mathematics

Article

# LDAShiny: An R Package for Exploratory Review of Scientific Literature Based on a Bayesian Probabilistic Model and Machine Learning Tools

**Javier De la Hoz-M [1,2,*], Mª José Fernández-Gómez [2,3] and Susana Mendes [4]**

[1] Facultad de Ingeniería, Universidad del Magdalena, Santa Marta 470004, Colombia
[2] Department of Statistics, University of Salamanca, 37008 Salamanca, Spain; mjfg@usal.es
[3] Institute of Biomedical Research of Salamanca, 37008 Salamanca, Spain
[4] MARE, School of Tourism and Maritime Technology, Polytechnic of Leiria, 2520-614 Peniche, Portugal; susana.mendes@ipleiria.pt
[*] Correspondence: jdelahoz@unimagdalena.edu.co

**Abstract:** In this paper we propose an open source application called LDAShiny, which provides a graphical user interface to perform a review of scientific literature using the latent Dirichlet allocation algorithm and machine learning tools in an interactive and easy-to-use way. The procedures implemented are based on familiar approaches to modeling topics such as preprocessing, modeling, and postprocessing. The tool can be used by researchers or analysts who are not familiar with the R environment. We demonstrated the application by reviewing the literature published in the last three decades on the species *Oreochromis niloticus*. In total we reviewed 6196 abstracts of articles recorded in Scopus. LDAShiny allowed us to create the matrix of terms and documents. In the preprocessing phase it went from 530,143 unique terms to 3268. Thus, with the implemented options the number of unique terms was reduced, as well as the computational needs. The results showed that 14 topics were sufficient to describe the corpus of the example used in the demonstration. We also found that the general research topics on this species were related to growth performance, body weight, heavy metals, genetics and water quality, among others.

**Keywords:** text mining; topic modeling; latent dirichlet allocation; automatic literature review

**Citation:** De la Hoz-M, J.; Fernández-Gómez, M.J.; Mendes, S. LDAShiny: An R Package for Exploratory Review of Scientific Literature Based on a Bayesian Probabilistic Model and Machine Learning Tools. *Mathematics* **2021**, *9*, 1671. https://doi.org/10.3390/math9141671

Academic Editor: Maria Laura Manca

Received: 17 May 2021
Accepted: 24 June 2021
Published: 16 July 2021

## 1. Introduction

A literature review is considered an integral part of the research process in any scientific area, and seeks to discover the relevant sources of a particular subject of study. Thus, it plays a crucial role since wisdom is generated through the process of interpretation and integration of existing knowledge [1].

Nowadays there is an increasing amount of scientific literature published in digital form in databases such as Scopus or Web of Science, to mention two of the most used by researchers [2]. Therefore, it can be inferred that there is a gap between the availability and use of information. A literature review in a conventional way is restricted, has a high cost in terms of of time, and has limited processing power, which leads researchers to restrict the amount of documents to review. Nowadays, machine learning approaches make it feasible to process huge amounts of data, allowing researchers to spend less time examining their findings. When human-assisted information processing, such as encryption, is replaced with computer-assisted processing, dependability improves and costs fall [3].

Asmussen and Muller [4] mention that the exploratory review of literature in a conventional way will soon become outdated because it is a process that has a high cost in time, with limited processing power, which leads researchers to restrict the amount of documents to be reviewed, which is a problem in the initial exploratory phase of an investigation since what is needed is an overview of the state of the art of research. The large amount of information available makes searching, retrieving and summarizing information cumbersome and challenging, so the use of tools capable of searching, organizing and summarizing a large collection of text documents in the scientific field is in demand.

In the open source environment R [5] in the Comprehensive Archive Network (CRAN) we can find a list of 59 packages related to natural language processing (NLP), eight of which implement the modeling of topics through latent Dirichlet assignment (LDA) [6]: lda collapsed Gibbs sampling methods for topic models [7]; lda.svi fit LDA models using stochastic variational inference [8]; ldaPrototype prototype of multiple LDA runs [9]; lda.svi LDA coupled with time series analyses [8]; ldatuning, tuning of the LDA models parameters [10]; LDAvis, interactive visualization of topic models [11]; topicdoc topic-specific diagnostics for LDA and Correlated Topic Models (CTM) topic models [12]; topicmodels [13] and textmineR [14] functions for text mining and topic modeling.

To date, there is no free statistical software package with a graphical user interface (GUI) where analysts and researchers can take advantage of the combined power of several packages to perform LDA-focused scientific literature reviews in an interactive (point-and-click) way. The LDAShiny application is primarily aimed at researchers who wish to use machine learning to explore a large number of documents (e.g., scientific articles) to identify research trends. This is beneficial for researchers who know little about the research field. The application allows a large number of documents to be grouped automatically in less time than if it were done manually, thus providing an overview of the directions of the investigation. Therefore, from the perspective of a literature review, this is valuable as the decision to include or exclude articles is made in a more informed way at a later stage.

This study presents the development of a computer tool for performing a literature review with a focus on topic modeling (a branch of unsupervised methods). It could help to reduce or to replace the time spent by the researcher at the computer by automatically generating review topics based on the statistical qualities of the documents utilized, without the need for prior classification, categorization, or labeling. Thus the possible bias due to subjective choices of the researchers could be avoided or minimized. Furthermore, historical and current research and trends in the field under study can be more easily synthesized.

There are several packages for modeling topics in the R environment. However, they require some statistical and machine learning skills that not all researchers possess [4]. Therefore, the main aim of LDAShiny was to make the typical LDA workflow easier to use, especially for those who are unfamiliar with R. With LDAShiny the analysis can be performed interactively in a web browser, which makes it easier for many more researchers to apply this technique to review the scientific literature.

Thus, and in order to facilitate the understanding of the work exposed here, the manuscript presents a section that introduce a quick overview of topic modeling with LDA. Then, the methods employed are presented (Section 3), followed by the detailed description of the LDAShiny GUI (Section 4). In Section 5, the use of the LDAShiny GUI using *Oreochromis niloticus* literature over the last three decades is explained. Finally, the conclusions are presented in Section 6.

