Supplementary Information to
“General Mechanism of Evolution Shared by Proteins and Words”

Contents

I. DETAILS ABOUT THE FIRST GLC

The goal of genetics-linguistics correspondence (GLC) is to quantify the universality shared by genetics and linguistics. Such a correspondence is built on the choice of unit that performs both generality and functionality. In the following, we will discuss some possible candidates besides what was introduced in the main text.

A. Order-of-magnitude estimate

To quantify the universality, a good starting point is order-of-magnitude estimate. A number written in scientific notation is $a \times 10^b$ where $1 \leq |a| < 10$ and the integer b denotes the order of magnitude. For example, the order of magnitude of $4123 = 4.123 \times 10^3$ is 3. Order of magnitude plays an important role to estimate the number of possible combinations for a group of elements.

Let us demonstrate this point of view. There are two people wanting to communicate via a sentence which contains 3 words. Assume this sentence is composed by selecting words from two different sets, either U or U'. For the first set U with 4 different elements (its size is zero order of magnitude), the number of possible sentences would be $4^3 = 64$. For the second set U' with 20 different elements (its size is one order of magnitude), the number of possible sentences becomes $20^3 = 8000$. In other words, the size of a set, i.e., cardinality, will greatly affect the complexity of creating a sentence.

Assume that two people reach a consensus by using identical sentence. If two people randomly create their own sentence with U, the probability that two sentences happen to be identical is $p(U) = (1/64)^2 \approx 2.4 \times 10^{-4}$. Whiles, the probability with U' would become $p(U') = (1/8000)^2 \approx 1.6 \times 10^{-8}$. Because the probabilities $p(U)$ and $p(U')$ are too small, it is impossible to verify all random generated sentences. To improve this situation, they may make an agreement that a consensus will be reached as long as their sentences are “similar”. Then it is crucial to introduce “the rules of formation” to regulate the composition of sentences. If they chose 10 sentences to be similar, the rule of formation for U must cut down at least $64 - 10 = 54$ kinds of sentences, while the rule for U' has to eliminate at least $8000 - 10 = 7990$ kinds. Therefore, the difference of cardinalities in order of magnitude will cause these two sentences to develop distinguish rules of formation.

B. Role of gene

Gene is a sequence of nucleotides in DNA that contains hereditary information to produce RNA, while RNA can be translated into protein. However, a single gene may encode multiple proteins due to alternative splicing[1, 2]. This characteristic makes the analysis of functions difficult. As the first attempt to build a general mechanism of evolution, we thus adopt protein sequence instead of gene to build GLC. Theoretically, it is possible to write down gene-Book, as in Eq. (7) in the main text. However, the inventories of block $\mathcal{B}$ and function $\mathcal{F}$ are not well defined here. We still do not know what exactly component and block in gene-Book are. To find out, one need to check the quantitative characteristics of newly defined units. This problem may leave for future research.

C. Alphabet is not elementary for the first GLC

Most languages have spoken form, but many of them do not have written form. Therefore, when deciding the basic elements of languages for GLC, we must consider the properties of spoken form. To connect written to spoken form, phonetic transcriptions can be used to find the phonemic inventories of writing systems. The basic unit of writing system is grapheme, while alphabet is the collection of graphemes for phonogram. Even though we only discuss the written form, alphabet cannot be the elementary set for all writing systems because it is solely for phonogram. The generality of alphabet in linguistics is much narrower than that of standard amino acid in genetics, thus losing its qualification to be the elementary set for the first GLC.

Based on the order-of-magnitude estimate, grapheme is also not a valid unit to establish first GLC in written language due to the large difference in its cardinalities among different languages. For example, there are more than 40000 kinds of character[3] in Chinese, but only 26 kinds of letters in English. Such quantitative difference makes it impossible to follow the traditional analysis - consider grapheme as the basic unit of written system - to construct GLC from written to spoken form. That is why we need to introduce syllagram to establish the GLC. We notice that, though there are 40000 kinds of character in Chinese, their pronunciation is composed of only 25 phonemes[4, 5]. This fact inspires us to construct GLC from spoken to written form.

We realize that phoneme can quantitatively establish the generality for spoken language because the cardinalities of phonemic inventories for world languages[6], such as English and Chinese, are indeed one order of magnitude[7]. Some languages with oversize phonemic inventories, like the number of phonemes in !Kung is 141[7], can be treated as exceptions because they have not endured the test of large users. Now we can use phonetic transcriptions to connect written with spoken form and make phoneme become a valid unit to perform the generality of language. Since the number of distinct phonemes and that of standard amino acids are both one order of magnitude, the first GLC can be established.

II. PREREQUISITE OF RANK-RANK ANALYSIS

In METHOD of the main text, we have introduced the tools to segment blocks and components. Here we will show how to construct the genome into text and what kinds of corpora are used in our work.

A. Life-Book

To apply linguistic methods on the genome and turn it into a protein sequence that contains information of composed domains, we need to construct a “Life-Book”. Our genome database is retrieved from Ensembl BioMart[23]. To create Book, you need to follow the below steps: First, choose your desired database. Second, select “Transcript stable ID” and “Interpro ID” as the attributes. The former corresponds to proteins, while the latter to domains. Third, click the “count” button. Forth, click the “result” button and export all results to XLS file.

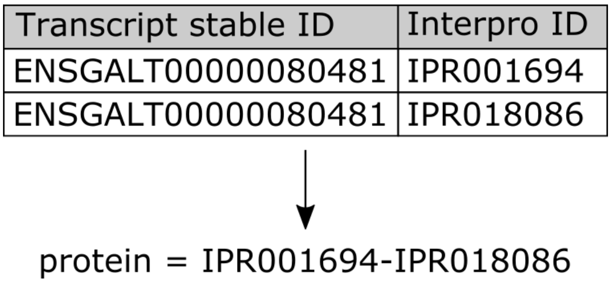
After download the excel file, our program can transform the XLS/XLSX file into Life-Book by representing protein as its composed domains as following:

$$\text{Book} = s_1\ s_2\ s_3\ ... \tag{1}$$

where the protein $s_j = I_{j1} - I_{j2} - ...$ and I_{jk} denotes the Interpro ID.

Note that we did not select the attributes of “Interpro start” and “Interpro end” which indicate the start and end of the domain in terms of the position of amino acid on the Ensembl peptide. It is because there are some situations that make the analysis of Life-Book difficult: (i) A domain is identified as a part of 3-D protein, its Interpro start and end may be discrete on the 1-D amino acid sequence. (ii) Interpro may use different models or experimental data to identify the position of a domain so that multiple choices of Interpro start and end are recorded in BioMart. (iii) The protein may exhibit a form, like A-B-A, in which the same domain (based on its structure, not function) repeats itself such that there are several Interpro starts and ends for one Interpro ID within a protein. It is hard to distinguish these situations when dealing with huge data on BioMart. As a result, the repeated Interpro IDs are treated as one ID if one selects Interpro start and end as the attributes. To save trouble, we do not suggest such attributes at the beginning.

A schematic for how our program constructs a Life-Book is exhibited in Fig. 1. For details of program, please refer to Ref. [24]. Genomes of several organisms are investigated, part of which are listed in Tab. I. All genomes used in this work are collected in Supplementary Data.



Transcript stable ID	Interpro ID
ENSGALT00000080481	IPR001694
ENSGALT00000080481	IPR018086

protein = IPR001694-IPR018086

FIG. 1: Schematic of how we construct Life-Book from data of biomart.

TABLE I: Statistical quantities of different organisms. Readers can find the original data by searching the key word in Reference column on Ensembl BioMart[23]. The FRD of all life-Books obey Zipf's law.

No.	Sample	Zipf b	r_g	SC	U_{Chain}	V_1	L	Reference
1	Alpaca	0.487	0.664	0.898	964	5407	6934	Alpaca
2	Stickleback	0.595	0.704	0.825	844	3577	6443	BROAD S1
3	Horse	0.614	0.732	0.508	596	2593	6189	EquCab3.0
4	Domestic cat	0.538	0.743	0.792	606	2667	5679	Felis_catus_9.0
5	Medium ground-finch	0.51	0.716	0.810	1029	3062	5396	GeoFor_1.0
6	Human	0.610	0.798	0.833	608	2468	7089	Human
7	Ciona intestinalis	0.57	0.700	0.881	686	4885	7851	KH
8	Prairie vole	0.521	0.672	0.840	1117	3586	5980	MicOch1.0
9	Northern white-cheeked gibbon	0.53	0.698	0.832	679	3325	6291	Nleu_3.0
10	Oryzias javanicus	0.58	0.819	0.642	664	2224	5677	OJAV_1.1
11	Chinook salmon	0.55	0.827	0.533	528	1969	5164	Otsh_v1.0
12	Chimpanzee	0.601	0.726	0.788	554	3009	6277	Pan_tro_3.0
13	Platyfish	0.569	0.731	0.692	796	2600	5379	Platyfish
14	Turkey	0.51	0.760	0.844	764	3050	5999	Turkey_5.1
15	Polar bear	0.581	0.730	0.718	692	2904	5858	UrsMar_1.0
16	Chicken	0.51	0.709	0.704	901	2798	5557	chicken
17	European seabass	0.55	0.845	0.590	576	1743	4947	seabass_V1.0
18	Zebrafish	0.565	0.715	0.906	747	3006	6235	zebrafish

B. Natural and artificial text

Besides Life-Book, we have also studied both real (natural text) and fake (artificial text) corpora in this work. Part of them are listed in Tab. II. All the references of real text can be found in Supplementary Data.

We are curious about what factors is crucial for the scaling structure (SS), such as grammar, types of writing, word formation, and frequency-rank distribution (FRD) of word (see Sec. IV D for conclusion). Relative to gene, it is easier and more convenient to control the factors that generate a text. Thus, we produce three kinds of fake corpora to search the origin of SS:

1. *Paper generator*

Paper generator is used to check the role of meaning of sentences in SS, while the sentences obey the grammar. We use SCIgen[26] to generate Computer-Science research papers consist of meaningless sentences. It applies context-free grammar to form all elements of the papers.

2. *N-gram generator*

N-gram generator is used to examine whether word formation and FRD of word affect SS. Based on Sinica Corpus[27], the N-gram generator[24] groups N random Chinese characters into one word. After building the word inventory, the user can arrange for FRD of these words to obey Zipf's or other distributions such as Gaussian. In addition, N-gram generator can constrain the supremum

$$U_{Chain} \equiv \sup\{Chain(r_b) \mid \vec{r} = (r_b, r_c) \in \mathrm{RRD}\} \tag{2}$$

where the definition of *Chain* function and RRD can be found in Eq. (8), the main text, and Sec. III.

To understand how N-gram generator works, let us give an example. For a 2-gram generator, it can group two unrelated Chinese characters, like 江 (means river) and 辣 (means spicy), into one fake word 江辣 (meaningless). In fact, using Chinese character as the syllagram is not necessary for checking the importance of word formation. This is because we aim to verify whether a text that is composed of fake words will destroy SS. Similar operation can also be done for given English syllagrams.

3. Mixed text

This is aimed to test whether mixing different writing styles will affect SS. We generate two different mixed texts: Newspaper[28] and Mix[29–31] (by various authors). The former is the collection of China Times, Apple Daily, Liberty Times, etc. over the period of 2016∼2017 (same writing styles and different authors), while the latter is the combination of three books (both writing styles and authors are different).

4. Excerpts

Excerpts of real texts are used to examine the role of book length L and size of word inventory V_1 in SS. In our work, we make excerpts of Mandarin Chinese novel, Frog, and English novel, Moby-Dick, with $L \sim 1300$.

III. DETAILS ABOUT THE SECOND AND THIRD GLC

The keys to build the second GLC are syllagram and rank-rank analysis. In this section, we describe more details about them.

A. Advantage to syllagram

The reason why we use syllagram instead of syllable to build GLC is explained in the main text. Besides, there is a major advantage to introduce syllagram: it is standardized. When deciding syllables in a word, the techniques and standards adopted to deal with speech data vary with different laboratories. By contrast, many written texts have been digitalized by character encodings, e.g., unicode and UTF-8, that make the determination of syllagram more convenient and standardized. As a result, syllagram is better for quantitative analysis.

We have to note that syllagrams are different from homonyms (words with same spelling and pronunciation, but different meaning), homophones (words with same pronunciation, but different meaning), or homographs (words with same spelling and pronunciation, but different meaning). First, a syllagram is not a word. Second, a syllagram only cares about the written form of syllables, not meaning or sound. For example, eye and I share the same syllables but different syllagrams.

The rigorous definition of syllagram has been given. The process of how we construct second and third GLCs via statistical evidence can be found in the main text.

B. Features of scaling structure

The fact that rank-rank distribution (RRD) exhibits SS for the genomes of 202 organisms, Mandarin Chinese, and English corpora have been examined in Supplementary Data (we check whether their SC value > 0.7, as defined in Sec. IV C). This leads us to conjecture that such a SS generally exists in genomes and natural language corpora.

