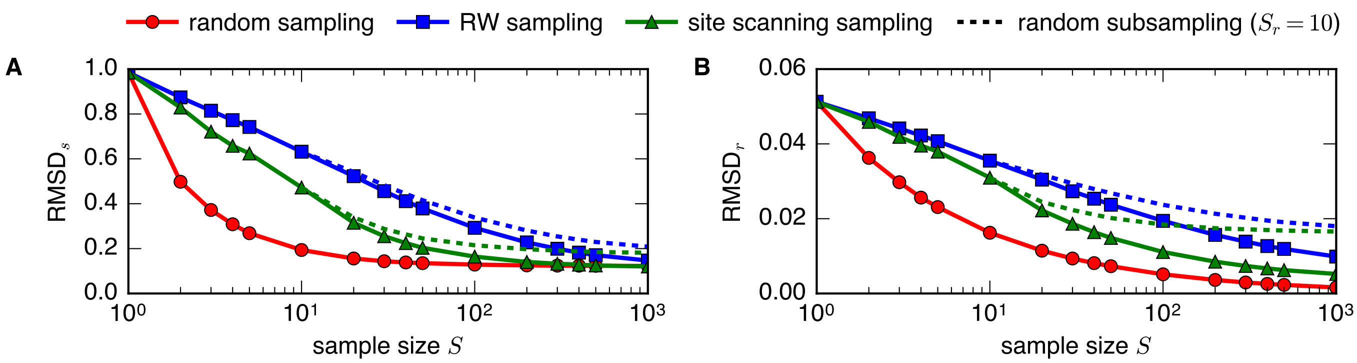

FIG. 3. Root-mean-square deviation (RMSD) (see eqs. (5) and (6)) between the (A) NC size (order of magnitude) and (B) NC robustness estimations and the true values for the 1000 largest NCs of the $L = 15$ RNA secondary structure GP map, as a function of sample size S, for uniformly random sampling, random walk (RW) sampling, and site scanning sampling. For the latter two, the dotted lines indicate random subsampling of fixed size $S_r = 10$ from respective full samples (sizes on the x-axis) larger than $S = 10$. For the NC size estimations, eq. (3) together with the found optimal value of $\alpha_{\mathrm{opt}} \approx 0.46$ for $L = 15$ (see Supplementary Information I) is used. The results confirm that apart from uniformly random sampling, site scanning sampling leads to better estimation results than RW sampling at least for smaller sample sizes. Furthermore, random subsampling of fixed size $S_r = 10$ from larger full samples ($S > 10$) produces better estimation results than simple full samples of size $S = 10$, i.e. the dashed lines lie significantly below the RMSD values at $S = 10$.

NC robustness. By definition, a simple random walk on a network is biased towards nodes with high degree, meaning a high genotype robustness for NC networks. This is likely the case for RW sampling and to a lesser degree also for site scanning sampling. We will come back to this result at a later point.

IV. APPLICATION TO FUNCTIONAL NON-CODING RNA SEQUENCES

In the second part of this article, we apply our framework to longer, naturally occurring functional non-coding RNA sequences, for which an exhaustive analysis is not feasible. We use the functional RNA database fRNAdb [31, 32] (`http://www.ncrna.org/`) as source. For a set of sequence lengths ranging from $L = 20$ to $L = 100$, we extract all sequences stored in the fRNAdb. In a filtering process, we remove all sequences for which the function or relevance of the secondary structure is not yet confirmed as well as all sequences that include non-standard nucleotides or for which *RNA.fold* returns the undefined phenotype (unbound structure). More details on the dataset and the filtering can be found in the Supplementary Information V. A similar dataset has been used for previous NN size estimations [21].

A. Benchmark

Since the true values of the NC or NN size and robustness of these sequences are not known, a new reference is required to benchmark our approach. We use the framework and software introduced by *Jörg et al.* [22], which can be used to estimate the NN size and robustness of a given RNA secondary structure. We will refer to the software as the *NNSE*. The *NNSE* has been used previously to estimate NN sizes of fRNAdb secondary structures [21, 22]. Similar to [21], we proceed as follows: First, we predict the secondary structure of the sequence of interest using *RNA.fold*. We then use this predicted secondary structure as an input for the *NNSE* using default parameters, apart from the number of measurements, which we set from 10 to 1 in order to reduce the computation time.

By default, the *NNSE* only returns a NN size estimate, which we label by $s_{\mathrm{NN,NNSE}}$. We apply the *NNSE* to all sequences from the fRNAdb dataset introduced before, but it does not converge in all cases (see Supplementary Information V). The *NNSE* can also return a NN robustness estimate in addition to the size estimate, which we label by $r_{\mathrm{NN,NNSE}}$. Because the robustness estimate requires a significant amount of additional computation time, we do not compute it for all sequences, but only for 100 randomly selected ones for each sequence length.

B. Modified sampling method

In the previous section, we showed that site scanning sampling outperformed RW sampling (see Figure 3), as it likely allows a broader exploration of the NC and therefore results in a more diverse sample of genotypes.

In order to reduce the computational costs, we consider an RNA-specific updated version of site scanning sampling. The detailed algorithm can be found in the Supplementary Information II B. Before, we could simply check if a mutation is neutral by checking if the mutated genotype is still in the fully known NC. Now, checking if a mutation is neutral – in principle – requires a call of *RNA.fold* on the mutated genotype and a comparison of the phenotype. The computational expense of this step increases with sequence length [22]. Therefore, in order to limit the number of calls to *RNA.fold*, we only consider mutations that do not break one of the six RNA secondary structure base pairs: CG, GC, AU, UA, GU, and UG. Whenever a one-point mutation affects a paired site, we only check the neutrality of the mutation if the mutated base pair is still one of the six base pairs, and otherwise consider the mutation as non-neutral. A similar restriction is used in the software of *Jörg et al.* [22]. We will refer to this approach below as 'accelerated site scanning sampling'.

As we showed above, random subsamples produce quite similar estimation results to the full samples from which they are taken, and better ones than full samples of the same size. Since subsampling reduces the number of genotypes for which the one-point mutational neighbourhood needs to be explored, it also significantly reduces the computational time required.

To even further reduce the computational costs, we apply the same base pair preserving principle as we employ for the accelerated site scanning sampling approach when measuring the one-point mutational neighbourhoods for the random subsample genotypes. We still go through all possible one-point mutations for every subsample genotype but whenever a one-point mutation affects a paired site, we only check if the mutation is neutral if the mutated base pair is still one of the six compatible combinations. This method is similar to how the mutational robustness is measured by the software of *Jörg et al.* [22]. More details on how we measure the number of neutral mutations per site at this stage, and on how we perform our simulations can be found in the Supplementary Information II B.

C. NC–NN extrapolation

No direct comparison between our NC estimates and the reference estimates from the *NNSE* is possible, because the latter returns NN size and robustness estimates, while our framework estimates NC size and robustness. For this reason, we consider extrapolations of our estimated NC characteristics to NNs.

We extrapolate our estimate of a NC size $s_{\mathrm{NC,est}}$ to a corresponding estimate of the NN size $s_{\mathrm{NN,est}}$ of the whole phenotype as follows:

$$s_{\mathrm{NN,est}} \approx s_{\mathrm{NC,est}} \cdot 2^n \tag{7}$$

where n is the number of base pairs in the secondary structure phenotype corresponding to the NC. This is based on the finding by *Schaper et al.* [23] that a NN 'typically fragments into at least 2^n NCs, often of similar size'. They come to this conclusion by studying the RNA secondary structure GP map exhaustively up to sequence length $L = 15$. In order to confirm their result, they consider sampling for length $L = 20$, for which they also find that for selected phenotypes, the sizes of the largest 2^n NCs do not differ more than one order of magnitude in most cases. However, it is a matter of debate whether this finding also holds for significantly longer sequence lengths. *García-Martín et al.* [27] argue that for longer sequence lengths 'either phenotypes are broken into few NCs, or one of these components is much larger than the others and dominates the abundance of the phenotype'. They argue based on their findings on sequence site versatility values across a range of sequence lengths and on work on genetic correlations in NCs by other authors.

Nevertheless, we stick to the extrapolation for all our considered sequence lengths up to $L = 100$. As we will discuss later, for our framework, a reasonable comparison with the reference NN size estimations by the *NNSE* and a determination of the functional relation for the correction parameter α is only possible when including the extrapolation.

For the robustness, we do not consider an extrapolation for the comparison. We simply assume that a NC robustness estimate $r_{\mathrm{NC,est}}$ by our framework is roughly an estimate of the NN robustness $r_{\mathrm{NN,est}}$, too:

$$r_{\mathrm{NN,est}} \approx r_{\mathrm{NC,est}} \tag{8}$$

D. Size estimation optimisation

Our NC size estimation (see eq. (3)) contains a correction parameter α, which we observed to be dependent on the sequence length for $L = 12$ to $L = 16$. In order to apply our approach to longer sequence lengths, we would need an appropriate value of α. To achieve this, we first search for the optimal value of α for every sequence length in our considered fRNAdb dataset. Then, we derive a functional relationship for α that we use to study our size estimations in more detail.

We search for the optimal value of α for a particular sequence length by minimising the order of magnitude root-mean-square deviation (RMSD$_s$) between our estimated and extrapolated NN sizes and the estimated NN sizes by the *NNSE* for all considered sequences from the fRNAdb of that sequence length. This is:

$$\mathrm{RMSD}_s = \sqrt{\frac{1}{N_L}\sum_{i=1}^{N_L}\left(\log_{10}\left(s_{\mathrm{NN,est},i}\right) - \log_{10}\left(s_{\mathrm{NN,NNSE},i}\right)\right)^2} \tag{9}$$

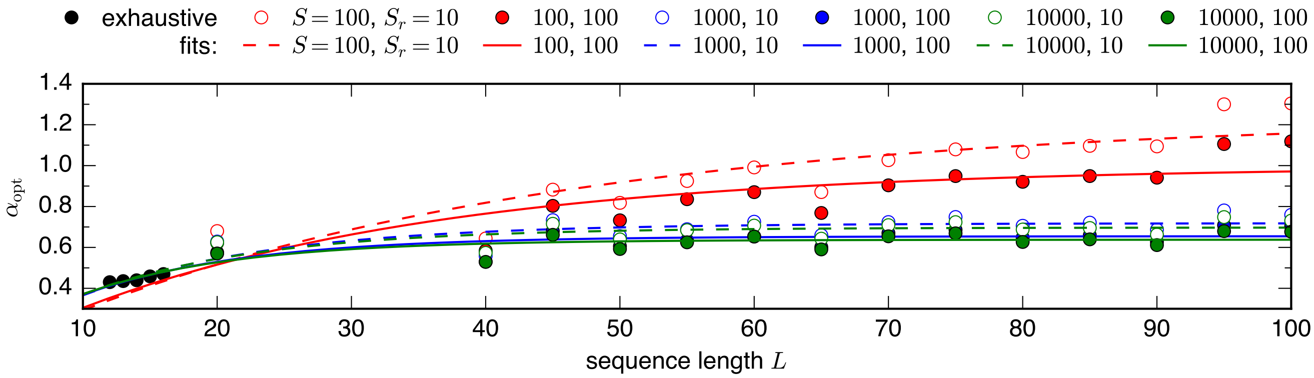

FIG. 4. Optimal values of the correction parameter α in the NC size estimation formula (see eq. (3)) that leads to minimum order of magnitude root-mean-square deviations (RMSD$_s$) (see eq. (9)) between the extrapolated NN size estimations by our framework and the reference estimations by the *NNSE* for all considered sequences from the fRNAdb for a given sequence length L for which the *NNSE* converges. The coloured dots indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. The black dots indicate the results for sequence lengths $L = 12$ to $L = 16$ from the exhaustive analysis in the Supplementary Information I. After an initial increase, the optimal α tends to saturate for longer sequence lengths. The lines indicate fits of the form $A\,(1 - \exp(-B \cdot L))$, for which the results can be found in Table I.

where the sum runs over all considered sequences from the fRNAdb of sequence length L for which the *NNSE* converges (total number: N_L). $s_{\mathrm{NN,NNSE},i}$ is the reference NN size estimate by the *NNSE* and $s_{\mathrm{NN,est},i}$ the extrapolated NN size estimate by our framework for sequence i, respectively.

For our framework, we estimate the NC size with eq. (3) (α dependent) and use the NC–NN extrapolation given by eq. (7). In Figure 4, we plot the obtained optimal values for α as a function of the sequence length and various sample and random subsample size combinations considered for our framework.

For fixed sequence length, the optimal α decreases and saturates with increasing sample size. While there is a significant difference between $S = 100$ and $S = 1000$ in particular for longer sequence lengths, there are only small differences between $S = 1000$ and $S = 10000$ for all sequence lengths, implying that our estimations do not change significantly above $S = 1000$. This is similar to the saturation in our NC size estimations with increasing sample size that we have seen before for $L = 15$. Similarly, increasing the random subsample size from $S_r = 10$ to $S_r = 100$ decreases the optimal α slightly.

We also add the optimal α values that we obtained for sequence lengths $L = 12$ to $L = 16$ from the exhaustive analysis in the Supplementary Information I. For the short sequence lengths, the optimal α tends to increase with sequence length, while for the longer ones, it tends to saturate. We introduced α in the NC size estimation formula to account for correlations between sequence sites. The findings suggest that these correlations only exist across a limited sequence range and do not increase further for sequences longer than this range. This is in agreement with findings by *García-Martín et al.* [27] that the average versatilities of paired and unpaired sites in RNA sequences do not change significantly for longer sequence lengths and show an asymptotic behaviour.