## 2. Topic Modelling for Exploratory Literature

Topic modeling is a classic problem in NLP and machine learning. It refers to a set of algorithms and statistical methods of learning, recognition and extraction that aim to analyze the hidden structure of a collection of documents to discover the topics, how they are related to one another and how they have evolved over time. It has the advantage of not requiring any prior annotations or document labeling because the topics emerge from the analysis of the original texts [15].

It has the advantage that no previous annotations or labeling of the documents are required. Its use spans practically every aspect of text mining and information processing, including text summarization, information retrieval and text classification [16]. Topic modeling allows us to organize and summarize electronic files in various formats (web pages, scientific articles, books, images, sound, videos and social networks) at a scale that would be impossible by human annotation [15].

Latent semantic analysis (LSA) [17] and probabilistic latent semantic analysis (PLSA) [18] are the predecessors of LDA However, considering that LDA is one of the most used methods [3,19,20], we decided on it due to its highly qualified ease of use, understanding and applicability [4].

LDA is a Bayesian variant of PLSA, based on a set of words assumption, which states that words in a text are interchangeable and that documents are represented as a series of individual words [6]. This algorithm was initially applied to text corpora but its use has been extended to images [21] and videos [22].

LDA is a generative model. In other words, it is a model that shows how data are produced, and once you have a model of how they are generated, you can know which target variable generated them. The Dirichlet distribution, which is a multivariate version of the beta distribution, is used by LDA to extract the features of the subjects and documents.

The generative process from which LDA assumes the documents come, is described as:

1. For every topic k:
    a. Draw a distribution over the words (i.e., vocabulary V) $\beta_k \sim Dir(\eta)$ [6]
2. For every document d:
    a. Draw a distribution over topics $\theta_d \sim Dir(\alpha)$ (i.e., per document topic proportion) [6]
    b. For each word w within document d:
        i. Draw a topic assignment, $z_{d,n} \sim Mult(\theta_d)$ (i.e., per-word topic assignment) [6]
        ii. Draw a word $w_{d,n} \sim Mult(\beta_{z,d,n})'$ [6,23,24].

Each topic k comes from a Dirichlet distribution $\beta_k \sim Dir(\eta)$, and is a multinomial distribution over the vocabulary. Furthermore, each document is represented as a topic distribution and originates from a $\theta_d \sim Dir(\alpha)$. The Dirichlet parameter $\eta$ defines the smoothing of the words within topics, and $\alpha$ the smoothing of the topics within documents [6]. The joint distribution of all the hidden variables $\beta_k$, $\theta_D$ (document topic proportions within D), $z_D$ (word topic assignments), and observed variables $w_D$ (words in documents), is expressed by Equation (1):

$$P(\beta_k\ ,\theta_D,\ z_D,w_D) = \prod_{k=1}^{K} P(\beta_k|\eta) \prod_{d=1}^{D} P(\theta_d|\alpha) \prod_{n=1}^{N} P(z_{d,n}|\theta_d)\ P(w_{d,n}|z_{d,n},\beta_k) \tag{1}$$

This shows the statistical assumptions behind LDA's generative process. The per-word topic assignment $z_{d,n}$ depends on the previously drawn (step 2.a.) per-document topic proportion $\theta_d$. Furthermore, the drawn word $w_{d,n}$ depends on the per-word topic assignment $z_{d,n}$ (step 2.b.i) and all the topics $\beta_k$ (we retrieve the probability of $w_{d,n}$ (row) from $z_{d,n}$ (column) within the K × V topic matrix). The latent variables are the per-word topic assignment, the per-document topic distribution and the topics, which are not observed. To infer the hidden structure using statistical inference, we would have to condition on the single seen variable, i.e., the words within the documents. This might be thought of as a reversal of the generative process [6].

Equation (2) expresses the posterior or conditional probability. Unfortunately, due to the denominator, this probability cannot be computed [6]. Therefore, machine learning algorithms have to be used to find approximations of the marginal probability of the observations $P(w_D)$, This marginal probability of the observations is the chance of seeing the observed corpus under any topic model [15].

$$P(\beta_k, \theta_D, z_D|w_D) = \frac{P(\beta_k, \theta_D, z_D, w_D)}{P(w_D)} \tag{2}$$

Although it is impossible to accurately calculate the posterior probability, statistical posterior inference can be used to obtain an approximate value close enough to the true value. Two main types of reasoning technique can be identified: sampling-based algorithms [25,26] and variational-based algorithms [26–28]. Sampling-based algorithms sample from the posterior, usually taking one variable at a time, fixing the other variables. Repeating this process for several iterations makes the inference process converge, so the sample values have the same distribution as if they came from the true posterior value. An example of a sampling-based algorithm is the Gibbs sampler (a full explanation about Gibb sampling can be found in Griffths and Steyvers [23]), a Markov chain Monte Carlo (MCMC) algorithm. Variational-based algorithms create a family of distributions that are closest (distance is measured with Kullback–Leibler (KL) divergence) to the true posterior. It should be noted that both variational and sampling-based algorithms provide similar accurate results [29].

The latent variables $\theta$ and $z$ are frequently used in inference to establish which subjects a document contains and from which subject a certain word in a document was derived. The variational posterior probability can be used to estimate latent variables on the premise that it is a reasonable approximation of the real posterior probability. If the variational expectation maximization (VEM) is employed for estimate, inference is always based on the variational posterior probabilities [13].

## 3. Materials and Methods

The methodology utilized to create the LDAShiny program is based on well-known topic modeling approaches to data cleansing and processing. The main contribution in this work is not to introduce new ways of processing data, but to learn how the methods are combined and how they can be easily used by researchers through the use of this application. The inspiration for the creation of LDAShiny can be found in Asmussen and Moller [4] who considered that the intelligent literature review process consists of three steps: preprocessing, topic modeling and post-processing.

In our proposal, the review process consists of four steps: preprocessing, inference, topic modeling and post-processing (Figure 1).