One cannot help marveling at this structure which is clearly not accidental. The imminent questions then are

1. What causes this universal phenomenon?
2. How it becomes evidence to construct GLC?

To answer question 1, we had been puzzled for a long time (see Sec. IV D) until we introduced a new concept, partition theory. It is useful for understanding the formation of SS, such as why RRD contains layer by layer, what causes RRD filled with “fog” which refers to the points not at any g_ℓ, and what is essential to retain SS.

The key to explain SS is to include the frequency of block ρ_x and component ρ_y, which are not shown in RRD. In other words, they are hidden information. To do this, we add the vertical V_m and horizontal lines H_n which are used to distinguish different groups that share the same frequency. That is, blocks in $(V_2, V_1]$ share the same frequency, so do blocks in $(V_3, V_2]$, etc. Same for components in $(H_2, H_1]$, $(H_3, H_2]$ and so on. Therefore, the data points $\vec{r} = (r_b, r_c)$ on RRD plot can be decomposed into rectangles

$$\{m, n\} \equiv \{\vec{r} \mid r_b \in (V_{m+1}, V_m], r_c \in (H_{n+1}, H_n]\}. \tag{3}$$

For small rank, namely low frequency components and blocks, there is a frequency-index identity[32] which is useful: the words with rank $x \in (V_{m+1}, V_m]$ exhibit the same frequency $\rho_x = m$, while the syllagrams with rank $y \in (H_{n+1}, H_n]$ also have frequency $\rho_y = n$. Same for proteins and domains in genome.

Once the auxiliary lines V_m and H_n are added, three features of SS appear.

1. There is no "fog" in the regions left of g_1 and between g_1 and g_2 where fog refers to the points not at any g_ℓ.
2. Each pair of V_m and H_n always crosses g_ℓ.
3. The upper left area in each rectangle is always devoid of fog.

The above features can be explained by the following observation. If a component with rank R_c only appears in one kind of block, it will be represented by just one point on RRD. In contrast, if R_c appears in multiple kinds of block, several points on RRD will share the same component. This implies that

$$\rho_y(\vec{R}) = \sum_{\vec{r} \in D_y(R_c)} \digamma(\vec{r})\rho_x(\vec{r}) \tag{4}$$

for the selected data point $\vec{R} = (R_b, R_c)$ where $D_y(R_c) \equiv \{\vec{r} = (r_b, r_c) \mid r_c = R_c\}$ indicates the blocks composed of the component with rank R_c, and $\digamma(\vec{r})$ denotes how many times the component r_c appears in block r_b, e.g., $\digamma((\text{ABA}, \text{A})) = 2$. This equation leads to an inequality that $\rho_x \leq \rho_y$ for any point $\vec{R}$.

Let's now go back to the first feature. The curve g_1 is unique because there is no fog in its neighborhood. We notice that the points on g_1 are all in the rectangles $\{1,1\}, \{2,2\}, ...,$ which imply $\rho_x = \rho_y$ and enforce that the component in rectangle $\{m,m\}$ contains only one point. This explains the lack of fog in the right region of g_1. Should there be any point in the rectangle $\{m,n\}$ where $m > n$, it would imply $\rho_x > \rho_y$ which contradicts $\rho_x \leq \rho_y$. This argument therefore rules out fog in the left region of g_1. What are those (block, component) that locate at g_1? They usually refer to special domains/syllagrams that appear in equally special proteins/words. For instance, (i) HHH domain 9 (IPR041692) just exists in SUPT6H protein (ENSGALT00000000200) for chicken genome (GRCg6a), (ii) fio only shows up in Ba-ruf-fio in Harry Potter 1, and (iii) 胺 (amine) is unique in 三聚氰胺 (Melamine) in 蛙 (Frog, Chinese novel).

The second feature is actually from the stipulation of plotting RRD: when two blocks or components share the same frequency, the one that appears earlier in the text will get a smaller rank. That means the data points must stack from the left bottom corner of each rectangle. If there is a new component added to the same rectangle, which simultaneously implies a new block, it will be placed in the upper right of old points and become a part of g_ℓ. This answers why V_m, H_n, and g_ℓ must cross at one point. Note that no matter whether a component is used in blocks with the same or a different ρ_x, its points will appear on the right of the earliest or frequent one. Associate this characteristic with the second feature, we conclude that the points in a rectangle will only locate at g_ℓ or its right. This is what the third feature describes.

C. Partition theory

So far, we have provided useful insights for the three main features of RRD for life and language. However, the composition of fog and why g_ℓ follows scaling relation are still unclear. To answer them, the concept of "number partition[33]" will come in handy. A partition of positive integer N is a way of writing N as a sum of positive integers. For example, $N = 3$ consists of three partitions $(1,1,1), (2,1)$, and (3). If $\digamma = 1$ in Eq. (4), the partition $(2,1)$ refers to a component with $\rho_y = 3$ that appears in two different blocks with $\rho_x = 2$ and 1, respectively.

In fact, Eq. (4) shows the partition of component with rank R_c when $\digamma = 1$. For $D_y = \{\vec{r}_1, ..., \vec{r}_k\}$ where $\vec{r}_i = (r_{ib}, R_c)$, a partition of $\rho_y(\vec{R})$ is $(\rho_x(\vec{r}_1), ..., \rho_x(\vec{r}_k))$. Now, let us demonstrate how the concept of partition helps understanding the formation of SS. Without loss of generality, we assume $\rho_x(\vec{r}_1) \geq ... \geq \rho_x(\vec{r}_k)$, which implies $r_{1b} \leq ... \leq r_{kb}$. Based on the second feature of SS, the point $\vec{r}_1$ will definitely be a part of g_ℓ. According to Fig. 1 in the main text, it is not hard to discover that g_ℓ in rectangles $\{m,n\}$ will follow the simple relation:

$$\ell = n - m + 1 \tag{5}$$

that leads to $\ell = \rho_y(\vec{R}) - \rho_x(\vec{r}_1) + 1$ when m, n are small enough[32]. Other $\vec{r}_i$ will mostly be in the fog, i.e., join the fog in rectangles $\{\rho_x(\vec{r}_i), \rho_y(\vec{R})\}$, or at g_ℓ by chance. We soon realize that the formation of SS requires a huge number of different partitions, which can be used to fill the scaling lines and fog. Otherwise, SS will be plagued by many vacancies and become imperceptible. For instance, if the components with $\rho_y = 3$ only have one kind of partition

(3), they will not contribute to the fog in $\{1,3\}$ and $\{2,3\}$ rectangles. So if the number of components that own few kinds of partitions is high, the fog will be diluted.

Although phenomenologically argued from simple facts, such as blocks comprise of components, the partition theory is useful for understanding the formation of SS. However, there are still some core issues that cannot be answered by the partition theory: (i) It is limited to $\digamma = 1$ in Eq. (4). (ii) It does not tell us what factor can decide the ingredients of sets D_x and D_y. (iii) Both Eq. (4) and partition theory can neither explain the origin of Eqs. (1, 5, 6) of the main text, nor generate them automatically by any given mechanism. These three issues inspire us to figure out the generalized mechanism of evolution that generates all the features of GLC. But in contrast to partition theory, the generalized mechanism is hard to intuitively explain why we have SS. A full understanding of GLC must contain both theories.

D. Chain and allocation functions

For question 2 in Sec. III B, we use statistical indices to be the evidence for GLC. There are two hidden properties. We realize the data points $\vec{r} = (r_b, r_c)$ on RRD can be grouped into

$$\begin{aligned} D_y(R_c) &\equiv \{\vec{r} = (r_b, r_c) \mid r_c = R_c\} \\ D_x(R_b) &\equiv \{\vec{r} = (r_b, r_c) \mid r_b = R_b\} \end{aligned} \tag{6}$$

where $\vec{R} = (R_b, R_c)$ is the selected point. The set $D_y(R_c)$ indicates the blocks composed of the component with rank R_c, while $D_x(R_b)$ refers to the components used in the block with rank R_b. The two hidden properties can be established through the following two definitions:

$$Allo(R_c) \equiv |D_y(R_c)| \tag{7}$$

$$Chain(R_b) \equiv \sum_{\vec{r} \in D_x(R_b)} Allo(r_c) \tag{8}$$

for $\vec{R} = (R_b, R_c)$ and $\vec{r} = (r_b, r_c)$, where $|D_y(R_c)|$ denotes the cardinality of $D_y(R_c)$. The allocation function $Allo(R_c)$ represents the ability of allocating a component R_c to other blocks, while the chain function $Chain(R_b)$ indicates how a block R_b is linked to other blocks. For examples, (i) if component A appears in blocks AB, AC, and KAD, then $Allo(\text{A}) = 3$; (ii) if component H appears only in either H or HH, then $Allo(\text{H}) = 1$; (iii) if component T appears in both T and TT, then $Allo(\text{T}) = 2$; (iv) $Chain(\text{ABC}) = Allo(\text{A}) + Allo(\text{B}) + Allo(\text{C})$; and (v) $Chain(\text{AA}) = Allo(\text{A})$.

By fitting real data for genomes and corpora, as in Extended Fig. 8, we observe that they satisfy two simple empirical relations (hidden properties):

$$Allo(y') = (-\alpha \ln y' + \beta)^2 \tag{9}$$

$$Chain(x') = -\gamma \ln x' + \omega \tag{10}$$

where $\vec{R}' = (x', y')$ is the new rank-rank vector depending on $(Chain, Allo)$ instead of (ρ_x, ρ_y), and $\alpha, \beta, \gamma, \omega > 0$ are fitting parameters.

Note that the last point at g_1, whose $(Chain, Allo) = (1,1)$, can be approximated as $\vec{R}_0 \approx (V_1, H_1)$. When $\vec{R}$ is transformed to $\vec{R}'$, $\vec{R}_0 \approx \vec{R}'_0$. From the definition of Eqs. (9, 10), we can deduce that $\beta/\alpha \approx \ln H_1$ and $\omega/\gamma \approx \ln V_1$ by taking $y' = H_1$ and $x' = V_1$. The V_1 and H_1 represent not only the boundaries of rectangle $\{1,1\}$, but also the size of block and component inventories in the Book. In Sec. IV B, we will show how this property help us obtain the analytic expressions of $\alpha, \beta, \gamma, \omega$.

For block and component, the $Chain$ and $Allo$ function unveil their hidden relationship, while ρ_x, ρ_y represent their individual properties. Combining these quantitative features together, we now have confidence in proposing that the second GLC is the similarity between domain and syllagram, while the third GLC is that between protein and word. Furthermore, the discovery of D_x and D_y inspire us to construct the multilayer network in the main text, which becomes our fundamental framework to study the characteristics of evolution.

IV. DESIGN STATISTICAL INDICES

To quantitatively describe the second and third GLC, we need to incorporate statistical indices. We use maximum likelihood estimation to find out the distribution for FRD of block ρ_x and component ρ_y. Besides, a new pattern reconstruction technique is created to quantify SS for RRD which is hidden behind a lot of messy points. Based on this technique, we are able to define the soundness-goodness value (SC value) for SS.

A. Maximum likelihood estimation for FRD and network

Instead of least squares estimation, we use maximum likelihood estimation to fit FRD and network degree distribution for statistical reasons[17]. Given a Book (as Eq. (7) of the main text) with length L, we measure the frequency of $s_1, ..., s_L$ and obtain a set of their ranks $X = [1, 2..., V_1]$ based on their frequency $\rho_x(1) > \rho_x(2) > ... > \rho_x(V_1) = 1$ where V_1 is the size of block inventory (total kinds of words/proteins). The normalized condition of probability gives:

$$1 = \sum_{k=1}^{V_1} \frac{\rho_x(k)}{L} = \frac{a}{L} h(V_1, b) \tag{11}$$

where the second equality holds for Zipf's law that $\rho_x(k) = ak^{-b}$ and $h(V_1, b) \equiv \sum_{k=1}^{V_1} k^{-b}$.

The likelihood $\mathcal{L}$ is the conditional probability of observing Book under a given probability distribution with parameter b[17, 18]:

$$\mathcal{L}(b|\text{Book}) = \prod_{k=1}^{V_1} P(x_k|b)^{\rho_x(k)} = L^{-L} \prod_{k=1}^{V_1} \left[\rho_x(k)\right]^{\rho_x(k)} \tag{12}$$

where $P(x_k|b) = \rho_x(k)/L$ denotes the normalized probability. To find the best b that makes theoretical values closest to data, we maximize $\mathcal{L}$[18]. Note that direct maximization is too huge to calculate. Instead, we maximize

$$\ln \mathcal{L} = -L \ln L + \sum_{k=1}^{V_1} \rho_x(k) \ln \rho_x(k). \tag{13}$$

Because the natural log is monotonic increasing, it will not change the finding of best b. The above equation has no simple solution, we can use the Python module, numpy, to do numerical calculation (see Ref. [24] for program).

