*Journal of the Brazilian Computer Society, 2026, 32:1,* doi: 10.5753/jbcs.2026.6198


# Biotext with SWeePtex: Bioinformatics Tricks to Perform Fast, Accurate, and Content-specific String Embedding

**Diogo de J. S. Machado** [ **Federal University of Paraná** | ***diogo.machado@ufpr.br*** ]
**Camilla R. De Pierri** [ **Federal University of Paraná** | ***camillareginatto.p@gmail.com*** ]
**Antonio C. da Silva Filho** [ **Federal University of Paraná** | ***antonio.camilofilho@gmail.com*** ]
**Flávia de F. Costa** [ **Federal University of Paraná** | ***flaviafc88@gmail.com*** ]
**Nelson A. de M. Lemos** [ **Federal University of Paraná** | ***lemos.nelson@gmail.com*** ]
**Camila P. Perico** [ **Federal University of Paraná** | ***camilapp94@gmail.com*** ]
**Letícia G. C. Santos** [ **Federal University of Paraná** | ***grazielalcs@gmail.com*** ]
**Maricel G. Kann** [ **University of Maryland** | ***mkann@umbc.edu*** ]
**Fábio de O. Pedrosa** [ **Federal University of Paraná** | ***fpedrosa@ufpr.br*** ]
**Roberto T. Raittz** [ **Federal University of Paraná** | ***raittz@ufpr.br*** ]

*Laboratory of Artificial Intelligence Applied to Bioinformatics, Graduate Program in Bioinformatics (PPGBioinfo), Federal University of Paraná (UFPR), Curitiba, PR, 81520-260, Brazil.*

**Received:** 03 July 2025 • **Accepted:** 14 January 2026 • **Published:** 04 May 2026

**Abstract.** The escalating demand for adaptable Artificial Intelligence (AI) systems presents a critical hurdle: generating efficient text embeddings tailored to specific problems. While Large Language Models (LLMs) excel in general contexts, they struggle in specialized domains due to their massive data requirements, opaque embedding strategies, and high computational costs. We introduce Biotext, featuring SWeePtex, a novel framework that adapts successful Bioinformatics techniques for text embedding. By converting text to the Biological Sequence-Like (BSL) format, our Python package enables the application of SWeeP, a tool originally developed for biological sequences, to create content-addressable vectors in natural language, employing the random projection paradigm. Using unsupervised machine learning, we validated this finding by analyzing data from 14,984 MEDLINE abstracts on the thioredoxin theme. Biotext, through SWeePtex, constructs a unified vector space for words and documents from scratch, capturing rich contextual relationships and offering scalable processing. Our usage example demonstrates that this Bioinformatics-inspired method effectively addresses key challenges in Natural Language Processing (NLP), providing interpretable, computationally efficient, and content-addressable linguistic representations for document exploration. Ultimately, Biotext demonstrates that bridging Bioinformatics and NLP yields powerful, efficient, and accessible text analysis tools that balance analytical power with interpretability, particularly valuable in specialized domains and resource-constrained environments. Biotext Python package is freely available at the PyPI repository.

**Keywords:** Text Mining, Vector Embedding, Bioinformatics, Random Projection

## 1 Introduction

Texts are a traditional medium for storing scientific knowledge. However, the ever-growing volume of textual data in public databases underscores the need for improved techniques in text manipulation and analysis [Tshitoyan *et al*., 2019]. In this context, Text Mining (TM) methodologies play a central role, enabling the extraction of structured information from unstructured text, linking human language with computational techniques [Hassani *et al*., 2020], and uncovering hidden insights within complex corpora.

A key aspect of TM is its reliance on string-based data representations, a characteristic it shares with other computational fields such as Bioinformatics. In both domains, entities – whether natural language texts or biological sequences in FASTA format – must be manipulated as symbolic strings. Consequently, techniques developed for mapping biological sequences to vector representations [Asgari and Mofrad, 2015; De Pierri *et al*., 2020; Leimeister *et al*., 2019] can be adapted to TM, thus broadening the repertoire of strategies for text analysis [Hassani *et al*., 2020; Lilleberg *et al*., 2015; Ma and Zhang, 2015].

These cross-domain parallels highlight a broader challenge: the need for transparent and controllable methods of text vectorization that yield interpretable embeddings. Although Large Language Models (LLMs) such as GPT [OpenAI *et al*., 2023; Radford and Narasimhan, 2018] can be used to derive embeddings, this is not their primary design goal, and their black-box complexity can obscure crucial choices in how texts are embedded. For specialized domains, an emerging alternative is to build models from scratch to mitigate the biases inherent in general-purpose training data [Li *et al*., 2025; Yao *et al*., 2022].

The quality of the resulting vector representations defines the semantic space on which all downstream tasks operate, effectively setting an upper bound on their performance [Mikolov *et al*., 2013b]. Embedding methods that faithfully encode domain-specific semantics, syntactic nuances, and conceptual relations can already unlock robust analyses when combined with more straightforward and interpretable mod-

els, a critical requirement as the field moves toward accessible and transparent tools [Hutson, 2024]. Thus, text vectorization is not merely a preprocessing step, but a central research problem that underpins efforts to achieve transparency, control, and trustworthy inference in Natural Language Processing (NLP) systems [Currie, 2023].

Motivated by the need for efficient, interpretable, and accessible embedding methods, this study investigates adapting SWeeP (Spaced Words Projection) – developed initially for biological sequence vectorization – to the domain of NLP. SWeeP [De Pierri *et al*., 2020] has proven efficient for representing biological sequences as compact vectors using FASTA-formatted data. Its successful application in various Bioinformatics studies [da Silva Filho *et al*., 2021; De Pierri *et al*., 2020; Perico *et al*., 2022; Raittz *et al*., 2021] motivates exploration of its broader potential in text processing and mining.

