The grammar of protein domain architectures

Supplementary information

Lijia Yu, Deepak Kumar Tanwar, Emanuel Diego S. Penha, Yuri Wolf, Eugene V. Koonin, Malay Kumar Basu

Contents

List of Figures

1 Supplementary methods

1.1 Domain structure determination

We used two methods to determine the domain composition of all the proteins encoded in each selected genome. In the first method, we used `hmmscan` from the HMMER (v. 3.1b)(1) package using default parameters to identify

domains in each genomes using Pfam-A database (release 30)(2). We then used a Perl script to parse the output of `hmmscan` to find non-overlapping regions on the protein sequence corresponding to each domain. The parsing is based on the following criteria:

1. "Independent e-value (i-Evalue)" for the domain hit is less than 0.001
2. Only "domain" and "family" were retained
3. When two hits overlap, only the one with the highest bit score ("full sequence") was kept.
4. Each Pfam ID was checked to determine whether it belongs to a clan. If so, then Pfam ID was replaced with the clan id.

In the second method, which was used mainly for the verification of the results, we used domain mappings of proteomes as provided by the Pfam directly. As Pfam did not provide data for all the species in UniProt, this data is a subset, including 426 eukaryotic, 2543 bacterial and 127 archaeal genomes, of the UniProt dataset. The results from the two methods were identical for all the overlapping species sets, therefore, unless otherwise specified, the presented results are from the first method using the larger dataset.

1.2 The n-gram model of a protein language

If a protein sequence is composed of domains $(d_1, \cdots, d_n)$ the probability of the whole protein is given by the chain rule of probability (3),

$$P(d_1, \cdots, d_n) = P(d_1) \times P(d_2 \mid d_1) \times P(d_3 \mid d_1, d_2) \times \cdots \times P(d_n \mid d_1, \cdots, d_{n-1}). \tag{1}$$

The chain rule shows that the joint probability of a series of domains can be determined by the multiplication of conditional probabilities, each denoting the probability of a domain given the series of preceding domains. In the case of protein domains, it is impossible to calculate the probability of a domain given every long string of preceding domains because of the scarcity of such data in a genome. However, the conditional probabilities can be approximated by a Markov process, where the probability of a given domain depends only on a limited number of preceding domains. The number of preceding domains determines the order: a first order Markov process considers one preceding domain (bigram), a second order considers two (trigram), and so on.

Modeling domains under a first order Markov process, where the probability of a domain (d_n) depends only on the one preceding domain (d_{n-1}) is called bigram model, and the domain pair (d_{n-1}, d_n) is called a bigram. The conditional probability of the domain (d_n) given the preceding domain (d_{n-1}) is as follows:

$$P(d_n \mid d_{n-1}) = \frac{P(d_{n-1}, d_n)}{P(d_{n-1})} \tag{2}$$

We can estimate this probability using the maximum likelihood estimation (MLE):

$$P_{MLE}(d_n \mid d_{n-1}) = \frac{C(d_{n-1}, d_n)}{C(d_{n-1})} \tag{3}$$

where $C(d_{n-1}, d_n)$ is the count of the bigram (d_{n-1}, d_n) in the genome, and $C(d_{n-1})$ is the count of all bigrams where the first domain is d_{n-1}, which is equivalent to the count of domain d_{n-1} in the genome.

1.3 Good-Turing Smoothing

The biggest problem with N-gram models is the sparsity of the data. Most of the domains are present in strictly constrained contexts and participate only in a restricted number of domain pairs. A large number of the conditional probabilities are, therefore, 0 in a genome. Some of the ways in which this sparsity of the data can be handled are described below.

The simplest way to tackle the 0 probabilities is to add 1 to all bigram counts. This "add-one" smoothing or "Laplace correction" has been used for many sparse datasets. However, because of the nature of the distribution of bigrams in a genome (power-law with extended tail of low counts (4, 5), the method adds too much weight to

missing data and cannot be used in N-gram modeling (3). In computational linguistics, numerous other methods have been developed to address this problems, such as Katz backoff (6), Kneser-Ney algorithm (7), etc.

In contrast to add-one smoothing where we add counts to bigrams with 0 count, discounting methods decrease counts for the existing bigrams to assign probabilities to unobserved bigrams. The most popular of all the discounting methods is Good-Turing smoothing (8). The first step in a Good-Turing estimate is to create a table of "frequencies of frequencies" where the observations are binned into J number of bins where $S(J)$ is the count of bigrams that are contained exactly J times in the genome. For example, $S(J_1)$ is the count of bigrams present exactly once in the genome (singletons); $S(J_2)$ is the count of bigrams present exactly twice, and so on. The Good-Turing method discounts the probability from existing bigrams in the genome to bigrams that are absent. If the MLE count (as in Eq. 3) of a bigram is C, then a Good-Turing count will be,

$$f_{GT} = (C+1)\frac{S(Jc+1)}{S(J_c)} \tag{4}$$

We used a modified version of the original Good-Turing smoothing called Simple Good-Turing (SGT)(9), where the estimate of the missing bigram is calculated from a linear regression of a log-log plot of frequency vs. frequency of frequency of bigrams in the genome. The method takes advantage of the power-law characteristic of this relationship. We used the public domain SGT calculator for this calculation (https://www.grsampson.net/D_SGT.c). SGT gives us the modified probabilities (P_{SGT}) of each bigram type in the genome and the total probability of all missing bigrams (P_0). After determining the P_{SGT}, these probabilities were renormalized so that they sum to 1, and the normalized probabilities were multiplied by the total bigram count to obtain the new SGT count for each bigram.

The Good-Turing smoothing assigns the probability estimate of all the missing bigrams in the genome to be $S(J_1)/N$ (given as P_0 in SGT), where J_1 is the count of singletons in the genome and N is the total number of bigrams in the genome. This probability space will be divided equally among the bigrams that are missing from the genome. If the number of domain types (number of unique domain families in a genome) is V (vocabulary), then, the number of all possible bigram types is V^2. If there are Bt types of bigrams in the genome, then, the number of missing types are $(V^2 - Bt)$. The total probability space P_0 will be equally divided amongst all the missing bigram. Therefore, the probability of a missing bigram is:

$$P(Missing) = \frac{P_0}{V^2 - Bt} \tag{5}$$

Once the Good-Turing counts are estimated, these counts were used to calculate the conditional probabilities as shown in Eq. 3.