For Zipf-Mandelbrot law (also named shifted-power law), we switch ρ_x to $\rho_x(k, c) = a(k + c)^{-b}$ and $h(V_1, b)$ to $h(V_1, b, c) = \sum_{k=1}^{V_1} (k + c)^{-b}$ in Eq. (11) then apply maximum likelihood estimation again. The above formula also works for the FRD of components ρ_y and degree distribution $P(d_c)$ of component network G_c.

When multiple models need to be selected, we can compare their AICc value[17]:

$$\text{AICc} = 2\kappa - 2 \ln \mathcal{L} + \frac{2\kappa(\kappa + 1)}{S - \kappa - 1} \tag{14}$$

where κ and S respectively denote the number of estimated parameters and sample size of the model, then choose the one with the smallest AICc, i.e., the least information loss.

B. Curve fit for *Chain* and *Allo*

Because $Chain(x\prime)$ and $Allo(y\prime)$ are not probability distribution and their error distribution are unknown, it is not suitable to use maximum likelihood estimation. As the first attempt, we use the non-linear least square fitting algorithm in the Python module scipy[19] to verify Eq. (9) and (10). The fitting curves are obtained by finding the best parameter vectors $\hat{\theta}_1 = (\alpha, \beta)$ and $\hat{\theta}_2 = (\gamma, \omega)$ that can minimize the following functions:

$$S_1(\hat{\theta}_1) = \sum_i \left[Allo(y\prime_i) - \hat{\mu}_1(\hat{\theta}_1, y\prime_i)\right]^2 \tag{15}$$

$$S_2(\hat{\theta}_2) = \sum_i \left[Chain(x\prime_i) - \hat{\mu}_2(\hat{\theta}_2, x\prime_i)\right]^2 \tag{16}$$

where $\hat{\mu}_1(\hat{\theta}_1, y\prime) = (-\alpha \ln y\prime + \beta)^2$ and $\hat{\mu}_2(\hat{\theta}_2, x\prime) = -\gamma \ln x\prime + \omega$ denote the fitting functions.

Beside numerical calculation, we can utilize the property mentioned in Sec. III D:

$$\frac{\beta}{\alpha} \approx \ln H_1; \ \frac{\omega}{\gamma} \approx \ln V_1 \tag{17}$$

so that the minimization of S_1 and S_2 become manually solvable. Now, $\hat{\mu}_1(\hat{\theta}_1, y\prime) \approx \hat{\mu}_1(\alpha, y\prime) = \alpha^2(-\ln y\prime + \ln H_1)$ and $\hat{\mu}_2(\hat{\theta}_2, x\prime) \approx \hat{\mu}_2(\gamma, x\prime) = \gamma(-\ln x\prime + \ln V_1)$. From $\partial S_i / \partial \hat{\theta}_i = 0$, we obtain

$$\sum_i 2\Big[Allo(y\prime_i) - \alpha^2(\ln H_1 - \ln y\prime_i)^2\Big]\alpha(\ln H_1 - \ln y\prime_i)^2 = 0 \tag{18}$$

$$\sum_i 2\Big[Chain(x\prime_i) - \gamma(\ln V_1 - \ln x\prime_i)\Big](\ln V_1 - \ln x\prime_i) = 0. \tag{19}$$

Since $\alpha \neq 0$, the solutions

$$\alpha = \sqrt{\frac{\sum_i Allo(y\prime_i)(\ln H_1 - \ln y\prime_i)^2}{\sum_i (\ln H_1 - \ln y\prime_i)^4}} \tag{20}$$

and

$$\gamma = \frac{\sum_i Chain(x\prime_i)(\ln V_1 - \ln x\prime_i)}{\sum_i (\ln V_1 - \ln x\prime_i)^2}. \tag{21}$$

C. De-noising and pattern reconstruction for scaling structure

The quantification of SS includes two steps:

1. Locate the points that fall on the upper envelopes
2. Calculate the total scaling ratio r_g (as in Fig. 2) and SC

In Step 1, separating points from fog is a hard quest. We found that dividing data points into different rectangles is helpful because the curve g_ℓ comes from the **upper envelopes** in rectangles $\{m, n\}, ..., \{m+i, n+i\}, ...$, where ℓ, m, n satisfy Eq. (5). To find g_ℓ, we need to remove fog as more as possible. First, for each block x in $\{m, n\}$, select the component with the highest y and discard other points. Second, utilize the property of being **concave down** to further filter these points. As can be seen in Fig. 1 of the main text, the envelopes can be approximately described as **concave functions** (see Appendix A) so we only need to look for the points above the diagonal line in each rectangle. In other words, we select the points (x, y) whose

$$y \geq y_{\text{diag}} = \Big(\frac{H_n - H_{n+1}}{V_m - V_{m+1}}\Big)(x - V_m) + H_n. \tag{22}$$

For $\{m+i, n+i\}$, just replace $\{m, n\}$ by $\{m+i, n+i\}$. In practice, deal with $i = 0 \sim 4$ is enough to cover g_ℓ because it is hard to observe g_ℓ in small rank region.

The above procedure eliminates most of the fog and greatly improves the efficiency of de-noising algorithm. However, in rare cases, the upper envelope in a rectangle is **not** concave down. Moreover, it may be locally convex or concave so that Eq. (22) work not well (see Appendix A and Fig. 7 for details). This issue can be modified by shifting our data with a concave function. Assume such situation happens in $\{m, n\}$. First, calculate the shifted points $\vec{r}_s = (x, y_s)$ of $\vec{r} = (x, y)$ where $y_s = y + \Delta y = y + (f_{cave} - y_{\text{diag}})$ and

$$f_{cave} = \tilde{a}(\delta)(x - V_m)(x - V_{m+1}) + \tilde{b}(\delta)(x - V_m) + \tilde{c}(\delta). \tag{23}$$

Second, select $\vec{r}$ if $\vec{r}_s$ satisfied Eq. (22). We name the above operations "concave shift". Now, no matter these envelopes are concave or convex, we have modified the raw data to new points $\vec{r}_0$ that contain little fog.

After removing most of messy points, we apply total variation reconstruction[20] to locate g_ℓ. The modified points $\vec{r}_0 = (x_0, y_0)$ can be rebuilt by finding the best reconstructed points $\hat{r} = (\hat{x}, \hat{y})$ which can minimize the following function:

$$\sum_i \left[\psi(\hat{x}_i - x_{0i}) + \psi(\hat{y}_i - y_{0i})\right] + \Gamma \cdot \phi_{tv}(\hat{r}) \tag{24}$$

where Γ denotes the regularization parameter (set as 1 in our study). The penalty function ψ (here we use Huber loss) and the regularization function ϕ_{tv} are defined as

$$\psi(D) \equiv \begin{cases} D^2 & \text{if } |D| \leq D_0 \\ D_0(2|D| - D_0) & \text{if } |D| > D_0 \end{cases} \tag{25}$$

$$\phi_{tv}(\hat{r}) \equiv \sum_i \left[|\hat{x}_{i+1} - \hat{x}_i| + |\hat{y}_{i+1} - \hat{y}_i| \right] \tag{26}$$

where D is the difference between reconstructed and original data, and the threshold D_0 is set to be 50 in our studies. Better than the method of least square, ψ can deal with outliers, while ϕ_{tv} makes reconstructed points more compact. One should note that the points $\vec{r_0}$ **must be sorted** according to their x_0 or y_0 value (we choose x_0 here); otherwise, ϕ_{tv} cannot work effectively since the reconstructed points are not compact. Figure 6 shows the outcome via the above technique.

In Step 2, we want to find an index to gauge the goodness of scaling. Let us recall what SS is. If different envelopes g_ℓ can be associated with a constant ratio

$$r_g = g_{\ell+1}(x)/g_\ell(x), \tag{27}$$

we shall call $\{g_\ell\}$ as SS. However, it is hard to find a common x on different g_ℓ in real cases. To correct this, we coarse-grain the reconstructed points on each g_ℓ into 101 points $\{(X_k, Y_k)_\ell\}$ where $k = 1 \sim 101$ (not 100, because we want the precision with 3 digits). The coarse-grain method includes three operations. First, determine the range of observation x_{ob}. Second, divide x_{ob} into 101 windows with size $W = [\inf(x_{ob}) - \sup(x_{ob})]/101$. Third, average the reconstructed points of g_ℓ in each windows to get $\{(X_k, Y_k)_\ell\}$. Based on the experience, we determine $x_{ob} = [V_1/4, V_1]$, so that $W = 3V_1/404$.

The set of scaling ratio $(r_g)_\ell$ and total scaling ratio r_g can be computed as

$$(r_g)_\ell \equiv \left\{ \frac{(Y_k)_{\ell+1}}{(Y_k)_\ell} \right\}_k, \; r_g \equiv \langle (r_g)_\ell \rangle + \sigma_g \tag{28}$$

where σ_g is total standard error from error propagation of each $(r_g)_\ell$

$$\sigma_g = \sqrt{\frac{\sum_\ell \rho(\sigma_\ell) \cdot \sigma_\ell^2}{\sum_\ell \rho(\sigma_\ell)}} \tag{29}$$

and $\langle \; \rangle$ denotes the weighted average

$$\langle Q_\ell \rangle \equiv \frac{\sum_\ell \rho(Q_\ell) \cdot \bar{Q}_\ell}{\sum_\ell \rho(Q_\ell)} \tag{30}$$

where σ_ℓ is standard error of $(r_g)_\ell$, $\rho(Q_\ell) \equiv$ size of $\{x | x \in Q_\ell, x \neq nan\}$ excludes the empty points on g_ℓ (comes from $(Y_k)_\ell = 0$ that makes $(Y_k)_{\ell+1}/(Y_k)_\ell = nan$), and $\bar{Q}_\ell \equiv$ arithmetic average of Q_ℓ.

Now, the soundness S and clearness C are defined as

$$S \equiv 1 - \frac{\sigma_g}{\langle (r_g)_\ell \rangle}, \; C \equiv \langle \left\{ \frac{\rho((r_g)_\ell)}{101} \right\}_\ell \rangle. \tag{31}$$

The higher S is, the more similar the function form of different g_ℓ is. The higher C is, the clearer those g_ℓ can be seen since g_ℓ contains fewer fog. A good scaling pattern should simultaneously exhibit the soundness S and clearness C, so we can define $SC \equiv S \times C$ as the index to quantify the goodness of scaling.

Figure 2 exhibits the SC value for different samples. In practice, we only consider $\ell = 2, 3, 4$. The g_1 is excluded because the components on g_1 are rarely used to compose other blocks (as explained in Sec. III B). In other words, they are unique and do not follow the scaling property. After calculate SC value for all samples, we find most of their $SC > 0.7$. Thus, it is reasonable to set $SC > 0.7$ as a criteria of SS.

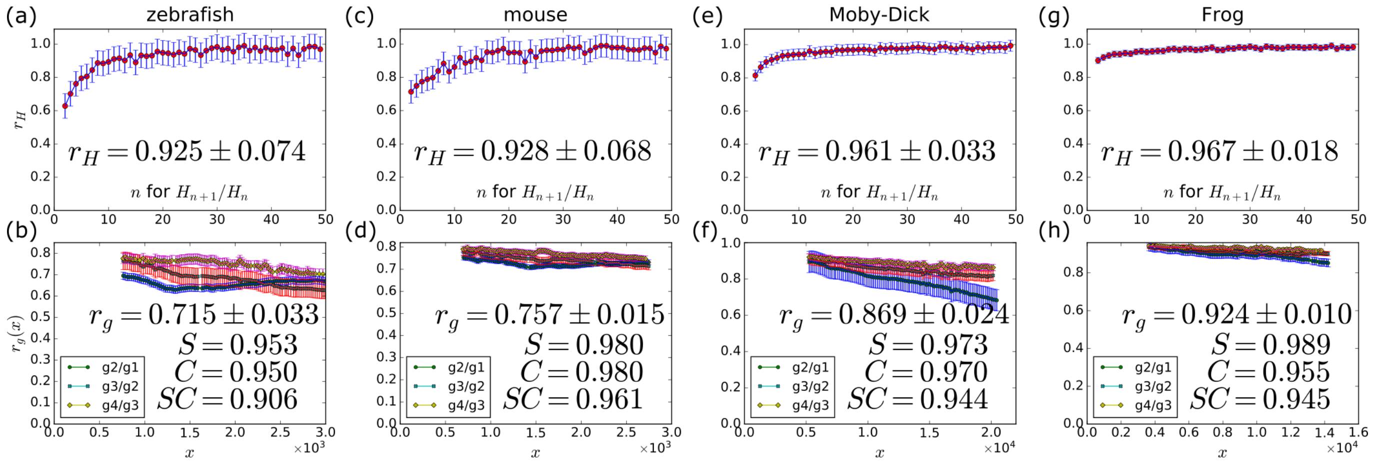


FIG. 2: The upper figures show the weighted average scaling relation of $r_H \equiv H_{n+1}/H_n$ (see Sec. VI A), while the lower ones depict r_g with standard error, σ_H and σ_g. Panels (a, b) are for gene of zebrafish, (c, d) for gene of mouse, (e, f) for English novel *Moby-Dick*, (g, h) for Chinese novel *Frog*. Definitions of r_g and SC, a measure of goodness of scaling, are given in Sec. IV C.

