The Open Bioinformatics Journal
DOI: 10.2174/0118750362484360260531193931, 2026, 19, e18750362484360

ISSN: 1875-0362
1

RESEARCH ARTICLE **OPEN ACCESS**

MetRef, an Auto-updatable Database for Reference/representative Genomes and their Complexity Metrics

Pablo Román-Escrivá[1,*], Vicente Arnau[1,2,3], Wladimiro Díaz-Villanueva[1,2,3] and Andrés Moya[1,2,3]

[1]*Institute for Integrative Systems Biology (I2SysBio), Spanish National Research Council (CSIC), University of València, 46980 València, Spain*
[2]*Biomedical Research Networking Center for Epidemiology and Public Health (CIBERESP), 28029 Madrid, Spain*
[3]*Area of Genomics and Health, Foundation for the Promotion of Sanitary and Biomedical Research of València Region (FISABIO), 46020 València, Spain*

Abstract:

Introduction: The use of genomic metrics to evaluate biological complexity is an important area of research in evolutionary biology; however, this area of study has been hindered by the need for high levels of computation and the lack of a uniform method of performing genomic complexity analyses. This research presents an auto-updating web-based application called MetRef (Metrics of Reference Genomes) that provides an interface to standardize and display genomic complexity analyses.

Methods: MetRef uses two key metrics: Genomic Signature (GS), which uses the normalized frequency of k-mers, and Biobit (BB), which is a combination of both the entropic and anti-entropic components. In order to reduce the computational overhead associated with k-mer counting, MetRef includes the specialized tool KMC 3. Additionally, the MetRef application utilizes a multi-layered architectural framework that incorporates Django, PostgreSQL, and Celery/Redis for managing high-quality assemblies from the NCBI RefSeq and Taxonomy databases.

Results: Using KMC 3 resulted in a reduction in processing time for the human genome from 36 hours to 5 minutes. Users can select comparative studies and receive automatic statistical reports that include Pearson correlation matrices, Principal Component Analysis (PCA), and ANOVA testing, with color-blind-friendly palettes.

Discussion: The advantages of MetRef to address two major limitations in genomic complexity research (computational efficiency and lack of uniform procedures) lie within its use of a special-purpose C++ counting engine (KMC 3) for calculating genetic complexities, and through the unification of calculation pipelines. As such, the MetRef removes inconsistencies generated from varying treatments of non-ACGT bases; therefore, all genetic complexities calculated with the same metrics can be compared directly across the entire tree of life.

Conclusion: In addition to these technical advantages, MetRef establishes a solid, scalable base for evolutionary genomics through an architecturally stable structure using Django and Celery / Redis. Furthermore, the system reduces the researcher's workload, providing them with automatically generated statistical reports and previously calculated, versioned data; thus enabling researchers to easily design and perform additional large-scale comparative studies of genetic complexities.

Keywords: Genomic complexity, Genomic signature, Biobit, Evolutionary genomics, Information theory, Database.

© 2026 The Author(s). Published by Bentham Open.
This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International Public License (CC-BY 4.0), a copy of which is available at: https://creativecommons.org/licenses/by/4.0/legalcode. This license permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

*Address correspondence to this author at the Institute for Integrative Systems Biology (I2SysBio), Spanish National Research Council (CSIC), University of València, 46980 València, Spain; E-mail: pablo.roman@uv.es

Cite as: Román-Escrivá P, Arnau V, Díaz-Villanueva W, Moya A. MetRef, an Auto-updatable Database for Reference/representative Genomes and their Complexity Metrics. Open Bioinform J, 2026; 19: e18750362484360. http://dx.doi.org/10.2174/0118750362484360260531193931

Received: February 04, 2026
Revised: April 07, 2026
Accepted: April 24, 2026
Published: November 13, 2026

Send Orders for Reprints to reprints@benthamscience.net

1. INTRODUCTION

The problem of understanding the complex nature of biological complexity from an evolutionary perspective is one of the most important and difficult issues in contemporary biology. The multi-dimensional nature of biological complexity makes it difficult to define and measure complexity, which leads to the need for measurable surrogates of complexity, *i.e.*, genomic complexity. A useful way to assess the complexity of the genome is to view the genome as a message. In this regard, the complexity of a genome can be assessed using the principles of information theory, with the benefit of being able to numerically quantify the complexity of the genome and thereby analyze genomic complexity as a component of the broader biological complexity of species, providing utility in evolutionary studies across species or in inclusive taxonomic groups [1-5].