1.4 Calculation of the difference and overlap between two entropy distribution

The statistical significance of the difference between the medians of two distributions was estimated by a permutation test. The categories of the data were randomly shuffled 1000 times, each time calculating the difference of the median between the two categories. The P-value of the observed difference between the medians of the two categories was then obtained from the distribution generated using random shuffling.

The overlap between two distributions was measured by calculating the fraction of discordant points in two distributions. To this end, all pairwise points in the two distributions were compared and the minimum of the following two values were calculated: (a) number of points in one distribution having values greater than the points in the other distribution and, (b) number of points in one distribution having values smaller than the points in the other distribution. The lower of these two values is a conservative estimate of the similarity between the two distributions. This number was then divided by the total number of comparisons. If two distributions are non-overlapping, all the points in one distribution will show the same trend and thus the number of discordant points will be 0. If two distributions are identical, the two numbers will be identical, with half of the points being discordant, and thus, the maximum overlap value calculated with this method will be 0.5.

An independent calculation of the overlap between two distributions was performed using the Bhattacharyya coefficient (10), a well-known measure of similarity between two distributions. To compare two distributions, each

was partitioned into optimal bins through the kernel density estimate using the R function `bw.nrd0`, and the similarity between the partitions was estimated using the formula:

$$BC(p,q) = \sum_{i=1}^{n} \sqrt{p_i q_i} \tag{6}$$

where p and q are the two distributions, n is the number of bins, and p_i and q_i are members of the corresponding distributions in the given bin.

1.5 Cross entropy (perplexity)

Given that N-gram language models are generative, "cross entropy" or "perplexity" is a measure of how well a given language model describes a language (3). If the sequence of domains in a genome is $D = d_1, d_2, \cdots, d_N$, the perplexity of the genome is measured as:

$$PP(D) = P(d_1, d_2, \cdots, d_N)^{-\frac{1}{N}} \tag{7}$$

$$= \sqrt[N]{\frac{1}{P(d_1, d_2, \cdots, d_N)}} \tag{8}$$

where, N is the total number of domains in the genome. Given the bigram model of the genome, the above equation can be expanded using the chain rule of probability:

$$PP(D) = \sqrt[N]{\prod_{n=1}^{N} \frac{1}{P(d_n \mid d_{n-1})}} \tag{9}$$

From this equation, it is clear that the higher the conditional probabilities of a bigram, the lower the perplexity. Because in a bigram model, the probability of a genome can be approximated by the sum of weighted probabilities of each bigram present in the genome, perplexity described by Eq. 8 can be rewritten in terms of entropy of the language model (for a detailed explanation of this relationship, please refer (3). Breiefly, if d_i is a domain, then the entropy of a unigram model is:

$$H_w = -\frac{1}{N} \sum_{N} count(d_i) \times log_2 P(d_i) \tag{10}$$

The right side of the equation is nothing but the log probability of the genome in terms of domains or bigrams, divided by number of domains and then taking the negative of the value.

$$H_w = -\frac{1}{N} log_2 P(d_1, d_2, \cdots, d_n) \tag{11}$$

$$= log_2 P(d_1, d_2, \cdots, d_n)^{-\frac{1}{N}} \tag{12}$$

This can be written as,

$$2^{H_w} = P(d_1, d_2, \cdots, d_n)^{-\frac{1}{N}} \tag{13}$$

$$= \sqrt[N]{\frac{1}{P(d_1, d_2, \cdots, d_N)}} \tag{14}$$

The right hand side of this equation is perplexity as shown in Eq. 8.

The perplexity can, therefore, be written in terms of entropy as,

$$PP(D) = 2^{H_w} \tag{15}$$

2 Supplementary table description

2.1 Table S1:

List of species and their taxonomic and genomics information. Column description:

1. Scientific Name
2. UniProt database ID
3. Taxonomy ID in NCBI Taxonomy database
4. Superkingdom (Bacteria/Archaea/Eukaryote) as provided by UniProt
5. Kingdom/Phyla information as provided by UniProt
6. Total amino acid count in the genome
7. Total protein count in the genome

2.2 Table S2:

Selected eukaryotic species for the phylogenetic analysis. Column description:

1. A shortened key used in the figure.
2. Taxonomy ID in NCBI Taxonomy database
3. UniProt database ID
4. Scientific name
5. Superkingdom information as provided by UniProt
6. Eukaryotic group information as provided by UniProt
7. Manually curated supergroups

2.3 Table S3:

Domain statistics and unigram and bigram entropy of each genome in the study. Column description:

1. Scientific name
2. UniProt ID
3. NCBI Taxonomy ID
4. Superkingdom information as provided by UniProt
5. Subdivision or kingdom or phyla information as provided by UniProt
6. Total domain count in the genome
7. Unique domain types in the genome
8. Total bigram count in the genome
9. Unique bigram types in the genome
10. Missing bigram types in the genome. Calculated as $(domain\ types)^2 - (bigram\ types)$.
11. Entropy of unigram model
12. Entropy of bigram model
13. Total domain count in the genome after adding N-C markers (see text)
14. Unique domain types in the genome after adding N-C markers (see text)
15. Total bigram count in the genome after adding N-C markers (see text)
16. Unique bigram types in the genome after adding N-C markers (see text)
17. Missing bigram types in the genome after adding N-C markers (see text). Calculated as $(domain\ types\ with\ NC)^2 - (bigram\ types\ with\ NC)$.
18. Entropy of unigram model after adding N-C markers (see text)
19. Entropy of bigram model after adding N-C markers (see text)

20. Bigram entropy of the genome after shuffling the domain (see text).
21. Relative entropy, calculated as (Column 11 - Column 12).
22. Relative entropy with N-C markers, calculated as (Column 18 - Column 19).
23. Relative entropy of shuffled genome, calculated as (Column 18 - Column 20).
24. Relative entropy of empirical and shuffled bigram models, calculated as (Column 20 - Column 19).