One might argue that the observed saturation of the optimal α values with increasing sequence length as well as the whole optimisation is affected and biased by our considered extrapolation from NC to NN size estimations, since both the correction parameter α as well as the extrapolation increase our NN size estimates. For this reason, we also tested the optimisation without including the extrapolation. The detailed results can be found in the Supplementary Information VI. For some of the sequence lengths and small sample or small random subsample sizes, no reasonable optima are reached, as our non-extrapolated size estimations do not increase with increasing α above a certain value of α and are too small compared to the reference NN size estimations by the *NNSE*. This is likely because the maximum factors for individual sites in our NC size estimation formula (see eq. (3)) are reached. In addition, if optima are found, the optimal α values are quite large and hard to reconcile with those found for sequence lengths $L = 12$ to $L = 16$. In all cases, the minimum RMSD$_s$ values are larger than the ones obtained when including the extrapolation. Due to the identified issues, we will stick to the extrapolation from NC to NN size estimates in the following.

As discussed, for sample sizes above about $S = 1000$, the optimal α seems to only depend on sequence length. A very simple functional relation that describes the initial increase and subsequent saturation with sequence length is the following:

$$\alpha(L) = A\,(1 - \exp(-B \cdot L)) \tag{10}$$

where A and B are parameters. This functional relation is a simplified version of the fit function considered in [27] to describe the sequence length dependence of the average

S	S_r	A fit	B fit
100	10	1.24 ± 0.08	0.027 ± 0.004
100	100	1.00 ± 0.05	0.036 ± 0.005
1000	10	0.72 ± 0.01	0.071 ± 0.006
1000	100	0.65 ± 0.01	0.083 ± 0.007
10000	10	0.70 ± 0.01	0.076 ± 0.006
10000	100	0.64 ± 0.01	0.088 ± 0.007

TABLE I. Fit parameter results of fits of the functional form $A\left(1-\exp\left(-B \cdot L\right)\right)$ to the data in Figure 4 for different sample sizes S and random subsample sizes S_r considered for our framework. The function describes the sequence length L dependence of the correction parameter α in the NC size estimation formula (see eq. (3)).

versatilities. A is the saturation value of α in the limit $L \to \infty$. In Figure 4, we fitted the functional relation to the data points. The results of the fit parameters and their errors are summarised in Table I.

From now on, we will work with the following values of A and B:

$$A = 0.68,\ B = 0.079 \tag{11}$$

which we obtain by averaging over the fit results for $S = 1000$ and $S = 10000$, i.e. the sample sizes for which we observe a saturation.

In principle, more complex functional relations could be fitted to the data as well. However, the used one is the simplest one that we have found. In addition, for sequence length $L = 12$ to $L = 16$, the α values obtained by eq. (10) and the above parameters for A and B are in good agreement with the optima found in the Supplementary Information I. Furthermore, as we will see later, for the longer fRNAdb sequences, α values obtained from the functional relation will lead to estimation results close to the optima, too.

E. Results

We now discuss the results for the sequences from the fRNAdb dataset using the derived analytical expression for the correction parameter α. In Figure 5, results for the NC size and robustness estimations for all sequences of length $L = 50$ and $L = 100$ in the dataset are shown using a sample size of $S = 1000$ and a random subsample size of $S_r = 10$. These results clearly reproduce the linear scaling of the NC robustness with the logarithm of the NC size. This is one of the properties previously found for GP maps on the phenotype and NC level [4, 14, 18, 22], and which here to some extent directly follows from the form of the estimation formulas that we consider. The results also highlight the remarkable range of estimated NC sizes, which span about 21 orders of magnitude for sequence length $L = 100$.

Next, we compare the extrapolated NN estimations for both sequence lengths with the reference estimations by the *NNSE*. The results are shown in Figure 6.

For the NN sizes, there is a good agreement between estimations by both frameworks. For $L = 50$, extrapolated estimations by our framework tend to be slightly larger, and for $L = 100$, slightly smaller on average than those by the *NNSE*. This likely arises from the fact that the α value determined from the functional relation is slightly larger or smaller than the optimal one found before, respectively (see Figure S7(B.i) compared to (A.i) in the Supplementary Information VII, where the average differences are shown for using the α values from the functional relation and the optimal ones, respectively). The results also indicate a larger spread of the estimations around the ideal case of full agreement for $L = 100$ compared to $L = 50$.

For the NN robustness, we again find a good agreement between estimations by both frameworks. For both sequence lengths, our estimations tend to be marginally larger on average than those by the *NNSE*.

In order to further quantify the agreement, we again study the root-mean-square deviation between our estimations and those by the *NNSE*.

For the NN size estimations, in Figure 7, the order of magnitude root-mean-square deviation (RMSD_s) (see eq. (9)) is shown as a function of the sequence length and various sample and random subsample size combinations considered for our framework. For Figure 7(A), the optimal α values from the optimisation (see Figure 4) are used for our NC size estimation, i.e. the α values that minimise the RMSD_s. In all cases, the RMSD_s tends to increase with sequence length. For fixed longer sequence lengths, it tends to decrease with increasing sample size from $S = 100$ to $S = 1000$, while there is only a marginal difference between $S = 1000$ and $S = 10000$ similar to the marginal differences of the respective optimal α values. Similarly, for the longer sequence lengths, the RMSD_s tends to slightly decrease with increasing random subsample size. For the short sequence lengths, there is nearly no difference in the RMSD_s between the sample and random subsample size combinations.

In Figure 7(B), the results are shown for using the α values from the derived functional relation for our NC size estimation, i.e. α values that only depend on the sequence length but not on the sample and random subsample size. In agreement with the fact that the parameters for the functional relation were determined by averaging the fit results for $S = 1000$ and $S = 10000$, the results for the RMSD_s are very similar to those in Figure 7(A) for both sample sizes. This also explains the stronger increase in the RMSD_s for $S = 100$ compared to before, since for this small sample size the functional relation for α differs from the respective optimal α values.

In order to check if the increase in the RMSD_s with sequence length is in line with the increase of the NC and so NN sizes with sequence length, we considered what we refer to as the relative RMSD_s. In this case,

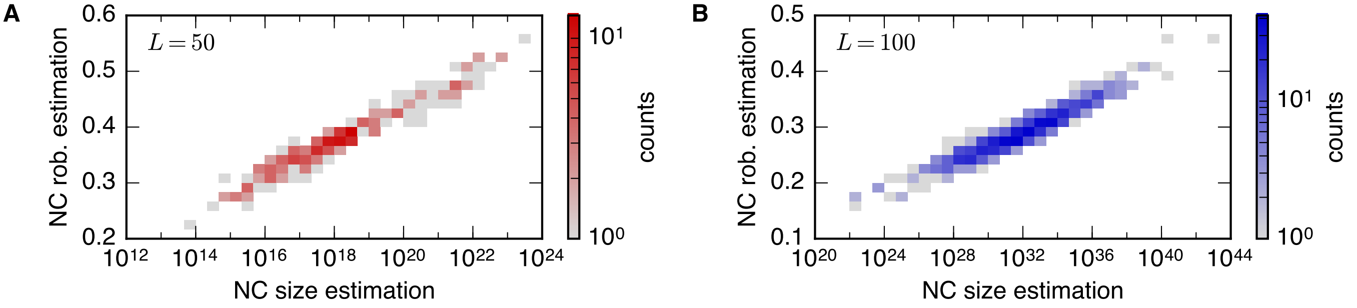

FIG. 5. NC robustness versus NC size estimations by our framework for all considered sequences from the fRNAdb of length (A) $L = 50$ and (B) $L = 100$. In both cases, accelerated site scanning sampling with a sample size of $S = 1000$ and random subsample size of $S_r = 10$ is used. For the NC size estimations, we use eq. (3) with α determined from eq. (10) and parameters from eq. (11). The NC robustness estimations clearly scale with the logarithm of the NC size estimations, which matches observations in the literature [4, 14, 18, 22].

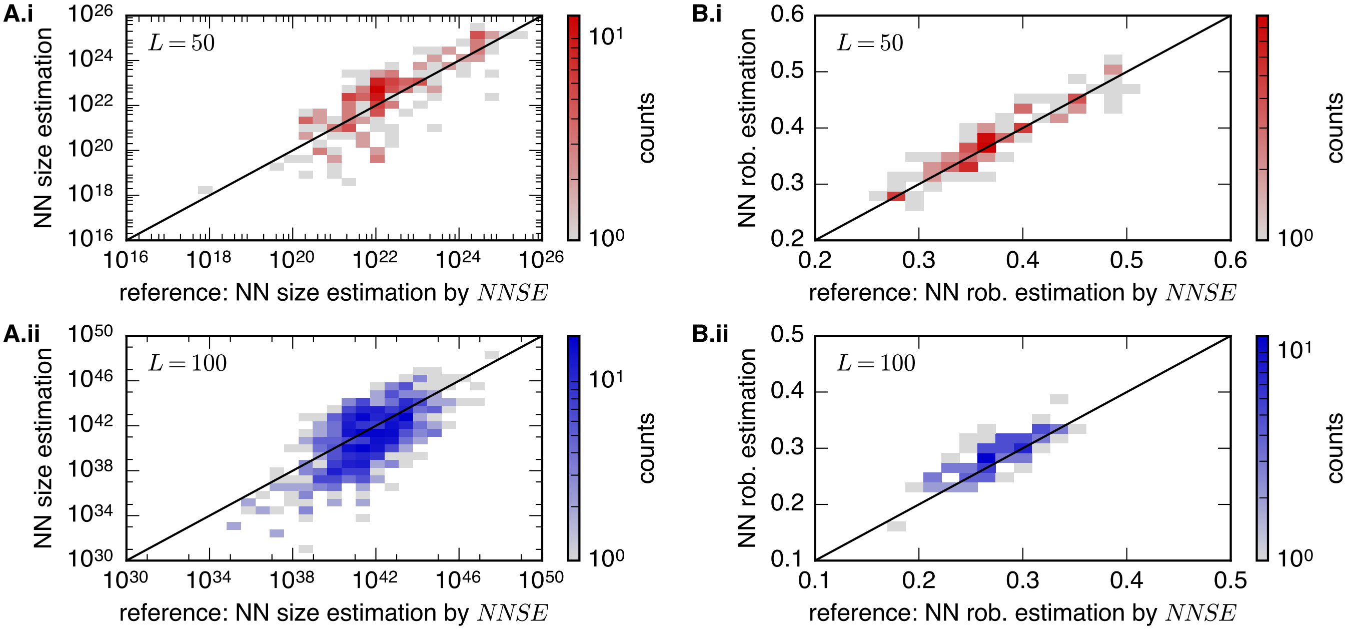

FIG. 6. Extrapolated (A) NN size and (B) NN robustness estimations by our framework versus the reference estimations by the *NNSE* for all considered sequences from the fRNAdb of length (i) $L = 50$ and (ii) $L = 100$. For the NN size estimations, all sequences are considered for which the *NNSE* converges, while for the NN robustness estimations, only 100 randomly selected sequences for each sequence length are considered. For our framework, in all cases, accelerated site scanning sampling with a sample size of $S = 1000$ and random subsample size of $S_r = 10$ is used. For the NC size estimations, before extrapolation, we use eq. (3) with α values from the derived functional relation (see eqs. (10) and (11)). The diagonal lines indicate the ideal case of zero difference between both estimation frameworks. Overall, there is a good agreement between estimations by both frameworks. For the NN size estimations, the spread around the diagonal increases for $L = 100$ compared to $L = 50$. The NN robustness estimations by our framework tend to be marginally larger than those by the *NNSE* for both sequence lengths.

we examine the deviation between our extrapolated NN size estimates from the reference NN size estimates by the *NNSE* relative to the latter ones. The results can be found in the Supplementary Information VII. We find that relative RMSD$_s$ is roughly constant and does not increase with sequence length.

