Methods xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Computational analysis of fitness landscapes and evolutionary networks from *in vitro* evolution experiments

Ramon Xulvi-Brunet [a,b,*,1], Gregory W. Campbell [b,1], Sudha Rajamani [c], José I. Jiménez [d], Irene A. Chen [b,*]

[a] Departamento de Física, Facultad de Ciencias, Escuela Politécnica Nacional, Quito, Ecuador
[b] Department of Chemistry and Biochemistry, Program in Biomolecular Science and Engineering, University of California, Santa Barbara, USA
[c] Indian Institute of Science Education and Research, Pune, India
[d] Faculty of Health and Medical Sciences, University of Surrey, Guildford, UK

ARTICLE INFO

Article history:
Received 13 February 2016
Received in revised form 16 May 2016
Accepted 18 May 2016
Available online xxxx

Keywords:
In vitro evolution
RNA selection
Fitness landscape

ABSTRACT

In vitro selection experiments in biochemistry allow for the discovery of novel molecules capable of specific desired biochemical functions. However, this is not the only benefit we can obtain from such selection experiments. Since selection from a random library yields an unprecedented, and sometimes comprehensive, view of how a particular biochemical function is distributed across sequence space, selection experiments also provide data for creating and analyzing molecular fitness landscapes, which directly map function (phenotypes) to sequence information (genotypes). Given the importance of understanding the relationship between sequence and functional activity, reliable methods to build and analyze fitness landscapes are needed. Here, we present some statistical methods to extract this information from pools of RNA molecules. We also provide new computational tools to construct and study molecular fitness landscapes.

1. Introduction

1.1. In vitro evolution and fitness landscapes of nucleic acids

Since its introduction in the early 1990s [1–3], *in vitro* evolution has proven to be a successful way to discover novel molecules that are suited to carry out specific biochemical functions. RNA is widely used in selection experiments because of its ability to carry out biochemical activities, both in modern biology and in the putative *RNA World* of early life, and the availability of methods to synthesize, amplify, and sequence pools of RNA. There are many examples of functional nucleic acid molecules that result from *in vitro* selections, which have been reviewed elsewhere [4]. For example, the *Spinach2* aptamer [5] regulates the fluorescence of a fluorophore via binding, providing an interesting alternative to GFP tagging. *In vitro* selection has been applied to discover a variety of ribozymes [6], deoxyribozymes [7], aptazymes [8], and riboswitches [9]. With appropriate polymerase enzymes, selection experiments can even be done with non-natural nucleic acids, as shown for a series of xeno-nucleic acids (XNAs). The growing body of work on *in vitro* selection of nucleic acids demonstrates the utility of this evolutionary technique.

Nucleic acid sequences can be thought of as occupying points in the discrete space of all possible sequences (4^L, where L is the sequence length and the number of dimensions in sequence space). For a given activity, each point is associated with a fitness value, which defines a *fitness landscape* [10,11] in which highly active sequences form peaks. If that activity were under selection, the evolution of a population of sequences would be largely governed by the topology of the fitness landscape [12]. The complete delineation of a fitness landscape is practically limited to short sequences (certainly $L < 30$), due to the exponential explosion in the number of possible sequences with increasing length. The fitness landscapes of short RNA macromolecules are of interest despite their small size, as short sequences can be functional (i.e., ribozymes and aptamers) and they are likely to be more abundant in an abiotic synthesis compared to long sequences. In chemical terms, the fitness landscape of sequences is essentially identical to a structure–activity relationship (SAR), where the structure is primary (i.e., sequence).

Detailed knowledge of a fitness landscape is of clear interest for basic understanding of the SAR and the evolution of novel functions. Data from *in vitro* selections, in general, are well suited to

* Corresponding authors at: Department of Physics, Faculty of Sciences, National Polytechnic School, Quito, Ecuador (R. Xulvi-Brunet) and Department of Chemistry and Biochemistry, Program in Biomolecular Science and Engineering, University of California, Santa Barbara, USA (I.A. Chen).
E-mail addresses: ramon.xulvi@epn.edu.ec (R. Xulvi-Brunet), chen@chem.ucsb.edu (I.A. Chen).
[1] R. Xulvi-Brunet and G. W. Campbell contributed equally to this work.

http://dx.doi.org/10.1016/j.ymeth.2016.05.012
1046-2023/

mapping fitness (or activity) landscapes. If the map possesses sufficient (e.g., base-by-base) resolution in sequence space, potential evolutionary pathways can be discovered by identifying networks of sequences that are connected by small mutational steps. The importance of understanding the relationship between functional activity and sequence, together with the potential to study evolutionary issues, provides motivation for constructing and analyzing maps of molecular fitness landscapes [13–15]. However, until the advent of high-throughput sequencing techniques, maps of the fitness landscape were difficult, if not impossible, to obtain, owing to the small amount of sequencing information available via low-throughput techniques.

High-throughput sequencing (HTS) has been used to delineate fitness landscapes localized around a known ribozyme [13]. In such work, the fitness of a given sequence during the selection can be determined by a direct comparison of the frequency of that sequence before and after selection, owing to the relatively small number of variants. In principle, this method could be extended to delineate the entirety of sequence space given unlimited sequencing capacity. However, in practice the technique has limited power to explore the vast volume of sequence space, because the number of sequences for which the fitness can be inferred cannot exceed the capacity of the sequencing (e.g., $< 10^{10}$ sequence reads for Illumina HTS systems at the time of this writing). In contrast, the number of sequences that can be included in the experimental selection, such as from a random pool, is much higher ($\sim 10^{16}$ molecules). The process of sequencing under-samples the diversity of the pre-selection pool, in particular. Therefore, a method for inferring the pre-selection frequency of any sequence would allow access to much larger volumes of sequence space, up to the experimental capacity of the selection. It must be noted, however, that post-selection frequencies must be measured rather than inferred, since we have little ability to predict the outcome of selection, so the limitation on sequencing capacity translates into a threshold fitness (i.e., frequency in the post-selection pool), below which sequences cannot be reliably detected. Although they are measured, post-selection frequencies may be distorted by biases associated with sample preparation and sequencing. We have discussed these issues in detail elsewhere [16]. Here, we briefly review these issues and describe the computational methods we have used to reconstruct fitness landscapes from pre- and post-selection frequencies [15] as a guide to potential practitioners. This includes expanded capabilities compared to our previously described platform.

1.2. Theoretical issues associated with estimating true pre-selection and post-selection frequencies

In any selection experiment, we need to *observe* the sequences contained in the experimental pool, both before and after selection. When the goal is to reliably construct the fitness landscape associated with the selection process, then observing sequences entails accurately counting (or estimating) the number of each type of molecule (i.e., unique sequence) present in those pools.

1.2.1. The pre-selection pool

If we want to map a complete fitness landscape, ideally we start with an initial pool of molecules that contains all possible unique sequences, with roughly equal representation. This requirement automatically limits the length of the random region of sequences that can be explored, since the number of possible unique sequences scales exponentially with the length of the random region. In practice, we are limited to randomized regions of 24 nucleotides if stochastic variations in the initial pool are to be minimized. A few more nucleotides can be added to the length of the random region at the expense of copy number in the initial pool, but 28 nucleotides ($\sim 10^{17}$) is probably the limit of laboratory feasibility. We assume (consistent with most selection experiments) that the sequence length is constant (or at least highly similar) across the selection pool. As an example, let us assume that the length of sequences in a given random pool is 24 nucleotides. This means that the number of unique sequences that would comprise the ideal, initial pool is roughly $4^{24} \simeq 10^{14}$. In order to accurately estimate the distribution of molecules (i.e., to find the copy number of each unique sequence), we would like to have numerous copies of each unique sequence. Let us assume that we want 100 copies of each of the 4^{24} unique sequences. In this case, we would need to count around 10^{16} total molecules in order to obtain the true distribution. Although this number of molecules can be handled experimentally in bulk, current sequencing technology cannot count this number. HTS might identify roughly 10^{10} molecules, which is several orders of magnitude smaller than the number of possible unique sequences (10^{14}). Thus, the number of molecules that we can experimentally count is much too small to directly measure the true distribution of molecules in a pool of this diversity.