2.4 Table S4

Log-linear regression statistics of entropy values and log converted biological variables. Column description:

1. Superkingdom/kingdom/phyla
2. Biological variables: domain type, domain count, protein count, and amino acid count
3. Slope of the linear regression line between unigram entropy vs log10(variable)
4. R^2 of the regression
5. P value of the regression
6. Slope of the linear regression line between bigram entropy vs log10(variable)
7. R^2 of the regression
8. P value of the regression

2.5 Table S5

The median unigram, bigram, shuffled bigram entropies, and three relative entropies in all supkerkingdoms and major subdivisions of life.

1. Superkingdoms
2. Kingdom or phylum
3. Median unigram entropy with N-C markers (bits/domain)
4. Median bigram entropy after shuffling the domains.
5. Median bigram entropy after adding N-C markers
6. Median relative entropies, calculated by taking the median of (Column 3 - Column 5) values for all genomes.
7. Median relative entropies of shuffled genome, calculated by taking the median of (Column 3 - Column 4) for all genomes.
8. Median relative entropies of bigram models, calculated by taking the median of (Column 4 - Column 5) for all genomes.

2.6 Table S6

Pairwise differences of the medians and calculation of overlap between relative entropy distributions in 3 superkingdoms. Column description:

1. First superkingdom
2. Second superkingdom
3. Whether N and C markers were included in the calculation (see Methods). "NC" indicates with these markers and "WNC" indicate without.
4. P-value for permutation test (see Methods).
5. Bhattacharyya coefficient
6. Calculation of overlap (see Methods). Max value possible is 0.5.

2.7 Table S7

Bigram identified using sPLS-DA feature selection. Column description:

1. Domain bigram
2. Evolutionary split in which the domain bigram was found to be a feature
3. Loading weights for the split using multiclass sPLS-DA

4. Loading weights for the split using binary sPLS-DA
5. Pfam domain Id for the first domain of the bigram
6. Pfam domain name for the first domain of the bigram
7. Pfam domain id for the second domain in the bigram
8. Pfam domain name for the second domain in the bigram
9. Top 5 gene ontology (GO) terms associated with proteins that contain the bigram

2.8 Table S8

Top 10 gene ontology (GO) terms in each GO categories associated with the proteins that contain the bigrams identified using sPLS-DA feature selection. Column description:

1. Evolutionary splits
2. GO categories
3. GO terms
4. Frequency of the GO terms

3 Supplementary figure legends

3.1 Supplementary Fig. S1

Log-linear regression of the unigram and bigram entropy with various genomic variables in the three superkingdoms of life. Each point corresponds to a genome. Each column corresponds to a superkingdom. Each row corresponds to a variable against which the regression of entropy is performed. These variables, from top to bottom, are: number of unique domain families in a genome, or domain type **(A to C)**, total domain count **(D to F)**, number of proteins **(G to I)**, and the total count of amino acids **(J to L)** in the genome. In each panel, the X-axis is converted to the log10 scale. The slope of the regression lines, the R2 and P values are indicated on top of each plot. Note the top panel if this figure is identical as the Fig. 1 in the main text.

3.2 Supplementary Fig. S2

Linear regression of the unigram and bigram entropy with various genomic parameters in three superkingdoms of life. Each point corresponds to a genome. Each column corresponds to a superkingdom. From left to right, they are Bacteria, Archaea, and Eukaryota. Each row is a specific parameter. From top to bottom they are: unique domain type in the genome **(A to C)**, total domain count **(D to F)**, number of proteins **(G to I)**, and the total count of amino acids **(J to L)** in the genome. The slope of the regression lines, the R^2 and P values are mentioned on the top of each plot.

3.3 Supplementary Fig. S3

Log-linear regression of the unigram and bigram entropy with various genomic parameters in prokaryotes. Each point is a genome. Columns indicate kingdoms or phyla. From left to right, they are Proteobacteria, Crenarchaeota, and Euryarchaeota. Each row is a specific parameter. From top to bottom they are: unique domain type in the genome **(A to C)**, total domain count **(D to F)**, number of proteins **(G to I)**, and the total count of amino acids **(J to L)** in the genome. In each figure the X-axis is converted to log10 scale. The slope of the regression lines, the R^2 and P values are mentioned on the top of each plot.

3.4 Supplementary Fig. S4

Log-linear regression of the unigram and bigram entropy with various genomic parameters in eukaryotes. Each point is a genome. Columns indicate kingdoms or phyla. From left to right, they are green plants (Viridiplantae), Fungi, and animals (Metazoa). Each row is a specific parameter. From top to bottom they are: unique domain type

in the genome **(A to C)**, total domain count **(D to F)**, number of proteins **(G to I)**, and the total count of amino acids **(J to L)** in the genome. In each figure the X-axis is converted to log10 scale. The slope of the regression lines, the R^2 and P values are mentioned on the top of each plot.

3.5 Supplementary Fig. S5

Distributions of unigram and bigram and relative entropies without using N-C markers (see Methods for details). **(A)** Density plot of entropy values in three superkingdoms. Each panel represents one superkingdom. From top to bottom: eukaryotes, archaea, and bacteria. Each peak is labeled with corresponding value and marked with dotted line. The inverted color boxed labels indicate the overall medians of the corresponding distribution. The median values are marked using solid line. The X-axis represents entropy in bits/domain. **(B)** Box plot of the unigram, bigram and relative entropies in three eukaryotic (green plants, fungi, animals); two archaeal groups (Crenarchaeota and Euryarchaeota); and six bacterial groups (Tenericutes, Acinetobacteria, Bacteroidetes, Firmicutes, Proteobacteria, and Cyanobacteria). Note the dotted horizontal lines representing the universal constant relative entropy. For calculation without N-C markers this is ~7 bits/domain.