A further alternative way to quantify the agreement is by considering the average difference and the standard deviation of the differences between NN size estimations by both frameworks. This is shown in the Supplementary Information VII similar to the cases in Figure 7. For using the α values obtained from the derived functional relation, the results highlight that for $S = 1000$ and $S = 10000$ the average difference fluctuates around zero, and that there is no significant trend across the considered range of sequence lengths. For a smaller random subsample size, the average difference tends to be slightly smaller, which means that our extrapolated NN size estimates are slightly

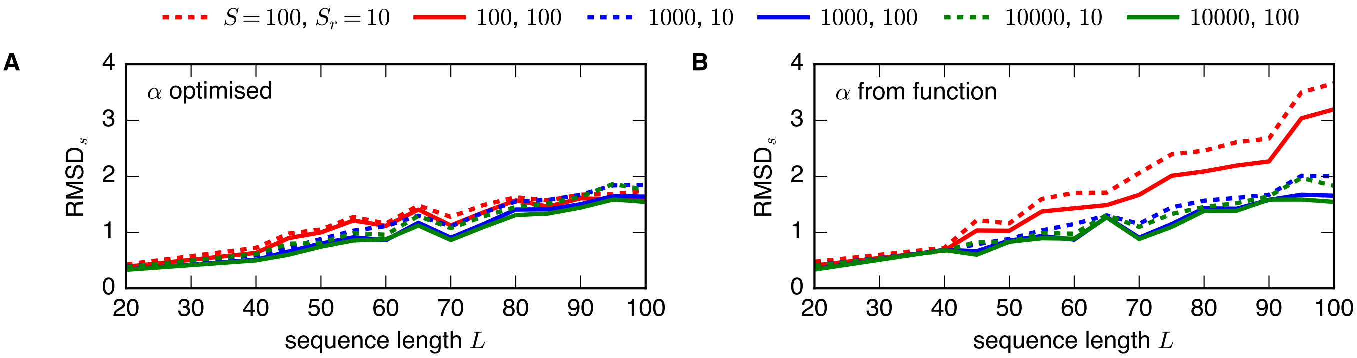

FIG. 7. Order of magnitude root-mean-square deviation (RMSD$_s$) (see eq. (9)) between the extrapolated NN size estimations by our framework and the reference estimations by the *NNSE* for all considered sequences from the fRNAdb for a given sequence length L for which the *NNSE* converges. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. For the NC size estimations, before extrapolation, we use eq. (3) and in (A) the α values from the optimisation (see Figure 4), i.e. those that minimise the RMSD$_s$, and in (B) the α values from the derived functional relation (see eqs. (10) and (11)). For the larger sample sizes $S = 1000$ and $S = 10000$, which have been used to derive the functional relation for α, the results for the RMSD$_s$ are very similar in (A) and (B). In all cases, the RMSD$_s$ increases with sequence length. Up to a certain amount, it decreases with increasing sample or random subsample size for the longer sequence lengths.

smaller on average. For $S = 100$, our extrapolated NN size estimates tend to be increasingly smaller on average than those by the *NNSE* with increasing sequence length, highlighting again that this sample size is too small. The results also show that in all cases the standard deviation of the differences significantly increases with sequence length, highlighting that it is mainly the spread of the differences that causes the increase in the RMSD$_s$ with sequence length. This likely originates from the fact that our NC size estimation formula (see eq. (3)) as well as the NN size estimation formula considered by the *NNSE* [22] consists of a product for which the number of factors increases with sequence length, leading to increasing uncertainties in the estimations themselves. This is in agreement with the roughly constant relative RMSD$_s$ across the sequence lengths as discussed before.

For the NN robustness estimations, we also consider the root-mean-square deviation RMSD$_r$, this is:

$$\mathrm{RMSD}_r = \sqrt{\frac{1}{N_{L,r}} \sum_{i=1}^{N_{L,r}} \left(r_{\mathrm{NN,est},i} - r_{\mathrm{NN,NNSE},i}\right)^2} \qquad (12)$$

where the sum runs over the $N_{L,r} = 100$ randomly selected sequences for each sequence length. Figure 8 shows the results. As an alternative, in Supplementary Information VII, the average difference and the standard deviation of the differences between estimations by both frameworks is shown.

The root-mean-square deviation RMSD$_r$ is roughly independent of the sequence length, and as seen before for the NN size estimations, decreases with increasing sample size up to a certain amount. An increase in the random subsample size also decreases the RMSD$_r$ to some extent. As can be seen in Figure 6 as well as Figure S8, our NN robustness estimates are marginally larger on average than those by the *NNSE*. There are two potential reasons. First, we actually estimate NC and not NN robustness and only assume that both are equal. In fact, the robustness of a NC might be larger than of its NN due to the fragmentation of the NN into NCs, leading to a marginal overestimation of our NN robustness estimates. Second, as addressed before, our site scanning sampling approach is likely biased towards network nodes with high degree and therefore with high robustness, potentially also leading to marginally overestimated NC and NN robustness.

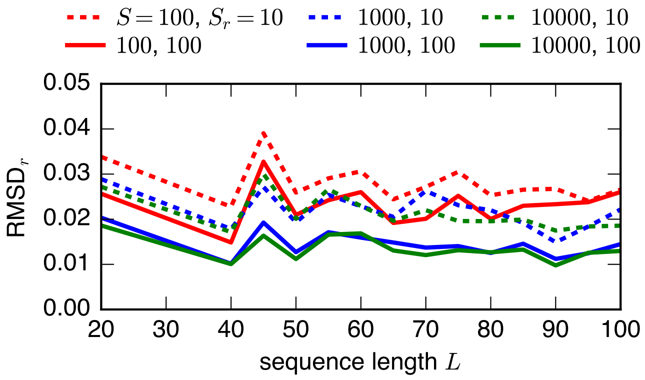

FIG. 8. Root-mean-square deviation (RMSD$_r$) (see eq. (12)) between the extrapolated NN robustness estimations by our framework and the reference estimations by the *NNSE* for 100 randomly selected sequences from the fRNAdb for a given sequence length L. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. The RMSD$_r$ is roughly independent of the sequence length. In all cases, up to a certain amount, it decreases with increasing sample or random subsample size.

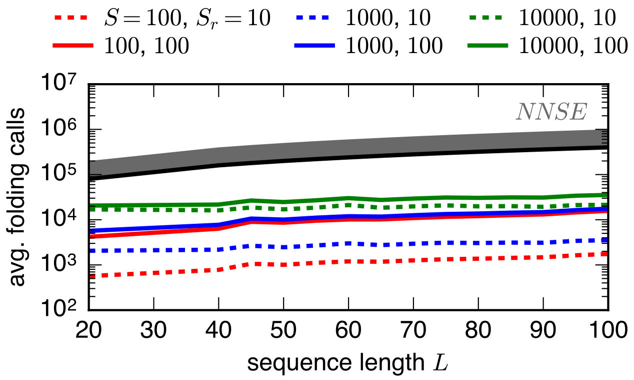

FIG. 9. Average number of calls of the folding function for one estimation run by our framework. The results are averaged over all considered sequences from the fRNAdb for a given sequence length L. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. As reference, the expected number of calls of the folding routine by the *NNSE* is shown: The black line corresponds to pure NN size estimations (see eq. (15)) and the shaded grey area to NN size and robustness estimations (see eq. (16), $\beta_r \in [0, 1]$). For our framework, the number of calls of the folding function can be several orders of magnitude smaller compared to the *NNSE*.

In Figures 7 and 8, as well as for the data points in Figure 4, there is a lack of monotonicity in the sequence length. This might arise due to certain biases in the sets of fRNAdb sequences considered for each sequence length. In addition, for the NN size estimates, it might also arise from the fact that different numbers of fRNAdb sequences are considered for each sequence length (see Table SII in the Supplementary Information V).

F. Computational costs

In a final step, we compare the computational costs of our approach to that of the benchmark. *Jörg et al.* [22] argue that for their *NNSE* nearly all of the CPU time is consumed by the RNA folding routine, the computational cost of which grows as L^3 with the sequence length L. Similarly, for our framework, most of the computation time is consumed by calling the folding function *RNA.fold*. Therefore, to compare computational costs, it is sufficient to compare the number of calls of the folding function during one estimation run. For experiments, similarly, the phenotype determination will likely be one of the most demanding steps. Figure 9 shows the average number of folding calls by our framework for one estimation run and as reference the expected number by the *NNSE*.

For our framework, NC size and robustness estimations require the same computational steps. Therefore, they can be computed together and are associated with the same computational costs. Furthermore, we can explicitly count the number of calls of *RNA.fold* while running our code. Considering a sample size of S and a random subsample size of S_r, the number of calls C for one estimation run for one sequence of length L can be described by:

$$C = \beta_S \cdot S + \beta_r \cdot S_r \cdot 3L \tag{13}$$

where $\beta_S \geq 1$ and $\beta_r \leq 1$ are parameters depending on the start genotype, NC, phenotype and sampling process.

The second term in eq. (13) corresponds to the measurement of the one-point mutational neighbourhoods of the S_r genotypes in the random subsample. In principle, this would require $S_r \cdot 3L$ times calling *RNA.fold*. However, since we restrict ourselves to RNA secondary structure compatible base pairs, fewer calls are required, given by the factor $\beta_r \leq 1$. It can be described by:

$$\beta_r = \frac{3(L-2n) + \gamma \cdot 2n}{3L} = 1 - \frac{2\,(3-\gamma)}{3}\frac{n}{L} \tag{14}$$

where n is the number of base pairs in the corresponding phenotype. For the $L - 2n$ unpaired sites, all 3 possible one-point mutations are checked for neutrality by calling *RNA.fold*. For the $2n$ paired sites, only the one-point mutations are checked for neutrality that lead to a compatible base pair, we describe this number by γ. Assuming that CG/GC base pairs are twice as frequent as AU/UA base pairs, and three times more frequent than GU/UG base pairs, the average number of one-point mutations for a paired site that still lead to a compatible base pair is given by $\frac{13}{22}$ and so $\gamma \approx \frac{13}{22} \approx 0.59$ (for more details see the Supplementary Information VIII). Comparing with the true number of calls of *RNA.fold* across our considered individual estimations shows that this is a valid approximation. For example, if half of the sites are paired, it is $\beta_r \approx 0.60$, highlighting the significant reduction in computational costs by restricting to RNA secondary structure compatible base pairs.

The first term in eq. (13) corresponds to the generation of a sample of size S by our accelerated version of site scanning sampling. Since not all one-point mutation steps considered in the sampling process are likely to be neutral, S or more calls of *RNA.fold* are required to generate a sample of size S. This is given by the factor $\beta_S \geq 1$.

For β_S, an analytical description has not been found so far, though some analytical considerations are possible. For an unpaired site, the number of calls to find a neutral one-point mutation approximately scales with $\frac{3}{\overline{x_j}}$, where $\overline{x_j}$ is the average number of neutral mutations for that site. For a paired site, the number of calls approximately scales with $\frac{\gamma}{\overline{x_j}}$, where γ is the number of compatible one-point mutations for a paired site as discussed before. However, β_S cannot be simply approximated by the sequence average of these values as our accelerated version of site scanning sampling is more advanced in the way that we do not test every site 'endlessly' until a neutral mutation is found but instead proceed to the next site if all mutations are tested and are not successful, making an analytical description harder. The sequence averaged numbers of calls of *RNA.fold* in Figure 9 suggest an average value of β_S ranging from about 1.5 to about 2, though it strongly depends on the concrete start genotype, NC, phenotype and sampling process considered.

Only the second term in eq. (13) scales directly with the sequence length. Since the scaling depends on the random subsample size, the costs can be significantly reduced by using a small random subsample – as addressed before – especially for longer sequences.

The *NNSE* software does not allow an explicit counting of the number of calls of the folding routine. Therefore, we deduce it with the best of our knowledge from the information given in [22] and the *C* code.

For a pure NN size estimation, using default parameters, we describe the number of calls $C_{\text{NNSE},s}$ of the folding routine for one estimation run for one structure of length L by:

$$C_{\text{NNSE},s} = (2000 + 2000) \cdot L \tag{15}$$

The first term corresponds to 2000 thermalisation and the second to 2000 measurement steps considered for each of the L shells (volume ratios) in the genotype space used to estimate the NN size.

A NN size together with a robustness estimation is associated with additional computational costs. In this case, the number of calls $C_{\text{NNSE},s+r}$ of the folding routine is given by:

$$C_{\text{NNSE},s+r} = (2000 + 2000) \cdot L + \beta_r \cdot 2000 \cdot 3L \tag{16}$$

The added term represents the measurement of the one-point mutational neighbourhood of the genotype in the innermost shell at every measurement step. The *NNSE* also restricts to RNA secondary structure compatible base pairs when measuring the neighbourhoods similarly as we do. Therefore, a similar reduction factor $\beta_r \leq 1$ is included. Its exact value cannot be determined as the *NNSE* allows no explicit counting, but it will likely be similar to our approximation considered before. In Figure 9, we plot eq. (16) for the full possible range $\beta_r \in [0, 1]$.

In both cases, the number of calls scales linearly with L. Comparison with eq. (13) for individual estimations as well as the average number of calls in Figure 9 highlights the significantly higher computational costs – up to several orders of magnitude – associated with the *NNSE* compared to our framework for the considered sample and random subsample size combinations. It should be noted that the *NNSE* also requires multiple calls of the inverse folding routine in the initialisation procedure in order to find sequences folding to the input structure. This is associated with additional computational costs, which we do not address here.

V. DISCUSSION AND CONCLUSION

In this article, we have introduced a framework for estimating large-scale properties in the genotype-phenotype (GP) map of RNA secondary structure – in detail the size and robustness of neutral components (NCs) – by only using small samples of genotypes. Our framework is novel and advantageous compared to existing estimation frameworks in several ways. Compared to the *NNSE* software of *Jörg et al.* [22], our framework allows estimates of NC instead of NN characteristics. The former are the more essential neutral units for evolving populations due to their full connectivity through one-point mutations. In addition, our framework can be computationally up to several orders of magnitude less expensive. In comparison to the NC size estimation method introduced by *García-Martín et al.* [27], our framework is – by default – designed to make estimates from small samples of genotypes. Furthermore, our framework allows NC size and robustness estimates at the same time, and no different (additional) measurements are required. In addition to the estimation framework, we also propose a novel sampling method to efficiently and broadly sample NCs: site scanning sampling, a periodic scanning of sequence sites for neutral mutations, which outperforms a simple random walk sampling in terms of the accuracy of estimates.

The considered method for estimating the NC robustness is simple and likely transferable to any other sequence-to-structure GP map without changes. It produces remarkably accurate estimates for small sample (and random subsample) sizes. With site scanning sampling, our NC robustness estimations tend to marginally overestimate on average the true NC robustness values for short sequence lengths or the respective NN robustness estimations by the *NNSE* for the longer, naturally occurring functional non-coding sequence lengths. However, this effect is marginal, and the robustness of a NC likely may be larger than of the respective NN as addressed in the article.