Although the pre-selection abundance of an individual 24-mer sequence cannot be directly measured by sequencing, one may calculate a good estimate for its abundance through the creation of a quantitative, semi-empirical model that describes how an initial pool of sequences is synthesized [15,16]. This is necessary since oligonucleotide synthesis is subject to chemical biases (e.g., in coupling efficiency), which prevent true randomness. This model describes the synthesis from a relatively small set of parameters, which themselves are estimated from statistical properties of the sequences present in the pool. For example, we may estimate the differential coupling efficiencies of A with A, A with C, A with G, etc., from sequence reads in the pre-selection pool. We can thus estimate the pre-selection abundances of sequences in the initial pool. Using model-selection criteria [17], we have found [15,16] that a reasonable model is one that assigns different reactivities to a growing chain depending on the identity of the last two nucleotides and of the incoming nucleotide. This model requires a relatively small number of parameters that are estimated from the sequences' statistics of the pre-selection pool.

Let us consider the case of direct synthesis of an RNA pool. To estimate the abundance of a sequence, we need to know the probability with which a nucleotide incorporates to a nascent sequence at any time. To do that, we first determine the probability with which the first two nucleotides are synthesized. We assume that the first nucleotide is attached to the solid support with equal probability for the four nucleotides, i.e., with $p_j = 1/4,\ \forall j \in \{\mathrm{G, A, C, U\ (or\ T)}\}$. (Current protocols allow the user to set, with reasonable accuracy, arbitrary values for p_j by manual mixing of resins.) Next, we assume that the probability that the second nucleotide is attached to the first is given by $\mathcal{P}(i|j) = r_{ij}C_i/\sum_{i'} r_{i'j}C_{i'}$, where C_i and $C_{i'}$, for $i, i' \in \{\mathrm{G, A, C, U\ (or\ T)}\}$, are the concentrations of the nucleotides in solution, and r_{ij} are 16 chemical reactivity parameters that account for the likelihood of dimerization between each potential pairing of the 4 nucleotides G, A, C, and U (or T) [16].

For the next and subsequent nucleotides, our assumption is that the probability that a nucleotide is attached to the nascent sequence is given by $\mathcal{P}(i|j,k) = r_{ijk}C_i/\sum_{i'} r_{i'jk}C_{i'}$, where r_{ijk} describes the likelihood that nucleotide i attaches to a chain whose second-to-last and last nucleotides are, respectively, j and k. Again, $C_{i'}, i' \in \{\mathrm{G, A, C, U(or\ T)}\}$, are the nucleotide concentrations in solution. The total number of parameters in this model is $4^2 + 4^3 = 80$. In order to estimate these 80 parameters, we use the

maximum likelihood estimator technique [18], which states that $\mathcal{P}(i|j)$ can be estimated as

$$\mathcal{P}(i|j) = \frac{d_{ji}}{\sum_{k=1}^{4} d_{jk}}, \quad (1)$$

and $\mathcal{P}(i|j,k)$ can be estimated as

$$\mathcal{P}(i|j,k) = \frac{t_{jki}}{\sum_{m,n=1}^{4} t_{jmn}}. \quad (2)$$

In these equations, d_{ji} and t_{jki} are the number of times a specific dimer or trimer, respectively, is found in the central part of the sequences in the synthesized pool. The reason to only count dimers and trimers in non-terminal regions of the sequences is that adapter ligation and reverse transcription can bias the first and last six nucleotides of the sequences [16]. Thus, we recommend estimating d_{ji} and t_{jki} by counting dimers and trimers whose nucleotides do not overlap with the terminal hexamers.

Once $\mathcal{P}(i|j)$ and $\mathcal{P}(i|j,k)$ have been estimated, we compute the probability of finding sequence $i_1 i_2 i_3 \ldots i_n, i_k \in \{\mathrm{G}, \mathrm{A}, \mathrm{C}, \mathrm{U(or\ T)}\}$, in the synthesized pool as

$$P_{i_1 i_2 i_3 \ldots i_n} = p_{i_1} \mathcal{P}(i_2|i_1)\mathcal{P}(i_3|i_2, i_1) \ldots \mathcal{P}(i_n|i_{n-1} i_{n-2}). \quad (3)$$

Note that, as discussed above, p_{i_1} is assumed to be 1/4.

1.2.2. The post-selection pool

High-throughput sequencing is necessary, as the amount of sequencing data limits the fitness threshold of the landscape map. However, several biases can arise during the workup for HTS. Possible biases included in the HTS workup include the following.

(1) 5′ **and** 3′ **adapter ligation**: Small segments of RNA (or DNA) with a specific known sequence are ligated to the 5′ and 3′ ends of the sequences in the selected pool. These segments are needed as constant regions, designed as binding sites for PCR primers, or to anneal the sequences to a support for sequencing. We had previously chosen an approach of ligating single-stranded adapter sequences to the randomized N24 sequences, in order to avoid effects from built-in PCR primer regions during the selection [15]. However, this introduces bias from the process of adapter ligation, which is catalyzed by T4 RNA ligase. The efficiency of ligating adapters to RNA or DNA sequences depends on the nucleotides at the terminal regions of those sequences. For instance, England and Uhlenbeck [19] and Middleton et al. [20], showed that sequences containing GC, AC, and GA at the 3′ end, and TT, TA, and AT at the 5′ end, were observed more frequently after ligation, while sequences with UU at the 3′ end and GG at the 5′ end were observed less frequently. This effect can be corrected for during computational analysis (see following section). Alternatively, PCR primer sequences can be built into the original design of the initial pool. The choice of approach depends on the goals of the experimenter. Our computational pipeline can be adapted to either approach.

(2) **Reverse transcription**: When the nucleic acid is RNA, reverse transcription is carried out to convert the RNA sequence into DNA, which is required for current HTS methods. Reverse transcription is initiated by a primer annealing to the 3′ end of a sequence. Therefore, the efficiency of the process may depend on the specific sequence to which the primer anneals. Taken together with adapter ligation, both protocol steps effectively make sequences which begin and/or end with a particular nucleotide combination more or less likely to be *observed*. Characterizing these effects allows correction of the distortion imposed by these biases [16].

The true abundance of sequences in a **pre-selection** pool can be computed via the model described in Section 1.2.1. Thus, no information about the ligation or reverse transcription bias is needed. Only two things need to be kept in mind: First, since the relative abundance of sequences reported by a sequencing device is somewhat inaccurate due to sequencing errors, those distortions need to be corrected. This can be done by any algorithm designed to correct sequencing bias (such as the one described below). Second, probabilities $\mathcal{P}(i|j)$ and $\mathcal{P}(i|j,k)$ have to be estimated, as described above, from dimers and trimers located at the central part of sequences (which are not affected by the combined bias of ligation and reverse transcription).

Post-selection pools, however, do need to be treated with some kind of correction method to estimate the true abundance of each sequence. The method that we propose uses the biases found in the 5' and 3' terminal regions of sequences in the pre-selection pool as the basis for correcting the post-selection pools. This involves the following steps:

I. Examine the relative abundance of all pre-selection sequences reported by the sequencing device and correct the observed abundance for sequencing errors (e.g., using the method described later in this section).
II. Classify the observed sequences of the pre-selection pool from step I according to the first and last 5 nucleotides at the terminal regions of each sequence. That is, arrange the sequences into the $1024 \cdot 1024 = 1048576$ potential sequence classes, such that all sequences in each class have the same first and last 5 nucleotides. Then, for each class l, compute the total abundance of that class; that is, sum the abundances of all sequences that belong to class l. Finally, divide the total abundance of each class l by the total number of observed molecules. This gives us the *observed* probability p_l^{ob} that a pre-selection sequence belongs to class l.
III. Using the model described in Section 1.2.1, compute the true abundance of the sequences in the pre-selection pool. Next, sort them into the same 1048576 sequence classes described above and sum the true abundances of all pre-selection sequences that belong to each class l. Divide the total abundance of each class l by the total number of molecules to obtain the probability p_l^{true} that a pre-selection sequence belongs to class l.
IV. For each sequence class, l, determine the correcting factor, f_l. We find f_l by dividing p_l^{true} by p_l^{ob}.
V. Examine the relative abundance of all post-selection sequences reported by the sequencing device and correct the observed abundance for sequencing errors (e.g., using the method described later in this section).
VI. We correct for the adapter ligation and reverse transcription biases of post-selection data as follows: First, we identify the class to which a post-selection sequence, s_k, belongs (step II); this determines which correcting factor, f_l, should be used. Second, we multiply the *observed*, post-selection abundance of s_k by that factor to obtain the corrected, post-selection abundance of the sequence.

(3) **PCR amplification**: PCR can be used to amplify the number of molecules we want to observe and to introduce additional constant regions. Ideally, all molecules would be increased by the same factor, so that a sequence distribution is not artificially distorted. However, although most sequences are indeed amplified by approximately the same factor, a

small subset may be multiplied much more or much less than expected (e.g., by an order of magnitude) [21]. If the goal is to accurately estimate a given sequence distribution, we recommend minimizing PCR amplification when possible.