TABLE II: Statistical quantities of some sample corpora used in our research. The column "Zipf b" denotes the fitting exponent for power law $P(x) = a/x^b$. The FRD of all corpora obey power law, except No. 1 (follow double power law $P(x) = a_1/x^{b_1} + a_2/x^{b_2}$), 2 (Gaussian $P(x) = (\sqrt{2\pi}\sigma)^{-1}\exp(-x^2/2\sigma^2)$), 3 (log-normal $P(x) = (x\sqrt{2\pi}\sigma)^{-1}\exp(-(\ln x)^2/2\sigma^2)$), and 4 (exponential $P(x) = a_3\exp(-a_3 x)$). The "r_g" and "SC" are defined in Sec. IV C, "U_{Chain}" is the supremum of *Chain* as defined in Sec. II B 2, "V_1" refers to the size of word inventory (total kinds of words), and "L" is the length of Book. The abbreviation "Sci." means scientific paper. More Book and other quantities (such as α, β) can be found in Supplementary Data.

No.	Sample	Zipf b	r_g	SC	U_{Chain}	V_1	L	Language	Reference
1	2-gram-DoublePowerLaw	1.4009	0.888	0.818	68	3307	200000	Fake	Sec. II B 2
2	2-gram-Gaussian	0.42	0.946	0.344	105	3560	20000	Fake	Sec. II B 2
3	2-gram-LogNormal	0.9614	0.884	0.855	181	4145	20000	Fake	Sec. II B 2
4	2-gram-Exponential	0.695	0.946	0.010	17	595	20000	Fake	Sec. II B 2
5	2-gram-Zipf	0.9521	0.854	0.844	89	3328	15000	Fake	Sec. II B 2
6	3-gram-Zipf	0.9540	0.867	0.877	170	3401	15000	Fake	Sec. II B 2
7	4-gram-Zipf	0.9544	0.881	0.886	284	3351	15000	Fake	Sec. II B 2
8	5-gram-Zipf	0.9575	0.893	0.877	318	3363	15000	Fake	Sec. II B 2
9	Chopsticks	0.848	0.762	0.288	129	908	2548	Sci. English	[39]
10	*Demi-Gods and Semi-Devils* 天龍八部	0.97246	0.954	0.900	1632	35222	695218	Mandarin	[37]
11	Empirical Tests of Zipf	0.866	0.799	0.240	91	799	2556	Sci. English	[40]
12	Excerpts from *Frog* 蛙	0.71	0.672	0.374	33	636	1224	Mandarin	[29], Sec. II B 4
13	Excerpts from *Moby-Dick*	0.77	0.705	0.198	58	652	1397	English	[34], Sec. II B 4
14	*Frog* 蛙	0.9545	0.924	0.945	697	14380	110578	Mandarin	[29]
15	LIGO	0.871	0.832	0.342	234	1288	4892	Sci. English	[38]
16	Mix	0.96327	0.927	0.941	999	22718	174588	Mandarin	Sec. II B 3
17	*Moby-Dick*	0.99194	0.869	0.944	2402	20687	203206	English	[34]
18	News	0.736	0.855	0.860	158	4142	12555	Mandarin	[28], Sec. II B 3
19	Paper-Generator1	0.8539	0.928	0.059	344	2127	26448	Fake Sci.	[26], Sec. II B 1
20	*The Hobbit*	0.9856	0.862	0.879	642	7686	94011	English	[35]
21	*Xu Zhimo poems* 徐志摩詩選	0.861	0.715	0.545	36	1234	3025	Mandarin	[36]

D. Trial and error for the origin of scaling structure

In this section, we want to present the process of trial and error that eventually aids us at locating the origin of SS. This section may help future researchers construct GLC at higher levels of organized unit.

In the early stage of our research, we found SS generally existed in different kinds of natural texts. To control the possible variables that affect SS, we turned our attention to the artificial texts (see Sec. II B). From hindsight, SS does not originate from those intuitive factors that were mentioned in Sec. II B. This conclusion is based on the answers of following questions with reasons:

TABLE III: This table determines that a sound SS, represented by a large SC value, relies on a large size of word bank V_1, and can still exist for fake corpora that consist of words composed of random syllagrams or do not follow rules of writing.

	natural corpora	scientific article	fake, No. 19 (real words)	fake, No. 1, 3, 5 ∼8 (fake words)
Use real words	yes	yes	yes	no
Obey grammar	yes	yes	yes	no
$V_1 > 1500$	yes	no	yes	yes
Zipfian or Zipf-like FRD	yes	yes	no	yes
$SC > 0.7$	yes	no	no	yes

1. Is SS a consequence of Zipf's law?
   Reason: The RRD is built from FRD. For natural texts, all of them exhibit Zipf's law for words. Therefore, we want to know whether Zipf's law is the only factor to produce SS.
2. What is the role of book length L?
   Reason: The L is a factor that affects the normalization of FRD. This is different from the Zipf's law, which mainly discusses the influence of exponent.
3. Is the grammar a deciding factor?
   Reason: Grammar affects how words compose a sentence. In other words, it regulates the rule whose level higher than the level of word formation.
4. Is SS common to different writing styles?
   Reason: Writing style is a much higher level variables than grammar, which affects the usage of words.

To clarify these questions, we summarize statistical quantities of different corpora in Table II and organize our analyses in Table III, while the goodness of SS is quantified by SC (most of our real data satisfy $SC > 0.7$), as defined in Sec. IV C.

First, the answer to question 1 is negative. It comes from two counterexamples: (i) An article consists of 1-gram words whose FRD follows Zipf's law, but its RRD is a straight line, i.e., no SS. (ii) Articles No. 1 and 3 does not follow Zipf's law, but their $SC > 0.7$, satisfy the requirement of SS. In fact, it is hard to directly distinguish whether they obey Zipf's law unless we really make the AIC test. In our case, double power law

$$P(x) = \frac{a_1}{x^{b_1}} + \frac{a_2}{x^{b_2}} \sim \frac{a_1}{x^{b_1}} \tag{32}$$

when $a_1 \gg a_2$ or $b_1 \ll b_2$ and log-normal

$$P(x) = \frac{e^{-(\ln x)^2/2\sigma^2}}{x\sqrt{2\pi}\sigma} \sim \frac{x^{-1}}{\sqrt{2\pi}\sigma} \tag{33}$$

when σ is large enough (this can be proved by take log on both side of log-normal distribution[13]). This counterexample implies a "Zipf-like" distribution is also acceptable to SS, not only Zipfian. Besides, we notice that No. 2 and 4, whose FRD is not Zipf-like, show no structure as in Fig. 3 (c, d). Thus the fact that **FRD is Zipfian or Zipf-like is merely a necessary, not sufficient, condition for SS**.

Second, SS is not guaranteed by a large L because No. 2 and 4 ($L = 20000$) do not exhibit SS, but No. 3, 5, 6, 7, 8 with equal or smaller article size ($L = 20000, 15000$) do. Nonetheless, a too small L is sure to ruin SS, as evidenced by No. 12 and 13 in Fig. 3 (a, b). Compare No. 12 and 13 to their original work, No. 14 and 17, we can confirm **a large L is merely a necessary, not sufficient, condition for SS**. Based on experience, we found $L > 5000$ is a rough threshold to produce SS.

Third, in response to question 3, No. 19, a man-made corpora based on grammar (see Sec. II B 1), has a very low SC. In contrast, No. 1, 3, 5, 6, 7, 8 that do not follow grammar acquire high SC. They indicate that **grammar and real word formation are inessential for SS**. Fourth, the answer to question 4 is a sound yes, as proven by the high SC for No. 16 and 18.

Last, but not the least, candidate to affect SS is V_1. In No. 9, 11, 15 (scientific paper), and 21 (poems), whose SC falls below 0.7, which proves that **a small word inventory is detrimental to SS**. In other words, the "jargon" limits V_1.

We should notice the discussion here only addresses SS, not for GLC. Although SS does not require word formation, **it is essential to other statistical features**, such as degree distribution of network and frequency distribution of the number of syllagrams in a word. We will demonstrate such viewpoint in Sec. V B.

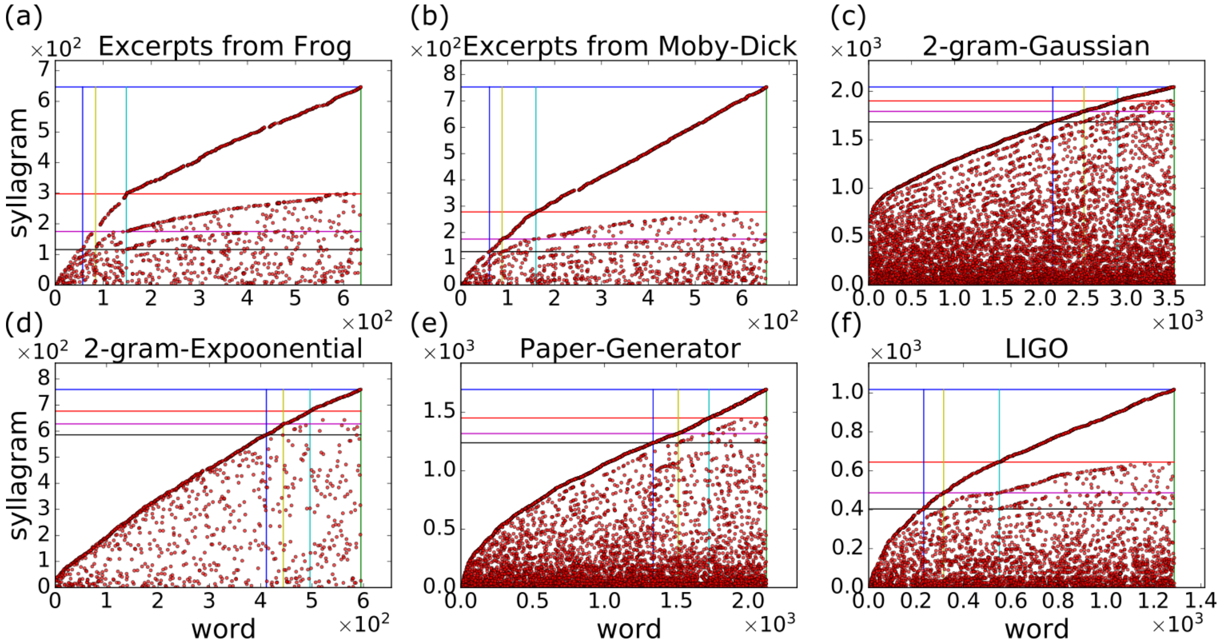


FIG. 3: Cases with $SC < 0.7$ due to different reasons: (a, b) L is too small, (c, d, e) FRD is neither Zipf nor Zipf-like, (f) V_1 is too small. Their parameters can be found in Tab. II. Note that the computer-generated paper in (e) is created by SCIgen[26] which has succeeded at deceiving the editors of many journals and conference organizers. But our RRD analysis can tell it from real paper because its RRD is not a scaling structure.

V. DETAILS ABOUT SIMULATION

To test our mechanism of evolution (see METHOD of the main text), we design a program[24] that can generate a Book. The user can define the following parameters: (i) length of Book L, (ii) effective connection z, (iii) system parameter λ, (iv) mutation probability $P_{\rm mu}$, (v) maximum repeat count T for mutation within a time step (see Sec. E of METHOD in the main text), and (vi) the prior probability distribution $\mathbb{P}_N$. In Supplementary Data, we record the parameters $(z, \lambda, P_{\rm mu}, T)$ via the filename of Book in the form like

1_495_005_1.txt

where the first number implies $z = 0.1L$, the second $\lambda = 0.495$, the third $P_{\rm mu} = 0.005$, and the last is the maximum repeat count T (see Sec. V A 3). The length L can be measured by the statistical program so it is not necessary to record. In the following simulation, we set the length $L = 10000$ and the probability $\mathbb{P}_N = (\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3, \mathbb{P}_4, \mathbb{P}_5) = (0.15, 0.40, 0.25, 0.15, 0.05)$.

A. Much faster evolution algorithm

A direct calculation for our theory in METHOD of the main text will **greatly** slow down the program because it involves too many matrix operations. To improve this, we can utilize the properties of **conserved change**, "no synonym" interpretation, and association matrix $\mathbf{A} = \{a_{ij}\}$.