Here, we formally present Biotext, a framework that integrates Bioinformatics and NLP through biological-sequence-inspired text processing. At its core, Biotext features SWeePtex, an adaptation of the SWeeP method for textual data. To achieve this, the framework implements two distinct text-to-BSL (Biological Sequence-Like) conversion strategies: AMINOcode (an amino acid-like encoding) and DNAbits (a nucleotide-like encoding). The Biotext is implemented in a Python package.

Before this study, we demonstrated the applicability of SWeeP to textual data by analyzing the Yoga literature [Ieger-Raittz *et al*., 2025]. In that research, the emphasis was placed primarily on content exploration rather than on the methodological characteristics of text vectorization. In the present work, however, our goal is to foreground the conceptual dimension of Biotext. For exemplification purposes, we applied unsupervised machine learning and geometric analysis of text vector representations to examine PubMed abstracts within a new domain: the thioredoxin literature. This experiment successfully revealed meaningful thematic patterns within the corpus.

Biotext uniquely combines NLP with biological sequence analysis using SWeeP. Subsequent sections delve into the vectorization of natural language (Section 1.1), provide a detailed explanation of the SWeeP vectorization method (Section 1.2), and position SWeePtex within the relevant specialized literature (Section 1.3).

## 1.1 Natural language vectorization

Natural Language Modeling (NLM) explores the intersection between computers and human language. Its primary objective is to equip machines with the capacity to understand, interpret, and generate text at a level that approaches human cognition [Jurafsky and Martin, 2025]. Over the past decades, the field has evolved considerably, transitioning from early symbolic and frequency-based representations to sophisticated neural architectures that capture semantic, syntactic, and contextual dependencies.

One of the first and most influential contributions to text vectorization is the Inverse Document Frequency (IDF), introduced by Jones [Jones, 1972]. By assigning lower weights to commonplace words, IDF highlights specific and informative terms within a corpus. When combined with Term Frequency (TF), the widely used TF-IDF representation becomes central to information retrieval and TM applications [Robertson, 2004]. Despite its simplicity, the conceptual foundation of IDF continues to underpin many current NLP techniques.

As research progressed, new approaches sought to address limitations of purely statistical models. Recurrent Neural Networks (RNNs) [Elman, 1990] enable the capture of temporal and semantic patterns by maintaining an internal state that, in the context of NLP, reflects the preceding context. This innovation allows models to process text sequentially, laying the necessary groundwork for the development of more complex architectures capable of representing long-range dependencies.

In parallel with advances in neural methods, random-projection methods also played a significant role in the evolution of text vectorization. Random Indexing [Kanerva, 1994] proposed representing concepts using randomly generated sparse and dense vectors inspired by cognitive models of human memory. This method facilitates compositionality and flexibility in developing new concepts [Kanerva, 1994; Kanerva *et al*., 2000]. Building on similar principles, Random Mapping [Kaski, 1998] applied the Johnson–Lindenstrauss lemma [Johnson and Lindenstrauss, 1984] to project high-dimensional representations into lower-dimensional spaces while preserving semantic structure. Its application to systems such as WEBSOM demonstrated that random projections can preserve thematic separability while achieving efficiency comparable to that of techniques such as Principal Component Analysis (PCA) [Kaski *et al*., 1998].

A significant shift occurred with the introduction of the neural probabilistic language model [Bengio *et al*., 2003], which pioneered the concept of continuous word representations, now known as "word embeddings". By learning distributed representations that capture the semantics of words in context, this approach opened the door to more expressive and generalizable vector spaces. Subsequent developments further refined this paradigm: Word2Vec [Mikolov *et al*., 2013a] introduced the CBOW and Skip-gram models, providing efficient and accurate embeddings via context-prediction tasks, while FastText [Bojanowski *et al*., 2016] extended this by modeling subword units, thereby increasing robustness in representing rare or morphologically rich words.

The advent of Transformer-based architectures marked another pivotal moment. GPT (Generative Pre-trained Transformer) [Radford and Narasimhan, 2018] used unidirectional masked self-attention to learn autoregressive representations optimized for predicting the next token. In contrast, BERT (Bidirectional Encoder Representations of Transformers) [Devlin *et al*., 2018] leveraged bidirectional attention, enabling the construction of highly context-aware embeddings by simultaneously attending to preceding and following words. Together, these models redefined the representational capacity of NLP systems and catalyzed the development of today's large-scale pretrained models.

Building on this trajectory, DeepSeek-R1 [DeepSeek-AI *et al*., 2025] introduced multi-stage reinforcement learning to enhance vector representations and reasoning capabilities. Distilled versions of the model transfer its high-dimensional internal structure to compact student architectures, such as

Qwen and Llama, enabling more lightweight systems to generate embeddings aligned with those of the teacher model while maintaining efficiency.

Supplementary Table S1 summarizes this historical progression, outlining the chronological development of key methods and concepts in Linguistic Modeling (LM). The table highlights how each methodological shift contributed to shaping the current landscape of computational linguistics.

Despite these substantial advances, the ongoing shift toward deep learning has increased the epistemological complexity of language models. As a result, direct interpretation of their internal representations has become increasingly complex, raising significant concerns about interpretability and reliability [Scorzato, 2024]. This growing complexity underscores the need for methodologies that balance expressive power with transparency, particularly in domains that require rigorous validation.