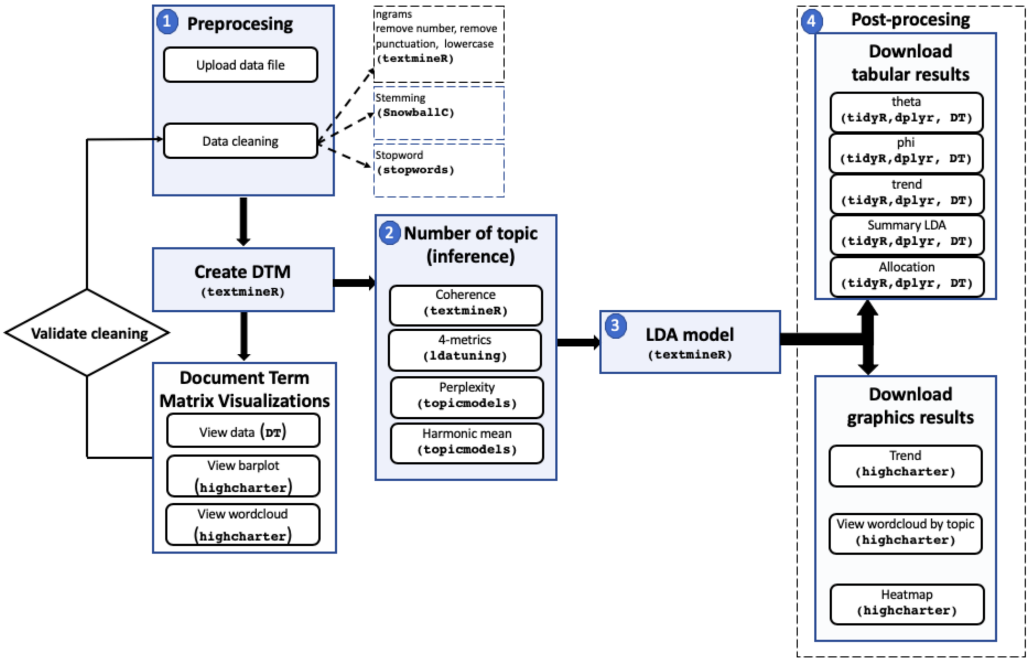


**Figure 1.** LDAShiny package outline. In parentheses are the main packages used.

### *3.1. Preprocessing*

Preprocessing consists of loading and preparing the documents for subsequent processes. This phase plays a very important role, being generally the first step in text mining techniques and applications [30]. Pre-processing seeks to normalize or convert the set of text to a more convenient standard form that allows the reduction of the data dimensionality of the data matrix by eliminating noise or meaningless terms. Within the pre-processing we have the "cleaning" in which the following tasks are performed:

- Tokenization, which is the procedure of separating morphemes (words). According to Jurafsky and Martin [31] it is beneficial in both linguistics and computer science.
- n-gram inclusion: an n-gram is a contiguous sequence of n words [32]. Although it is more usual to analyze individual words, in some cases, such as in the life sciences, incorporating bigrams would be advantageous because scientific names of species are made up of two words. In LDAShiny we can work with unigrams, bigrams or trigrams (three words frequently occurring).
- Remove numbers, despite the fact that numbers are frequently thought to be uninformative, there are some areas of knowledge where numbers can provide valuable information, for instance, in legislative matters, bills or decrees can be significant with respect to content legislation. That is why in the developed application the researcher can decide whether or not to eliminate the numbers.
- Remove StopWord, a term coined by Luhn [33]. The procedure consists of discarding words that have no lexical meaning and that appear in texts very frequently (such as articles and pronouns). There are many potential StopWord lists, however, we restrict ourselves to a pre-compiled list of words provided by the R StopWord [34]. LDAShiny allows performing this procedure in 14 languages Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Norwegian, Portuguese, Romanian, Russian, Spanish, and Swedish.
- Stemming, which is the simplest version of lemmatization. It consists of reducing words to basic forms [35]. Although it is often used as a reduction technique, it must be used carefully, since it could combine words with different meanings, for example in the phrases "college students partying", and "political parties", stemming would reduce partying and parties as the same basic form.
- Remove infrequently used terms (sparsity). This procedure is very useful because it allows removing the terms that appear in very few documents before continuing with the successive phases. Among the reasons for this procedure is the computational feasibility, as this process drastically reduces the size of the matrix without losing significant information and can also eliminate errors in the data, such as misspelled words. This only applies to terms that comply with:

  $$df(t) > N(1 - sparce) \tag{3}$$

  where df is the frequency of documents of the term t and N is the number of vectors. For example, if the sparse value is 0.99, the terms that appear in more than 1% of the documents are taken. As a general rule, terms that appear in less than 0.5–1 percent of the articles should be discarded [19,36,37]. However, there has been no systematic examination of the implications of this pre-processing decision on the analyses' final phase.
- Eliminating blank spaces and punctuation characters, as well as lowering the entire text, are other standard procedures used to prevent a word from being counted twice due to capitalization.

The cleaning process must be validated. However, to date there has been no scientific way to establish when this process ends, so the process must be iterative, since it is not possible to guarantee an identical cleaning procedure when conducting an exploratory review [4]. Once the pre-processing phase is completed, the document-term matrix (DTM) is obtained as input data for topic models.

### *3.2. Inference*

LDA is a model for latent variables using correlations between words and latent semantic topics in a collection of documents [38]. This implies that the parameter k (number of topics) of the algorithm is crucial and must be established beforehand, since the validity of the results obtained depends largely on the inference process of the model. In theoretical terms, a very large number of topics will produce overly specific topics, while conversely, a very small number would handle broad and heterogeneous themes [39].