As we mentioned in Fig. 12 of the main text, the step "Add $s_{q(t)}$ and build its FC" contains three operations. Under the condition of **conserved change**, the operations "Decide $N_{q(t)}$ via $\mathbb{P}_N$" and "Build FC network" can be pre-calculated with the step "Create Book $= s_1...s_{q_0}$ and association matrix $\mathbf{A}$" in Fig. 2 of the main text (we will talk about this later) so that we only have to deal with "Add $s_{q(t)}$" in the loop of time $t = 1 \sim L - q_0$.

In Fig. 4 below, there are two steps that affect the content of Book, i.e., sequential variation: "Add $s_{q(t)}$" (length of Book is changed) and "Mutation" (length of Book is not changed). Because the mutation may repeat many times within a time step, we denote the time-quantities as $Q^{[t,\tau]}$ or $Q(t, \tau)$ where the time step $t = 1 \sim L - q(0)$ and repeat counts $\tau = 0 \sim T$. To simplify the notations, the shorthand for time-quantities:

$$\begin{aligned} \text{before mutation, } & Q^{[t]}, Q(t) \equiv Q^{[t,0]}, Q(t, 0) \\ \text{after mutation, } & \tilde{Q}^{[t]}, \tilde{Q}(t) \equiv Q^{[t,T]}, Q(t, T). \end{aligned} \tag{34}$$

In the following, we symbolize

$$\begin{aligned} \text{Book}^{[t,\tau]} &\equiv s_1 \ ... \ s_j \ ... \ s_{q(t,\tau)} \\ \text{functions } \mathcal{F}^{[t,\tau]} &= \{f_1, ..., f_j, ..., f_{q(t,\tau)}\} \\ \text{block inventory } \mathcal{B}^{[t,\tau]} &= \{b_1, ..., b_i, ..., b_{p(t,\tau)}\} \end{aligned} \tag{35}$$

where the indices $1 \leq i \leq p$ and $1 \leq j \leq q$ are used to describe block and function, respectively. For q, $q(t) = \tilde{q}(t)$ since "Mutation" is defined under the condition that the length of Book is not changed; while there is no such a restriction for p.

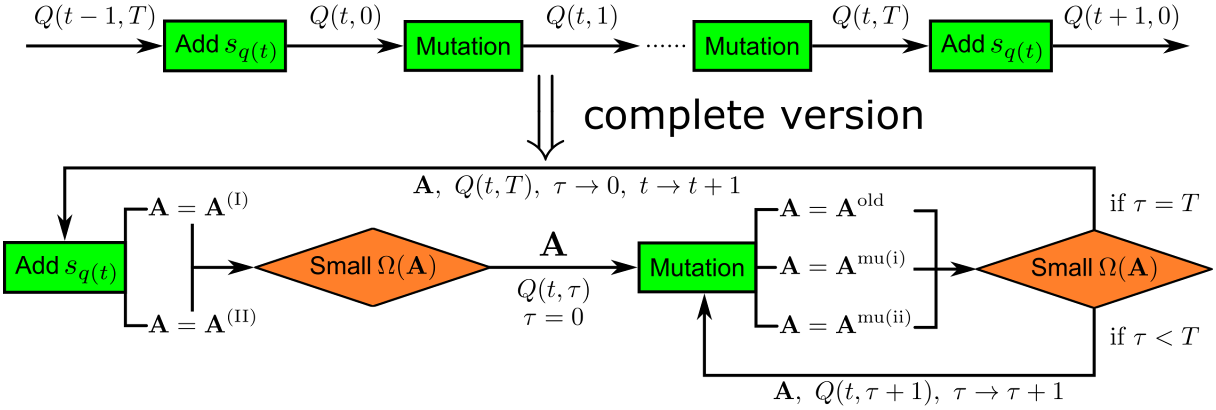

$Q(t-1,T)$
Add $s_{q(t)}$
$Q(t,0)$
Mutation
$Q(t,1)$
......
Mutation
$Q(t,T)$
Add $s_{q(t)}$
$Q(t+1,0)$
complete version
$\mathbf{A},\ Q(t,T),\ \tau \to 0,\ t \to t+1$
Add $s_{q(t)}$
$\mathbf{A} = \mathbf{A}^{(\mathrm{I})}$
$\mathbf{A} = \mathbf{A}^{(\mathrm{II})}$
Small $\Omega(\mathbf{A})$
$\mathbf{A}$
$Q(t,\tau)$
$\tau = 0$
Mutation
$\mathbf{A} = \mathbf{A}^{\mathrm{old}}$
$\mathbf{A} = \mathbf{A}^{\mathrm{mu(i)}}$
$\mathbf{A} = \mathbf{A}^{\mathrm{mu(ii)}}$
Small $\Omega(\mathbf{A})$
if $\tau = T$
if $\tau < T$
$\mathbf{A},\ Q(t,\tau+1),\ \tau \to \tau+1$

FIG. 4: The flowchart on the top row is a simple version of the complete one below it. Both demonstrate our evolution algorithm in the loop of time, where the notation $a \to b$ means a is replaced by b. The step "Add $s_{q(t)}$" provides two modes to select the direction of evolution via the principle of least effort ("Small $\Omega(\mathbf{A})$"). Afterward, the information of Book $(\mathbf{A}, Q(t,0))$ will be sent to the next step. Different from "Add $s_{q(t)}$", the step "Mutation" is a small probability event. Each block of Book has the mutation probability P_{mu}. Once it happens, there are three possible variations to be selected from via the principle of least effort (details can be found in Sec. V A 3). For Book evolves within a time step, we set the maximum repeat count as T. After $\tau = T$, the information of Book $(\mathbf{A}, Q(t,T))$ will be sent to the next loop (next time step).

1. *Build function connection network*

In our mechanism of evolution, most changes of $\mathbf{A}$ involve function connection (FC). We exemplify our simulation as a minimal reproducible example in **conserved increase**. The length of Book $q(t) = q(t-1)+1 = q(0)+t = q_0+t$. The new function does not affect old FCs. That means we can pre-calculate the FC network instead of evaluating it every time step. In the simplest simulation to test how FC influences evolution, we assume the FCs between different functions are random (a uniform FC is obviously nonsense). To construct such a FC network, there are 3 operations.

First operation: decide the number of component of each function based on the prior probability distribution $\mathbb{P}_N$. We use a one-dimensional array to save this information:

$$\vec{N}^{[t=0,\tau=0]} = \left[N_1^{[0]}, ..., N_L^{[0]}\right] \tag{36}$$

where N_j represents the number of component of s_j. In our simulation, the maximal number of components in a block $N_{\max} = \sup\{N_j \mid N_j \in \vec{N}\} = 5$.

Second operation: generate $L \times L$ random hollow symmetric matrices $\Lambda^{(k)}$ for layer $k = 1 \sim 5$

$$\Lambda_{\mu,\nu}^{(k)} = \begin{cases} 0 & \text{if } \mu = \nu \\ \Lambda_{\nu,\mu}^{(k)} \in [0,1] & \text{if } \mu \neq \nu \end{cases} \tag{37}$$

where $1 \leq \mu, \nu \leq L$. The FC at $k-$position between f_μ and f_ν can be calculated as

$$C_{\mu,\nu}^{(k)}(t,\tau) = \begin{cases} \Lambda_{\mu,\nu}^{(k)} & \text{if } N_\mu^{[t,\tau]}, N_\nu^{[t,\tau]} \geq k \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

which comes from the fact that $C^{(k)}_{\mu,\nu} = 0$ once s_μ or s_ν does not have the $k-$position component. For example, s_μ has 3-components and s_ν has 5, then $\min(N_\mu, N_\nu) = \min(3, 5) = 3$ and $C^{(k)}_{\mu,\nu} = 0\ \forall k > 3$. When $\mu = \nu$, $C^{(k)}_{\mu,\nu} = 0$. **Note**: do not pre-calculate $C^{(k)}_{\mu,\nu}$ via this formula, calculate it when needed.

Third operation: sum up $\Lambda^{(k)}$ with different number of layers:

$$\Lambda^d = \sum_{k=1}^{d} \Lambda^{(k)} = \begin{cases} \Lambda^{(1)} & \text{if } d = 1 \\ \Lambda^{d-1} + \Lambda^{(d)} & \text{otherwise} \end{cases} \tag{39}$$

where $d = 1 \sim 5$ denotes the total number of layers (sum with the iteration formula will be much faster.)

There are three situations that involve FC network. In "Add $s_{q(t)}$", the Book needs to decide the direction of evolution from (I) co-option or (II) *de novo*. In "Mutation", the *de novo* like mutation needs FC. Let us talk about them later.

2. Direction of evolution

Under the "no synonym" interpretation, the probability of producing block b_i for time step t is

$$P^{[t]}(b_i) = \frac{1}{q(t)} \sum_{j=1}^{q(t)} a^{[t]}_{ij} = \frac{\rho^{[t]}(b_i)}{q(t)} \tag{40}$$

where $q(t)$ is the size of functions at time t, $b_i \in \mathcal{B} = \{b_1, ..., b_{p(t)}\}$, and the frequency of $\mathcal{B}$ is a vector $\vec{\rho}^{\,[t]}(\mathcal{B}) = \{\rho^{[t]}(b_i)\} \equiv \{\sum_{j=1}^{q(t)} a^{[t]}_{ij}\} = \{\rho^{[t]}(b_1), ..., \rho^{[t]}(b_{p(t)})\}$. Similarly, the conditional probability of assigning block b_i to function f_j is

$$P^{[t]}(f_j|b_i) = \frac{a^{[t]}_{ij}}{\sum_{\mu=1}^{q(t)} a^{[t]}_{i\mu}} = \frac{a^{[t]}_{ij}}{\rho^{[t]}(b_i)}. \tag{41}$$

Apply Eq. (40) and define the "total entropy"

$$h^{[t]}_p(\mathcal{B}) \equiv -\sum_{i=1}^{p(t)} \rho^{[t]}(b_i) \log_{p(t)} \rho^{[t]}(b_i) = -\sum_{i=1}^{p(t)} \sigma^{[t]}_p(b_i) \tag{42}$$

where $\sigma^{[t]}_p(b_i) \equiv \rho^{[t]}(b_i) \log_{p(t)} \rho^{[t]}(b_i)$ symbolizes the "entropy" of block b_i, then the individual effort becomes

$$\mathcal{H}^{[t]}_p(\mathcal{B}) \equiv -\sum_{i=1}^{p(t)} P^{[t]}(b_i) \log_{p(t)} P^{[t]}(b_i) = \frac{1}{q(t)} \left[h^{[t]}_p(\mathcal{B}) + \sigma^{[t]}_p(q) \right] \tag{43}$$

where $\sigma^{[t]}_p(q) \equiv q(t) \log_{p(t)} q(t)$ denotes the "entropy" of length. An interesting discovery is that $-h^{[t]}_p(\mathcal{B}) - \sigma^{[t]}_p(q)$ is similar to Eq. (13).

Apply Eq. (41) and the identity

$$a^{[t]}_{ij} \log_{q(t)} a^{[t]}_{ij} = 0 \tag{44}$$

(for $a_{ij} = 0$, proved by $\lim_{x\to 0^+} x \log x = 0$), then the collective effort for b_i becomes

$$\mathcal{H}^{[t]}_q(\mathcal{F}|b_i) \equiv -\sum_{j=1}^{q(t)} P^{[t]}(f_j|b_i) \log_{q(t)} P^{[t]}(f_j|b_i) = \log_{q(t)} \rho^{[t]}(b_i); \tag{45}$$

while the total collective effort

$$\mathcal{H}^{[t]}_q(\mathcal{F}|\mathcal{B}) \equiv \sum_{i=1}^{p(t)} P^{[t]}(b_i) \mathcal{H}^{[t]}_q(\mathcal{F}|b_i) = \sum_{i=1}^{p(t)} \left[\frac{\rho^{[t]}(b_i)}{q(t)} \right] \mathcal{H}^{[t]}_q(\mathcal{F}|b_i) = \sum_{i=1}^{p(t)} \frac{\sigma^{[t]}_q(b_i)}{q(t)} = \frac{-\log_{q(t)} p(t)}{q(t)} h^{[t]}_p(\mathcal{B}) \tag{46}$$

where $\sigma_q^{[t]}(b_i) \equiv \rho^{[t]}(b_i) \log_{q(t)} \rho^{[t]}(b_i) = (\log_q p)\sigma_p^{[t]}(b_i)$ denotes the “entropy” of block b_i with varied length $q(t)$.

Once knowing $h_p^{[t]}(\mathcal{B})$, we can obtain the total effort via simple formulation

$$\Omega_\lambda^{[t]}(\mathbf{A}) = \lambda \mathcal{H}_q^{[t]}(\mathcal{F}|\mathcal{B}) + (1-\lambda)\mathcal{H}_p^{[t]}(\mathcal{B}) = \frac{1}{q(t)} \Big\{ \Big[1 - \lambda \big(1 + \log_q p\big)^{[t]} \Big] h_p^{[t]}(\mathcal{B}) + (1-\lambda)\sigma_p^{[t]}(q) \Big\} \tag{47}$$

where system parameter $0 \le \lambda \le 1$. **<u>Note</u>**: although we wrote down the above formulae for $Q^{[t]}$, they can also be applied to $Q^{[t,\tau]}$.