### 1.2 Sequence vectorization with SWeeP

SWeeP (Spaced Words Projection) [De Pierri *et al*., 2020] is a method that efficiently represents biological sequences as compact vectors using FASTA-formatted data as input. SWeeP operates in two steps: 1) creation of a High-Dimensional Vector (HDV); and 2) pseudo-random projection of the HDV into a Lower-Dimensional Vector (LDV or SWeeP vector).

SWeeP exhibits robust performance even for long, structurally complex sequence patterns, which are typically challenging to process directly within FASTA-based workflows. The method is an efficient, quick, and effortless way of adapting sequences to machine learning tasks. SWeeP is efficient for sequence comparisons, as demonstrated in previous studies [da Silva Filho *et al*., 2021; De Pierri *et al*., 2020; Perico *et al*., 2022; Raittz *et al*., 2021].

### 1.3 Theoretical context

The evolution of language modeling has progressed from classic statistical methods such as IDF [Jones, 1972] to sophisticated modern systems, including GPT [Radford and Narasimhan, 2018], BERT [Devlin *et al*., 2018], and DeepSeek-R1 [DeepSeek-AI *et al*., 2025] (Supplementary Table S1). This trajectory reveals a clear trend toward increasing complexity in capturing intricate semantic patterns.

SWeePtex emerges as a distinct approach grounded in random-projection principles. Although aligned with techniques such as Random Indexing [Kanerva, 1994] and Random Mapping [Kaski, 1998; Kaski *et al*., 1998], the method establishes its own niche by combining the efficiency of early random-projection strategies with a BSL encoding layer, thus reducing the overall complexity of vectorization.

Early neural models, including recurrent networks [Elman, 1990] and the Neural Language Model [Bengio *et al*., 2003], were groundbreaking in pioneering learned distributed representations. The Transformer architecture introduced attention mechanisms, enabling deep and comprehensive contextual modeling. Nevertheless, the models still required substantial computational resources for training and deployment. SWeePtex, on the other hand, focused on embeddings and adopted a distinctive paradigm by using random projection directly, thereby circumventing the demands of deep training.

## 2 Method

The Biotext framework comprises two encoding methods, AMINOcode and DNAbits, that convert natural language into Biological Sequence-Like (BSL) formats. The encoded sequences are compatible with SWeeP vectorization allowed by the SWeePtex method. Those vectors allow for downstream tasks, including semantic embedding and machine learning applications.

### 2.1 Approaches

This study focuses on vectorizing BSL-formatted texts using the SWeeP method and defines SWeePtex. The resulting vectors are suitable for a wide range of analytical techniques, including machine learning, PCA, graph-based methods, and semantic similarity assessment. Furthermore, converting the text into a BSL format from AMINOcode or DNAbits can already enable the use of other Bioinformatics tools, such as sequence alignment and mutation-frequency analysis, although this may require appropriate adaptations.

The AMINOcode substitutes characters with patterns from a pre-defined dictionary, assigning generic encoding to untreated characters. Its variants include a reduced version that generalizes numbers and punctuation, along with a detailed version that explicitly encodes these elements. In contrast, DNAbits uses the American Standard Code for Information Interchange (ASCII), allowing a full representation of all characters. These methods form an increasing sequence with respect to both information preservation and string length. The selection of a coding scheme depends on the specific special characters that must be represented. For this study, SWeePtex used the detailed AMINOcode, as it adequately represented the character set typically found in the scientific literature. Alternative schemes remain available for future projects.

#### 2.1.1 AMINOcode

AMINOcode is the method to encode natural language text in a format based on amino acid sequence representation in FASTA format. It offers two encoding options: reduced and detailed. In reduced encoding, the resulting sequence has the smallest possible size, but it loses information that cannot be recovered during decoding. The detailed options increase the string size by retaining information for additional characters. Supplementary Table S2 shows the character substitution rules.

Supplementary Table S3 provides an example of sentence encoding using the AMINOcode, illustrating both reduced and detailed options and their respective decodings, highlighting differences in information loss. The decoding with AMINOcode does not preserve information about capital letters or characters outside the rules defined by the dictionary. However, it can maintain numbers and major punctuation types in the detailed version.

### 2.1.2 DNAbits

DNAbits encodes natural language text into a DNA-sequence representation in FASTA format. The output string from DNAbits is longer than that of AMINOcode but preserves all the information in ASCII, as shown in Supplementary Table S4. DNAbits processes text by converting each character into its 8-bit binary representation and then splitting each byte into four 2-bit pairs. Each pair of bits is then translated to nucleotides: "00" becomes "A", "10" becomes "C", "01" becomes "G", and "11" becomes "T". For example, the binary representation of the character representing the letter "a" (byte corresponding to the numeric value 97) is "10-00-01-10", yielding "CAGC" after DNAbits encoding.

DNAbits can also be used for alternative amino acid encoding, converting each encoded text into a protein sequence in the three possible reading frames and concatenating the three resulting sequences.

### 2.1.3 SWeePtex

Biotext utilizes SWeeP for text vectorization. Once the texts are encoded with AMINOcode or DNAbits, applying SWeeP is straightforward and requires no additional preparation. This specific combination of BSL encoding and SWeeP is referred to as SWeePtex. The resulting vectors are versatile and can be directly employed in various applications, including machine learning techniques, PCA, graph construction, matrix operations, text comparison, semantic analysis, or other text-based tasks that require vector representation.