(4) **Sequencing**: DNA sequences consisting of known constant regions flanking the initially randomized region are sequenced by HTS. This step can generate millions to billions of sequence reads. Sequencing technology sometimes misreads nucleotides, which effectively blurs the sharpness of the observed distribution in sequence space. However, this blurring effect can be corrected when accurate models are developed to describe how and why nucleotides are occasionally read incorrectly [22–28]. Therefore, it is possible to quantify the effects of those errors on a real distribution of sequences, and thus recover true sequence distributions from observed distributions by designing appropriate algorithms. In practice, the error rate of the sequencing technique should be known so that appropriate corrections can be made to the fitness landscape. We use what is probably the simplest model that can quantitatively describe the sequencing bias associated with the identification of nucleotides by a sequencing device. This model is a first approximation to deal with sequencing errors. More specific models associated with specific sequencing devices may be necessary to improve accuracy.

Our simple model assumes that the probability of a sequencing device misreading a nucleotide is constant and independent of the type of nucleotide to be read, the sequential position of the nucleotide, and the neighboring nucleotides in the sequence. Additionally, we assume that when an erroneous read occurs, the probability of reporting each of the three possible incorrect nucleotides is 1/3, i.e., any of the other three nucleotides are equally likely to be falsely reported. Under these assumptions [16], the total number of unique sequences containing n sequencing errors is given by

$$\mathcal{N}(n) = \frac{3^n L!}{n!(L-n)!}, \tag{4}$$

and the probability of finding a particular sequence containing n errors is

$$P(n) = p^{L-n}\left(\frac{1-p}{3}\right)^n. \tag{5}$$

where L is the length of the sequence and p is the probability that a nucleotide is read correctly.

A simple algorithm to correct for sequencing errors in this model is based on the following idea: for any sequence i, the expected number of copies we observe is equal to the true number of copies multiplied by the probability that it is correctly read, plus the number of variants of other sequences that are erroneously identified as sequence i. In mathematical terms,

$$\langle n_i^{ob}\rangle = n_i^{re}P(0) + \sum_j n_j^{re}P(d_{ij}), \tag{6}$$

where n_i^{ob} is the observed abundance of sequence i (after sequencing), n_i^{re} is the true abundance of i (before sequencing), and $P(d_{ij})$ is the probability given by Eq. (5), where d_{ij} is the distance between sequences i and j.

The correction algorithm is as follows:

Step 4.1. First, the sequences are sorted according to their observed abundance. This results in a list of sequences such that the first sequence is the most abundant, the second is the next most abundant, and so on.

Step 4.2. The second stage of the algorithm is an iterative procedure that performs the following steps on each item in the sorted list described above. At each iteration i, the algorithm estimates the true abundance of the ith sequence in the list as described below:

Step 4.2.1. For each sequence j, $j \neq i$, the quantity

$$c_{ji} = \mathrm{int}\left(\frac{n_j^{ob}}{p^L}\left(\frac{1-p}{3}\right)^{d_{ij}} p^{L-d_{ij}}\right) \tag{7}$$

is computed. $n_j^{ob}/P(0) = n_j^{ob}/p^L$ can be seen as the expected true abundance, n_j^{re}, of sequence j, provided that it is sufficiently different from other sequences. This is certainly an approximation; sequence j might actually have some real neighboring sequences. However, this approximation, $n_j^{re} \simeq n_j^{ob}/p^L$, is the key assumption in this heuristic algorithm. $((1-p)/3)^{d_{ji}}p^{(L-d_{ji})}$ gives the probability that sequence j is misread as sequence i. Finally, $\mathrm{int}(x)$ is just the nearest integer function, which converts any real number x into its nearest integer. Thus, the quantity c_{ji} is an approximation of the expected number of observed copies of sequence i that arises from misreads of sequence j.

Step 4.2.2. The quantity $c_i = \sum_j c_{ji}$ is computed. To a first approximation, c_i gives the expected increase in abundance of sequence i due to erroneous readings of the rest of the sequences.

Step 4.2.3. The observed abundance n_i^{ob} is corrected by subtracting c_i from it; thus, we can write $n_i^c = n_i^{ob} - c_i$, where n_i^c is the corrected abundance of sequence i. This step reduces the abundance of n_i^{ob} by the expected contribution from all other sequences in the distribution.

Step 4.2.4. If the corrected abundance n_i^c satisfies $n_i^c \geqslant 0$, then n_i^c is updated by the next formula: $n_i^u = \mathrm{int}(n_i^c/p^L)$, where n_i^u is the updated abundance of sequence i. Note that the effect of this final correction simply converts the copies of sequence i that were lost due to misreading back into sequence i. If $n_i^c \leqslant 0$, then final updated n_i^u is simply set to zero, $n_i^u = 0$. The role of function $\mathrm{int}(x)$ is to yield a final integer number of copies for each sequence.

In [16], we showed from simulation data that this simple algorithm is sufficiently good to recover a true, unobserved population of sequences from an observed population that had been subject to sequencing errors. With real data, the quality of recovery of the true sequence population depends on how accurately the simple model discussed above describes the misreading process.

To summarize, all four steps listed above are potential sources of experimental bias. As a result, *observed*, post-selection abundances of unique sequences are, in general, different from true, post-selection abundances. As has been shown in [15,16], while the majority of observed abundances are similar to the estimated true abundances, for some sequences these may differ by up to a few orders of magnitude. Therefore, we apply computational techniques to correct for the biases.

Constructing reliable and detailed fitness landscapes demands (1) technology that can identify large numbers of sequences, and (2) computational protocols to assess and correct for experimental biases. Here, we detail the computational protocols used in our laboratory, which have been implemented on the Galaxy bioinformatics platform. These protocols include some recent improvements to our previously published methods [15].

2. Methods

In this section, we first describe the algorithms we use to construct fitness landscapes. We then describe their practical implementation using tools on the Galaxy platform (www.galaxyproject.org) [29]. Our tools are available at http://galaxy-chen.cnsi.ucsb.edu:8080/.

2.1. Estimating the fitness of the experimentally selected sequences

Step 1. First, address the sequencing errors. Following the ideas presented in Section 1.2.2, correct both the observed pre-selection and observed post-selection sequence distributions obtained from the sequencing device in order to remove as much of the effect of sequencing errors as possible.

Step 2. Estimate probabilities $\mathcal{P}(i|j)$ and $\mathcal{P}(i|j,k)$, according to Section 1.2.1, in order to build the model that computes the initial pre-selection sequence distribution.

Step 3. For the post-selection sequences, address biases due to the adapter ligation and reverse transcription. Following the algorithm described in Section 1.2.2, obtain the corresponding distribution corrected for adapter ligation and reverse transcription, if appropriate. Note that the corrected distribution contains only sequences that meet a certain fitness threshold.

Step 4. Extract the abundance of each sequence i present in the corrected, post-selection pool. Next, sum these abundances of all sequences to generate the normalization factor *nf*. Divide the abundance of each sequence i by *nf* in order to obtain the "normalized abundance", i.e., the frequency of sequence i after selection, p_i^{post}.

Step 5. For any sequence i present in the corrected, post-selection distribution, compute the initial frequency of sequence i in the pre-selection pool, p_i^{pre}. We can estimate that initial frequency by using the model mentioned previously in Section 1.2.1.

Step 6. The fitness f_i of each sequence i may be defined as p_i^{post}/p_i^{pre}. In this case, the fitness is defined as the relative enrichment between these two conditions. However, the user may wish to define fitness differently to suit his or her needs.

2.2. Clustering of related sequences

At this point, the sequencing data are in the form of a list of unique sequences and their accompanying estimated fitness. For most purposes, these sequences should be organized with respect to similarity. A central metric is therefore the *distance* between points in sequence space. One of the most often used distances is the Hamming distance [30]: the number of positions at which two sequences are different, assuming that both sequences are equal in length. We can generalize this definition to make it applicable to sequences of different length, as deletions and insertions can occur during *in vitro* evolution. First, we find all possible alignments of the shorter sequence with the longer one. Then, the minimum number of positions at which the shorter sequence differs from the aligned segment of the longer one, plus the difference in lengths, yields the generalized Hamming distance between the two sequences. Alternatively, we can use another definition of distance: the minimum number of operations, i.e., substitutions, insertions, and deletions, we must use in order to convert one sequence into another. This distance, which may always be used regardless of any length difference between the sequences, is also referred to as the *edit* distance [31,32].