3.6 Supplementary Fig. S6

Multiclass bigram feature selection using Sparse Partial Discriminant Analysis (sPLS-DA). Weighted bigram probabilities were calculated from each species and sPLS-DA were carried out using multiple classes, each representing a single phylogenetic group, as outcome vector. The analyses were carried out in a nested hierarchical manner, beginning with the all the kingdoms **(A)**. In the subsequent rounds, analyses were carried out on following kingdoms and subdivisions: prokaryotes **(B)**; Archaea **(C)**; eukaryotes **(D)**; and Opisthokonta **(E)**. For each analysis, biplot with component 1 on X-axis and component 2 on Y-axis is shown. The ellipses represent 95% confidence area of each cluster.

3.7 Supplementary Fig. S7

Selected bigram features using Sparse Partial Discriminant Analysis (sPLS-DA). For each analysis in (**Fig S6**), the bigram variables were selected for component that showed maximum separation between the supergroups. The barplots show loading values on X-axis for each selected bigram on Y-axis. The color of the bar indicates the classification group shown in the legend. From top to bottom and left to right, the barplots show the selected features and their loading weights for the following classifications: **(A)** All kingdoms. Features selected on component 1 separating eukaryotes and prokaryotes; **(B)** Prokaryotes. Featureas selected on components 1 separating Bacteria and Archaea; **(C)** Archaea. Features selected component 1 separating Crenarchaeota vs Euryarchaeota; **(D)** Eukaryotes. Features selected on component 2 separating Viridiplantae (green plants) and Opisthokonta (Fungi plus animals); **(E)** Opisthokonta. Features selected on component 1 separating Fungi and Metazoa (animals).

Supplementary references

1. Eddy SR (2011) Accelerated Profile HMM Searches. *PLOS Computational Biology* 7(10):e1002195.

2. Finn RD, et al. (2010) The Pfam protein families database. *Nucleic Acids Res* 38(Database issue):D211–222.

3. Jurafsky D, Martin JH (2008) *Speech and Language Processing* (Prentice Hall). 2nd Ed.

4. Basu MK, Poliakov E, Rogozin IB (2009) Domain mobility in proteins: Functional and evolutionary implications. *Brief Bioinform.* doi:10.1093/bib/bbn057.

5. Basu MK, Carmel L, Rogozin IB, Koonin EV (2008) Evolution of protein domain promiscuity in eukaryotes. *Genome Res* 18(3):449–61.

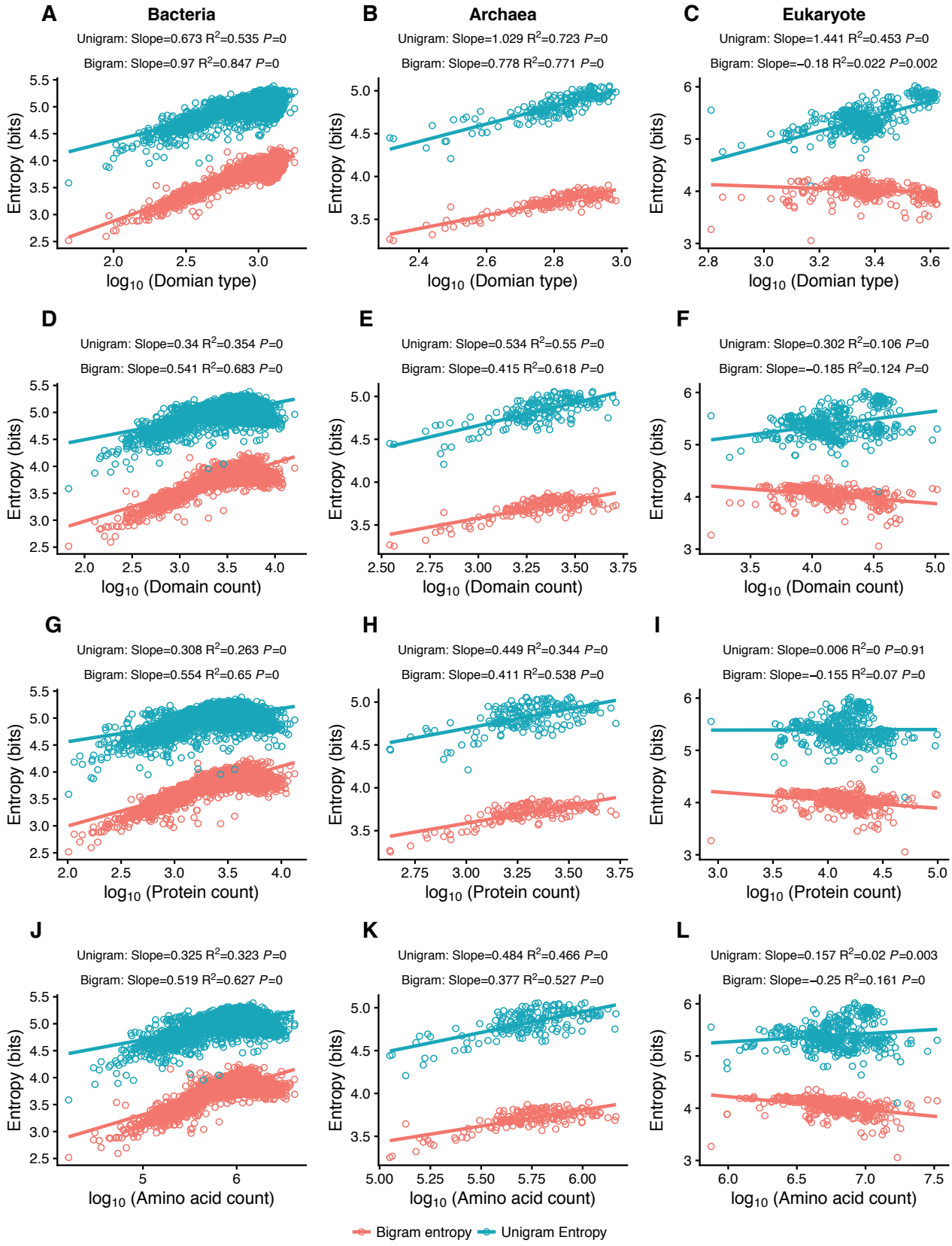
6. Katz SM (1987) Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Trans Acoust Speech Signal Process* 35(3).