In conserved change when $t-1 \to t$, the evolution may adopt (I) co-option or (II) *de novo*. We need to compare the total effort for (I) and (II) to select the mode with the higher survival rate.

For mode (I) co-option, $p(t) = \tilde{p}(t-1)$, we build the total FC vector for $f_{q(t)}$:

$$\vec{C}_q(t) = \big[C_{q,1}(t), ..., C_{q,q-1}(t)\big] = \big[C_{q,\mu}(t)\big]_\mu = \big[\Lambda_{q,\mu}^{d(t)}\big]_\mu \tag{48}$$

where $C_{q,\mu}(t) \equiv \sum_{k=1}^{d(t)} C_{q,\mu}^{(k)}(t)$, $d(t) = \min(N_q^{[t]}, N_\mu^{[t]})$, $1 \le \mu \le q(t-1) = q-1$, and the third equality comes from Eq. (39). So the probability that $f_{q(t)}$ uses s_j is (see Eq. (19) of the main text):

$$P_{q,j}^{old}(t) = \frac{C_{q,j}(t)}{\sum_\mu C_{q,\mu}(t)} \tag{49}$$

where $C_{q,j}, C_{q,\mu} \in \vec{C}_q$. Assume the new function $f_{q(t)}$ chooses an old block s_ζ on Book, then (we label $b_\nu = s_\zeta$) $N_q^{[t]} \to N_\zeta^{[t]}$ and

$$a_{ij}^{[t]} = \begin{cases} 1 & \text{if } i = \nu \ \&\ j = q(t) \\ 0 & \text{if } i \neq \nu \ \&\ j = q(t) \\ \tilde{a}_{ij}^{[t-1]} & \text{otherwise} \end{cases} \Rightarrow \rho^{[t]}(b_i) = \begin{cases} \tilde{\rho}^{[t-1]}(b_i) + 1 & \text{if } i = \nu \\ \tilde{\rho}^{[t-1]}(b_i) & \text{otherwise} \end{cases}. \tag{50}$$

Using the above iteration, $h_p^{[t]}(\mathcal{B})$ can be calculated via Eq. (42). We can substitute this result into Eq. (47) to obtain the total effort of co-option $\Omega_\lambda^{[t]}(\mathbf{A}^{(\mathrm{I})})$ at time t where the upper symbol $^{(\mathrm{I})}$ means all the calculations are done in mode (I). **<u>Note</u>**: in fact, once $N_q^{[t]} \neq N_\zeta^{[t]}$, the total FC will also be changed because $\min(N_q^{[t]}, N_\mu^{[t]}) \neq \min(N_\zeta^{[t]}, N_\mu^{[t]})$. But for the simplest simulation, $\vec{C}_q$ is used only once in co-option and not in other places. We do not need to revise $\vec{C}_q$.

For (II) *de novo*, we build the $k-$position FC vector for $f_{q(t)}$ (via Eq. (38)):

$$\vec{C}_q^{(k)}(t) = \big[C_{q,1}^{(k)}(t), ..., C_{q,q-1}^{(k)}(t), z\big] \tag{51}$$

where the q^{th} term z is the effective connection. The probability of “€creating”€ the k^{th} component for $f_{q(t)}$ is (see Eq. (21) of the main text)

$$P_{q,new}^{(k)}(t) = \frac{z}{z + \sum_\mu C_{q,\mu}^{(k)}(t)}, \tag{52}$$

while the probability that $f_{q(t)}$ uses a k^{th} component in s_j (see Eq. (22) of the main text):

$$P_{q,j}^{(k)}(t) = \frac{C_{q,j}^{(k)}(t)}{z + \sum_\mu C_{q,\mu}^{(k)}(t)} \tag{53}$$

where $P_{q,new}^{(k)}(t) + \sum_{j=1}^{q(t-1)} P_{q,j}^{(k)}(t) = 1$. To decide what $s_{q(t)}$ should be, one needs to run the component selection for $k = 1 \sim N_q^{[t]}$.

There are two possible results of the component selections. First, “create” an already existing block s_ζ, then the computation of total effort will follow mode (I). In such a case, it does not mean that mode (I) equals (II). Because the probabilities of selecting old block in these two modes are different, it would be hard to simultaneously select the same block and obtain the same effort in both modes. Second, create a new kind of block $b_{p(t)}$, i.e., $p(t) = \tilde{p}(t-1)+1$, then

$$a^{[t]}_{ij} = \begin{cases} 1 & \text{if } i = p(t) \ \& \ j = q(t) \\ 0 & \text{if } i \neq p(t) \ \& \ j = q(t) \\ 0 & \text{if } i = p(t) \ \& \ j \neq q(t) \\ \tilde{a}^{[t-1]}_{ij} & \text{otherwise} \end{cases} \Rightarrow \rho^{[t]}(b_i) = \begin{cases} 1 & \text{if } i = p(t) \\ \tilde{\rho}^{[t-1]}(b_i) & \text{otherwise} \end{cases}. \tag{54}$$

Again, $h^{[t]}_p(\mathcal{B})$ can be calculated via Eq. (42), and the total effort of *de novo* $\Omega^{[t]}_\lambda(\mathbf{A}^{(\mathrm{II})})$ at time t can be obtained by substituting $h^{[t]}_p(\mathcal{B})$ into Eq. (47) where the upper symbol $^{(\mathrm{II})}$ means all the calculations are done in mode (II).

The direction of evolution is decided according to the principle of least effort. We compare Ω_λ for both modes and select the smaller one as the evolution mode at time t. After this step, saving the time-quantities $\mathcal{B}^{[t]}$, $\vec{\rho}^{\ [t]}$, $\vec{N}^{[t]}$, and $\mathrm{Book}^{[t]}$ for the next sequential variation.

3. *Mutation*

Now we come to the next step: mutation. We use "for loop" from s_1 to $s_{q(t)}$ with mutation probability P_{mu} for each s_j. There are two possible variations of mutation: (i) co-option like and (ii) *de novo* like. Ideally, we have to build a list for all the mutated s_j with randomly assigned variations. For example, there is a $L = 5$ Book (0: no change, 1: co-option like, 2: *de novo* like) whose

$$\text{mutation list with variations} = \{1, 0, 0, 2, 0\}. \tag{55}$$

We compare such a sequential variation with the original book through the principle of least effort. However, this is not necessary to test our general evolution mechanism in the simplest case. Because the program will cost an incredibly long time to test a huge amount of possible sequential variations - just like a real process of mutation. Instead, we run three variations for each mutated s_j, then pick up the best one. After we complete such an execution for Book (from $j = 1 \sim q$), the repeat count of mutation τ will increase by one. By setting a maximum repeat count $T \geq \tau$, we can control the amount of mutation.

In (i) co-option like mutation, the size of $\mathcal{B}$ will not change. For **the simplest case** (our simulation), we assume the probability that $s_j \to s_\xi$ obeys a uniform distribution:

$$P(s_j \to s_\xi) = \frac{1}{\text{size of Book} - 1} = \frac{1}{q(t) - 1} \tag{56}$$

where $1 \leq j,\ \xi \leq q(t)$ and $j \neq \xi$ (this is why the denominator is $q(t) - 1$). When $s_j \to s_\xi$ happens,

$$\begin{aligned} N^{[t,\tau]}_j &\to N^{[t,\tau]}_\xi \\ \rho^{[t,\tau]}(s_j) &\to \rho^{[t,\tau]}(s_j) - 1 \\ \rho^{[t,\tau]}(s_\xi) &\to \rho^{[t,\tau]}(s_\xi) + 1. \end{aligned} \tag{57}$$

The total effort $\Omega_\lambda(\mathbf{A}^{\mathrm{mu(i)}})$ can be obtained through Eq. (47).

In (ii) *de novo* like mutation, a new block b_{new} that does not exist in $\mathcal{B}$ will be created to substitute s_j. To do this, we build the $k-$position FC vector for f_j (via Eq. (38)):

$$\vec{C}^{(k)}_j(t,\tau) = \left[C^{(k)}_{j,1}(t,\tau), ..., C^{(k)}_{j,q(t)}(t,\tau), z\right] \tag{58}$$

where the $(q+1)^{th}$ term z is the effective connection. The probability of "creating" the k^{th} component for f_j is

$$P^{(k)}_{j,new}(t,\tau) = \frac{z}{z + \sum_\mu C^{(k)}_{j,\mu}(t,\tau)}, \tag{59}$$

while the probability that f_j uses a k^{th} component in s_j:

$$P^{(k)}_{j,\xi}(t,\tau) = \frac{C^{(k)}_{j,\xi}(t,\tau)}{z + \sum_\mu C^{(k)}_{j,\mu}(t,\tau)} \tag{60}$$

where $P^{(k)}_{j,new}(t,\tau) + \sum_{\xi=1}^{q(t)} P^{(k)}_{j,\xi}(t,\tau) = 1$. To decide what b_{new} should be, one needs to run the component selection for $k = 1 \sim N^{[t,\tau]}_q$.

Similar as *de novo*, the *de novo* like mutation also has a small probability to create an existing block. But in order to make a difference from co-option like mutation in the simplest case, we forbid the case that $b_{\text{new}} \in \mathcal{B}$. In other words, the program needs to run the component selection of b_{new} until $b_{\text{new}} \notin \mathcal{B}$.

When $s_j \to b_{\text{new}}$ and $b_{\text{new}} \notin \mathcal{B}$, the size of $\mathcal{B}$ may increase

$$p(t,\tau) \to \begin{cases} p(t,\tau) & \text{if } \rho^{[t,\tau]}(s_j) > 1 \\ p(t,\tau) - 1 & \text{if } \rho^{[t,\tau]}(s_j) = 1 \end{cases} \tag{61}$$

where s_j disappear from the Book in the second case. Besides, the frequency of block also changes

$$\begin{aligned} \rho^{[t,\tau]}(s_j) &\to \rho^{[t,\tau]}(s_j) - 1 \\ \rho^{[t,\tau]}(b_{\text{new}}) &\to 1. \end{aligned} \tag{62}$$

After obtaining the above information, we can calculate the total effort $\Omega_\lambda(\mathbf{A}^{\text{mu(ii)}})$ via Eq. (47).

The survival of variations of mutation depends on the principle of least effort for co-option like $\Omega_\lambda(\mathbf{A}^{\text{mu(i)}})$, *de novo* like mutation $\Omega_\lambda(\mathbf{A}^{\text{mu(ii)}})$, and original sequence $\Omega_\lambda(\mathbf{A}^{\text{old}})$. The winner will survive and wait for the next sequential variation. **Note**: since we only allow $b_{\text{new}} \notin \mathcal{B}$ in *de novo* like mutation, the total effort $\Omega_\lambda(\mathbf{A}^{\text{mu(ii)}})$ can be calculated from Eqs. (61, 62) before running the component selection of b_{new}. Our program runs the selection **after** *de novo* like mutation survives.

After finding out the best variations for all the muted s_j from $j = 1 \sim q$, we obtain the time-quantities after mutation (as shown in Fig. 4) as

$$Q(t,\tau+1) = Q(t,\tau) \text{ after mutation from } j = 1 \sim q \tag{63}$$

where $Q = \rho$, $\vec{N}$, and Book. What about $Q = \mathcal{B}$? The readers may notice a problem in our algorithm of mutation: there will be a lot of "empty" blocks in $\mathcal{B}$, which means they are not used in Book (frequency equals zero). Here we make a wild guess: when $q(t)$ becomes $q(t+1)$, these "empty" blocks will be discarded from $\mathcal{B}$ and not used in the future. Therefore, at the end of mutation in each time step, the algorithm needs to regulate $\mathcal{B}$ and $\rho(b_i)$. Briefly, delete those b_i whose $\rho(b_i) = 0$ after $\tau = T$.

B. Analysis of parameters

We demonstrate some results of our simulation in Tab. IV and Fig. 5. For samples No. 1, 2, 3, 7, and 8, their FRD of block ρ_x act like language ($b = 0.8 \sim 1$), while for samples No. 4, 5, 6, 9, 10, 11, and 12, their exponents act like protein sequence ($b = 0.42 \sim 0.7$). Roughly speaking, when z increases, the FRD of component ρ_y becomes more curved. When P_{mu} increases, the exponent b also grows up. But for more tests about how change of $(z, \lambda, P_{\text{mu}}, T)$ affects the statistical properties of GLC is still a puzzle for future researches.

