Università di Torino

Doctoral School in Life and Health Sciences

PhD Programme in Complex Systems for Life Sciences

Extracting information from simple statistical laws in complex component systems.

Supervisor:
Michele Caselle

Student:
Andrea Mazzolini

Main collaborators:
Matteo Osella
Marco Cosentino Lagomarsino

Opponents:
Marco Gherardi
Antonio Scialdone

February 2018

Abstract

In several systems modularity is a crucial ingredient at the basis of complexity and heterogeneity. Prominent examples are genomes, which can be viewed as an assembly of genes, written texts composed of words, or man-made systems built from basic modules. Our work proposes a general theoretical tool to represent modular systems, called *component system*, where sets of objects are a collection of elementary components, exactly as LEGO toys are made of bricks. The present thesis focuses on emerging statistical patterns/laws that component systems show. These patterns, in principle, can unveil information about underlying architectural constrains, or the system generative process, thus leading to a better comprehension of modular structures. We present five works in this context, which addresses different approaches to understand the origin of these simple patterns, and what they can teach us about empirical systems. In short, the first works tackles the problem about dependencies between statistical laws, providing new insights about the well-known "U" shaped distribution of shared genes in genomes, which can be mainly considered as a derivative phenomenon of a scale-free Zipf's law. The next three works are dedicated to reproduce statistical patterns by means of stochastic growth models. In particular, we show that statistical laws of a component system can be linked to topological properties of functional dependencies between the system-components. We also investigated how such patterns can emerge as a consequence of the "sample space reducing" mechanism, i.e. a recently introduced model based on the assumption that the number of potentially new components reduces as the system evolves. The fourth work considers the growth of the object "vocabulary" as a function of its "size", well-known in linguistics as the Heaps' law. We show that empirical systems display a non-trivial and universal vocabulary's fluctuations scaling, which can be generated by specific conditions of the innovation dynamics. Finally, the fifth work tackles the problem of ranking the system-components according to their "importance" (with a similar spirit of the Google PageRank). To this end, we employ a non-linear iterative algorithm to efficiently rank species in mutualistic ecological network, according to their importance in determining the system robustness.

Acknowledgements

I wish to express my sincere gratitude to all the people, who helped me to shape this project and guided me during these three year of Phd, giving also a fundamental contribution to my personal development as a scientist. Namely, Matteo Osella, who is the guy behind most of the ideas of the work, Michele Caselle for the valuable guidance and advise, Marco Cosentino Lagomarsino and Marco Gherardi for the countless useful discussions and suggestions. I would like also to acknowledge il Nonno, Eleonora De Lazzari, Jacopo Grilli and Matteo Cereda for their collaboration.

In addition to those ones already mentioned, I sincerely thank all the other members the bio-physics group in Torino, and all the Phd students of the "basement", not only for the scientific discussions, but also for making the physics department of Torino a very friendly environment.

Finally, I would like to thank my family and all my friends from Novara and Torino for their fundamental support and help.

Contents

Chapter 1

Summary of the work

1.1 Purpose of the work

The present thesis describes the research work that I carried out during my PhD program in complex systems for life sciences. As a general context, the project studies the emerging statistical patterns/laws of complex systems. This field of research has grown in the last twenty/thirty years as a consequence of the increasing number of data that new technologies can extract from empirical systems and process. Examples of involved disciplines are biology, ecology, economics, linguistics or social sciences. Basically, the approach consists of computing simple qualities across all the "microscopic" entities, giving rise to statistical patterns which describe the system as a whole. Resembling the concept of macroscopic observables in thermodynamics, the obtained statistical laws ignore a lot of "microscopic" details, providing a global description in simple mathematical terms. This often highlights surprising regularities across very different fields, and, in principle, may unveil fundamental properties of the system and how it has been generated. For example, one can study a large ensemble of genomes and their composition in terms of gene families. This structure has been shaped by evolution, and therefore its emerging statistical properties contain information about the organisms evolutionary history and the evolutionary forces acting on them. Analogously, the statistical laws shown by the occurrences of words among books are related to fundamental properties of human language, and how it has evolved trying to optimize communication efficiency (Section 2.2 will present a more extended introduction to statistical laws).

We approached the problem of studying statistical laws proposing a general theoretical framework, called *component system.* It takes advantage of the modular structure shown by several empirical systems. For instance, genomes can be viewed as a collection of genes, exactly as books are made of words or man-made buildings are composed of basic modules. In other words, the component-system framework describes whatever ensemble of *realizations/objects* made of *elementary components.* Such a description can be applied to a huge variety of systems, belonging to very different fields. Some examples are genomics, linguistics, technological systems, biology, ecology and economy.

Several well-known statistical laws find a natural definition within the component-system representation. We can then investigate statistical patterns within a general and abstract framework, which allows us to compare the laws between very diverse fields. This may highlight general features across systems, maybe generated by statistical constrains of the modular representation, or due to universal mechanisms. On the contrary, the specificities should be related to system-specific properties, unveiling functional features. A further advantage of this approach is that it creates bridges between different disciplines. As a consequence, techniques and ideas specific of a certain field can be extended and generalized to all the modular systems.

The present thesis will describe five works within this context. Trying to summarize their common purpose in one sentence, we want to understand the origin and the meaning of the statistical and topological properties of empirical modular systems by employing the component system framework. We can then categorized the five works in three more refined topics:

- **Dependencies between statistical laws**. Considering different statistical patterns of complex systems, a first crucial question is about the dependencies between them. If they are independent, one can expect that they highlight different aspects of the system under study, otherwise laws can emerge as a statistical (or null) consequence of other laws, carrying redundant information. This problem is better addressed in the first work, "U-shaped law as a statistical consequence of the Zipf's law", which focuses on the relationship between two important statistical patterns. The work proposes a null-model which takes into account such dependency, providing new insights about the origin of so called "U"-shaped law very famous in genomics.

- **Generative process of component systems**. A classical approach to extract information form statistical laws is to build mathematical models reproducing the observed patterns with the minimal set of ingredients and parameters. If a model succeeds, it is reasonable to assume that such ingredients are at the basis of the system generative process, providing then information on how the system evolves. The next three works of this thesis, "Zipf and Heaps laws from dependency structures", "Heaps and U-shaped laws in sample space reducing processes", and "the Heaps' law fluctuations unveil information on the innovation dynamics", focus on this aspect. Specifically, the first one considers a model based on functional dependencies between components, and how the statistical patterns can emerge as a consequence of the topological properties of the dependency network. The second work focuses on a recently introduced stochastic model based on the shrinking of the state space as the system evolves, and how different laws can emerge form that assumption. Finally, the third study is dedicated to the investigation of a specific observable, namely the fluctuation scaling of the so-called Heaps' law, which is strictly related to the innovation dynamics of the system.

- **Ranking of components**. The fifth work, "towards the optimal ranking in ecological mutualistic networks", tackles the problem of ranking the system components and realizations according to a certain definition of importance. Specifically, we look for the most important species in a mutualisitic ecosystem (which can be represented as a component system) whose extinction would lead to a faster collapse of the system. To this end, we generalize a non-linear iterative algorithm recently introduced in economics.

The thesis is arranged as follows: the current chapters will continue presenting the abstracts of all the five different works. Then, chapter 2 describes the component-system framework, and presents some well-known statistical laws, focusing on how they find a representation in our framework. These concepts are at the basis of the five works, which are presented in the next five chapters: 3, 4, 5, 6, 7.

1.2 Abstracts

1.2.1 U-shaped law as a statistical consequence of the Zipf's law

Authors: Andrea Mazzolini, Marco Gherardi, Michele Caselle, Marco Cosentino Lagomarsino, Matteo Osella.

Here we focus on the statistics of shared components, i.e., the distribution of the number of basic components shared by different system-realizations, such as the number of common bricks found in different LEGO sets, common genes in different organisms or common words in different books. The shared components distribution is well-known in genomics and technological system, showing a characteristic and apparently universal "U"-shape. The common approach to understand the meaning of this distribution is to investigate generative models leading to the "U" shape under system-specific functional constraints. However, considering a simple null model based on random extractions of the component, we prove that this law can be qualitatively generated assuming the heterogeneity of the component abundances, whose broad distribution is known as Zipf's law in several contexts. Moreover, the null model provides analytical estimates of component occurrence features, such its power law decay and its "core component" fraction, and how these features are related to the Zipf's law and the system parameters. The range of validity of such predictions seems to be very wide, since they are confirmed in all the considered data: bacterial genomes, LEGO sets, and book chapters. Therefore, efficient detection of specific architectural features and system functional information from the occurrence distribution have to keep these emergent regularities into account. In this way, system-specific statistical anomalies can be identified as deviations from null predictions. These deviations can highlight functional constraints, as we show for the illustrative case of bacterial genome composition.

1.2.2 Zipf and Heaps laws from dependency structures

Authors: Andrea Mazzolini, Jacopo Grilli, Eleonora De Lazzari, Matteo Osella, Marco Cosentino Lagomarsino, Marco Gherardi.

Zipf's and Heaps' laws are broadly studied examples of statistical laws, concerning the distribution of component abundances, and their number as a function of system size. Interestingly, they show emergent regularities quantitatively invariant across very different systems. Despite the effort, the debate is still open regarding their origin, robustness, and universality. In this work, we propose a positive model, based on the concept of dependency structures between components, which constrain the statistical properties of the component system under study, leading to the two quantitative laws. Such dependency structures (i.e., networks encoding the dependency relations between the components in a system) were proposed recently as organizing principles underlying some of the regularities observed, specifically the so-called "U"-shaped component frequency distribution. However, only binary descriptions (absence/presence of components) have been utilized. Here, we consider a simple model that generates, from a given ensemble of dependency structures, a statistical ensemble of sets of components, allowing for components to appear with any multiplicity. A mean-field analytical approach (analogous to what is called Zipfian ensemble in the linguistics literature) is able to capture the relevant laws involving the occurrence and the abundance of the components, as we show by comparison with numerical computations. In particular, we recover a power-law Zipf rank plot (with a set of core components) and a Heaps law displaying three consecutive regimes (linear, sub-linear and saturating) that we characterize quantitatively.

1.2.3 Heaps and U-shaped laws in sample space reducing processes

Authors: Andrea Mazzolini, Alberto Colliva, Michele Caselle, Matteo Osella.

This work aims to analyse a generative model for complex component systems called Sample Space Reducing process, SSR. The basic model idea is that as the system evolves, the number of possible state diminishes, i.e. the state space shrinks. This provides a novel mechanism for the generation of power laws, reproducing the Zipf's law. We show that, in addition to the Zipf's law, other well-known scaling laws may emerge as a consequence of the sample space reducing assumption: the sub-linear growth of the vocabulary size as a function of the text size, i.e. the Heaps' law, and the U-shaped distribution of the fraction of realizations which share a component. Within the model framework, the behavior of both these laws can be predicted analytically, and it is in agreement with random sampling models which assume the Zipfs law. We also apply the SSR model in a specific example, trying to reproduce the Zipf's and the Heaps' laws of a book, which are qualitatively in agreement with the empirical ones. At the same time, statistical laws which look at correlations between words, showing non-trivial behaviour in books, cannot be reproduced by the model.

1.2.4 How the Heaps' law fluctuations are related to the innovation dynamics

Authors: Andrea Mazzolini, Alberto Colliva, Michele Caselle, Matteo Osella.

Even though the average behaviour of the Heaps' law has been extensively studied in linguistics, there are very few attempts to characterize the full statistics of the book vocabularies at fixed text-length. Here, tacking-advantage of the component system framework, we tackle the study of the Heaps' law variance in linguistic and genomic databases. The fluctuation scaling show a surprising Taylor's law with exponent 2, which seems to be very general in all the datasets and deviates from random-models predictions. The origin of this non-trivial scaling is analysed in the context of duplication-innovation models, allowing us to connect the Heaps' variance with the innovation dynamics. Our findings shows that the scaling is correctly reproduced by models in which the probability of discovering new components is linear with the vocabulary (i.e. the number of distinct components/words), as in the Chinese restaurant process. This suggests a rich-gets-richer mechanism in terms of vocabulary usage: the more realizations have a rich vocabulary, the more they will discover new components. This mechanism can find interesting context-specific interpretations. For example, we speculate that it may be originated by phylogenetic history in genomes, or the topical aspect of written texts in linguistics.

1.2.5 Towards the optimal ranking in ecological mutualistic networks

Authors: Andrea Mazzolini, Matteo Osella, Michele Caselle.

The stability of mutualistic ecological systems relies on a network of complex interactions between species. A challenging problem is determining the "importance" of any given species when evaluating the system resilience and robustness to perturbations. This task has been tackled with promising results by using a non-linear iterative algorithm originally introduced in economics, called *fitness-complexity map.* Specifically, from the binary component system composed of countries and exported products, the algorithm highlights the most "important" nations in terms of non-monetary competitiveness. Surprisingly, the same algorithm applied to mutualistic systems (which share the same structure as a binary component system) ranks species according to their ecological "importance" in a very efficient way. Our work proposes a one-parameter generalization of the fitness-complexity map. The free parameter allows us to tune the concept of "importance" assigned to species (or countries), which is instead fixed in the standard map. Testing the algorithm in mutualistic systems, we find that the generalization leads to much better performances, which can be quantitatively evaluated by computing the so-called extinction area. An unexpected consequence of the optimal ranking is that, rearranging the adjacency matrix of the system according to the obtained ranking, a surprising geometric pattern emerges, which seems to really capture the "nested" architecture of the system.

Chapter 2

Statistical laws in component systems

2.1 The component system framework

Several empirical systems are modular, that is to say that the complex entities of the systems can be viewed as a collection of basic building blocks. For instance, LEGO toys can be regarded as an assembly of bricks (Figure 2.1a), books are a collection of words (Fig. 2.1b), and genomes are composed of protein domain families (Fig. 2.1c). We call this structure *component system*. Modular representations of complex systems have emerged in diverse fields, some examples are genomics [1, 2], quantitative geography [3, 4], linguistics [5, 6, 7], and technological systems [2, 8]. However, these works use the modular representation to study problems within the specific contexts, without referring to a general and abstract structure. A first aim of this thesis is to define such generalization (i.e. the component system framework) and to take advantage of the bridges and links between distant fields that it creates.

The precise mathematical definition of a component system will be presented in Section 2.1.1, but in several cases such systems can be simply identified with the *component matrix*, Figure 2.2a. The matrix elements indicate the quantity of the *component* (e.g. LEGO brick, word, protein domain family) in the *realization* (e.g. LEGO toy, book, genome). Note that a component matrix can be interpreted as the adjacency matrix of a weighted bipartite network, Figure 2.2b, where the two layers of nodes are the realizations and the components.

Her we introduce an important class of component systems, which are characterized by the order of the component within the realization. For example in written texts the order is naturally defined as the position of the words in the text string (Fig. 2.2c), or in LEGO toys the order can be obtained by the time in which a brick joins the construction following the

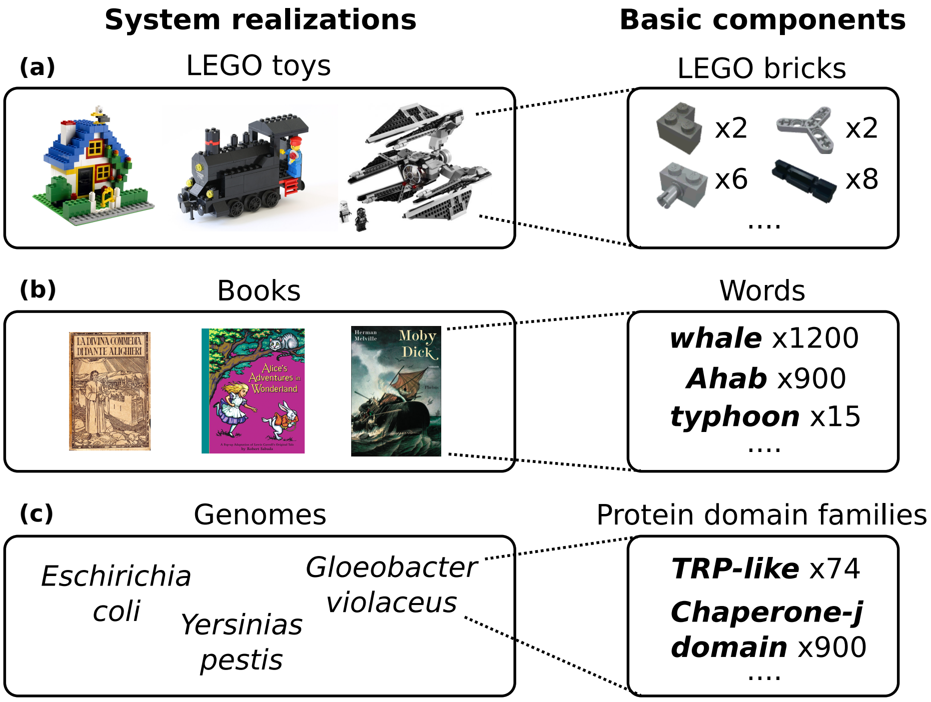
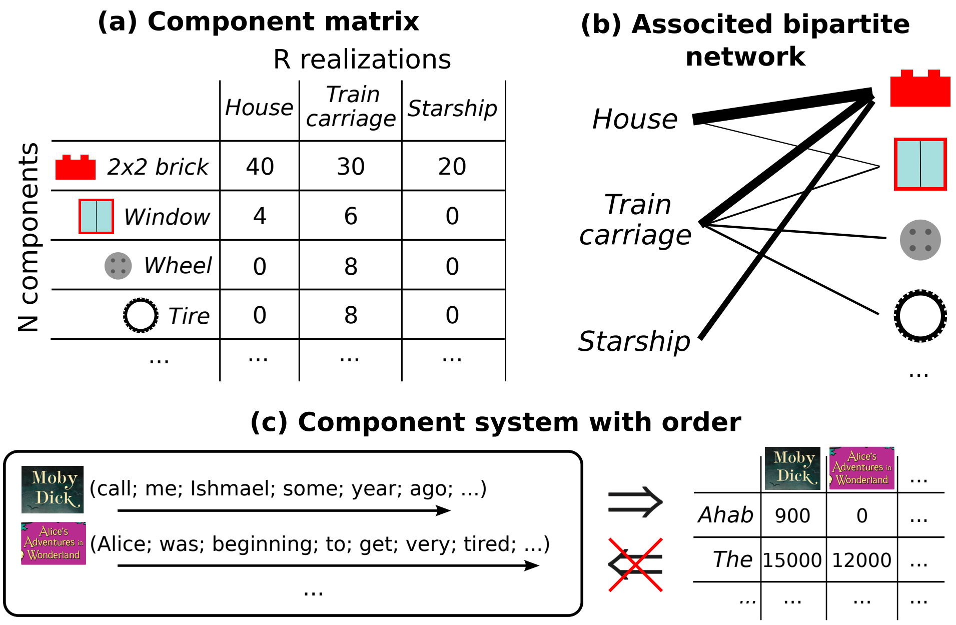


Figure 2.1: **Three examples of component systems**

instruction manual. In such cases, we refer to *component system with order* (defined in Section 2.1.3). Note that the component-order information is not encoded in the component matrix, which can be still associated to this kind of systems, but it does not provide a complete description. It is important to introduce component systems with order because some later definitions and results are defined only for this special class.

It is worth mentioning that the component-system-description leads a simplification of the original system complexity. For instance, if one describes books just as a bag of words, then he ignores the super-structures, such as chapters or paragraphs, as well as the statistics of the letters and the syllabuses. To take into account the more complex architecture of empirical systems one can generalize our framework introducing multiple matrices, each associated to a different "resolution". For example, in the cited linguistic case, the elementary components can be words, letters, or syllabuses, while the system-realizations can be books, chapters or paragraphs. Similarly, in genomics the basic components of genomes can be genes, protein domains, or whatever kind of component inside the cell. At the same time, genomes can be chosen as single species but also at different taxonomic levels. In such a way, all the different choices of the basic components and the system-realizations lead to different component matrices, implying that



(a) Component matrix

(b) Associted bipartite network

(c) Component system with order

	House	Train carriage	Starship
2x2 brick	40	30	20
Window	4	6	0
Wheel	0	8	0
Tire	0	8	0
...	...	...	...

Moby Dick (call; me; Ishmael; some; year; ago; ...)

Alice's Adventures in Wonderland (Alice; was; beginning; to; get; very; tired; ...)

...

	Moby Dick	Alice's Adventures in Wonderland	...
Ahab	900	0	...
The	15000	12000	...
...	...	...	...

Figure 2.2: **Mathematical representation of component systems.** Panel (a) shows the component matrix associated to a toy example of LEGO buildings. Each row represents a component, each column a realization, and the entries are the instances of the component in the realization. Panel (b) displays the component system representation as a weighted bipartite network, where the first layer of nodes are the realizations, the second layer the components, and the link weight indicates the abundance of the component in the realization. Finally, panel (c) shows an example component system with order, specifically a set of books. The component matrix can be derived from this representation, but the vice-versa is not true.

several matrices can be associated to the same system. The matrix-set provides, in principle, a more detailed description of the system and a more complete view of its complexity. However, when one perform a quantitative analysis of the system, he has to take into account the properties of each matrix and how they vary at the different resolutions, making the investigation much more complicated (with respect to a single matrix). Moreover, the "excess" of information could hide the essential features which one wants to study. This highlight a typical trade-off in complex system: the system representation must simplify the original complexity as much as possible (in order to allow mathematical tractability) but without loosing the essential properties of interest. Typically, a priori, there is not a general answer about the right "level of simplification" to chose (the most detailed description is not always the optimal choice), and it depends on the features of interest.

In this thesis we will show that a single component matrix contains a large amount of information by itself, and our analysis will consider only this simplest description. However the study of multiple "resolutions" could provide an interesting extension of the framework.

2.1.1 Mathematical notation

We define a component system as a set of R realizations: $\{r_1, r_2, \ldots, r_R\}$. Each realization is a set: $r_j = \{x_1, x_2, \ldots, x_{s_j}\}$, where s_j is the size of the realization. The variables x_k are instances of the elementary components: $x_k \in \{c_1, c_2, \ldots, c_N\}$. The set of unique components $\{c_1, c_2, \ldots, c_N\}$ is called *system vocabulary*, and its cardinality, N is the *system vocabulary size.* To illustrate this notation with an example, let us consider an ensemble of books. Each book is a realization r_j, which is a string of word-tokens. Then, each word-token, x_k, is an instance of the unique words-types which compose the vocabulary $\{c_i\}$.

This set of realizations can be completely described by the associated *component matrix.* The matrix elements are defined through the relation: $n_{ij} = \sum_{x_j \in \{x_1, \ldots, x_{s_j}\}} \delta_{x_j, c_i}$. Therefore n_{ij}, with $i = 1, \ldots, N$, $j = 1, \ldots, R$, represents the number of instances of the component i in the realization j, called also "local abundance". In Figure 2.2a is shown an illustrative example.

In the following, we present some definitions based on the component matrix $\{n_{ij}\}$, which are summarized in the table 2.1. The *global abundance* of a component is defined as the total number of times that the component appears in the ensemble: $a_i = \sum_j n_{ij}$. Considering the analogy with a bipartite network, Figure 2.2, the abundance is exactly the strength of the node. The abundance normalized by the total number of components is called frequency: $f_i = \frac{a_i}{\sum_i a_i}$. A second observable is the component *occurrence* o_i, which is instead defined as the fraction of realizations in which the component is found, thus $o_i = \frac{1}{R} \sum_j (1 - \delta_{n_{ij},0})$. The occurrence is proportional to the node degree of the associated bipartite network. Considering the realizations, the counterpart of the abundance is the realization *size*: $s_j = \sum_i n_{ij}$, which represents the total number of components in the realization j. Note that sometimes it will be used l instead of s. The distinction is that s refers to the final size of a realization, while l is the "partial" size that a realization have during a dynamical growth process, $l \in [1, s]$. Finally, the last quantity is the number of distinct component in a realization, called realization *vocabulary*: $v_j = \sum_i (1 - \delta_{n_{ij},0})$. Again, we change the notation for the partial vocabulary during a growth process, using the letter h, with $h \in [1, v]$.

Table 2.1: **Some key quantities of a component system.**

	Symbol	Formula
System vocabulary size	N	
Number of realizations	R	
Component abundance	a_i	$\sum_{j=1}^{R} n_{ij}$
Component frequency	f_i	$\frac{a_i}{\sum_{i=1}^{N} a_i}$
Component occurrence	o_i	$\frac{1}{R}\sum_{j=1}^{R}(1-\delta_{n_{ij},0})$
Realization size	s_j (or l_j)	$\sum_{i=1}^{N} n_{ij}$
Realization vocabulary	v_j (or h_j)	$\sum_{i=1}^{N}(1-\delta_{n_{ij},0})$

2.1.2 Binary component systems

This subsection introduces a special case of component system called *binary component system*. It is defined by the fact that a component in a realization can be only present or absent, implying that it is described by a binary matrix, $n_{ij} \in \{0,1\}$. A typical example is a group of personal computers (i.e. the system-realizations) and the software packages installed on them (i.e. the system-components). Each software cannot be installed more than one time, implying that the system is binary. Furthermore, whatever component system can be "binarized" choosing a threshold θ on the matrix elements, and fixing the new elements as $n'_{ij} = H(n_{ij} - \theta)$, where $H(x)$ is the Heaviside function. It is important to note that a binary component system has the following properties: the component occurrence is proportional to the abundance, $a_i = Ro_i$, and the realization size is equal to the realization vocabulary, $s_j = v_j$. As a consequence, a couple of statistical laws (introduced later) are trivial, and several results derived in this thesis become trivial.

2.1.3 Component order in realizations

As discussed previously, some systems have the additional property that their components are ordered. A typical example is an ensemble of written texts as shown in Fig. 2.2c. In a peak of creativity, we call this class of systems *component systems with order*. This is defined considering the system-realization as an ordered sequence of component-instances (instead as a set of component-instances): $(x_1, x_2, \ldots, x_{s_j})$, where, as before, $x_k \in \{c_1, c_2, \ldots, c_N\}$. Clearly the component matrix can be derived from the sequences ($n_{ij} = \sum_{x_j \in (x_1,\ldots,x_{s_j})} \delta_{x_j,c_i}$), but the vice-versa is not true. Indeed, a different order of a realization-sequence is associated to the same column of the component matrix. Note that a component system with order is no longer equivalent to a bipartite network.

Using the component-matrix-definition or the ordered-sequence-definition will depend on the system under study (whether or not the component order is well defined) and the property under analysis (if it is related with the component order). For the sake of simplicity, we will use the component-matrix definition whenever it will be possible.

Systems with partial order

This thesis will discuss only the two classes of component systems described so far, that are systems with order, and without order. However, they can be viewed as two extremes of a more general mathematical structure, which will be briefly presented in this paragraph, and could provide a really interesting generalization of the framework. We can refer to this generalization as *component system with partial order*, because it considers realizations as *partially ordered sets* [9]. Trying to give the basic ideas of this mathematical object, the set $r = \{x_1, x_2, \ldots, x_s\}$ (i.e. the realization composed of its component instances) can be associated to a second set named partial order P. The elements of this second set are binary relations between the elements of the first set r. These binary relations (or "order relations") are indicated with $x_i < x_j$, where the symbol "<" can be interpreted as x_i is followed by x_j. Note that the meaning of numerical inequality typically associated to "<" can be viewed as a particular case of this definition. To be a "partial order", P must satisfy the condition of *antisymmetry* (if $x_i < x_j$ then the element $x_j < x_i$ does not exist) and *transitivity* (if $x_i < x_j$ and $x_j < x_k$ then $x_i < x_k$). Given such a definition, the component system without order can be recovered simply considering an empty set of order relations P. Differently, the component systems with order is obtained when there exists an order relation between every pair of elements of r, defining the so-called *total order*, which implies that all the elements can be written in a unique ordered sequence, as the words in a book. An important example of totally ordered set is $\mathbb{N}$, where the order relations are the numerical inequalities ($0 < 1$, $0 < 2$, $1 < 2$, ...), which allows the natural numbers to be written in the sorted sequence: 0, 1, 2, 3, ...

Therefore, the intermediate case (between the total and the absence of order) provides a less constraining definition of order, which could find application also in the component system framework. For example, LEGO toys are perfect examples of realizations with partial order. Think for instance to a small village composed of different houses made of LEGO bricks. Bricks needed to build a certain house have order relations (the doors and the walls must be built before the roof). On the contrary, the bricks belonging to different houses do not have such relations. This follows from the fact that the final village can be obtained independently of which house is built first. Therefore the component system with partial order can potentially provide a more reliable description of systems in several contexts. One can wonder, for example, whether its properties can be translate into statistical laws, and than compared between different systems unveiling crucial information and regularities.

2.2 Statistical laws in complex systems

A possible way to characterize a complex system is looking at its global statistical patterns. These patterns can be defined as distributions of simple quantities computed among all the entities of the system, or specific dependencies between such quantities. This approach has been widely used across different fields. For example, in genomics the increasing number of complete genome sequences has made possible the computation of statistical patterns across very different organisms. This has led to the discover of "universal" laws [10], in the sense that they seem to be conserved in every region of the phylogenetic tree. Some examples are the log-normal distribution of the evolutionary rates [11], the power law distributions of gene family sizes [12], or the category-dependent power law scaling of the number of genes in a category with the genome size [13]. These universal regularities can be reproduced with simple mathematical models which mimic the evolutionary process, linking therefore the universal patterns to fundamental evolutionary mechanisms [10, 14, 15, 1, 16].

The emerging statistical patterns have been extensively investigated also in quantitative linguistics [5, 17]. The most famous example is the Zipf's law, studied in the '30s by George Zipf [18]. It states that the ranked abundances of words follow a power law decay with exponent -1, i.e. the abundance of the r-th most abundant word in a text is inversely proportional to its rank r (this law and its interpretation will be described more in detail in section 2.2.1). Other examples are the Menzerath-Altmann law [19], according to which the increase of a linguistic construct results in a decrease of its constituents, and the Heaps' law [20], which states that the number of different words in a text scales with the text size as a sub-linear power law. In addition to genomics and linguistics, statistical laws has been studied and characterized in countless fields, such as ecology [21, 22], economics [23], technological systems [24, 8], and social systems [25].

The main purpose of this thesis is to employ the component system framework to address the study of these statistical patterns, indeed, most of the cited laws can be defined within our representation. In this way, such laws can be studied from a more general perspective, comparing their features between all the modular systems. Such comparison may highlight general behaviours related to universal mechanisms, as well as context-dependent features, which should lead to a better understanding of the system-specific functional properties. Furthermore, the developed common theoretical language can help the exchange of ideas, models and data-analysis techniques between distant fields and communities of researchers.

The following subsections describe three important statistical laws and how they find a description within the component system framework.

2.2.1 The Zipf's law

As anticipated above, the Zipf's law is one of the most famous statistical laws, introduced in linguistics in the '30s. Using the component system notation, one can define two versions of the Zipf's law. The first definition, probably the most common one, regards the rank plot of the component local-abundances in a single realization, panels (a), (b), (c) of Figure 2.3. Specifically, given the list of the number of instances of each component in the realization j, sorted in descending order: $(n_{1j}, n_{2j}, \dots, n_{v_j j})$, where $n_{rj} \geq n_{r'j}, \forall r, r'$, the rank plot is defined as the variation of n_{rj} as a function of the rank r (i.e. the position in the sorted list). The observation made by George Zipf [18] is that the number of times that a word appears in the text j is inversely proportional to its rank:

$$n_{rj} \propto \frac{1}{r} \tag{2.1}$$

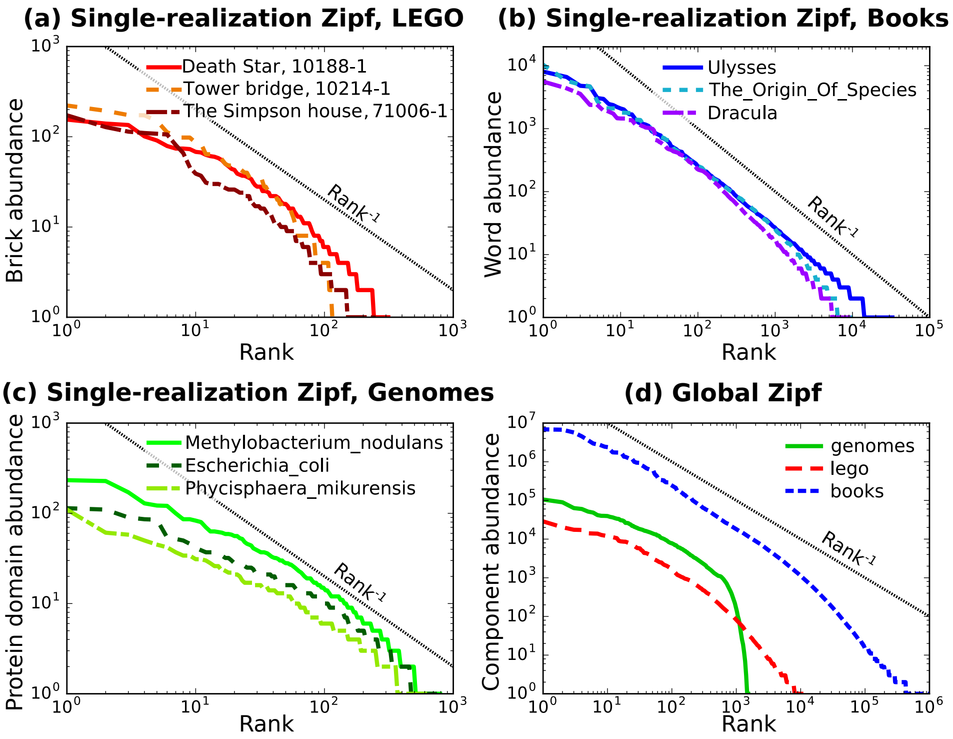
One can also define the Zipf's law of the entire component system, Figure 2.3d, as the rank plot of the global component abundances a_i (see table 2.1), where, again, the Zipf's expectation is: $a_r \propto r^{-1}$.

Looking at the Figure 2.3, it is evident that the power law decay can deviate from the classical prediction. Indeed, modern formulations consider the Zipf's law as a power law function with a generic exponent less than -1 [5]:

$$n_{rj} \propto \frac{1}{r^\gamma}, \qquad \gamma \geq 1 \tag{2.2}$$

However, even this expression is not satisfactory in several cases. In literature there are several work looking for the most suitable generalization. We do not enter into this discussion, but we cite a common definition used in linguistics [7], which is a double power law, where the scaling of the low ranks is r^{-1}, while for higher ranks the exponent becomes smaller.

The intriguing aspect of the law is that its shape seems to be qualitative similar across a wide variety of complex systems [26, 27]. Indeed, in addition to the words in texts, the Zipf's law is shown also, for example, by the population in different cities [26], the gene family sizes in genomes



(a) Single-realization Zipf, LEGO

(b) Single-realization Zipf, Books

(c) Single-realization Zipf, Genomes

(d) Global Zipf

Figure 2.3: **Global and single-realization Zipf's law** The panels (a), (b), (c) show the single-realization Zipf's law for three examples of three datasets: 2613 Lego toys, 3036 books (Gutenberg project), and an ensemble of 1060 bacterial genomes (the datasets are described in details in the appendix A). All the empirical curves are compared with the theoretical slope r^{-1} (black dotted lines). Panel (d): global Zipf's law as the global abundance rank plot (ranked list of the variable a_i in the table 2.1) for the three considered datasets.

[12], the number of papers that scientists write [28], and the firm sizes [29]. From the law's discovery until now, a large body of theoretical work have addressed the origin of the law, and, in general, of power law distributions. Several models are based on stochastic processes which mimic the growth of the system, and try to reproduce the component usage heterogeneity using the most simple possible ingredients. In this regard, the most famous mechanism is *preferential attachment*, first introduced with the Yule-Simon process [30, 31], and extensively used, for instance, at the basis of evolutionary models in genomics [32, 15, 33], or for text generation [34, 7]. Other examples are the random typewriter model [35], models based on multiplicative processes [36, 37, 38], or the sample space reducing mechanism [39]. A second different approach explains the Zipf's law assuming optimization principles: regardless of the microscopic growth dynamics, the system aims

to optimize a certain function which, in turns, imposes constraints on the system statistical properties. For example, in linguistics the optimization of the communication efficiency leads to the power law of the word abundance rank plot [40, 41]. Similar arguments have been also applied in other fields, where the system tries to maximize its performance/robustness under uncertain conditions [42, 43]. A third approach refers to the concept of *self-organized criticality*. Generally speaking, this approach states that complex systems naturally pose themselves in a critical state, showing similar properties of statistical mechanical systems near the phase transition [44]. Within this framework the Zipf's emerges as the probability distribution of the microscopic states, which takes a power law shape [45]. Finally, we cite a couple of recent works which study the Zipf's law in a really general setting, deriving it as a consequence of a random partitioning of items into groups [46], or due to the presence of fluctuating hidden variables [47].