There are a variety of metrics that can be used to determine the optimal number of topics. In our package we implement the following:

- perplexity defined by [6] for a set of text of M documents as:

$$perplexity(D_{text}) = exp\left(\frac{\sum_{d=1}^{M}\log(w_d)}{\sum_{d=1}^{M} N_d}\right) \tag{4}$$

  where $N_d$ is the number of words in the d-document of the text corpus $D_{text}$ and $w_d$ is the $d^{th}$ document in the corpus. It is monotonically decreasing and algebraically equivalent to the inverse of the geometric mean probability per word. When comparing several models, the one with the lowest value of perplexity is considered the best [6].
- marginal likelihood that can be approximated by harmonic mean. This method has first been applied by Griffiths and Steyvers in their 2004 Bayesian approach, in order to find the optimal number of topics [23,40].
- coherence [41]. It is based on the distribution hypothesis [42] which states that words with similar meanings tend to coexist in similar contexts. The procedure used for this metric is based on the TextmineR package [14], which implements a thematic coherence measure based on probability theory and consists of fitting several models and calculating the coherence for each of them. The best model will be whichever offers the greatest measure of coherence.
- other metrics can be found in the ldatuning package Arun 2010 [43], CaoJuan 2009 [44], Deveaud 2014 [45], Griffiths 2004 [23]. The approach of these metrics is simple and they are based on finding extreme values (minimization Arun 2010 and CaoJuan 2009; maximization Deveaud 2014 and Griffiths 2004).

For a further description of each of the metrics used by the application, it is recommended to review the corresponding articles.

### *3.3. Latent Dirichlet Assignment (LDA) Model*

Once the number of topics has been determined, LDAShiny proceeds to execute the LDA model. Some parameters such as the number of iterations can be modified by a number of iterations greater than that used to make the inference. As a result, the modeling DTM is reduced to two matrices. The first one, theta, has rows that indicate the distribution of topics on documents $P(topic_k|document_d)$. The second one, phi, has rows that indicate the distribution of words on topics $(token_v|topic_k)$.

### *3.4. Post-Processing*

This step involves processing the results and obtaining a description of the topics. The distribution of topic terms does not come with a semantic interpretation. However, depending on the frequency of the words, the topics can be labeled correctly in most cases. Lewis, Zamith, and Hermida [46] mention that algorithmic analyses have a very limited capacity to understand latent meanings in human language, so manual labeling is considered a standard [47]. However, in the latter case, the labeling can provide different topic labels depending on the researcher. The textmineR [14] package provides a topic labeling based on a naive labeling algorithm built on bigrams. However, as mentioned, these algorithms have limited capabilities, but may well serve as a guide.

Once all the topics have been labeled, with the help of the theta matrix, the procedure continues assigning documents to each topic, classifying them according to the highest probability of each document for each topic. In this way the documents will also be grouped.

Labelling requires validation by an expert in the field of research, otherwise mislabeled topics and an invalid result could be obtained [4].

In order to facilitate the characterization of the topics in terms of their trends, the simple regression slopes for each theme are used. The year is the dependent variable and the proportions of the topics in each year the response variable [23]:

$$\theta_k^y = \frac{\sum_{m \in y} \theta_{mk}}{n^y} \tag{5}$$

where $m \in y$ represent the articles published in a certain year and, $\theta_{mk}$ the proportion of the k-topic and $n^y$ the total number of articles published in the year $y$ [48]. Topics whose regression slopes are positive (negative) at a statistical significance level are interpreted as increasing (declining) their interest respectively, and if the slopes are not significant, the topics will be classified as fluctuating trends.

## 4. LDAShiny Graphical User Interface (GUI)

The LDAShiny is web-based and has been developed in R using the shiny [49] web application framework. LDAShiny provides an integrated platform for exploratory review of scientific information, offering a number of options to manage, explore, analyze and visualize data. This is particularly beneficial to researchers who are not as familiar with R, or programming in general, but wish to use the methods described here.

The LDAShiny package is accessed from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=LDAShiny. To install, load, and launch it, type the following in R:

- R > install.packages(“LDAShiny”)
- R > library(“LDAShiny”)
- R > LDAShiny::runLDAShiny()

The GUI proposed in this work provides a menu that, from top to bottom, guides the user through the analysis:

- About: this panel serves as the software’s introduction page. The application’s general information, as well as the software’s goal, are displayed in English and Spanish.
- Data input and preprocessing: this provides an interface for users to load the data to be analyzed. In addition, there are also different options to perform preprocessing.
- Document term matrix visualizations: the matrix of terms and documents can be viewed both in tabular and graphical form in this menu. The tabular data can be downloaded in csv xlxs or pdf format or can be copied to the clipboard. The graphics (barplot or wordcloud) can be downloaded in .png, .pdf, .jpeg, .svg, and .pdf format.
- Number of topics (inference): The options to set the input parameters of each of the metrics used to find the number are available in this menu.
- LDA model tutorial: this menu offers a vignette (in English and Spanish) with videos that serve as a quick guide where the basic steps to use the software are explained.

Table 1 lists the details of each panel or menu.