7. Kneser R, Ney H (1995) Improved backing-off for m- gram language modeling. *IEEE ICASSP-95* 1:181–184.

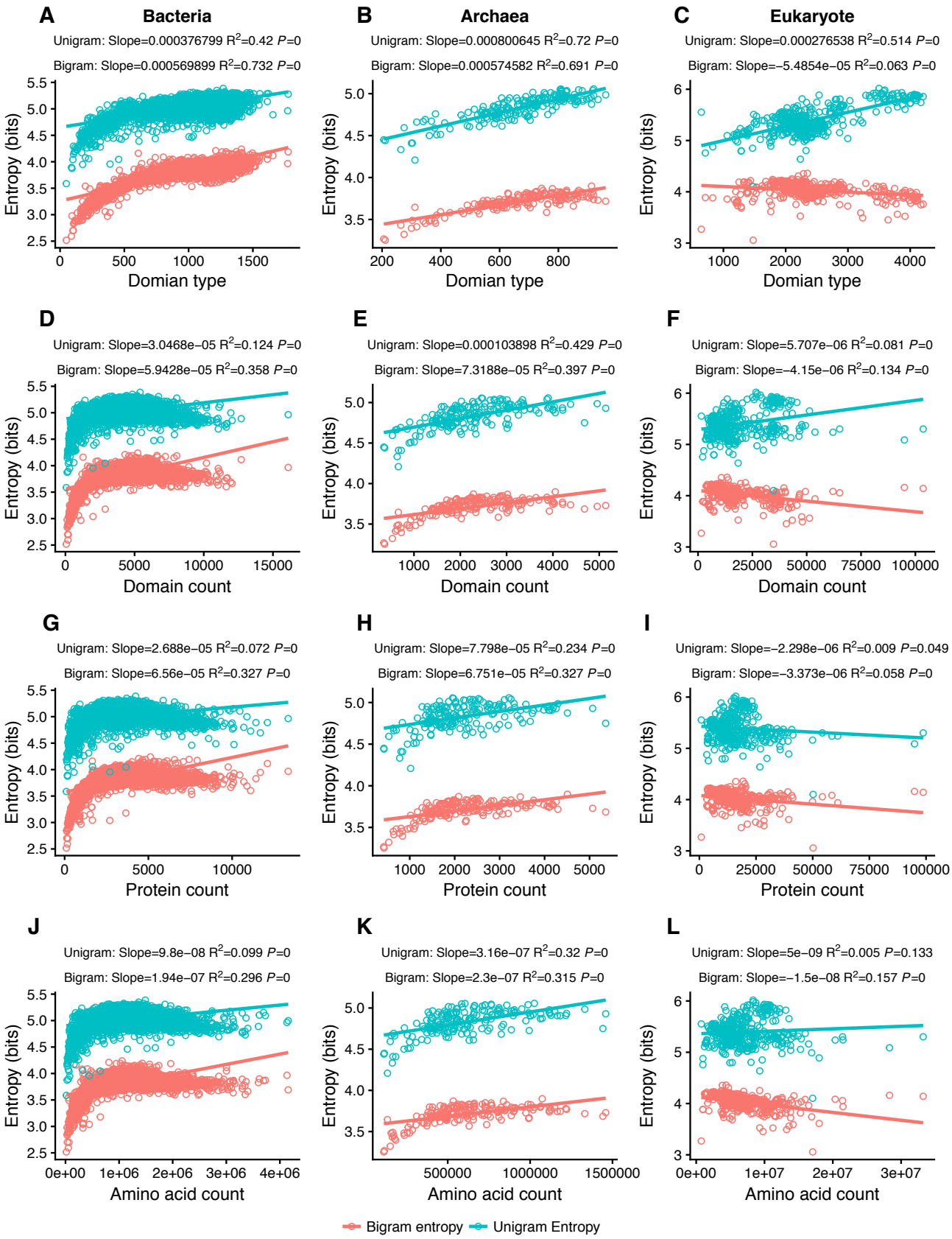
8. Good IJ (1953) The population frequencies of species and the estimation of population parameters. *Biometrika* 40(3):237–264.