The distance metric used for building and analyzing a fitness landscape can heavily influence the topological features of the landscape. A simple example illustrates this point. Consider the following two sequences: GCCAGUGGCUUAGAACGGCAUGGGAC and CCAGUGGCUUAGAACGGCAUGGGACG (note that one is a circular permutation of the other). The Hamming distance between these sequences is 19, while the edit distance between them is just 2. Thus, depending on the definition of distance we use, these two sequences may either be isolated from each other in the landscape, or they may be part of the same peak. When the goal is to group two sequences closely in the landscape if it is likely they have a similar secondary structure, or if the operations of the edit distance are biochemically reasonable, then edit distance is usually the best metric to consider, although it is more computationally taxing than Hamming distance. In general, we choose to use the edit distance, unless the edit distance is expected to give the same outcome as the Hamming distance, in which case the Hamming distance is faster to compute. Our algorithms allow either choice.

Once the distance metric has been chosen, sequences can be grouped into clusters (which may represent fitness peaks, depending on whether sequence space was sufficiently covered) using the following simple algorithm. First, sort all sequences in the corrected, post-selection distribution according to their fitness values. Then, take the highest fitness sequence and compute the distance from that sequence to every other sequence in the distribution. If the distance is less than or equal to a certain pre-defined distance (e.g., 4), consider that sequence to be part of the same potential cluster as the initially chosen highest fitness sequence. If the total number of clustered sequences is larger than a pre-defined cutoff value (e.g., 3), define the first *cluster* (or peak) as the group of sequences formed by the highest fitness sequence (the summit of the peak) and the sequences found to cluster with it. To find the rest of the clusters, follow the same procedure as many times as necessary: Take the highest fitness sequence that is not yet included in a cluster; compute the distances between that sequence and the other sequences in the distribution; find the sequences in the potential cluster; and if the number of clustered sequences is larger than the previously chosen cutoff, define the corresponding cluster (or peak) as the set of sequences formed by the selected highest fitness sequence (the summit) and its cluster of related sequences.

2.3. Discovery of evolutionary pathways

Because the experimental pre-selection pool often provides only a sparse sample of sequence space, and because of the previously discussed subsampling due to sequencing limitations, the fitness landscape constructed from a typical selection experiment is often incomplete, consisting mostly of a set of peaks with different heights and shapes distributed across sequence space, likely in an apparently unrelated fashion. Sometimes, however, with sufficient sequencing depth and coverage of the pre-selection pool, or with certain topologies of the fitness landscape, it is possible to find pathways consisting of high-fitness molecules that make stepwise connections between seemingly distant peaks [15]. Identifying such potential evolutionary pathways is of fundamental interest. In principle, an evolutionary pathway in the data indicates a specific series of mutational steps that could occur during evolution without passing through a deep fitness 'valley'.

The discovery technique that we describe is adapted from modern network theory, and is related to the so-called network tomography problem [33,34]. We start by constructing a network whose nodes, individual sequences in the landscape, will be linked by a network edge if the distance between them is less than or equal to a given number, corresponding to the size of permissible mutational steps. If, for example, we assume that this value is one, then all sequences that are one mutation (i.e., one edit) apart will be linked by a network edge. Once such a network has been constructed, we proceed as follows: Consider a peak center in this network; this initial sequence will be the root node, and we define it as shell number 0 of the tomography procedure. Then, all edges

starting at the root node will be followed until all nodes (i.e., sequences) that are a distance of one from the root node are reached and identified. This new set of nodes is then defined as shell number 1. Next, all edges leaving a node in shell 1 are followed, and all new sequences reached are labeled as nodes of shell number 2. This same procedure is carried out for shell 2, in order to identify the sequences that are 3 mutations away from the root sequence, i.e., shell number 3. The procedure is then repeated until no more new sequences are found by following the edges leaving the last shell. If, by carrying out this algorithm, another "summit sequence" is found to be in any of the shells, then there must exist a pathway between the root sequence and the summit sequence found in that shell; that is, there must exist a pathway between the two peaks. The length of the pathway will be equal to the shell number in which the second summit sequence was found.

Different networks can be constructed within the same landscape, depending on the value of the maximum step size. By following the algorithm outlined above, we can discover pathways between peaks that can be traversed with steps of length 1, or lengths less than or equal to 2, or 3, etc. By increasing the step size, we are essentially allowing larger and larger mutational jumps between neighboring sequences in these pathways. This step size can be set in our algorithm to a value felt to be appropriate by the researcher.

To illustrate how the network changes when the step size is increased, we present some results from one of our *in vitro* directed evolution experiments. In the experiment, selection pressure favoring the ability to bind GTP was applied to a pool of RNA sequences (roughly 24 nucleotides long) over the course of several rounds; the goal was to determine the fitness landscape with respect to GTP-binding [15]. Analysis of the experiment showed that 11 fitness peaks were present in one of the post-selection pools. Using this pool as an example, we determined how step size affects the network connectivity, i.e., the number of pairs of peaks joined by at least one pathway (*pn*). As the step size increases, *pn* increases toward a maximum, which occurs when all peaks are connected to each other. This can be seen in Fig. 1, which shows that there are only 6 connected pairs when the step size is low, but at larger step size values, *pn* rapidly increases toward the maximum. This behavior can be related to a percolation threshold phase change in the evolutionary network.

One question that may arise is whether the pathways are statistically significant or not, compared to what one would expect by chance. One way to approach this is the following: Consider all sequences in the post-selection distribution that do not belong to a peak. For each of those sequences, generate a random sequence of the same length. Then replace all the actual sequences that lie outside of peaks with the randomly generated sequences. Next, carry out the algorithm described above to discover the potential pathways. Repeat this procedure many times in order to compute the probability that a random pathway is found between two given peaks. If the probability is too low (in terms of *p*-value or any other statistical measure), then the pathway found in the actual post-selection data is statistically significant. It is very likely, for a functional selection, that the statistical significance tested in this way will be quite high for a pathway that was detected in the data, since the absolute number of unique sequences resulting from the selection is likely to be very low compared to a percolation threshold for the network [35]. Thus the chance that a pathway is discovered by chance is very low, at least for low step sizes. Alternatively, if the data are available, another test of whether the detected pathway is real is whether it is reproducible in a replicate data set. Finally, other clues may be used, such as whether a biochemical explanation for the pathway exists (e.g., presence of a conserved motif).

2.4. Practical implementation

We implemented our methods using the Galaxy bioinformatics platform [29]. Before using our tools, any paired end FASTQ data should be joined and groomed, and any reverse complements should be converted to align with the expected sequencing output. The workflow consists of the following tools.

Tool 1. **Extractor**. The Extractor tool extracts desired sequences from a standard Illumina output (FASTQ) file (Fig. 2). It recognizes desired sequences based on a defined 3′-tag appearing in the relevant sequences. The 3′-tag is also removed (5′-tags are left on the sequences, but can be removed by the 5′-Extractor tool). Sequences containing motifs which are suspected to be contaminants, derivatives of tag sequences, etc., can also be removed. The Extractor tool returns a list of raw sequences (not FASTQ format).

Tool 1A. **Format converter**. If desired, the raw sequences (e.g., output from the Extractor) can be converted back into FASTQ format (e.g., if needed as input for the 5′-Extractor tool). This tool interconverts raw sequences, FASTA, and FASTQ files.

Tool 1B. **5′-Extractor**. Like the main Extractor tool, this tool extracts sequences based on the 5′-tag sequence, removes the 5′ tag, and returns the list of raw sequences.

Tool 2. **Counter.** The Counter tool returns a list of unique sequences including the copy number (uncorrected) (Fig. 3). It also returns the total number of unique sequences and the total number of sequence reads. The output file can now be used as input for the Corrector, Basic Statistics, or Landscape Constructor tools.

Tool 3. **Corrector.** The Corrector tool applies three possible corrections to the data, as desired, to account for sequencing errors, ligation and reverse transcription biases, and biases in the synthesis of the initial pool ('fitness correction') (Fig. 4). Any combination of these corrections may be applied. Note that the first correction requires knowledge of the error rate of sequencing. Note that the second and third corrections require input of the counted sequence data from the pre-selection pool. Use of this tool is optional if no corrections are desired.