### 2.1.4 SWeePtex embedding

To create context-aware word and document embeddings that leverage bidirectional relationships within a given corpus, we designed a protocol based on SWeePtex. The corpus is a set of documents that represent the subject of the analysis. The embedding process starts with creating preliminary vector embeddings for each document using SWeePtex. The embedding of each word is derived by averaging the preliminary vectors of the corresponding records in which the word appears. Similarly, the final document embeddings are obtained by averaging the word vectors of all words in the document, leveraging precomputed word embeddings. Figure 1 illustrates the steps of the method.

The actual embedding program checks for the availability of a preexisting word list. If such a list exists, it is loaded directly for use. If not, the program creates a new word list by extracting all unique terms from the corpus. Next, the program begins mapping, establishing bidirectional relationships between words and documents.

## 2.2 Usage example: "thioredoxin"

This section describes an experiment designed to showcase SWeePtex's capabilities for integrating machine learning and geometric procedures for text analysis and visualization. Here, we propose a pipeline for analyzing biomedical literature from MEDLINE abstracts.

The topic of "thioredoxin" serves as a practical demonstration case, chosen because of the research group's genuine

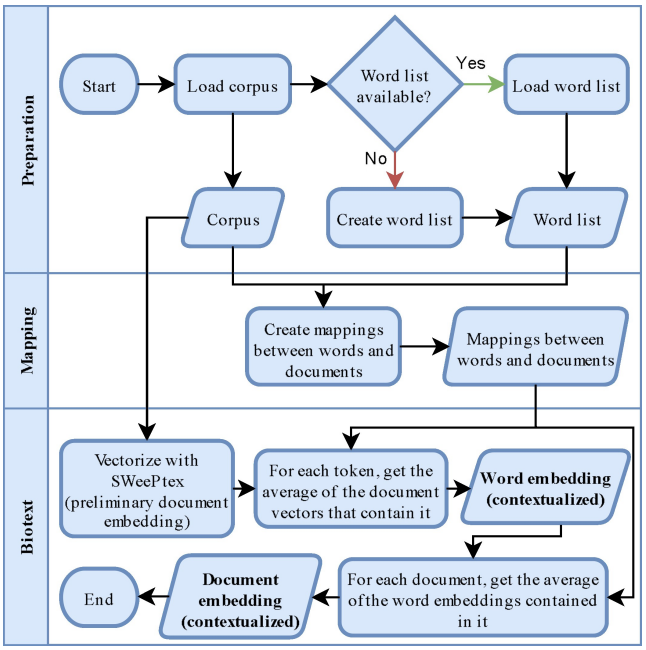


**Figure 1.** SWeePtex embedding flowchart. This figure illustrates the SWeePtex embedding technique. It involves three key steps: (1) loading the textual corpus, (2) creating a bidirectional mapping between words and documents, and (3) computing embeddings for both words and documents through vector averaging operations.

interest arising from a previous study [Rubel *et al.*, 2016]. This selection demonstrates the protocol's applicability to real-world research questions.

The dataset is a MEDLINE (PubMed format) file constructed from a PubMed search on the topic "thioredoxin", comprising 14,984 entries (documents). The corpus is converted to lowercase and tokenized to ensure consistency in subsequent analyses. The search details are in Supplementary Table S6.

Term Frequency-Inverse Document Frequency (TF-IDF) norms are computed for all words in the corpus, with a threshold of greater than or equal to 0.5 applied to select domain-relevant vocabulary.

Semantic representations of words and documents are generated using Biotext embeddings with AMINOcode to convert text into BSL format and SWeePTex to vectorize. In this study, the SWeePtex output is set to 1,200 features.

The PCA and Multi-Layer Perceptron (MLP) autoencoder bottleneck technique [Hinton and Salakhutdinov, 2006] is applied to reduce the dimensionality of word and document embeddings to 50. The goal is to evaluate the behavior of the two reduction techniques.

For unsupervised analysis, the optimal number of clusters for words and documents is determined using the elbow method, followed by $k$-means clustering, for the three representations: the full SWeePtex output, with 1,200 features; the PCA reduced, with 50 features; and the MLP reduced, also with 50 features. All clusterings are evaluated using the Cosine Silhouette, Euclidean Silhouette, Calinski-Harabasz, and Davies-Bouldin metrics. If different data representations yield different numbers of clusters, a secondary clustering is performed using the representation that produces the best Cosine Silhouette score to ensure comparability. Visual comparisons are performed with this fixed number of clusters to

improve comparability.

At the document level, the most salient term trios are extracted using TF-IDF cutoffs and stop-word filtering. Document clusters are exemplified by items selected using two approaches: (1) centroid-proximate, where documents closest to the cluster's geometric center (centroid) are chosen for their typicality and representativeness of core themes, and (2) randomly sampled, where documents are selected arbitrarily to capture broader variability within the group.

This pipeline illustrated most of the resources available through the Biotext package, as well as other potential analyses that can be performed on the generated data. The results are presented in multiple formats, including dimensional-reduction visualizations, cluster-annotation plots, frequency-based word clouds, and structured tabular data summarizing key findings at both the word and document levels.

Finally, a manual analysis of document clusters is performed using the generated material to identify the pattern recognized by unsupervised machine learning.

Additionally, to test the intuitive consistency of vector operations, proximity searches for the target words "leukemia" and "pregnancy" are performed for word-to-word and word-to-document relationships.

### 2.3 Implementation

The approaches are implemented as Python 3 [Van Rossum and Foundation, 2025] modules in the Biotext package. A pipeline script written in the same language automates the usage example experiment.