**Table 1.** Details and description of LDAShiny panel.

| Panel | Item | Menu | Description |
|---|---|---|---|
| Preprocessing | Upload data file | Use example data set? | Check box indicating whether a file that comes with the package. |
| | | Choose csv file | Clicking the Browse button will load local data files in csv format |
| | | Header | Checkbox indicating if the first line of the file contains the names of the columns |
| | | stringAsFactors | String as factors. |
| | | Separator | Field separating character. |
| | | Select | PickerInput presents the loaded dataset and displays it in the Statistical summary table view. |
| | Data cleaning | Incorporate information | Clicking three times the Incorporate information button will load the data into preprocessing. |
| | | Select id document | PickerInput for specifying vector of names for documents. |
| | | Select document vector | PickerInput for specifying character vector of documents |
| | | Select publish year | PickerInput for specify the vector containing the year the document was published |
| | | ngrams | Radio buttons to specify the type of ngram to use (unigram, bigram or trigram). |
| | | Remove number | Checkbox to specify whether or not to delete the numbers in thecorpus (if clicked it will remove the numbers). |
| | | Select language for stopword | PickerInput to specify the language used in the stopword removal (the list contains 14 languages to choose from). |
| | | Stop Words | Text field to include additional stop words to remove (words must be separated by commas). |
| | | Stemming | Checkbox if clicked, stemming is performed |
| | | Sparsity | Slider to select sparse parameter. |
| | | Create Document-Term Matrix DTM | After clicking the Create DTM button, a spinner will be displayed during the process. Once finished, a table with the dimensions of the created matrix is displayed. |
| Document Term Matrix Visualizations | | View Data | Clicking the View Data button will be display a summary. Also shown are a series of buttons that allow downloading in csv, xlxs or pdf formats, print the file Print, copy it Copy to the clipboard, and a button to configure the number of rows Show to be used in the summary. |
| | | View barplot | Clicking the View barplot button will be display a barplot. The number of bars can be configured using the slider shown in the Dropdown button, Select number of term. In the upper right part of the graph (export button), clicking on it, you can download the graph in different formats (.png, .jpeg, .svg and .pdf) |
| | | View wordcloud | Clicking the View wordcloud button will be display a wordcloud. The number of words can be configured by the slider shown in the Dropdown button Select number of term In the upper right part of the graph (export button), clicking on it, you can download the graph in different formats (.png, .jpeg, .svg and .pdf). |
| Number of topic (inference) | Tab Coherence | Iterations | Numeric input parameter that specifies how many iterations will be performed |

**Table 1.** *Cont.*

| Panel | Item | Menu | Description |
|---|---|---|---|
| | | Burn-in | Numeric input parameter that specifies how many burn-in for posterior sampling will be performed |
| | | Hyper-parameter | Numeric input parameter that specifies the alpha value of the Dirichlet distribution. |
| | Tab 4-metrics | Estimation method | There are two radio buttons to select the estimation algorithm, Gibb for Gibbs sampling and VEM for variational expectation maximization |
| | Tab Perplexity | Iteration, Burn-in, and Thin | These parameters control how many Gibbs sampling draws are made. The first burning iterations are discarded and then every thin iteration is returned for each iterations |
| | Tab Harmonic mean | Iteration, Burn-in, and Keep | If a keep parameter was given, the log-likelihood values of every keep iteration, are contained. |
| LDA model | Run model | | The input parameters are the number of topics (K), number of iterations and the alpha parameter of the Dirichlet distribution. Clicking the Run LDA Model button, a spinner will be displayed. Once the process is complete, a table will be displayed that includes coherence score, prevalence, and 10 top-terms for each topic. Also shown are a series of buttons that allow downloading in csv, xlxs or pdf formats, print the file Print, copy it Copy to the clipboard and a button to configure the number of rows (Show rows). |
| | Download tabular results | theta | Clicking on the theta button, a table will be displayed that includes topic, document and theta |
| | | phi | Clicking on the phi button, button, a table will be displayed that includes topic, term and phi |
| | | trend | Clicking on the trend button, a table showing the results of a simple linear regression (intercept, slope, test statistic, standard error and *p*-value) where the year is the dependent variable and the proportions of the topics in the corresponding year is the response variable. |
| | | Summary LDA | Clicking on the Summary LDA button, three sliders will be shown at the top, this allows the summary configuration: Select number of labels, Select number top terms, and Select assignments the latter is a documents by topics matrix similar to theta. This will work best if this matrix is sparse, with only a few non-zero topics per document |
| | | Allocation | Clicking on the Allocation button, a table will be shown where the user can find the documents that can be organized by topic. Thanks to the slider located at the top one we can choose the number of documents per topic to be displayed. |
| | Download graphics | trend | Clicking on the trend button, a line graph will be shown (one line for each topic) where time trends can be visualized. The graphic is interactive, clicking on the lines they will be removed or displayed as the user decides. |
| | | View wordcloud by topic | Clicking the View wordcloud by topic button will be display a wordcloud. In the drop-down button you can select the topic from which we want to generate the wordcloud, also, in the slider you can select the number of words to show |
| | | heatmap | Clicking the heatmap button will display a heatmap. The years are shown on the $x$-axis, the $y$-axis shows the topics and the color variation represents the probabilities. |

## 5. Demonstration of LDAShiny GUI

To demonstrate how the GUI is used, an exploratory review of scientific texts referring to the species *O. niloticus* was carried out. This species is used when considering that aquaculture research involves very diverse areas (engineering, ecology, biology, physiology, economics, environmental and political sciences, among others), which in most cases must be developed together to successfully produce a specific species at the industry level. It was assumed that an exploratory review of the literature on the species was necessary and that the number of documents to be reviewed was too large to carry out a manual review.

The inclusion criteria focused on selectin those research articles in which information about this species was discussed, using its scientific name as a keyword. Likewise, it was decided to take into account documents in which the name of the species was mentioned either in the title, in abstract or as keyword, ensuring that the largest number of potentially relevant documents was included.

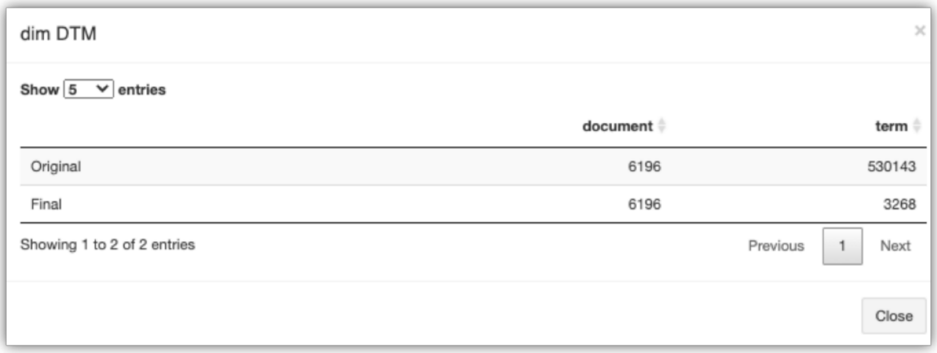
The search for articles was carried out through Scopus database considering that it supports the downloading of metadata batches of the articles (which speeds up data collection). Furthermore, it is one of the databases most used by researchers [2]. A number of 6196 abstracts of articles were found (in the last three decades 1991–June 2020). This number of documents makes an individual exploratory review too time consuming, so the set of articles considered provides a good example to test the application. The file used for the demonstration can be downloaded at the link https://github.com/JavierDeLaHoz/o.niloticus/blob/main/O.niloticus.csv.