Even though the list above is far from being exhaustive, it provides a sketch of the wide variety of different mechanisms proposed in the last 30-40 years to better understand Zipf's laws. This reflects the great interest of the scientific community about the law universality, which is still central in the current debate in complex systems.

2.2.2 The Heaps' law

A second famous law of quantitative linguistics is the Heaps' law (called also Herdan's law), which is typically defined as the sub-linear power-law scaling of the number of distinct words with the text size [20]. Using the component system notation, this law can be defined from a single-realization perspective and from the global one (analogously to the Zipf). It is important to point out that the single-realization definition requires that the order of the components within the realization is well-defined (see paragraph 2.1.3). Then, given a realization as $(x_1, \dots, x_{s_j})$, the scaling of the number of distinct components, h, in the first l components of the sequence (according to the given order) defines the Heaps' law, which has the following behaviour:

$$h(l) \propto l^{\mu}, \qquad 0 < \mu < 1 \tag{2.3}$$

where $1 \leq l \leq s_j$, $1 \leq h \leq v_j$. The maximal values v_j and s_j refer to the final vocabulary and the final size of the realization (see table 2.1). Some empirical examples are shown in Figure 2.4a.

The global Heaps' law has two alternative definitions. The first one is the scatter-plot of the realizations sizes $\{s_j\}$ (on the x-axis) versus the realization vocabularies $\{v_j\}$ (y-axis), as shown in the Figure 2.4b,c,d. Note that each point corresponds to a realization, and its coordinates, (s_j, v_j), identify exactly the final point of the single-realization vocabulary trajectory: $v_j = h(l)|_{l=s_j}$ (if it can be defined). Importantly, realization-sizes and

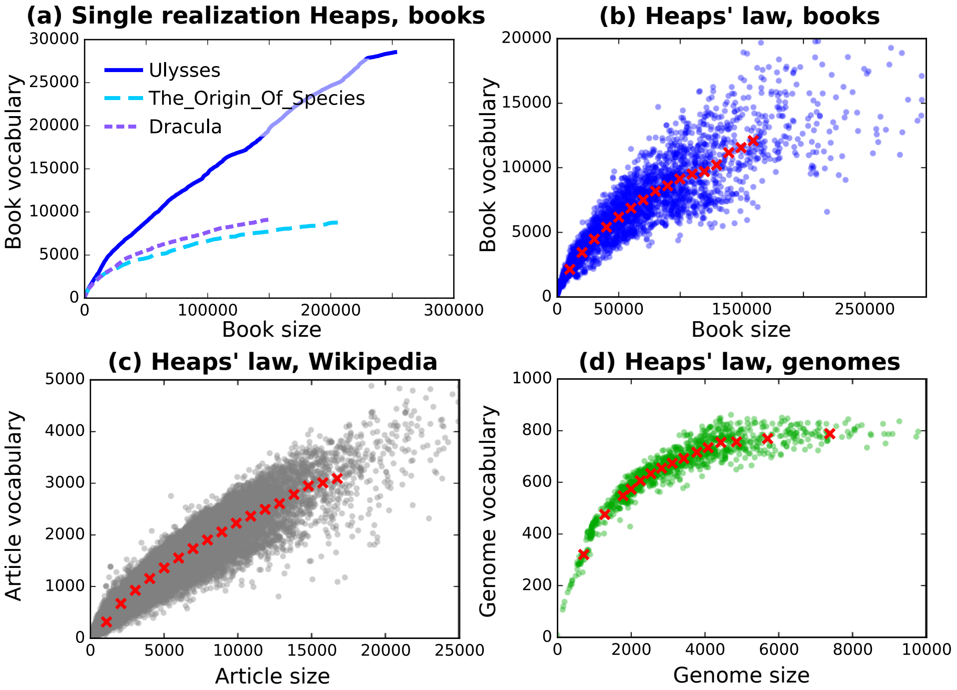


Figure 2.4: **Single-realization and global Heaps' law** The panel (a) shows the single-realization Heaps for three books ("Ulysses" by James Joyce, "The origin of species" by Charles Darwin and "Dracula" by Bram Stoker). The lines correspond to the number of distinct word in the first red words. The panels (b), (c), (d) display the global Heaps for three datasets. Genomes and books are to the same data used in the Figure 2.3. The details of the Wikipedia dataset can be found in the appendix A as well. The dots correspond to the system-realizations having a certain size (x-axis) and a certain vocabulary (y-axes), while the red crosses are the average of the cloud.

realization-vocabularies can be derived from the component matrix, implying that this definition does not require the component-order information. This property is crucial to extend the law to all the systems where the component-order is not well defined, for example in genomes ([33], and Fig. 2.4d). On the contrary, when the component-order is known, one can define the global Heaps' law as the ensemble of all the single-realization trajectories. This second definition is much more informative than the size-vocabulary scatter plot. Indeed, if one wants to study the statistics of h at a given size l, here the number of samples is equal to the number of realizations having size larger than l (while the former definition contributes only with the realizations having exactly size l).

Since there are two alternative definitions of the global law, the question about their equivalence immediately arises. Even though a rigorous mathematical analysis for this purpose is lacking, in linguistic datasets it seems to be true (as discussed in A.2.4). Because of that, for the rest of the thesis we will assume this equivalence, employing the first or the second definition depending on the convenience of the moment (the first can be extended to all the "non-ordered" systems, the second leads to a larger statistics). As for the single-realization Heaps, the average of the global law is expected to shows the same scaling of the equation (2.3).

The Heaps' law has gained interest in the scientific community because of its universal behaviour, becoming one of the most important statisical laws in quantitative linguistics [20, 5], and infometrics [48]. It is worth mentioning that this law has been observed also in the occurrences of tags for online resources [49], the keywords of scientific papers [50], the growth of the vocabulary in modern Java, C++ and C programs [51], and even in the protein domain families in genomes [33]. In addition to that, this law finds some concrete applications, for example in infometrics, where it was used for optimization of the memory allocation [52], and also in linguistics for the estimation of the vocabulary size of language [53].

Several works try to address the law origin. In this regard, we can classify these attempts into two categories. The first approach considers stochastic growth models, trying to generate the Heaps' law with the minimal set of ingredients (similar to the stochastic models discussed in the previous section for the Zipf). In particular, a generalization of the Yule-Simon model can lead to the observed scaling of the Heaps' law and Zipf's law at the same time [34]. A further generalization of the model [7], which introduces the distinction between core and non-core words, leads to a more accurate fit the two empirical laws, reproducing the observed "double-scaling" behaviour. In genomics, the Chinese Restaurant Process (CRP) was used to reproduce the global Heaps' law and the single-realiazation-Zipf shown by the protein domain families in genomes [33]. The second approach considers the Heaps' law as a derivative phenomenon of the Zipf's law [54]. Some of these models are called "Zipfian ensembles", and assume that the occurrence of every possible word is governed by random extractions [55, 6] (or a Poisson process [56]) with probabilities proportional to the word abundances. Since the word-abundance statistics is coded by the Zipf's law, the Heaps' law emerges as a statistical consequence of the power law Zipf. Under this setting, it can be proven the exponents of the Zipf, γ (equation (2.2)), and the Heaps' law, μ (equation (2.3)), are asymptotically connected by a simple expression: $\gamma = \frac{1}{\mu}$.

2.2.3 The distribution of shared components

The third and final law introduced in this section is known in genomics as "gene frequency distribution" or "U"-shaped law [57, 2]. Note that in the section 2.1.1 we already used the term “frequency” with a different meaning (as the normalized component abundance). Therefore, for the rest of the thesis we will refer to the “U”-shaped law as the “distribution of shared components” or “occurrence distribution”. As this latter name suggests, within the component system framework the law is defined as the probability distribution of the variable “occurrence” (table 2.1). Specifically, the occurrence of a component is the fraction of realizations in which the component is present, or, in other words, the fraction of realizations which share the component (e.g. $o_i = 1$ means that i is present in all the realizations, while if $o_i = \frac{1}{R}$, the component is present only in one object of the system).

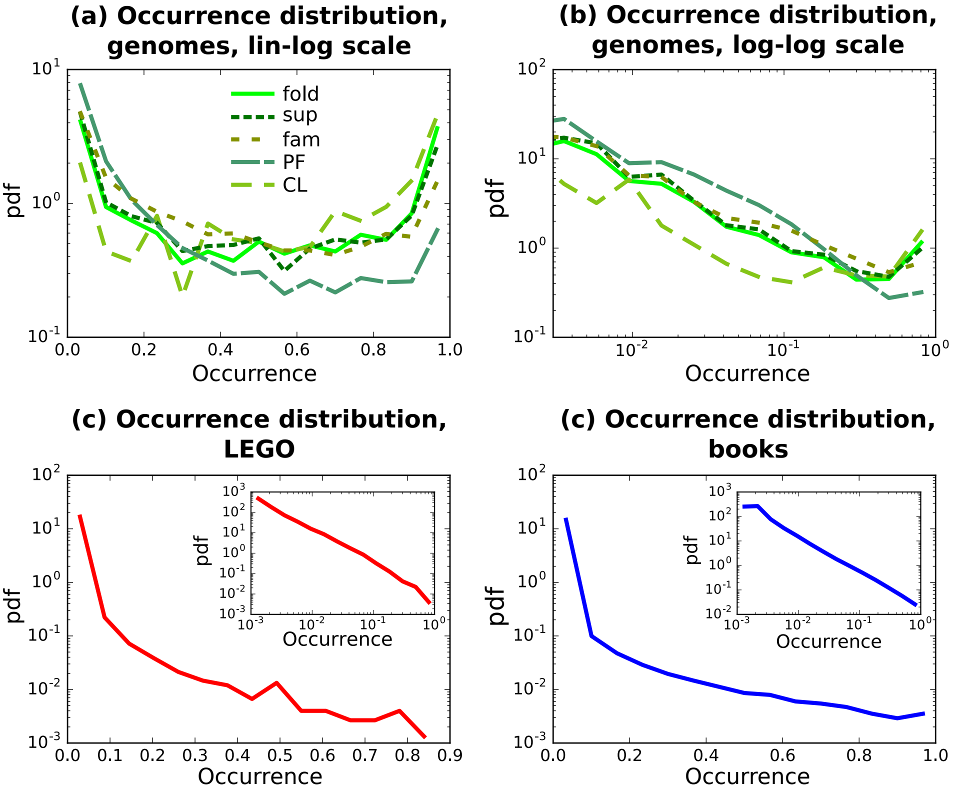


Figure 2.5: **Distribution of shared components.** The distribution is shown for the genome dataset, panel (a) (lin-log scale) and (b) (log-log scale), where we considered five different classifications of the protein domain families (see Appendix A for the details). Panels (c) and (d) show the distribution of shared components for the LEGO and the book datasets respectively, where the insets display the same curve in log-log scale.

The distribution of shared components is well-known in genomics because of its characteristic “U” shape (Figure 2.5a), which seems to be universal across several taxonomic levels [57, 58, 59, 60]. This shape implies that there are two enriched groups of basic components (typically protein domain families): the rare ones, which are present in a small fraction of genomes (occurrence near 0, peak on the left), and the common/core families, which tend to be present in all the species (occurrence near 1, peak on the right). From the Figure 2.5a it is clear that the shape is conserved even for different classifications of the protein domain families (the five datasets are described in the appendix A). The panels (c) of Figure 2.5 shows that the occurrence distribution in LEGO toys does not present this characteristic shape, while the linguistic dataset, panel (d), displays an asymmetric shape with a very small peak associated to the core words. The second general feature is the power law decay at low values of the component occurrence [2], which is present in all the considered data (Figure 2.5b,c,d), and also in technological systems [2].

In evolutionary genomics, the origins of this statistical law are still under discussion, and the debate is mainly focused on the importance of natural selection in shaping this pattern. In particular, the “U”-shaped law has been rationalized theoretically by population dynamics models combining birth-death and gene transfer events [59], evolutionary models on the phylogenetic tree which assume the “infinitely many genes” hypotheses [61, 62], infinite allele models based on the distribution of gene replacement rates and the phylogenetic tree [60], or as a consequence of functional dependencies among different components [2].

For component systems outside of genomics, the distribution of shared components remains under-explored.

Chapter 3

U-shaped law as a statistical consequence of the Zipf's law

Authors: Andrea Mazzolini, Marco Gherardi, Michele Caselle, Marco Cosentino Lagomarsino, Matteo Osella.

3.1 Introduction

This first work is dedicated to the distribution of shared components (or component occurrence distribution), described in section 2.2.3. As said before, this statistical law is well-known in genomics [57, 58, 59, 60], presenting a power law decay followed by a peak at maximal occurrence. This peculiar "U"-shape seems to be a very general feature of genomic datasets, giving rise to questions about its origin and its relationship with the evolutionary process. Almost all the attempts to reproduce the empirical distribution focus on the system generative models [59, 61, 62, 60], trying to identify the crucial mechanisms of evolution at the basis of the law shape. Here we employ an alternative approach. First, we study the occurrence distribution within the component system framework, allowing us to generalize the law and to compare its features between systems of very different origin, such as texts or LEGO toys (see Figure 2.5 and 3.1a-d). Second, instead of focusing on the system-specific generative process, we claim that this law is a general derivative phenomenon of the component abundance statistics (i.e. the Zipf's law). In particular, using theoretical model based on random sampling of components from their overall abundances, we show that a distribution of shared components with a power-law behavior naturally emerges, and its properties are mainly fixed by the component abundance statistics. In other words, given a component system with a heterogeneous component usage (i.e. a power law Zipf, but also other broad distributions, as shown later), the U-shaped law arises as a statistical consequence, regardless of the system generative process. These predictions are confirmed in the three analyzed datasets: an ensemble of genomes (described in the appendix A.1.1), a set of book chapters (appendix A.2.1) and a group of LEGO toys (appendix A.3), suggesting that the strong relationship between occurrence and abundance statistics is a general phenomenon of any modular systems.

The third important point of this work is that, even though the distribution generated from the random sampling model provides a very good approximation of the empirical law, there can be small deviations. In this regard, it is useful to consider the model-distribution as a first order approximation which takes into account the system "structural information" (i.e. the abundance statistics and a couple of other properties described later). The small deviations between the model and the data are then independent of this general information, and highlight system-specific and functional information.

This concepts are presented in following sections. The first one, 3.2, describes the random sampling model, how it is related to the abundance statistics, and how it reproduces the empirical distribution of shared components through simulations. Section 3.3 focuses on analytic calculations, which allows us to better understand how the features of the distribution of shared components (e.g. the power law decay and the peak at maximal occurrence) are related to the Zipf's law and other system properties. Then, section 3.4 tests this analytic predictions on the datasets, and, finally, in 3.5, we show an example of deviation between the model and the empirical law in genomics, which highlight a functional property of the system.

3.2 The random sampling model generates the distribution of shared components

As anticipated in the introduction, in order to identify the statistical consequences of the abundance statistics on the distribution of shared components, a suitable model is needed. Referring to the component system framework introduced in the section 2.1, we would like to generate R system realizations starting from a fixed component frequency set $\{f_i\}$, $i = 1, \ldots, N$. To this end, we employ a random-sampling procedure, similar to the linguistic models used to generate the Heaps' law as a consequence of the Zipf's law [55, 54, 56, 6]. The artificial realizations are built through an iterative random extraction (with replacement) of components from their frequencies $\{f_i\}$ (note that the frequencies can be considered probabilities, since $f_i > 1$ and $\sum_i f_i = 1$). Each realization size s specifies the number of random extractions. Therefore, given the list of component frequencies $\{f_i\}$ and the list of realization sizes $\{s_j\}$, the random sampling model generates a new system without any additional functional information or constraint. In Figure 3.1e is shown a schematic representation of the model.

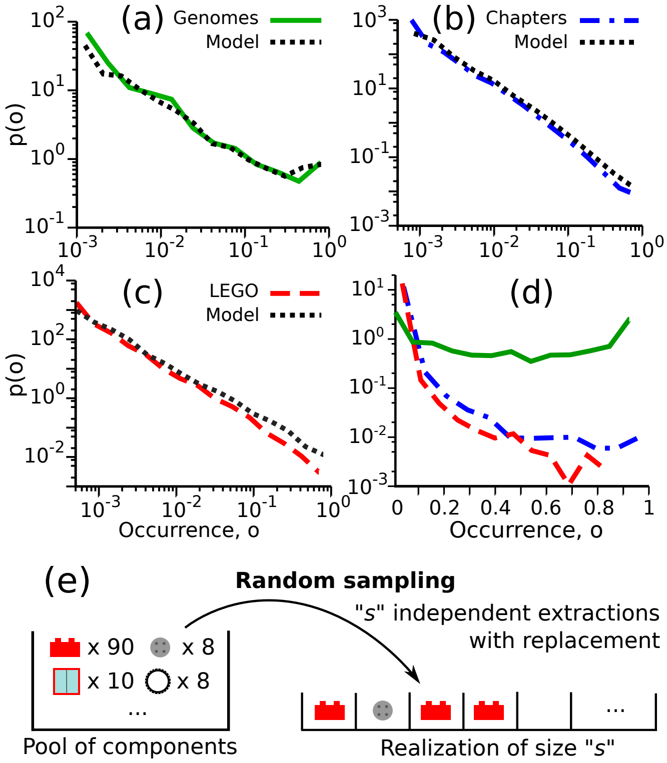


Figure 3.1: **The random-sampling model captures the main features of the empirical statistics of shared components.** The plots show the normalized distribution $p(o)$ of component occurrences for the three datasets: genomes (a) book chapters (b) and LEGO sets (c). The log-log scale highlights the power-law like decay. The black dashed lines represent the prediction of the random-sampling model assuming the empirical component frequencies and realization sizes. The model reproduces very well the power-law decay, but may differ quantitatively from the empirical laws in the high-occurrence region. Panel (d) plots the same quantities in log-lin scale, to highlight the quantitative differences between systems and the presence/absence of a peak of core components. (e) Scheme of the random-sampling process: samples of size s are generated from independent draws from the "universe" of all possible components with their specific abundances. Therefore, the probability of a component extraction is proportional to its global abundance, i.e., the sum of its abundances over all realizations of the systems.

Figure 3.1 compares the empirical occurrence distributions with simulations of a random sampling, where the frequency and size lists are chosen equal to the empirical ones. The null-model curves (dashed lines) provide very good approximations of the empirical laws, particularly for low component occurrences. Additionally, the model matches well the power law decay with the system-specific exponent. Finally, the model predicts also the qualitative behaviour of core components (those with high occurrence), and specifically that only genomes show a U-shaped distribution of shared components. The relative core sizes of the three systems are also well approximated, although there are some quantitative deviations from the empirical values that will be addressed in detail in section 3.5. These results suggest that the shape of the distribution of shared components in the three widely different empirical systems considered here is well described by a random-sampling model that only conserves the empirical component frequencies, and the realization sizes.

3.3 On the origin of the "U"-shape

This section is dedicated to understand what kind of "U"-shaped law can be expected for a given distribution of component frequencies. To address this question, we have computed analytically the distribution of shared components under general prescriptions for the component frequency distributions within the random-sampling model.

3.3.1 Occurrence distribution in the random sampling model

The iterated extractions of the model define a multinomial process, according to which the probability of a specific configuration $(n_1, n_2, \dots, n_N)$, where n_i is the number of the components with frequency f_i, is:

$$P(n_1, n_2, \dots, n_N; s) = \frac{s!}{\prod_{i=1}^{N} n_i!} \prod_{i=1}^{N} f_i^{n_i} \ , \tag{3.1}$$

under the constraint that $\sum_{i=1}^{N} n_i = s$. Since the marginal distribution for a single component is a binomial one, the probability q_i that a component of rank i is present in a realization of size s_j is $q_i(s_j) = 1-(1-f_i)^{s_j}$. Therefore, the expectation value for the occurrence of component i (see table 2.1) over a set of R realizations is:

$$o_i = \frac{1}{R}\sum_{j=1}^{R} q_i(s_j) = 1 - \frac{1}{R}\sum_{j=1}^{R}(1-f_i)^{s_j} \ . \tag{3.2}$$

In order to obtain the probability distribution associated to this rank representation, one can use the fact that the rank of a component with occurrence o is the number of components with occurrence higher than o. In fact, these naturally correspond to components with higher frequency and thus lower rank. Therefore, we can write the rank $i(o)$ as

$$i(o) = \text{rank}(o) = \sum_{o'=o}^{o_1} Np(o') \simeq N \int_{o}^{o_1} p(o')do' \ , \tag{3.3}$$

where o_1 is the highest possible occurrence, which corresponds to the component of rank 1. The function $i(o)$ is simply the inverse function of Eq. (3.2). From the approximate integral representation of $i(o)$, the occurrence probability distribution $p(o)$ is defined by the simple relation $\frac{di(o)}{do} = -Np(o)$.

Occurrence constraints

This paragraph is dedicated to understand the range of values in which the variable occurrence is correctly defined for a generic component system (not necessarily a random sampling). Clearly the occurrence assumes values in the interval $[1/R, 1]$, i.e. the minimal value is the presence in a single realization, while $o = 1$ if the component appears in all the realizations. However, given a certain list of frequencies and sizes, additional constraints are in place. The first one is a frequency-dependent upper boundary: if a component has abundance a, then it cannot appear in a number of realizations greater than a, implying $Ro \leq a$. Since the frequency is $f_i = a_i / \sum_j s_j$, and the occurrence cannot exceed 1, this translates into the relation:

$$o_i \leq \min\{ \langle s \rangle f_i \ , \ 1 \}, \tag{3.4}$$

where $\langle s \rangle$ is the average size. It can be noted that the expected value of the random sampling formula 3.2 for very small f can be approximated to $o_i \approx \langle s \rangle f_i$, overlapping the upper boundary. This is due to the fact that it is very unlikely that a component with very few instances appear more than one time in a single realization.

The second constraint defines instead a lower boundary: if the abundance is greater than the sum of the sizes of the n most large realizations, than the component must be present in at least n realizations. An implicit expression for this condition is:

$$\text{if } \sum_{k=1}^{n} s_k < a_i \leq \sum_{k=1}^{n+1} s_k \ \text{ then } \ Ro > n, \tag{3.5}$$

where the sizes s_k are sorted in descending order: $s_k \geq s_{k'}$ for $k < k'$. In the case of equally-sized realizations, an explicit solution can be easily found, reading: $o_i > \mathrm{int}(fR)/R$, where $\mathrm{int}(x)$ indicates the integer part of the real number x.

3.3.2 Explicit expressions and power law scaling from power law and exponential rank distributions

Explicit solutions for the occurrence distribution can be derived assuming a simple scenario, in which all realizations have the same size s, and the component frequency statistics follows a specific function. We first consider

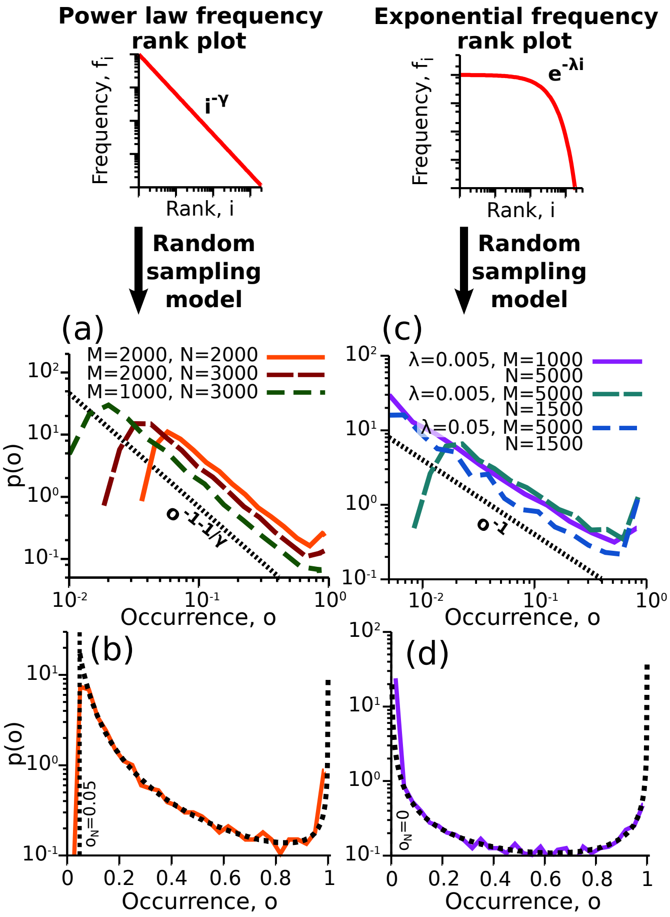


Figure 3.2: **Power-law decaying and U-shaped component occurrence distributions may descend from both power-law and exponential distributed universe component frequencies.** (a): A power-law rank-plot for the frequency (and thus for the abundance), whose exponent is $-\gamma$ ($\gamma = 1.2$ in the plot), produces a power-law decay of the component occurrence distribution with exponent $-1 - \frac{1}{\gamma}$, independently of the realization size s and the number of components N (for sufficiently large values of these parameters). (b): Agreement between the theoretical prediction of Eq (3.7) (black line) and a simulated random sampling with parameters $R = 1000$, $N = 2000$, $\gamma = 1.2$, $s = 2000$ (the black vertical dashed line is the left boundary of the $p(o)$ domain). Panels (c) and (d) are the counterpart of (a) and (b) for an exponential frequency rank plot. In this case $p(o)$ always decreases with exponent -1, for every value of λ, s, and N (sufficiently large). Parameter values: $R = 1000$, $N = 2000$, $\lambda = 0.005$, $s = 5000$.

the empirically relevant case of a power-law frequency rank plot (Figure 3.2, left panel) defined by

$$f_i = \frac{1}{\alpha} i^{-\gamma}, \qquad \alpha = \sum_{i=1}^{N} i^{-\gamma} \ . \tag{3.6}$$

Under these assumptions and using Eq. (3.2) and (3.3), the exact expression of the occurrence distribution can be calculated:

$$p(o) = \frac{(1-o)^{\frac{1}{s}-1}}{\gamma s N \alpha^{\frac{1}{\gamma}} \left(1-(1-o)^{\frac{1}{s}}\right)^{\frac{1}{\gamma}+1}} \ . \tag{3.7}$$

The distribution is defined in the interval of occurrences $[o_N; o_1]$, where o_i is computed by Eq. (3.2). This expression can be simplified taking the limit of large realizations $s \gg 1$:

$$p(o) = k(\gamma, s, N) \frac{(1-o)^{-1}}{\gamma\left(-\log(1-o)\right)^{1+\frac{1}{\gamma}}}, \tag{3.8}$$

where k is defined as:

$$k(\gamma, s, N) = \frac{s^{\frac{1}{\gamma}}}{\alpha(\gamma, N)^{\frac{1}{\gamma}} N}. \tag{3.9}$$

It is worth pointing out that in this limit the number of independent parameters defining the occurrence distribution reduces to two: γ and k, which is a combination of s (realization size), N (system vocabulary) and γ. A test of these “rescaling property” is shown in Figure 3.3. The expression can be further approximated for low values of the occurrence, $o \ll 1$, leading to the empirically observed power-law decay:

$$p(o) \simeq k(\gamma, s, N) o^{-\frac{1}{\gamma}-1} \ , \tag{3.10}$$

where the power-law exponent depends only on the exponent γ of the frequency rank-plot. The agreement between these predictions and simulations are shown in Figure 3.2a,b.

Analogous calculations can be performed assuming a frequency distribution described by an exponential rank plot $f_i \sim e^{-\lambda i}$ (right panels of Figure 3.2). In this case, the distribution of shared components, for large enough realizations, $s \gg 1$, has the expression:

$$p(o) \simeq \frac{(1-o)^{-1}}{N\lambda \log\left[(1-o)^{-1}\right]}. \tag{3.11}$$

which is a function of a single effective parameter λN, and does not depend on the realization sizes s. In other words, the shape of the distribution, and whether it is clearly U-shaped, only depend on the decay of component frequencies and on the total number of components. In fact, occurrence distributions corresponding to different exponential frequency rank plots collapse if λN is constant, even if the realizations have widely different size. This is shown in Figure B.1 of Appendix B.1.

Interestingly, for rare families the above expression simplifies again to a power-law decay

$$p(o) \simeq \frac{1}{N\lambda} o^{-1}, \tag{3.12}$$

with a "universal" exponent -1. This indicates that also systems with a heterogeneous but more compact frequency distribution are expected to show a power-law decay in the occurrence distribution.

3.3.3 When the "U" shape emerges: the core size

We now turn our attention to the conditions for a U-shaped distribution of shared components in the random-sampling model. In particular, we can now estimate the "core size" by computing the fraction of components with occurrence greater that a given threshold θ_c as a function of the only two effective parameters γ and k. Considering the random sampling from a power law rank distribution, the core size estimate can be derived integrating Eq. (3.7) from θ_c to the maximum occurrence o_1. Taking the $s \gg 1$, this quantity reads

$$\begin{cases} c = 1 & \text{if} \quad o_N \geq \theta_c \\ c = k\left[-\log(1-\theta_c)\right]^{-\frac{1}{\gamma}} & \text{otherwise} \end{cases}, \tag{3.13}$$

where o_N is the left boundary of the occurrence distribution, corresponding to the component with lowest frequency. Starting from this estimate of the core size, Figure 3.3ab shows how the scaling property of the Eq. (3.8) is verified in simulations.

Fig. 3.3c compares the analytical predictions for the core size with simulations for different values of γ, showing perfect agreement. Equally, one can obtain analytical estimates for the fraction of rare components (occurrence below a fixed threshold), which are tested in Fig. 3.3d. Thus, with increasing k, core families increase linearly with a γ-dependent slope until all components are shared, and concurrently rare components decrease linearly until they hit zero (when the lower cut-off of occurrence exceeds the chosen threshold value). Component number and realization size only enter through the combination defined by the rescaling parameter k. This phenomenology fully characterizes the distribution of shared components with varying parameters.

In Appendix B.1 is shown a similar analysis for the occurrence distribution generated by an exponential frequency rank plot, deriving the analogous equation of (3.13).

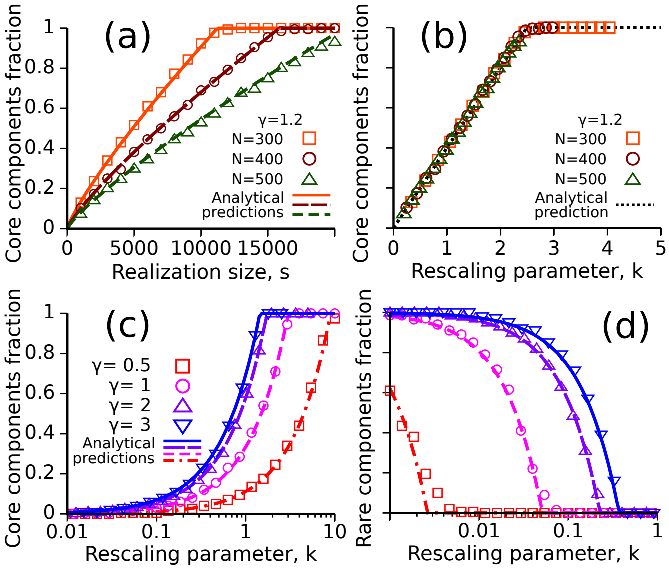


Figure 3.3: **Scaling of the distribution of shared components and fraction of rare and core components.** (a) The fraction of core components (defined by the occurrence threshold $o > \theta_c = 0.95$) for a power-law component frequency distribution with exponent $\gamma = 1.2$, plotted as a function of component size s for three values of realization number N. (b) Collapse of the curves shown in panel (a) when plotted as a function of the rescaled parameter k, defined in Eq. (3.9). (c) and (d): fraction of core and rare ($o < 0.05$) plotted as a function of k for different values of γ. For sufficiently large k (i.e. typically when s dominates over N), the fraction of core components saturates to 1. Conversely, the fraction rare components drops to zero for increasing k. Symbols refer to numerical simulations of the random-sampling model, while the lines are the theoretical predictions of Eq. (3.13).

3.4 Predictions confirmed in the data

3.4.1 Power law decay of the occurrence distribution

One can ask whether the general analytical predictions discussed in the previous section can be applied to empirical data. In particular, we first asked how the power-law decay exponent of the distribution of shared components relates to the component frequency rank plot in empirical systems, and if this relation follows our analytical prediction (Eq. (3.10)). An analytical mapping would give a more synthetic and powerful description than the direct simulations discussed in Section 3.2. Importantly, the analytical formulas for the distribution of shared components are derived under the hypothesis of a pure power-law or exponential component frequency rank plot. However, the three empirical datasets show a more complex phenomenology, and can be better approximated with a double-scaling power-law frequency rank distribution (see Figure 2.3 and panels a,c,e of Figure 3.4). To override this issue, we restricted the frequency rank plot range in which the predictions are applicable. The procedure to perform this comparison is described below and applied in Fig. 3.4. First, we chose an arbitrary threshold θ_r defining the rare components and we mapped it to the frequency rank plot (assuming the model), by using the inverse function of Eq. (3.2). The frequency rank associated to the occurrence threshold θ_r, $i(\theta_r)$ in the figure, is the rank above which the model prediction for the decay of the distribution of shared components should apply as long as $i(\theta_r)$ does not cross the position of the change in scaling. In other words, since in the model there is a monotonic relation between occurrence and frequency (Eq. (3.2)), all components with rank greater than $i(\theta_r)$ (and frequency smaller that $f_{i(\theta_r)}$) are assumed to be the components with occurrence lower than θ_r. We then estimated the behaviour of the frequency rank plot in the high-rank region (after $i(\theta_r)$) as the best fit with a power-law function or an exponential. This leads to a prediction for the decay exponent of the distribution of shared components (using Eq. (3.10) or Eq. (3.12) for the exponential case) in the range $[o_N, \theta_r]$. Fig. 3.4 shows that the predicted decay exponents correspond well with the data.

3.4.2 Scaling relationship for the core size

The random-sampling model also gives qualitative analytical predictions for the expected fraction of core components, and thus for the expected shape of the distribution of shared components for a given empirical system. While the analytical relations between exponents applied in Figure 3.4 do not depend on the realization sizes, the analytical formulas for the fraction of core components (see e.g. Eq. (3.13)) were derived assuming realizations of fixed size s. The actual size distributions for the three empirical systems are quite broad (Figure A.1, A.3, and A.5 in Appendix A), but we can still use the analytical framework to get an estimate of the core fraction considering the average realization size of each empirical system. Following the same line of reasoning as for the low-occurrence tail of the distribution of shared components, we can use a restricted region of the frequency rank plot. In this case, the low-rank region (with exponent around 1 for all the datasets) is expected to contain the core components. Therefore, the parameter γ can be fixed to 1, implying that the fraction of core components, given by Eq. (3.13), should be simply proportional to the rescaling parameter k (Eq. (3.9)). However, the function $\alpha(\gamma, N)$, which is present in the definition of k and defined in Eq. (3.6), takes an approximately constant value with respect to N for large values of N, as it is the case for the empirical examples considered. As a consequence, the core fraction should be simply proportional to $\frac{s}{N}$.