VI. AN APPROXIMATE ANALYTIC FORM OF SCALING CURVE

Although Sec. IV C provides a good method to facilitate the identification of points on the scaling lines, we want to find a physical meaning for the scaling curves. To do this, an analytic form is helpful.

A. scaling function

Based on the scaling property, we assign $g_\ell(x) = a_\ell g(x)$. The second feature mentioned in Sec. III B implies

$$H_n = a_\ell g(V_{n-\ell+1}). \tag{64}$$

Let $m = n - \ell + 1$, then ℓ, m, n satisfy Eq. (5). By comparing Eq. (64) for each horizontal and vertical lines, we get

$$\frac{H_{n+1}}{H_n} = \frac{g(V_{m+1})}{g(V_m)} = \frac{a_{\ell+1}}{a_\ell} = r \tag{65}$$

TABLE IV: Statistical quantities and parameters of some simulation samples. As in the beginning of Sec. V, we set the prior probability distribution $\mathbb{P}_N = [0.15, 0.40, 0.25, 0.15, 0.05]$. The filename is used to record $(z, \lambda, P_{\rm mu}, T)$. More corpora and other quantities (such as α, β) can be found in Supplementary Data.

No.	Sample	Zipf b	r_g	SC	U_{Chain}	V_1	L
1	004_49_0015_1	0.9512	0.691	0.735	750	4121	10000
2	004_49_001_1	0.894	0.746	0.820	529	4042	10000
3	005_495_001_1	0.868	0.775	0.749	749	4095	10000
4	01_42_0001_1	0.47	0.847	0.873	290	3686	5000
5	01_43_0005_1	0.63	0.832	0.885	294	2783	5000
6	01_48_0005_1	0.649	0.853	0.867	269	2751	5000
7	01_495_0004_3	0.900	0.766	0.723	944	3994	10000
8	01_495_0011_1	0.878	0.762	0.799	849	4124	10000
9	01_49_0005_1	0.713	0.826	0.812	506	4878	10000
10	03_44_0005_1	0.54	0.806	0.848	119	2911	5000
11	03_49_0005_1	0.67	0.807	0.889	252	4994	10000
12	1_49_0005_1	0.646	0.743	0.898	98	5104	10000

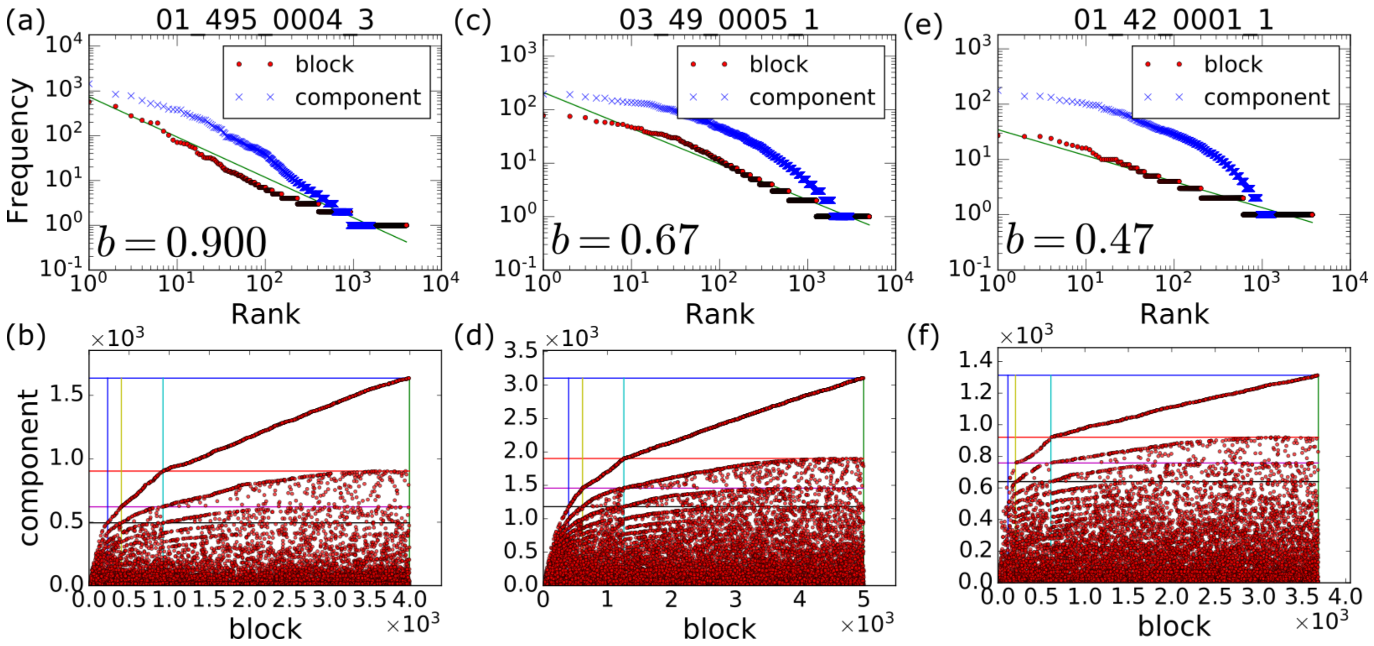


FIG. 5: The upper figures show the FRD of simulation with different parameters, while the lower ones depict their RRD.

where the ratio r is a constant of ℓ, m, n. The evidence of this nontrivial consequence is shown in Fig. 2. This formula gives two kinds of r: r_g and r_H. Most of the scaling region corresponds to small n, thus r_g covers more data than r_H. Additionally, $r_H \approx$ constant is merely a statistical outcome[54]. In conclusion, we tend to use r_g as r.

Since the mid-point of each rectangle $(\bar{x}_m, \bar{y}_n) = (\frac{V_{m+1}+V_m}{2}, \frac{H_{n+1}+H_n}{2})$ roughly falls on the scaling line, it can be shown that

$$a_\ell g(\bar{x}_m) \approx \bar{y}_\ell \approx H_1(1+\frac{1}{r})r^{\ell-1}r^{a\bar{x}_m^{-b}} \tag{66}$$

where the last approximation is done by applying Eq. (5) to r^{n-m} and frequency-index identity[32] with Zipf's law to r^m. This equation can be used to derive the form of scaling function as

$$g(x) \approx \frac{H_1}{a_1}(1+\frac{1}{r_g})r^{\rho_x} \Rightarrow y \approx A r_g^{\rho_x}. \tag{67}$$

This function actually gives a rule of block-composition related to the usage rate of components. It can be observed from the differential form of Eq. (67), $d\rho_x \propto dy/y$, that reveals the increase of block usage ($d\rho_x$) is proportional to the difference in popularity among components (dy) weighted by the inverse of their rank ($1/y$). This is similar to the rich-get-richer spirit of the preferential attachment[55] that leads to scale-free behavior. Like Sec. IV D, this conclusion was obtained in the early stage of our research. The rich-get-richer spirit eventually leads us to the concept of function connection in our evolution mechanism.

B. fitting the scaling function

We use Eq. (67) with free parameters A and a in $\rho_x(x) = ax^{-b}$ to fit the scaling lines. This procedure includes the following processes:

(a) Prepare the input.
(i) use the reconstructed points $\hat{r}$ as in Fig. 6 (a) to be data points, instead of the coarse-grained points.
(ii) the exponent b of ρ_x comes from the maximum likelihood estimation, as in Eq. (13).
(iii) the total scaling ratio r_g was calculated by Eq. (28).

(b) Filter out the data whose block rank $x \leq 0.25V_1$ (most of the scaling region.)

(c) Select g_2, g_3, and g_4 alternatively to be the base of scaling function.

(d) Fit each base by standard error optimization and use it to define other scaling lines. For example, if g_2 is the base, then $g_3 = r_g \cdot g_2$ and $g_4 = r_g^2 \cdot g_2$.

(e) Calculate total standard error of base g_ℓ,

$$Dev(g_\ell) \equiv \sum_{i=2}^{4} \sum_{x} \Big[y_i(x) - g_i(x) \Big]^2 = \sum_{i=2}^{4} \sum_{x} \Big[y_i(x) - r_g^{i-\ell} g_\ell(x) \Big]^2 \tag{68}$$

where $y_i(x)$ denotes real data.

(f) The g_ℓ that enjoys the smallest $Dev(g_\ell)$ will consequently be chosen as the best base.

(g) If 0.25 in (b) is insufficient to complete the optimization, decrease it until the optimization succeeds.

Figure 6 (b) shows the result of the above fitting processing. To avoid overfitting, the statistical method AIC is employed. It is a direct measurement of information loss and emphasizes the need to strike a balance between model simplicity and goodness of fit. According to AIC, our processes are better than trying to fit $g_2 \sim g_4$ separately with Eq. (67) because the former only uses 2 parameters but the latter needs 6.

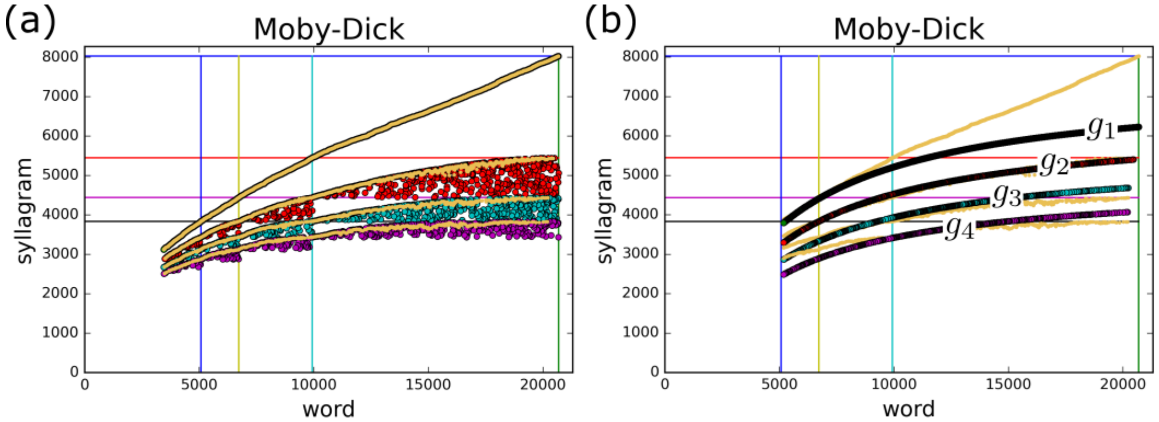


FIG. 6: (a) The yellow points compose the upper envelopes of RRD of Moby-Dick. It is the outcome of the denoising process via Eqs. (22∼26). (b) The curves $g_1 \sim g_4$ result from the scaling functions which fit the yellow reconstructed points in (a), where g_2 is the base.

Appendix A: Concave shift and locally convex problem

In Sec. IV C, we utilize the property of being concave down to locate the points on the envelopes. The term "concave down" means

$$\frac{d^2 g_\ell(x)}{dx^2} \leq 0. \tag{A1}$$

Thus we can use concave functions to describe the upper envelope. Some readers may notice there is a similar concept in geometry: **convex envelope** or **convex hull** that denotes the the smallest convex set (it would be unique) to contain a given set[56]. A possible idea to get g_ℓ is to apply convex hull algorithm in each rectangles. However, if someone really does so, the number of selected points will be too few to construct a clear upper envelope, namely, the result is under-represented and C value will be low. To avoid confusion, we use the term "upper envelope" to describe g_ℓ instead of convex envelope or convex hull. When we said "the envelope is convex/concave", it means "the envelope can be described approximately by a convex/concave function".

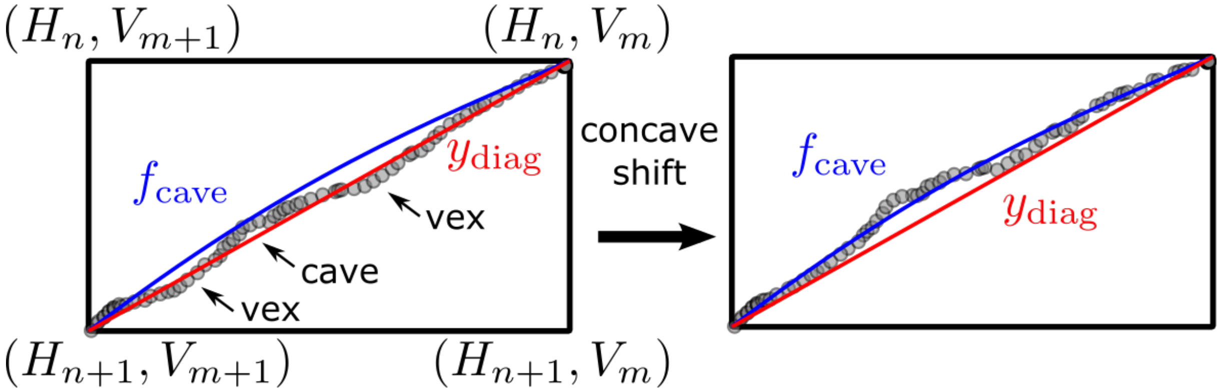


FIG. 7: A schematic of using concave shift to solve the locally convex/concave problem. The red line depicts y_{diag} in Eq. (22), while the blue one exhibits f_{cave} in Eq. (23). The abbreviation vex/cave denotes convex/concave. In the left panel, some parts of the upper envelope are under y_{diag} so that they fail the test of Eq. (22). With the aid of concave shift, the upper envelope can be selected through Eq. (22).