### 5.1. Preprocesing

The required dataset must be in a wide format (one article or abstract per row). Upload the *O. niloticus* data file to LDAShiny from the Upload Data panel. Next, on the Data cleaning panel, click the Incorporate information button, and then specify the columns for id document (Title in our case). Select document vector (Abstract), and select the year of publication (Year), Then click on the checkbox to select ngram (Bigrams). Remove the numbers, select the language for the stopwords and include the words you want to remove. In our example we use, in addition to the default list, a pre-compiled list called SMART (System for the Mechanical Analysis and Retrieval of Text) from the stopword package. In addition, some terms detected in the validation were also removed, such as all the terms with two letters and also the following words: article, articles, author, authors, blackwell, copyright, fish, francis, international, journal, licensee, nature, nile, niloticus, objective, oreochromis, present, press, published, publishing, reserved, result, resulted, results, rights, science, showed, significant, significantly, sons, springer, study, taylor, tilapia, total, verlag and wiley. The complete list of stopwords used in the example can be found at https://github.com/JavierDeLaHoz/stopword/blob/main/stopword.csv.

For this example, no stemming was performed and the Sparsity slider used was 99.5%, that is, the terms that appeared in more than 0.5%. Finally, the Create DTM button was clicked, after cleaning, 530,143 unique terms remained in the corpus, however the procedure reduced the number of unique terms to 3268, greatly reducing computational needs (Figure 2).

The resulting DTM matrix can be previewed in the Document Term Matrix Displays panel, in both tabular and graphical form (Figure 3). The information presented in tabular form contains the terms (term), their frequency of appearance (term_freq) and how many documents these terms appear (doc_freq). In addition, idf is the inverse frequency of the document, which measures if the term is common or not in the document collection. It is obtained by dividing the total number of documents by the number of documents that contain the term, and then the logarithm of that quotient is taken. We observe that words such as growth, levels, higher, protein, control weight, species, effects, days and observed, are the most frequent terms that appear the most in the evaluated documents (Figure 3).


dim DTM

Show 5 entries

|  | document | term |
|---|---|---|
| Original | 6196 | 530143 |
| Final | 6196 | 3268 |

Showing 1 to 2 of 2 entries

**Figure 2.** Document term matrix dimensions before (original) and after preprocessing (final).

The information on the frequency of terms can also be seen graphically in the form of a barplot or wordcloud. In both options the user can configure the number of words to display (Figure 3).

This statistical description, in the collection of articles, can provide a specific but limited overview of a particular field of research. As result, the words found in the evaluated articles represent the variety of topics investigated for *O. niloticus.*