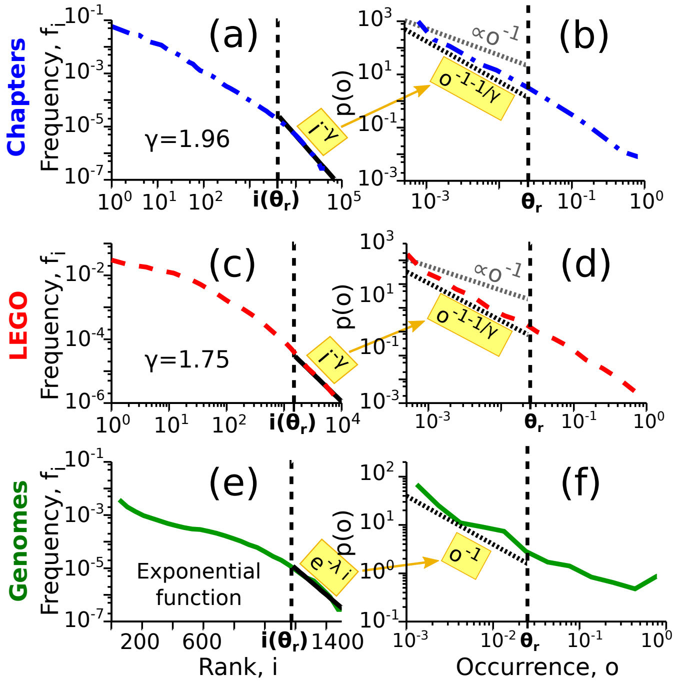


Figure 3.4: **The relation between the exponents of frequency rank plot and occurrence distribution is satisfied in all the three datasets.** The plots consider the low occurrence region, below the arbitrary threshold $\theta_r = 0.025$ which corresponds to the high-rank region above $i(\theta_r)$ in the frequency rank plot (see main text). Panel (a) and (b) refer to book chapters, for which the tail of rank plot is a power law with exponent $\gamma = 1.96$, which implies a power law decay of $p(o)$ with exponent $1+\frac{1}{\gamma} = 1.51$. Panel (c) and (d) show the LEGO dataset ($\gamma = 2.8$, $1 + \frac{1}{\gamma} = 1.36$). Panel (e) and (f) correspond to protein domains in genomes, where the best fit of the tail region the rank plot is an exponential function (note that (e) is in linear-logarithmic scale), which implies a power law decay with exponent -1.

This estimate can be used to explain why the core fraction is much larger in genomes than in the other two empirical systems (see Figure 3.1d). In fact, genome sizes are typically of the same order as the total number of families ($s \simeq 3000$, $N = 1531$) leading to a large expected core. By comparison, book chapters have similar realization sizes but a much larger vocabulary ($N \simeq 50000$), and LEGO sets have very small sizes ($s \simeq 100$) compared to vocabulary size ($N \simeq 13000$).

More in general, Eqs. (3.9) and (3.13) lead to a scaling estimate (dependent on the decay of the frequency rank plot) as a function of the system parameters s and N, which can be applied to data, in order to generate expectations for the core components. For example, for Zipf-like (exponent -1) frequency distributions, we expect the absolute number of core components to be linearly dependent on the average size of realizations s, and essentially insensitive to the vocabulary size N and the total number of realizations R.

The predicted linear relation between core fraction and average realization sizes is tested in Figure 3.6a for prokaryotic genomes and seems accurately verified. However, the fraction of core components predicted by the random-sampling model is actually much smaller than the empirical one. This highlights the presence of additional functional constraints and/or specific correlations in the empirical system that the model can not capture. Section 3.5 addresses this point more in detail.

3.4.3 Core size scaling with the system-vocabulary in genomes

Besides the growth of the core size fraction with the typical realization size, the relation (3.13) states that the core size is inversely proportional to the number of different components present in the ensemble N. This holds true under the condition that N is sufficiently large such that $\alpha(\gamma, N)$ is approximately constant. To test this prediction we take advantage of the concept of multiple resolution mentioned in Section 2.1 applied to the genomic dataset. Indeed, there not exists a unique prescription to build the protein-domain-family classification. Even following the same rules for defining similarity between families (such as evolutional and functional similarity used in the SUPERFAMILY database [63]), one can tune the "parameters" to obtain different resolutions. For example, when the constraints are weaker, the average number of protein domains in a family enlarges, while the total number of families, N, diminishes. This allows us to compare ensembles having different vocabulary sizes. It is important to mention that to test the dependency on N, the different systems must conserve the other parameters, that are the realization size and the Zipf's law exponent. These two conditions are satisfied in the five considered protein domain family classifications described in Appendix A.1. In fact, they come from two databases (SUPERFAMILY and PFAM) whose collection of genomes share a similar size distribution (Figure A.1a). Moreover, the exponents of the global Zipf's laws for low ranks (those affecting the core components as explained in the section before) are qualitatively the same (Figure A.1b).

Figure 3.5 displays the core size fraction as a function of the vocabulary size for the five ensembles, showing a qualitative agreement with the equation (3.13). The multiplicative coefficient is roughly fixed choosing s as the average value of the size distribution, while γ and α are estimated by the fit of the "average rank plot" (for each rank we consider the average of the five frequencies with that rank). Even though the theoretical expectation is derived under very strong assumptions, it reproduces quite well the data,

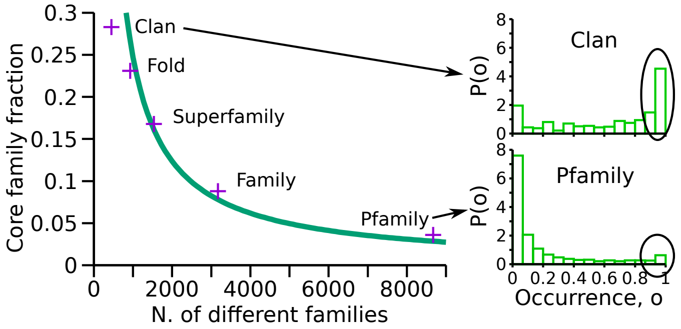


Figure 3.5: **The random sampling model predicts the core size scaling with the system vocabulary.** Fraction of the common families (threshold: $\theta_c = 0.95$) as a function of the total number of different families, N, for five different protein domain family classifications (family, superfamily and fold classifications from the SUPERFAMILY database, and Pfamily and clan from the PFAM database, Appendix A.1). The empirical values are compared with the random sampling prediction (continuous line), which scales as N^{-1}, according to the equation (3.13.)

providing a further proof that the distribution of shared components and its features can be reliably predicted from the abundance and size statistics.

3.5 Deviations from the model highlight functional information

Beyond the striking agreement with null predictions for shared components, the deviations from sampling can be used to quantify specific functional and architectural features of a component system. While the scope of this work is to highlight the common trends and their origins, here we discuss a simple procedure to quantify deviations from the random sampling, applied below to a specific example. Let us consider a generic component system and its associated random sampling (computed from the frequencies and sizes of the original system). Then one can test the deviation in a small bin of occurrence (or a slice of the the occurrence distribution) $[o_1, o_2]$ with the Z-score:

$$Z(o_1, o_2) = \frac{n^{\text{data}}(o_1, o_2) - \langle n^{\text{sampling}}(o_1, o_2)\rangle}{\sigma^{\text{sampling}}(o_1, o_2)}, \tag{3.14}$$

where $n^{\text{data}}(o_1, o_2)$ is the number of components with $o \in [o_1, o_2]$ in the empirical system, $\langle n^{\text{sampling}}(o_1, o_2)\rangle$ is the average of the same quantity over an ensemble of sampling models, and $\sigma^{\text{sampling}}(o_1, o_2)$ their standard deviation. The value of $Z(o_1, o_2)$ can be associated to a p-value quantifying the statistical significance of the deviation.

3.5.1 Deviations of protein domain categories

Of the three data sets considered here, the case where the clearest deviations emerge are genomes. For example, Figure 3.6a illustrates how the random sampling underestimates the empirical core size by a constant offset, for genomes of increasing size. Generally speaking, this larger core of components is due to the components that tend to occur in most realizations, but in few copies. The natural explanation is that there are specific basic functions that are essential for all (or most) genomes, but the domains involved in these functions are not necessarily needed in many copies per genome, and thus their presence in all realizations does not simply correlate with high global abundances as the random sampling would entail [64].

To test this hypothesis, we divided the domain families in functional categories (see Appendix A.1.1 for the functional annotation), and computed the Z-score 3.14 for components of each family in different bin of occurrences. The result of this analysis is reported in Figure 3.6b. Different parts of the distribution of shared components are indeed enriched in components of different biological functions with respect to the random-sampling expectation. In particular, protein domains that play a functional role in information processes, (such as DNA transcription, and DNA replication) are clearly enriched in the core. At the same time, they seem statistically under-represented at occurrences around 0.6. These two deviations can be explained as two sides of the same coin if this category contains domain families that empirically occur in all genomes but in a single copy per genome. Indeed, the global frequency (i.e, across all genomes) of families that are both single-copy and ubiquitous is $f = \frac{R}{Rs} = 1/s$. Therefore, their occurrence predicted by the random-sampling model is $o = 1 - (1 - \frac{1}{s})^s = 1 - e^{s\log(1-\frac{1}{s})} \simeq 1 - e^{-1} \simeq 0.6$ (where the rough approximation holds for large enough s), thus naturally leading to an excess of those families in the core and to a depletion around $o \simeq 0.6$.

The observation of a strong presence of protein domains related to basic cellular function in the core genome is not new [64]. However, the random-sampling model allows in principle to distinguish families whose presence in the core could be simply explained by their high abundance in the pan-genome and thus it would be expected also in a simple scenario of random gene exchange. Finally, the observed correlation between biological functions and deviations from random sampling predictions seems coherent with a picture, recently proposed [60], in which natural selection and functional constraints have played an important role in defining the empirical U-shaped distribution of gene occurrences.

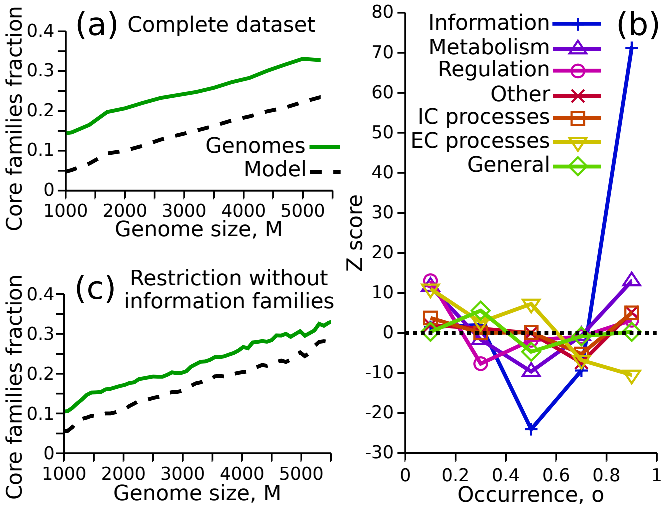


Figure 3.6: **Specific functional constraints can be detected by deviations from the predictions of a random sampling.** a) Fraction of common protein domain families as a function of the genome sizes. Each point of the curves corresponds to the core families ($o > \theta_c = 0.95$) given the occurrence distribution of a genomes' subset whose sizes are inside a certain window. The average of the size windows defines the x axis. b) Enrichment analysis in the occurrence distribution for specific functional categories. Considering domain families relative to a single functional category, their relative component occurrence distribution was evaluated for an ensemble of systems built with a random sampling. From this, the average value and the standard deviation for the expected fraction of components at each occurrence value o can be calculated. This provides a measure (Z score) of over- or under-representation of domain families belonging to each functional category in the empirical dataset. c) Excluding from the analysis the domain families associated to information processes (i.e., DNA replication, transcription and translation) significantly reduces the offset between the random-sampling prediction and the empirical trend.

3.6 Discussions

This work employs a simple statistical model based on random sampling of components to describe the distribution of shared components in complex component systems. A similar approach was employed in quantitative linguistics to explain the so-called “Heaps' law” (Section 2.2.2) while assuming Zipf's law for component frequencies [56, 48, 55, 54, 6]. We extended the model to show that there is a general link between the Zipf's law of the system (i.e. the rank plot of the component abundances) and the statistics of shared components, regardless of the mechanisms that generate Zipf's law. Consequently, models or generative processes able to explain the heterogeneity in component abundance implicitly carry information about the statistics of shared components.

The model can be also investigated analytically, characterizing the occurrence distribution features and how they depend on the abundance statistics and few others component-system parameters. Specifically, its power law decay is related to Zipf's law behavior (Eq. (3.10) and Fig. 3.2), while the fraction of common components (or core size) depends on the realization sizes, the system vocabulary and the Zipf's law exponent (Eq. (3.13) and Fig. 3.3). These results have been derived within an over-simplified setting with respect to the data, but the fact that the power law exponent does not depend on the size distribution allows us to apply the prediction to the empirical curves, Fig. 3.4. By contrast, the formula (3.13) cannot be directly confront to the data, but it provides still qualitative predictions about the scaling of the core size with the realizations size, Fig. 3.6a, and the system vocabulary, Fig. 3.5.

It is important to point out that the relationship between the two statistical laws is solid in three very different empirical systems (LEGO sets, genomes, book chapters), and it is confirmed by simulations, Fig. 3.1 as well as theoretical predictions (Section 3.4). Therefore, it is reasonable to claim that the link between the two statistical laws is a "universal" property of modular systems. This justifies the introduction of the concept of "component system", which can capture in a unified framework a large class of complex systems which may show convergent phenomena. Furthermore, component-system-representation has been proven useful since we took advantage of the bridges between distant fields that it creates, for example, extending the random-sampling approach developed in quantitative linguistics to our specific problem, and generalizing the distribution of shared component in the context of natural languages.

As said above, the main result of this work is that the patterns of shared components can be largely predicted by the null model. However "small deviations" can emerge. We have considered here a specific example for the case of shared protein domain families in genomes (Fig. 3.6), but this question still needs to be approached systematically. In this specific case, core components are particularly enriched by specific functional classes of components with respect to the sampling prediction. In evolutionary terms, the random sampling defines a scenario in which the pan-genome fully determines the overall abundance of the gene families in each genome, while in empirical bacterial genomes genome-specific functional constraints are clearly in place [1, 65, 66]. Deviations from the null scenario can thus highlight the role of selection for specific functions, supporting from a different perspective the idea that the empirical U-shaped gene occurrence distribution is affected by selective rather than neutral processes [60, 62, 61, 59]. In general, the random sampling model generates a "first order approximation" of the distribution of shared components, based on a purely random scenario constrained only by the component-abundance and realization-size statistics. As a consequence, the deviations highlight system-specific constraints, which can be informative about functional properties of the system. In our view, the study the occurrence distribution should always be approached through the comparison with the null-prediction, in order to take into account the "redundant" information contained already in the abundance statistics. In this regard, model-data deviations provides "higher order" observables which can be used to validate generative models.

Chapter 4

Zipf and Heaps laws from dependency structures

Authors: Andrea Mazzolini, Jacopo Grilli, Eleonora De Lazzari, Matteo Osella, Marco Cosentino Lagomarsino, Marco Gherardi.

4.1 Introduction

Dependency structures have emerged recently as a promising framework for the rationalisation of the regularities observed in complex systems [2]. They have been proposed in various contexts and forms, and have helped achieving remarkable results, for instance in the scope of preference prediction [67], or for addressing causality in financial data [68]. A dependency structure is a directed graph (most often, but not necessarily, acyclic), whose nodes are the components (e.g. genes, Linux packages) and whose links are the dependency relations occurring between them. A component depends on another if it is not functional unless the latter is present, for example, genes involved in metabolism typically depend on chemical reactions involving other metabolic genes, or specific software packages need other low-level packages (e.g. Python, gzip) to be functional.

In this work we assume that the component system is constrained by the underlying dependency structure between the system-components. To this end, a system-realization is constructed with the following prescription: if a certain node/component belongs to the realization, then all its direct and indirect dependencies must be included in the realizations. This simple model allows to link statistical laws of component systems to topological properties of the dependency structure. For instance, a broad ensemble of dependency networks has the property that the number of total dependencies of each node is scale-free. This topological property explains the fat-tailed distribution of shared components, both in genomes and technological systems [2]. A limitation of this model is that the components are constrained to have binary abundance in a single realization, i.e., to either be present in one copy or to be absent (see Section 2.1.2). Such a description is expected to be accurate for some components (e.g., for software packages) but is a rough approximation for those systems where components appear with non-negligible abundances (e.g., coarse-grained evolutionary families of genes such as superfamilies, and words in a text).

The work presented in this section extends the model proposed in [2] to the case where components appear with non-trivial abundances. This allows us to explore how dependency structures affect abundance-related features, such as Heaps' (Section 2.2.2) and Zipf's laws (Section 2.2.1).

4.2 Model

4.2.1 The dependency network

In order to generate a meaningful dependency structure, we use the model introduced in [2] and briefly summarized here. Such a network defines dependency relationships between components, and it will be used as input of the novel generative process described in the next section.

A dependency structure is a directed acyclic graph $\mathcal{G}$ on a given set of nodes/components $\{c_1, \dots, c_N\}$. An edge $i \to j$ between two nodes i and j represents the relation "i depends on j", which means that i is not functional without j (e.g. a software package i that depends on a low-level package j). Such a relation can be more or less strict depending on the system; for instance it is enforced in software operating systems, where a package cannot function unless all its dependencies are installed [69], but not in metabolic networks, where alternative pathways can be chosen [1]. We assume here strict unbroken dependencies. Notice that acyclicity of $\mathcal{G}$ is not stringently necessary; however, as will be clear in the following, a cycle in $\mathcal{G}$ would behave as a single node in the model.

The network is generated through a very simple growth process with an incremental node-addition mechanism. Starting with an initial set of m nodes without links, called *seeds*, (where we typically consider $m = 1$ for all the following calculation), the full network is built node by node, where at each step of time a new node enter the system. The new node is linked to d randomly chosen existing nodes, where d is a random variable of mean D (typically extracted from a Poisson distribution). The process is stopped when the network reaches the total number of components N (therefore after $N - m$ time steps). The Figure 4.1a shows an illustrative example with $N = 9$ nodes and $m = 3$ seeds. In this specific case, the variable d is considered deterministic and always equal to 2. As a consequence, all the nodes have out-degree 2 except for the three seeds. Note also that labelling each node in order of appearance $t = m+1, \dots, N$, the network satisfies the property that for each link $t \to t'$, the time t is always less than t'. This

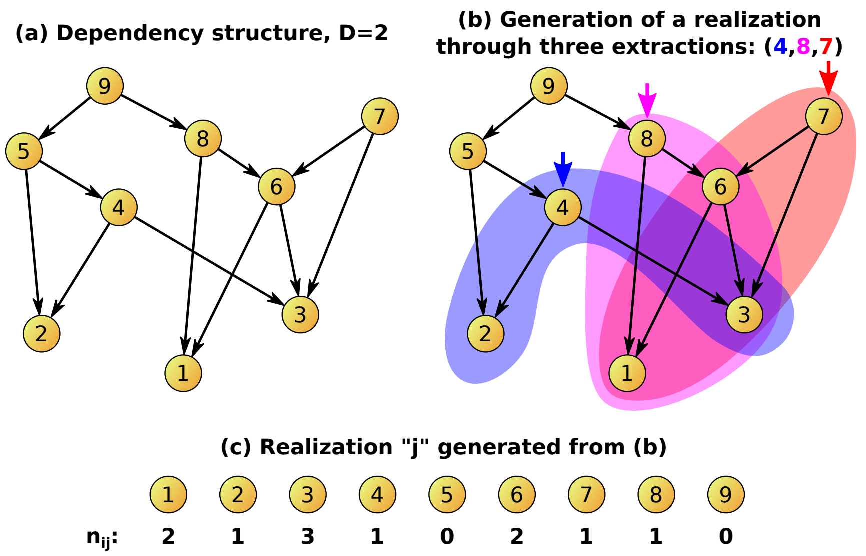

(a) Dependency structure, D=2
(b) Generation of a realization through three extractions: (4,8,7)
(c) Realization "j" generated from (b)
1 2 3 4 5 6 7 8 9
n_{ij}: 2 1 3 1 0 2 1 1 0

Figure 4.1: **Component-system-realization constrained to the dependency structure.** a) Toy example of a dependency network generated through the model described in the main text with $N = 9$ components, three seeds $\{1, 2, 3\}$, and fixed out-degree $D = 2$. b) Illustrative example of the generative model for a system-realization constrained to the dependency network of the panel (a). Three precursor components are chosen at random $\{4, 8, 7\}$. For each extraction, the node and all its dependencies (its forward cone) are added to the realization (each forward cone is illustrated with a coloured area). At the end of the process the obtained realization has a number of instances of components as shown in the panel (c).

implies that the obtained graph is acyclic.

4.2.2 Component system from the dependency network

This section introduces a possible model to generate a component system constrained by a dependency structure. Although whatever dependency network can be used as input of the model, the current analysis considers the network defined in [2] and described in the previous section. This is motivated by the fact that it shows statistical properties qualitatively similar to the empirical ones.

Before explaining the details of the algorithm, it is crucial to introduce the definitions of *forward cone* and *backward cone* of the dependency network $\mathcal{G}$. Given a node c, we define its forward cone, $\wedge(c)$, the set of all nodes c' such that there exists at least one directed path in $\mathcal{G}$ starting from c and arriving at c'. On the other hand, the backward cone $\vee(c)$ of c is the set of all nodes c' such that there exists a path from c' to c. In other words, $\wedge(c)$ is the set of all components from which c depends on (directly or indirectly), whereas $\vee(c)$ is the set of the nodes that depend on c (directly or indirectly).

Let us fix a positive integer ρ, which represents the number of "precursor" components determining a realization. For each realization, the ρ precursors are chosen randomly and independently among the nodes of $\mathcal{G}$. Then the corresponding realization is produced by taking all the components from which the precursors depend on (i.e. belonging to their forward cones). A toy example is shown in Figure 4.1b, where the $\rho = 3$ precursors are $\{4, 8, 7\}$, and their forward cones are highlighted with areas having different colours. At each extraction, each node within the cone is added to the realization, whose final configuration of component abundances is shown in the panel 4.1c. Given this recipe, one can generate a component system composed of R realizations according to a set of numbers of precursors, $\{\rho_1, \dots, \rho_R\}$. Note that the realization sizes s_j are a stochastic variable depending on the number of precursor ρ_j. Specifically, the average size depends linearly on the number of precursors through the relation:

$$E_{\mathcal{G}}[s_j] = \rho_j \langle | \wedge | \rangle_{\mathcal{G}}, \tag{4.1}$$

where the expected size is the average over an ensemble of realization with same ρ_j and fixed dependency network $\mathcal{G}$, while $\langle | \wedge | \rangle_{\mathcal{G}} = \frac{1}{N+m} \sum_{i \in \mathcal{G}} | \wedge (i)|$ is the average of the number of nodes in the forward cones of the network $\mathcal{G}$.

The model in [2] can be viewed as a special case of the presented algorithm for $\rho = 1$. It is important to point out that, differently from [2], here the components have "multiplicity" within the realization (i.e. the system is not binary), allowing us to compute the so-called Zipf's and Heaps' laws.

4.3 The Zipf's law from the dependency structure

Let us consider a a component system constrained by a dependency structure $\mathcal{G}$ and composed of R realizations generated through a fixed number of precursors ρ. As defined in Table 2.1, the abundance of the node/component i is the total number of times that it appears in the system. It can be computed by knowing the probability of choosing a cone that contains i: $\frac{|\vee(i)|}{N}$, where $|\vee(i)|$ is the size of the backward cone of the component. The expected abundance of i is then:

$$a_i = R\rho \frac{|\vee(i)|}{N}, \tag{4.2}$$

where $R\rho$ is the total number of extraction, which also fixes an upper boundary to this quantity.

Sorting the component abundances in descending order, the indexes in this sorted list become the "ranks" of the components, defining the Zipf's law. Following [2], an approximate relation can be derived between $|\vee(i)|$ and the component rank i. Indeed, one can derive the functional form of the backward cone size of a node added at time t in the network (the one added at the t-th step of the construction, when a network of size $t-1$ has already been generated). This result can be obtained by writing an equation based on the observation that the backward cone of the t-th node is the union of the backward cones of all the nodes that, at later times t', will directly attach to the t-th node. Neglecting the intersections between these cones allows to write the recursion

$$|\vee(t)| = 1 + \sum_{t'=t+1}^{N} \frac{D}{t'} \left|\vee(t')\right| , \tag{4.3}$$

where the factor D/t' estimates the probability that the t'-th node attaches to the t-th node. By approximating the sum by an integral and taking a derivative with respect to t, one obtains a differential equation that is solved by $|\vee(t)| = (N/t)^D$. For small t, however, $(N/t)^D$ is greater than the size of the network N. In fact, the relation can hold only down to a cut-off $t_{\min}$, which can be estimated by the condition that the whole network depends on the $t_{\min}$-th node, i.e., $(N/t_{\min})^D = N$, which gives $t_{\min} = N^{1-1/D}$. For any node below $t_{\min}$, the size of its backward cone is $\approx N$:

$$|\vee(t)| \approx \begin{cases} N & t < N^{1-1/D} \\ (N/t)^D & t \geq N^{1-1/D} \end{cases} \tag{4.4}$$

Identifying the rank of a node with the time in which it enters the network (the first introduced nodes have larger abundance, on average), and putting together Eq. (4.2) and (4.4), one obtains the expression for the Zipf's law:

$$a_i \approx \begin{cases} R\rho & i < N^{1-1/D} \\ R\rho N^{D-1} i^{-D} & i \geq N^{1-1/D}. \end{cases} \tag{4.5}$$

This relation has the form of a Zipf power-law (with exponent $-D$) with an initial set of of $N^{1-1/D}$ components having maximal abundance. Figure 4.2a compares the analytical form (4.5) with the results of simulations, showing good accord, especially in the behaviour of the fat tail. The transition between the core and the tail, instead, is less sharp than predicted. This is tied to the fact that the relation $|\vee| = (N/t)^D$ starts to break down before reaching N, and saturates more smoothly than in the approximation made above.

In addition to the abundance, this model allows us to compute the component occurrence (which defines the distribution of shared components discussed in Sections 3 and 2.2.3). In particular, we can ask whether the occurrence-abundance relation is equivalent to the universal (or "null") prediction under the random sampling assumption, Equation (3.2). This would

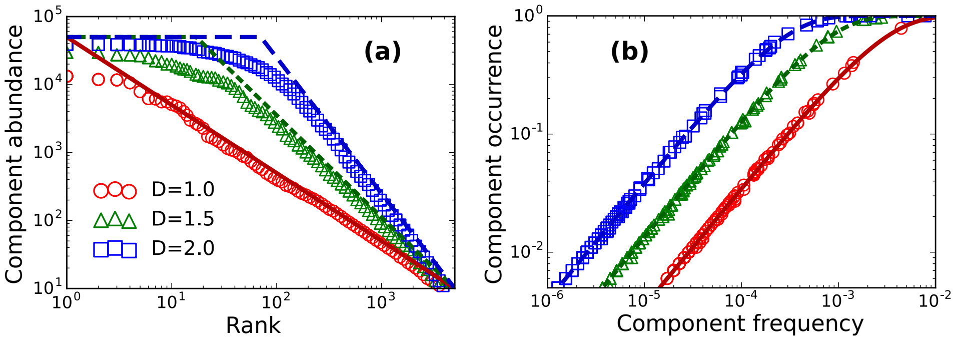


Figure 4.2: **Zipf's law prediction and occurrence-abundance relation.** Zipf's laws of component systems constrained to dependency networks are shown in panel (a). Three systems at different D are compared (the other parameters are fixed: $R = 1000$, $\rho = 50$, $N = 5000$). The lines follow the theoretical prediction (4.5), which define a "saturation" region for ranks greater than $N^{1-1/D}$, meaning that all these nodes are directly and indirectly connected to all the N nodes. This region is followed by a power law decay whose exponent is fixed by D. Panel (b) shows the occurrence-frequency dependence (where the frequency is the abundance divided by the total number of components in the system). This behaviour is well fitted by Equation (3.2), implying that the occurrence statistics is determined by a random sampling of components.

imply that the occurrence statistics can be predicted only by the knowledge of the Zipf's law (main result of Chapter 3), and it is independent of the detailed structure of the dependency network. In fact, we show here that a simple probabilistic argument gives a relation that is asymptotically equivalent to Eq. (3.2) (in the case of R realizations having the same size s). In the limit of large N, we can assume that the occurrence of a component i is equal to the probability of choosing i at least once in a single realization:

$$o_i = 1 - \left(1 - \frac{|\vee(i)|}{N}\right)^{\rho}. \tag{4.6}$$

Using the expression (4.5), and introducing the normalized abundance: $f_i = a_i / \sum_i a_i = a_i/(sR)$, one obtains:

$$o_i = 1 - \left(1 - \frac{s}{k} f_i\right)^{\rho}.$$

Finally, one can distinguish two regimes: if $f_i \ll 1$, the expression simplifies as follows:

$$o_i \simeq 1 - e^{-s f_i} \simeq 1 - (1 - f_i)^s$$

which is equivalent to the formula (3.2) for large s. In the case of $f_i \lesssim 1$, assuming $\rho \gg 1$, the occurrence saturates to 1, as for the "null" prediction in the large s limit. Figure 4.2b shows that a scatter-plot of occurrence versus abundance occurrence in simulations perfectly matches the theoretical curve (3.2). This fact allows us to use the results presented in Chapter 3 to characterize the "U"-shaped distribution of the model. For example, the exponent of the rare-components-region should be $1 - 1/D$ (using the result 3.10).

4.4 The Heaps' law from the dependency structure

4.4.1 Analytical derivation

This section is dedicated to the derivation of the so-called Heaps' law (Section 2.2.2), $h(s)$, i.e. the number of unique components in a realization of size s. This calculation can be performed in a mean-field approximation, where the correlations between nodes are neglected. Note that the fact that the occurrence-abundance relation follows the random sampling prediction (Figure 4.2b) suggests that this assumption is justified. A second hypotheses is that the abundance rank plot has the expression (4.5). Even though simulated systems may show small deviation from the prediction in the "saturation" regime (Figure 4.2a), the expression catches the essential features of the law (i.e. the power law scaling, and deviation from this scaling at low ranks).

In order to derive the number of different components in a realization of size s, let us consider the probability of drawing the node added at time t:

$$p(t) = \frac{1}{\Omega} \left| \vee(t) \right| , \tag{4.7}$$

which is proportional to the size $|\vee(t)|$ of its backward cone. In a continuous approximation, the normalization Ω can be fixed by the normalization condition $\int_1^N p(t) \, \mathrm{d}t = 1$, which reads (for $D > 1$):

$$\Omega = \frac{ND}{D-1} \left(N^{1-1/D} - 1 \right) \approx \frac{DN^{2-1/D}}{D-1}, \tag{4.8}$$

where the approximation holds true for $N^{1-1/D} \gg 1$. Analogously to $q_i(s_j)$ in Equation (3.2), we now consider the probability that the t-th node in the network is present in a realization of size s:

$$q_t(s) = 1 - (1 - p(t))^s \,.$$

A mean-field estimate of h (i.e. the number of distinct components) can then be obtained as

$$h(s) = \sum_{t=1}^{N} q_t(s) \approx \int_1^N q_t(s) \, \mathrm{d}t = N - \int_1^N (1 - p(t))^s \, \mathrm{d}t, \tag{4.9}$$

which can be written down using the explicit expressions $p(t)$ and considering 1 negligible with respect to $N^{1-1/D}$:

$$h(s) = N - N^{1-1/D}\left(1 - \frac{N}{\Omega}\right)^{s} - \mathcal{H}\left(s, N^{1-1/D}, \frac{N^D}{\Omega}\right), \tag{4.10}$$

where the last addendum corresponds to the summation (B.7) in Appendix (approximated as an integral):

$$\mathcal{H}\left(s, N^{1-1/D}, \frac{N^D}{\Omega}\right) = \int_{N^{1-1/D}}^{N} \left(1 - \frac{1}{\Omega}\left(\frac{N}{t}\right)^{D}\right)^{s} \mathrm{d}t. \tag{4.11}$$

As discussed in Appendix B.2, the term above can be approximated for $s \gg 1$ and $N \gg 1$, Eq. (B.8). Substituting this approximation in the expression (4.10), one finally obtains:

$$\begin{aligned} h(s) = N - N^{1-1/D}\left(1 - \frac{N}{\Omega}\right)^{s} - \\ - \frac{N}{D}\left(\frac{s}{\Omega}\right)^{1/D}\left(\Gamma\left(-\frac{1}{D}, \frac{s}{\Omega}\right) - \Gamma\left(-\frac{1}{D}, \frac{Ns}{\Omega}\right)\right), \end{aligned} \tag{4.12}$$

where Γ is the incomplete Gamma function. Fig. 4.3 shows that the analytical mean-field expression (4.12) nicely matches the results of numerical simulations of the model.

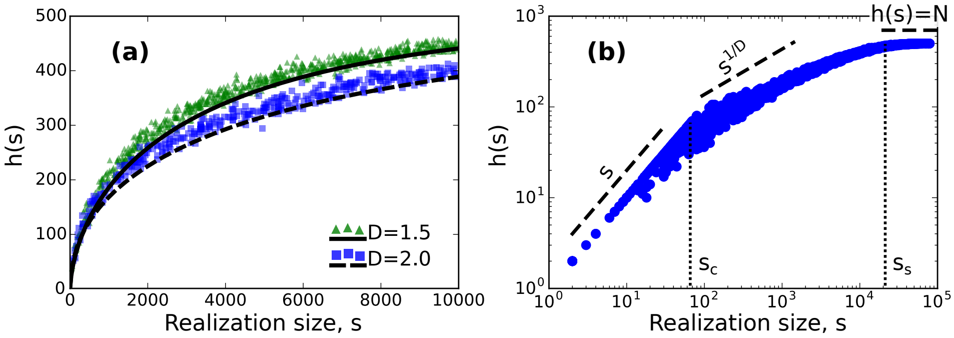


Figure 4.3: **Heaps' law from dependency structures.** A component system constrained by a dependency structure generates the well-know sub-linear scaling of the number of different components with the realization size. The expression (4.12) approximately reproduces the average of the simulations, panel (a), with parameters: $N = 500$, $R = 2000$, and ρ uniformly distributed in $[1, 2000]$. The panel (b) shows the law in double logarithmic scale, highlighting the three expected regimes: the linear growth for $s < s_\mathrm{c}$, Eq. (4.14), the sub-linear scaling,, and the saturation for $s > s_\mathrm{s}$, Eq. (4.16) (parameters: $N = 500$, $D = 2$, $R = 2000$, $\rho \in [1, 3000]$).

4.4.2 Linear, sub-linear, and saturating regimes

Most of the studies about the Heaps' law consider the limit of an infinite vocabulary of the component-universe [55, 56], leading to the classical prediction that the law grows as a sub-linear power-law function. This is justified by the fact that literary texts and, in general, component systems show this phenomenology, i.e. the vocabulary of the "universe" is actually infinite, or the data-set is a sample of the universe with a vocabulary much smaller that the original one. However, the presented model allows us to fully characterize the vocabulary scaling.

Naively, if a realization is constructed by incremental addition of randomly chosen components, one expects $h(s)$ to be approximately linear for small s, as it is unlikely to draw the same component twice. At the same time, this innovation probability decreases with s, up to a point where approximately all components in the universe will have been included, and $h(s)$ will saturate to N. This behaviour is brought out clearly by plotting $h(s)$ in log-log scale (see Fig. 4.3b). There emerge three distinct regimes: a linear increase for small s, a saturation to N for large s, and an intermediate regime where the sub-linear increase of $h(s)$ appears to be well described by the classical power law prediction $h(s) \approx s^{1/D}$ (where the exponent is fixed by the Zipf's law) [52, 55, 56].

