Wai Yee Low

Population entropies estimates of proteins

Proceedings of the 3rd ISM International Statistical Conference 2016: Bringing Professionalism and Prestige in Statistics, as published in AIP Conference Proceedings, 2017 / vol.1842, iss.1, pp.020004-1-020004-7

Corrigendum: “Population Entropies Estimates of Proteins,” Wai Yee Low, AIP Conf. Proc. **1842**, 020004 (2017)

This article, which was originally published online on 12 May 2017, contained an error in Eq. (1), where the summation sign was missing. The corrected equation appears below.

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 p_i \qquad (1)$$

Population Entropies Estimates of Proteins

Wai Yee Low[1, a)]

[1]*Centre for Bioinformatics, Perdana University,*
Jalan MAEPS Perdana, 43400 Serdang, Selangor, Malaysia

[a)]Corresponding author: lloydlow@perdanauniversity.edu.my

Abstract. The Shannon entropy equation provides a way to estimate variability of amino acids sequences in a multiple sequence alignment of proteins. Knowledge of protein variability is useful in many areas such as vaccine design, identification of antibody binding sites, and exploration of protein 3D structural properties. In cases where the population entropies of a protein are of interest but only a small sample size can be obtained, a method based on linear regression and random subsampling can be used to estimate the population entropy. This method is useful for comparisons of entropies where the actual sequence counts differ and thus, correction for alignment size bias is needed. In the current work, an R based package named EntropyCorrect that enables estimation of population entropy is presented and an empirical study on how well this new algorithm performs on simulated dataset of various combinations of population and sample sizes is discussed. The package is available at https://github.com/lloydlow/EntropyCorrect

INTRODUCTION

Shannon entropy [1] is essentially a measure of uncertainty in a data set and in the field of bioinformatics, it has been successfully used to quantify protein variability. High entropy of a particular site in a protein alignment corresponds to high uncertainty and hence a lower predictability of the actual amino acid type. For a given multiple protein sequence alignment, the Shannon entropy (H) per alignment column number is calculated as

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 p_i \quad (1)$$

where H is the entropy value,
X is the alignment column number,
N is the number of amino acid types and a gap character (21)*,
p_i is the frequency of i^{th} character in N
*note that it is optional whether a gap is included as a character.

The entropy, H is an interesting measure of variability because it takes into account both the number of different amino acid types and their frequencies. In a 21 characters system, H ranges from 0 (only single character per alignment column) to 4.392 (equiprobability of all 21 characters). Although arbitrary, some researchers consider $H \leq 2$ as conserved and values of $H > 2$ as variable [2].

The literature in biology is replete with example uses of entropy calculation such as vaccine design [3], identification of antibody binding sites [4] and exploration of 3D protein structural properties [5],[6]. To make the entropy calculations easier, various software tools can be used for this purpose such as Sequence variability server (http://imed.med.ucm.es/Tools/svs_help.html), HIV sequence database (http://www.hiv.lanl.gov/),

The 3rd ISM International Statistical Conference 2016 (ISM-III)
AIP Conf. Proc. 1842, 020004-1–020004-7; doi: 10.1063/1.4982834
Published by AIP Publishing. 978-0-7354-1512-6/$30.00

BioPhysConnectoR (https://cran.r-project.org/web/packages/BioPhysConnectoR/), H-BloX [7], CoeViz [8] and AVANA [9].

The usual way to begin analysis on the entropies of a sample of protein sequences starts with an alignment. This process can be achieved by using software such as CLUSTALW [10], CLUSTALOMEGA [11] and MUSCLE [12]. After alignment, users may wish to manually edit it and remove gaps, which can be a tedious process that is prone to human errors. The estimated entropies on the protein alignment belong to a sample but often times the population entropies are of interest. For example, if a vaccine is designed to target conserved sites estimated from a small random sample of viruses, it will be of interest to know if the sites are also likely conserved at the population level. The relationship of the protein variability estimates from a sample and how it relates to the population variability parameter is not well understood.

An example use of population entropies estimates is the correction of alignment size bias in calculation of entropies. This problem is due to the presence of truncated protein sequences (i.e. not all of the protein sequences are of the same length) and thus in the protein alignment, some sites have a higher sequence count whereas others have less. Additionally, researchers may also wish to compare variability of proteins that are homologous but exists as separate alignment files with different sequence counts in each file. In these cases, a method that takes care of alignment size bias allows for a better comparison of the estimated protein variability.