Tool 4. **Landscape constructor.** The Landscape Constructor Tool accepts the file previously output by the Counter or Corrector tool. The Landscape Constructor clusters the sequences according to relatedness and determines the presence of evolutionary pathways (Fig. 5). It takes a counted (and optionally corrected) list of sequences and makes pairwise comparisons between each set of two sequences. If one sequence can be converted into the other within a specific number of insertions, deletions, or substitutions, they are part of the same cluster (or fitness peak). Within a cluster, the sequences are sorted by decreasing (corrected) copy number. This tool also performs basic network analysis on the landscape to determine various measures of peak connectivity. The tool calculates the clustered list of sequences with their fitness, the histogram of fitness, the distances among members of the same cluster, the distances between cluster centers, and potential evolutionary pathways between clusters. Note that several parameters can be adjusted, including the definition of clusters, Hamming vs. edit distance, and the number of edits allowed in one step of the evolutionary pathway.

Tool 5. **Basic statistics.** The output from the Counter tool can also be used for basic characterization of the sequencing data. This is useful as a quick check of the randomness and quality of the pool over time and the progress of the selection. The Basic Statistics tool gives a count of *n*-mers within the pool ($n = 1$ to 7), a histogram of sequence counts, a histogram of lengths, the frequency of each monomer at each position, and the frequency of each dimer sequence (16 possible) at each position.

Please cite this article in press as: R. Xulvi-Brunet et al., Methods (2016), http://dx.doi.org/10.1016/j.ymeth.2016.05.012

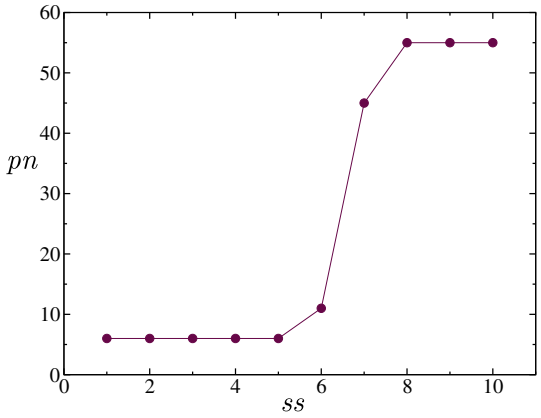

Fig. 1. Number of pairs of peaks joined by at least one pathway (pn) as a function of the step size (ss). Note that, since these data represent 11 peaks, the maximum number of pathways among the peaks is $11 * (11-1)/2 = 55$. These data are taken from Replicate 1 of the experiment described previously [15]. Note that peaks and pathways in the analysis here were defined according to a Hamming distance metric, for computational expediency while performing multiple analyses. Analysis based on Hamming distance and edit distance both identify major peaks, but analysis based on Hamming distance is less sensitive to minor peaks.

3. Discussion

3.1. The definition of a fitness peak

Our operational definition of peak requires a few assumptions. We use 'cluster' and 'peak' interchangeably when discussing complete fitness landscapes. However, in the case of a landscape which cannot be completely explored *in vitro* (e.g., having more than 30 random sites), sequences may cluster without revealing the entire peak. In this case, 'cluster' may be the preferred term.

A group of neighboring sequences is defined as a peak only if the number of sequences in the group exceeds a certain cutoff value, a number chosen by the researcher. This number may be chosen to minimize spurious clusters, for example. If the group of neighboring sequences is smaller than the cutoff value, it is not classified as a peak. The chosen cutoff may be sufficiently small (e.g., 1), such that a peak is defined as any set of sequences that meet the distance criteria for inclusion in the peak (see below). It is the responsibility of the researcher to choose a reasonable cutoff value, depending on the goals of the study and the statistics of the post-selection distribution.

In addition, a cluster is defined as those sequences that are within a certain cutoff distance (edit or Hamming) from the "summit", which is defined as the highest fitness sequence of that cluster. A small cutoff, for instance, a cutoff distance of 2, will tend to portray fitness landscapes consisting of dense, small peaks that may be in close proximity to each other in the landscape. Conversely, a large cutoff will tend to build fitness landscapes with a small number of sparsely populated, large peaks. Again, it is the responsibility of the researcher to choose a suitable value for the cutoff distance. We have found that the number of identified peaks does not depend strongly on the cutoff distance for a reasonable range of values; beyond a certain point, however, the peaks rapidly merge. Finally, note that this simple peak definition may place two peaks next to each other, such that sometimes they might even

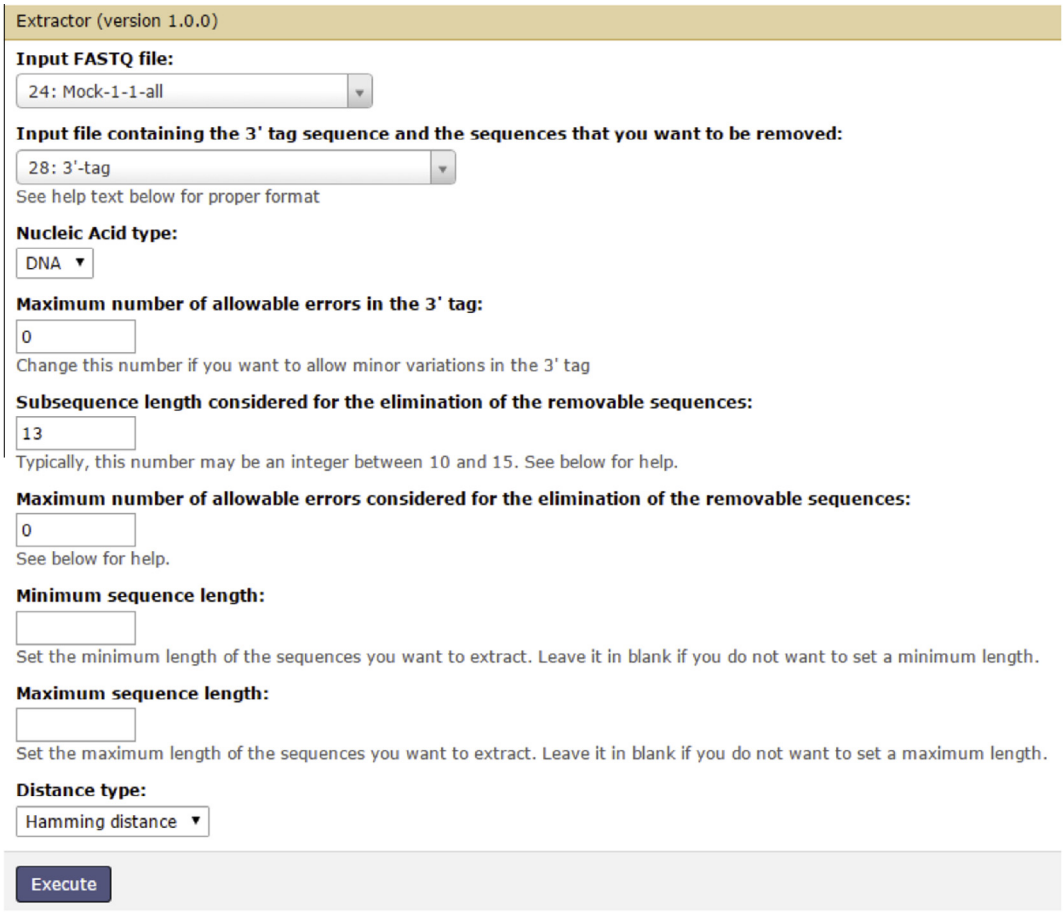

Fig. 2. Extractor screenshot.

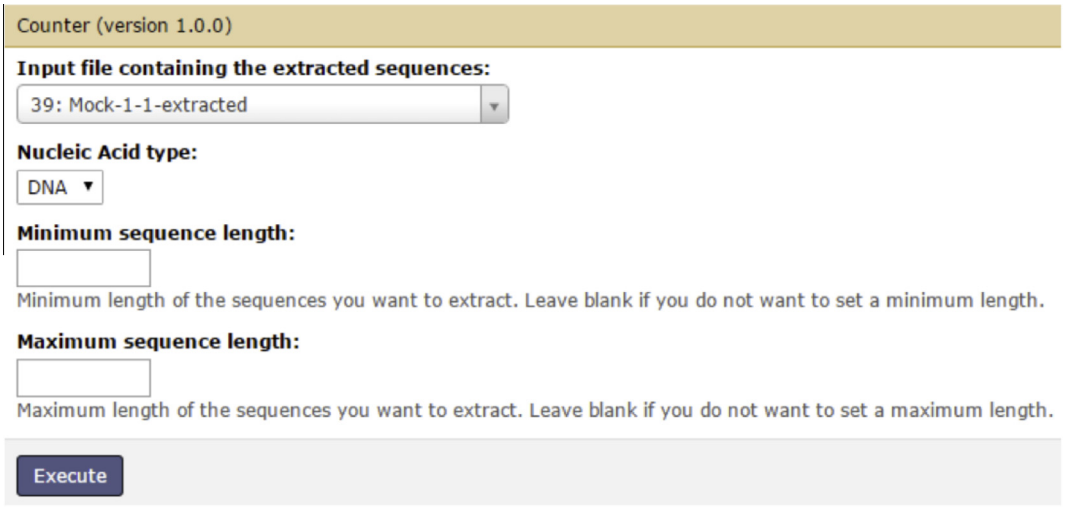

Fig. 3. Counter screenshot.