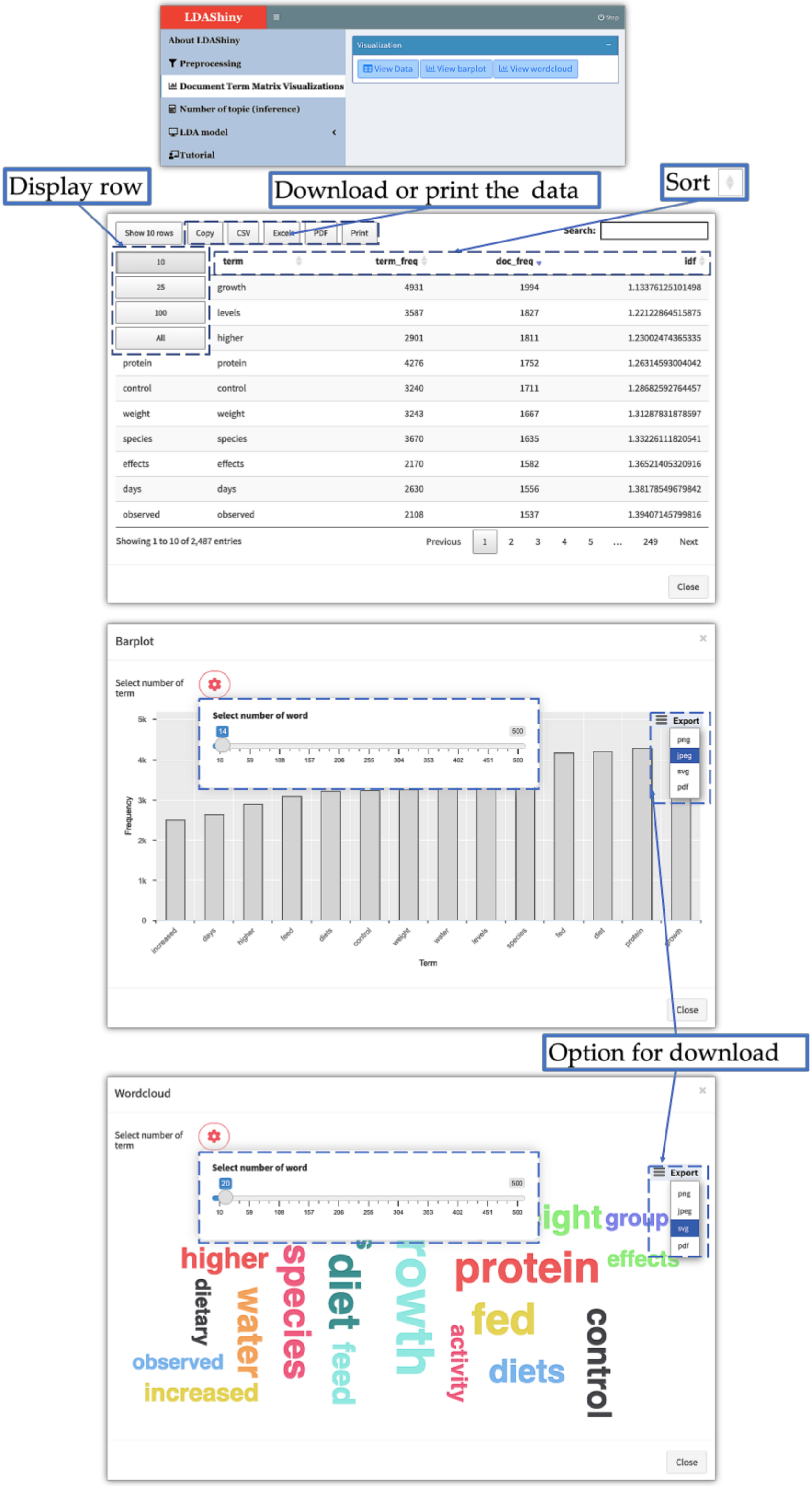
### *5.2. Number of Topics (Inference)*

Once the DTM matrix has been obtained, the next step is to determine the optimal number of topics. A very small number of topics can generate broad and heterogeneous topics. By contrast, a high number of k will produce themes that are too specific and in both cases the interpretation is complicated [39]. Therefore, the least number of topics was preferred as the intention is to provide an overview of the usefulness of the LDAShiny GUI. The highest quality LDA model can be determined using different metrics such as topic coherence [40]. This is a measure of the quality of a model topic from the point of view of human interpretability. Some authors consider it to be a more appropriate measure than computational metrics, such as perplexity [50] and likelihood of holdout data [24]. It should be noted that finding the number of topics is a computational expensive procedure and, although LDAShiny uses parallelism, the procedure may take anywhere from a few minutes to even a couple of days. It depends on the size of the DTM, the number of models (number of topics to evaluate), and the number of cores on the computer (LDAShiny works with the total number of cores).

In the left margin of Figures 4–7 the configuration options for each of the metrics used to calculate the number of topics are shown. The graphic outputs of each one appear on the right one. In every scenario, the amount of time it took to complete the inference is displayed. The time elapsed for estimating the number of topics in each of the metrics was 13,922, 2276, 5832 and 2755 s for “coherence”, “4-metrics”, “perplexity” and “Harmonic mean”, respectively.

However, it should be noted that the times required are very dependent on the size of the DTM matrix, the number of iterations used (in all cases of the example there were 1000 iterations except for 4-metric, which uses 2000 by default), and the number of central processing unit (CPU) cores available (in our case a laptop with four cores was used).

Regarding the number of topics, the metrics Griffiths 2004, CaoJuan 2009, Arun 2010, Perplexity and Harmonic mean agreed to stablish that the number of suitable topics is between 45 and 50, while Deveaud 2014 showed 35 and Coherence 14 Topics. However, there are considerations that must be addressed when using LDAShiny. There is no common accepted way to choose the number of topics in a topic model. Thus, finding the right number of topics can be quite complex [4].

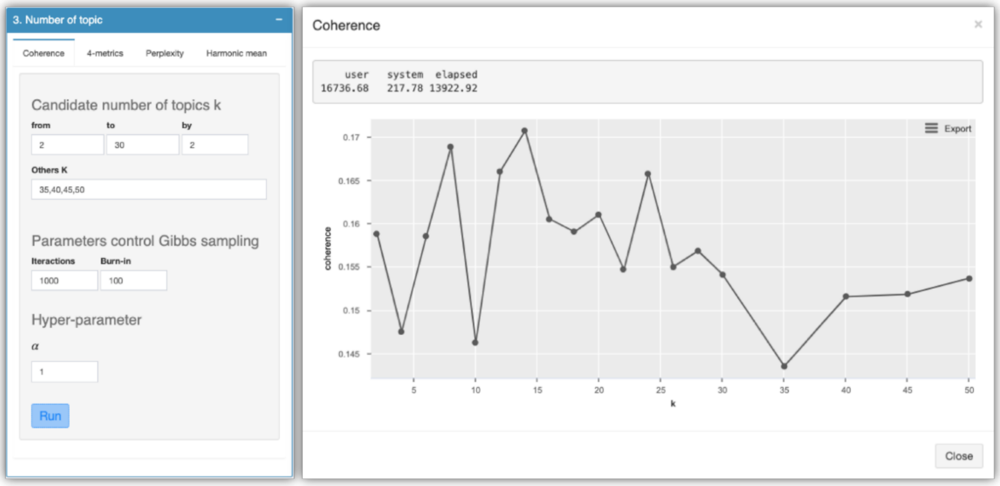


| | term | term_freq | doc_freq | idf |
|---|---|---|---|---|
| | growth | 4931 | 1994 | 1.13376125101498 |
| | levels | 3587 | 1827 | 1.22122864515875 |
| | higher | 2901 | 1811 | 1.23002474365335 |
| protein | protein | 4276 | 1752 | 1.26314593004042 |
| control | control | 3240 | 1711 | 1.28682592764457 |
| weight | weight | 3243 | 1667 | 1.31287831878597 |
| species | species | 3670 | 1635 | 1.33226111820541 |
| effects | effects | 2170 | 1582 | 1.36521405320916 |
| days | days | 2630 | 1556 | 1.38178549679842 |
| observed | observed | 2108 | 1537 | 1.39407145799816 |