9. Gale WA, Sampson G (1995) Good-turing frequency estimation without tears. *J Quant Linguist* 2(3):217–237.

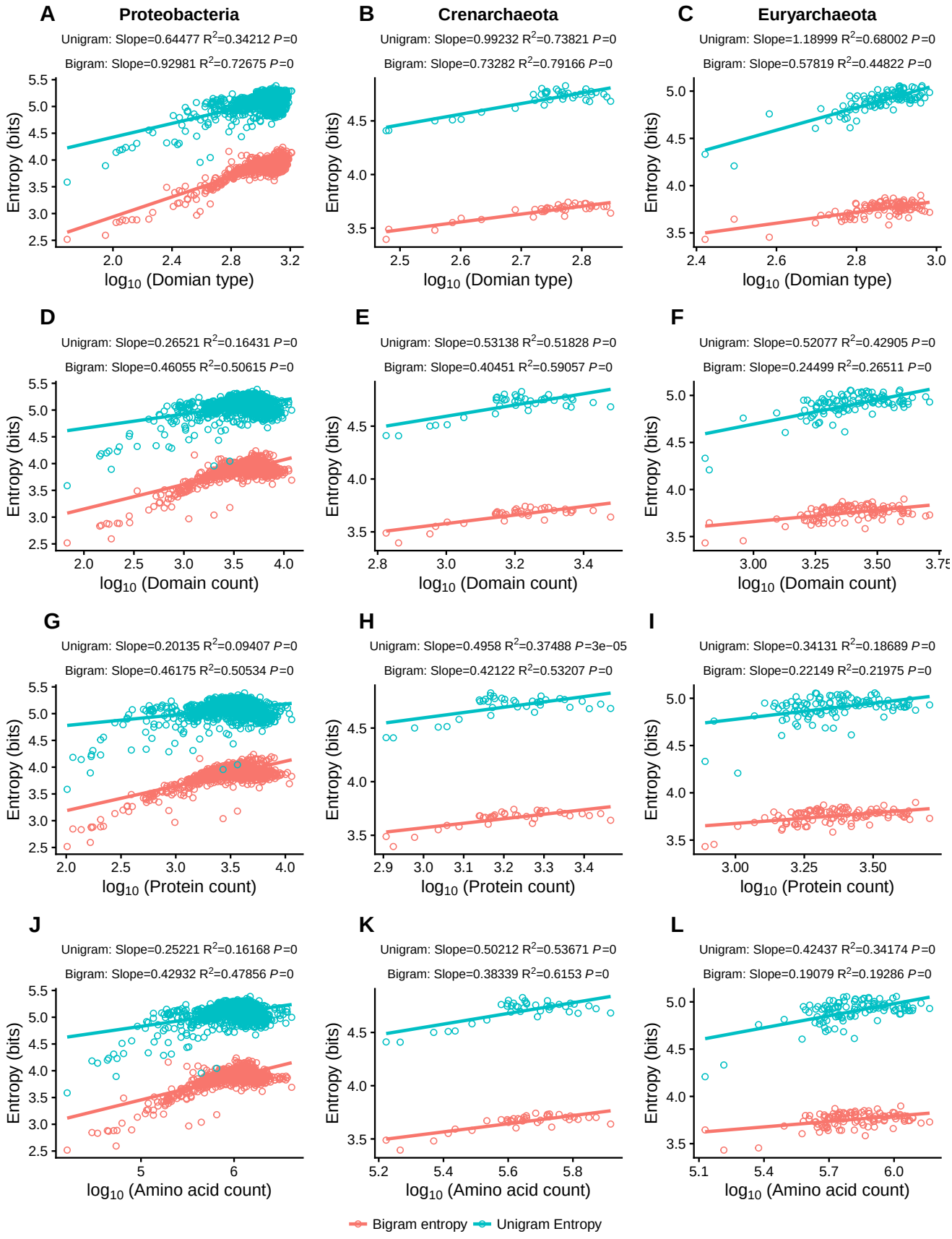
10. Bhattacharyya A (1943) On a measure of divergence between two statistical populations defined by their probability distributions. *Bull Calcutta Math Soc* 35:99–109.