As a result of this area of study, there have been several metrics developed to quantify the genomic complexity of DNA sequences. Specifically, two of the most commonly used metrics in evolutionary genomics research are Genomic Signature (GS) and Biobit (BB).

GS is a metric of genomic complexity based on the content of k-mers (sub-sequences of length k) within a given genome [6]. BB is conceptualized as a logistical map that balances the entropic and anti-entropic components of a genome [7, 8].

These metrics have been successfully applied to analyze shifts in complexity in specific biological contexts, as in our prior work investigating the evolutionary impact of endosymbiosis on genome complexity [9].

Despite the theoretical value of these metrics [10], their practical application faces two primary challenges. First, the core step of k-mer frequency calculation is computationally intensive for large eukaryotic genomes, which is identified as the major bottleneck. Second, the lack of a complete, published implementation leads to discrepancies in results across research groups due to varied handling of procedural steps (*e.g.*, treatment of non-ACGT bases or the random genome generation process), resulting in non-comparable values.

To address these limitations, this work pursued two main objectives. First, to achieve a robust and efficient calculation pipeline for GS and BB. Second, and most importantly, to develop MetRef (Metrics of Reference Genomes), a centralized, standardized, and constantly updated web application for the management, analysis, and visualization of these complexity metrics alongside key genomic parameters. MetRef unifies metric implementation and curates reference data, providing a ready-to-use platform that streamlines the research flow.

2. MATERIALS AND METHODS

2.1. Genomic Complexity Metrics

GS quantifies the difference between the observed frequency of a k-mer and the expected frequency for a randomly generated genome of the same size. For an alphabet of four characters (DNA, ACGT), and a word size of k characters, there is a maximum of 4^k possible k-mers. The value of GS for a given k (GS_k) is calculated using genomic signature Eq. (**1**) as follows:

$$GS_k = \frac{1}{\sum_{j=1}^{4^k} n_j} \cdot \sum_{i=1}^{4^k} |n_i - EV| \quad \textbf{(1)}$$

Where n_i is the observed number of k-mers for a specific sequence, and EV (Expected Value) is used to center the value on the mean, calculated as $EV = \sum_{j=1}^{4^k} \frac{n_j}{4^k}$. The final value is normalized by $\sum_{j=1}^{4^k} \cdot n_j$ To enable comparison between genomes of different lengths. To find the optimal GS value for a genome, a random genome (GS_r) of the same size as the study genome (GS_g) is generated, and a preliminary score ($GS_p = GS_g - GS_r$) is calculated starting at $k = 2$. The procedure is repeated, incrementing k up to 16, at which point the local maxima of GS_p are identified and selected.

BB compares a given genome G against an equifrequent random genome of the same length. The metric combines the entropic component ($E(G)$) and the anti-entropic component ($A(G)$) of the genome, such that E(G) + A(G) = L(G) (where $L(G) = \log_4(G)$). The Biobit score $BB(G)$ is calculated non-linearly using Biobit Eq. (**2**) as:

$$BB(G) = \sqrt{L(G)} \sqrt{\frac{A(G)}{L(G)}} \left(1 - 2\frac{A(G)}{L(G)}\right)^3 \quad \textbf{(2)}$$

The development of MetRef addressed two core technical requirements: achieving efficiency in the base metric calculation and establishing a robust, web-accessible system for data management and analysis.