Our NC size estimation method cannot be directly transferred to other sequence-to-structure GP maps, though the basic ideas behind the method are likely to be universal. The NC size estimation formula includes a correction factor that accounts for correlations between sequence sites 'suppressing' the average number of neutral mutations per site. We derive a sequence length dependent functional relation of the correction factor by optimising the NC size estimations with respect to the true values for short sequence lengths and the extrapolated NN size estimations with respect to those by the *NNSE* for the longer sequence length. Here, specifically for RNA, we find that the optimal correction factor saturates with increasing sequence length, which suggests that the correlations between sequence sites are limited to a certain range. For other sequence-to-structure GP maps, the functional relation for α would need to be modified, but the basic structure of the estimation formula can likely be used in the same way. Using the derived functional relation, our extrapolated NN size estimations are in good agreement with those by the *NNSE* for the longer sequences up to length $L = 100$.

A few caveats remain: Firstly, the used extrapolation from NC to NN sizes is based on findings by *Schaper et al.* [23] on NN fragmentation in the case of RNA secondary structure. We use it across all considered sequence lengths. However, as addressed in the article, it is a matter of debate whether this extrapolation still holds for longer sequence lengths. The fact that our approach yields more accurate NN size estimates when using the extrapolation indicates that it does, but more research is needed. Secondly, we use as reference the *NNSE*, which has not yet been benchmarked itself for longer RNA sequence lengths for which the true values are not known. Using the default settings for all sequence lengths might lead to a systematic error in the estimations. Thirdly, all of our analysis as well as the *NNSE* is based on using the *ViennaRNA* package as prediction software and defining the phenotype by the minimum free energy secondary structure. This algorithm is not 100% accurate and may deviate from the secondary structure of real biological RNA molecules, especially for longer sequence lengths.

Compared to the *NNSE*, the computational costs – in terms of the number of calls of the folding function – can be up to several orders of magnitude smaller for our framework. A major contributing factor is the fact that small random subsamples of larger site scanning samples are usually sufficient for accurate estimates. This also opens a potential experimental realisability. If a small and diverse sample of genotypes from the same NC is known (either through sampling and random subsampling or through a set of previous experiments), only the one-point mutational neighbourhoods of these genotypes need to be measured for neutrality to calculate the estimates. For long sequence lengths, this still can be a quite large number of genotypes to be tested. However, our framework only requires the knowledge whether or not the phenotype changes by a one-point mutation, and not which concrete alternative phenotype appears for a non-neutral mutation. In addition, for RNA secondary structure, some one-point mutations can be a priori called non-neutral if they violate compatible base pair nucleotide combinations, and neutrality might be measured by investigating changes in biological function related to RNA secondary structure, which likely could further reduce the experimental workload.

The introduced framework enables new applications. It allows the quantitative and qualitative comparison of the NCs of functional non-coding RNA sequences, which may yield new insights into the evolution of RNA. In particular, it can allow estimates for long RNA sequences, for which an exhaustive analysis is far from being feasible, or for which using the *NNSE* exceeds a reasonable computation time. In future studies, the framework could also be applied and adapted to other sequence-to-structure GP maps, like that of the HP model, the Polyomino model [13, 14], or to more complex maps such as protein secondary structure, thereby fostering the general understanding of GP maps for long, non-exhaustively analysable sequence lengths.

An unsolved problem that could be addressed in the future is the estimation of NC evolvability from samples of genotypes. Evolvability differs from size and robustness in the way that it not only depends on the NC itself but also its explicit mutational neighbourhood, i.e. all distinct alternative phenotypes surrounding the NC. Therefore, estimating the evolvability of a NC just from a sample of genotypes is a more challenging problem.

Author contributions

MW and SEA designed the analysis, MW carried out the analysis, MW and SEA wrote the manuscript.

Competing interests

We declare we have no competing interests.

Data accessibility

The code to generate the data shown in this article, the data itself as well as a standalone program to make NC estimates starting from a given input RNA sequence can be accessed at: `https://github.com/mw636/NC-sample-est.git`.

Funding

MW was supported by the EPSRC and the Gatsby Charitable Foundation. SEA was supported by the Gatsby Charitable Foundation.

[1] P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker, Proceedings of the Royal Society of London. Series B: Biological Sciences **255**, 279 (1994).
[2] W. Fontana, BioEssays **24**, 1164 (2002).
[3] A. Wagner, Proceedings of the Royal Society B: Biological Sciences **275**, 91 (2008).
[4] J. Aguirre, J. M. Buldú, M. Stich, and S. C. Manrubia, PLOS ONE **6**, 1 (2011).
[5] E. Ferrada and A. Wagner, Biophysical Journal **102**, 1916 (2012).
[6] J. L. Payne and A. Wagner, Science **343**, 875 (2014).
[7] S. Ciliberti, O. C. Martin, and A. Wagner, Proceedings of the National Academy of Sciences **104**, 13591 (2007).
[8] J. L. Payne and A. Wagner, PLOS Computational Biology **9**, 1 (2013).
[9] J. F. Matias Rodrigues and A. Wagner, PLOS Computational Biology **5**, 1 (2009).

[10] K. F. Lau and K. A. Dill, Macromolecules **22**, 3986 (1989).

[11] D. J. Lipman, W. J. Wilbur, and J. M. Smith, Proceedings of the Royal Society of London. Series B: Biological Sciences **245**, 7 (1991).

[12] H. Li, R. Helling, C. Tang, and N. Wingreen, Science **273**, 666 (1996).

[13] I. G. Johnston, S. E. Ahnert, J. P. K. Doye, and A. A. Louis, Phys. Rev. E **83**, 066105 (2011).

[14] S. F. Greenbury, I. G. Johnston, A. A. Louis, and S. E. Ahnert, Journal of The Royal Society Interface **11**, 20140249 (2014).

[15] M. A. Fortuna, L. Zaman, C. Ofria, and A. Wagner, PLOS Computational Biology **13**, 1 (2017).

[16] J. Maynard Smith, Nature **225**, 563 (1970).

[17] M. Eigen, R. Winkler-Oswatitsch, and A. Dress, Proceedings of the National Academy of Sciences **85**, 5913 (1988).

[18] S. F. Greenbury, S. Schaper, S. E. Ahnert, and A. A. Louis, PLOS Computational Biology **12**, 1 (2016).

[19] S. E. Ahnert, Journal of The Royal Society Interface **14**, 20170275 (2017).

[20] S. Schaper and A. A. Louis, PLOS ONE **9**, 1 (2014).

[21] K. Dingle, S. Schaper, and A. A. Louis, Interface Focus **5**, 20150053 (2015).

[22] T. Jörg, O. C. Martin, and A. Wagner, BMC Bioinformatics **9**, 464 (2008).

[23] S. Schaper, I. G. Johnston, and A. A. Louis, Proceedings of the Royal Society B: Biological Sciences **279**, 1777 (2012).

[24] S. F. Greenbury and S. E. Ahnert, Journal of The Royal Society Interface **12**, 20150724 (2015).

[25] S. Manrubia and J. A. Cuesta, Journal of The Royal Society Interface **14**, 20160976 (2017).

[26] M. Weiß and S. E. Ahnert, Journal of The Royal Society Interface **15**, 20170618 (2018).

[27] J. A. García-Martín, P. Catalán, S. Manrubia, and J. A. Cuesta, EPL (Europhysics Letters) **123**, 28001 (2018).

[28] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster, Monatshefte für Chemie / Chemical Monthly **125**, 167 (1994).

[29] I. L. Hofacker, Nucleic Acids Research **31**, 3429 (2003).

[30] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker, Algorithms for Molecular Biology **6**, 26 (2011).

[31] T. Kin, K. Yamada, G. Terai, H. Okida, Y. Yoshinari, Y. Ono, A. Kojima, Y. Kimura, T. Komori, and K. Asai, Nucleic Acids Research **35**, D145 (2007).

[32] T. Mituyama, K. Yamada, E. Hattori, H. Okida, Y. Ono, G. Terai, A. Yoshizawa, T. Komori, and K. Asai, Nucleic Acids Research **37**, D89 (2008).

[33] J. A. Capitán, J. Aguirre, and S. Manrubia, Chaos, Solitons & Fractals **72**, 99 (2015).

[34] S. Manrubia and J. A. Cuesta, Journal of The Royal Society Interface **12**, 20141010 (2015).

[35] J. Aguirre, P. Catalán, J. A. Cuesta, and S. Manrubia, Open Biology **8**, 180069 (2018)

Supplementary Information

I. NC SIZE ESTIMATION VALIDATION

In section II A, we introduce a formula to estimate NC sizes (see eq. (3)) from a sample of genotypes from a NC of interest. In the following, we will validate this formula, particularly in regards of the included correction parameter α.

Here, in all steps, we do not consider a sample of genotypes from a NC but instead the full set of genotypes of the NC, the largest possible sample. I.e. we compute the average number of neutral mutations and its standard deviation for a site j as follows:

$$\overline{x_j} = \frac{1}{G_{\mathrm{NC}}} \sum_{i=1}^{G_{\mathrm{NC}}} x_{i,j} \tag{S1}$$

$$\sigma_j = \sqrt{\frac{1}{G_{\mathrm{NC}}} \sum_{i=1}^{G_{\mathrm{NC}}} \left(x_{i,j} - \overline{x_j}\right)^2} \tag{S2}$$

where both sums run over all genotypes of the NC of interest (total number: G_{NC}). Note the difference of the standard deviation to the sample standard deviation in eq. (2) for ‘incomplete’ samples.

As basis, we consider the NCs of the exhaustively analysable RNA secondary structure GP maps of sequence lengths $L = 12$, $L = 13$, $L = 14$, $L = 15$ and $L = 16$. In all cases, we ignore the undefined phenotype (unbound structure), leaving 431, 1236, 3311, 8792, and 23091 NCs, respectively.

A. Optimisation

In the NC size estimation formula, we include a correction factor α that we assume to be dependent on the sequence length L and that accounts for the amount of correlations between sites. In this step, we compute the optimal value of α for each sequence length. In detail, we are interested in minimising the order of magnitude root-mean-square deviation (RMSD$_s$) between estimated and true sizes of all NCs. This is:

$$\mathrm{RMSD}_s = \sqrt{\frac{1}{N_L} \sum_{i=1}^{N_L} \left(\log_{10}\left(s_{\mathrm{NC,est},i}\right) - \log_{10}\left(s_{\mathrm{NC,true},i}\right)\right)^2} \tag{S3}$$

where the sum runs over all NCs of the GP map of sequence length L (total number: N_L). $s_{\mathrm{NC,true},i}$ and $s_{\mathrm{NC,est},i}$ are the true and estimated NC size of NC i, respectively. The NC size estimations are computed using eq. (3) (α dependent) and the average numbers of neutral mutations per site and their standard deviations as introduced above. We find $\alpha_{\mathrm{opt}} \approx 0.43$ ($L = 12$), $\alpha_{\mathrm{opt}} \approx 0.44$ ($L = 13$), $\alpha_{\mathrm{opt}} \approx 0.44$ ($L = 14$), $\alpha_{\mathrm{opt}} \approx 0.46$ ($L = 15$) and $\alpha_{\mathrm{opt}} \approx 0.47$ ($L = 16$). In Figure S1(A), we plot for all considered sequence lengths how the RMSD$_s$ varies with the correction parameter including the optimum. In all cases, the optimised RMSD$_s$ is quite small with about 0.1 orders of magnitudes.

B. Validation

In Figure S1(B), we plot the NC size estimations versus the true NC sizes for all NCs for each sequence length, respectively, using the determined optimal values of α. The results highlight the very good agreement between the estimated and true NC sizes.