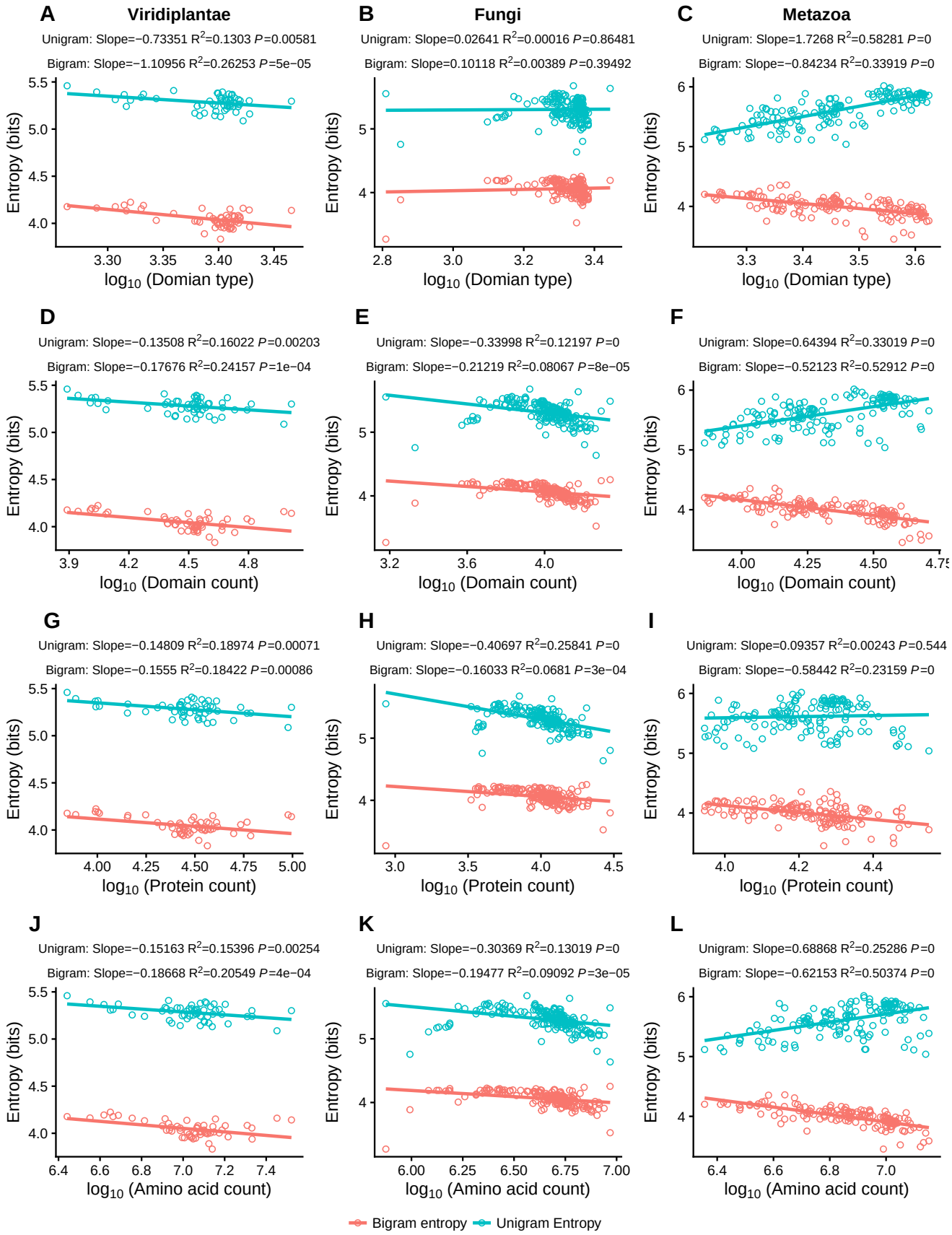
Supplementary Fig. S1



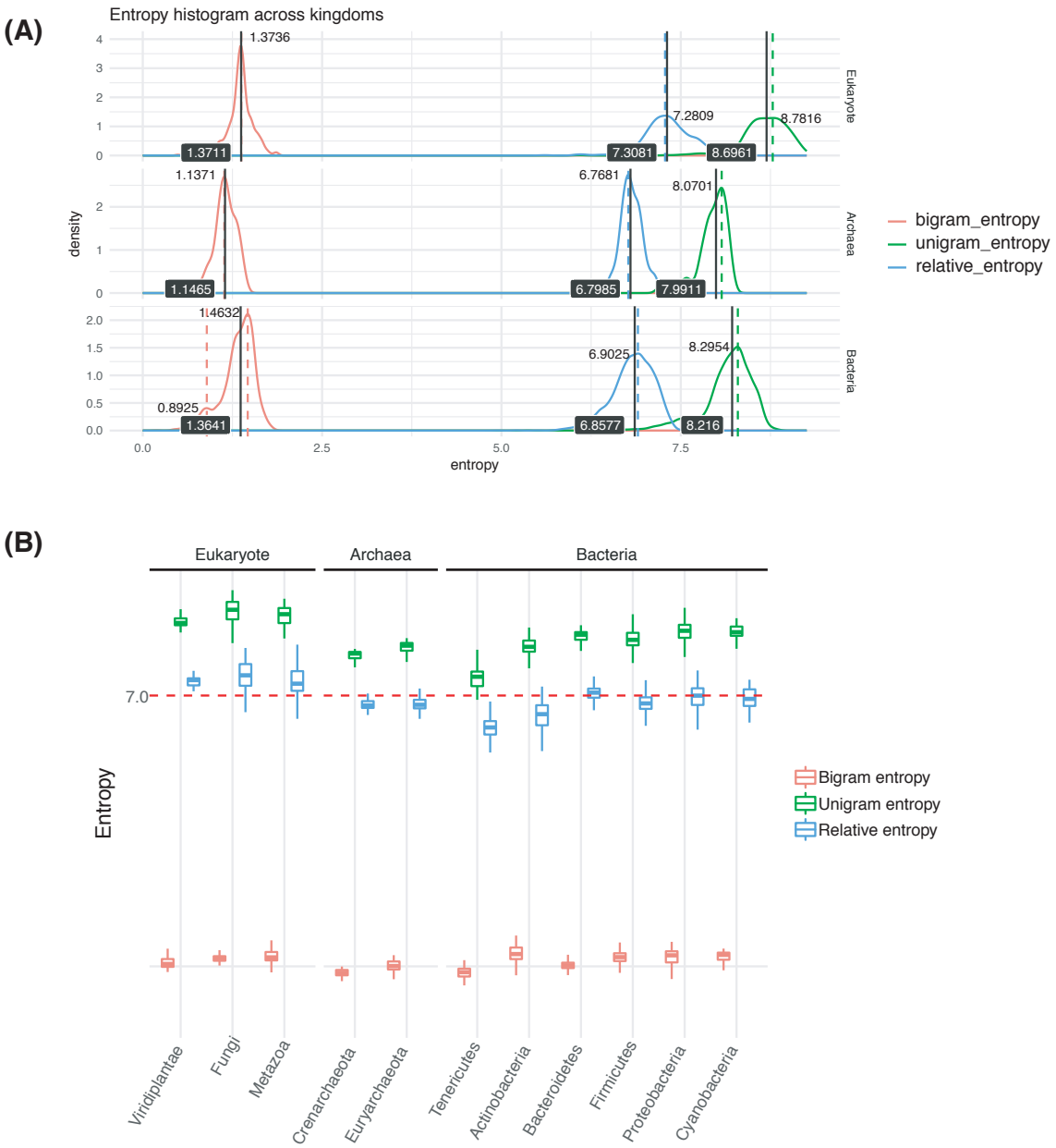
Supplementary Fig. S2



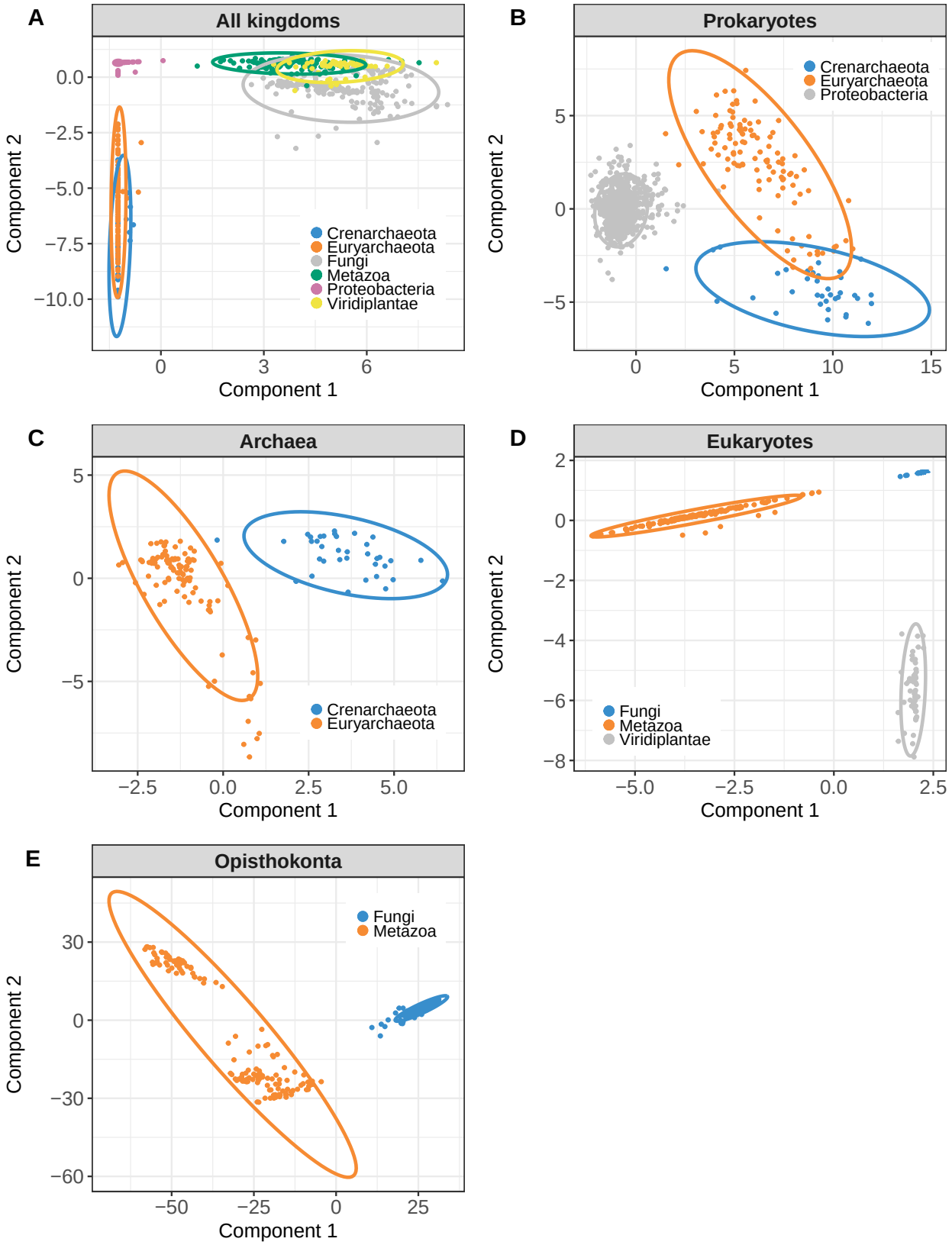