2.2. Metric Calculation Efficiency

The major computational bottleneck in calculating both GS and BB was the generation and storage of the k-mer frequency tables. Initial optimization attempts using standard Python and Bash proved unsuccessful, confirming the need for a specialized solution.

The successful solution was to integrate the external, specialized tool: KMC 3 [11, 12]. KMC 3 was chosen based on comparative performance and its ability to guarantee exact k-mer counts, unlike other faster tools that provide approximations. KMC 3 uses a RADIX-sort-based algorithm to fragment the input genome into bins for highly parallelized processing [13].

2.3. MetRef System Architecture

MetRef is a client-server web application implemented with the Django framework [14]. The system's organization is designed for robustness and high concurrency [15, 16].

- Front-end Access: NGINX [17] acts as a reverse proxy, receiving external requests and forwarding them to the application [18].
- Application Server: Gunicorn [19] serves as the WSGI connector, linking NGINX to the Python-based Django application.

- Data Persistence: PostgreSQL [20] is a relational object database, chosen for its efficiency in complex data analysis and its use of MVCC (Multi-Version Concurrency Control) for efficient concurrency management.

2.4. Asynchronous Processing

A critical feature of MetRef is the use of asynchronous task management.

Task Management: Celery [21] manages computationally intensive background operations. This includes the crucial biannual database updates and the asynchronous generation of large genome lists, which would otherwise trigger a timeout for the user.

Broker/Cache: Redis [22] is utilized as the in-memory data store for application caching and as the message broker for Celery [23].

This complete architecture, as shown in Fig. (**1**), ensures the application is secure, responsive, and capable of efficiently handling large-scale genomic data.

The goal of MetRef is to collect an extensive number of genomic complexity measures and integrate them into a common, structured framework that can be searched easily.

2.5. Data Source and Curation

MetRef uses RefSeq assemblies found at NCBI [24, 25] to populate its database. The system downloads the RefSeq assembly summary file and then applies several filters in order to use only the data needed for genomic complexity analysis:

- Only include genomes that have been classified as “reference genome” or “representative genome”.
- Exclude genomes that have been specifically excluded from RefSeq (typically due to chimera formation, contamination, fragmentation or size abnormalities).
- Include only high-quality assemblies (either “Chromosome” or “Complete Genome”), since metric calculations are extremely sensitive to either sequencing errors or missing information.
- Focus on Archaea, Bacteria, and Eukaryota, while excluding viruses and organisms without a classification.
- Additional fields that are retrieved from the RefSeq summary file and inserted into both the Genome and Traits tables include: accession, category, assembly_level, group, genome_size, gc_percent, total_gene_count, protein_coding_gene_count, and non_coding_gene_count.