In order to quantitatively characterize these three distinct regimes, it is useful to apply the recurrence relation of the incomplete Gamma function (Eq. (B.11)) to Eq. (4.12). This leads to the following expression:

$$\frac{h(s)}{N} = 1 - e^{-\frac{s}{\Omega}} + \left(\frac{s}{\Omega}\right)^{\frac{1}{D}} \left(\Gamma\left(1 - \frac{1}{D}, \frac{s}{\Omega}\right) - \Gamma\left(1 - \frac{1}{D}, \frac{sN}{\Omega}\right)\right). \quad (4.13)$$

The initial linear regime holds true if the term $\frac{sN}{\Omega}$ is much less then one, or, in other words, if:

$$s \ll s_c = \frac{\Omega}{N} = \frac{D}{D-1} N^{1-1/D}, \quad (4.14)$$

which allows us to express the difference between the two incomplete upper Gamma function as an incomplete lower Gamma function, and then to apply the expansion for $\frac{sN}{\Omega} \ll 1$:

$$\Gamma\left(1 - \frac{1}{D}, \frac{s}{\Omega}\right) - \Gamma\left(1 - \frac{1}{D}, \frac{sN}{\Omega}\right) \approx \gamma\left(1 - \frac{1}{D}, \frac{sN}{\Omega}\right) \approx \frac{D}{D-1}\left(\frac{sN}{\Omega}\right)^{1-\frac{1}{D}}.$$

Combining the identity above and the equation (4.13), one finds the expected linear growth:

$$h(s) \approx s. \quad (4.15)$$

The opposite regime appears when all the different components have been extracted, and the realization vocabulary is then equal to the vocabulary of the "universe" N. This happens for:

$$s \gg s_s = \Omega = \frac{D}{D-1} N^{2-1/D}, \tag{4.16}$$

or, equivalently, $\frac{s}{\Omega} \gg 1$. Under this condition, one can approximate the upper Gamma function at the first power of its asymptotic series, neglecting terms of order $\frac{\Omega}{s}$ or smaller. This eventually leads to:

$$h(s) \approx N. \tag{4.17}$$

In the intermediate region of these two extremes the vocabulary grows sub-linearly. As said above, the classical prediction for such a region is a power law function with an exponent equal to $1/D$. We can recover that imposing $s_c \ll s \ll s_s$. The first incomplete Gamma function of Eq. (4.13) becomes approximately equal to the Euler Gamma with argument $1 - 1/D$, while the second one can be approximated again with the asymptotic series. After some calculations one finally finds:

$$h(s) \approx \Gamma\left(1 - \frac{1}{D}\right) N \left(\frac{s}{\Omega}\right)^{\frac{1}{D}}. \tag{4.18}$$

4.5 Discussion

The main purpose of the present work is to establish a quantitative relationship between statistical properties of a component system and its underlying dependency structure, i.e. the functional dependencies between its components. To this end, we introduced a generative model (Section 4.2.2) which generates a new component system considering a dependency network, and constraining the system to obey such dependencies. Although whatever network can be used as a input of the model, we focused on the dependency structure defined in [2]. There are two main reasons for that. The first one is that the generative model of the network is extremely simple (defined only by two parameters), though leading to non-trivial topological properties. The second reason is that the work [2] employs such a network to take into account statistical properties of binary component systems with promising results. By means of our model, we can extend these results to component systems with abundances (i.e. non-binary), which show a richer set of statistical patterns, including the Zipf's and Heaps' laws. Both these two laws can be analyzed by simulations and approximate calculations as shown in Sections 4.3, 4.4. Specifically, the model reproduces a scale-free rank plot of the abundances with an exponent related to the average out-degree of the dependency network, Eq. 4.5 and Fig. 4.2a. The model also leads to an approximate prediction of the sub-linear scaling of the realization-vocabulary with the size, $h(s)$, Eq. 4.12 and Fig. 4.3a, and allows us to identify the different scaling regimes, Fig. 4.3b: linear ($s < s_\mathrm{c}$), sublinear ($s_\mathrm{c} < s < s_\mathrm{s}$), and at saturation ($s > s_\mathrm{s}$).

A possible question is whether the relationships between statistical laws predicted by random models are satisfied in the presented framework. We proved in Chapter 3 that the occurrence distribution can be viewed as a derivative phenomenon of the Zipf's law. This specific prediction seems to be confirmed also here, as shown by Figure 4.2b, where the occurrence-frequency scatter plot follows the random sampling expectation. Analogously, the Heaps' law can be viewed as a statistical consequence of the abundance statistics (as discussed in Section 2.2.2). Here, the analyical prediction of $h(s)$ is derived neglecting correlations (and leading to good approximations of the simulations), therefore suggesting that the null Heaps-Zipf relation is satisfied as well. The fact that the null-predictions give good results seems to be in contradiction with the inner definition of the model, which assumes dependencies between components, and therefore the presence of non-trivial correlations (which instead a random model neglects). We can better understand the problem looking at the mutual information between pair of components, which can be used as a measure of correlation:

$$I(i,j) = \sum_{x,y \in \{0,1\}} p_{x,y}(i,j) \log \left(\frac{p_{x,y}(i,j)}{p_x(i) p_y(j)} \right) \tag{4.19}$$

where $p_0(i)$ is the fraction of realizations in which the component c_i is absent, while $p_1(i) = 1 - p_0(i)$ is the fraction of realization in which c_i is present (exactly equal to the definition of occurrence, o_i, of Table 2.1). $p_{x,y}(i,j)$ defines the co-occurrences of the components c_i and c_j, for example $p_{0,0}(i,j)$ is the fraction of realizations in which they are both absent, or $p_{0,1}(i,j)$ is the fraction where c_i is present and c_j is absent. Note that the mutual information is maximal, and equal to the Shannon entropy of one of the nodes, when the two components have the same co-occurrence pattern: $p_{x,y}(i,j) = p_x(i)\delta_{x,y} = p_x(j)\delta_{x,y}$, while it is null if the two components are independently distributed among the realizations: $p_{x,y}(i,j) = p_x(i)p_y(j)$. Figure 4.4 shows the distribution the mutual information comparing a system generated through our model (blue line) and its "reshuffeld" version (grey dotted line, using the random sampling described in Section 3.2). Most of the component-pairs in both the two distributions have very small mutual information and generate a large peak around 0 (note that the y-axis scale is logarithmic). This can explain the fact that relationships between statistical laws can be predicted with null models, indeed the majority of the pair-correlations are indistinguishable from a random sampling. However, there is a small but significant difference between the two distributions: the model shows a fat-tail, which clearly deviate form the reshuffling, highlighting the presence of the expected correlations generated by the dependency structure.

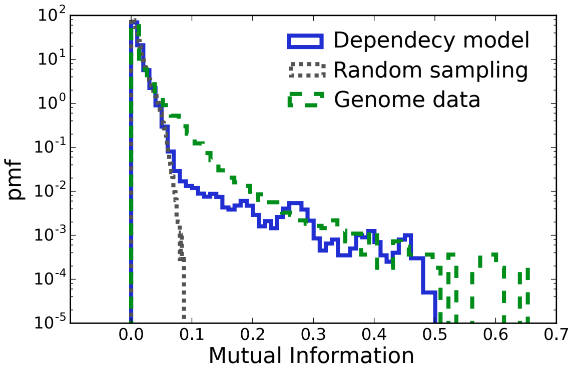


Figure 4.4: **Mutual information distribution.** The panel shows the distribution of the mutual information between each pair of nodes (Eq. (4.19)) in three systems: one generated through the dependency-network model ($D = 1$, $N = 2000$, $R = 1000$, $\rho \in [50, 1000]$), a random sampling at fixed abundance and size distribution of the previous system, and the protein domains-genomes matrix (Appendix A.1.1).

Figure 4.4 also shows the distribution of the empirical genomic system, which displays a similar behaviour of the dependency-network-model, deviating from the null-distribution with a fat-right-tail. This suggests the presence of dependency relations (or at least non-trivial correlations between components) also in genomes. However, it is important to point out that the presented model (at least in its current basic formulation) is not a suitable reference framework for genomes. This because it does not reproduce the empirical phenomenology of the abundance rank distribution (power-law decay + exponential cut-off as shown in Fig. 2.3d and A.1b). Since the dependency network is strongly affected by this distribution, we cannot infer properties of the empirical dependency structure by the comparison with the discussed model. Therefore, Fig. 4.4 provides just an hint that there are non-trivial dependencies in genomes which a random model cannot take into account.

Chapter 5

Heaps and U-shaped laws in sample space reducing processes

Authors: Andrea Mazzolini, Alberto Colliva, Michele Caselle, Matteo Osella.

5.1 Introduction

As discussed in Section 2.2.1, the Zipf's law is a central topic in the study of complex systems. A large variety of models have been proposed to understand its origin the reason of its generality across very different systems. In recent works [39, 70, 71] Corominas-Murtra and co-workers introduced a simple stochastic model, called Sample Space Reducing process, SSR, which joins the mechanisms for the Zipf's law generation. The model is based on the idea that, given a finite set of states, the number of possible visiting states diminishes as the time passes, with a sort of shrinking of the configuration space. Intuitively, this mechanism resembles ageing systems, where the number of "new states/possibilities" reduces as the system ages, or also the writing process, where, after each written word, the number of possible following words reduces as a consequence of significance or syntactic constrains (i.e. the state space shrinks). The SSR process provides a minimal description for all the system characterized by a reduction of the state space during their evolution, linking this mechanism to the generation of power laws in a very elegant way (the definition of this model is presented in Section 5.2).

The simplest implementation of this mechanism leads to the classical Zipf's law (with exponent -1) [39, 70], which is proven to be very general also for non-homogeneous visiting probabilities of the state space [71]. Recently, the author have proposed also a generalization which combines the

SSR mechanism with a multiplicative process (called Sample Space Reducing Cascading process, SSRC), allowing them to recover the full spectrum of scaling exponents of the Zipf's law [72]. A further generalization including a space-dependent noise leads to several other classes of statistical distributions different from power laws [73].

The present work addresses the question whether the discussed mechanism is able to take into account other statistical patterns shown by complex systems, and therefore to what extent the SSR model reproduces the global statistical properties of empirical systems.

The first statistical law considered here is the Heaps' law, which describe how the number of different words in a text grows with the text size (see Section 2.2.2). Typically this law presents a sub-linear growth (approximately a power law function with exponent lesser than 1), which is essentially due to the fact that, after reading few words in a book, there is an high chance that the next word is new, while as the number of read words increases this chance decreases, because the most frequent words have been already discovered and the vocabulary enlarges only by reading specific and rare words. Therefore, the Heaps' law is strictly related with the statistics of the word frequencies, coded by the Zipf's law, and indeed the relationship between the two laws is indeed investigated in some papers [55, 54, 56, 6]. The section 5.3 will show that the SSR model is able to catch this statistical pattern, reproducing the same features of the empirical Heaps' law. Moreover the Zipf-Heaps relationship known in literature is satisfied also in the SSR process, allowing us to derive a reliable analytical estimation of the Heaps' average of the model.

In addition to the Heaps' law, the section 5.4 shows that the SSR model can generate the "U"-shaped law or distribution shared components (see Section 2.2.3). In particular, the SSR model can be used to generate a component system, where the different realizations are different instances of the model, and the system-components are the states of the sample-space. This allows us to compute the component-occurrence distribution as a consequence of the sample state reducing mechanism. As for the Heaps' law, an analytical expectation of "U" shaped law will be derived in the section 5.4, linking its features to the model parameters.

Finally, the section 5.5 will discuss a linguistic example (the book "The Origin of Species"), showing that the model is able to take into account its Zipf's law and Heaps' law, but not other quantities related to temporal correlations between words. These "higher order" statistical properties cannot be generated through the SSR mechanism, posing an interesting challenge on finding the minimal ingredients at the basis of their non-trivial behaviour.

5.2 The Sample Space Reducing Process

The basic definition of the sample space reducing (SSR) process [39], is the following: given a system composed of N states labelled from 1 to N, at the first time step one of those states is chosen randomly, for instance the number k. At the second step only the first $k-1$ elements are accessible, and the second state is drawn randomly among them. The process iterates, each time with a sort of shrinking of the accessible states, and ends when the number 1 is chosen. In other words, a single instance of the SSR process, which is denoted with ϕ, generates a strictly decreasing sequence of the selected states, which always ends with 1.

We can define a generative stochastic process through independent repetitions of ϕ, where a repetition means that, once the state 1 is selected, the process restarts. The repetition of ϕ for S times can be used to create a certain realization adding each selected state/component during the process to the realization. At the end of these iterations, the realization is a string of several components, such as a book composed of words, and the different words are the different states of the process. Clearly, the components labelled with low numbers are chosen more frequently, and for instance the component 1 has the abundance equals to S because is selected at each run of ϕ. On the other hand, the component N can be chosen only at the first step of ϕ with probability $\frac{1}{N}$, leading to an expected abundance equal to $\frac{S}{N}$. Indeed, it can be proven that the average abundance of the i-th component is proportional to i^{-1} in the limit $S \to \infty$, which is the well-known Zipf's law.

A more general model can be defined adding a multiplicative process to the SSR prescription [72]. This new model is called Sample Space Reducing Cascade process, SSRC, and its definition is described in the following (a schematic representation is shown in Fig. 5.1). Given a space composed again of N states, at the first time step μ balls are thrown at random, hitting independently one of the states ($\{c_1, \ldots, c_\mu\}$, $c_i \in \{1, \ldots N\}$) with uniform probability ($P(c_k = i) = \frac{1}{N}$). Then, at the next step, each of these μ balls, for example that one in c_k, splits up into new μ balls, which are thrown at random under the SSR condition: only the first $c_k - 1$ states are accessible. When a ball reaches the state 1, it is removed from the process. Eventually, all the generated balls will hit the state 1, concluding then a "cascade". After that, the process can restart repeating the move of the first step. Note that the model is equivalent to the SSR model for $\mu = 1$. It can be proven that, asymptotically, the number of times that the state i is selected is proportional to $i^{-\mu}$, reproducing whatever exponent of the Zipf's law. This result is true for every $\mu > 0$, where the processes with a non-integer μ can be defined extracting the number of new balls at each step from a distribution with average μ.

For the rest of this paper we will refer to the notation $\phi_s^{(\mu)}$, which indi-

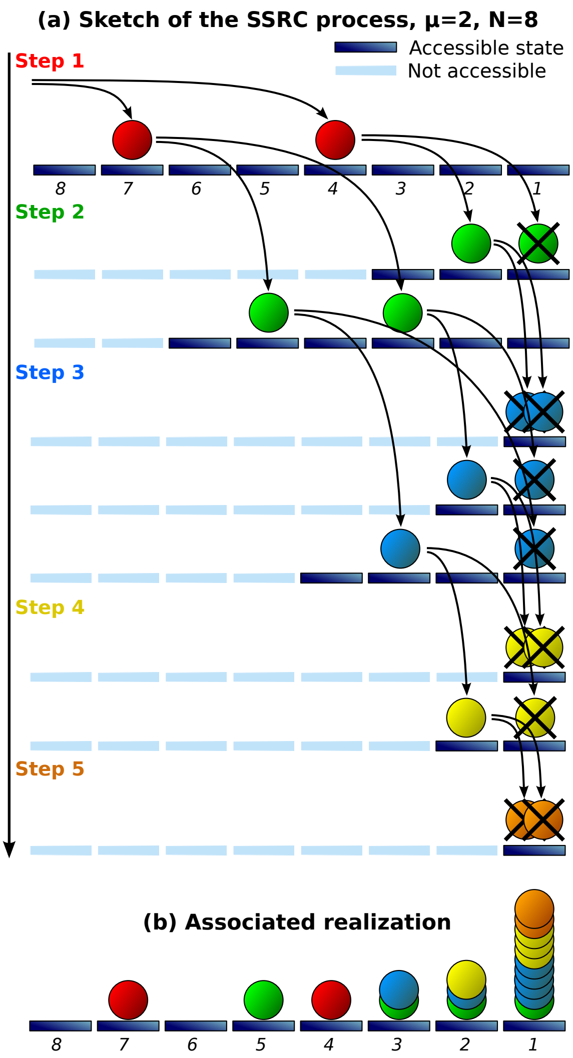


Figure 5.1: **Schematic representation of the SSRC processes.** a) At the first step all the state are accessible and μ of them are chosen with uniform probability (red balls in the states 7 and 4). At the next step, each ball divides into μ new balls, which jump “forward” to a state whose index is lesser than the one occupied by the original ball (e.g. the red ball in the state 4 splits up into two other balls, which can access only to the states $\{1, 2, 3\}$). When the state 1 is chosen, the ball does not divide anymore. When all the balls hit the state 1 a “cascade” finishes, and the process restarts throwing at random μ balls as in the step 1 of the figure. The SSR model can be associated to the growth process of a certain realization (e.g. a book) in which the selected states (e.g. the words) are added to it, as shown in the panel (b).

cates a SSRC process that stops after s drawn components. The value of s will be chosen always much greater than 1, regime in which the component abundances follow approximately a power law function. Note that the indication $\phi_s^{(\mu)}$ does not specify completely the process. Indeed, in the general case of a real positive μ, one has to define the distribution from which the number new of balls is extracted at each iteration (which must have average μ). To avoid ambiguity, for the rest of this paper a Poisson distribution will be always chosen (if not specified differently), also in the case of integer values of μ. It must be pointed out that the following results for the Heaps' law and the "U" shaped law are derived in a mean field approximation, implying that they depend on the average μ but not on the other moments of the distribution. However, in general, the statistical properties of the string of components can change varying the distribution shape, as for the inter-occurrence distance shown in the section 5.5.

5.3 The sample space reducing mechanism generates the Heaps' law

In linguistics, the single-realization Heaps' law is defined as the number of different words, $h(l)$, in the first l words of a text. Note that this law can be computed only if the order of the states/components in the realization is defined. In this regard, within a SSR realization the order is given by the time in which the states are selected during its run. As shown in Fig. 5.2, the SSR process $\phi_s^{(\mu)}$ shows the expected power law distribution of the state abundances (the Zipf's law), panel a, and it generates also a Heaps' law displaying the sub-linear growth typical of empirical systems, panel b. For each set of parameters we have plotted four independent trajectories, giving a qualitative idea of the dispersion around the average. The rest of this section is dedicated to derive the analytical formula of the Heaps' average $\langle h(l)\rangle$.

It is known that a SSR model, $\phi_s^{(\mu)}$, with $s \to \infty$, generates the following occupation probability of a generic state i, as derived in [72]:

$$p^{(\mu)}(i) = \frac{i^{-\mu}}{\alpha}, \qquad \alpha = \sum_{k=1}^{N} k^{-\mu}. \tag{5.1}$$

Assuming that the realization grows through independent extractions of the components with probabilities fixed by the expression above, the chance of choosing for the first time the component i at after l extracted components is given by:

$$\left(1 - p^{(\mu)}(i)\right)^{l-1} p^{(\mu)}(i),$$

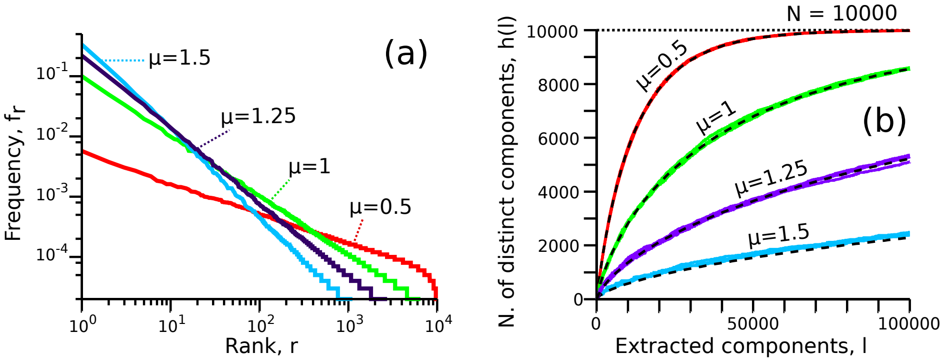


Figure 5.2: **Zipf's law and Heaps' law of the SSRC model.** The panel (a) shows the rank plot of the component frequencies of four realization of a SSRC model, where the number of states chosen at each time step is drawn from a poissonian distribution with average μ. The simulations confirm the theoretical expectation (5.1), in which the power law exponent is equal to the parameter μ. The number of different components, $h(l)$, grows in a sub-linear fashion with the number of extracted components, l, as shown in the panel (b). All the trajectories saturate to the asymptotic value $h(l) = N$ (black dotted line), where N is the second parameter of the model representing the total number of components, i.e. the dictionary size, and is chosen equal to 10000 for all the simulations. The black dashed lines, which overlap the heaps trajectories, are computed with the analytical function (5.4), providing a very good estimate of the mean value of the Heaps' process.

which implies that, after l steps, the component i is selected at least one time with probability:

$$q_l^{(\mu)}(i) = \sum_{k=1}^{l} \left(1 - p^{(\mu)}(i)\right)^{k-1} p^{(\mu)}(i) = 1 - \left(1 - p^{(\mu)}(i)\right)^{l}, \tag{5.2}$$

and then, in a mean field approximation we can write down the expected value of the Heaps' curve, i.e. the number of different components after l selected components:

$$\langle h(l) \rangle = \sum_{i=1}^{N} q_l^{(\mu)}(i) = N - \sum_{i=1}^{N} \left(1 - \frac{i^{-\mu}}{\alpha}\right)^{l}. \tag{5.3}$$

This expression can be simplified under the approximation of large l and large N. To this end, the summation appearing in the formula above can be expressed as the summation (B.7) in Appendix B.2, specifically as $\mathcal{H}(l, 1, 1/\alpha)$. Using the result (B.9) and omitting the expected value notation, one finds:

$$h(l) \approx N - \frac{1}{\mu}\left(\frac{l}{\alpha}\right)^{1/\mu}\Gamma\left(-\frac{1}{\mu}, \frac{l}{N^{\mu}\alpha}\right), \tag{5.4}$$

where Γ is the incomplete Gamma function. This function is verified in Fig. 5.2, where it overlaps the simulation trajectories. As for the the Zipf's law, Eq. (5.1), this estimate depends only on the average μ of the new-balls-distribution.

For $\gamma > 1$ it is also possible to recover the classical sub-linear power law growth of the Heaps' law [55]. Indeed, when $\frac{l}{N^{\mu}\alpha} \ll 1$ (i.e. realization far away from the saturation point) the result (B.12) leads to:

$$h(l) \approx \left(\frac{l}{\alpha}\right)^{1/\mu}\Gamma\left(1 - \frac{1}{\mu}\right), \tag{5.5}$$

where Γ is the Euler Gamma function, and the exponent of growth is the inverse of the Zipf's law exponent.

$$\mathcal{H}(s, a, c) \approx \frac{(cs)^{1/\gamma}}{\gamma}\Gamma\left(-\frac{1}{\gamma}, cN^{-\gamma}s\right). \tag{5.6}$$

5.4 Distribution of shared components form an ensemble of SSR realizations

Given an ensemble of realizations, the distribution of shared components is defined as the distribution of the occurrences, i.e. the fraction of realizations which contain the component (see table 2.1). As discussed previously (see Section 2.2.3), several genomic studies investigate the law for different strains and phyla, finding a “universal” shape characterized by a power law decay followed by a peak at maximal occurrence. Here we tackle the question whether the sample space reducing mechanism can reproduce a similar behaviour. It is worth mentioning that the components/states of the SSR model are “labelled”, i.e. each one is identified by an index from 1 to N, allowing us to study the properties of each component across different realizations. In other words, given an ensemble of SSR strings generated independently and sharing the same state space, one can study the behaviour of a generic component in each realizations, for example computing its total abundance or the fraction of realizations in which it is present (i.e. its occurrence). Other well-known generative models, such as the Chinese Restaurant Process [74] or the Simon's model [31], do not show this property. Indeed, when a new component is added to the realization, it has no relationship with the components in other realizations. Therefore, differently from these models, the SSR process provides a natural framework to study the component occurrence statistics.

In order to reproduce the distribution, we generate a set of R realizations in a state space composed of N states/components. Each realization is a sequence of components provided by the process $\phi_s^{(\mu)}$. The Fig. 5.3a shows three examples of the occurrence distribution, $P(o)$, computed for three values of μ. All the three examples show the characteristic U-shape present in empirical data: a peak at the minimal occurrence, a second peak at $o = 1$, and a power law decay for small occurrences (inset of Fig. 5.3a).

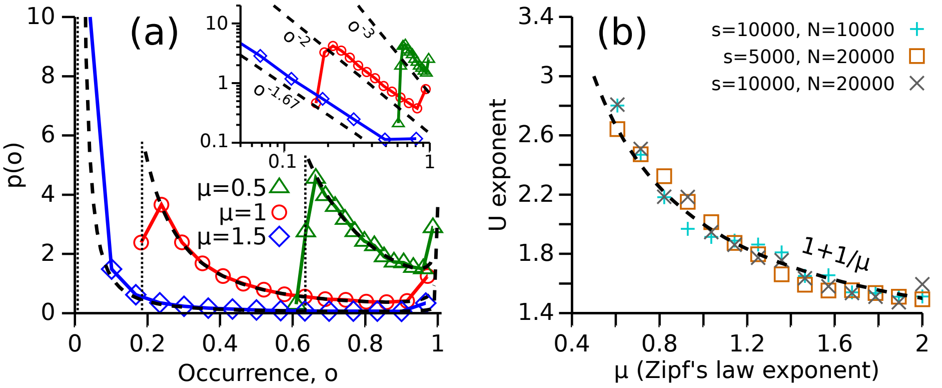


Figure 5.3: **Component occurrence distribution.** The first panel shows the component occurrence distribution for three ensembles of $R = 1000$ realizations of the SSRC process. Each ensemble has a different value of μ, while the other two parameters are fixed: $s = 10000$ and $N = 10000$. The three distributions are in good agreement with the analytical predictions (5.8) (dashed black curves), whose left boundaries, o_{left}, is indicated with vertical dotted lines. The inset shows the same distribution in logarithmic scale, displaying the power law decay, with an exponent given by the relation (5.10). In the panel (b) this exponent is computed for different ensemble parameters (μ on the x-axis, s and N are indicated in the legend), and they are compared with the theoretical expectation (black dashed line), which is independent of s and N. Each dot is obtained through a least square fit of the occurrence distribution. The fitted region is: $[o_{left}+\epsilon_1; o_{right}-\epsilon_2]$, where o_{left}/o_{right} are the left/right domain boundaries (equation (5.9)), while ϵ_1/ϵ_2 are two positive constants, which have been manually tuned in order to remove the “finite-size” cut-off for occurrences near o_{left}, and the increasing part on the right side of the distribution.

Using a similar approach of the previous section, these features and their relationship with the model parameters can be derived analytically. Here the relevant observable is the component occurrence, which can be computed considering the probability that the component i is present in a realization of size s. This probability is given by the expression (5.1), leading to the following expected value:

$$E[o_i] = \frac{1}{R}\sum_{j=1}^{R} q_s^{(\mu)}(i) = 1 - \left(1 - \frac{i^{-\mu}}{\alpha}\right)^s . \qquad (5.7)$$

Note that here we are considering the most simple case in which the probabilities $q_s^{(\mu)}(i)$ are identical for each realization (all of them have the same s and μ). Therefore they do not depend on the index j and the summation is trivial. In general, one can generate an ensemble of realizations with different sizes $\{s_j\}$, and different duplication parameters $\{\mu_j\}$, providing much more complicated scenarios.

This expression is equivalent to the expected occurrence derived in Chapters 3, Eq. (3.2) for random extractions at fixed power law abundance distribution with exponent μ. Therefore we can extend the results derived in that work to the present study. In particular, the analytical formula for the component occurrence distribution is:

$$p(o) = \frac{(1-o)^{\frac{1}{s}-1}}{\mu s N \alpha^{\frac{1}{\mu}} \left(1-(1-o)^{\frac{1}{s}}\right)^{\frac{1}{\mu}+1}}, \qquad (5.8)$$

which is defined in the interval $[o_{left}, o_{right}]$, where:

$$\begin{aligned} o_{left} &= E[o_N] = 1 - \left(1 - \frac{N^{-\mu}}{\alpha}\right)^s , \\ o_{right} &= E[o_1] = 1 - \left(1 - \frac{1}{\alpha}\right)^s . \end{aligned} \qquad (5.9)$$

This expressions are verified in the Fig. 5.3a, main panel, (dashed lines, the vertical dotted lines represent the left domain boundaries). Note that for $s \gg 1$ one has: $o_{right} \simeq 1$, indeed the three examples in the figure are in this regime ($s = 10000$), showing $o = 1$ as right boundary.

It can be proven that the occurrence distribution decays as a power law function for rare components imposing the limit $o \ll 1$ and $s \gg 1$ in (5.8), which becomes:

$$p(o) \simeq \frac{s^{\frac{1}{\mu}}}{\alpha^{\frac{1}{\mu}} \mu N} o^{-\frac{1}{\mu}-1}. \qquad (5.10)$$

Therefore the exponent of the occurrence distribution decay is dependent only on μ through a very simple relationship verified in the Fig. 5.3b.

5.5 Comparison with data: where the SSR model succeeds and where it fails

In the following, the SSR process is used as a generative model for the book "The Origin of Species". In principle, the model can mimic the growth process of whatever "entity" made of elementary components, such as, for example, a genome made of genes or a man-made building made of basic modules. However, a linguistic example is a natural choice since the writing process resembles the sample space reducing mechanism.

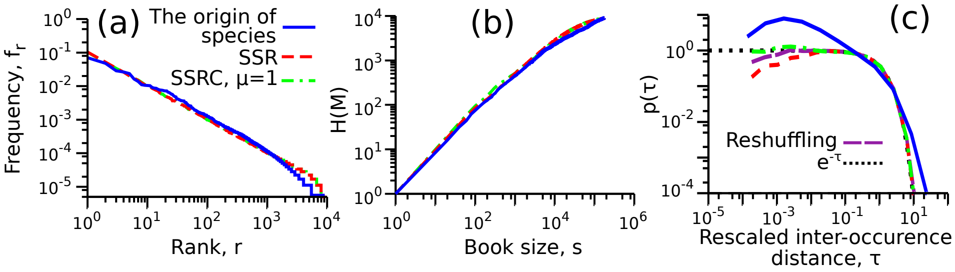


Figure 5.4: **Comparing the SSR model the Origin of Species** The panel (a) shows the Zipf's law of the Origin of Species (blue line), of a basic SSR process (red dashed line), and a SSRC model where the number of new balls are extracted from a Poisson distribution with average $\mu = 1$ (green line-dot line). The number of different words N and the text size s of the two models are the same of the book ($N = 9132$, $s = 178820$). The models generate a power law Zipf with exponent equals to -1, which is qualitatively similar to the real rank distribution. Analogously, the panel (b) displays the qualitative agreement between the Heaps' laws. The plot (c) is the rescaled inter-occurrence distance distribution, equation (5.11), computed considering only the words with abundance greater than 2 and lesser than 1000. The distribution is shown for the book, the two SSR models, and also for the reshuffled string of the SSR model (violet dashed line) and the theoretical expectation of a random process (black dotted line). The curves generated through the SSR, the SSRC and the reshuffled list are the average over 20 independent realizations.

The purpose of the section is to compare the empirical statistical patterns with those generated by the model, specifically the Zipf's law, the Heaps' law, and the inter-occurrence distance (defined below), looking for the statistical properties that arise as a consequence of the sample space reducing assumption. To this end, we have chosen the most simple definition of the model, which is the basic SSR formulation (beginning of Section 5.2), and a SSRC model with $\mu = 1$. Note that the difference between the two models is the distribution from which the number of new balls is generated, specifically a delta function centred in 1 for the basic SSR, and a Poisson distribution with average 1 for the SSRC. The Fig. 5.4a,b shows that the models generate statistical laws qualitatively similar to the empirical ones, specifically a power law Zipf's function and a sub-linear growth of the vocabulary as a function of the book size. It is worth mentioning that the overlap between the curves of the two models is in agreement with the theoretical expectation that the Zipf's and Heaps' law are independent of the shape of the distribution of new balls (but they depend only on the average which is $\mu = 1$ in both the cases).

However, even though the SSRC model is able to reproduce two important statistical patterns of texts, it does not take into account the properties related to temporal correlations between components. Such correlations between words have been extensively studied in linguistics, showing interesting non-trivial features [75, 76, 77]. A simple example of these statistical patterns is called inter-occurrence distance [78, 6]. It is intimately related to the words auto-correlations, and describes the clustering of words in certain regions of texts. For instance, if a character appears only in one chapter of a book, its name is localized in a small region, and completely absent in other parts of the book. This cannot be taken into account by a random model without correlations, which would predict that each word is homogeneously scattered across all the book. In order to quantify this phenomenon, given a given word $i \in V$, one can compute the number of other words between two consecutive instances of i, and weight this value with the frequency f_i. Specifically the inter-occurrence distance of the word i between its $(k-1)$th and kth appearance is:

$$\tau_k = (l_k - l_{k-1})\, f \tag{5.11}$$

where l_k represents the position of the kth appearance of the word, and l_0 is the beginning of the book. It can be proven that for a random generation of the book (e.g. assuming a Poisson growth process), the stochastic variable τ_k follows approximately an exponential distribution with average 1, independently of the word frequency f and the position k. The Fig. 5.4c shows the comparison between the empirical curve and those generated by the two models. As a null expectations the panel displays also the curve generated by the reshuffled list of the SSR process, which is very similar to the expected exponential function, but with a small deviation at small values of τ_k because of the presence of a frequency-dependent lower boundary in the formula (5.11) ($\tau_k^{(min)} = f$). Only the empirical text presents an enrichment at small and large distances with respect the null prediction, which is due to the word clustering discussed above.

Looking at the figure, it can be also noted that the two versions of the model show slightly different inter-occurrence distributions at small τ. This implies that, even though the Zipf and the Heaps' laws depend only on the average of the distribution of new balls μ, in general, other statistical patterns can depend on the shape of this distribution.

5.6 Discussion

In summary we have shown that the sample space reducing mechanism, in addition to the Zipf's law, generates two other important statistical laws: the sub-linear vocabulary growth as a function of the entity size (the Heaps' law) and the U-shaped distribution of shared components, reproducing qualitatively features observed in empirical data. Moreover, the analytical behaviour of the three laws can be derived, linking their properties with the model parameters, equations (5.4), (5.8), (5.10). This extends the range of applicability of the SSR process, which can be used as a benchmark for the study of these statistical laws, joining the models used to reproduce the Heaps' curve [79, 33, 7] and the distribution of shared components [2, 59, 62, 60, 61]. It is remarkable that in the SSR formulation the state space is shared by all the independent realizations of the model, allowing us to study the statistics of the components across realizations, and, in particular, the occurrence distribution. Differently, this is not possible in several commonly used innovation-duplication models, such as the Chinese Restaurant process or the Simon's model.

We have also considered a statistical law based on the temporal autocorrelations between components, named the inter-occurrence distance distribution, equation (5.11) [78, 6]. Its shape in texts is characterized by an enrichment at short inter-occurrence distances, which is not shown by the process, displaying instead a similar behaviour of random models (i.e. without correlations). The non-trivial shape of this distribution highlights a complex generative dynamics posing an interesting challenge on finding the minimal ingredients to generate the observed pattern. A possible way could be considering a process with memory [80, 81] which should lead to the temporal autocorrelations at the basis of this behaviour.

It can observed that the absence of such correlations in the model suggests why the obtained predictions about the Heaps' and the U laws (equations (5.4), (5.8)) are so accurate. Indeed, to derive those equations, it is assumed that at each step the states are extracted independently with a probability fixed only by the Zipf's law (this is necessary to write down the equation (5.2)). The fact that the SSR does not show correlations between components confirms that this hypothesis can be applied, and consistently, the obtained predictions reproduce the simulations. In other words, the SSR model is, somehow, a recipe to generate a realization with a power law abundance distribution, without introducing other constrains on the order of appearance of the components, allowing us to predict the Heaps' laws and the U-shaped distribution with a method based on random and uncorrelated extractions of the components. It can be also noted that models used to establish a null relationship between Zipf's and Heaps' laws are based on similar prescriptions, for example extracting with replacement components with probabilities fixed by a power law function [55], or considering a Poisson growth process where the arrivals of different components follow a rate given by the Zipf's law [56]. As a consequence, in both the two cases the derived Heaps' law are asymptotically equivalent to (5.3). Similarly, the SSR occurrence distribution is equivalent to the one generated by a random sampling from a power law function, Section 3.

Chapter 6

The Heaps' law fluctuations unveil information on the innovation dynamics

Authors: Andrea Mazzolini, Alberto Colliva, Michele Caselle, Matteo Osella.