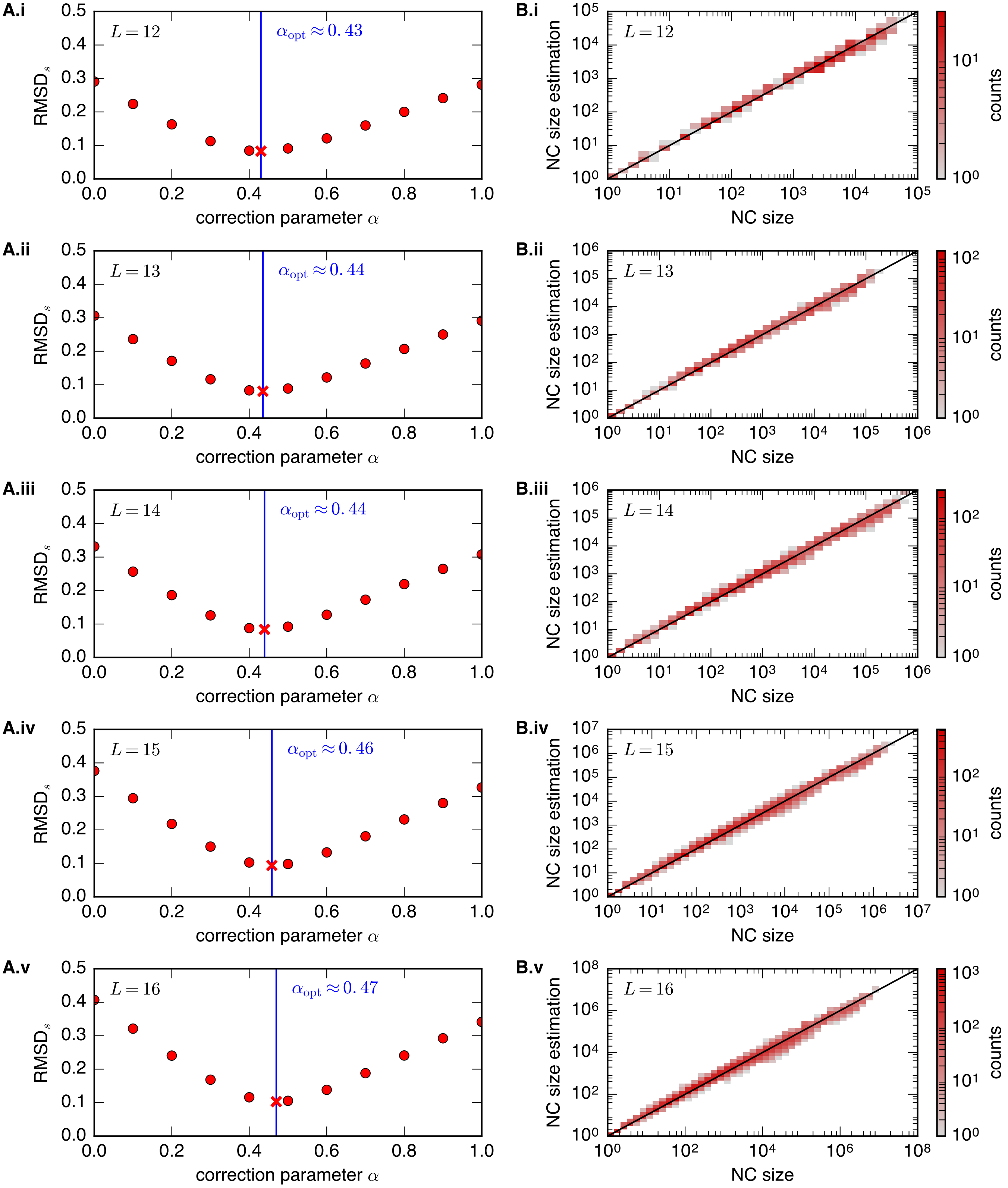

FIG. S1. (A) Order of magnitude root-mean-square deviation RMSD$_s$ (see eq. (S3)) between estimated and true NC sizes versus the correction parameter α considered in the NC size estimation formula (see eq. (3)). The cross marks the optimum such that RMSD$_s$ is minimised. (B) Estimated versus true NC sizes using the respective optimal value of α in the NC size estimation formula (see eq. (3)). In all cases, all NCs of the RNA secondary structure GP map of sequence length (i) $L = 12$, (ii) $L = 13$, (iii) $L = 14$, (iv) $L = 15$ and (v) $L = 16$ are considered, and the full set of NC genotypes instead of a sample is used for the NC size estimation.

II. METHODS

For the generation of a genotype sample from a NC of interest and the subsequent measurements of the one-point mutational neighbourhoods, we consider different methods and algorithms. In the following, they will be specified. Furthermore, we will specify how we perform our simulations. Throughout, we will use the following terminology:

$\mathcal{NC}$: NC of interest = set of all NC genotypes

$\mathcal{S}$: sample

S: (desired) sample size

$\mathcal{S}_r$: random subsample

S_r: (desired) random subsample size

p: phenotype

g: genotype

$g[j]$: letter at jth site (counting from 1) of genotype g

x_g: number of neutral mutations for genotype g

$x_g[j]$: number of neutral mutations for jth site (counting from 1) for genotype g

$\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$: alphabet (for RNA: $\mathcal{A} = \{$A,C,G,U$\}$)

a: letter

k: alphabet size

L: sequence length

A. Methods for exhaustively analysable NCs

1. Sampling

In section III, we apply our framework to the NCs of the exhaustively analysable $L = 15$ RNA secondary structure GP map. We consider three main sampling methods: uniformly random sampling, random walk (RW) sampling, and site scanning sampling. For the latter two, we additionally consider a random subsampling step. Throughout the following, it is assumed that the full GP map, i.e. the mapping of each genotype, and the full NC of interest are known.

```
Algorithm Uniformly random sampling
Input: $\mathcal{NC}$, $S$
Output: $\mathcal{S}$
    $\mathcal{S} \leftarrow S$ random elements $\in \mathcal{NC}$          ▷ $S$ random genotypes from NC
```

```
Algorithm RW sampling
Input: $\mathcal{NC}$, $S$, $\mathcal{A}$, $L$
Output: $\mathcal{S}$
    $g_1 \leftarrow$ random element $\in \mathcal{NC}$          ▷ random initial genotype from NC
    $\mathcal{S} \leftarrow [g_1]$          ▷ sample genotype list
    $g_{\text{ref}} \leftarrow g_1$          ▷ reference genotype that is mutated
    while len($\mathcal{S}$) $< S$ do
        $j \leftarrow$ random element $\in \{1, 2, \ldots, L\}$          ▷ random site
        $a \leftarrow$ random element $\in \mathcal{A} \setminus \{g_{\text{ref}}[j]\}$          ▷ random letter
        $g_{\text{mut}} \leftarrow g_{\text{ref}}$
        $g_{\text{mut}}[j] \leftarrow a$          ▷ one-point mutation
        if $g_{\text{mut}} \in \mathcal{NC}$ then          ▷ check if mutated genotype in NC
            $\mathcal{S}$.append($g_{\text{mut}}$)          ▷ append mutated genotype to sample
            $g_{\text{ref}} \leftarrow g_{\text{mut}}$          ▷ update reference genotype
        end if
    end while
```

Note: A generated sample $\mathcal{S}$ not necessarily consists of distinctive genotypes.

Algorithm Site scanning sampling

```
Input: $\mathcal{NC}$, $S$, $\mathcal{A}$, $L$
Output: $\mathcal{S}$
    $g_1 \leftarrow$ random element $\in \mathcal{NC}$                    ▷ random initial genotype from NC
    $\mathcal{S} \leftarrow [g_1]$                                          ▷ sample genotype list
    $g_{\text{ref}} \leftarrow g_1$                                         ▷ reference genotype that is mutated
    $j \leftarrow 1$                                                        ▷ site that is mutated
    while len($\mathcal{S}$) $< S$ do
        $\mathcal{A}^* \leftarrow \mathcal{A}$                              ▷ 'help alphabet'
        while $\mathcal{A}^* \neq \{g_{\text{ref}}[j]\}$ do
            $a \leftarrow$ random element $\in \mathcal{A}^* \setminus \{g_{\text{ref}}[j]\}$   ▷ random letter
            $g_{\text{mut}} \leftarrow g_{\text{ref}}$
            $g_{\text{mut}}[j] \leftarrow a$                                ▷ one-point mutation
            if $g_{\text{mut}} \in \mathcal{NC}$ then                       ▷ check if mutated genotype in NC
                $\mathcal{S}$.append($g_{\text{mut}}$)                      ▷ append mutated genotype to sample
                $g_{\text{ref}} \leftarrow g_{\text{mut}}$                  ▷ update reference genotype
                $j \leftarrow j + 1 \mod (L + 1)$                           ▷ update site
                break
            else
                $\mathcal{A}^* \leftarrow \mathcal{A}^* \setminus \{a\}$    ▷ update 'help alphabet'
            end if
        end while
        if $\mathcal{A}^* = \{g_{\text{ref}}[j]\}$ then
            $j \leftarrow j + 1 \mod (L + 1)$                               ▷ update site
        end if
    end while
```

Note: A generated sample $\mathcal{S}$ not necessarily consists of distinctive genotypes.

Algorithm Random subsampling

```
Input: $\mathcal{S}$, $S_r$
Output: $\mathcal{S}_r$
    $\mathcal{S}_r \leftarrow S_r$ random elements $\in \mathcal{S}$        ▷ $S_r$ random genotypes from sample
```

Note: Given the input sample $\mathcal{S}$, the random subsample $\mathcal{S}_r$ not necessarily consists of distinctive genotypes.

2. One-point mutational neighbourhood measurement

For the NC estimations, the sample average and sample standard deviation of the number of neutral mutations per site are required. Therefore, the number of neutral mutations per site for all sample (random subsample) genotypes is required. For the analysis in section III, we measure the one-point mutational neighbourhood and the number of neutral mutations per site for a genotype as outlined in the algorithm below. Once repeated for every sample (random subsample) genotype, we compute the sample average and sample standard deviation according to eqs. (1) and (2).

Algorithm One-point mutational neighbourhood measurement

```
Input: $g_{\text{ref}}$, $\mathcal{NC}$, $\mathcal{A}$, $L$
Output: $x_{g_{\text{ref}}}$
    $x_{g_{\text{ref}}} \leftarrow [0] \cdot L$                 ▷ set number of neutral mutations for every site to 0
    for $j \in \{1, 2, \ldots, L\}$ do
        for $a \in \mathcal{A} \setminus \{g_{\text{ref}}[j]\}$ do
            $g_{\text{mut}} \leftarrow g_{\text{ref}}$
            $g_{\text{mut}}[j] \leftarrow a$                        ▷ one-point mutation
            if $g_{\text{mut}} \in \mathcal{NC}$ then               ▷ check if mutated genotype in NC
                $x_{g_{\text{ref}}}[j] \leftarrow x_{g_{\text{ref}}}[j] + 1$   ▷ update number of neutral mutations for $j$th site
            end if
        end for
    end for
```

3. Simulations

For the results shown in Figures 2, 3 and S2, we perform our simulations as follows. For each of the considered NCs, and for each of the sampling methods, we generate 100 independent samples, respectively, for a range of sample sizes, respectively. For the different sample sizes, independent sampling procedures are considered, i.e. samples of larger size do not (necessarily) include the genotypes from samples of smaller size. For RW sampling and site scanning sampling, we additionally consider random subsampling of fixed size, i.e. whenever the sample size is larger or equal than the considered random subsample size, for each sample, we additionally randomly select the fixed number of random subsample genotypes uniformly from it.

B. Methods for functional non-coding RNA sequences

1. Sampling

In section IV, we apply our framework to longer, naturally occurring functional non-coding RNA sequences, for which an exhaustive analysis is not feasible and the full NCs are not known. In this case, we consider an RNA-specific accelerated version of site scanning sampling in order to reduce computational costs, i.e. calls of the function *RNA.fold* to find the phenotype of a genotype. The algorithm starts from a given genotype from the fRNA database [S1, S2].

Algorithm Accelerated site scanning sampling (for functional non-coding RNA sequences)

```
Input: $g_1$, $S$, $\mathcal{A}$, $L$                                    ▷ $g_1$: initial genotype from fRNAdb
Output: $\mathcal{S}$
    $\mathcal{S} \leftarrow [g_1]$                                        ▷ sample genotype list
    $p_{\text{ref}} \leftarrow RNA.fold(g_1)$                             ▷ find reference phenotype of initial genotype
    $g_{\text{ref}} \leftarrow g_1$                                       ▷ reference genotype that is mutated
    $j \leftarrow 1$                                                      ▷ site that is mutated
    while len($\mathcal{S}$) $< S$ do
        $\mathcal{A}^* \leftarrow \mathcal{A}$                            ▷ ‘help alphabet’
        while $\mathcal{A}^* \neq \{g_{\text{ref}}[j]\}$ do
            $a \leftarrow$ random element $\in \mathcal{A}^* \setminus \{g_{\text{ref}}[j]\}$   ▷ random letter
            $g_{\text{mut}} \leftarrow g_{\text{ref}}$
            $g_{\text{mut}}[j] \leftarrow a$                              ▷ one-point mutation
            if $j$ unpaired site in $p_{\text{ref}}$ or ($j$ paired site in $p_{\text{ref}}$ and mutated base pair $\in$ {CG,GC,AU,UA,GU,UG}) then
                $p_{\text{mut}} \leftarrow RNA.fold(g_{\text{mut}})$      ▷ find phenotype of mutated genotype
                if $p_{\text{mut}} = p_{\text{ref}}$ then
                    $\mathcal{S}$.append($g_{\text{mut}}$)                ▷ append mutated genotype to sample
                    $g_{\text{ref}} \leftarrow g_{\text{mut}}$            ▷ update reference genotype
                    $j \leftarrow j + 1 \mod (L + 1)$                     ▷ update site
                    break
                else
                    $\mathcal{A}^* \leftarrow \mathcal{A}^* \setminus \{a\}$   ▷ update ‘help alphabet’
                end if
            else
                $\mathcal{A}^* \leftarrow \mathcal{A}^* \setminus \{a\}$       ▷ update ‘help alphabet’
            end if
        end while
        if $\mathcal{A}^* = \{g_{\text{ref}}[j]\}$ then
            $j \leftarrow j + 1 \mod (L + 1)$                             ▷ update site
        end if
    end while
```

Note: A generated sample $\mathcal{S}$ not necessarily consists of distinctive genotypes.

On the generated sample $\mathcal{S}$, we apply random subsampling as specified before.

```
Algorithm Random subsampling
Input: $\mathcal{S}$, $S_r$
Output: $\mathcal{S}_r$
    $\mathcal{S}_r \leftarrow S_r$ random elements $\in \mathcal{S}$        ▷ $S_r$ random genotypes from sample
```

Again, given the input sample $\mathcal{S}$, the random subsample $\mathcal{S}_r$ not necessarily consists of distinctive genotypes.