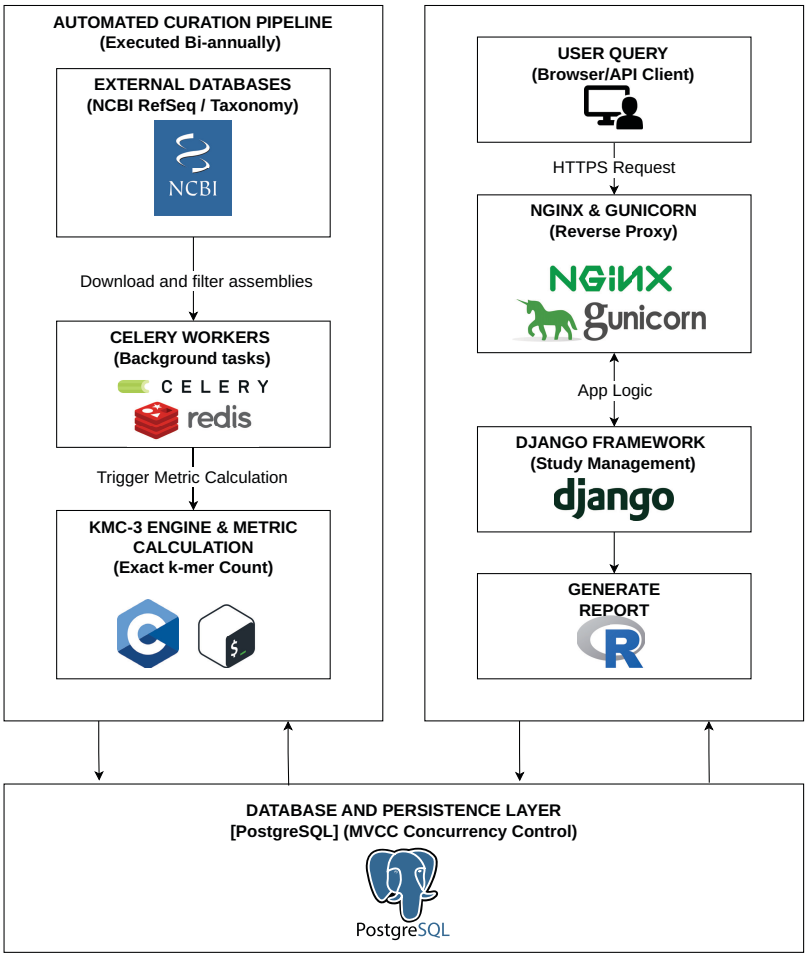


Fig. (1). Complete architecture of *MetRef*.

2.6. Taxonomic Integration and Versioning

In addition to providing users with easy-to-use options to select genomes for analysis, MetRef also links genome data with the NCBI Taxonomy Database [26]. To accomplish this integration, MetRef uses the downloaded and processed taxonomy files: merged.dmp, names.dmp, and nodes.dmp.

During the database loading process, for every genome:

The system checks whether the genome's tax_id exists in the Taxon table.

If the tax_id (and its parent taxa up to the Superkingdom level) does not exist, new Taxon records are created recursively. The Taxon table stores the tax_id, parent_tax_id, name, and rank.

The relational database model is built on several key tables.

- Version: Tracks periodic updates, set at a bi-annual frequency. Maintaining previous versions ensures reproducibility for extending older studies.
- Study: Records user-defined research projects, linking a title, a specific Version, and selected Taxon records to a User.
- GenomeList: The core mapping table. For each selected Taxon, an asynchronous task generates a GenomeList that references all subordinate Genome objects associated with the chosen Version.

The whole relational model is referenced in Fig. (**2**).

3. RESULTS

3.1. Metric Calculation Efficiency

Integrating KMC 3 into the metric pipeline resulted in a massive reduction in processing time and resource usage: the calculation of BB for the human genome decreased from 36 hours (1 CPU, 150 GB RAM) to just 5 minutes (8 CPUs, 40 GB RAM). This optimization ensured efficient metric calculation and allowed the process to handle compressed genome files directly.



Fig. (2). Relational database model of *MetRef*.

3.2. Study Management

The application interface allows users to create a new Study *via* a form, specifying a title, a Version, and selecting at least one Taxon. The system enforces that the chosen taxa cannot be parent or descendant taxa of each other, ensuring they represent independent comparative groups. Due to the potential size of the resulting genome list, its generation is handled by a Celery asynchronous task. If the list takes time to populate, a temporary waiting message is shown to the user until the list status is updated.

MetRef provides two core outputs to the user: direct access to the raw, pre-calculated data and an automated statistical analysis report.

3.3. Data Availability and Access

MetRef is designed to provide a Traits object with the pre-computed metrics and genomic parameters for each genome in a study defined by the user. The traits object contains the pre-calculated metrics and parameters for all selected genomes as a downloadable csv file. These parameters include:

Complexity Metrics: GS (Genomic Signature) and BB (Biobit), plus the k value used to calculate them (k_{GS} and k_{BB1}).

Genomic Parameters: Size, ACGT base count, GC%, entropy, gene count (total, protein coding and non-protein coding), and the proportion of hapax sequences (a sequence that appears only one time).