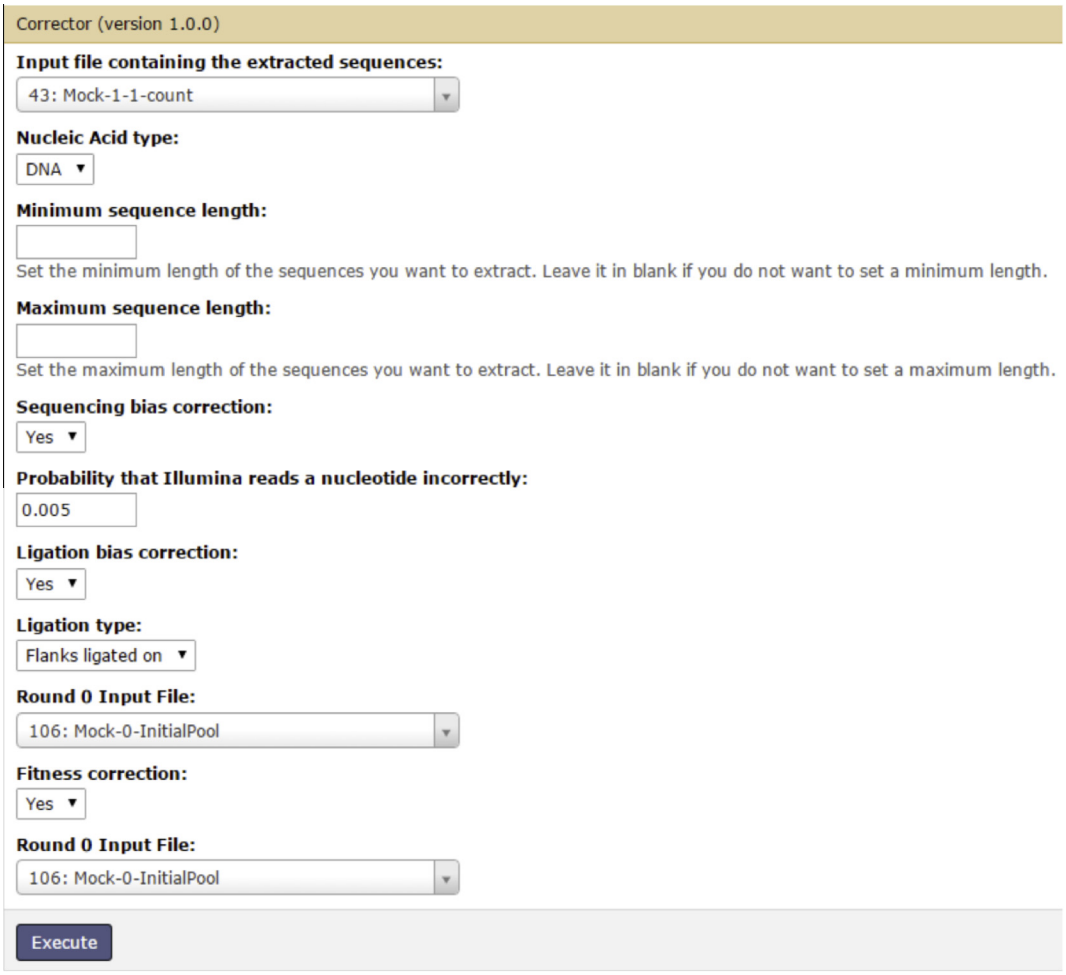

Fig. 4. Corrector screenshot.

share some sequences (in our algorithm for landscape construction, sequences may be part of more than one peak). The likelihood of this scenario can be tuned by modifying the cutoff distance, as mentioned above.

The advantage of defining peaks in this way is that it limits the computations that are required. Considering that calculating edit distances is computationally expensive, this turns out to be an important advantage. The choice of cutoff values may also be evaluated empirically by testing a range of values to ensure that the overall picture of the resulting fitness landscape, including the shape and distribution of peaks, is sufficiently robust and meaningful.

Please cite this article in press as: R. Xulvi-Brunet et al., Methods (2016), http://dx.doi.org/10.1016/j.ymeth.2016.05.012

The alternative to a simple peak definition, such as the one proposed here, is a definition based on sophisticated clustering techniques [36]. Such techniques are usually computationally expensive in and of themselves, certainly more expensive than the simple algorithm discussed above. Also, although some clustering techniques might potentially offer some advantages over our definition, it is worth noting that all clustering techniques also establish some kind of arbitrary cutoff, either regarding the number of clusters that are allowed to be found, or regarding the distances that separate clusters. Establishing those corresponding suitable cutoffs usually entails judgment on the part of the researcher, just as in our more expedient definition of a peak.

We can offer some rough guidelines that may be helpful in determining a suitable cutoff distance. One way to decide the value of this cutoff could be based on how the *number of peaks* varies as a function of the cutoff distance. From the preceding discussion, it is clear that, as the cutoff distance progressively increases, peaks will tend to merge (and become increasingly sparse). Thus, a reasonable cutoff distance *cd* should satisfy the condition that, for distances greater than or equal to *cd*, peaks should not merge for a reasonably long range of cutoff values, i.e., the number of peaks should be relatively constant for some range of cutoff distances.

Fig. 6 shows, for data corresponding to a real selection experiment, how the number of peaks typically varies as a function of the cutoff distance. The picture suggests, concordant with our previous discussion, that a cutoff distance of 3 (or 4) is likely the most reasonable option. Note that there are 13 peaks for cutoff values of 1 and 2 (which are unreasonably small cutoff values, given the possibility of mutations and sequencing errors), but some of the peaks merge when the cutoff increases to 3. The number of peaks remains constant for values between 3 and 8 (inclusive), which is depicted in the figure as a relatively long plateau. Then, starting at a cutoff distance of 9, peaks continue to merge as the cutoff increases.

3.2. Reproducibility of the reconstructed fitness landscape

If the goal of an *in vitro* selection is to generate molecules with a given function, reproducibility of the selection and the generated fitness landscape is not a primary concern, as any individual candidate sequence will be tested for activity. Or, if only a sparse sample of sequence space was taken, then reproducibility in a replicate experiment is not necessarily expected, since a second preselection pool is unlikely to contain substantial overlap in sequences with the first. However, if high coverage of sequence space was expected, and constructing the fitness landscape was a goal of the experiment, then a replicate experiment should yield similar results. The utility of a replicate is twofold. First, it establishes the reproducibility of the selection itself, which is particularly useful when fitness is understood to be survival through the selection (which is affected by multiple factors), as opposed to molecular activity. Second, a replicate aids in characterizing the level of experimental noise, which affects the identification of peaks and evolutionary pathways. The level of noise essentially determines the fitness threshold for reliable detection. A comparison of replicates will show the conservation of major, high-fitness peaks, but lower-fitness peaks may not be found in both replicates. While sequences falling below this transition may be biochemically significant, their detection is unreliable. One may think of this threshold fitness as a 'sea level' below which a fitness peak or evolutionary pathway cannot be reliably seen. The biochemical activity level to which this fitness threshold corresponds may not be knowable in advance. In addition, whether reproducibility in replicates can be observed also depends on the topology of the fitness landscape. If the selection yields a very flat landscape with a large number of peaks of similar fitness, it is