In the Biotext package, the NumPy library [Harris *et al*., 2020] is used for vector manipulation procedures. The results are made publicly available through the PyPI repository. Biopython [Cock *et al*., 2009] is used to manipulate FASTA files. The Scikit-learn library [Pedregosa *et al*., 2011] is used to implement machine learning in the experimental script. Graphs are generated using Matplotlib [Hunter, 2007] and Wordcloud [Mueller, 2026], while scatterplot refinement is performed using the adjustText [Flyamer *et al*., 2024] package. The Natural Language Toolkit (NLTK) [Bird *et al*., 2009] stopwords list is considered when necessary. The dataset used is constructed from the PubMed database [Canese and Weis, 2002].

## 3 Result

### 3.1 Python package

The Biotext Python package is freely available for installation via PyPI (https://pypi.org/project/biotext). This implementation provides researchers with full access to the SWeePtex methodology in a user-friendly format. The package handles all stages of the analysis, from raw text input to final output visualization.

Table 1 summarizes the core modules of the Biotext package, accompanied by minimal working examples. These demonstrate BSL encoding (with both the AMINOcode and DNAbits options) and the construction of a vector space.

The package includes the exact executable pipeline used for the thioredoxin experiment, ensuring the reproducibility of the published results. All analysis steps, from text processing to final visualizations, can be replicated and reproduced. The documentation provides detailed guidance for running the complete experiment.

In addition to supporting reproducibility, the architecture offers substantial flexibility for new applications. Researchers can adjust encoding schemes, modify projection dimensions, or implement custom similarity metrics to refine their analyses. This modular design accommodates both immediate use and extensive customization for specialized needs.

### 3.2 Thioredoxin result

The implemented Text Mining pipeline generated multiple insightful visualizations and quantitative outputs, revealing patterns across the 14,984 MEDLINE abstracts analyzed. The complete execution takes approximately 1 hour and 48 minutes on a personal computer (details in Supplementary Table S5), with less than 18 minutes dedicated to Biotext embedding (words and documents) and the remainder allocated to pre- and post-processing tasks.

The experimental material is publicly available through the Zenodo repository (https://doi.org/10.5281/zenodo.17810310), comprising the main execution script, supporting utility modules, configuration files, and all output results utilized in this study, allowing full reproducibility of our findings while also providing researchers with adaptable components for future applications.

The elbow method determines the optimal number of clusters for words and documents (Supplementary Figures S1 and S2). For words, the optimal number is 9 clusters for the complete SWeePtex vectors, 16 clusters for the PCA-reduced vectors, and 3 clusters for the MLP-reduced vectors. For documents, all three vector representations result in 3 clusters. In accordance with the methodological criterion, clustering with 3 clusters is also performed for the complete and PCA-reduced SWeePtex representations.

Quantitatively, MLP-based reduced vectorization achieves the best performance, with a cosine silhouette score of 0.6552 for word clustering and 0.7136 for document clustering. This superiority is consistent with the results across all other evaluated metrics. The complete results are detailed in Supplementary Table S7.

Clustering results are visualized through scatter plots for words (Supplementary Figure S3) and documents (Supplementary Figure S4). The document clusters are further illustrated using word clouds (Supplementary Figure S5).

The full experimental output includes tables of the representative document examples and the most frequent term trios for each cluster across all three vectorization methods. For brevity, the Supplementary Material presents only the tables from the best-performing method, the MLP-based reduced vectors (Supplementary Tables S9 and S10). A preliminary interpretation of the document clustering pattern is also provided (Supplementary Table S8).

For target word analysis, the scatter plots display the 10 closest words from the word-to-word search (Supplementary Figures S6 and S7). The word-to-document search is visu-

**Table 1.** Biotext Python package summary. This table summarizes the Biotext modules designed for generating Bioinformatics-inspired text encodings and embeddings at both the word and document levels.

| Module | Description | Example |
|---|---|---|
| **aminocode** | Encodes text using amino acid-like representations. | `encoded = aminocode.encode_string(`<br>`  "Hello world!"`<br>`)`<br>`# Output:`<br>`# 'HYELLYQYSYWYQRLDYPW'` |
| **dnabits** | Encodes text using binary DNA-like representations. | `encoded = dnabits.encode_string(`<br>`  "Hello world!"`<br>`)`<br>`# Output:`<br>`# 'AGACCCGCATGCATGCTTGCAAGAT`<br>`#  CTCTTGCGATCATGCACGCCAGA'` |
| **sweeptex** | Generates document vectors using the SWeePtex algorithm. | `embeddings = sweeptex(`<br>`  ["Text 1", "Text 2"],`<br>`  emb_size=1200`<br>`)`<br>`# Output array shape:`<br>`# (2, 1200)` |
| **sweeptex_emb** | Generates context-aware embeddings for words and documents. | `results = biotext_emb(`<br>`  ["First doc", "Second doc"],`<br>`  return_doc_emb=True,`<br>`  return_word_emb=True`<br>`)`<br>`# Output keys:`<br>`# - Document embeddings`<br>`#  results['doc_emb']`<br>`# - Word embeddings`<br>`#  results['word_emb']` |

alized via word clouds generated from the 30 most similar documents (Supplementary Figures S8 and S9). Both figures present results for all three vector representations. For the operation using the complete SWeePtex vector, which retrieves the documents intuitively most related to each target word, the 30 nearest documents per word are listed in Supplementary Table S11.

The embedding visualization corresponding to the best quantitative results is shown in Figure 2. Panel A displays the resulting word clusters, with each cluster labeled by its most representative terms, as determined by centroid proximity. Panels B and C visualize the semantic neighborhoods of the target words “leukemia” and “pregnancy”, respectively.