6.1 Introduction

The approach of the previous two chapters is based on specific assumptions for the system growth, namely the presence of a dependency structure between components, and the sample space reducing mechanism. We then study whether these assumptions are able to reproduce the empirical statistical laws. The present work focuses again on generative models for component systems, but using a different perspective. The starting point is a single and specific observable which shows a non-trivial and universal behaviour across systems. Then, given a wide class of models, we look for the minimal condition that a model must satisfy to explain such observable. In this way, the identified minimal ingredient suggests a key mechanisms at the basis of the empirical systems growth.

More specifically, here we study the fluctuations of the Heaps' law. As discussed in Section 2.2.2, the Heaps' law is largely studied in linguistics, and defines how the vocabulary of system-realizations scales with the realization-sizes. Almost all the studies about this law consider the average behaviour of the vocabulary, which follows a sub-linear growth typically approximated as a power law function (with an exponent between zero and one). However, the huge number of digitalized books, allows, in principle, to characterize the full statistics of the number of different words at a given book-size. Even though this kind of analysis may improve the applications of the Heaps' law in linguistics and memory allocation (see Section 2.2.2), as far as we know, there is only one attempt in this direction [82]. Interestingly, the authors have found that the Heaps' law variance scales quadratically with the average in three different linguistic datasets. This behaviour is called Taylor's law (i.e. the power law scaling of the variance with the average). It was first introduced in ecology [83, 22], where the variable of interest is typically the density of a censused population, whose statistics is computed over a set of different spatial or temporal samples. Several models have been proposed to take into account the super-linear scaling of the variance with the average population density, e.g. [84, 85, 86]. Beyond ecology, this deviation from the classical Poisson-variable expectation (where the variance scales linearly with the average) has been found across several other complex systems, from biology to physics [87, 88, 89, 90].

In [82], the explanation of the quadratic fluctuations of Heaps' law is based on the "topical" aspect of written language, that is to say that the different topics of a text ensemble increase the variability of vocabulary usage. To prove this statement, the authors employ the Latent Dirichlet Allocation (LDA) model [91] to infer the topic composition of a given text ensemble, and also the topic-dependent word frequency distributions (i.e. each topic is characterized by a certain Zipf's law). Performing then a random sampling of words (equivalent to the model presented in Sec. 3) from the topic-dependent frequencies, they recover the empirical fluctuation scaling. As a null-comparison, a random sampling from the global frequency distribution would lead to a variance always less than the average.

Here we focus on the same observable of [82], showing that the Taylor's law with exponent 2 is not only present in linguistics, but also in genomics, suggesting that it is a universal feature of Heaps' law of component systems. We then address the question of its origin using an alternative way (but not necessary in contrast with [82]), i.e. looking for the minimal generative model which reproduces the Taylor's law. Our investigation proves that only a certain class of duplication-innovation growth models is able to take into account a super-linear scaling. These findings suggest that empirical systems grow with a rich-gets-richer mechanism in terms of "vocabulary richness".

6.2 Universal Taylor's law of the vocabulary growth

A universal scaling shown by complex component systems is the sub-linear growth of the number of distinct components, h, with the realization size, s, i.e. the Heaps' law. Figure 6.1 shows this statistical patterns in three datasets: books from the Gutenberg database (panel a, datasets described in Appendix A.2.2), genomes made of protein-domain families (panel b, Superfamily database, Appendix A.1.1), and Wikipedia articles (panel c, Appendix A.2.3).

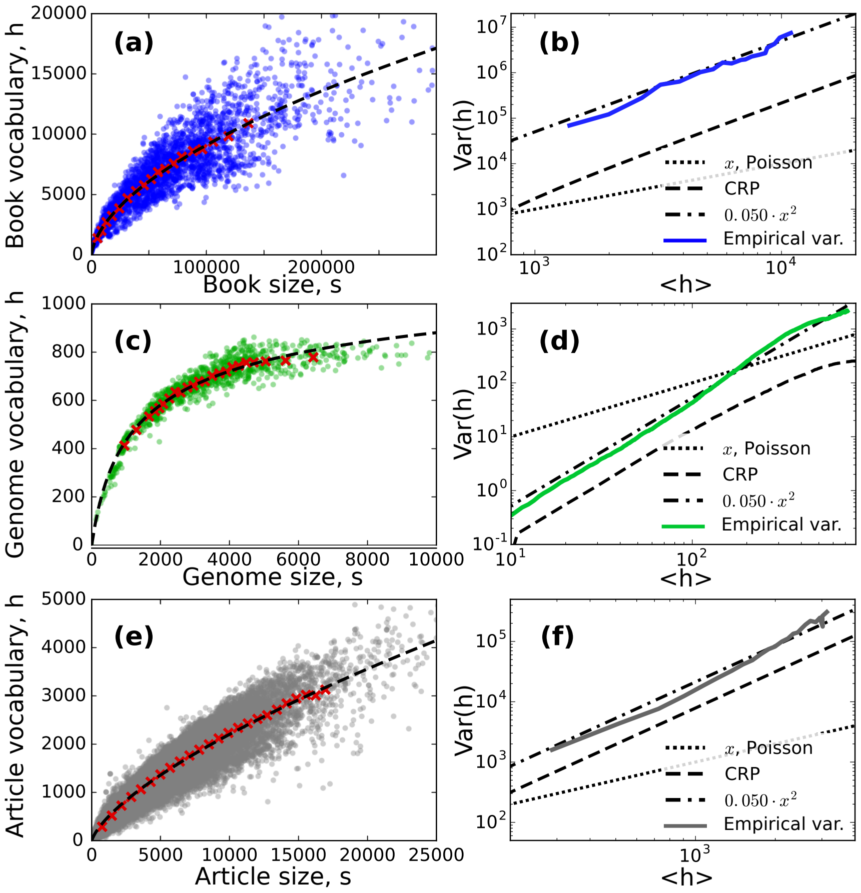


Figure 6.1: **Heaps' law and its fluctuation scaling shown in three datasets.** Data: 3036 books from the Gutenberg database (panels a,b), 1060 bacterial genomes from the Superfamily database (c,d), 8334005 articles from the English Wikipedia (e,f). The Heaps' law (panels a,c,e) shows the growth of the vocabulary size h with the realization size s. The average of h over x-axis bins (red crosses) grows in a sub-linear fashion, and it is well fitted by the mean-field CRP prediction (6.9), with parameters: $\alpha = 0.57$, $\theta = 110$ for books, $\alpha = -0.31$, $\theta = 436$ for Superfamily, $\alpha = 0.68$, $\theta = 27$ for Wikipedia. The right panels (b,d,f) display the variance of h, i.e. the dispersion of the vocabulary size at fixed entity size, as a function of the average, growing approximately as a parabolic function (continuous lines). The procedure to compute these lines is discussed in the main text. The empirical variance is compared with a classical Poisson's process, $\mathrm{Var}[h] = \mathrm{E}[h]$ (dotted lines), and the variance of a simulated CRP (dashed lines) with the parameters fixed by the fit of the average . The dash-dot lines are the best fit of the empirical variance assuming a quadratic scaling: $\mathrm{Var}[N] = c \cdot \mathrm{E}[N]^2$.

Here we focus on the fluctuations of Heaps' law around its average, that is how much realizations of similar size are diverse in terms of vocabulary richness. Intuitively, in the linguistic case, diversity of vocabulary usage means that there are books with a tendency to repeat always the same words (and then having a small vocabulary), as well as books which try to use as many different words as possible.

Before describing the Heaps variance in datasets, it is worth mentioning a technical point about its computation. As described in Section 2.2.2, given an ensemble of realizations, the global Heaps' law can be defined in two alternative ways: (a) it can be the scatter plot of the realization-vocabulary versus the realization-size (where each dot is a realization, as in Fig. 6.1a,c,e), or (b) one can compute the vocabulary-size trajectory for each realization, and then consider the law as the ensemble of all the trajectories. Note that for this latter prescription one needs a certain definition of the component-order within a realization (e.g. in books it is naturally defined as the temporal order in which the words appear). Summarizing, according to (b), each realization contributes to the Heaps' law with the entire trajectory $h(l)$ with $l \in [1, s]$, while, according to (a), the realization contributes only with the last point of the trajectory, $h(s)$ (which is always known regardless of the component-order). Importantly, even though the definitions (a) and (b) may not be equivalent in terms of h statistics, for the linguistics datasets it seems to be true, as discussed in Appendix A.2.4. This allows us to study the vocabulary statistics unambiguously in the linguistic data. The genomics case needs a separate discussion. The number of genomes in the ensemble (around 1000) is not sufficient to have a solid statistics of the vocabulary fluctuation scaling considering the definition (a) (Fig. A.2a shows the variance computed with this procedure). The definition (b) would lead to a much larger number of samples per size-bin, but since the protein domain order is not well-defined, the genome-trajectories cannot be computed. To overcome this problem we adopted the procedure described in Appendix A.1.3, in which we defined artificial trajectories of genome growth through a weighted under-sampling of the protein domains, allowing us to use (b) and obtain a smoother line for $\mathrm{Var}(h)$, panel 6.1d.

The fluctuation scaling of the Heaps' law in the three datasets are shown in Fig. 6.2b,d,f, where the vocabulary variance, $\mathrm{Var}(h)$, at fixed size-bin is plotted as a function of the vocabulary average in that bin, $\langle h \rangle$. The clear quadratic scaling in all the three panels allows us to identify a general Taylor's law for the vocabulary fluctuations in component systems:

$$\mathrm{Var}(h) \propto \langle h \rangle^2. \tag{6.1}$$

Interestingly, a classical poissonian random variable (dotted lines) cannot predict this scaling, showing a linear growth. Moreover, also the random sampling model (described in Section 3.2) deviates from the empirical data.

Indeed, the variance of the Heaps' law can be computed (as discussed in Appendix B.3, Eq. (B.15)) and reads:

$$\mathrm{Var}[h(l)] = \langle h(l) \rangle - \sum_{i=1}^{N} q_i(l)^2, \tag{6.2}$$

where $q_i(l)$ is the probability of extracting at least one time the component with a given frequency f_i after l extractions, Equation (B.13). Since the second term is always negative, the variance is always bounded by the linear growth, and cannot reproduce the Taylor's law. The deviation from this null-predictions says that the observed pattern is due to non-trivial features of component system statistics, requiring more refined models to be taken into account.

6.3 Duplication-innovation models

To explain the Taylor's law (6.1) we considers a simple but wide class of models called duplication-innovation processes. Some of the most famous examples are the Yule-Simon model [30, 31], the Pólya's urn [92, 93], and the Chinese Restaurant process [94, 95, 96]. These are used, for instance, in genomics to mimic the evolutionary process of organisms [15, 97, 98], or in linguistics to generate books and their statistical patterns [34, 7]. They are based on the idea that a system-realization grows adding components through two moves: a duplication of an existing component-instance, or a discovery of a new component. Actually, a third move is often considered: the deletion of a component. However, we will show that duplication and innovation are sufficient to fully characterize the dynamics at the basis of the observation (6.1).

A generic duplication-innovation model can be defined as follows. Let us consider the notation described in 2.1.3, where a realization is a sequence of component-instances $(x_1, x_2, \ldots)$, and such instances belong to the set of unique components $x_k \in \{c_1, c_2, \ldots\}$, where the cardinality of this latter set is called vocabulary size. At the first time-step, the realization is composed of one instance of the component c_1. Its associated sequence is then: (x_1), the unique-components set is $\{c_1\}$, and trivially $x_1 = c_1$. At the generic time step l a new component-instance is added to the realization and the sequence enlarges by one unit:

$$(x_1, \ldots, x_{l-1}) \to (x_1, \ldots, x_{l-1}, x_l).$$

An innovation event occurs with probability p^{new}, meaning that the instance x_l is a new component, i.e. it is not present in the set of unique components at the previous step: $x_l \notin \{c_1, \ldots, c_{h(l-1)}\}$. Therefore, also the component-set enlarges by one unit, which is the new introduced component: $c_{h(l)} = x_l$.

With probability p^{old}, one of the existing component-instances duplicates. In such a case, the unique-component-set does not change, and x_l is chosen equal to one of the existing components according to a rule prescribed by the specific model. To summarize:

$$\begin{array}{lll} & \text{Probability} & \text{Vocabulary size} \\ \text{Innovation} & p^{new} & h(l) = h(l-1) + 1 \\ \text{Ducplication} & p^{old} & h(l) = h(l-1). \end{array} \tag{6.3}$$

Of course $p^{new} + p^{old} = 1$. Note also that without deletion events, the realization size corresponds exactly to the time step l of the model.

Duplication-innovation models are typically used to reproduce the power law behavior of the component frequency/abundance distributions within a realization [30, 31, 95, 34, 33]. Indeed, they are a perfect framework to include the "preferential attachment" assumptions, which can be encoded in the duplication probabilities, leading to scale-free patterns. Note that these results consider the "local" abundance of a component within a realization ($n_i = \sum_k \delta_{x_k,c_i}$), not the "global" abundance a_i defined in Table 2.1. For the rest of this chapter we will always refer to this "local" definition of abundance.

In addition to the abundance distribution, these models can be also used to investigate the statistics the number of unique components as a function of the realization size: $h(l)$, which exactly defines the Heaps' law trajectory. For a generic duplication-innovation model, the average $h(l)$ can be computed with a mean-field approximation, by knowing that in a single step of time the average increment of the number of different components corresponds to the innovation probability: $\langle h(l) - h(l-1) \rangle = p^{new}(l)$. This leads to the following mean-field formula:

$$\frac{\mathrm{d}\langle h(l) \rangle}{\mathrm{d}l} = p^{new}(l). \tag{6.4}$$

Even though this expression is extremely useful for the average, a mean field analysis cannot be performed to compute the variance of $h(l)$ (the actual quantity which we are interested in). In this regard, alternative analytical approaches will be shown later on.

The remaining of this section will describe three innovation-duplication models, which have been used in previous papers to reproduce the sub-linear behaviour of the Heaps' law average and the power law distribution of the abundances. We are going to analyze the Heaps law fluctuations of such models, in order to better understand how the variance is affected by the model-specific assumptions.

6.3.1 Three models for the sub-linear growth of the vocabulary

Generalized Simon's model

The first model is based on the Yule-Simon's process [30, 31, 99], which can reproduce the power law abundance distribution of words (the single-realization Zipf's law) but not the sub-linear vocabulary growth. The authors in [34] proposed then the following generalization for the innovation and duplication probabilities:

$$p^{new} = \beta\, l^{\mu-1} \qquad p_i^{old} = \frac{1-\beta\, l^{\mu-1}}{l} n_i, \tag{6.5}$$

where $0 \leq \beta \leq 1$, $0 < \mu < 1$, and $p^{old} = \sum_i p_i^{old} = 1 - \beta\ l^{\mu-1}$, which correctly normalizes adding p^{new}. Note that the duplication probability of a component is proportional to the local abundance n_i, encoding a "preferential attachment" mechanism, which eventually leads to the Zipf's law $(n_i \propto i^{1/\mu})$.

The Heaps average can be easily computed with the mean field formula (6.4), leading to:

$$\langle h(l)\rangle \approx \frac{\beta}{\mu}\, l^{\,\mu}, \tag{6.6}$$

which is exactly the wanted sub-linear power law growth. However, looking at the variance of $h(l)$, Fig. 6.2a, one finds that it is lower than the poissonian line $\mathrm{Var}(h) = \langle h\rangle$, and therefore cannot reproduce the quadratic scaling. Using the same procedure of Section 6.3.2, the analytical formula for the variance can be computed (for $l \gg 1$, and approximating the summation with an integral):

$$\mathrm{Var}[h(s)] \approx \frac{\beta}{\mu}\, l^{\,\mu} - \frac{\beta^2}{(2\mu+1)}\, l^{\,2\mu+1} \leq \langle h(l)\rangle, \tag{6.7}$$

where the first addendum is exactly the average (6.6), and the second term is always positive (in absolute value). Therefore, this proves that the variance of the generalized Simon's model is always bounded by the poissonian line.

Chinese Restaurant Process

Analogously to the generalized Simon's model, the Chinese Restaurant process (CRP) [94, 95, 96], has been used to reproduce the power law Zipf and the sub-linear Heaps. In particular, it was used to fit the statistical patters of an ensemble of bacterial genomes [33]. The model is typically defined as a customer seating-plan in a Chinese restaurant. When a customer enter the restaurant, he can choose to sit at an already occupied table, preferring the tables with a higher number of people (preferential attachment), or he can choose to sit at an unoccupied table. The first move corresponds to a duplication event (the abundance of a table/component increases), while the second move is an innovation event (a new table/component is discovered). This can is formalized with the following probabilities:

$$p^{new} = \frac{\theta + \alpha h}{\theta + l} \qquad p_i^{old} = \frac{n_i - \alpha}{\theta + l}, \tag{6.8}$$

where $\theta > 0$, $0 \leq \alpha < 1$, and $p^{old} = \sum_i p_i^{old} = \frac{l - \alpha h}{\theta + l}$. Again, using the mean field approximation, the average number of unique components can be computed:

$$\langle h(l) \rangle \approx \frac{1}{\alpha} \left((\alpha + \theta) \left(\frac{\theta + l}{\theta} \right)^{\alpha} - \theta \right), \tag{6.9}$$

which (for large l) scales as a power law with exponent α. In Appendix B.4, Eq. (B.24), we derived the exact expression for the average vocabulary (without the mean field approximation) leading to the same scaling. Therefore, similarly to the generalized Simon's model, the CRP leads to the empirical sub-linear power-law Heaps' law. However, the innovation probabilities in the two processes have different dependencies, generating the power law growth through diverse mechanisms. Both the two p^{new}s decrease as l increases (a necessary condition for the sub-linear growth). But, in the generalized Simon's model, the innovation probability is a power law function of the realization-size, and this directly leads to the power-law growth of h. Instead, the CRP innovation rate is a balance between the realization-size (at denominator) and the vocabulary-size (at numerator), which (in a less intuitive way) constrains the vocabulary size to grow as a power-law.

These two different mechanisms at the basis of the vocabulary growth are asymptotically equivalent looking at the average, but they lead to completely different results considering the variance of h. Indeed, in Fig. 6.2 the CRP line shows a quadratic scaling of Var(h) as a function of $\langle h \rangle$. This behavior is confirmed also by the analytical resolution of the vocabulary size statistics shown in Appendix B.4, and, in particular, by the analytitcal expression of the variance (B.26) for $l \gg 1$:

$$\text{Var}[h(l)] \approx \left(\frac{(\theta + \alpha)\Gamma(\theta + \alpha)^2}{\Gamma(\theta + 2\alpha)\Gamma(\theta + 1)} - 1 \right) \langle h(l) \rangle^2. \tag{6.10}$$

Generalized Pólya's urn

The third model is based on the definition of the Pólya's urn [92, 93]. The authors of [79] generalized the well-known model with the assumption that the space of new possible components enlarges when a novelty occurs (i.e. an innovation event). Specifically, at the initial time, the urn contains h_0 distinct components. At each further time, a uniform random extraction is performed: the selected component is added to the realization and then put back in the urn together with ρ additional instances of its type (this leads to the preferential attachment). In the case of a novel component (i.e. not present in the realization-sequence), $\nu + 1$ brand new distinct components are added to the urn. In such a way, a novelty increases the possibility of discovering other novelties. This definition leads to the following innovation-duplication probabilities:

$$p^{new} = \frac{h_0 + \nu h}{h_0 + (\nu+1)h + \rho l} \qquad p^{old} = \frac{\rho n_i + 1}{h_0 + (\nu+1)h + \rho l} \tag{6.11}$$

The mean field expression for the Heaps' law average reads (for large l, and $\rho > \nu$):

$$h(l) \approx (\rho - \nu)^{\frac{\nu}{\rho}} l^{\frac{\nu}{\rho}}, \tag{6.12}$$

where, again, the power-law growth with an exponent lesser than one is recovered. This could have been predicted looking at the innovation probability in the large l limit, $p^{new} \approx \frac{\nu}{\rho}\frac{h}{l}$. This expression is equivalent to the innovation probability of the Chinese restaurant process (again for large l) identifying $\frac{\nu}{\rho}$ with α, and therefore one can expect the same scaling behaviour. This equivalence seems to be present also for the variance of h. Even though, we were not able to derive an exact analytic prediction, the Figure 6.2a shows the quadratic scaling as in the CRP and in the empirical systems.

6.3.2 A necessary condition for the quadratic scaling

The vocabulary size statistics is independent of the specific mechanism of the component duplication (defined by p_i^{old}). Indeed, the growth of $h(l)$ is driven only by the innovation probability, which determines whether $h(l)$ grows by one unit (probability equals to p^{new}), or remains constant $(1 - p^{new})$. Let us consider the probability of having h different tables at time l: $P(h,l)$. The recurrence relation for this quantity reflects the considerations above:

$$\begin{aligned} P(h, l+1) &= P(h,l) + p^{new}P(h-1,l) - p^{new}P(h,l) \\ &= p^{new}P(h-1,l) + (1-p^{new})P(h,l) \end{aligned} \tag{6.13}$$

with the initial condition $P(h,1) = \delta_{h,1}$.

Therefore, also the mechanism behind the quadratic scaling of the Heaps' fluctuations should be encoded in p^{new}. Looking at the three described models, the Chinese restaurant process and the generalized Pólya's urn succeed in reproducing the empirical scaling, while the generalized Simon's model fails. The key ingredient in the innovation probabilities seems to be the dependency on h, present only in the two successful models (6.8), (6.11). Here, we are going to prove this statement. Specifically, we will show that if p^{new} does not depend on h, then its variance is always bounded by the line

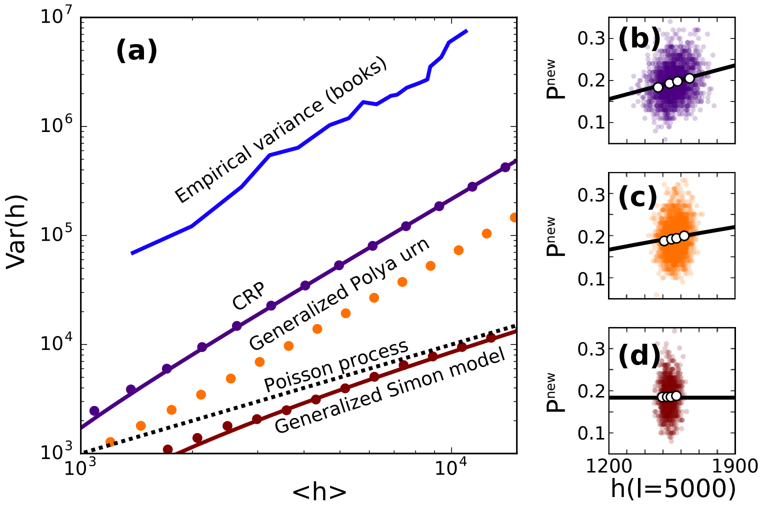


Figure 6.2: **Fluctuation scaling and innovation rates for the three considered models.** The model parameters are fixed by the average heaps fit of the Gutenberg database ($\alpha = 0.57$, $\theta = 110$ for the CRP, $h_0 = 7.43{\cdot}10^4$, $\nu = 147$, $\rho = 250$ for the generalized Polya's urn, and $\beta = 10.8$, $\mu = 0.59$ for the generalized Simon's model). In panel (a) the dots refer to the simulations, while the continuous lines are given by the analytical formula (6.7) and (6.10). The blue line is the variance of the Gutenberg dataset (Fig. 6.1), which scales as the square of the average, like the CRP and the Polya's urn, but with a greater multiplicative coefficient. The Simon's model shows a sub-poissonian scaling as expected. The quadratic scaling is related to the increasing innovation rate with the vocabulary size, panels (b), (c), and (d). Indeed, the p^{new} grows with h in the simulations of the CRP and the Polya's urn, while it is constant for the Simon's model. The black lines are the known innovation probabilities (6.5), (6.8), (6.11), which are in agreement with the binning averages (white dots).

$\mathrm{Var}(h) = \langle h \rangle$. This is equivalent to state that the innovation-probability dependency on h is a necessary condition for the quadratic scaling.

To prove the thesis we look for the recurrence relation of the variance starting from (6.13). First, let us multiply both the terms of by h, and then apply the summation over h:

$$\sum_{h=1}^{\infty} hP(h, l+1) = \sum_{h=1}^{\infty} h\left[p^{new}(l)P(h-1, l) + (1 - p^{new}(l))P(h, l)\right].$$

Note that the left term becomes the average of h at time $l + 1$. Since p^{new} does not depends on the vocabulary size (by hypothesis), it can filter outside the summation. This property is crucial to find the following simple expression for the average (by knowing that $P(0,l) = 0$ and $\sum_h P(h,l) = 1$):

$$\begin{aligned} \langle h(l+1) \rangle &= \langle h(l) \rangle + p^{new}(l) = \ldots = \\ &= \sum_{k=1}^{l} p^{new}(k) + 1. \end{aligned} \tag{6.14}$$

In a similar way, one can find the second moment of h (multiplying by h^2 instead of h):

$$\langle h(l+1)^2 \rangle = \langle h(l)^2 \rangle + p^{new}(l)^2 \; (2\langle h(l) \rangle + 1),$$

and putting together the first and the second moment, the variance reads:

$$\begin{aligned} \mathrm{Var}[h(l+1)] &= \mathrm{Var}[h(l)] + p^{new}(l) \; (1 - p^{new}(l)) = \ldots = \\ &= \sum_{k=1}^{l} p^{new}(k) \; (1 - p^{new}(k)) = \\ &= \langle h(l+1) \rangle - 1 - \sum_{k=1}^{l} p^{new}(k)^2 \\ &< \langle h(l+1) \rangle, \end{aligned} \tag{6.15}$$

where the third line is written using (6.14). Since the innovation probability is greater than zero, it is clear that the variance is always less than the average. Therefore, the hypothesis that p^{new} does not depend on the vocabulary size implies that the variance scaling of the process cannot cross the line $\mathrm{Var}(h) = \langle h \rangle$, and therefore the model cannot reproduce the empirical quadratic scaling.

6.4 Empirical data suggests a rich-gets-richer mechanism in terms of vocabulary richness

The previous section proved that the innovation probability must depends on the vocabulary size in order to give rise to the Taylor's law (6.1). The simplest dependency on h is the linear one, shown, for example, by the Chinese restaurant process (6.8) (which, at the same time, must be inversely proportional to l for taking into account the sub-linear scaling of the average). We proved that, in this case, the variance grows quadratically with the average (6.10). Also the innovation probability of the generalized Pólya's urn depends linearly on h (in the limit $l \gg 1$), and consistently, the model shows the parabolic Taylor's law (Fig. 6.2). Unfortunately, our analytic investigation can handle only this special case of p^{new} linearly dependent on h (the mathematical "trick" which allows us to solve the CRP variance works only for this linear case, see Appendix B.4), and we do not have any analytical predictions for more complex cases. However, the data seems to be well described by this simple scenario. One can obtain an estimation of the innovation probability looking at the discrete derivative of a single-realization trajectory:

$$p^{new} \approx \left\langle \frac{h(l+\Delta l) - h(l)}{\Delta l} \right\rangle, \tag{6.16}$$

which well approximates p^{new} for small Δl. Figure 6.3 shows the numerically computed innovation probability which grows linearly (on average) with the vocabulary size. As a comparison, we also compute the discrete innovation probabilities for the vocabulary trajectories of the three models, as shown in Figure 6.2b,c,d. As expected the CRP and the generalized Pólya's urn have an asymptotic linear dependency (well fitted by their innovation probability formula).

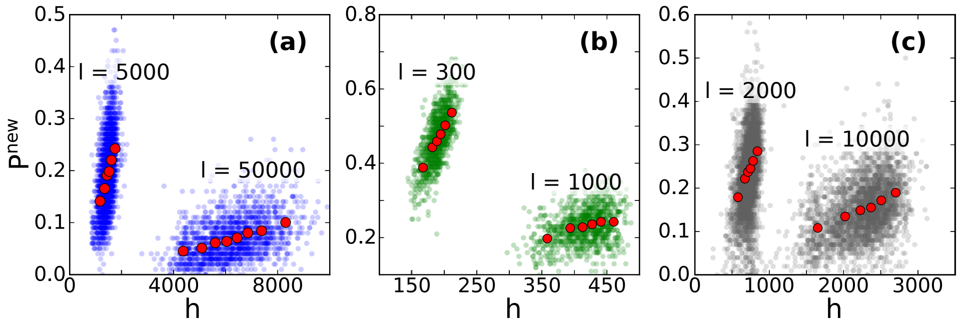


Figure 6.3: **Innovation rates in data show a linear dependency on the vocabulary-size.** Innovation probabilities versus vocabulary-sizes for the three datasets: (a) Gutenberg database's books, (b) Superfamily's genomes, (c) Wikipedia's articles. The scatter plot has been computed considering the ensemble of vocabulary trajectories $h(l)$, and, for each trajectory, we calculated the discrete derivative (6.16) as an estimation of the innovation probability. The red dots are the binned averages, where the x-axis has been divided into bins which have the same number of samples.

Putting all together, we can conclude that the Taylor's law shown by data can be explained by innovation-duplication models with innovation rates linearly dependent of the vocabulary size. This simple mechanism reflects the fact that realizations with a large vocabulary tend to innovate more with respect realizations with similar size but a smaller vocabulary. Imagining several realizations growing together in the size-vocabulary plane, the ones that stay above at the beginning (with a larger vocabulary) tend to innovate more than the others, growing faster because of this self-reinforcement mechanism. At the same time, realizations with small vocabulary grow much slower, leading, at the end, to a variance of the ensemble much larger than the one generated by vocabulary-independent innovation rates. In other words, rich realizations in terms of vocabulary size get richer, acquiring novel components with higher probability than the poor ones.

6.5 Discussion

In this work, the component system framework allows us to generalize the word-vocabulary fluctuation scaling known in linguistics [82]. Interestingly, the scaling is conserved also in genomic systems (Taylor's law with exponent 2), deviating from random predictions in a general and non-trivial way. To explain such observation we employ a duplication-innovation model framework, which establishes a solid connection between the emerging fluctuation scaling of component-system vocabularies and the "microscopic" innovation dynamics of the generative process. Specifically, the minimal hypothesis to reproduce the Taylor's law is that the innovation probability of empirical systems must be linearly dependent on the vocabulary size, as in the Chinese restaurant process. This suggests that empirical systems grows with a rich-gets-richer mechanism in term of vocabulary usage, where the realizations with a larger vocabulary will innovate more than those ones with a smaller number of distinct components (but similar size).

The explanation of Taylor's law proposed here is based on the assumption that a component systems evolves according to a duplication-innovation model. However, in principle, other classes of models can succeed as well. For example, in ecology Taylor's law is very popular, and one can take inspiration from one of the several models employed in that context. Here we consider a recent work as a comparison [86], which derives the quadratic scaling of the variance assuming a very general multiplicative process for the population in a Markovian environment. It must be pointed out that in such a model the variable of reference is the population density. According to the process, it grows proportionally to the density at the previous step and to the environmental state. This model can be safely applied to the Heaps' law growth (i.e. substituting the density with the vocabulary size), and it would provide a valid explanation for the vocabulary fluctuation scaling. However, while the multiplicative process in a stochastic environment has a direct interpretation in ecology, it provides a much less intuitive description of the vocabulary diversity evolution. On the contrary, duplication-innovation models are a natural way to imagine the component systems growth. Moreover, they not only describe how the diversity grows (the vocabulary size is a marginal statistic of the complete process), but they also take into account several other properties of component realizations, such as the abundance statistics (differently from the cited multiplicative process). Summing up, both the methods take into account Taylor's law, but duplication-innovation models (and specifically the CRP) provide a more natural representation of the component system growth, allowing us to make broader predictions, and to give a direct interpretation of the microscopic generative mechanism.

Coming back to the rich-gets-richer mechanism discussed above, one can try to interpret this behaviour in specific empirical systems. For example, in genomics it can be due to specific mechanisms of the evolutionary process, or in linguistics to "topicality", as discussed in the next paragraphs.

6.5.1 Interpretation in genomics

Before speculating on the possible evolutionary mechanisms which can generate the "rich-gets-richer vocabulary diversity" in genomics, here we discuss a testable prediction of this behaviour. It is known that genomes evolve along a phylogenetic tree. Then, let us consider a speciation event. Because of random fluctuation, it can happen that the ancestors of two species have different vocabularies, for example the first organism is "richer" $h_1 > h_2$. We also assume that these two ancestor gives rise two different phyla. Since we are hypothesizing that "diversity generates diversity", we can expect that all the descendant of the first phylum will have a larger vocabulary then the organisms of the second one. Therefore, it can be expected that each phylum (or whatever taxonomic classification) has a specific trend in terms of vocabulary diversity.

Figure 6.4 shows Heaps' law of genomes, colouring the organisms of the three largest phyla in our dataset. Roughly, the behaviour seems to be in agreement with the statement above: the Actinobacteria tend to stay below the average, the Proteobactoeria above, while the Firmicutes are mostly central. To better highlight these trends, we introduce the relative vocabulary size, which quantifies the distance of each bacteria from the global average, $\langle h \rangle$, in units of standard deviation $\sigma[h] = \sqrt{\mathrm{Var}[h]}$:

$$\tilde{v}_j = \frac{v_j - \langle h \rangle}{\sigma[h]}. \tag{6.17}$$

We expect this metric to be independent of the realization size, allowing us to compare all the different organisms together. The probability distribution of the relative vocabulary for the three phyla is shown in the figure inset. As expected, the three distribution separate, showing a specific trend for each taxonomic group. We can also employ a two samples Kolomgorv-Smirnov test to reject the null hypothesis that the distributions are identical ($D = 0.71$, p-value $= 3 \cdot 10^{-35}$ for Proteobacteria vs. Actinobacteria; $D = 0.3$, p-value $= 2 \cdot 10^{-11}$ for Proteobacteria vs. Firmicutes; $D = 0.47$, p-value $= 7 \cdot 10^{-13}$ for Firmicutes vs. Actinobacteria). Moreover, the same test has been applied to the six largest phylum, testing if their relative vocabulary sizes belong to the global distribution of all the other species (results in Table 6.1).

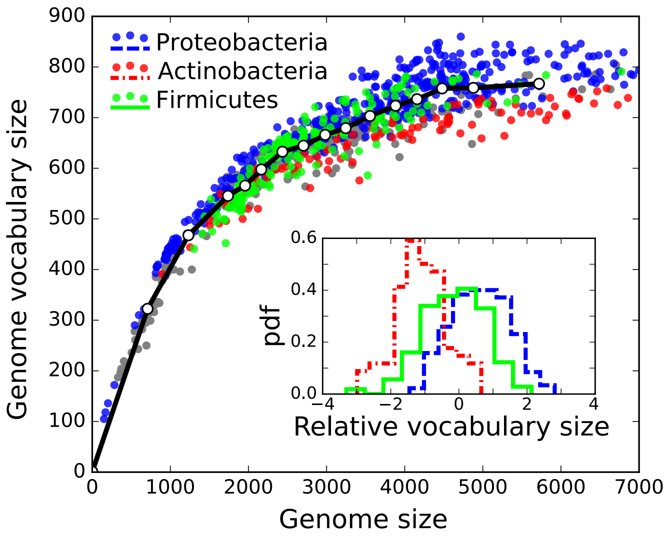


Figure 6.4: **The vocabulary of different phyla show separate trends.** The main panel shows the Heaps's law of the genome dataset, highlighting the bacteria of the three largest phyla: Proteobacteria (in blue, 416 organisms), Firmicutes (in green, 195 organisms) and Actinobacteria (in red, 93 organisms). All the other 356 genomes are in gray, while the white dots connected by black lines are the binning average, where a bin on the x-axes conserves the number of samples. The inset shows the relative vocabulary size for the three phyla, Eq. 6.17.

Table 6.1: Two sample Kolmogorov-Smirnov test, where the first sample is the relative vocabulary size of a taxonomy, compared with the relative vocabulary size of all the other genomes.