MetRef thus addresses the need for an integrated system capable of providing immediate access to pre-calculated, standard values.

3.4. Preliminary Statistical Analysis

MetRef creates a complete PDF report of preliminary statistical analysis on the users' requests. The MetRef reports can be produced by MetRef only when there are at least thirty organisms involved in a Study that has been created by the user; this capability was added because it is a very common requirement to do some form of systematic analysis prior to further investigation of data sets. All of the reports are developed using R scripts.

Correlations: A Pearson correlation matrix is developed to determine the linear relationship between various genomic parameter metrics. The Holm-Bonferroni correction [27] for the Family-Wise Error Rate (FWER) is applied to all correlations reported within the study. A reference to a correlation matrix from one of the reports is provided in Fig. (**3**).

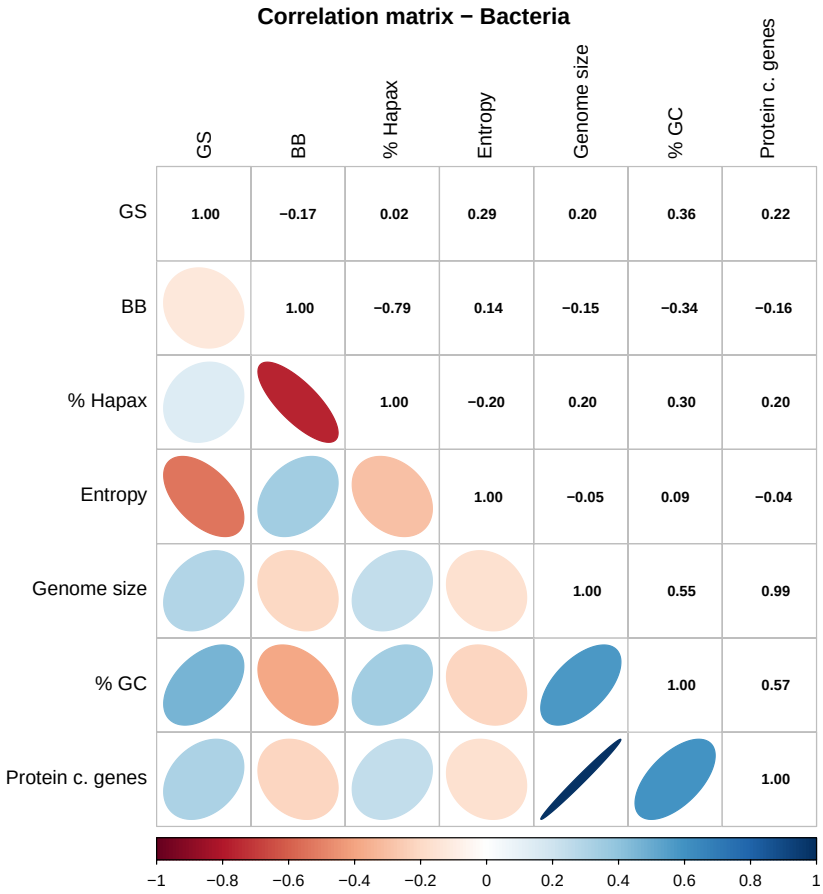


Fig. (3). Correlation matrix extracted from a report.

Data Visualizations (Scatter Plots): Using the ggplot2 [28] R library, scatter plots are used to provide a graphical representation of relationships between pairs of variables. Reference to a scatter plot from one of the reports is provided in Fig. (**4**).

Principal Component Analysis (PCA): If there are seventy or more organisms within the Study, then PCA is conducted. PCA is critical in the reduction of dimensionality and visualization of groups of organisms. Reference to a PCA from one of the reports is provided in Fig. (**5**).



Fig. (4). Scatter plot extracted from a report.



Fig. (5). PCA extracted from a report.