# 4 Discussion

## 4.1 Conceptual analysis

Biotext with SWeePtex demonstrates how Bioinformatics methodologies can effectively address key challenges in NLP. Although the BSL format has appeared in third-party implementations [Araujo *et al*., 2022], SWeePtex is, to our knowledge, the first implementation specifically designed for text vectorization. It provides interpretable, content-addressable representations that enable direct vector manipulation during literature exploration, while maintaining computational efficiency and ensuring continuity in randomness-based linguistic representations.

SWeeP is efficient for biological analyses, and its potential extends to NLP through Biotext. This approach aligns with a strategy in language models: the use of random projections for representation. For example, Kanerva [1994] illustrates how random representation reflects human conceptual cognition. However, despite this theoretical foundation, the predominant technology in NLP largely follows alternative directions.

Transformer-based models, while effective, require substantial computational resources, making the development of custom models challenging. To address this limitation, the recent study by DeepSeek [DeepSeek-AI *et al*., 2025] proposes model distillation as a strategy to reduce computational costs and underscores its importance for future research. In contrast, SWeePtex revisits Kanerva’s concept and implements it within a Bioinformatics framework, thereby maintaining continuity in randomness-based linguistic representation.

Random approaches and methods based on Artificial Neural Networks (ANN) are not mutually exclusive; they represent complementary research directions. Patterns identified through random exploration can be refined and structured using ANNs, thereby enabling synergistic integration that


**Figure 2.** Visualization of the Thioredoxin usage example result via scatter plots. This figure presents two-dimensional projections (reduced via PCA) of data processed through the described pipeline. Semantic clustering and proximity calculations use SWeePtex feature vectors processed by the MLP-based technique. Panel A shows word clusters identified by the optimal configuration, with each cluster labeled by its most representative terms. Panels B and C illustrate the semantic neighborhoods of "leukemia" and "pregnancy", respectively.

improves solution design. This combined methodology represents a potential path for future development.

## 4.2 Thioredoxin insights

The thioredoxin experiment shows how SWeePtex handles large data volumes and extracts relevant contextual information. The analysis is performed on a substantial collection of documents, a volume impractical for manual evaluation, and the results are human-interpretable.

Clustering using MLP-reduced vectors yields better quantitative results (Supplementary Table S7). Hinton and Salakhutdinov [2006] demonstrates that the dimensionality reduction approach using an autoencoder neural network can be stronger than PCA reduction. Specifically, in the SWeePtex case, we consider the possibility that the MLP adds characteristics to the new representation that relate patterns not explicitly captured by the averaging approach.

A visual inspection of the scatter plots of the word clusters reveals broadly similar patterns across the three vector representations tested (Supplementary Figure S3), despite differences in dimensionality. Since MLP-reduced vectors (Figure 2A) produced the strongest quantitative results, we used this as the basis for the evaluation, from which the following interpretation arises: Cluster 0 is the most specific grouping highly technical terms related to molecular structure, enzyme mechanisms, and biochemistry (e.g. catalytic, disulphide, isomerase); Cluster 1 comprises intermediate-level terms that define specific biological systems, disease models, and experimental contexts (e.g., hepatocytes, apoptosis, mice); and Cluster 2 contains general methodological and discourse-linking terms (e.g., or, evidence, over), forming the

universal scaffold of scientific writing.

The structure of the clustering words reveals a linguistic gradient, starting with broad contextual language, advancing to concepts of the biological domain, and concluding with precise structural and mechanistic terminology. However, it represents only one possible semantic interpretation of the clusters, and other valid perspectives could likely be identified.

In contrast to word-level evaluation, document-level analysis using the same MLP-reduced projection offers a broader view of how the thioredoxin-related literature is organized (Supplementary Table S8). Three thematic domains emerged from the clustering patterns. Cluster 0 focuses on Fundamental Biochemistry and Enzyme Mechanisms, bringing together studies on structural biology, catalytic cycles, protein dynamics, and mechanistic descriptions of redox enzymes and cofactors. Cluster 1 groups topics associated with Chemical Biology and Environmental Applications, spanning chemical interventions in biological systems, including drug discovery, enzyme inhibitors, antimicrobial agents, and environmental detoxification processes. Cluster 2 encompasses research on Disease Mechanisms and Therapeutic Approaches, reflecting research in translational biomedical science, including investigations of pathological processes and assessments of therapeutic strategies in cellular and animal models.

In the word-to-word search analysis of the target words "leukemia" and "pregnancy", no significant qualitative difference is intuitively perceptible among the three vector representations tested; the nearby terms are easily interpretable in all cases (Supplementary Figures S6 and S7, Figures 2B and 2C). However, for word-to-document search, as is evident from word clouds (Supplementary Figures S8 and S9), clustering with vectors reduced by the MLP did not yield documents strongly related to the search term. That is, the MLP projection harmed the comparability between word and document vectors. Between the full SWeePtex vector and the one projected by PCA, coherence is visible in the word cloud, and, looking at the documents returned by proximity, the full vector looks the most coherent (Supplementary Table S11).

When using the full-vector approach, the system can associate documents containing both "leukemia" and "leukaemia" spellings, confirming that it interprets semantic relationships rather than relying on exact term matches. For "leukemia", it retrieves studies on thioredoxin reductase inhibitors and drug resistance, while for "pregnancy", it returns the literature on preeclampsia and gestational diabetes. These results demonstrate the system's utility in biomedical text analysis, as it identifies contextually relevant documents without depending on exact term occurrences.