Taxonomy	**Size**	**D**	**p-value**
Proteobacteria	416	0.37	$4 \cdot 10^{-30}$
Firmicutes	195	0.14	$3 \cdot 10^{-3}$
Actinobacteria	93	0.57	$2 \cdot 10^{-25}$
Bacteroidetes/Chlorobi group	72	0.31	$4 \cdot 10^{-6}$
Tenericutes	31	0.15	0.49
Spirochaetes	31	0.36	$4 \cdot 10^{-4}$

Therefore, the "diversity-generates-diversity" behaviour implies the quadratic scaling of the Heaps' variance (Fig. 6.1d), and also the "separation" of vocabularies of different phyla (Fig. 6.4). Following this line of reasoning, a question immediately arises: what is the evolutionary mechanism at the basis of such a behaviour? In this respect, here we briefly address two possible ideas, which could be analyzed and tested in a quantitative way in future works. The first considers the feedback between the environment and gene repertoire. Coming back to the speciation event mentioned above, if the first ancestor has a greater vocabulary of gene families, it can be able to colonize an environment requiring a high diversity. Then, it is reasonable to assume that, since this environment favours the vocabulary diversity, the organism is subject to selective pressure, which acts in a way to increase further its vocabulary size and the one of its descendants (the rich gets richer). In other words, species which can colonize more complex environments increases further their repertoire to fully exploit it. A second interesting explanation is about the mechanism of innovation. It is known that an important way to acquire new genes is duplicating existing ones and than mutating the obtained copies [100]. Let us then assume that, first, innovation events are obtained mainly by this mechanism, and, second, that each gene family can generate a finite small number of different new families. Within this scenario the "rich-gets-richer" behaviour can be recovered, indeed if there are very few families, the number of potential new families is low, implying that there are few possibilities to enlarge the vocabulary size. On the contrary, if the number of different families is large, the space of possible new families is much greater , favouring then the innovation events. Of course, a more exhaustive investigation is needed to better characterize the validity of the two assumptions, and a quantitative analysis is necessary to understand their consequences on the innovation dynamics. A possible objection to the first hypothesis is that the horizontal transfer is the dominant evolutionary force in prokariots [101], making negligible the contribute of innovation-by-duplication to the vocabulary growth. However it is biased for phylogenetic dinstance [102], implying that, probably, considering group of evolutionary similar organisms as "super-genomes", innovation-by-duplication becomes a relevant force.

6.5.2 Interpretation in linguistics

As said in the introduction, the quadratic fluctuations in linguistics can be explained by the topical aspect of texts [82]. We speculate that the "innovation reinforcement" mechanism proposed here is not in contradiction with "topicality", but instead it is an effective consequence of this. Our naive picture of how texts grow and generate Taylor's law is based on the fact each text is associated to a specific mixture of topics, and this composition determines the innovation rates. We expect that a text associated with a lot of different topics expresses a great variety of concepts, which need a large number of different words. This, in turns, implies that its innovation rate is larger with respect texts composed of few topics. As a consequence, if one compares two partial trajectories with different vocabulary size, the one with larger vocabulary should be associated to a greater number of topics an it will continue to grow faster. This effectively gives rise to the rich-gets-richer mechanism observed in Fig. 6.3. However, this is just an intuitive idea, and the scenario does not necessarily imply an innovation probability linearly dependent on the vocabulary size. Further investigations are needed to better quantify the relation between innovation dynamics and topicality.

Chapter 7

Towards the optimal ranking in ecological mutualistic networks

Authors: Andrea Mazzolini, Matteo Osella, Michele Caselle.

7.1 Introduction

Instead of focusing on emerging statistical laws, this final chapter aims to extract information from component systems exploiting its "topological structure". This has been inspired by recent works introduced in economy and ecology [103, 104], where looking at bipartite systems (specifically binary bipartite networks, equivalent to binary component systems, Section 2.1.2) they aim to identify the "most important" realizations and components. The meaning of realization-"importance" is based on two properties:

(1) The vocabulary of the realization, i.e. how many different components it contains. Looking at the system as a bipartite network, this property is the node/realization degree.

(2) The fact that the realization contains rare components (with low occurrence/degree), and those components tend to be present only in other "important" realizations.

Note that there is a sort of circularity in this definition. Indeed, looking at *(2)*, a realization is "important" if it contains components present in other "important" realizations. However, this can be elegantly translated into a recursive algorithm called *fitness-complexity map* [103], which, at the stationary state, provides a score for each realization related to the coupling of the properties above (this algorithm will be presented in Section

7.2). The original field of application has been economics, in particular the binary-bipartite system of countries (the realizations) and their exported products (the components). In this context, the "importance" of a country highlights its non-monetary competitiveness [103, 105, 106, 107]: in order to have a high score, the country must export a lot of different products (high vocabulary/degree, property *(1)*), and those products must be "complex" (they are exported only by few and important countries, property *(2)*).

Clearly the fitness-complexity map can be extended to all the binary component systems. Indeed, a second example [108] is a system of countries contributing to scientific topics. The element of the country-topic matrix is 1 if a nation provides a relevant contribution to that particular topic (that is quantified on basis of scientific paper citations). Here the map helps to evaluate the scientific competitiveness of a nation. A totally different kind of systems are mututalistic ecological networks, where a set of active species can interact with a second set of passive species. For example, considering interactions between plants and pollinators, the map ranks the animal pollinators according to their ecological importance within the ecosystem [104].

The present work starts from the fitness-complexity map employed by all the cited works, and defined in Section 7.2. We then propose a one-parameter generalization of the map (Section 7.3), which among all the possible generalizations shows useful symmetry properties. The free parameter controls the balance between the statement *(1)* and *(2)*, while in the standard-map it is fixed. Mutualistic ecological systems will be used as a benchmark to evaluate our proposal. Indeed, in such systems, a quantitative measure of the goodness of a ranking can be defined, the so-called *extinction area*. Taking advantage of this, we show that the generalization gives better results than the standard map in several cases (Section 7.4). The final section, 7.5, is dedicated to illustrate a curious geometric pattern shown by binary-component matrices ordered according to the algorithm scores. We prove that the origin of the pattern is intimately related to the *extinction area* maximization.

7.2 Fitness-complexity map in component systems

The *fitness-complexity* map was introduced in recent years with the purpose to quantify the non-monetary competitiveness of a country on the basis of its exported products [103, 106, 107]. The exported basked of countries can be represented as a binary-component system (Section 2.1.2), where countries (the system-realizations) are a collection of their exported products (the system-components). Specifically, the binary component matrix has elements n_{ij} equal to 1 if the country i exports the product j, and 0 otherwise. The fitness-complexity map takes as input this matrix, and defines a non-linear iterative process where two vectors of scores are updated until convergence. The first vector quantifies the *fitness* of countries, related to the economic strength of a nation. The second score is associated to products. Its called product-*complexity*, and encodes the fact that some goods provide more competitiveness than others. Starting form an initial condition of fitness $F_i^{(0)}$, for each country i, and complexity $Q_j^{(0)}$, for each product j (we will use vectors of ones), the values of these two observables evolve in time following the equations below (which encode the statements *(1)* and *(2)* described in the introduction):

$$\begin{cases} \tilde{F}_i^{(t)} = \sum_j n_{ij} Q_j^{(t-1)} & F_i^{(t)} = \tilde{F}_i^{(t)} / \langle \tilde{F}^{(t)} \rangle \\ \tilde{Q}_j^{(t)} = \left(\sum_i n_{ij} \left(F_i^{(t-1)} \right)^{-1} \right)^{-1} & Q_j^{(t)} = \tilde{Q}_j^{(t)} / \langle \tilde{Q}^{(t)} \rangle, \end{cases} \tag{7.1}$$

where, at each step of time, F and Q are updated according to the expressions on the left, and normalized through the formula on the right, so that their average is 1 at each step of time. Looking at the first equation, it is clear that the fitness of a country is proportional to the sum of the complexities of its exported products, i.e. producing a lot of products with high complexity will lead to a high fitness. At the same time, the product-complexity evolves in a way that if the product is made by very few countries with high fitness (few and small addenda at the denominator of the second line) its complexity will be high. On the contrary, if a lot of nations are able to make the product, including countries with low fitness, then its complexity is expected to be low. These rules couple fitness and complexity in a non-linear way, leading, after a sufficient number of iterations, to a stationary state which defines then final scores: F_i^*, Q_j^*. As discussed in the introduction, even thought this metric was introduced specifically in an economic context, it provides remarkable results also in other fields and, in particular, in ecology [104]. Specifically, in mutualistic ecological systems, where the "realizations" are active species while the "components" are passive species. A typical example is a system of plants (passive species/components) which interact with the animal pollinators (active species/realizations). Interactions between two species (one active i and one passive j) is represented with $n_{ij} = 1$ in the binary component matrix. Applying the fitness-complexity map to such a system, the fitness of an active species seems to be significantly related to its ecological importance within the ecosystem, while the complexity of a passive species to its vulnerability to system perturbations.

Since we will use this ecological system as a reference example, it is necessary to point out that "animal pollinator importance" and "plant vulnerability" can a be more meaningful nomenclature than "animal fitness" (which can be confused with the evolutionary meaning) and "plant complexity". However, at the same time, we want to maintain the standard mathematical notation F and Q, referring to the two vector of scores. Therefore, for the rest of the work, we will employ both the two nomenclatures. To avoid confusion it is sufficient to be aware that the term fitness is unrelated by its evolutionary-biology meaning, and the pairs complexity - vulnerability and fitness - importance are synonymous.

7.3 Map generalization

7.3.1 Definition and limit cases

We propose a one-parameter generalization of the fitness-complexity map 7.1 which reads as follows:

$$\begin{cases} \tilde{F}_i^{(t)} = \sum_j n_{ij} \left(Q_j^{(t-1)}\right)^{\gamma} & F_i^{(t)} = \tilde{F}_i^{(t)} / \langle \tilde{F}^{(t)} \rangle \\ \tilde{Q}_j^{(t)} = \left(\sum_i n_{ij} \left(F_i^{(t-1)}\right)^{-\gamma}\right)^{-1} & Q_j^{(t)} = \tilde{Q}_j^{(t)} / \langle \tilde{Q}^{(t)} \rangle, \end{cases} \tag{7.2}$$

where the new parameter is the exponent γ, and the standard map is recovered for $\gamma = 1$. Even if the map is well defined for each real value of γ, in the current work we will consider only the case the case $\gamma \geq 0$. This because in the opposite regime, $\gamma < 0$, the concept of fitness and complexity does not find a direct interpretation in the considered ecological systems.

In principle, one could generalize the map (7.1) in a lot of different ways, for example one possibility is discussed in [109]. Our proposal, aside from the useful properties which we will show in the next sections, seems to be a natural choice because of its equivalence to the following symmetric map:

$$\begin{cases} \tilde{F}_i^{(t)} = \sum_j n_{ij} \left(S_j^{(t-1)}\right)^{-\gamma} & F_i^{(t)} = \tilde{F}_i^{(t)} / \langle \tilde{F}^{(t)} \rangle \\ \tilde{S}_j^{(t)} = \sum_i n_{ij} \left(F_i^{(t-1)}\right)^{-\gamma} & S_j^{(t)} = \tilde{S}_j^{(t)} / \langle \tilde{S}^{(t)} \rangle \end{cases} \tag{7.3}$$

where we impose the substitution: $\tilde{S}_j = \tilde{Q}_j^{-1}$. Since the new observable S is the inverse of the complexity (for $\gamma = 1$), we can call it *simplicity*. Considering an ecological system, S is the opposite of the vulnerability of a plant species, therefore it is related to its ecological strength and importance within the system, becoming, somehow, the counterpart of the fitness for the passive species. In addition to aesthetic reasons, the symmetric shape of this map helps us to interpret its behaviour in limit cases. The first immediate observation is that $\gamma = 0$ is trivial: fitness and simplicity are no longer coupled, and after the first iteration they become proportional to the species degree (i.e. with how many other species it interacts). Therefore, considering values of γ much less than one, the map assigns more importance (i.e. a large fitness) at those active species which interact with a lot of passive ones, without considering their vulnerability. In other words, the property *(1)* of the introduction dominates over *(2)*.

In order to understand the opposite limit: $\gamma \to \infty$, let's consider the approximation for large γ of the equation 7.3, first line:

$$\tilde{F}_i^{(t)} \propto \left(\min_{j \in J(i)} S_j^{(t-1)} \right)^{-\gamma} \quad \langle \tilde{F}_i^{(t)} \rangle \propto \left(\min_{j} S_j^{(t-1)} \right)^{-\gamma}$$

where, given an active species (pollinator) i, $J(i)$ is the set of passive ones (plants) linked to i ($n_{ij} = 1$ if and only if $j \in J(i)$). The expression on the left means that the non normalized fitness of each pollinator, $\tilde{F}_i^{(t)}$ is determined only by the plant with minimum simplicity (i.e. maximal vulnerability/complexity) with which it interacts. Indeed, in this limit all the other addenda associated with *simpler* plants are negligible. Let us also consider the fitness average, on the right, which is proportional to the sum of all the non-normalized fitness. Since each $\tilde{F}_i^{(t)}$ is dominated by the less simple plant that contains, the sum over all the animals i is dominated by the less simple plant among all. As a consequence, since the actual value of fitness is the ratio between the terms above, only the pollinators linked to the less simple plant (the most vulnerable) have a positive normalized fitness, which instead goes to zero for all the other animals. For the map symmetry, this reasoning is true also for the plant simplicities $S_j^{(t)}$, which are dominated by the less fit animals. Of course we cannot have an intuitive comprehension of all the dynamics in this limit, and also we cannot make a prediction about which animals will have the greater or the lower fitness, however we can state that, in this regime, the fitness of an animal is determined only by the most vulnerable plants with which interacts (property *(2)* of the introduction) independently of how many links it has.

Therefore, we have a naive understanding of the generalized map behaviour: the variation of γ tunes the map of being in an intermediate state between the two limit cases. For large gammas the feature that dominates the fitness dynamics is the pollinator connection with the most complex plants, *(2)*. Decreasing γ, the plant complexities loose importance, and progressively the pollinator degree becomes the only relevant property, *(1)*. The dynamics is then completely dominated by the degree for $\gamma \ll 1$. The standard map with exponent $\gamma = 1$ seems to be an intermediate case between the two extremes, giving relevance both to the species degree and the connection with the most complex plants. Nevertheless, we do not know a priori which is the best choice of γ in determining the fitness rank in a given context, and sometimes the best results are obtained for γ different from 1, as we will see later.

7.3.2 Map convergence phenomenology at different exponents

Before testing the generalized map in ecological systems (next Section), it is worth analysing the map convergence properties. The map outputs are the stationary values of the trajectories $F_i^{(t)}$ and $Q_j^{(t)}$, which can show three different behaviours: (a) the convergence to a stationary positive fixed point, (b) the convergence to a fixed point which is zero, and (c) the absence of a fixed point, for example showing oscillations. Note that the divergence to infinite values is not allowed since the trajectories are normalized at each time step. The most informative scenario is obtained if all the trajectories converge to positive values, (a), allowing us to assign a real score to each species. This is not always possible since some trajectories often tend to zero, (b), implying that all those species have the same null score and cannot be compared. However, in these cases one can consider the species ranking, putting at higher positions those with higher fitness, while, if the trajectories go to zero, those with a "slower" decay will be ranked at higher position (a more detailed explanation of the ranking computation is illustrated in Appendix C.1). This is the typical scenario that one finds, and therefore the typical outcome of the procedure is a ranking of species. Finally, the case (c) neither allows us to to associate a score to the realizations/components, nor can be used to compute the species ranking, and therefore it does not provide any useful information. However, although there are no mathematical proofs that a fixed point always exists, all the studied empirical matrices for a wide range of the parameter γ, never show oscillations in the fitness/complexities trajectories. The convergence behaviour clearly depends on the input matrix, and, as far as we know, there are no possibilities to predict a priori how many (a) or (b) trajectories there will be. In [110] there is an exhaustive discussion about the relation between the input binary matrix and the map convergence outcome for the standard fitness-complexity map.

Moreover, the convergence behaviour depends on the chosen exponent γ, as shown the the figure 7.1, where we analyze the plant pollinator matrix introduced in Appendix A.4 (Robertson 1929). An informative observable describing the convergence scenario is the fraction of trajectories which tend to positive fixed points, f_c (i.e. the number of (a) cases over the total number of cases), which is displayed as a function of the map exponent in Figure 7.1a. We know that for exponents much less than one the animal fitness becomes proportional to the degree, implying that all the fixed points are positive, $f_c = 1$. This convergence scenario is conserved increasing the exponent from zero to values near 1, for instance at $\gamma = 0.75$ in figure 7.1b. Approaching 1, a small fraction of trajectories begins decaying to zero, in particular for $\gamma = 1$, panel 7.1c, about the 2 percent of the animals have a null fitness. Moving from 1 towards a "critical" exponent, the fraction of positive trajectories decreases, while the convergence time enlarges, as shown in 7.1d, for $\gamma = 1.1$. Finally, after a discontinuous transition, only few animals converge to positive fitness, panel 7.1e for $\gamma = 1.3$, while the majority have a null decaying fitness.

From a statistical mechanics perspective, the plot 7.1a resembles a first

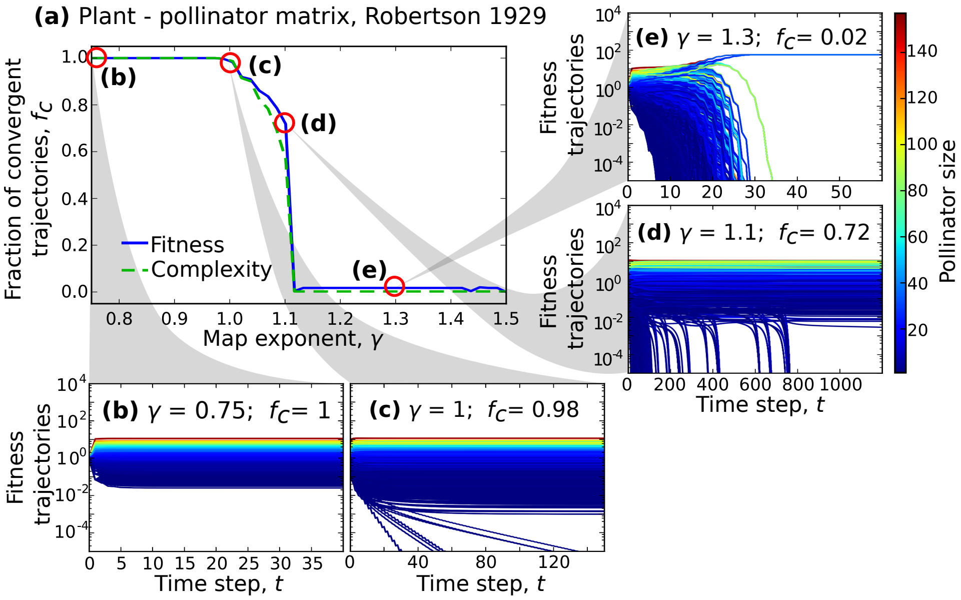


Figure 7.1: **Convergence phenomenology for fitness trajectories.** Panel (a) shows the fraction of trajectories which converge to positive stationary values as a function of the map exponent γ. The blue line refers to the pollinator fitness, while the green one to the flower complexity (data: plant-pollinator matrix Robertson 1929). The four red circles are associated to the fitness trajectories at specific exponents, shown in the other four plots. In particular, panel (b) is the outcome of the map at $\gamma = 0.75$, where all the trajectories converge to positive values. The standard map, $\gamma = 1$, is shown in panel (c), while plot (d) is near the critical exponent: the convergence time is much larger and an intermediate number of positive trajectories survive. Increasing then the exponent by an small quantity, the map enter the regime where only very few pollinator show positive fitness, as in panel (e).

order phase transition where the temperature is the map exponent and the order parameter is the fraction of trajectories which converge to positive values, f_c. This phenomenon is present across all the different empirical cases that we have considered (see Figure C.4 in Appendix). Indeed, in general, for exponents less than 1, all the trajectories converge to positive values, while, after the discontinuous transition, only very few trajectories do not drop to zero dominating all the other species. The critical value of gamma is dataset specific, and depends on a lot of different properties of the input binary matrix, including the matrix size, the ratio between height and width, the density of ones, and the degree distribution. As a consequence, the identification of general common trends is really complex. Analytic results can be derived only for very simple input matrices, for example, large uniform random matrices, as discussed in Appendix C.3. Within this very simple setting, we computed a condition for the trajectory convergence, Eq. (C.2), which provides an estimate of how the critical exponent depends on the matrix parameters (the size, and the density of ones), tested in Fig. C.2.

Interestingly, in many ecological cases the transition is exactly at the value of the standard map, $\gamma = 1$, and in general the critical exponent is always localized not far away from 1 (Fig, C.4). One can speculate that around the phase transition the algorithm could work better, since it is more sensitive to the non-trivial long-range correlation of the system. If this naive statement is assumed true, one can notice that the standard map (with $\gamma = 1$) gives interesting results because it is put exactly near this critical transition.

7.4 Looking for the specie ranking which maximizes the extinction area

As discussed in [104], the standard fitness-complexity map ($\gamma = 1$) provides a really informative ranking of species based on their importance within the ecosystem. The authors evaluated the map ranking using an observable called extinction area [111], whose computation through a toy example is shown in the figure 7.2. Basically, given an animal pollinator ranking, let's say (A_1, A_2, A_3), at the first step the first animal, A_1, and all its links are removed from the system, and, as a consequence, a certain number of plants remain without links and get extinct, like F_1 in the panel 7.2b. Removing all the animals according to the ranking and counting the extinct plants at each step, one can draw the plots 7.2c, which keeps track of the fraction of extinct plants as a function of the fraction of removed animals. The extinction area is defined as the integral of this curve, and quantifies, somehow, the velocity of the whole ecosystem extinction given a certain ranking of species importance. The fitness-complexity map provides a pollinator ordering which outperforms other algorithms used in the ecological context in terms of extinction area maximization [104].

The figure 7.2d shows that the generalized map (7.2) performs even better than the standard map. Indeed, varying the exponent γ the map mechanism changes, leading to different rankings, and, therefore different extinction areas. In particular, in the mutualistic network Robertson 1929 the extinction area as a function of the map exponent has a maximum at $\gamma \simeq 1.12$, where the area is significantly larger than the value computed with the standard map (i.e. at $\gamma = 1$). In general, there are no theoretical argument which allows us to identify the value of γ which maximizes the extinction area, and very often one have to explore a certain range of γ to find it. As a general remark from the analysed cases, the area maximum

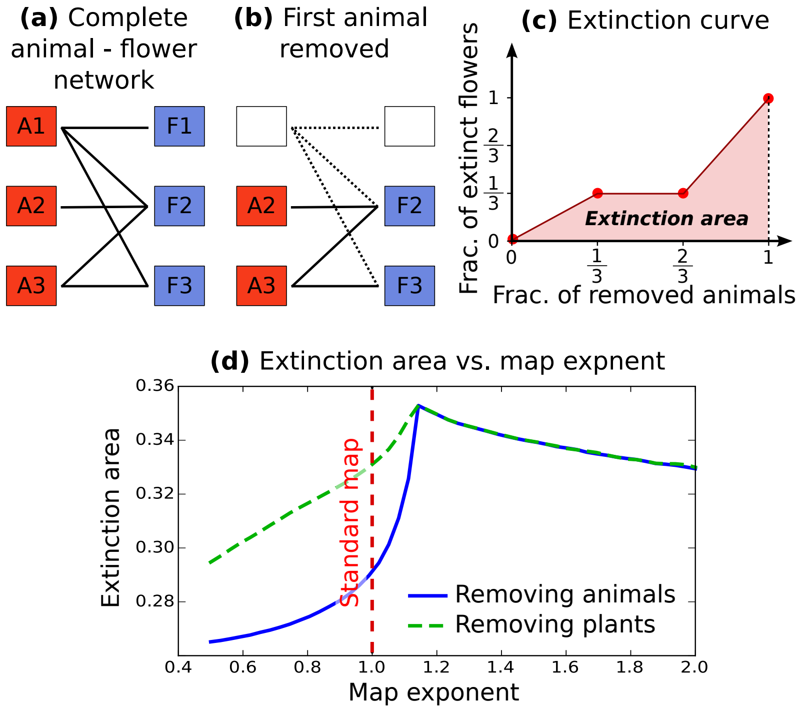


Figure 7.2: **Extinction area maximization.** The panel (a), (b) and (c) illustrate a toy example of the extinction area computation. Starting from the complete network, (a), animals are removed according to a certain ranking. In (b) the animal A_1 (the first of the ranking) and its links have been deleted, leading to the extinction of the flower F_1 which remains without links. The extinction area is the integral of the curve in (c), where at each animal removal step (x-axis) the fraction of extinct flowers is computed (y-axis). The panel (d) shows the extinction area for different map exponents γ of the mutualistic system Robertson 1929. The blue continuous line is the area computed in the same way of the toy example above, where the animal ranking is provided by the fitness at the given γ. The green dash-dot line is computed using the opposite procedure: the plants are removed following the complexity ranking keeping track of the extinct animals. The red dashed line highlights the standard map exponent which provides a smaller extinction area.

is typically greater or equal than 1 (see some examples in Figure C.5). We also compared the map performance with the generalized map proposed in [109]. Appendix C.2 shows that the algorithm (7.2) always finds a larger or equal extinction area than [109].

One can wonder if the obtained ranking is really the optimal one, or, in other words, whether it is possible to find a ranking not accessible to the generalized fitness-complexity map leading to better values of extinction areas. We approached this problem through the genetic algorithm described in Appendix C.4. Even though we explored only a limited class of matrices (having small dimensions), the best extinction area of the generalized map always equals the best outcome of the genetic algorithm (Fig. C.3a). At the same time, the map outperforms the genetic algorithm in terms of computational time (Fig. C.3b).

7.5 Matrix packing

Looking at the adjacency matrix shape after the ordering of the rows and the columns according to the map ranking, a geometric pattern emerges. In Figure 7.3 is shown a mutualistic network matrix, where the ones are represented with little black boxes, and the zero entries are in white. Rows and columns are ordered differently, specifically using the ranking generated by the generalized map with the indicated exponent γ. At first sight, it's quite surprising that the matrix assumes a curious configuration for the ranking which maximizes the extinction area. In particular, it seems that the area of contiguous zeros from the bottom right corner is maximized, and it is separated by the top left area by a continuous border of ones.

Actually, the extinction area and the emergent area of contiguous zeros are mathematically related as it is shown below. Therefore this emergent pattern and the extinction area maximization are two sides of the same coin. In order to prove this equivalence, let us define l_j as the row-index of the last non-zero column j element. For example, in the figure 7.4a, $l_1 = l_4 = 2$, $l_2 = 4$, and $l_3 = 1$. Then the total number of consecutive zeros from the bottom of each column is:

$$A^{(c)} = \sum_{j=1}^{R}(N - l_j) \tag{7.4}$$

where the sum is over all the R column and N is the number of rows. Note that we have used the superscript (c) to distinguish between the area from the matrix bottom, and the area for the matrix right, $A^{(r)}$, i.e. the number of consecutive zeros form the right side of each row. Clearly, $A^{(r)}$ is exactly $A^{(c)}$ of the transposed matrix. We want to establish a relation between these quantities and the extinction areas.

Let us consider the first step to compute the extinction area: the removal of the first pollinator in the ranking. The plants which remain without links define the fraction of extinct plants/columns during the first iteration: $e_1^{(c)}$. Considering the binary matrix, this procedure is equivalent to remove the first row (the first pollinator), and to count the columns composed only of zeros (the plants remaining without links). In the example, after the removal of the first row, only the third column remains without ones, and therefore

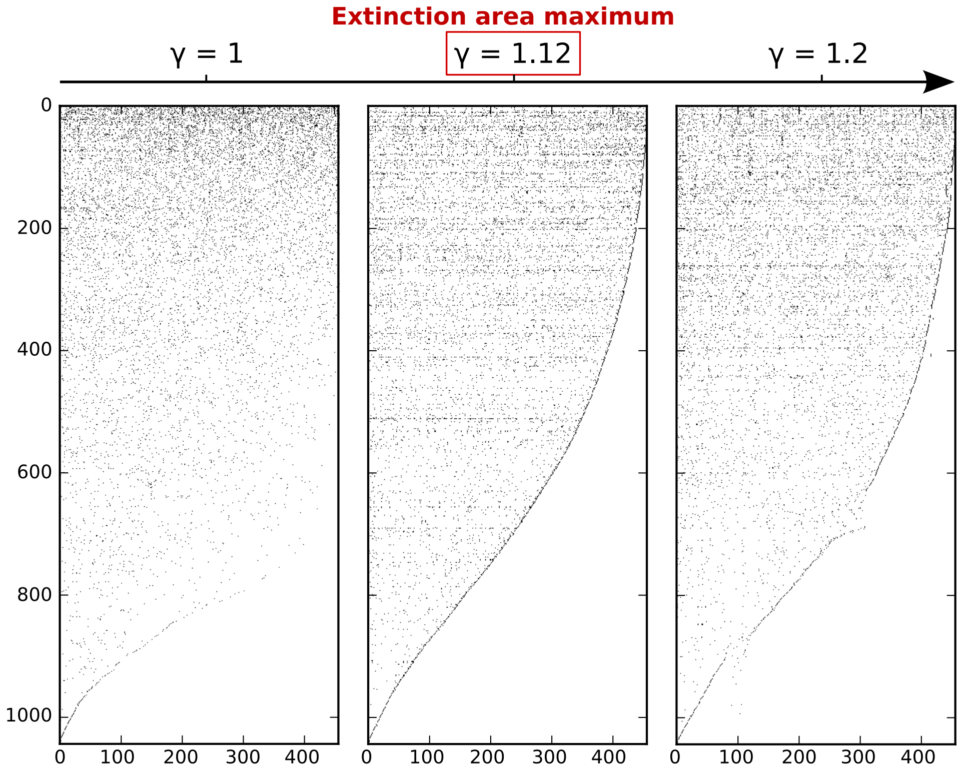


Figure 7.3: **The extinction area maximum provides the best matrix packing.** Interaction matrix matrix with rows and columns sorted according to the F and Q ranking for three different map exponents (data: the plant-pollinator system Robertson 1929). A black dot corresponds to the entry 1, while a white one in an entry 0. $\gamma \simeq 1.12$ is the value which maximizes the extinction area as shown in the figure 7.2d. The maximum extinction area corresponds to the visually best matrix "packing", in the sense that the area of consecutive zeros from the bottom right corner is maximized.

the associated plant gets extinct. In general, the fraction of extinct plant after the first removal step reads as follows:

$$e_1^{(c)} = \frac{1}{R}\sum_{j=1}^{R} \delta_{l_j,1}$$

where the Kronecker delta is equal to 1 if $l_j = 1$, which means that the summation counts the columns which are all zeros except for the first removed element. Iterating this procedure, the fraction of extinct plants at the k-th removal step is:

$$e_k^{(c)} = e_{k-1}^{(c)} + \frac{1}{R}\sum_{j=1}^{R} \delta_{l_j,k} = \frac{1}{R}\sum_{j=1}^{R} \theta(k - l_j) \tag{7.5}$$

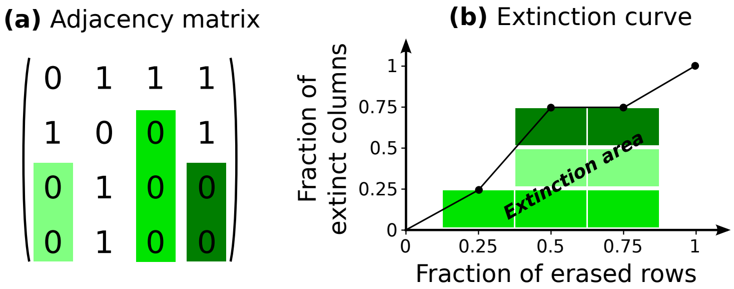


Figure 7.4: **Equivalence between the area of zeros form the matrix bottom, (a), and the extinction area computed removing rows, (b).** If the first row is removed, the third column gets extinct, and therefore the extinction curve increases by $\frac{1}{4}$. Note that this contribution to the curve is represented by a box with the same shade of green of the extinct column. The removal of the second row leads to the extinction of the first and the fourth columns, therefore the extinct plant are now 3 as the boxes under the second point in (b). The same happens for the third row removal. Looking at the panel (b), each column contributes to the extinction area with a number of boxes equals to its number of consecutive zeros from the bottom, and this implies the proportionality between the two areas.

which is the fraction at the previous step, plus the new extinct columns. The expression on the right is obtained writing down the explicit expression of e_{k-1} and all the other terms for previous removal steps. Here the theta function is 1 only if the last non-zero element is lesser or equal than the number of removed columns k. The sequence of e_k from $k = 1$ to N defines the extinction curve, implying that the extinction area reads:

$$E^{(c)} = \frac{1}{N}\sum_{k=1}^{N} e_k^{(c)} = \frac{1}{NR}\sum_{j=1}^{R}(N - l_j) = \frac{A^{(c)}}{NR} \tag{7.6}$$

where we have used the expression 7.5 and the fact that $\sum_i^N \theta(i - l_j) = N - l_j$, proving the equivalence between the extinction area and the area of contiguous zeros. An intuitive derivation of this relation is shown in the caption of the figure 7.4. It is straightforward to derive the same equivalence between the extinction area removing the columns, $E^{(r)}$, and the area of consecutive zeros from the right side of the matrix, $A^{(r)}$.

To summarize, we proved that a ranking of the rows defines a certain area of contiguous zeros from the matrix bottom, which is proportional to the extinction area removing rows. At the same time, the ranking of the columns defines the area of zeros from the right side of the matrix, equivalent to the extinction area removing columns. These two statements prove that the extinction area and the area of contiguous zeros are connected, but the origin of the actual shape in Fig 7.3 is not fully understood yet. Specifically, it is not clear why the two areas of contiguous zeros are equal and bounded by a continuous monotonic border of ones. This can be a consequence of the simultaneous maximization of $A^{(r)}$ and $A^{(c)}$ provided by the generalized fitness-complexity algorithm, but at the moment a demonstration is lacking, and the problem still under study.

7.6 Discussion

In conclusion, this work proposes and analyses a generalization of the fitness-complexity algorithm, Eq. (7.2), which, among several possible generalizations, can be expressed in a useful symmetric formula Eq. (7.3). As for the classical map, it leads to a ranking of realizations and components according to a certain definition of "importance". Intuitively, in our proposal, the concept of realization-importance is a balance between two ideas: *(a)* the degree of the realization, and *(b)* the connection with "complex" components. The free parameter γ determines the weight of these two properties. In particular for $\gamma = 0$ the importance (or fitness) is defined by *(a)*, while for large γ, only *(b)* becomes relevant. The standard map is recovered for $\gamma = 1$, providing then a fixed definition of "importance". Mutualistic ecological systems are a perfect benchmark to test the ranking of species. Indeed, each order can be quantitatively evaluated computing the extinction area (Section 7.4). The analysis highlights that the generalization typically finds better extinction areas than the standard map (with an optimal γ dependent on the dataset). Moreover, for a limited class of small matrices, Appendix C.4 shows that a genetic algorithm cannot find better areas than the generalized map, suggesting that our proposal computes a solution very close to the optimal one in a really short time (with respect the genetic algorithm for example).

An interesting observation is that the fraction of convergent trajectories to positive values shows a discontinuous transition varying γ, Fig 7.1 and C.4 (resembling a first order phase transition). One can argue that the algorithm tuned around the transition better catches the non-trivial correlations between components, providing therefore better results. This seems to be in agreement with the extinction area maximization. Indeed, in almost all the considered cases, the gamma maximizing the extinction area is very close to the critical γ (see Figure C.4 and C.5 in Appendix). Among the different examples shown in Figure C.4, panel (e) display the country-product matrix studied in [103]. The fact that the discontinuous transition is at $\gamma \approx 1.2$ suggests that the generalized map could provide better results also in economic contexts (where all the analysis have been performed with the standard map).

A final remark regards the surprising geometric shape in Figure 7.3, which we showed to be connected to the extinction area maximization. The authors of [104] already notice that ordering rows and columns according to the standard fitness-complexity map enhance the “nested” structure of the system. Our generalization provide even better matrix packing, as shown in Figure 7.3, where the best configuration is for $\gamma \approx 1.2$. This implies that the generalized map can have implications in better defining the concept of nestedness, since, despite its popularity, there are still ambiguities in its definition [112].

Appendix A

Datasets

A.1 Genomics

The genomic dataset considers genomes composed of families of protein domains. Protein domains are the basic modular topologies of proteins [113], which can be considered independently folded and thermodynamically stable. These domains can be grouped into families according to functional and evolutionary similarities. Different domains of the same family can be found in each genome in the same or different proteins.