Supplementary Fig. S3



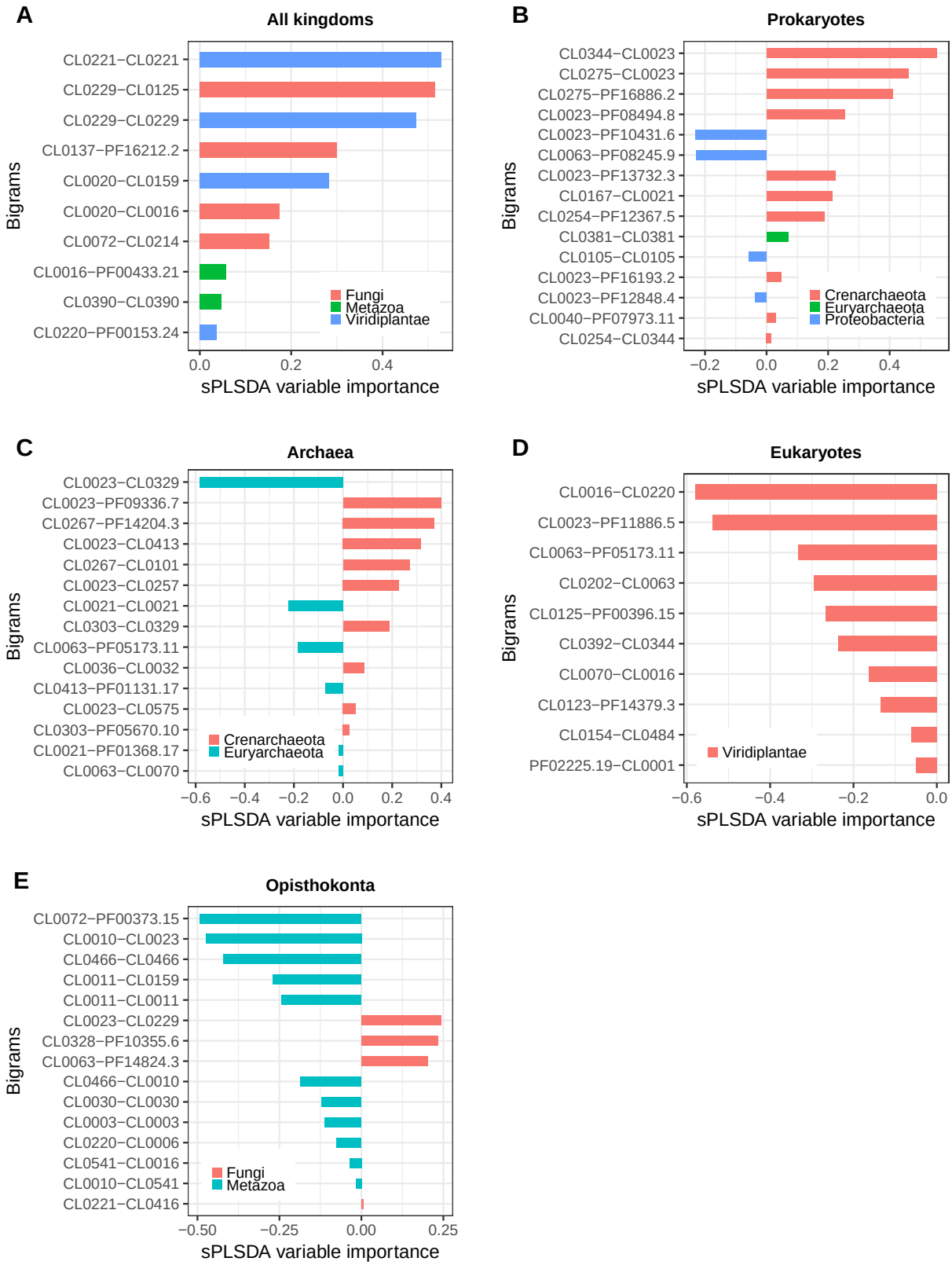
Supplementary Fig. S4



Supplementary Fig. S5



Supplementary Fig. S6



Supplementary Fig. S7