Finally, the thioredoxin usage example shows an adaptable pipeline that can be extended to other research topics. To ensure reproducibility, we made all experimental material, including execution scripts, utility modules, and generated output, available. The scripts are thoroughly documented, enabling third-party users to modify and reimplement them.

# 5 Conclusion

The integration of Bioinformatics and NLP in SWeePtex offers a computationally efficient approach that addresses limitations of traditional Transformer models, particularly in interpretability, reliability, and resource utilization. SWeePtex utilizes content-addressable vectors, enabling direct manipulation and transparency in tasks such as literature exploration. This approach is based on Bioinformatics, which provides a foundation for vectorizing text. The efficient structuring of text into a compact vector offers a representation that is immediately applicable and easily integrable into modern downstream Artificial Intelligence (AI) applications and NLP pipelines.

SWeePtex can be seamlessly integrated into existing text-processing pipelines to broaden the range of semantic analyses they support. Its efficiency is underpinned by the organization of words and texts within a unified vector space. This core mechanism has already demonstrated its semantic power, as evidenced by our earlier study [Ieger-Raittz *et al*., 2025], which successfully used it to perform vector search and create effective mappings for the Yoga literature domain. This existing framework validates the potential of projection-based methods for extracting deep textual meaning, thereby setting the stage for applying SWeePtex to other specialized corpora.

Looking ahead, a key opportunity is to explore text augmentation techniques to further enhance SWeePtex's capabilities. This approach entails the programmatic generation of specialized text by integrating Large Language Models (LLMs) into the SWeePtex environment. Using local corpus knowledge captured by SWeePtex, the integrated LLM could generate text focused on specific aspects or nuances within the corpus, effectively enriching it for targeted analytical tasks. This synergy aims to combine the contextual depth of LLMs with the computational efficiency of SWeePtex.

# Declarations

## Authors' Contributions

DJSM, CRDP, LGCS, and RTR contributed to the conceptualization of the study. DJSM, CRDP, and RTR contributed to methodology, formal analysis, investigation, and writing (original draft). DJSM and RTR contributed to software development. DJSM, CRDP, ACSF, FFC, NAML, CPP, MGK, FOP, and RTR contributed to writing (review and editing). RTR is responsible for supervision. All authors read and approved the final manuscript.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

The authors acknowledge the support of the Federal University of Paraná (UFPR), the Technology and Professional Education Sector (SEPT) at UFPR, the Artificial Intelligence Applied to Bioinformatics group at UFPR, and the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES). FFC acknowledges

the support from the National Institute of Science and Technology (INCT CERBC, grant no. 406645/2022–1).

## Funding

This research is funded by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES).

## Availability of data and materials

The Biotext Python package is available in the PyPI repository (https://pypi.org/project/biotext). Datasets and experimental materials, including the main script, utility modules, and output results used in this paper, are available in the Zenodo repository (https://-doi.org/10.5281/zenodo.17810310).

# References

Araujo, J. D., Santos-e Silva, J. C., Costa-Martins, A. G., Sampaio, V., de Castro, D. B., de Souza, R. F., Giddaluru, J., Ramos, P. I. P., Pita, R., Barreto, M. L., Barral-Netto, M., and Nakaya, H. I. (2022). Tucuxi-blast: Enabling fast and accurate record linkage of large-scale health-related administrative databases through a dna-encoded approach. *PeerJ*, 10:e13507. DOI: 10.7717/peerj.13507.

Asgari, E. and Mofrad, M. R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10(11):e0141287. DOI: 10.1371/JOURNAL.PONE.0141287.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155. Available at:`https://dl.acm.org/doi/10.5555/944919.944966`.

Bird, S., Klein, E., and Loper, E. (2009). Natural language processing with python. Book.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. DOI: 10.48550/arxiv.1607.04606.

Canese, K. and Weis, S. (2002). Pubmed: The bibliographic database. Available at:`https://www.ncbi.nlm.nih.gov/books/NBK153385/pdf/Bookshelf_NBK153385.pdf`.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and De Hoon, M. J. L. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. 25(11):1422–1423. DOI: 10.1093/bioinformatics/btp163.

Currie, G. M. (2023). Academic integrity and artificial intelligence: is chatgpt hype, hero or heresy? *Seminars in Nuclear Medicine*, 53(5):719–730. DOI: 10.1053/j.semnuclmed.2023.04.008.

da Silva Filho, A. C., Marchaukoski, J. N., Raittz, R. T., Pierri, C. R. D., de Jesus Soares Machado, D., Fadel-Picheth, C. M. T., and Picheth, G. (2021). Prediction and analysis in silico of genomic islands in aeromonas hydrophila. *Frontiers in Microbiology*, 12. DOI: 10.3389/fmicb.2021.769380.

De Pierri, C. R., Voyceik, R., Santos de Mattos, L. G. C., Kulik, M. G., Camargo, J. O., Repula de Oliveira, A. M., de Lima Nichio, B. T., Marchaukoski, J. N., da Silva Filho, A. C., Guizelini, D., Ortega, J. M., Pedrosa, F. O., and Raittz, R. T. (2020). Sweep: representing large biological sequences datasets in compact vectors. *Scientific Reports*, 10(1):91. DOI: 10.1038/s41598-019-55627-4.

DeepSeek-AI *et al*. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. DOI: 10.48550/arXiv.2501.12948.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. DOI: 10.48550/arxiv.1810.04805.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211. DOI: 10.1207/s15516709cog1402_1.