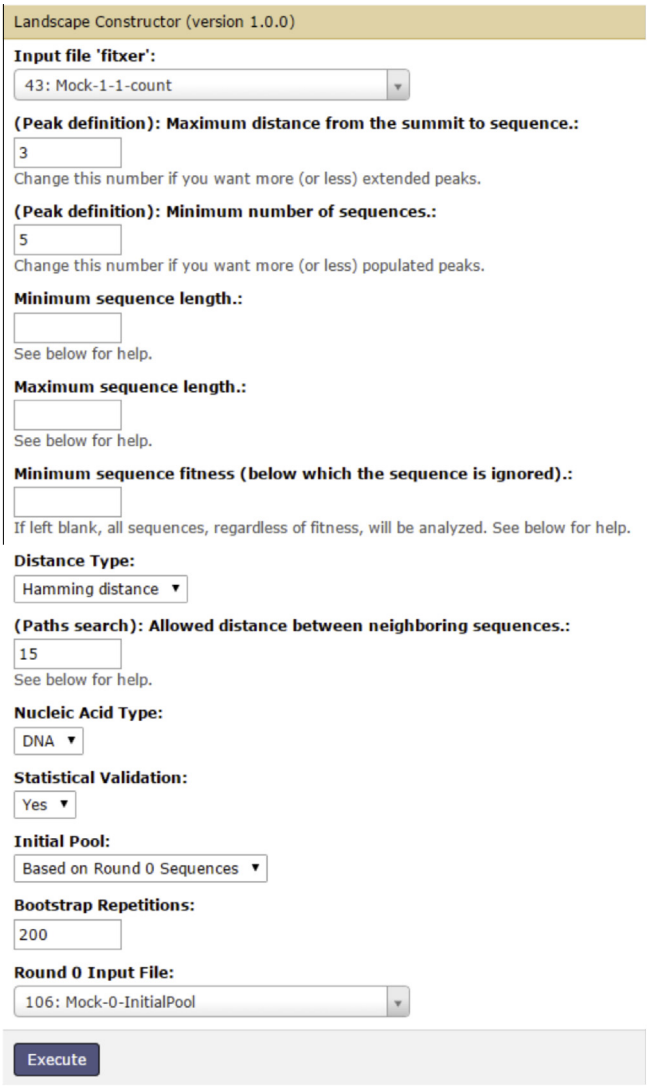

Fig. 5. Landscape constructor screenshot.

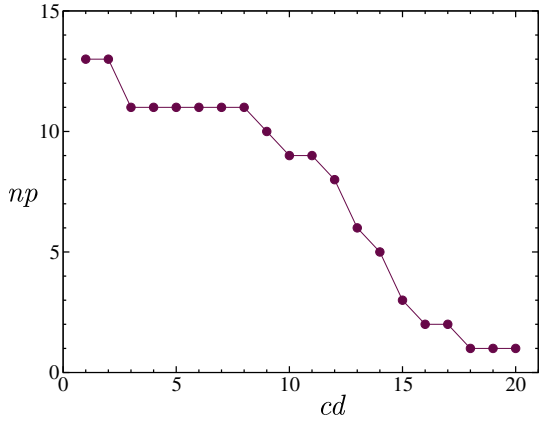

Fig. 6. Number of peaks (*np*) as a function of the cutoff distance (*cd*). The minimum number of sequences needed to define a peak was 5, and the distance definition was the Hamming distance. The data correspond to the data used in Fig. 1. The behavior of the curve is robust to increases in the minimum number of sequences used to define a peak (for reasonable values) or the type of distance (Hamming vs. edit). Note the plateau between $cd = 3$ and $cd = 8$.

possible that HTS data will provide only a subsample of the fitness peaks, such that a second replicate will reveal a different subsample of the peaks. In this situation, the sequences in the post-selection pool may be identical in the two replicates (e.g., with $> 10^{10}$ sequences), but their identity cannot be determined because of limitations on HTS sequencing ($< 10^{10}$ reads). If this should occur, the stringency of the selection may be increased to differentiate among the fitness of these peaks, if experimental replication of the fitness landscape is important.

3.3. Graphical representation of the fitness landscape

The number of dimensions (i.e., the number of independent variables) in a fitness landscape in sequence space is determined by the length of the sequences. Since the sequence length in a selection experiment is greater than two or three, it is impossible to plot a landscape in a Euclidean space of dimension two or three. Representation of the landscape therefore requires special consideration. Several groups have already addressed this important question [37,13,14] and have developed different computational tools, usually based on principle components analysis, to graphically represent appropriate aspects of the fitness landscape. However, when the landscape includes many different, essentially unrelated, peaks (e.g., [15]), it is not represented well by a small number of principal components. We used a customized network representation in this case. Depending on the goal, similar representations may be created using the online resource Cytoscape (www.cytoscape.org) [38].

3.4. Comment on evolutionary fitness and biochemical activity

Selection experiments are usually intended to discover RNA (or DNA) sequences with a certain biochemical activity. It is sometimes assumed that the fittest sequences, i.e., those achieving highest abundance and/or highest rate of increase, are also the most biochemically active (e.g., having the highest affinity or catalytic rate). Whether fitness correlates well with activity in practice depends on the experimental parameters. We have focused on fitness landscapes, but one may also conceptualize an analogous landscape for biochemical activity (an 'activity landscape'). If it is of specific interest to use the fitness landscape to construct the activity landscape, the extent to which the fitness landscape (which is measurable through sequencing) corresponds to the activity landscape should be evaluated. For example, several exemplar sequences could be identified and tested individually for activity to understand the correlation between the biochemical measurement and the evolutionary fitness. Indeed, the corrections we have described here are intended to improve this correlation by removing knowable biases in the fitness measurement. Furthermore, we suggest that methods of estimating fitness based on the rates of increase may improve upon those based on abundance alone.

3.5. Recent improvements to the fitness landscape algorithms

We previously developed algorithms in FORTRAN to carry out the computational tasks described here [15,16]. Recently we translated these algorithms into a suite of computational tools on the Galaxy bioinformatics platform, primarily to improve accessibility to potential users. The new format is web-based and user-friendly. The updated tools also integrate a number of improvements. These include the ability to analyze long sequences (up to 1000 nucleotides), to alter the step size in the discovery of evolutionary pathways, to turn on and off options as needed for a custom experiment, and several improvements that result in faster runtimes.

Researchers interested in using these tools may visit http://galaxy-chen.cnsi.ucsb.edu:8080/ or contact us for more information.

4. Conclusion

The robust re-construction of fitness landscapes from data acquired from *in vitro* evolution experiments is a topic of fundamental interest to both evolutionary biologists and synthetic biologists seeking new functional molecules. Several technical challenges exist in the process of mapping fitness landscapes. Overcoming these challenges requires creating quantitative models to accurately describe the synthesis of nucleic acids pools and characterize the ways in which necessary experimental protocols distort the estimation of molecular distributions. These models may be used as the basis for correcting for these unwanted effects. Most computational time, however, is devoted to construction of the landscape and determination of evolutionary pathways. Finally, we detail an expansion and implementation of our methods in the online, user-friendly Galaxy platform.

Acknowledgements

We acknowledge support from the Center for Scientific Computing at the CNSI and MRL: an NSF MRSEC (DMR-1121053) and NSF CNS-0960316. The authors thank the Simons Foundation (Grant No. 290356), the Searle Scholars Program, the Hellman Fellows Fund, and the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. R. Xulvi-Brunet expresses his gratitude to the Prometeo Project of the Ministry of Education, Science, Technology, and Innovation of Ecuador for the support of this project. Sudha Rajamani is grateful to IISER Pune for financial support. J. Jiménez acknowledges the support of BBSRC grant BB/M009769/1.

References

[1] A.D. Ellington, J.W. Szostak, In vitro selection of RNA molecules that bind specific ligands, Nature 346 (6287) (1990) 818–822, http://dx.doi.org/10.1038/346818a0.
[2] C. Tuerk, L. Gold, Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase, Science 249 (4968) (1990) 505–510, http://dx.doi.org/10.1126/science.2200121.
[3] D.L. Robertson, G.F. Joyce, Selection in vitro of an RNA enzyme that specifically cleaves single-stranded DNA, Nature 344 (6265) (1990) 467–468, http://dx.doi.org/10.1038/344467a0.
[4] S. Klussmann, The Aptamer Handbook: Functional Oligonucleotides and their Applications, Wiley-VCH (2006), http://dx.doi.org/10.1002/3527608192.
[5] R.L. Strack, M.D. Disney, S.R. Jaffrey, A superfolding Spinach2 reveals the dynamic nature of trinucleotide repeat-containing RNA, Nat. Methods 10 (12) (2013) 1219–1924, http://dx.doi.org/10.1038/Nmeth.2701.
[6] G.F. Dolan, A. Akoopie, U.F. Muller, A faster triphosphorylation ribozyme, PLoS ONE 10 (11) (2015) e0142559, http://dx.doi.org/10.1371/journal.pone.0142559.
[7] L. Chan, K. Tram, R. Gysbers, J. Gu, Y. Li, Sequence mutation and structural alteration transform a noncatalytic DNA sequence into an efficient RNA-cleaving DNAzyme, J. Mol. Evol. (2015) 1–9, http://dx.doi.org/10.1007/s00239-015-9712-x.
[8] K. Sefah, J.A. Phillips, X. Xiong, L. Meng, D. Van Simaeys, H. Chen, J. Martin, W. Tan, Nucleic acid aptamers for biosensors and bio-analytical applications, The Analyst 134 (2009) 1765–1775, http://dx.doi.org/10.1039/b905609m.
[9] L. Martini, A.J. Meyer, J.W. Ellefson, J.N. Milligan, M. Forlin, A.D. Ellington, S.S. Mansy, In vitro selection for small-molecule-triggered strand displacement and riboswitch activity, ACS Synth. Biol. 4 (2015) 1144–1150, http://dx.doi.org/10.1021/acssynbio.5b00054.
[10] S. Wright, The roles of mutation, inbreeding, crossbreeding, and selection in evolution, Proc. Sixth Int. Congress Genet. 1 (1932) 356.
[11] J. Maynard Smith, Natural selection and the concept of a protein space, Nature 255 (1970) 563, http://dx.doi.org/10.1038/225563a0.