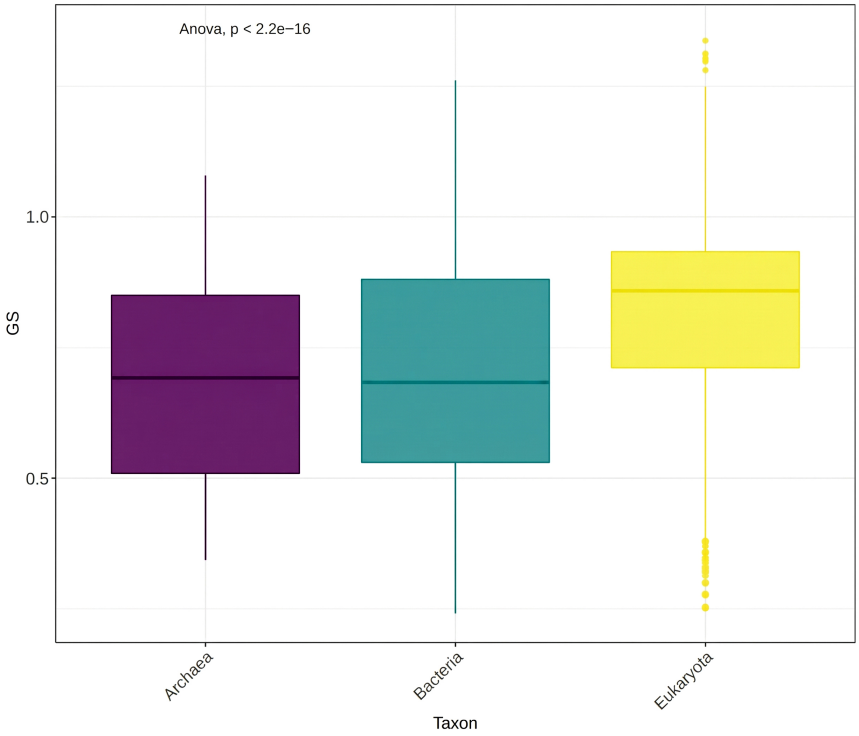


Fig. (6). ANOVA test extracted from a report.

Comparing Means (Box Plots): Box plots are generated to graphically show the distributions of metric values among the selected taxa. When comparing only two taxa, a t-test is used to compare means. If three or more taxa are being compared, then an Analysis of Variance (ANOVA) test is used to compare the means of each taxon. Reference to an ANOVA from one of the reports is provided in Fig. (**6**).

All generated plots will utilize the viridis color palette, which is a color palette designed to assist individuals who have difficulty seeing colors (*i.e.*, color blind), to ensure that all individuals can easily interpret all plots [29].

3.5. System Performance and Scalability

To assess how well our system could handle both a lot of users at once and be responsive to those users, we did stress tests on our site with 100 concurrent users each sending a total of 5,000 ab requests. The average request time for MetRef was 15.32 milliseconds and had a total throughput of 6527.30 requests per second. With no failures, these results showed that the architecture we are currently using (NGINX/Gunicorn) is very robust and will perform well under a high number of concurrent users, making a large amount of requests without an impact on overall performance or service quality.

4. DISCUSSION

MetRef has successfully addressed the two main issues of genomic complexity analysis: 1) Computational inefficiency, and 2) Lack of procedural standards.

4.1. Biological Relevance of Complexity Metrics

Genomic Signature (GS) and Biobit (BB) metrics developed by MetRef can be used to measure the biological relevance of complexity. Genomic Signature (GS) measures how different or dissimilar a k-mer distribution is compared to a completely random distribution, indicating some level of constraint due to evolution and/or structural bias. Biobit (BB) provides an alternate way to assess the degree of non-linearity in a genome's informational capacity by balancing both entropic and anti-entropic contributions. These metrics were previously shown in prior studies of bacterial endosymbiotic events to be responsive to evolutionary changes, including a reduction in genome size and increased randomness of sequences.