A potential solution was well explained in a paper on vaccine design [3], where essentially the method relies on extrapolation to infinite size population entropy value obtained from a linear regression model [13]. The current work considers the infinite size population entropies estimates so that it can be used to address alignment size bias issue. Such population entropies estimates are called as entropy correction in this work. An empirical study was also conducted on simulated populations and samples to compare how well the corrected entropy performed versus uncorrected entropy (i.e. straight from equation 1).

METHODOLOGY

An R package named EntropyCorrect was designed to address the problem of alignment size bias, which essentially requires the use of entropy estimated from an infinite size population. In order to make it clear to the users how this method works, first the step-by-step entropy calculation for a sample sequences (test.fasta) is illustrated in Table 1(the file is attached together with the package). The concept of how entropy is calculated when the window size of interest is not the typical monomer but rather other sizes, such as a nonamer that is used in the field of immunology, is also given. This feature that allows entropy calculation for varying window sizes in a protein alignment is available in EntropyCorrect but many software that perform similar calculation do not have it and only focused on a window size of 1.

Implementation

Now that it is clear how entropy is calculated from a protein alignment, the next feature to implement is entropy correction, which is illustrated in Fig. 1. The method requires at least 10 sequences in the alignment file for random subsample collection and entropy calculation. The random subsampling without replacement occurs for each n = 10 sequences, until it reaches the maximum number of sequences. Therefore, for a protein alignment with 20 sequences, there will be 11 samples collected, each with a sample size that corresponds to 10, 11, 12, …, 20. For each of the collected sample, an entropy is calculated as per equation 1. Then a linear model is fitted using the lm() function in R with entropy as the dependent variable, 1/n as the independent variable, and the intercept value is retrieved, which corresponds to the estimate where $n \rightarrow \infty$. Therefore, the retrieved intercept value is the entropy estimate when the population is of infinite size, which for this paper is considered as the corrected entropy value.

Table 1. A step-by-step calculation of entropy from an alignment of 10 sequences.

A) A sample protein alignment

Sequences	Alignment position								
	1	2	3	4	5	6	7	8	9
A	L	K	N	I	D	Y	E	T	V
B	L	K	G	I	D	Y	E	T	V
C	L	K	N	I	D	Y	E	I	V
D	L	K	G	I	D	Y	E	T	V
E	L	K	S	I	D	Y	E	T	I
F	L	K	G	I	D	Y	E	I	V
G	L	K	G	I	D	Y	E	T	V
H	L	K	G	I	D	Y	E	T	V
I	L	K	G	I	D	Y	E	T	V
J	L	K	G	I	D	Y	E	T	V

B) Example entropy calculation for a monomer (i.e. window size, n = 1)

Alignment position	i	Unique monomer	Count	p_i	H(X)
1	1	L	10	1	0
2	1	K	10	1	0
3	1	N	2	0.2	1.157
	2	G	7	0.7	
	3	S	1	0.1	

C) Example entropy calculation for a nonamer (i.e. window size, n = 9)

Alignment position	i	Unique nonamer	Count	p_i	H(X)
1-9	1	LKNIDYETV	1	0.1	1.771
	2	LKGIDYETV	6	0.6	
	3	LKNIDYEIV	1	0.1	
	4	LKSIDYETI	1	0.1	
	5	LKGIDYEIV	1	0.1	

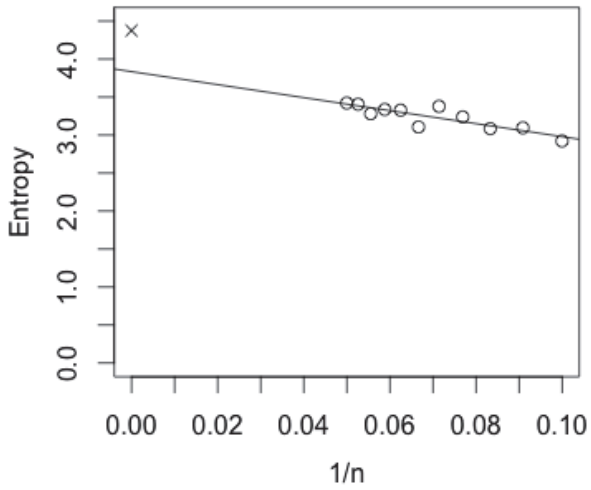

FIGURE 1. An illustration of the concept behind entropy correction. The population is set at a size of 1000 and the individual amino acids are sampled with equiprobability. Since all members of the population are known, the population entropy can be calculated, which is represented by the 'x' mark on the figure. From this population, a sample size of 20 is drawn. Starting from n = 10, subsamples are drawn until it reaches the sample size of 20 in this case. From each of the subsample, entropy is calculated. A linear regression model is then applied for entropy against 1/n.