2. One-point mutational neighbourhood measurement

For the analysis in section IV, we also consider an RNA-specific accelerated version of the one-point mutational neighbourhood measurement for all sample (random subsample) genotypes in order to reduce computational costs. We proceed as outlined in the algorithm below. This algorithm is similar to how the mutational robustness is measured by the software of *Jörg et al.* [S3]. Once repeated for every sample (random subsample) genotype, we compute the sample average and sample standard deviation according to eqs. (1) and (2).

```
Algorithm One-point mutational neighbourhood measurement (for functional non-coding RNA sequences)
Input: $g_{\mathrm{ref}}$, $p_{\mathrm{ref}}$, $\mathcal{A}$, $L$
Output: $x_{g_{\mathrm{ref}}}$
    $x_{g_{\mathrm{ref}}} \leftarrow [0] \cdot L$        ▷ set number of neutral mutations for every site to 0
    for $j \in \{1, 2, \ldots, L\}$ do
        for $a \in \mathcal{A} \setminus \{g_{\mathrm{ref}}[j]\}$ do
            $g_{\mathrm{mut}} \leftarrow g_{\mathrm{ref}}$
            $g_{\mathrm{mut}}[j] \leftarrow a$        ▷ one-point mutation
            if $j$ unpaired site in $p_{\mathrm{ref}}$ or ($j$ paired site in $p_{\mathrm{ref}}$ and mutated base pair $\in$ {CG,GC,AU,UA,GU,UG}) then
                $p_{\mathrm{mut}} \leftarrow RNA.fold(g_{\mathrm{mut}})$        ▷ find phenotype of mutated genotype
                if $p_{\mathrm{mut}} = p_{\mathrm{ref}}$ then
                    $x_{g_{\mathrm{ref}}}[j] \leftarrow x_{g_{\mathrm{ref}}}[j] + 1$        ▷ update number of neutral mutations for $j$th site
                end if
            end if
        end for
    end for
```

3. Simulations

For the results shown in Figures 4, 5, 6, 7, 8, 9, S3, S4, S5, S6, S7, S8 and Table I, we perform the simulations for our framework as follows. For each of the considered functional non-coding RNA sequences, we generate a sample using the accelerated version of site scanning sampling, for a range of sample and random subsample sizes, respectively. For each sample size, an independent sampling procedure is considered, i.e. a sample of larger size does not (necessarily) include the genotypes from a sample of smaller size. Then, for each fixed sample size, the random subsamples of different size are randomly selected uniformly from the same sample, respectively.

III. RNA $L = 15$ EXAMPLE NC DATASET

In section III B in Figure 2, we consider NC estimations for nine example NCs of the $L = 15$ RNA secondary structure GP map. Details of these NCs are listed in Table SI.

NC rank	phenotype	NC size	NC robustness
1	((....)).......	2329155	0.6112
125	..(((......))).	735321	0.4691
250	..((((....)))).	465730	0.4575
375	.((((.....)))).	327873	0.4164
500	((((.......))))	176336	0.3967
625	.(((...))).... .	129243	0.4887
750	.(((((...))))).	96418	0.3747
875	..(((((...)))))	57450	0.4081
1000	..(((((...)))))	38579	0.3844

TABLE SI. Details of the nine example NCs of the $L = 15$ RNA secondary structure GP map considered in Figure 2.

IV. ADDITIONAL FIGURES: APPLICATION TO EXHAUSTIVELY ANALYSABLE NCS

In section III B, we quantify the quality of our NC size and robustness estimations for the considered NCs of the $L = 15$ RNA secondary structure GP map by considering the root-mean-square deviation (RMSD) from the true values. An alternative way is to consider the average difference ($\overline{\mathrm{D}}$) and the standard deviation (σ_{D}) of the differences between our estimated and true values. This is:

$$\overline{\mathrm{D}_s} = \frac{1}{N_{R_{\max}}} \sum_{i=1}^{N_{R_{\max}}} \left(\frac{1}{N_S} \sum_{j=1}^{N_S} \left(\log_{10} \left(s_{\mathrm{NC,est},i,j} \right) - \log_{10} \left(s_{\mathrm{NC,true},i} \right) \right) \right) \tag{S4}$$

$$\sigma_{\mathrm{D}_s} = \sqrt{ \frac{1}{N_{R_{\max}}} \sum_{i=1}^{N_{R_{\max}}} \left(\frac{1}{N_S} \sum_{j=1}^{N_S} \left(\log_{10} \left(s_{\mathrm{NC,est},i,j} \right) - \log_{10} \left(s_{\mathrm{NC,true},i} \right) - \overline{\mathrm{D}_s} \right)^2 \right) } \tag{S5}$$

$$\overline{\mathrm{D}_r} = \frac{1}{N_{R_{\max}}} \sum_{i=1}^{N_{R_{\max}}} \left(\frac{1}{N_S} \sum_{j=1}^{N_S} \left(r_{\mathrm{NC,est},i,j} - r_{\mathrm{NC,true},i} \right) \right) \tag{S6}$$

$$\sigma_{\mathrm{D}_r} = \sqrt{ \frac{1}{N_{R_{\max}}} \sum_{i=1}^{N_{R_{\max}}} \left(\frac{1}{N_S} \sum_{j=1}^{N_S} \left(r_{\mathrm{NC,est},i,j} - r_{\mathrm{NC,true},i} - \overline{\mathrm{D}_r} \right)^2 \right) } \tag{S7}$$

The first sum runs over the NCs from rank 1 to $N_{R_{\max}} = 1000$ as we restrict ourselves to the 1000 largest ones for the quantification as discussed in the main article. The second sum runs over the $N_S = 100$ independent samples considered for each NC, respectively. For the NC size estimations, we consider order of magnitude differences. In Figure S2, these quantities are shown for the considered sampling methods as a function of the sample size and with and without random subsampling.

For the NC size estimations, in all cases, over a range of sample sizes, the average difference $\overline{\mathrm{D}_s}$ between our estimations and the true values tends to increase with sample size before saturating at around zero. This means that our NC size estimates are on average too small for small sample sizes, and tend to increase on average with sample size before saturating around the true values. This increase presumably originates from our NC size estimation formula (see eq. (3)). The larger the sample of genotypes, the more likely that there is a diversity between these genotypes and so the more likely that the average numbers of neutral mutations per site deviate to some extent from the most frequent values of 0 and 3 as well as that their standard deviations deviate to some extent from 0, leading to a more likely larger NC size estimate (see e.g. $(1 + 0.1)(1 + 2.9) = 4.29 > 4 = (1 + 0)(1 + 3)$). Building on this argument, the differences in the results between the sampling methods are likely caused by the differences in the diversity between the sample genotypes generated by these methods, as discussed in the main article.

For the NC robustness estimations, for uniformly random sampling, the average difference $\overline{\mathrm{D}_r}$ is constant and around zero for all sample sizes. This is because the random selection of genotypes together with our NC robustness estimate as calculated in eq. (4) is in line with the definition of the NC robustness. However, for both RW sampling and (to a lesser degree) site scanning sampling, the $\overline{\mathrm{D}_r}$ increases and marginally deviates from zero with increasing sample size. This means that our NC robustness estimates tend to be marginally larger on average than the true values in the considered sample size range for these two sampling methods. As mentioned in the article, this likely arises from the fact that both sampling approaches are to some extent biased towards network nodes with high degree, which correspond to genotypes with high robustness. However, it should also be noted that the standard deviations are larger or roughly equal than the deviation of the average differences from zero in all cases.

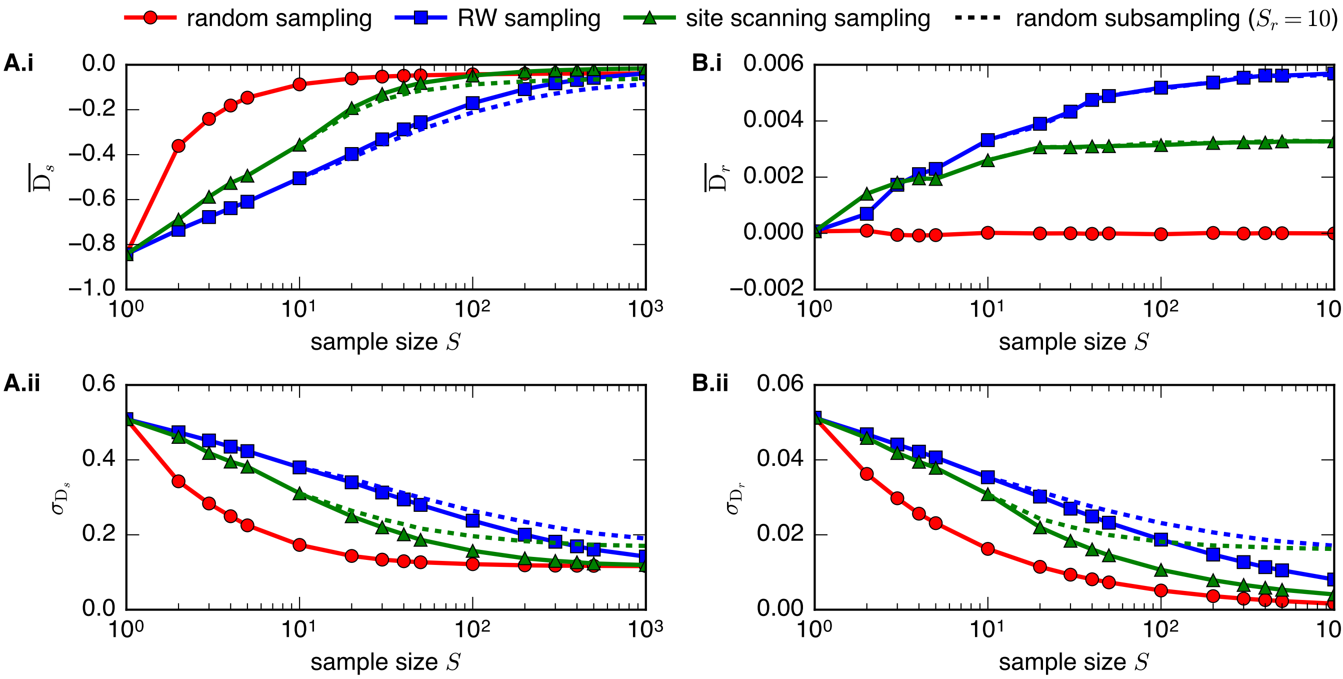

FIG. S2. (i) Average difference $\overline{\mathrm{D}}$ (see eqs. (S4) and (S6)) and (ii) standard deviation σ_{D} of the differences (see eqs. (S5) and (S7)) between the (A) NC size (order of magnitude) and (B) NC robustness estimations and the true values for the 1000 largest NCs of the $L = 15$ RNA secondary structure GP map, as a function of sample size S, for considering uniformly random sampling, random walk (RW) sampling, and site scanning sampling. For the latter two, the dotted lines indicate random subsampling of fixed size $S_r = 10$ from respective full samples (sizes on the x-axis) larger than $S = 10$. For the NC size estimations, eq. (3) together with the found optimal value of $\alpha_{\mathrm{opt}} \approx 0.46$ for $L = 15$ (see Supplementary Information I) is used.

V. FRNADB DATASET

In section IV, we consider longer, naturally occurring functional non-coding RNA sequences. We use the functional RNA database fRNAdb [S1, S2] as source, which can be accessed at `http://www.ncrna.org/`. It is an archived database, which was last updated in 2011. We accessed it on October 3, 2018. A summary of the number of considered sequences from the fRNAdb and their processing can be found in Table SII.

As the starting point, we extract all sequences stored in the fRNAdb of lengths $L = 20$, $L = 40$, $L = 45$, $L = 50$, $L = 55$, $L = 60$, $L = 65$, $L = 70$, $L = 75$, $L = 80$, $L = 85$, $L = 90$, $L = 95$ and $L = 100$ (see second column). Then, we filter the sequences. First, we remove all sequences that are labelled as 'putative' (see third column). Second, from the remaining sequences, we remove all that are labelled as 'piRNA' (see fourth column). For both sequence types, the function or relevance of the secondary structure is not yet confirmed [S4]. Third, from the remaining sequences, we remove all that are incompatible (see fifth column), i.e. sequences that include non-standard nucleotides or for which *RNA.fold* returns the undefined phenotype (unbound structure). A similar dataset has been used for previous NN size estimations [S4].

To all remaining sequences, we apply our introduced estimation framework to estimate the NC size and robustness (see sixth column). To the same sequences, we also apply the *NNSE* by *Jörg et al.* [S3] to estimate the NN sizes of the phenotypes (see seventh column). Not for all input, the *NNSE* converges and returns an estimate (see eighth column). To 100 randomly chosen sequences from those remaining after filtering, we apply the *NNSE* (with relevant setting changed) to additionally estimate the NN robustness of the phenotypes (see ninth column).

L	fRNAdb	filtering			NC estimations	*NNSE*		
	# seq.	# putative	# piRNA	# incompatible	# applied	# applied	# converging	# add. rob.
20	14350	-2384	-617	-4153	7196	7196	7190	100 (random)
40	662	-493	-1	-4	164	164	164	100 (random)
45	537	-397	0	-5	135	135	135	100 (random)
50	475	-265	0	-3	207	207	207	100 (random)
55	509	-292	0	-2	215	215	215	100 (random)
60	354	-246	0	-1	107	107	104	100 (random)
65	543	-212	0	-8	323	323	314	100 (random)
70	2304	-203	0	-28	2073	2073	2062	100 (random)
75	1430	-183	0	-11	1236	1236	1222	100 (random)
80	566	-187	0	-3	376	376	373	100 (random)
85	919	-195	0	-6	718	718	696	100 (random)
90	770	-224	0	-16	530	530	522	100 (random)
95	607	-230	0	-5	372	372	362	100 (random)
100	939	-216	0	-7	716	716	697	100 (random)

TABLE SII. Summary of the number of considered sequences from the fRNAdb and their processing.