A.1.1 Superfamily classification

We used the classification of protein domains into families from the SUPERFAMILY database [63] considering a set of $R = 1061$ prokaryotic genomes ("realizations") and a total number of different families $N = 1531$ ("components"). As a functional annotation of protein domains in SUPERFAMILY, we considered the SCOP annotations mapped into 7 general function categories, as developed by C. Vogel [114].

A.1.2 Comparison between different classifications

In order to compare different family classifications we employed the three classifications provided by the Superfamily database, which are "fold", "superfamily" (the main classification described above), and "family". The "fold" classification is the widest, in the sense that the families are defined with weaker evolutionary and functional constraints. As a consequence, the number of protein domains in a family is typically larger and the number of different families, N, is less than the other two classifications. The "family" classification is the less wide.

Together with the Superfamily database we also employed the PFAMILY database [115, 116], and its two different classifications: CLAN (the wider) and PFAM. It is important to note that the list of genomes considered in the PFAMILY and SUPERFAMILY database is slightly different. However the size distribution of the genomes is similar, allowing us to compare the two databases in Section 3.4.3.

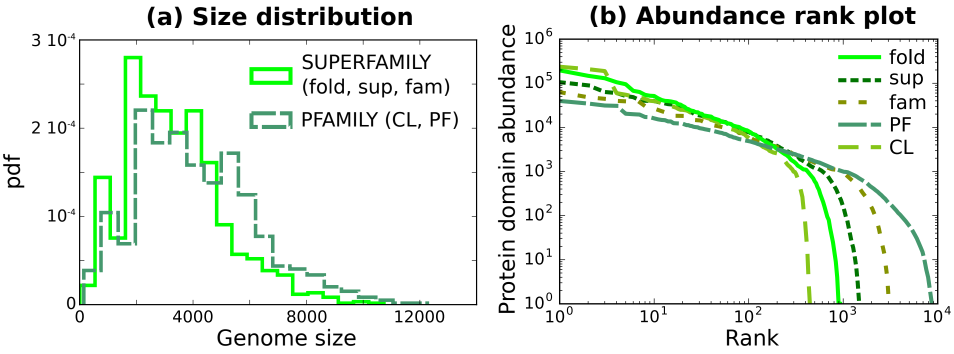


Figure A.1: **Genome size distribution and abundance rank plot for the SUPERFAMILY and PFAM databases.**

A.1.3 Weighted under-sampling for Heaps' law trajectories

Here we describe a procedure to generate an artificial vocabulary-size trajectory for a single genome (i.e. a single realization Heaps' law). The vocabulary-size trajectory needs an ordering of the components within the realization. We naively generate this order assuming that a genome "grows" through an under-sampling (without replacement) of its own protein domain families. In such a way, the components with higher abundance will be selected in the first positions, and at the end of the sampling the genome j will have the original abundance list $\{n_{ij}\}$. We also introduce an second important ingredient: if a protein domain is not drawn yet in the sampling procedure, its extraction probability is determined not only by its abundance n_{ij}, but also by its occurrence in the ensemble. For example, the first instance of a core-component (with high occurrence, and probably very important for the organism survival), has a higher probability of being extracted than a specialized component.

According to those assumptions, we want to generate the ordered string of components of the genome j: $(x_1, \ldots, x_{s_j})$, where $x_k \in \{c_i\}$, with $i = 1, \ldots, N$. The component abundance (inside the genome) is given by $\{n_{ij}\}$, and the global occurrence by $\{o_i\}$. At each iteration t of the algorithm, a new component x_t is added to the sequence $(x_1, \ldots, x_{t-1})$, and it is selected among $\{c_i\}$ with the following probability:

$$P(x_t = c_i) \propto \begin{cases} n'_{ij} + \omega o_i & \text{if } c_i \notin (x_1, \ldots, x_{t-1}) \\ n'_{ij} & \text{otherwise} \end{cases} \tag{A.1}$$

which is normalized at every step dividing by $\sum_i P(x_t = c_i)$. n'_{ij} is the remaining number of the component i in the genome j. At the first step, the list $\{n'_{ij}\}$ is set equal to the abundances $\{n_{ij}\}$. At each next step, after the extraction of the component c_i, the number of remaining component is updated as: $n'_{ij} = n'_{ij} - 1$, in such a way, at the end of the procedure $n'_{ij} = 0$, $\forall i$. Note that the algorithm depends on one parameter ω which determine the weight of the occurrence in the first extraction of a component. For example, if $\omega = 0$ the algorithm is just a uniform under-sampling without replacement, while if $\omega \gg 1$ the first extraction of a component is determined only by its occurrence.

The aim of this procedure is to create an ensemble of vocabulary-size trajectories which allows us to have enough statistics to compute the Heaps' law variance as a function of the average (Figure 6.1d). Indeed, the same function computed with the vocabulary-size scatter plot (one dot for each genome) shows a very noisy trend, Figure A.2a (even if the super-linear scaling is still recognizable). Of course, one has to fix the parameter ω. To this end, we assumed that the Heaps' law defined as an ensemble of trajectories must reproduce the statistics of the vocabulary-size scatter plot. We then have chosen ω to minimize the distance between the binned average of h (computed with the scatter plot, red crosses in Fig. A.2b) and average of the ensemble of trajectories. This procedure leads to the optimal value $\omega = 0.91$. The Figure A.2b shows some examples of average of trajectories for different values of ω.

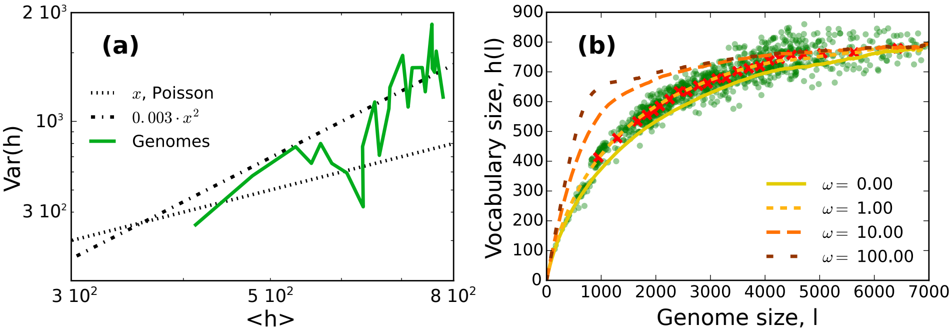


Figure A.2: Panel (a): Heaps' law variance of the genome dataset computed considering the vocabulary-size scatter plot. This observable is discussed in Chapter 6.2. Panel (b): the Heaps' law and its binned average (red crosses) are compared with the average of four ensembles of trajectories computed by the under-sampling procedure described in this section. The four ensembles correspond to four choices of ω. $\omega = 0$ is a pure under-sampling algorithm.

A.2 Linguistics

A.2.1 Chapters from Gutenberg project

A first linguistic corpus is composed by $R = 1721$ book chapters of several English books randomly chosen from the most popular ones in the database “http://www.gutenberg.org”. In this first linguistic dataset, we defined chapters as realizations, instead of entire books, to obtain a corpus with a range of sizes (total number of components per realization, shown in Figure A.3) comparable to the one of genomes (Figure A.1, SUPERFAMILY) and LEGO toys (Figure A.5). The complete list of books considered is reported in Table A.1. The elementary components are defined as the words according to the following rules: the words are separated by whitespace or non-alphanumeric or non-underscore character, all the capital letters are converted into lower-case letters, all the characters different from ascii lower-case chars are removed (such as numbers or the punctuation).

Table A.1: **List of the books whose chapters compose the analysed linguistic corpus.**

Title	**Author**
Alice’s adventures in wonderland	Lewis Carroll
Anna Karenina	Lev Nikolayevich Tolstoy
A tale of two cities	Charles Dickens
Dracula	Bram Stoker
Emma	Jane Austen
Great expectations	Charles Dickens
Les miserables	Victor Hugo
Moby Dick	Herman Melville
Notre-Dame de Paris	Victor Hugo
Pride and prejudice	Jane Austen
The adventures of Tom Sawyer	Mark Twain
The count of Monte Cristo	Alexandre Dumas
The man in the iron mask	Alexandre Dumas
The picture of Dorian Gray	Oscar Wilde
The three musketeers	Alexandre Dumas
War and peace	Lev Nikolayevich Tolstoy

A.2.2 Books from Gutenberg project

This dataset is composed of 3036 books taken from the same database above (http://www.gutenberg.org). We used the parsed dataset provided by [117], where the meta-data, the licence information and the transcriber’s notes have been removed. In order to generate the component-system-realization, and therefore to transform the book in a “list of words” we applied the rules described above for the chapters.

A.2.3 Wikipedia

The analysed set of English Wikipedia articles can be freely downloaded from the Wikipedia dumps: “https://dumps.wikimedia.org/enwiki/”. In order to parse the raw articles we used the software provided at the following url: “http://attardi.github.io/wikiextractor/”. Then, the cleaned articles were parsed with the same rules described for the book chapters.

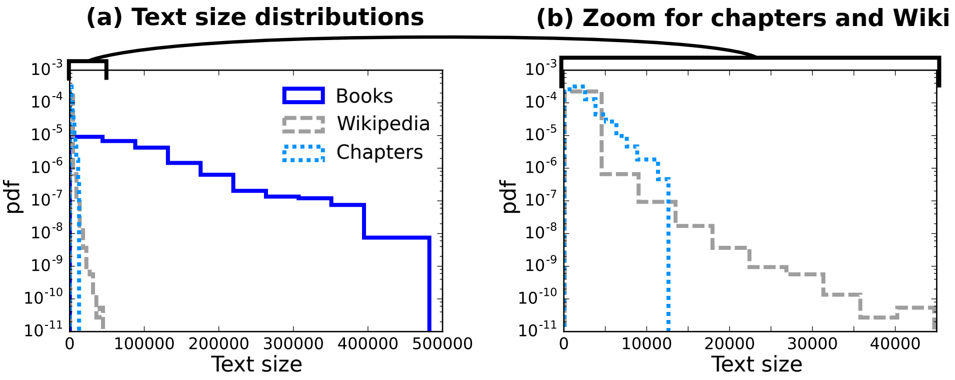


Figure A.3: **Text size distribution for the linguistic databases.**

A.2.4 Single-realization and trajectory definitions of the Heaps’ law in linguistics

As discussed in Section 2.2.2 and 6.2, the global Heaps’ law of a component system can be defined as the scatter-plot of sizes and vocabularies, or as the ensemble of the single realization-trajectories. Figure A.4 compares the first momentum, panel (a), and the variance, panel (b), of the vocabulary size statistics h of the two definitions. The two panels show a good accord, verifying the equivalence of the two prescriptions. For the comparison has been used the Gutenberg dataset composed of 3036 books. Of course, one can compare the two definition only when the component-order is well-defined, as in the considered linguistic ensemble.

A.3 LEGO

The composition in bricks of several LEGO sets ($R = 2820$) can be freely downloaded from “http://rebrickable.com”. We excluded from the analysis

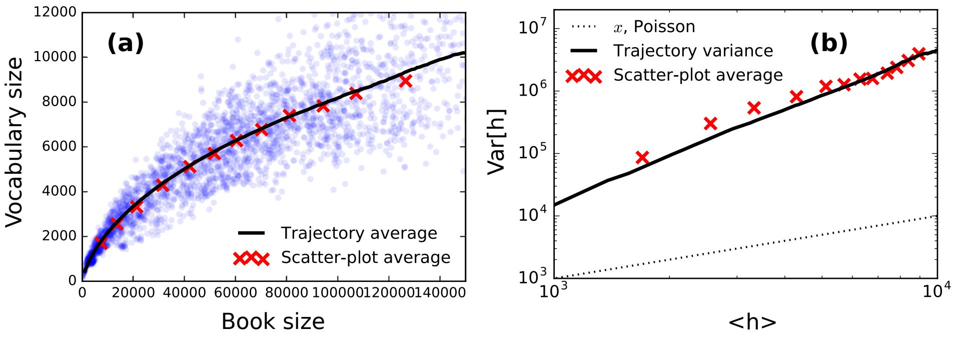


Figure A.4: **Heaps' law average and variance for its two definitions.** Panel (a) compares the binned average of the scatter plot definition, red crosses, and the average of the trajectory ensemble, black line. Similarly, panel (b) displays the variance. The considered dataset is the ensemble of 3036 books from the Gutenberg database.

LEGO sets belonging to the category of "LEGO Technic" since, by construction, they share a very small number of bricks with the classic LEGO toys. Similarly, we did not consider LEGO sets with less than 80 components or belonging to the categories "Educational and Dacta" and "Supplemental" in order to exclude sets that are actually collections of spare parts or additional bricks for other sets.

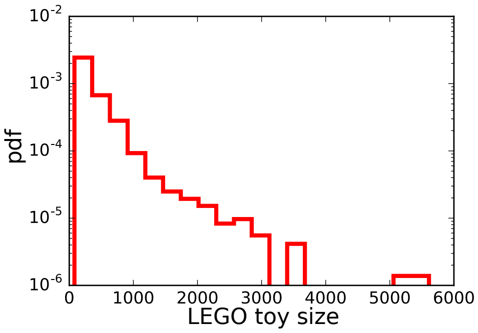


Figure A.5: **LEGO toy size distribution.**

A.4 Mutualistic ecological systems

Ecological interaction matrices are taken from the Interaction Web Database, IWDB (https://www.nceas.ucsb.edu/interactionweb/), an open access database which shares a large number published data on species interaction networks. In particular, the main ecological example discussed in Chapter 7 is an interaction plant-pollinator matrix based on the work of C. Robertson in the

1929, which studied the interactions between flowers and insects in Carlinville, Illinoise. This dataset was validated and updated in [118], collecting the data of 1429 animal species visiting flowers of 456 plant species.

Appendix B

Calculations

B.1 Core size form exponential rank distribution

The mathematical calculation described in the section 3.3 of the main text can be applied to an exponential rank distribution of the form

$$f_i = \frac{1}{\beta} e^{-\lambda i}, \qquad \beta = \sum_{i=1}^{N} e^{-\lambda i}. \tag{B.1}$$

Considering a random sampling of R realizations with fixed size s, one finds:

$$p(o) = \frac{(1-o)^{\frac{1}{s}-1}}{\lambda s N \left(1 - (1-o)^{\frac{1}{s}}\right)}. \tag{B.2}$$

Imposing the condition $s \gg 1$ this equation takes the form

$$p(o) \simeq \frac{(1-o)^{-1}}{N\lambda \log\left[(1-o)^{-1}\right]}, \tag{B.3}$$

which provides a good approximation for the overall distribution shape as a function of one single effective parameter $k = N\lambda$.

In the $s \gg 1$ limit, the occurrence extreme values are $o_1 \simeq 1$ and $o_N \simeq 0$. This implies that the distribution is well defined over all possible values of occurrence. Figure B.1 shows the rescaling properties of Eq. (B.3) by testing its independence on s (panel a) and by varying N and λ while keeping their product constant (panel b).

For rare families, one can further approximate the expression for $p(o)$ finding the expected power-law decay with exponent -1:

$$p(o) \simeq \frac{1}{N\lambda} o^{-1}. \tag{B.4}$$

We now analyse the properties of the fraction of core components, i.e., those with occurrence greater than the threshold θ_c. In order to derive the core size one has to integrate the distribution described by Eq. (B.2) from $o = \theta_c$ to the maximum occurrence value o_1 (whose formula can be obtained from Eq. (3.2) of the main text).

The result reads:

$$\begin{cases} c = 1 & \text{if} \quad o_N \geq \theta_c \\ c = -\frac{1}{N\lambda}\left(\lambda + \log\beta + \log\left[1 - (1-\theta_c)^{\frac{1}{s}}\right]\right) & \text{otherwise.} \end{cases} \tag{B.5}$$

In the limit of large s this expression becomes

$$c \simeq -\frac{1}{N\lambda}\left(\lambda + \log\beta + \log\left[\log(1-\theta_c)^{-1}\right] - \log s\right), \tag{B.6}$$

which further simplifies only when the logarithm of s becomes dominant over the other terms.

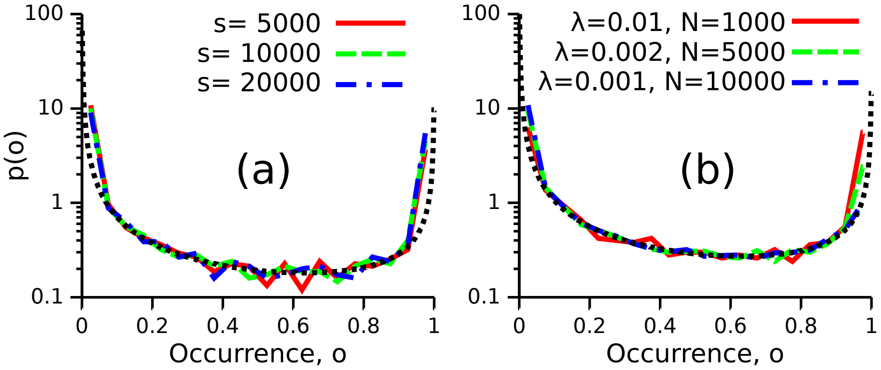


Figure B.1: **Rescaling property of the occurrence distribution generated by an exponential frequency rank distribution.** a) The global shape of the distribution does not depend on s in the limit $s \gg 1$, as shown by the Eq. (B.3). The three curves are computed at fixed values of $\lambda = 0.01$ and $N = 1500$, and the black dotted line is the prediction of Eq. (B.3). The only effective parameter determining the U-shape (for $s \gg 1$) is the product λN. Indeed, panel b shows that the distribution does not change its shape while varying N and λ if their product is kept to a constant.

It is worth mentioning that the expression above does not show rescaling properties, even in the regime $s \gg 1$, and this may seem to be in contradiction with Eq. (B.3). Nevertheless, this apparent inconsistency is basically due to the singular behaviour of the occurrence distribution in $o \simeq 1$. In the large s limit, the right boundary can be expressed as $o_1 = 1 - \epsilon$, where ϵ is an infinitesimal term depending on s and λ, whose effect on the overall distribution shape is negligible (Eq. (B.3)). However, the core size is defined as the integral of the distribution. Therefore, the variation of $p(o_1)$ due to a change in s or λ provides a sufficiently large contribution (because of the function singular behaviour) which compensates the infinitesimal variation of o_1. Finally, this leads to a finite contribution to the integral and thus to the core size as it is defined in the main text. In general, this finite contribution has a non-trivial dependency on the parameters, explaining why the equation (B.6) does not show the rescaling property.

B.2 Heaps' integral

Here we discuss a summation that typically appears during the Heaps' law derivation which assumes a random sampling of components from a power-law frequency distribution:

$$\mathcal{H}(s,a,c) = \sum_{i=a}^{N} \left(1 - ci^{-\gamma}\right)^{s}, \tag{B.7}$$

where s is the number of extracted components, γ is the Zipf's law exponent, c is the normalization coefficient of the frequency rank plot, and N is the vocabulary size. The parameter a defines the onset of the power law region. Here we want to approximate the summation above under the conditions $s \gg 1$ and $N \gg 1$.

First we express the summation as an integral:

$$\mathcal{H}(s,a,c) \approx \int_{a}^{N} \left(1 - ci^{-\gamma}\right)^{s} \, di + \epsilon,$$

where the error ϵ can be evaluated with the Euler-Maclaurin formula. In the present case, $s \gg 1$ and $N \gg 1$ lead to a negligible value of ϵ.

The next approximation concerns the integrand, and it is based on the observation that $ci^{-\gamma}$ is typically much less than one (since $\sum_{i=a}^{N} ci^{-\gamma} \leq 1$). This allows us to express the integrand as an exponential function:

$$\mathcal{H}(s,a,c) \approx \int_{a}^{N} \exp\left[-ci^{-\gamma}s\right] \, di.$$

Note that imposing $s \gg 1$, the approximation above holds true also for $ci^{-\gamma} \lesssim 1$ because the large exponent drives those terms to negligible values. In other words, the integral is dominated by the terms with $i \approx N$, for which the condition $ci^{-\gamma} \ll 1$ is always satisfied (we are considering $N \gg 1$).

The expression above can be evaluated with the change of variables: $x = ci^{-\gamma}s$, leading to:

$$\mathcal{H}(s,a,c) \approx \frac{(cs)^{1/\gamma}}{\gamma} \int_{cN^{-\gamma}s}^{ca^{-\gamma}s} e^{-x} x^{-1-1/\gamma} \, dx.$$

Using the definition of the incomplete Gamma function $\Gamma(n,t) = \int_t^\infty e^{-x}x^{n-1}dx$, the formula above can be written down as:

$$\mathcal{H}(s,a,c) \approx \frac{(cs)^{1/\gamma}}{\gamma}\left(\Gamma\left(-\frac{1}{\gamma}, cN^{-\gamma}s\right) - \Gamma\left(-\frac{1}{\gamma}, ca^{-\gamma}s\right)\right). \tag{B.8}$$

A further approximation can be applied if $ca^{-\gamma}s \gg 1$. In such a case, the second Gamma function is negligible:

$$\mathcal{H}(s,a,c) \approx \frac{(cs)^{1/\gamma}}{\gamma}\Gamma\left(-\frac{1}{\gamma}, cN^{-\gamma}s\right). \tag{B.9}$$

One can also try to evaluate the expression above in the limit cases. The key parameter is $cN^{-\gamma}s$, which defines when the realization size dominates over the vocabulary (saturation regime, $cN^{-\gamma}s \gg 1$), or the vocabulary can be considered infinite ($cN^{-\gamma}s \ll 1$). In the first case, it is immediate to see that:

$$\mathcal{H}(s,a,c) \approx 0 \qquad \text{if } cN^{-\gamma}s \gg 1. \tag{B.10}$$

The opposite limit (system far away from the saturation) can be evaluated considering the recurrence relation of the incomplete gamma function:

$$\Gamma(n+1,t) = n\Gamma(n,t) + t^n e^{-t}. \tag{B.11}$$

Which implies:

$$\mathcal{H}(s,a,c) \approx Ne^{-cN^{-\gamma}s} - (cs)^{1/\gamma}\Gamma\left(1-\frac{1}{\gamma}, cN^{-\gamma}s\right),$$

and imposing the limit (when $\gamma > 1$):

$$\mathcal{H}(s,a,c) \approx N - (cs)^{1/\gamma}\Gamma\left(1-\frac{1}{\gamma}\right) \qquad \text{if } cN^{-\gamma}s \ll 1. \tag{B.12}$$

where $\Gamma\left(1-\frac{1}{\gamma}\right)$ is the Euler Gamma function.

B.3 Heaps variance in the random sampling model

The random sampling model is defined in Section 3.2. It states that a realization of size s is generated through s independent extractions from the pool of components, where the extraction probabilities are fixed by the component frequencies $\{f_i\}$ ($i = 1, \ldots, N$, $\sum_i f_i = 1$). The Heaps' law counts the number of different components after the first l selected components. It can be derived considering the probability of extracting at least one time the component with frequency f_i after l extractions:

$$q_i(l) = 1 - (1-f_i)^l. \tag{B.13}$$

Then, the stochastic variable describing the Heaps' process is:

$$h(l) = \sum_{i=1}^{N} x[q_i(l)], \tag{B.14}$$

where $x[p]$ is a binary random variable which is 1 with probability p and 0 with probability $1 - p$. Its first two statistical moments are:

$$\langle x[p] \rangle = p, \qquad \langle x[p]^2 \rangle = p.$$

Given (B.14), the average of the vocabulary size is:

$$\langle h(l) \rangle = \sum_{i=1}^{N} \langle x[q_i(l)] \rangle = \sum_{i=1}^{N} q_i(l),$$

while the second moment reads:

$$\langle h(l)^2 \rangle = \sum_{i=1}^{N} q_i(l) \left(1 + \sum_{j=1, j \neq i}^{N} q_j(l) \right),$$

Finally, putting together the first and the second moment, one can easily derive the variance:

$$\mathrm{Var}[h(l)] = \sum_{i=1}^{N} q_i(l) \left(1 - q_i(l)\right). \tag{B.15}$$

B.4 Chinese Restaurant process: number of tables statistics

B.4.1 CRP notation

The Chinese Restaurant Process, CRP, is an innovation-duplication growth model for component systems. It can be visualized as a restaurant sitting plan, according to which:

- At the initial time one person is placed at one table.
- At each further time, s, a new person enter the restaurant and takes a seat at a new table with probability p_{new}, or at an occupied table with probability p_{old} ($p_{new} + p_{old} = 1$).

By this definition and since deletion events are not considered (e.g. a person leaves the restaurant), the time step, s, corresponds to the number of person in the restaurant at that time. The state of the process at time s is completely defined by the set of table sizes: $\{n_i\}$, where n_i is the number of person at the table i, $i = 1, \dots h$, and $\sum_{i=1}^{h} n_i = s$. Note that the number of table, h, is a random time-dependent variable.

The two-parameters CRP is defined by the following probabilities [95]:

$$p_{new} = \frac{\theta + \alpha h}{\theta + s} \qquad p_{old}^{(i)} = \frac{n_i - \alpha}{\theta + s}, \tag{B.16}$$

where $0 \leq \alpha \geq 1$ and $\theta > -\alpha$. It is immediate to verify that $p_{old} = \sum_{i=1}^{h} p_{old}^{(i)}$ satisfies $p_{old} + p_{new} = 1$.

This kind of model with s steps clearly generates a component entity of size s, where the components are the tables, each of them with a certain abundance (i.e. the number of persons at that table). Note that the probability of a table growth is proportional to the table size, n_i, leading to a preferential attachment mechanism generating a power law component abundance distribution, typical feature of complex component systems. In this section, however, we are going to describe the Heaps' law generated by such process, which is the characterization of the random variable $h(s)$, i.e. the number of different tables/components at time/size s. In particular we will derive the exact and the asymptotic value of its average and its variance.

B.4.2 Implicit expression of the moment generative function

The probability of having h different tables at time s, $P(h, s)$, is governed by the following recurrence relation:

$$\begin{cases} P(h, s+1) = \dfrac{\theta + \alpha(h-1)}{\theta + s} P(h-1, s) + \dfrac{s - \alpha h}{\theta + s} P(h, s) \\ P(h, 1) = \delta_{h,1} \end{cases}, \tag{B.17}$$

where $\delta_{h,1}$ is the Kronecker delta. From this equation one can derive the recurrence relation for the moment generative function, $G(z, s) = \sum_{h=1}^{\infty} P(h, s) e^{hz}$, summing both the equation terms over h and multiplying for e^{hz}.

$$\begin{cases} (s + \theta) G(z, s+1) = \left(\theta e^z + s\right) G(z, s) + \alpha \left(e^z - 1\right) \dfrac{\partial G(z, s)}{\partial z} \\ G(z, 1) = e^z \end{cases}. \tag{B.18}$$

We now define the operator $\mathcal{L}_s$ groping together to all the terms acting on $G(z, s)$ on the right side of the equation above:

$$\mathcal{L}_s = s + \theta e^z + \alpha \left(e^z - 1\right) \frac{\partial}{\partial z}, \tag{B.19}$$

so that the recurrence equation becomes: $(\theta + s) G(z, s) = \mathcal{L}_{s-1} G(z, s-1)$. This allows us to write down an implicit expression for the generative function iterating the relation:

$$G(z, s) = \frac{1}{(\theta + 1)_{s-1}} \mathcal{L}_{s-1} \mathcal{L}_{s-2} \dots \mathcal{L}_1 e^z, \tag{B.20}$$

where the notation $(\theta+1)_{s-1}$ refers to the rising factorial:

$$(x)_N = x(x+1)\dots(x+N-1) = \frac{\Gamma(x+N)}{\Gamma(x)}. \tag{B.21}$$

Here we are interested in the first and second moment of $P(h,s)$, which can be obtained from the generative function by the formula: $E[h(s)^k] = \partial_z^k G(z,s)|_{z=0}$. In order to do this, we will not compute the explicit expression of $G(z,s)$ (which seems to be a very tough task), instead we will apply the formula directly to the implicit form of the generative function, (B.20), taking advantage of the following property of the operator $\mathcal{L}_s$ when $z=0$:

$$\begin{aligned}\mathcal{L}_s f(z)|_{z=0} =& (s+\theta e^z)\, f(z)|_{z=0} + \alpha\,(1-e^z)\,\frac{\partial}{\partial z} f(z)|_{z=0} = \\ =& (s+\theta) f(0),\end{aligned} \tag{B.22}$$

which is true for every derivable function $f(z)$ in 0. As described in the next sections, this property leads to strong simplifications which allows us to obtain the exact expression for the first moment and the variance.

B.4.3 Heaps' law first moment

Using Equation (B.20), the first moment of the Heaps' law reads:

$$E[h(s)] = \frac{1}{(\theta+1)_{s-1}} \frac{\partial}{\partial z} \mathcal{L}_{s-1}\mathcal{L}_{s-2}\dots\mathcal{L}_1 e^z|_{z=0}.$$

In order to use the property (B.22), the derivative must filter towards the right end of the equation, so that there are no operators acting on $\mathcal{L}$. To this end, one has to use commutator:

$$\left[\frac{\partial}{\partial z}, \mathcal{L}_s\right] = \frac{\partial \mathcal{L}_s}{\partial z} - \mathcal{L}_s \frac{\partial}{\partial z} = \mathcal{L}_0 + \alpha \frac{\partial}{\partial z}.$$

Therefore the action of the derivative on $\mathcal{L}_{s-1}$ leads to the expression:

$$E[h(s)] = \frac{1}{(\theta+1)_{s-1}} \left(\mathcal{L}_0 + (\mathcal{L}_{s-1}+\alpha)\,\frac{\partial}{\partial z}\right) \mathcal{L}_{s-2}\dots\mathcal{L}_1 e^z|_{z=0}.$$

Iterating this procedure over all the operators $\mathcal{L}$ and using the notation $\mathcal{L}_s^\alpha = \mathcal{L}_s + \alpha$, one finds the following expression:

$$E[h(s)] = \frac{1}{(\theta+1)_{s-1}} \left(\sum_{i=1}^{s-1} L^\alpha_{s-1,i+1}\mathcal{L}_0 L^0_{i-1,1} + L^\alpha_{s-1,1}\frac{\partial}{\partial z}\right) e^z|_{z=0},$$

where we have defined the new operator $L^\alpha_{i,j}$ as:

$$\begin{cases} L^\alpha_{i,j} = \displaystyle\prod_{k=j}^{i} (\mathcal{L}_k + \alpha) & \text{for } i \geq j \\ L^\alpha_{i,j} = 1 & \text{for } i = j-1 \end{cases}.$$

We now can take advantage of the property (B.22), and, for example, the terms outside the summation becomes:

$$\begin{aligned} L^{\alpha}_{s-1,1}e^z|_{z=0} &= (s-1+\theta+\alpha)L^{\alpha}_{s-2,1}e^z|_{z=0} = \\ &= (s-1+\theta+\alpha)(s-2+\theta+\alpha)L^{\alpha}_{s-3,1}e^z|_{z=0} = \\ &= \ldots = (\theta+\alpha+1)_{s-1}, \end{aligned}$$

which is a rising factorial, (B.21). Applying the property to all the terms, after some mathematical manipulations, the expression for the average Heaps' curve becomes:

$$E[h(s)] = \frac{\theta}{(\theta)_s}\sum_{i=0}^{s-1}(\theta+\alpha+i+1)_{s-i-1}(\theta)_i,$$

which is an hypergeometric series that can be computed with numerical methods, [119], (we used the Wolfram Mathematica software). The final exact expression for the Heaps' law average is then:

$$E[h(s)] = \begin{cases} \dfrac{\theta}{\alpha}\left(\dfrac{(\theta+\alpha)_s}{(\theta)_s} - 1\right) & \text{if } \alpha > 0 \\ \displaystyle\sum_{i=0}^{s-1}\frac{\theta}{\theta+i} & \text{if } \alpha = 0 \end{cases}. \tag{B.23}$$

Asymptotically, i.e. for $s \to \infty$, for $\alpha > 0$, one can rewrite the rising factorials in terms of gamma functions, (B.21), and then use the Stirling expansion. The result reads:

$$E[h(s)] \approx \frac{\Gamma(\theta+1)}{\alpha\Gamma(\theta+\alpha)}s^{\alpha}, \tag{B.24}$$

which is a power law increasing function as expected from mean field arguments [33]. The asymptotic behaviour for $\alpha = 0$ can be obtained with the integral approximation of Equation (B.23) presenting a logarithmic growth.

B.4.4 Heaps' law second moment and variance

To derive the second moment the procedure is similar, with the difference that now the first derivative in z have to be substituted with the second derivative:

$$E[h(s)^2] = \frac{1}{(\theta+1)_{s-1}}\frac{\partial^2}{\partial z^2}\mathcal{L}_{s-1}\mathcal{L}_{s-2}\ldots\mathcal{L}_1 e^z|_{z=0}.$$

As described in the previous section, the second derivative must filter towards e^z, and this can be done by using the following commutator:

$$\left[\frac{\partial^2}{\partial z^2}, \mathcal{L}_s\right] = \mathcal{L}_0 + (\alpha + 2\theta e^z)\frac{\partial}{\partial z} + 2e^z\alpha\frac{\partial^2}{\partial z^2}. \tag{B.25}$$

After some math, the second moment becomes:

$$E[h(s)^2] = \frac{1}{(\theta+1)_{M-1}} \left(L^{2\alpha e^s}_{M-1,1} \frac{\partial^2}{\partial s^2} + \sum_{i=1}^{M-1} L^{2\alpha e^s}_{M-1,i+1} \right.$$
$$\left. \left(\mathcal{L}_0 L_{i-1,1} + (2\theta e^s + \alpha) \sum_{j=1}^{i-1} L^{\alpha}_{i-1,j+1} \mathcal{L}_0 L_{j-1,1} + L^{\alpha}_{i-1,1} \frac{\partial}{\partial s} \right) \right) e^s.$$

Now the property B.22 can be safely applied. After some manipulation and computing numerically two hypergeometric series, one finds the analytical expression for the second moment ($\alpha > 0$):

$$\begin{aligned} E[h(s)^2] &= \frac{\theta}{\alpha^2(\theta)_s} \left((\alpha+\theta)(\theta+2\alpha)_s - (\alpha+2\theta)(\theta+\alpha)_s + +\theta(\theta)_s \right) \\ &\approx \frac{\theta+\alpha}{\alpha^2} \frac{\Gamma(\theta+1)}{\Gamma(\theta+2\alpha)} s^{2\alpha}, \end{aligned} \tag{B.26}$$

where the second formula is the asymptotic expression for $s \to \infty$, obtained using the Stirling expansion for the gamma function.

Finally, the variance of the Heaps' process can now be derived, $\mathrm{Var}[H(M)] = E[H(M)^2] - E[H(M)]^2$, and, for large M, it reads:

$$\begin{aligned} \mathrm{Var}[h(s)] &\approx \frac{\Gamma(\theta+1)}{\alpha^2} \left(\frac{\theta+\alpha}{\Gamma(\theta+2\alpha)} - \frac{\Gamma(\theta+1)}{\Gamma(\theta+\alpha)^2} \right) s^{2\alpha} = \\ &\approx \left(\frac{(\theta+\alpha)\Gamma(\theta+\alpha)^2}{\Gamma(\theta+2\alpha)\Gamma(\theta+1)} - 1 \right) E[h(s)]^2, \end{aligned} \tag{B.27}$$

which shows the Taylor's law observed in empirical systems. This formula is not valid for $\alpha = 0$, however in such a case p_{new} no longer depends on h, implying that the growth of the variance is bounded by the Poisson's scaling, result 6.15.

Appendix C

Fitness-Complexity algorithm

C.1 Computational issues

The fitness - complexity map is simulated through the iteration of the equation (7.2) until the stationary state is reached. A first technical issue is the presence of trajectories going to zero, since the presence of null fitness/complexity terms lead to infinite addenda, and therefore to computational errors. In order to fix this problem we define a lower boundary for each fitness and each complexity trajectory, such that values of F_i and Q_j lesser than the boundary are substituted with a value equal to the boundary. In this way, the decaying trajectories cannot cross the boundary, and once they reach the chosen threshold, they are imposed to be constant for all the remaining time steps. In this work the boundary value is equal to 10^{-100}. Testing different boundary values, it seems that the map does not change its behavior for sufficiently low values, lesser than 10^{-20}, implying that this alteration should not affect the map dynamics.