There are other features implemented in the package that helps take care of some of the common manipulation to the protein alignment such as gap removal, setting a reference sequence and writing the changed alignment file out in FASTA format. The full arguments accepted by the package can be found by running ?EntropyMSA on the R console after installation.

Empirical study on simulated dataset

Researchers usually only estimate protein variability from a sample, which is likely much smaller in size when compared to the actual population. In order to test how well corrected entropy performed relative to the uncorrected version, population of various sizes at different entropy levels were selected for an empirical investigation. Population sizes of 1000, 10000, 100000, 1000000, and 10000000 were simulated. For each population size, four different types of populations each with entropies of H = 1.584, H = 2.578, H = 3.575, and 4.375 were created. The entropies were chosen to reflect populations with different variability. Then for each population, five different random sample sizes of 20, 40, 60, 80 and 100 were chosen. From these samples, the uncorrected and corrected entropies were calculated.

RESULTS

The result of entropy correction versus uncorrected is summarized as boxplots in Fig. 2. From this figure, it could be observed that the corrected entropies were closer to the real population entropies parameters than the uncorrected entropies. This was true for almost all sample sizes examined that ranged from 20 to 100. When the true population entropy was low, which in terms of the protein alignment was reflected by the presence of only a few amino acid types, the difference between corrected and uncorrected entropies was negligible. However, if the population entropies were high, corrected entropies were closer to the true population entropies because without correction, small sample sizes tend to underestimate the true level of population variability.

A real example on the use of EntropyCorrect package is given in Fig. 3. The full code to regenerate the figure is given in the package itself under the example usage section, which allows future users to generate similar plots. The protein alignment presented here is an example from mammalian glutathione S-tranferase omega class.

FIGURE 2. There were five population sizes simulated ranging from a thousand to ten millions. For each population size, there were four different entropies level: low entropy, moderate entropy, high entropy, and very high entropy. From each population, five different random samples of sizes 20, 40, 60, 80 and 100 were drawn and their corrected and uncorrected entropies were calculated. The solid lines in each of the four panels are the population entropies.

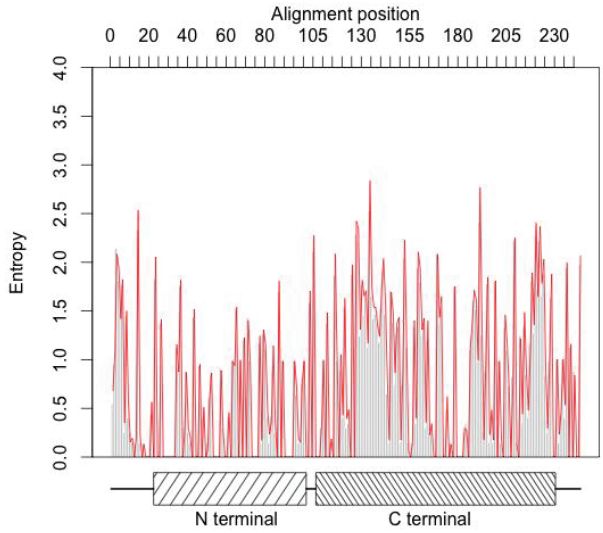

FIGURE 3. An example use of the EntropyCorrect package. The entropy across a protein alignment with a reference sequence human GSTO1. The uncorrected entropy values are in grey bars whereas the corrected values are in red lines.

DISCUSSION

If the aim of calculating entropy from a protein alignment is to get closer to the population variability parameter, from the results of simulated populations of various entropies and sample sizes, it would be better to use the corrected entropy values. Given that users may also be working with protein alignments that contain truncated sequences, correction for alignment size bias is needed and thus using entropy values that are all infinite population size estimates are better. This study considered populations that ranged from a thousand to ten millions but some populations are much larger than this range. However, if the generality of the results obtained here could be extrapolated to much larger population sizes, such as those in billions, the expectation is uncorrected entropies will still tend to underestimate variability. Therefore, the corrected entropies should estimate the population variability much better in these cases. An interesting application of the presented method is to estimate the human proteins variability accurately especially given that the population size of humans is known. In this case, the actual code of the software needs to be manipulated to report the entropy when the intercept of the linear regression line matches 1/N, where N is the known population size. From here, the minimum random sample size that is required to estimate human proteins variability accurately can be predicted.

The inception of this software package was motivated by the fact that R provides the necessary environment to calculate entropies as well as produce publication ready figures. Besides, R is cross platforms and there are other packages such as msa that can handle the protein alignment step [14]. Altogether, these packages enable protein variability investigation without the need to switch software. Additionally, given the powerful features of base R and other additional packages, the users can just focus on learning one software environment well.