If all parts of the upper envelope are above y_{diag}, we can reconstruct it through Eq. (22). But if most parts of the envelope are under y_{diag}, as in Fig. 7, it is very hard to reconstruct the whole envelope. We name such issue as a locally convex problem. In our data, some life-Books have the locally convex problem, e.g., the human. In Fig. 7, we can see how concave shift helps the finding of upper envelope. The parameters $\tilde{a}$, $\tilde{b}$, and $\tilde{c}$ in Eq. (23) are obtained by assuming f_{cave} pass (V_m, H_n), (V_{m+1}, H_{n+1}), and $(\frac{1}{2}(V_m + V_{m+1}), H_{n+1} + \frac{1+\delta}{2}(H_n - H_{n+1}))$ so that

$$\begin{aligned} \tilde{a}(\delta) &= 2\delta \frac{H_{n+1} - H_n}{(V_{m+1} - V_m)^2} \\ \tilde{b}(\delta) &= \frac{H_{n+1} - H_n}{V_{m+1} - V_m} \\ \tilde{c}(\delta) &= H_n. \end{aligned} \tag{A2}$$

The function f_{cave} is concave and y_{diag} is linear. Since the second derivative of shift $D_x^2 \Delta y = D_x^2(f_{cave} - y_{\text{diag}}) \leq 0$, $\Delta y = f_{cave} - y_{\text{diag}}$ is still concave. This is why we call Δy the concave shift. The parameter δ (default $= 15\%$) is the user control parameter to decide the magnitude of Δy. The higher the δ is, so is the middle point of f_{cave}. In other words, the parabola f_{cave} is more curved to include more points near the upper envelope.

After explaining the method of concave shift, the final question is how we know most parts of the envelope are under y_{diag}. It can be done through the following steps.

(a) Divide a rectangle into N_s (default $= 2$) sections so that an upper envelope becomes N_s local envelopes.

(b) Define parameter $c_\%$ (default $= 5\%$) to evaluate how many points are needed to represent a local envelope. The least number to represent a local envelope in the rectangle $\{m, n\}$ which is divided into N_s sections will be

$$N_{\text{local}} = c_\% \times \text{the least number to represent a section} = c_\% \times \frac{V_m - V_{m+1}}{N_s}. \tag{A3}$$

(c) For each section, we use Eq. (22) to find the modified points $\{\vec{r}_0\}$. There are two situations:
(i) it is okay if $|\{\vec{r}_0\}| \geq N_{\text{local}}$ points in a section, where $|\{\vec{r}_0\}|$ denotes the cardinality of $\{\vec{r}_0\}$.
(ii) if not, it means Eq. (22) can not find enough modified points. We apply concave shift to calculate the shifted points $\vec{r}_s = \vec{r} + (0, \Delta y) = (x, y + \Delta y)$ and select those $\vec{r}$ which pass the test of Eq. (22) to be $\{\vec{r}_0\}$.

(d) Move on to the next section and repeat step (c).

For example, $V_m = 1100, V_{m+1} = 500, N_s = 4$, the section length is $(1100 - 500)/4 = 150$. If we expect that a local envelope needs at least $c_{\%} = 5\%$ to be presented, then $N_{\text{local}} = 5\% \times 150 = 7.5$. The size of $\{\vec{r}_0\}$ has to be no less than 8 to represent a local envelope.

Now we have obtained the modified points $\{\vec{r}_0\}$. The remain procedures to construct a upper envelope can be found in Sec. IV C.

[1] Black, D. L. Protein Diversity from Alternative Splicing. *Cell* **103**, 367-370 (2000).

[2] Baralle, F. E. & Giudice, J. Alternative splicing as a regulator of development and tissue identity. *Nat. Rev. Mol. Cell Biol.* **18**, 437–451 (2017).

[3] Kangxi Zidian (Tongwen Shuju edition). Available in `https://ctext.org/dictionary.pl?if=en`

[4] Hua Lin. A Grammar of Mandarin Chinese, ch. 2. Lincom Europa, Germany (2001).

[5] San, Duanmu. Chinese (Mandarin): Phonology. Encyclopedia of Language and Linguistics, 2nd Edition, p. 351-355. Oxford, UK: Elsevier Publishing House (2006).

[6] de Mejía & Anne-Marie. Power, Prestige, and Bilingualism: International Perspectives on Elite Bilingual Education. Multilingual Matters. pp. 47–49 (2002).

[7] Crystal, D. The Cambridge Encyclopedia of Language (3rd ed.), p. 173., Cambridge (2010).

[8] Gimona, M. Protein linguistics — a grammar for modular protein assembly? *Nat. Rev. Mol. Cell Biol.* **7**, 68 (2006).

[9] Doolittle, R. F. The multiplicity of domains in proteins. *Annu. Rev. Biochem.* **64**, 287 (1995).

[10] Koonin, Eugene V., Wolf, Yuri I. & Karev, Georgy P. The structure of the protein universe and genome evolution. *Nature* **420**, 218 (2002).

[11] Searls, David B. The language of genes. *Nature* **420**, 211 (2002); The Linguistics of DNA. *American Scientist* **80**, 579 (1992).

[12] Scaiewicz, Andrea & Levitt, Michael. The Language of the Protein Universe. *Curr. Opin. Genet. Dev.* **35**, 50 (2015).

[13] Mitzenmacher, M. A Brief History of Generative Models for Power Law and Lognormal Distributions. Internet Mathematics **1**, 226-251 (2004).

[14] Yu, Lijia et al. Grammar of protein domain architectures. *PNAS* **116**, 3636 (2019).

[15] Cavnar, W. B. & Trenkle, J. M. N-Gram-Based Text Categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 161-175 (Las Vegas, NV, 1994).

[16] Zipf, G. K. Human Behavior and the Principle of Least Effort (Addison-Wesley, Boston, 1949).

[17] Tsai, S. T. et al. Power-law ansatz in complex systems: Excessive loss of information. *Phys. Rev. E* **92**, 062925 (2015).

[18] Newman, M. E. J. *Power laws, Pareto distributions and Zipf's law.* Contemporary Physics **46**, 323 (2005).

[19] See `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html` for the curve fitting algorithm.

[20] Boyd, Stephen & Vandenberghe, Lieven. Convex Optimization (Cambridge University Press, Cambridge, 2009).

[21] Mandelbrot, B. The Fractal Geometry of Nature (Freeman, San Francisco, 1983).

[22] Randolph, Mark A. Syllable-based Constraints on Properties of English Sounds. Ph.D. thesis, MIT (1989).

[23] Ensembl Biomart. Retrieved from `http://www.ensembl.org/biomart/martview` (2020).

[24] Wang, Li-Min & Wu, Shan-Jyun. Genetics-Linguistics-Correspondence, Open source project on Github: `https://github.com/FireIceMan/GLC_framework`.

[25] Numba: A High Performance Python Compiler. Retrieved from `https://numba.pydata.org/` (2022).

[26] Stribling, J., Krohn, M. & Aguayo, D. SCIgen - An Automatic CS Paper Generator. Retrieved from `https://pdos.csail.mit.edu/archive/scigen/` (2017).

[27] Academia Sinica. Sinica Corpus. Retrieved from `http://godel.iis.sinica.edu.tw/CKIP/engversion/20corpus.htm` (2017).

[28] Collected by Tsai, S. T. News collection from China Times, Apple Daily, Liberty Times, etc. (2016, 2017). This text file is included in Supplementary Data.

[29] Mo Yan 莫言. *Frog* 蛙. ISBN: 9787532136766 (Shanghai Wenyi, Shanghai, 2009).

[30] Yu Guangzhong 余光中. *Selected Poetry of Yu Guangzhong* 余光中詩選. ISBN：9576742730 (Hongfan Bookstore, Taipei, 2006).

[31] Chang Show-Foong 張曉風. *Zhang Xiaofeng Prose Collection* 張曉風散文集. ISBN: 9574441407 (Chiu Ko Publishing, Taipei, 2004).

[32] For rare events, experience tells us that their ρ_x and ρ_y behave like the indices, i.e., start orderly from $1, 2, 3, ...$. When the events are frequent, the difference of their ρ_x and ρ_y between successive ranks can be bigger than 1. This leads to the breakdown of this identity. Most region in RRD are rare event and follow this identity, so that $m \approx \rho_x(\bar{x}_m)$. Combining this approximation with Zipf's law, we gain $m \approx a\bar{x}_m^{-b}$.

[33] Weisstein, Eric W. "Partition Function P." From MathWorld–A Wolfram Web Resource. `https://mathworld.wolfram.com/PartitionFunctionP.html`.

[34] Herman Melville. *Moby-Dick.* Urbana, Illinois: Project Gutenberg. Retrieved from `https://www.gutenberg.org/ebooks/2701` (2008).

[35] Tolkien, J. R. R. *The Hobbit.* ISBN: 0618260307 (HMH Books, Boston, 2002).

[36] Xu Zhimo 徐志摩. Selected poems. Retrieved from `http://w3.loxa.com.tw/fxp6033/poet01.htm` (2017).

[37] Jin Yong 金庸. *Demi-Gods and Semi-Devils* 天龍八部. ISBN: 9789573256748 (Yuan-Liou Publishing, Taipei, 1996).

[38] Abbott, B. P.*et al.* Observation of Gravitational Waves from a Binary Black Hole Merger. Phys. Rev. Lett. **116**, 061102 (2016).

[39] Tsai, S. T. *et al.* Acoustic Emission from Breaking a Bamboo Chopstick. Phys. Rev. Lett. **116**, 035501 (2016).

[40] Maillart, T. , Sornette, D. Spaeth, S. & Von Krogh, G. Empirical Tests of Zipf's law Mechanism In Open Source Linux Distribution. Phys. Rev. Lett. **101**, 218701 (2008).

[41] Wu Jingzi 吳敬梓. *The Scholar* 儒林外史. Retrieved from `https://zh.wikisource.org/zh-hant/%E5%84%92%E6%9E%97%E5%A4%96%E5%8F%B2` (2017).

[42] Lung Yingtai 龍應台. 野火集. ISBN: 9576070589 (Eurasian Press, Taipei, 1985).

[43] Haruki Murakami 村上春樹. Kafka on the Shore (Chinese version) 海邊的卡夫卡. ISBN:2966622172 (China Times Publishing, Taipei, 2017).

[44] Jin Yong 金庸. *Mandarin Duck Blades* 鴛鴦刀. ISBN: 9789573256748 (Yuan-Liou Publishing, Taipei, 1996).

[45] Ni Kuang 倪匡. Wisely 衛斯理系列-第二種人. ISBN:9576455936 (Storm & Stress Publishing, Taipei, 1995).

[46] Zheng Chou-yu 鄭愁予. Selected poems. Retrieved from `http://w3.loxa.com.tw/fxp6033/poet03.htm` (2017).

[47] Rowling, J. K. *Harry Potter and the Sorcerer's Stone.* ISBN:0439554934 (Arthur A. Levine Books, New York, 1997).

[48] Rowling, J. K. *Harry Potter and the Chamber of Secrets.* ISBN:0439064864 (Arthur A. Levine Books, New York, 1999).

[49] Rowling, J. K. *Harry Potter and the Prisoner of Azkaban.* ISBN:043965548X (Scholastic, New York, 2004).

[50] Rowling, J. K. *Harry Potter and the Goblet of Fire.* ISBN:0439139600 (Scholastic, New York, 2002).

[51] Rowling, J. K. *Harry Potter and the Order of the Phoenix.* ISBN:0439358078 (Scholastic, New York, 2002).

[52] Rowling, J. K. *Harry Potter and the Half-Blood Prince.* ISBN:0439785960 (Scholastic, New York, 2006).

[53] Rowling, J. K. *Harry Potter and the Deathly Hallows.* ISBN:0545010225 (Scholastic, New York, 2007).

[54] The $r_H \equiv H_{n+1}/H_n$ as in Fig. 2. When n is small, r_H is not a constant. But as n increases, the amount of components that share the same ρ_y becomes less and less. That means, the difference between H_{n+1} and H_n also decreases. Mathematically, $r_H = (H_n - \Delta H_n)/H_n = 1 - (\Delta H_n/H_n) \to 1$ as n increases.

[55] Barabási, A.-L. & Albert, R., *Science* **286**, 509 (1999).

[56] Wikipedia, Convex hull. retrieved from `https://en.wikipedia.org/wiki/Convex_hull` (2022).