Another computational issue is the identification of the stationary condition, at which the map iteration stops. We employ the following simple procedure: at each time step we compute the average differential of the fitness trajectory ensemble, which reads as follows:

$$\langle d \rangle = \frac{1}{R} \sum_{i=1}^{R} \mid F_i^{(t)} - F_i^{(t-1)} \mid,$$

where N is the total number of realizations/active species. The map stops if one of the two following conditions are satisfied: $\langle d \rangle < \theta$, where the threshold is chosen equal to 10^{-6}, or the number of iterations exceeds a limit value (about 2000 iterations). The latter condition is required in particular rare cases where the relaxation time becomes extremely large, such as if there are power law decaying trajectories, or when a stationary state not exists.

A third crucial point is the ranking computation, i.e. the ordering of the nodes based on the map fitness, and therefore related to the node importance within the system. In a scenario where all the fitness trajectories converge to positive values, this ranking can be computed trivially. However the presence of trajectories which decay to zero is very common, and clearly these nodes cannot be compared on the basis of their stationary value, which is zero for all of them. In such cases, we distinguish between the nodes with positive stationary fitness, and the nodes with trajectory decaying to zero. The nodes in the first group are put at the top positions of the ranking, and they are ordered according to their stationary fitness value. The decaying trajectories are put at lower ranks, and, among this second group, the nodes are ranked according to the "velocity" of decaying. In order to determine which trajectory goes faster to zero, we consider a fitness threshold (very small but greater than the lower boundary described above, we have chosen 10^{-80}), assuming that the trajectories which cross this threshold first (at smaller time steps) have a faster decay, and therefore they are put at lower positions.

C.2 Comparison with the Mariani map

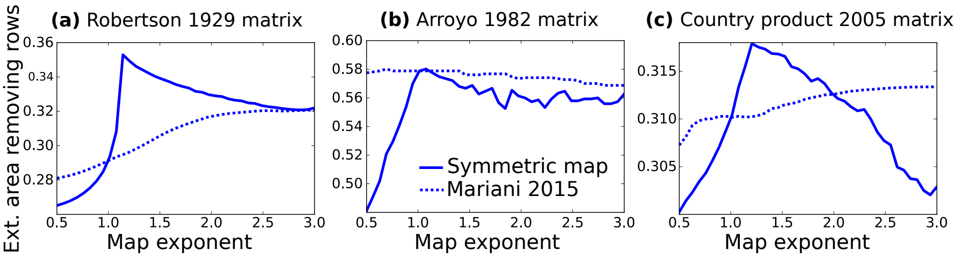


Figure C.1: **Comapring two fitness-complexiy map generalizations.** Extinction area computed through the Mariani generalization [109] and the symmetric one, for three different datasets. Consistently for $\gamma = 1$ both the algorithms are equivalent to the standard one and find the same value of extinction area. Panel (a) and (b) are mutualisitic system, dataset A.4, while panel (c) is the country-product matrix described in [103] for the 2005.

The map generalization proposed in [109] reads as follows:

$$\begin{cases} \tilde{F}_i^{(t)} = \sum_j n_{ij} Q_j^{(t-1)} & F_i^{(t)} = \frac{\tilde{F}_i^{(t)}}{\langle \tilde{F}^{(t)} \rangle} \\ \tilde{Q}_j^{(t)} = \left(\sum_i n_{ij} \left(F_i^{(t-1)} \right)^{-\gamma} \right)^{-\frac{1}{\gamma}} & Q_j^{(t)} = \frac{\tilde{Q}_j^{(t)}}{\langle \tilde{Q}^{(t)} \rangle}, \end{cases} \tag{C.1}$$

which is different from our symmetric generalization because of the inversion of the complexity exponent. Clearly, they find different F^* and Q^* rankings, and therefore different extinction areas, as shown in the figure C.1 as a function of the map exponent γ. From the examples considered, it seems that the generalization (7.2) finds always a larger or equal extinction area maximum. The two maximums coincide only for $\gamma = 1$, where the two maps are equivalent to the classical algorithm.

C.3 Critical exponent in uniform random matrices

In the following the generalized symmetric map (7.2) is applied to a large uniform random matrix with dimensions $d_1 \times d_2$ ($d_1, d_2 \gg 1$). The matrix elements are Bernoulli random variables, such that $n_{ij} = 1$ with probability p and $n_{ij} = 0$ with $1 - p$. Starting from the initial conditions $F_i^{(0)} = 1$ and $S_j^{(0)} = 1$ the first map step leads to sums of independent and identically distributed random variables, for example the non-normalized fitness reads: $\tilde{F}_i^{(1)} = \sum_i^{d_2} n_{ij}$. By using the central limit theorem one can obtain the following expressions (using the symmetric map (7.3)):

$$\begin{cases} F_i^{(1)} = 1 + \sqrt{\frac{1-p}{d_2 p}}\eta \\ S_j^{(1)} = 1 + \sqrt{\frac{1-p}{d_1 p}}\eta, \end{cases}$$

where η is the standard normal random variable ($\langle \eta \rangle = 0$ and $\mathrm{Var}(\eta) = 1$). At the next step the non-normalized fitness reads:

$$\tilde{F}_i^{(2)} = \sum_i^{d_2} n_{ij} \left(1 + \sqrt{\frac{1-p}{d_1 p}}\eta \right)^{-\gamma}$$

and a similar equation can be found for the simplicity $\tilde{S}_j^{(2)}$. The two expressions can be simplified under the following conditions:

$$\gamma\sqrt{\frac{1-p}{d_1 p}} \ll 1; \qquad \gamma\sqrt{\frac{1-p}{d_2 p}} \ll 1 \tag{C.2}$$

Approximating at the first order according to the conditions above and applying the central limit theorem again, after some algebra one finds that $F_i^{(2)}$ and $S_j^{(2)}$ become equal to their counterparts at the previous step shown in the equation above. This implies that the map converges immediately, and the stationary values follows a normal distribution which does not depend on γ. Moreover the condition (C.2) provides information about the convergence behaviour of the map, defining a region of parameters in which all the trajectories are positive. Indeed the critical exponent scaling with the matrix sizes and the density of ones are tested in the figure C.2 satisfying the prediction.

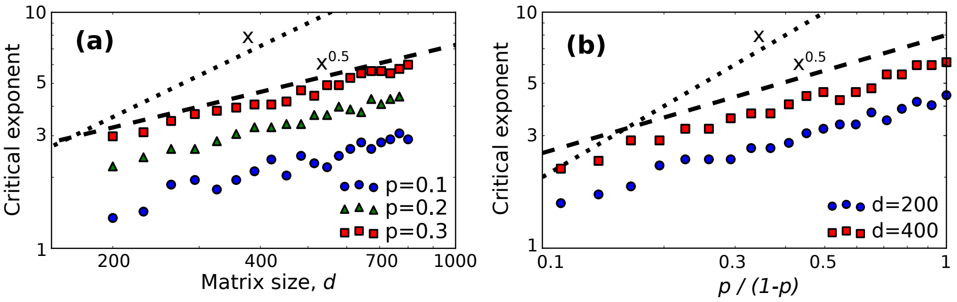


Figure C.2: **Critical exponent in unifrom random matrices.** The relation (C.2)can be used to predict the scaling of the map critical exponent, defined as the largest value of γ for which all the trajectories converge. The panel (a) tests the scaling in squared matrix having d rows and columns for three values of the matrix density p, while (b) considers the critical exponent as a function of a certain combination of p. The observed growth as the square root of the x-axis variable in both the cases is in agreement with the theoretical relation (C.2).

C.4 Genetic algorithm

Here we compare a genetic algorithm and the generalized fitness complexity map (7.2) in terms of finding a ranking of realizations which maximizes the extinction area (Fig. 7.2). The two algorithms are applied to a set of random binary square matrices with growing size. Let us call d the matrix size, as the number of rows and columns. The ensemble of random matrices is defined by the condition that a generic element n_{ij} is equal to 1 with probability $\frac{2ij}{d(d+1)}$, where $i = 1, \dots d$ and $j = 1, \dots d$, and zero otherwise. In this way the average of the row i degree is $d_i = \sum_j p_{ij} = i$, and similarly $d_j = j$, implying that there is heterogeneity of degrees (resembling empirical cases). In Figure C.3 is shown the best extinction area (a) and the computational time (b) as a function of the matrix size d, for the the two algorithm and the standard fitness-complexity map.

The genetic algorithm is defined as follows. The starting population is composed of of K row rankings generated at random (i.e. each on is a random permutation of the row indexes). At each time step two rankings are drawn from the population and the two associated extinction areas are computed. The row ordering with the lesser extinction area is substituted by a copy of the other "fitter" ranking, and this new copy can mutate with probability μ, where a mutation means that two row indexes exchange their position in the ranking. The algorithm stops if there is no improvement of the extinction area for t_{stop} iterations.

In order to compute Figure C.3 we need also a procedure to find the exponent of the generalized fitness complexity map which maximizes the

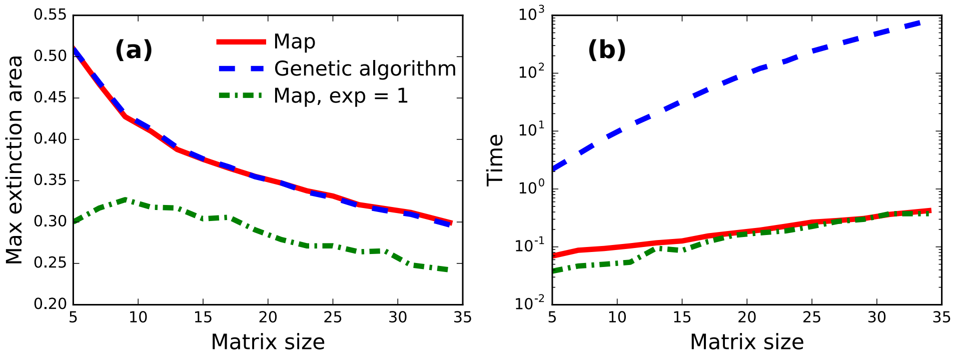


Figure C.3: **Comparison between a genetic algorithm, the standard fitness-complexity map and our generalization.** On the left there is the best extinction area found by each algorithm for increasing matrix size d, while on the right is shown the computational time (in seconds). The two quantities are the average over an ensemble of 200 random matrices for each size. The definition of the algorithms and procedure of the matrix generation are explained in the main text. The parameters of the genetic algorithm are: $K = 500$, $\mu = 0.05$, and $t_{stop} = 10^4$, while for the generalized map $\gamma_0 = 0.8$, $\Delta_0 = 0.2$, $\hat{\Delta} = 0.01$.

extinction area. We employ a method inspired by the bisection algorithm. This algorithm is based on the assumption that the extinction area grows monotonically from the map exponent $\gamma = 0$, to the area maximum, which seems to be approximately true in a lot of empirical cases, Figure C.5. The method is the following: at a generic step t, the extinction area is computed for an exponent γ_t and a second exponent $\gamma_{t+1} = \gamma_t + \Delta_t$. If the extinction area at γ_{t+1} is greater than the area of the first exponent, this means that I am approaching the maximum, and at the next time step we consider γ_{t+2}, that is $\gamma_{t+2} = \gamma_{t+1} + \Delta_{t+1}$, with $\Delta_{t+1} = \Delta_t$. Otherwise, if the area at γ_{t+1} is lesser, then the algorithm has surpassed the maximum, and the increment becomes smaller and changes sign: $\Delta_{t+1} = -\frac{\Delta_{t+1}}{2}$. The method stops if the exponent increment becomes smaller the a certain threshold $\hat{\Delta}$.

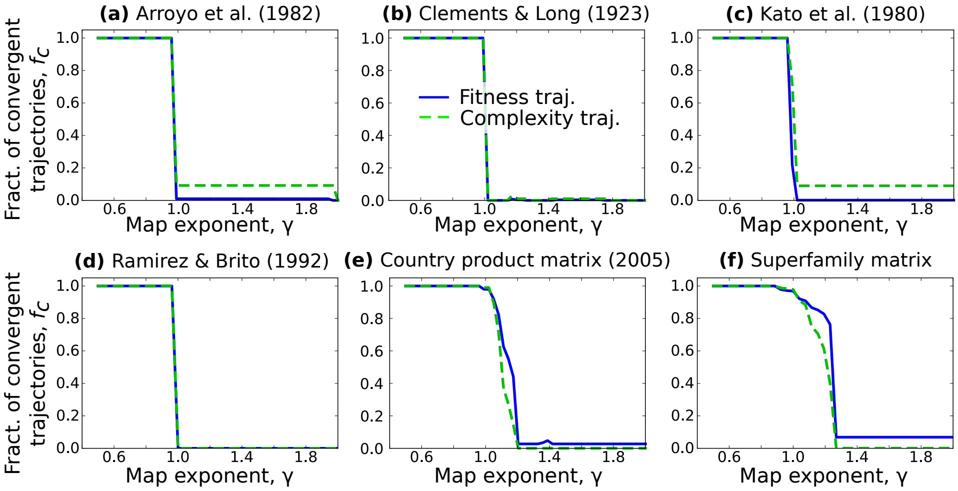


Figure C.4: **Fraction of convergent trajectories as a function of the map exponent in different datasets.** All the datasets display a phase transition in the number of positive stationary trajectories, and the critical exponent varies from one (*(a)*, *(b)*, *(c)*, *(d)*) to larger values (*(e)*, *(f)*). The first four panels refer to ecological mutualistic systems taken from the Interaction Web Database, whose dimensions are: 87 x 98, 96 x 276, 93 x 679, 33 x 53 respectively. The fifth panel is the country - product matrix described in [103] for the 2005, composed of 148 countries and 1176 exported products. The last plot refers to the binarized genomic system described in A.1.1.

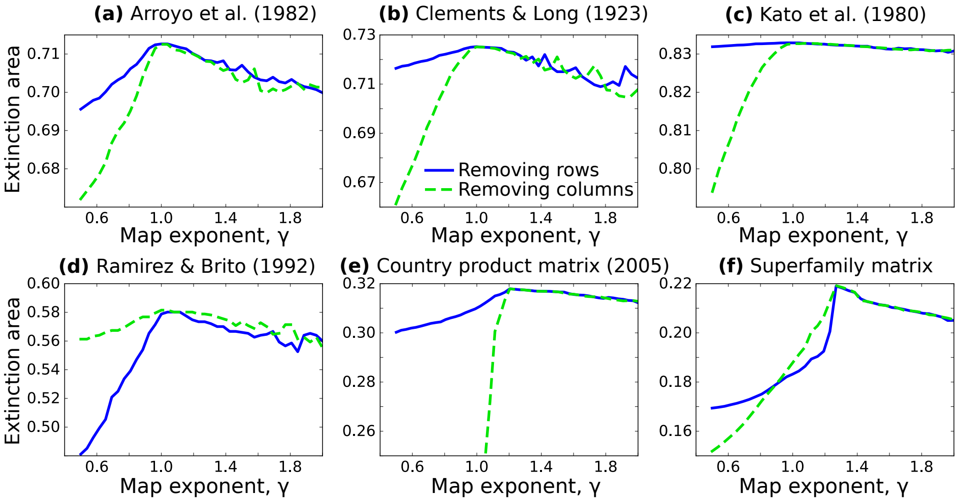


Figure C.5: **Extinction area vs. map exponent for the six datasets described in the figure C.4.**

Bibliography

[1] Sergei Maslov, Sandeep Krishna, Tin Yau Pang, and Kim Sneppen. Toolbox model of evolution of prokaryotic metabolic networks and their regulation. *Proceedings of the National Academy of Sciences*, 106(24):9743–9748, 2009.

[2] Tin Yau Pang and Sergei Maslov. Universal distribution of component frequencies in biological and technological systems. *Proceedings of the National Academy of Sciences*, 110(15):6235–6239, 2013.

[3] L. M. A. Bettencourt, J. Lobo, D. Helbing, C. Kühnert, and G. B. West. Growth, innovation, scaling, and the pace of life in cities. *Proc. Natl. Acad. Sci. USA*, 104(17):7301–7306, 2007.

[4] Rémi Louf and Marc Barthelemy. Modeling the polycentric transition of cities. *Physical review letters*, 111(19):198702, 2013.

[5] E. G. Altmann and M. Gerlach. Statistical laws in linguistics. In *Creativity and Universality in Language*, pages 7–26. Springer, 2016.

[6] Francesc Font-Clos and Álvaro Corral. Log-log convexity of type-token growth in zipfs systems. *Physical review letters*, 114(23):238701, 2015.

[7] Martin Gerlach and Eduardo G Altmann. Stochastic model for the vocabulary growth in natural languages. *Physical Review X*, 3(2):021006, 2013.

[8] Marco Gherardi, Salvatore Mandrà, Bruno Bassetti, and Marco Cosentino Lagomarsino. Evidence for soft bounds in ubuntu package sizes and mammalian body masses. *Proceedings of the National Academy of Sciences*, 110(52):21054–21058, 2013.

[9] Ben Dushnik and Edwin W Miller. Partially ordered sets. *American journal of mathematics*, 63(3):600–610, 1941.

[10] Eugene V Koonin. Are there laws of genome evolution? *PLoS computational biology*, 7(8):e1002173, 2011.

[11] Nick V Grishin, Yuri I Wolf, and Eugene V Koonin. From complete genomes to measures of substitution rate variability within and between proteins. *Genome research*, 10(7):991–1000, 2000.

[12] M. A. Huynen and E. van Nimwegen. The frequency distribution of gene family sizes in complete genomes. *Mol Biol Evol*, 15(5):583–589, May 1998.

[13] Erik van Nimwegen. Scaling laws in the functional content of genomes. *Trends in genetics*, 19(9):479–484, 2003.

[14] Yuri I Wolf, Pavel S Novichkov, Georgy P Karev, Eugene V Koonin, and David J Lipman. The universal distribution of evolutionary rates of genes and distinct characteristics of eukaryotic genes of different apparent ages. *Proceedings of the National Academy of Sciences*, 106(18):7273–7280, 2009.

[15] Georgy P Karev, Yuri I Wolf, Andrey Y Rzhetsky, Faina S Berezovskaya, and Eugene V Koonin. Birth and death of protein domains: a simple model of evolution explains power law behavior. *BMC evolutionary biology*, 2(1):18, 2002.

[16] Jacopo Grilli, Bruno Bassetti, Sergei Maslov, and Marco Cosentino Lagomarsino. Joint scaling laws in functional and evolutionary categories in prokaryotic genomes. *Nucleic acids research*, 40(2):530–540, 2011.

[17] Damián H Zanette. Statistical patterns in written language. *arXiv preprint arXiv:1412.3336*, 2014.

[18] George Kingsley Zipf. The psycho-biology of language. 1935.

[19] G. Altmann. Prolegomena to menzeraths law. *Glottometrika*, 2(2):1–10, 1980.

[20] G. Herdan. *Quantitative Linguistics*. Butterworth Press, Oxford, 1964.

[21] Kevin J Gaston. Global patterns in biodiversity. *Nature*, 405(6783):220–227, 2000.

[22] P. A. Marquet, R. A. Quiñones, S. Abades, F. Labra, M. Tognelli, M. Arim, and M. Rivadeneira. Scaling and power-laws in ecological systems. *Journal of Experimental Biology*, 208(9):1749–1769, 2005.

[23] Xavier Gabaix. Power laws in economics and finance. *Annu. Rev. Econ.*, 1(1):255–294, 2009.

[24] I Present. Cramming more components onto integrated circuits. *Readings in computer architecture*, 56, 2000.

[25] Xavier Gabaix. Zipf's law for cities: an explanation. *The Quarterly journal of economics*, 114(3):739–767, 1999.

[26] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[27] Michael Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251, 2004.

[28] Alfred J Lotka. The frequency distribution of scientific productivity. *Journal of the Washington academy of sciences*, 16(12):317–323, 1926.

[29] R. L. Axtell. Zipf distribution of us firm sizes. *Science*, 293(5536):1818–1820, 2001.

[30] G Udny Yule. A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical transactions of the Royal Society of London. Series B, containing papers of a biological character*, 213:21–87, 1925.

[31] Herbert A Simon. On a class of skew distribution functions. *Biometrika*, 42(3/4):425–440, 1955.

[32] J. Qian, N. M. Luscombe, and M. Gerstein. Protein family and fold occurrence in genomes: power-law behaviour and evolutionary model. *J Mol Biol*, 313(4):673–681, Nov 2001.

[33] Marco Cosentino Lagomarsino, Alessandro L Sellerio, Philip D Heijning, and Bruno Bassetti. Universal features in the genome-level evolution of protein domains. *Genome biology*, 10(1):R12, 2009.

[34] Damián Zanette and Marcelo Montemurro. Dynamics of text generation with realistic zipf's distribution. *Journal of quantitative Linguistics*, 12(1):29–40, 2005.

[35] George A Miller. Some effects of intermittent silence. *The American journal of psychology*, 70(2):311–314, 1957.

[36] Elliott W Montroll and Michael F Shlesinger. On 1/f noise and other distributions with long tails. *Proceedings of the National Academy of Sciences*, 79(10):3380–3383, 1982.

[37] Moshe Levy and Sorin Solomon. Power laws are logarithmic boltzmann laws. *International Journal of Modern Physics C*, 7(04):595–601, 1996.

[38] Ofer Malcai, Ofer Biham, and Sorin Solomon. Power-law distributions and levy-stable intermittent fluctuations in stochastic systems of many autocatalytic elements. *Physical Review E*, 60(2):1299, 1999.

[39] B. Corominas-Murtra, R. Hanel, and S. Thurner. Understanding scaling through history-dependent processes with collapsing sample space. *Proc. Natl. Acad. Sci. USA*, 112(17):5348–5353, 2015.

[40] Benoit Mandelbrot. An informational theory of the statistical structure of language. *Communication theory*, 84:486–502, 1953.

[41] R. F. i Cancho and R. V. Solé. Least effort and the origins of scaling in human language. *Proc. Natl. Acad. Sci. USA*, 100(3):788–791, 2003.

[42] J. M. Carlson and J. Doyle. Highly optimized tolerance: A mechanism for power laws in designed systems. *Phys. Rev. E*, 60(2):1412, 1999.

[43] E. Fabrikant, A.and Koutsoupias and C. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the internet. *Automata, languages and programming*, pages 781–781, 2002.

[44] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Phys. Rev. Lett.*, 59(4):381, 1987.

[45] Thierry Mora and William Bialek. Are biological systems poised at criticality? *Journal of Statistical Physics*, 144(2):268–302, 2011.

[46] S. K. Baek, S. Bernhardsson, and P. Minnhagen. Zipf's law unzipped. *New J. Phys.*, 13(4):043004, 2011.

[47] David J Schwab, Ilya Nemenman, and Pankaj Mehta. Zipfs law and criticality in multivariate data without fine-tuning. *Physical review letters*, 113(6):068102, 2014.

[48] Harold Stanley Heaps. *Information retrieval: Computational and theoretical aspects.* Academic Press, Inc., 1978.

[49] C. Cattuto, V. Loreto, and L. Pietronero. Semiotic dynamics and collaborative tagging. *Proc. Natl. Acad. Sci. USA*, 104(5):1461–1464, 2007.

[50] Zi-Ke Zhang, Linyuan Lü, Jian-Guo Liu, and Tao Zhou. Empirical analysis on a keyword-based semantic system. *The European Physical Journal B-Condensed Matter and Complex Systems*, 66(4):557–561, 2008.

[51] Hongyu Zhang. Discovering power laws in computer programs. *Information Processing & Management*, 45(4):477–483, 2009.

[52] R. Baeza-Yates and G. Navarro. Block addressing indices for approximate text retrieval. *J. Assoc. Inf. Sci. Technol.*, 51(1):69–82, 2000.

[53] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182, 2011.

[54] Linyuan Lü, Zi-Ke Zhang, and Tao Zhou. Zipf's law leads to heaps' law: Analyzing their relation in finite-size systems. *PloS one*, 5(12):e14139, 2010.

[55] Dick C van Leijenhorst and Th P Van der Weide. A formal derivation of heaps' law. *Information Sciences*, 170(2):263–272, 2005.

[56] I. Eliazar. The growth statistics of zipfian ensembles: Beyond heaps law. *Physica A*, 390(20):3189–3203, 2011.

[57] Marie Touchon, Claire Hoede, Olivier Tenaillon, Valérie Barbe, Simon Baeriswyl, Philippe Bidet, Edouard Bingen, Stéphane Bonacorsi, Christiane Bouchier, Odile Bouvet, et al. Organised genome dynamics in the escherichia coli species results in highly diverse adaptive paths. *PLoS genetics*, 5(1):e1000344, 2009.

[58] Eugene V Koonin and Yuri I Wolf. Genomics of bacteria and archaea: the emerging dynamic view of the prokaryotic world. *Nucleic acids research*, 36(21):6688–6719, 2008.

[59] Bart Haegeman and Joshua S Weitz. A neutral theory of genome evolution and the frequency distribution of genes. *BMC genomics*, 13(1):196, 2012.

[60] Alexander E Lobkovsky, Yuri I Wolf, and Eugene V Koonin. Gene frequency distributions reject a neutral model of genome evolution. *Genome biology and evolution*, 5(1):233–242, 2013.

[61] F. Baumdicker, W. R. Hess, and P. Pfaffelhuber. The infinitely many genes model for the distributed genome of bacteria. *Genome Biol. Evol.*, 4(4):443–456, 2012.

[62] R. E. Collins and P. G. Higgs. Testing the infinitely many genes model for the evolution of the bacterial core genome and pangenome. *Mol. Biol. Evol.*, 29(11):3413–3425, 2012.

[63] Derek Wilson, Martin Madera, Christine Vogel, Cyrus Chothia, and Julian Gough. The superfamily database in 2007: families and functions. *Nucleic acids research*, 35(suppl 1):D308–D313, 2007.

[64] Eugene V Koonin. Comparative genomics, minimal gene-sets and the last universal common ancestor. *Nature reviews. Microbiology*, 1(2):127, 2003.

[65] Nacho Molina and Erik van Nimwegen. Scaling laws in functional genome content across prokaryotic clades and lifestyles. *Trends in genetics*, 25(6):243–247, 2009.

[66] Jacopo Grilli, Mariacristina Romano, Federico Bassetti, and Marco Cosentino Lagomarsino. Cross-species gene-family fluctuations reveal the dynamics of horizontal transfers. *Nucleic acids research*, 42(11):6850–6860, 2014.

[67] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1(Oct):49–75, 2000.

[68] Dror Y Kenett, Michele Tumminello, Asaf Madi, Gitit Gur-Gershgoren, Rosario N Mantegna, and Eshel Ben-Jacob. Dominating clasp of the financial sector revealed by partial correlation analysis of the stock market. *PloS one*, 5(12):e15032, 2010.

[69] Miguel A Fortuna, Juan A Bonachela, and Simon A Levin. Evolution of a modular software network. *Proceedings of the National Academy of Sciences*, 108(50):19985–19989, 2011.

[70] Stefan Thurner, Rudolf Hanel, Bo Liu, and Bernat Corominas-Murtra. Understanding zipf's law of word frequencies through sample-space collapse in sentence formation. *Journal of the Royal Society Interface*, 12(108):20150330, 2015.

[71] B. Corominas-Murtra, R. Hanel, and S. Thurner. Extreme robustness of scaling in sample space reducing processes explains zipfs law in diffusion on directed networks. *New J. Phys.*, 18(9):093010, 2016.

[72] B. Corominas-Murtra, R. Hanel, and S. Thurner. Sample space reducing cascading processes produce the full spectrum of scaling exponents. *arXiv preprint arXiv:1703.10100*, 2017.

[73] B. Corominas-Murtra, R. Hanel, L. Zavojanni, and S. Thurner. How noise determines the statistics of simple path dependent systems. *arXiv preprint arXiv:1706.10202*, 2017.

[74] Jim Pitman. Exchangeable and partially exchangeable random partitions. *Probability theory and related fields*, 102(2):145–158, 1995.

[75] E. G. Altmann, G. Cristadoro, and M. Degli Esposti. On the origin of long-range correlations in texts. *Proc. Natl. Acad. Sci. USA*, 109(29):11582–11587, 2012.

[76] Renaud Lambiotte, Marcel Ausloos, and Mike Thelwall. Word statistics in blogs and rss feeds: Towards empirical universal evidence. *Journal of Informetrics*, 1(4):277–286, 2007.

[77] E. Alvarez-Lacalle, B. Dorow, J. P. Eckmann, and E. Moses. Hierarchical structures induce long-range dynamical correlations in written texts. *Proc. Natl. Acad. Sci. USA*, 103(21):7956–7961, 2006.

[78] E. G. Altmann, J. B. Pierrehumbert, and A. E. Motter. Beyond word frequency: Bursts, lulls, and scaling in the temporal distributions of words. *PLoS One*, 4(11):e7678, 2009.

[79] Francesca Tria, Vittorio Loreto, Vito Domenico Pietro Servedio, and Steven H Strogatz. The dynamics of correlated novelties. *Scientific reports*, 4:5890, 2014.

[80] C. Cattuto, V. Loreto, and V. D. P. Servedio. A yule-simon process with memory. *EPL*, 76(2):208, 2006.

[81] Fang Wu and Bernardo A Huberman. Novelty and collective attention. *Proceedings of the National Academy of Sciences*, 104(45):17599–17601, 2007.

[82] M. Gerlach and E. G. Altmann. Scaling laws and fluctuations in the statistics of word frequencies. *New Journal of Physics*, 16(11):113010, 2014.

[83] L. R. Taylor. Aggregation, variance and the mean. *Nature*, 189(4766):732–735, 1961.

[84] AM Kilpatrick and AR Ives. Species interactions can explain taylor's power law for ecological time series. *Nature*, 422(6927):65–68, 2003.

[85] Joel E Cohen, Meng Xu, and William SF Schuster. Stochastic multiplicative population growth predicts and interprets taylor's power law of fluctuation scaling. *Proceedings of the Royal Society of London B: Biological Sciences*, 280(1757):20122955, 2013.

[86] Andrea Giometto, Marco Formentin, Andrea Rinaldo, Joel E Cohen, and Amos Maritan. Sample and population exponents of generalized taylors law. *Proceedings of the National Academy of Sciences*, 112(25):7755–7760, 2015.

[87] Stefano Galluccio, Guido Caldarelli, Matteo Marsili, and Y-C Zhang. Scaling in currency exchange. *Physica A: Statistical Mechanics and its Applications*, 245(3-4):423–436, 1997.

[88] Zoltán Eisler, Imre Bartos, and János Kertész. Fluctuation scaling in complex systems: Taylor's law and beyond. *Advances in Physics*, 57(1):89–142, 2008.

[89] A. Fronczak and P. Fronczak. Origins of taylors power law for fluctuation scaling in complex systems. *Physical Review E*, 81(6):066112, 2010.

[90] J. Ramsayer, S. Fellous, J. E Cohen, and M. E. Hochberg. Taylor's law holds in experimental bacterial populations but competition does not influence the slope. *Biology letters*, 8(2):316–319, 2012.

[91] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[92] George Pólya. Sur quelques points de la théorie des probabilités. *Ann. Inst. H. Poincaré*, 1(2):117–161, 1930.

[93] Norman Lloyd Johnson and Samuel Kotz. Urn models and their application; an approach to modern discrete probability theory. 1977.

[94] Jim Pitman and Marc Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.

[95] Jim Pitman et al. Combinatorial stochastic processes. 2002.

[96] Bruno Bassetti, Mina Zarei, Marco Cosentino Lagomarsino, and Ginestra Bianconi. Statistical mechanics of the chinese restaurant process: Lack of self-averaging, anomalous finite-size effects, and condensation. *Physical Review E*, 80(6):066118, 2009.

[97] A Angelini, A Amato, G Bianconi, B Bassetti, and M Cosentino Lagomarsino. Mean-field methods in evolutionary duplication-innovation-loss models for the genome-level repertoire of protein domains. *Physical Review E*, 81(2):021919, 2010.

[98] Antonio Rosanova, Alberto Colliva, Matteo Osella, and Michele Caselle. Modelling the evolution of transcription factor binding preferences in complex eukaryotes. *Scientific Reports*, 7, 2017.

[99] Benoit B Mandelbrot and Roberto Pignoni. *The fractal geometry of nature*, volume 173. WH freeman New York, 1983.

[100] Jianzhi Zhang. Evolution by gene duplication: an update. *Trends in ecology & evolution*, 18(6):292–298, 2003.

[101] Eugene V Koonin, Kira S Makarova, and L Aravind. Horizontal gene transfer in prokaryotes: quantification and classification. *Annual Reviews in Microbiology*, 55(1):709–742, 2001.

[102] Luigi Grassi, Jacopo Grilli, and Marco Cosentino Lagomarsino. Large-scale dynamics of horizontal transfers. *Mobile genetic elements*, 2(3):163–167, 2012.

[103] Andrea Tacchella, Matthieu Cristelli, Guido Caldarelli, Andrea Gabrielli, and Luciano Pietronero. A new metrics for countries' fitness and products' complexity. *Scientific reports*, 2:723, 2012.

[104] Virginia Domínguez-García and Miguel A Muñoz. Ranking species in mutualistic networks. *Scientific reports*, 5, 2015.

[105] Andrea Tacchella, Matthieu Cristelli, Guido Caldarelli, Andrea Gabrielli, and Luciano Pietronero. Economic complexity: conceptual grounding of a new metrics for global competitiveness. *Journal of Economic Dynamics and Control*, 37(8):1683–1691, 2013.

[106] Matthieu Cristelli, Andrea Gabrielli, Andrea Tacchella, Guido Caldarelli, and Luciano Pietronero. Measuring the intangibles: A metrics for the economic complexity of countries and products. *PloS one*, 8(8):e70726, 2013.

[107] Matthieu Cristelli, Andrea Tacchella, and Luciano Pietronero. The heterogeneous dynamics of economic complexity. *PloS one*, 10(2):e0117174, 2015.

[108] Giulio Cimini, Andrea Gabrielli, and Francesco Sylos Labini. The scientific competitiveness of nations. *PLoS One*, 9(12):e113470, 2014.

[109] Manuel Sebastian Mariani, Alexandre Vidmer, Matsúš Medo, and Yi-Cheng Zhang. Measuring economic complexity of countries and products: which metric to use? *The European Physical Journal B*, 88(11):1–9, 2015.

[110] Emanuele Pugliese, Andrea Zaccaria, and Luciano Pietronero. On the convergence of the fitness-complexity algorithm. *arXiv preprint arXiv:1410.0249*, 2014.

[111] Stefano Allesina and Mercedes Pascual. Googling food webs: can an eigenvector measure species' importance for coextinctions? *PLoS computational biology*, 5(9):e1000494, 2009.

[112] Werner Ulrich, Mário Almeida-Neto, and Nicholas J Gotelli. A consumer's guide to nestedness analysis. *Oikos*, 118(1):3–17, 2009.

[113] Christine A. Orengo and Janet M. Thornton. Protein families and their evolution-a structural perspective. *Annu Rev Biochem*, 74:867–900, 2005.

[114] Christine Vogel and Cyrus Chothia. Protein family expansions and biological complexity. *PLoS Comput Biol*, 2(5):e48, 2006.

[115] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, et al. The pfam protein families database. *Nucleic Acids Res.*, 32(suppl_1):D138–D141, 2004.

[116] Robert D Finn, Alex Bateman, Jody Clements, Penelope Coggill, Ruth Y Eberhardt, Sean R Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, et al. Pfam: the protein families database. *Nucleic acids research*, 42(D1):D222–D230, 2013.

[117] Shibamouli Lahiri. Complexity of word collocation networks: A preliminary structural analysis. pages 96–105, April 2014.

[118] John C Marlin and Wallace E LaBerge. The native bee fauna of carlinville, illinois, revisited after 75 years: a case for persistence. *Conservation Ecology*, 5(1):9, 2001.

[119] Marko Petkovsek, Herbert S Wilf, and Doron Zeilberger. *A+ B*. A K. Peters, 1996.