4.2. Advantages & Performance

MetRef was optimized through the inclusion of KMC 3, which utilized a RADIX sort-based algorithm to facilitate high levels of parallelism for k-mer counting.

4.3. Standardization and Reproducibility

The use of a web-based interface allows MetRef to ensure that all processed data comes from a standardized pipeline. Therefore, it is possible for users to reproduce results with very high accuracy. The combination of using Django, PostgreSQL, and asynchronous processing capabilities of Celery/Redis enables the system to operate reliably at high concurrency levels, supporting the generation of large-scale comparative reports.

CONCLUSION

MetRef is an integrated, self-automating, and standardized reference platform that facilitates the measurement of genomic complexity. In addition to standardizing the application of the metric (the actual calculation), MetRef also produces automatic statistical reporting of correlation coefficients (Pearson R) and principal component analyses (PCA), as well as ANOVA, which greatly enhances the productivity of the research process for evolutionary biologists.

Currently, MetRef is fully operational and ready for use by both internal and external researchers. The next stage of development for MetRef will be the integration of additional metrics used to measure genomic complexity, as well as enhancements to the user interface with respect to facilitating batch uploads of user-generated genomes.

AUTHORS' CONTRIBUTIONS

The authors confirm contribution to the paper as follows: P.E.: Study conception and design; P.E., V.A and W.V.: Analysis and interpretation of results; P.E.: Draft manuscript; P.E. and A.M.: Review manuscript; A.M.: Supervision. All authors reviewed the results and approved the final version of the manuscript.

LIST OF ABBREVIATIONS

BB	=	Biobit
GS	=	Genomic Signature

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

Not applicable.

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The MetRef web application is freely available for access at:https://www.metrefdb.com.

The data used to generate different images correspond to the first version which can be find on the link below: GitHub at:https://github.com/XXX/metref.

FUNDING

This research was funded by 'Ministerio de Ciencia, Innovación y Universidades,' grant number PID2019-105 969GB-I00, and by Generalitat Valenciana, grant number CIPROM/2021/042. PRE has an FPU Grant (FPU21/03813) from the "Ministerio de Universidades', Spain.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Adami C. What is complexity? BioEssays 2002; 24(12): 1085-94. http://dx.doi.org/10.1002/bies.10192 PMID: 12447974

[2] Adami C. The evolution of biological information: How evolution creates complexity, from viruses to brains. Princeton University Press 2024.

[3] Zander RH. Shannon entropy and informational redundancy in minimally monophyletic bryophyte genera. Plants 2025; 14(19): 3066. http://dx.doi.org/10.3390/plants14193066 PMID: 41095207

[4] Bernaola-Galván PA, Carpena P, Gómez-Martín C, Oliver JL. Genome divergence based on entropic segmentation of DNA. Entropy 2025; 27(10): 1019. http://dx.doi.org/10.3390/e27101019 PMID: 41148977

[5] Koonin EV. The meaning of biological information. Philos Trans-Royal Soc, Math Phys Eng Sci 2016; 374(2063): 20150065. http://dx.doi.org/10.1098/rsta.2015.0065 PMID: 26857678

[6] de la Fuente R, Díaz-Villanueva W, Arnau V, Moya A. Genomic signature in evolutionary biology: A review. Biology 2023; 12(2): 322. http://dx.doi.org/10.3390/biology12020322 PMID: 36829597

[7] Bonnici V, Manca V. Informational laws of genome structures. Sci Rep 2016; 6(1): 28840. http://dx.doi.org/10.1038/srep28840 PMID: 27354155

[8] Manca V, Bonnici V. Informational genomics. Infogenomics: The Informational Analysis of Genomes. Cham: Springer Nature Switzerland 2023; pp. 67-112. http://dx.doi.org/10.1007/978-3-031-44501-9_4

[9] Román-Escrivá P, Bernabeu M, Paganin E, *et al*. Metrics of genomic complexity in the evolution of bacterial endosymbiosis. Biology 2025; 14(4): 338. http://dx.doi.org/10.3390/biology14040338 PMID: 40282203