VI. OPTIMISATION WITHOUT NC–NN SIZE EXTRAPOLATION

In section IV D, we optimise our size estimation and search for the optimal value of the correction parameter α in the NC size estimation formula (see eq. (3)) for every sequence length in our considered fRNAdb dataset and sample and random subsample size combination, by extrapolating our NC size to NN size estimates (see eq. (7)) and minimising the order of magnitude root-mean-square deviation (RMSD_s) (see eq. (9)) from the estimated NN sizes by the *NNSE*. In the following, we discuss the results of following the same optimisation procedure, but without considering the NC–NN extrapolation, i.e. simply setting for our framework:

$$s_{\mathrm{NN,est}} \approx s_{\mathrm{NC,est}} \tag{S8}$$

In Figure S3, we plot the obtained optimal values for α as a function of the sequence length and the sample and random subsample size combinations in a similar way as we did in Figure 4, where we considered the NC–NN extrapolation. We find that the found optimal values for α strongly vary with sample and random subsample size. In particular, for fixed small sample or small random subsample sizes, the found optimal values for α are identical for many of the sequence lengths, respectively. In these cases, no reasonable optima are reached. This is because our non-extrapolated size estimations do not increase with increasing α above a certain value of α and are too small compared to the reference NN size estimations by the *NNSE* (for details see provided dataset). As discussed in the article, this likely results from the fact that the maximum factors for individual sites in our NC size estimation formula (see eq. (3)) are reached. The obtained identical values for the optimal α are likely an artefact of the used Python function *scipy.optimize.minimize* due to the saturation of our size estimates with increasing α. The results also highlight that the optimal α values – if reasonable optima are found – are quite large and hard to reconcile with those found for sequence lengths $L = 12$ to $L = 16$, not allowing an appropriate functional relation to be fitted to the data.

In Figure S4, we compare the found minimum order of magnitude root-mean-square deviation (RMSD_s) (see eq. (9)) for considering the NC–NN size extrapolation (Figure S4(A), identical to Figure 7(A)) and for not considering it (Figure S4(B)). In all cases without the extrapolation, the minimum RMSD_s values are larger.

An alternative way to quantify the deviation is to consider the average difference ($\overline{\mathrm{D}_s}$) and the standard deviation (σ_{D_s}) of the differences between the size estimations by both frameworks. This is:

$$\overline{\mathrm{D}_s} = \frac{1}{N_L}\sum_{i=1}^{N_L}\left(\log_{10}\left(s_{\mathrm{NN,est},i}\right) - \log_{10}\left(s_{\mathrm{NN,NNSE},i}\right)\right) \tag{S9}$$

$$\sigma_{\mathrm{D}_s} = \sqrt{\frac{1}{N_L}\sum_{i=1}^{N_L}\left(\log_{10}\left(s_{\mathrm{NN,est},i}\right) - \log_{10}\left(s_{\mathrm{NN,NNSE},i}\right) - \overline{\mathrm{D}_s}\right)^2} \tag{S10}$$

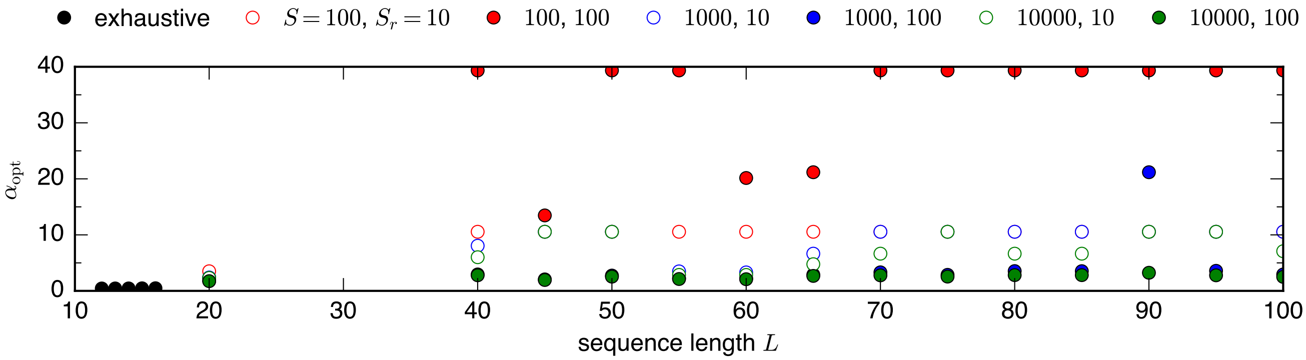

FIG. S3. Optimal values of the correction parameter α in the NC size estimation formula (see eq. (3)) that leads to minimum order of magnitude root-mean-square deviations (RMSD_s) (see eq. (9)) between the NN size estimations by our framework and the reference estimations by the *NNSE* for all considered sequences from the fRNAdb for a given sequence length L for which the *NNSE* converges. Different to Figure 4, we do not consider an extrapolation from NC to NN sizes in our framework, i.e. we simply set them equal to each other (see eq. (S8)). The coloured dots indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. The black dots indicate the results for sequence lengths $L = 12$ to $L = 16$ from the exhaustive analysis in the Supplementary Information I. There is a strong variation in the optimal α with sample and random subsample size. In addition, for fixed small sample or small random subsample sizes, the optimal α values are identical for many of the sequence lengths, respectively. This is an artefact as no reasonable optima are reached in these cases. No appropriate fits to the data are possible.

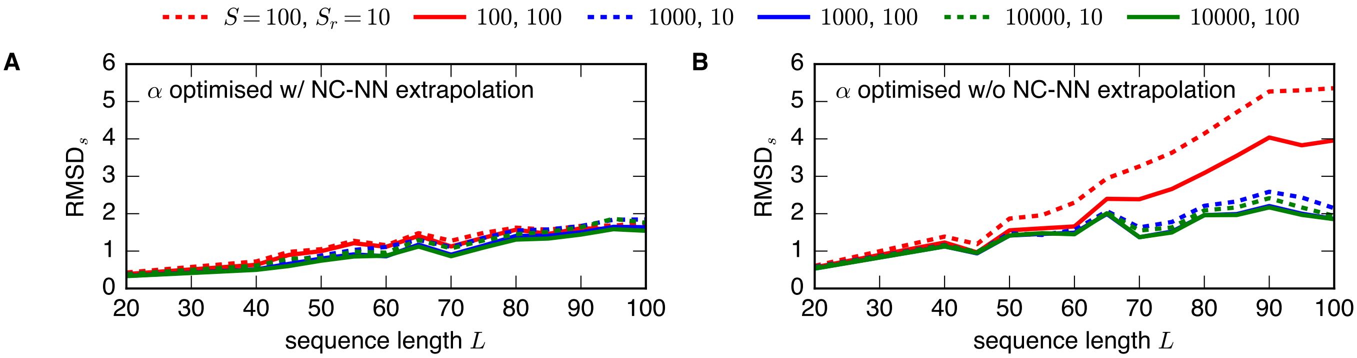

FIG. S4. Minimum order of magnitude root-mean-square deviation (RMSD$_s$) (see eq. (9)) between the optimised NN size estimations (via optimised α values for the NC size estimations) by our framework and the reference estimations by the *NNSE* for all considered sequences from the fRNAdb for a given sequence length L for which the *NNSE* converges, (A) with considering a NC–NN size extrapolation (see eq. (7)) and (B) without considering a NC–NN size extrapolation (see eq. (S8)) for our framework. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. In all cases, the minimum RMSD$_s$ is larger if no NC–NN size extrapolation is considered compared to the case when it is considered.

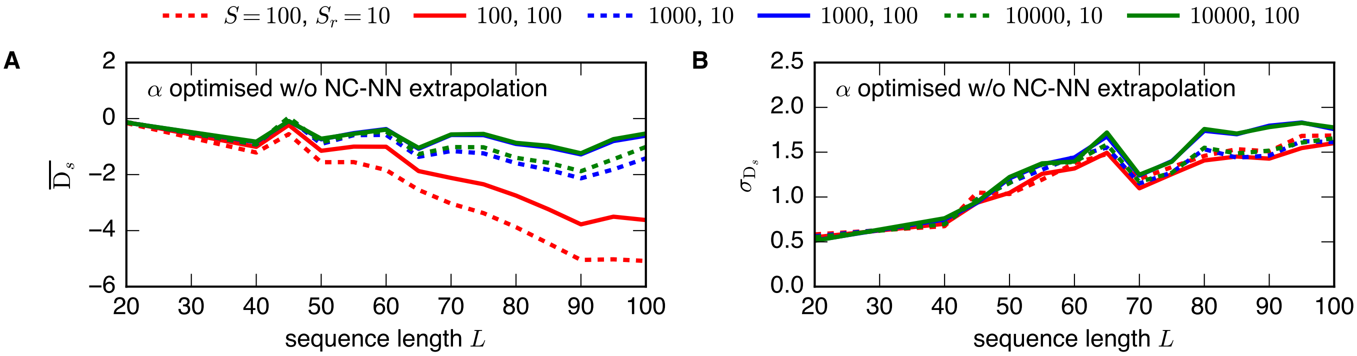

FIG. S5. Order of magnitude (A) average difference $\overline{\mathrm{D}_s}$ (see eq. (S9)) and (B) standard deviation σ_{D_s} of the differences (see eq. (S10)) between the optimised non-extrapolated NN size estimations (via optimised α values for the NC size estimations) by our framework and the reference estimations by the *NNSE* for all considered sequences from the fRNAdb for a given sequence length L for which the *NNSE* converges. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. In comparison to the reference NN size estimates by the *NNSE*, in all cases, our optimised size estimates are too small.

where the sums run over all considered sequences from the fRNAdb of sequence length L for which the *NNSE* converges (total number: N_L). $s_{\mathrm{NN,NNSE},i}$ is the reference NN size estimate by the *NNSE* and $s_{\mathrm{NN,est},i}$ the NN size estimate by our framework for sequence i, respectively. In Figure S5, the results are shown using the optimal α values obtained without considering the NC–NN size extrapolation. The results further highlight that in this case our optimised size estimates are in all cases on average too small compared to the reference NN size estimates by the *NNSE*, with the average differences significantly exceeding the standard deviations of the differences for sample size $S = 100$ and the larger sequence lengths.

Due to all of these observations, we come to the conclusion that the NC–NN size extrapolation is required in our framework, i.e. NNs significantly fragment into NCs, at least for sequence lengths up to $L = 100$.

VII. ADDITIONAL FIGURES: APPLICATION TO FUNCTIONAL NON-CODING RNA SEQUENCES

In section IV E, we quantify the agreement between the extrapolated estimations by our framework and the reference ones by the *NNSE* by considering the root-mean-square deviation (RMSD).

For the NN size estimations, a first alternative way is to consider the relative order of magnitude root-mean-square deviation (RMSD$_s$), which is given by:

$$\text{relative RMSD}_s = \sqrt{\frac{1}{N_L}\sum_{i=1}^{N_L}\left(\frac{\log_{10}(s_{\text{NN,est},i}) - \log_{10}(s_{\text{NN,NNSE},i})}{\log_{10}(s_{\text{NN,NNSE},i})}\right)^2} \tag{S11}$$

where the sum runs over all considered sequences from the fRNAdb of sequence length L for which the *NNSE* converges (total number: N_L). $s_{\text{NN,NNSE},i}$ is the reference NN size estimate by the *NNSE* and $s_{\text{NN,est},i}$ the extrapolated NN size estimate by our framework for sequence i, respectively. For our framework, we estimate the NC size with eq. (3) (α dependent) and use the NC–NN extrapolation given by eq. (7). In Figure S6, the respective results are shown in a similar way as for Figure 7. For Figure S6(A), the optimal α values from the optimisation (see Figure 4) are used for our NC size estimation, i.e. the α values that minimise the (absolute) RMSD$_s$, while for Figure S6(B), the α values from the derived functional relation (see eqs. (10) and (11)) are used. For the former, in all cases, the relative RMSD$_s$ is roughly constant across the considered range of sequence lengths. For the latter, this is true as well except for sample size $S = 100$, for which the relative RMSD$_s$ is increasing with sequence length, though this small sample size has also not been taken into account for the derivation of the functional relation. The constancy for the other sample sizes demonstrates that the increase of the (absolute) RMSD$_s$ with sequence length is in line with the increase of the NC sizes and so NN sizes with sequence length.

A further alternative way is to consider the average difference ($\overline{\text{D}}$) and the standard deviation (σ_{D}) of the differences between the estimations by both frameworks. For the NN size estimations, as already stated in the Supplementary Information VI, this is given by eqs. (S9) and (S10). In Figure S7, the results are shown. Again, for Figure S7(A), the optimal α values from the optimisation are used for our NC size estimation and for Figure S7(B), the α values from the derived functional relation. In the former case, the average difference $\overline{\text{D}_s}$ is approximately zero across all sequence lengths and sample and random subsample size combinations, highlighting that the optimisation not only leads to minimal RMSD$_s$, but also close to optimum agreement between NN size estimations by both frameworks on average. In the latter case, for sample sizes $S = 1000$ and $S = 10000$, which have been used to derive the functional relation for α, the average difference mainly 'fluctuates' around zero and there is no significant trend across the considered range of sequence lengths. For a smaller random subsample size, the average difference tends to be slightly smaller, i.e. our extrapolated NN size estimates slightly smaller. For $S = 100$, which has not been taken into account for the derivation of the functional relation, the average difference tends to significantly decrease with increasing sequence length, highlighting that this sample size is too small.