The current implantation of EntropyCorrect allows the users to select their own window size in the alignment for entropy calculation. This feature is useful for researchers such as those who find nonamers of viral proteins potentially better represent the surface peptides that actually interact with Human Leukocyte Antigen and T Cell Receptor core binding domains [3]. With the selection of a window size larger than one, the *H(X)* in equation 1 will change because p_i is no longer based on just the monomer. For example, in a nonamer setting, the maximum for H(X) is $\log_2(20^9)$ if we assume the use of only 20 amino acids.

Future version release of the package will include additional topics on entropy as well as adding features requested by users. For example, although the entropy method is straightforward at estimating variability, it is oblivious to incorporating evolutionary information such as some residues are more likely to mutate and thus should be emphasized if such residues are seen as conserved. To address this problem, joint entropy with BLOSUM-proportional probabilities was proposed [15] and perhaps this is a good additional feature for future version.

CONCLUSION

EntropyCorrect is an R package that performs both uncorrected and corrected entropy calculations. In addition, it handles gap removal, an option to include reference sequence, and window size setting. Results from simulation of various population and sample sizes showed the corrected entropies are closer to the population entropies than uncorrected ones.

AVAILABILITY AND REQUIREMENTS

Project name: EntropyCorrect
Project home page: https://github.com/lloydlow/EntropyCorrect
Operating system(s): Platform independent
Programming language: R
Other requirements: R 3.2.2 or higher
License: GNU GPL v. 2.
Any restrictions to use by non-academics: none

ACKNOWLEDGMENTS

The author is grateful to Dr. Mohammad Asif Khan and his lab members for sharing ideas on entropy correction.

REFERENCES

1. C. Shannon , A mathematical theory of communication. Bell Syst Tech J **27**, 379–423, 623–656 (1948).
2. H-H. Bui , J. Botten, N. Fusseder, V. Pasquetto, B. Mothe, M. J. Buchmeier, et al., Protein sequence database for pathogenic arenaviruses. Immunome Res. BioMed Central **3**(1), 2007.
3. A. M. Khan, O. Miotto, E. J. M. Nascimento, K. N. Srinivasan, A. T. Heiny, G. L. Zhang et al., Conservation and variability of dengue virus proteins: implications for vaccine design. PLoS Negl Trop Dis **2**(8), 272 (2008).
4. K. Pan and M. W. Deem, Quantifying selection and diversity in viruses by entropy methods, with application to the haemagglutinin of H3N2 influenza. J R Soc Interface **8**(64), 1644–1653 (2011).
5. H. Liao, W. Yeh, D. Chiang, R. L. Jernigan and B. Lustig, Protein sequence entropy is closely related to packing density and hydrophobicity. Protein Eng Des Sel **18**(2), 59–64 (2005).
6. R. P. Bywater , Prediction of protein structural features from sequence data based on Shannon entropy and Kolmogorov complexity. PLoS One **10**(4), e0119306 (2015).
7. J. Zuegge, M. Ebeling and G. Schneider, H-BloX: visualizing alignment block entropies. J Mol Graph Model **19**(3), 304–6 (2001).
8. F. N. Baker and A. Porollo, CoeViz: a web-based tool for coevolution analysis of protein residues. BMC Bioinformatics **17**, 119 (2016).
9. O. Miotto, A. Heiny, T. Tan, J. T. August, V. Brusic, S Baigent, et al., Identification of human-to-human transmissibility factors in PB2 proteins of influenza A by large-scale mutual information analysis. BMC Bioinformatics **9** (Suppl 1):S18 (2008)
10. M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, et al., Clustal W and Clustal X version 2.0. Bioinformatics. Oxford University Press **23** (21), 2947–2948 (2007).
11. F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, et al., Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol. EMBO Press **7**(1), 539 (2011).
12. R. C. Edgar, MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. Oxford University Press **32**(5), 1792–1797(2004).
13. L. Paninski, Estimation of Entropy and Mutual Information. Neural Comput. MIT Press **15**(6), 1191–253 (2003)
14. U. Bodenhofer, E. Bonatesta, C. Horejš-Kainrath and S. Hochreiter, msa: an R package for multiple sequence alignment. Bioinformatics. Oxford University Press **31**(24), 3997–3999 (2015).
15. I . Mihalek, I. Res, O. Lichtarge, C. Shannon, W. Weaver, P. Shenkin, et al., Background frequencies for residue variability estimates: BLOSUM revisited. BMC Bioinformatics **8**(1), 488 (2007).