[12] S. Kauffman, S. Levin, Towards a general theory of adaptive walks on rugged landscapes, J. Theor. Biol. 128 (1) (1987) 11–45, http://dx.doi.org/10.1016/S0022-5193(87)80029-2.
[13] J.N. Pitt, A.R. Ferré-D'Amaré, Rapid construction of empirical RNA fitness landscapes, Science 330 (2010) 376, http://dx.doi.org/10.1126/Science.1192001.
[14] J.N. Pitt, I. Rajapakse, A.R. Ferré-D'Amaré, SEWAL: an open-source platform for next-generation sequence analysis and visualization, Nucleic Acids Res. 38 (22) (2010) 7908–7915, http://dx.doi.org/10.1093/nar/gkq661.
[15] J.I. Jiménez, R. Xulvi-Brunet, G.W. Campbell, R. Turk-MacLeod, I.A. Chen, Comprehensive experimental fitness landscape and evolutionary network for small RNA, Proc. Nat. Acad. Sci. USA 110 (2013) 14984–14989, http://dx.doi.org/10.1073/pnas.1307604110.
[16] R. Xulvi-Brunet, G.W. Campbell, S. Rajamani, J.I. Jiménez, I.A. Chen, Quantitative analysis of synthesized nucleic acid pools, in: J. Carballido-Landeira, B. Escribano (Eds.), Nonlinear dynamics in Biological Systems, Sema Simai Springer Series, vol. 7, 2016 (Chapter 2).
[17] K.P. Burnham, D.R. Anderson, Model Selection and Multimodel Inference: A Practical Information-theoretic Approach, Springer, 2002. ISBN: 0-387-95364-7.
[18] A.W.F. Edwards, Likelihood, Expanded ed., Johns Hopkins University Press, 1992.
[19] T.E. England, O.C. Uhlenbeck, Enzymatic oligoribonucleotide synthesis with T4 RNA ligase, Biochemistry 17 (11) (1978) 2069–2076.
[20] T. Middleton, W.C. Herlihy, P.R. Schimmel, H.N. Munro, Synthesis and purification of oligoribonucleotides using T4 RNA ligase and reverse-phase chromatography, Anal. Biochem. 144 (1) (1985) 110–117.
[21] S.G. Acinas, R. Sarma-Rupavtarm, V. Klepac-Ceraj, M.F. Polz, PCR-induced sequence artifacts and bias: insights from comparison of two 16S rRNA clone libraries constructed from the same sample, Appl. Environ. Microbiol. 71 (12) (2005) 8966–8969, http://dx.doi.org/10.1128/AEM.71.12.8966-8969.2005.
[22] F. Meacham, D. Boffelli, J. Dhahbi, D.I.K. Martin, M. Singer, L. Pachter, Identification and correction of systematic error in high-throughput sequence data, MBC Bioinf. 12 (2011) 451, http://dx.doi.org/10.1186/1471-2105-12-451.
[23] K. Nakamura, T. Oshima, T. Morimoto, S. Ikeda, H. Yoshikawa, Y. Shiwa, S. Ishikawa, M.C. Linak, A. Hirai, H. Takahashi, Md. Altaf-Ul-Amin, N. Ogasawara, S. Kanaya, Sequence-specific error profile of Illumina sequencers, Nucl. Acids Res. 39 (13) (2011) e90, http://dx.doi.org/10.1093/nar/gkr344.
[24] J.C. Dohm, C. Lottaz, T. Borodina, H. Himmelbauer, Substantial biases in ultra-short read data sets from high-throughput DNA sequencing, Nucl. Acids Res. 36 (16) (2008) e105, http://dx.doi.org/10.1093/nar/gkn425.
[25] M. Schirmer, U.Z. Ijaz, R. D'Amore, N. Hall, W.T. Sloan, C. Quince, Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform, Nucl. Acids Res. 43 (6) (2015) e37, http://dx.doi.org/10.1093/nar/gku1341.
[26] S. Alon, F. Vigneault, S. Eminaga, D.C. Christodoulou, J.G. Seidman, G.M. Church, E. Eisenberg, Barcoding bias in high-throughput multiplex sequencing of miRNA, Genome Res. 21 (9) (2011) 1506–1511, http://dx.doi.org/10.1101/gr.121715.111.
[27] M. Hafner, N. Renwick, M. Brown, A. Mihailović, D. Holoch, C. Lin, J.T. Pena, J.D. Nusbaum, P. Morozov, J. Ludwig, T. Ojo, S. Luo, G. Schroth, T. Tuschl, textitRNA-ligase-dependent biases in miRNA representation in deep-sequenced small RNA cDNA libraries, RNA 17 (9) (2011) 1697–1712, http://dx.doi.org/10.1261/rna.2799511.
[28] W.H. Thiel, T. Bair, K.W. Thiel, J.P. Dassie, W.M. Rockey, C.A. Howell, X.Y. Liu, A. J. Dupuy, L. Huang, R. Owczarzy, M.A. Behlke, J.O. McNamara, P.H. Giangrande, Nucleotide bias observed with a short SELEX RNA aptamer library, Nucleic Acid Ther. 21 (4) (2011) 253–263, http://dx.doi.org/10.1089/nat.2011.0288.
[29] B. Giardine, C. Riemer, R.C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor, W. Miller, W.J. Kent, A. Nekrutenko, Galaxy: a platform for interactive large-scale genome analysis, Genome Res. 15 (10) (2005) 1451–1455, http://dx.doi.org/10.1101/gr.4086505.
[30] R.W. Hamming, Error detecting and error correcting codes, Bell Syst. Tech. J. 29 (2) (1950) 147–160, http://dx.doi.org/10.1002/j.1538-7305.1950.tb00463.x.
[31] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Sov. Phys. Doklady 10 (8) (1966) 707–710.
[32] G. Navarro, A guided tour to approximate string matching, ACM Comput. Surv. 33 (1) (2001) 31–88, http://dx.doi.org/10.1145/375360.375365.
[33] Y. Vardi, Network tomography: estimating source-destination traffic intensities from link data, J. Am. Stat. Assoc. 91 (433) (1996) 365–377, http://dx.doi.org/10.1080/01621459.1996.10476697.
[34] R. Xulvi-Brunet, W. Pietsch, I.M. Sokolov, Correlations in scale-free networks: tomography and percolation, Phys. Rev. E 68 (2003) 036119, http://dx.doi.org/10.1103/PhysRevE.68.036119.
[35] S. Gavrilets, J. Gravner, Percolation on the fitness hypercube and the evolution of reproductive isolation, J. Theor. Biol. 184 (1) (1997) 51–64, http://dx.doi.org/10.1006/jtbi.1996.0242.
[36] B.S. Everitt, S. Landau, M. Leese, D. Stahl, Cluster Analysis, fifth ed., Wiley, 2011. Wiley series in probability and statistics.
[37] D.M. McCandlish, Visualizing fitness landscapes, Evolution 65 (6) (2011) 1544–1558, http://dx.doi.org/10.1111/j.1558-5646.2011.01236.x.
[38] P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, Cytoscape: a software environment for integrated models of biomolecular interaction networks, Genome Res. 13 (11) (2003) 2498–2504, http://dx.doi.org/10.1101/gr.1239303.

Please cite this article in press as: R. Xulvi-Brunet et al., Methods (2016), http://dx.doi.org/10.1016/j.ymeth.2016.05.012