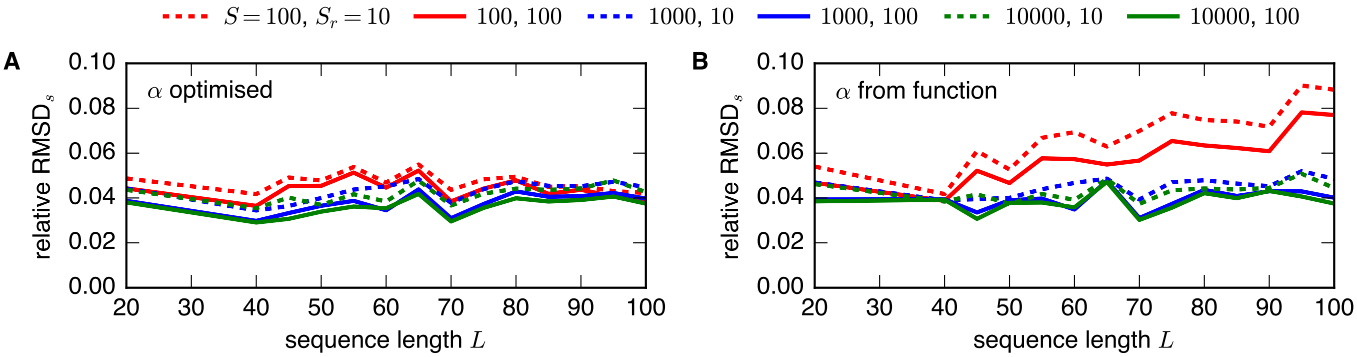

FIG. S6. Relative order of magnitude root-mean-square deviation (RMSD$_s$) (see eq. (S11)) between the extrapolated NN size estimations by our framework and the reference estimations by the *NNSE* for all considered sequences from the fRNAdb for a given sequence length L for which the *NNSE* converges. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. For the NC size estimations, before extrapolation, we use eq. (3) and in (A) the α values from the optimisation (see Figure 4), i.e. those that minimise the (absolute) RMSD$_s$, and in (B) the α values from the derived functional relation (see eqs. (10) and (11)). In all cases, except of (B) and sample size $S = 100$, which has not been taken into account for the derivation of the functional relation for α, the relative RMSD$_s$ is roughly constant with sequence length, and so the increase in (absolute) RMSD$_s$ in line with the increase of the NC sizes and so NN sizes with sequence length.

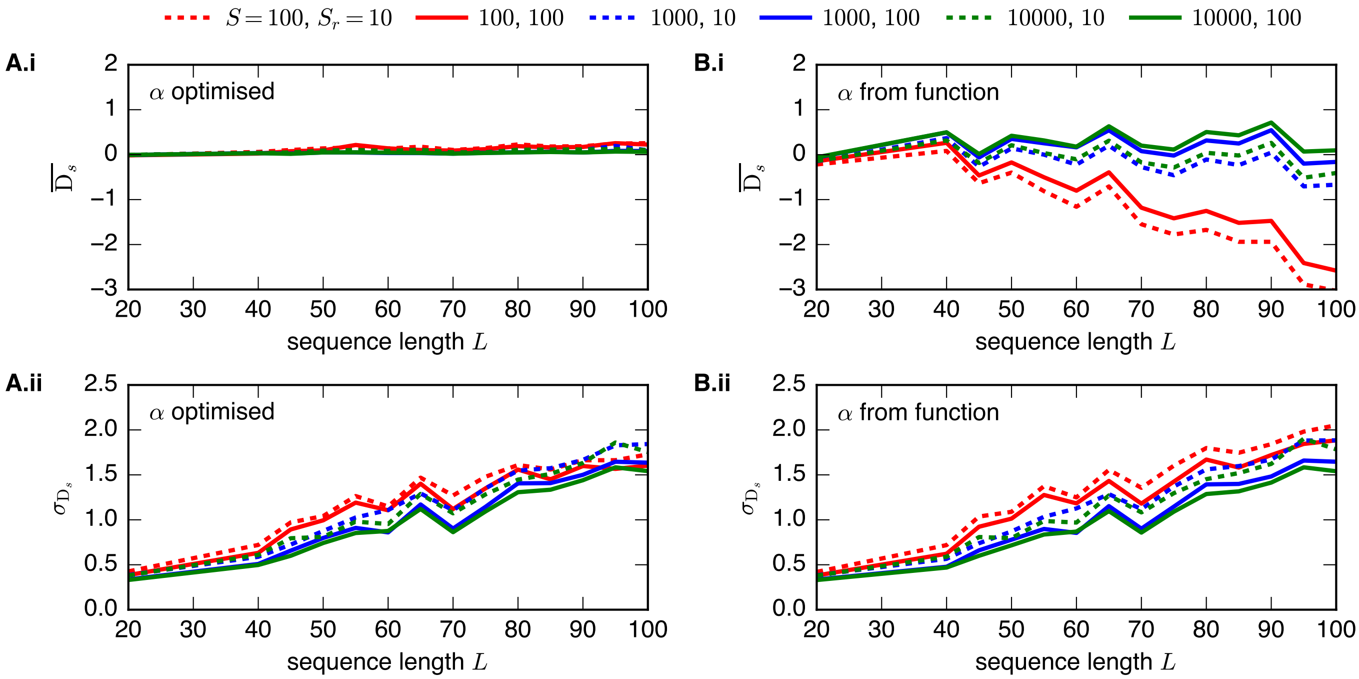

FIG. S7. Order of magnitude (i) average difference $\overline{\mathrm{D}_s}$ (see eq. (S9)) and (ii) standard deviation σ_{D_s} of the differences (see eq. (S10)) between the extrapolated NN size estimations by our framework and the reference estimations by the *NNSE* for all considered sequences from the fRNAdb for a given sequence length L for which the *NNSE* converges. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework. For the NC size estimations, before extrapolation, we use eq. (3) and in (A) the α values from the optimisation (see Figure 4), and in (B) the α values from the derived functional relation (see eqs. (10) and (11)).

In Figure S7, the results also show that in all cases the standard deviation σ_{D_s} of the differences significantly increases with sequence length, highlighting that it is mainly the spread of the differences that causes the increase in the RMSD$_s$ with sequence length. As mentioned in the article, this likely originates from the fact that our NC size estimation formula (see eq. (3)) as well as the NN size estimation formula considered by the *NNSE* [S3] consists of a product for which the number of factors increases with sequence length, leading to increasing uncertainties in the estimations themselves. It should be noted that for $S = 1000$ and $S = 10000$, in all cases, the standard deviation of the differences is larger than the deviation of the average difference from zero, showing that the deviation is not significant.

For the NN robustness estimations, the alternative error measures are given by:

$$\overline{\mathrm{D}_r} = \frac{1}{N_{L,r}} \sum_{i=1}^{N_{L,r}} \left(r_{\mathrm{NN,est},i} - r_{\mathrm{NN,NNSE},i} \right) \tag{S12}$$

$$\sigma_{\mathrm{D}_r} = \sqrt{\frac{1}{N_{L,r}} \sum_{i=1}^{N_{L,r}} \left(r_{\mathrm{NN,est},i} - r_{\mathrm{NN,NNSE},i} - \overline{\mathrm{D}_r} \right)^2} \tag{S13}$$

where the sums run over the $N_{L,r} = 100$ randomly selected sequences for each sequence length, for which we use the *NNSE* to additionally estimate the NN robustness. In Figure S8, the results are shown.

In most cases, the average difference $\overline{\mathrm{D}_r}$ is marginally larger than zero, i.e. our NN robustness estimates are marginally larger on average than those by the *NNSE*. As mentioned in the article, there are two potential reasons. First, we actually estimate NC and not NN robustness and only assume that both are equal. In fact, the robustness of a NC might be larger than of its NN due to the fragmentation of the NN into NCs, leading to a marginal overestimation of our NN robustness estimates. Second, the considered site scanning sampling is likely to some extent biased to network nodes with high degree and so robustness, potentially leading to marginally overestimated NC and so NN robustness by our method. Similarly, the standard deviation σ_{D_r} of the differences is roughly independent of the sequence length and decreases with increasing sample or random subsample size up to a certain amount. Again, in all cases, the standard deviation of the differences is larger than the deviation of the average difference from zero, showing that the deviation is not significant.

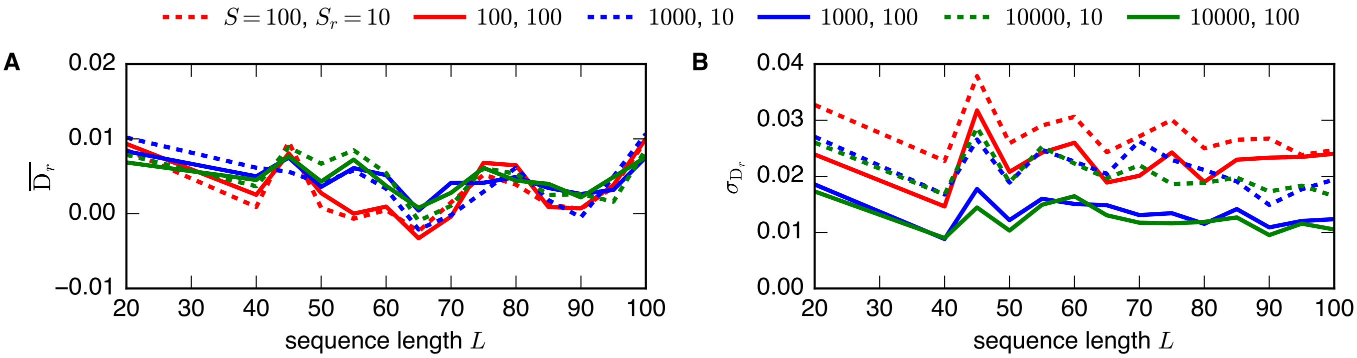

FIG. S8. (A) Average difference $\overline{\mathrm{D}_r}$ (see eq. (S12)) and (B) standard deviation σ_{D_r} of the differences (see eq. (S13)) between the extrapolated NN robustness estimations by our framework and the reference estimations by the *NNSE* for 100 randomly selected sequences from the fRNAdb of given fixed sequence length L. The coloured lines indicate results for different sample sizes S and random subsample sizes S_r considered for our framework.

VIII. RNA SECONDARY STRUCTURE ‘COMPATIBLE’ BASE PAIR MUTATIONS

In section IV F and eq. (13), we describe the computational costs for one estimation run for our framework in terms of the number of calls of the folding function *RNA.fold*. The second term in eq. (13) corresponds to the measurement of the one-point mutational neighbourhoods of the S_r genotypes in the random subsample. When doing so, we restrict ourselves to RNA secondary structure compatible base pairs {CG,GC,AU,UA,GU,UG}, i.e. for a paired site, we only check if a one-point mutation is neutral if the mutated base pair is still one of the six compatible combinations. For a paired site, we describe the number of one-point mutations that are compatible by γ. In the following, we outline how to derive an average approximate value for γ.

In simple algorithmic approaches to estimate the free energy of an RNA secondary structure, it is assumed that a single CG/GC base pair is about two times as strong (two times the energy contribution) as a AU/UA base pair, and about three times as strong (three times the energy contribution) as a GU/UG base pair [S5]. Based on this, we simply assume that a CG/GC base pair is twice as frequent than a AU/UA base pair, and three times more frequent than a GU/UG base pair. This implies that for a base pair the probability to be formed out of CG or GC is $\frac{3}{11}$, respectively, to be formed out of AU or UA is $\frac{1}{2} \cdot \frac{3}{11}$, respectively, and to be formed out of GU or UG is $\frac{1}{3} \cdot \frac{3}{11}$, respectively. Using this, the weighted average of the number of compatible one-point mutations for a paired site is given by $\frac{13}{22}$ and so $\gamma \approx \frac{13}{22} \approx 0.59$. See Table SIII for a detailed calculation.

base pair site 1	base pair site 2	assumed probability	# compatible mutations site 1	# compatible mutations site 2
C	G	$\frac{3}{11}$	1 (C→U)	0
G	C	$\frac{3}{11}$	0	1 (C→U)
A	U	$\frac{1}{2} \cdot \frac{3}{11}$	1 (A→G)	0
U	A	$\frac{1}{2} \cdot \frac{3}{11}$	0	1 (A→G)
G	U	$\frac{1}{3} \cdot \frac{3}{11}$	1 (G→A)	1 (U→C)
U	G	$\frac{1}{3} \cdot \frac{3}{11}$	1 (U→C)	1 (G→A)
	weighted average		$\frac{13}{22} \approx 0.59$	$\frac{13}{22} \approx 0.59$

TABLE SIII. Detailed calculation of the weighted average number of compatible one-point mutations for a paired site in an RNA secondary structure phenotype, i.e. the weighted average number of one-point mutations of a paired site that still lead to one of the six compatible base pairs {CG,GC,AU,UA,GU,UG}.

[S1] T. Kin, K. Yamada, G. Terai, H. Okida, Y. Yoshinari, Y. Ono, A. Kojima, Y. Kimura, T. Komori, and K. Asai, Nucleic Acids Research **35**, D145 (2007).

[S2] T. Mituyama, K. Yamada, E. Hattori, H. Okida, Y. Ono, G. Terai, A. Yoshizawa, T. Komori, and K. Asai, Nucleic Acids Research **37**, D89 (2008).

[S3] T. Jörg, O. C. Martin, and A. Wagner, BMC Bioinformatics **9**, 464 (2008).

[S4] K. Dingle, S. Schaper, and A. A. Louis, Interface Focus **5**, 20150053 (2015).

[S5] F. J. Burkowski, *Structural Bioinformatics: An Algorithmic Approach*, 1st ed. (Chapman & Hall/CRC, 2008)