Flyamer, I. *et al*. (2024). Phlya/adjusttext: 1.3.0. *Zenodo*. DOI: 10.5281/zenodo.14019059.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with numpy. *Nature 2020 585:7825*, 585(7825):357–362. DOI: 10.1038/s41586-020-2649-2.

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., and Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, 4(1):1. DOI: 10.3390/bdcc4010001.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507. DOI: 10.1126/science.1127647.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95. DOI: 10.1109/MCSE.2007.55.

Hutson, M. (2024). Forget chatgpt: why researchers now run small ais on their laptops. *Nature*, 633(8030):728–729. DOI: 10.1038/D41586-024-02998-Y.

Ieger-Raittz, R., De Pierri, C. R., Perico, C. P., de Fatima Costa, F., Bana, E. G., Vicenzi, L., de Jesus Soares Machado, D., Marchaukoski, J. N., and Raittz, R. T. (2025). What are we learning with yoga? mapping the scientific literature on yoga using a vector-text-mining approach. *PLOS ONE*, 20(5):e0322791. DOI: 10.1371/JOURNAL.PONE.0322791.

Johnson, W. B. and Lindenstrauss, J. (1984). *Extensions of Lipschitz mappings into a Hilbert space*, page 189–206. American Mathematical Society. DOI: 10.1090/conm/026/737400.

Jones, S. K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21. DOI: 10.1108/eb026526.

Jurafsky, D. and Martin, J. H. (2025). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models. Available at:`https://web.stanford.edu/~jurafsky/slp3/`.

Kanerva, P. (1994). *The Spatter Code for Encoding Concepts at Many Levels*, page 226–229. Springer London, London. DOI: 10.1007/978-1-4471-2097-1_52.

Kanerva, P., Kristoferson, J., and Hols, A. (2000). Ran-

dom indexing of text samples for latent semantic analysis. Available at:https://escholarship.org/uc/item/5644k0w6.

Kaski, S. (1998). Dimensionality reduction by random mapping: fast similarity computation for clustering. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, page 413–418. DOI: 10.1109/IJCNN.1998.682302.

Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. (1998). Websom – self-organizing maps of document collections. *Neurocomputing*, 21(1):101–117. DOI: 10.1016/S0925-2312(98)00039-3.

Leimeister, C. A., Schellhorn, J., Dörrer, S., Gerth, M., Bleidorn, C., and Morgenstern, B. (2019). Prot-spam: fast alignment-free phylogeny reconstruction based on whole-proteome sequences. *GigaScience*, 8(3):1–14. DOI: 10.1093/GIGASCIENCE/GIY148.

Li, S., Hu, R., and Wang, L. (2025). Efficiently building a domain-specific large language model from scratch: A case study of a classical chinese large language model. DOI: 10.48550/arXiv.2505.11810.

Lilleberg, J., Zhu, Y., and Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. *Proceedings of 2015 IEEE 14th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2015*, page 136–140. DOI: 10.1109/ICCI-CC.2015.7259377.

Ma, L. and Zhang, Y. (2015). Using word2vec to process big text data. *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, page 2895–2897. DOI: 10.1109/BIGDATA.2015.7364114.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. DOI: 10.48550/arXiv.1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. DOI: 10.48550/arXiv.1310.4546.

Mueller, A. C. (2026). Wordcloud. Available at:https://github.com/amueller/word_cloud.

OpenAI *et al*. (2023). Gpt-4 technical report. DOI: 10.48550/arXiv.2303.08774.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., M., B., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830. DOI: 10.48550/arxiv.1201.0490.

Perico, C. P., Pierri, C. R. D., Neto, G. P., Fernandes, D. R., Pedrosa, F. O., de Souza, E. M., and Raittz, R. T. (2022). Genomic landscape of the sars-cov-2 pandemic in brazil suggests an external p.1 variant origin. *Frontiers in Microbiology*, 13. DOI: 10.3389/fmicb.2022.1037455.

Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training. Available at:https://api.semanticscholar.org/CorpusID:49313245.

Raittz, R. T., Pierri, C. R. D., Maluk, M., Batista, M. B., Carmona, M., Junghare, M., Faoro, H., Cruz, L. M., Battistoni, F., de Souza, E., de Oliveira Pedrosa, F., Chen, W. M., Poole, P. S., Dixon, R. A., and James, E. K. (2021). Comparative genomics provides insights into the taxonomy of azoarcus and reveals separate origins of nif genes in the proposed azoarcus and aromatoleum genera. *Genes*, 12:1–21. DOI: 10.3390/genes12010071.

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520. DOI: 10.1108/00220410410560582.

Rubel, E. T., Raittz, R. T., Coimbra, N. A. d. R., Gehlen, M. A. C., and Pedrosa, F. d. O. (2016). Proclat, a new bioinformatics tool for in silico protein reclassification: case study of drab, a protein coded from the dratgb operon in azospirillum brasilense. *BMC bioinformatics*, 17(Suppl 18). DOI: 10.1186/S12859-016-1338-5.

Scorzato, L. (2024). Reliability and interpretability in science and deep learning. *Minds and Machines*, 34(3):1–31. DOI: 10.1007/S11023-024-09682-0.

Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K. A., Ceder, G., and Jain, A. (2019). Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98. DOI: 10.1038/s41586-019-1335-8.

Van Rossum, G. and Foundation, P. S. (2025). Python language reference. Available at:https://docs.python.org/3/reference/.

Yao, X., Zheng, Y., Yang, X., and Yang, Z. (2022). Nlp from scratch without large-scale pretraining: A simple and efficient framework. DOI: 10.48550/arXiv.2111.04130.