[10] Moya A, Oliver JL, Verdú M, *et al*. Driven progressive evolution of genome sequence complexity in Cyanobacteria. Sci Rep 2020; 10(1): 19073. http://dx.doi.org/10.1038/s41598-020-76014-4 PMID: 33149190

[11] Kokot M, Długosz M, Deorowicz S. KMC 3: Counting and manipulating *k* -mer statistics. Bioinformatics 2017; 33(17): 2759-61. http://dx.doi.org/10.1093/bioinformatics/btx304 PMID: 28472236

[12] Devadoss J, Rexie JAM, Raimond K. Deterministic techniques for K-Mer frequency counting. 2024 Asia Pacific Conference on Innovation in Technology (APCIT). MYSORE, India, 26–27 July 2024, pp. 1-8. http://dx.doi.org/10.1109/APCIT62007.2024.10673584

[13] Wu L, Zhou M, Xu W, Venkat A, Rosing T, Skadron K. Abakus: Accelerating k-mer counting with storage technology. ACM Trans Archit Code Optim 2024; 21(1): 1-26. http://dx.doi.org/10.1145/3632952

[14] Django. Available from: https://djangoproject.com

[15] Du J. Programming and Development of Web Automation Application Platform Based on Django. Proceedings of the 2024 2nd International Conference on Artificial Intelligence, Systems and Network Security,. New York, NY, USA, 2025, pp. 197-200. http://dx.doi.org/10.1145/3714334.3714368

[16] Bednarz B, Miłosz M. Benchmarking the performance of Python web frameworks. J Comput Sci Inst 2025; 36: 336-41. http://dx.doi.org/10.35784/jcsi.7738

[17] NGINX documentation. Available from: https://nginx.org/en/docs/

[18] Li W, Pan M. Study on Nginx load balancing algorithm based on ESXi performance counter with random operator. Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering,. Xiamen, China, 2024, pp. 862-866. http://dx.doi.org/10.1145/3650400.3650546

[19] Gunicorn. Available from: https://gunicorn.org/
[20] PostgreSQL: The world's most advanced open source relational database. Available from: https://www.postgresql.org/
[21] Celery - Distributed Task Queue. Available from: https://docs.celeryq.dev/en/stable/
[22] Redis Docs. Available from: https://redis.io/docs/latest/
[23] Ivan Solis Zacapantzi A, Ramos Ramos J, Juan Hernandez Mora J, Estela Islas Flores B. Design of an asynchronous architecture based on django and websocket for interactive applications. Int J Sci Res 2025; 692-694: 692-4. http://dx.doi.org/10.21275/SR251108101843
[24] Goldfarb T, Kodali VK, Pujar S, *et al*. NCBI RefSeq: Reference sequence standards through 25 years of curation and annotation. Nucleic Acids Res 2025; 53(D1): D243-57. http://dx.doi.org/10.1093/nar/gkae1038 PMID: 39526381
[25] Tatusova T, DiCuccio M, Badretdin A, *et al*. NCBI prokaryotic genome annotation pipeline. Nucleic Acids Res 2016; 44(14): 6614-24. http://dx.doi.org/10.1093/nar/gkw569 PMID: 27342282
[26] Schoch CL, Ciufo S, Domrachev M, *et al*. NCBI Taxonomy: A comprehensive update on curation, resources and tools. Database 2020; 2020: baaa062. http://dx.doi.org/10.1093/database/baaa062 PMID: 32761142
[27] Holm S. A simple sequentially rejective multiple test procedure. Scand J Stat 1979; 65-70. http://dx.doi.org/10.2307/4615733
[28] ggplot2. Available from: https://ggplot2.tidyverse.org/
[29] Rocchini D, Chieffallo L, Thouverai E, *et al*. Under the mantra: 'Make use of colorblind friendly graphs'. Environmetrics 2024; 35(6): e2877. http://dx.doi.org/10.1002/env.2877