Showing 1 to 10 of 2,487 entries

Previous 1 2 3 4 5 … 249 Next

**Figure 3.** Document term matrix display options.



3. Number of topic

Coherence 4-metrics Perplexity Harmonic mean

Candidate number of topics k

from: 2 to: 30 by: 2

Others K: 35,40,45,50

Parameters control Gibbs sampling

Iteractions: 1000 Burn-in: 100

Hyper-parameter

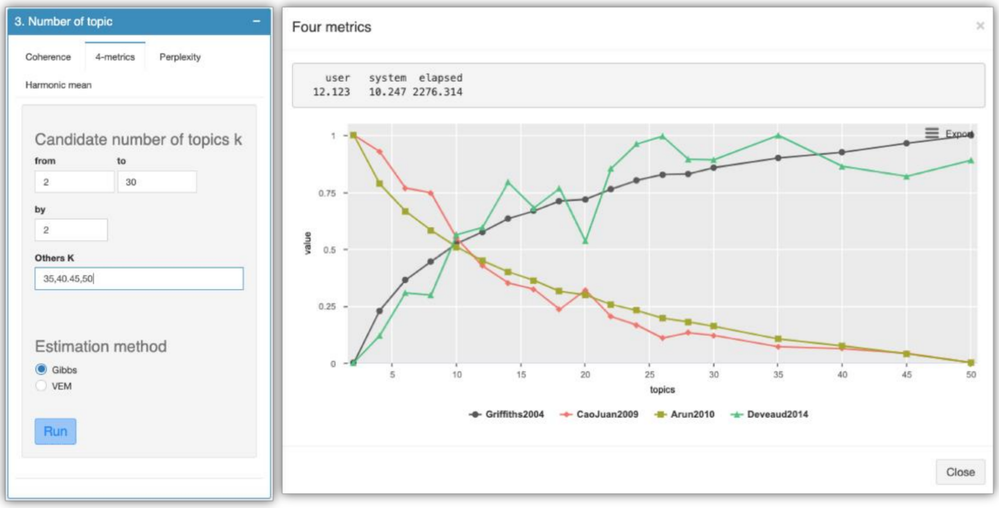
$\alpha$: 1

Run

Coherence

```
    user   system  elapsed
16736.68   217.78 13922.92
```

**Figure 4.** Configuration options used to calculate the number of topics (coherence method).



3. Number of topic

Coherence 4-metrics Perplexity Harmonic mean

Candidate number of topics k

from: 2 to: 30

by: 2

Others K: 35,40,45,50

Estimation method

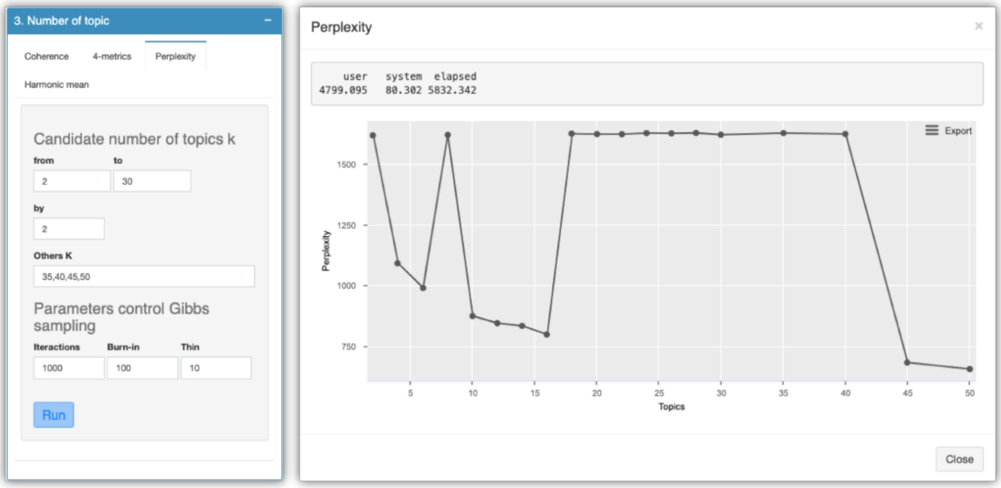
● Gibbs
○ VEM

Run

Four metrics

```
  user   system  elapsed
12.123   10.247 2276.314
```

**Figure 5.** Configuration options used to calculate the number of topics (comparison of four metrics).



3. Number of topic

Coherence 4-metrics Perplexity Harmonic mean

Candidate number of topics k

from: 2 to: 30

by: 2

Others K: 35,40,45,50

Parameters control Gibbs sampling

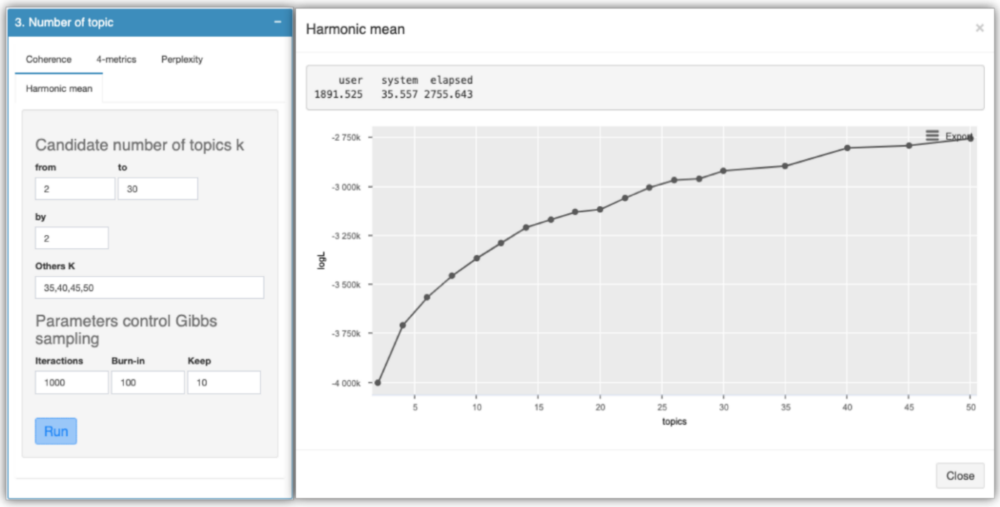
Iteractions: 1000 Burn-in: 100 Thin: 10

Run

Perplexity

```
    user  system  elapsed
4799.095  80.302 5832.342
```

**Figure 6.** Configuration options used to calculate the number of topics (perplexity).



Candidate number of topics k

from 2

to 30

by 2

Others K 35,40,45,50

Parameters control Gibbs sampling

Iteractions 1000

Burn-in 100

Keep 10

Run

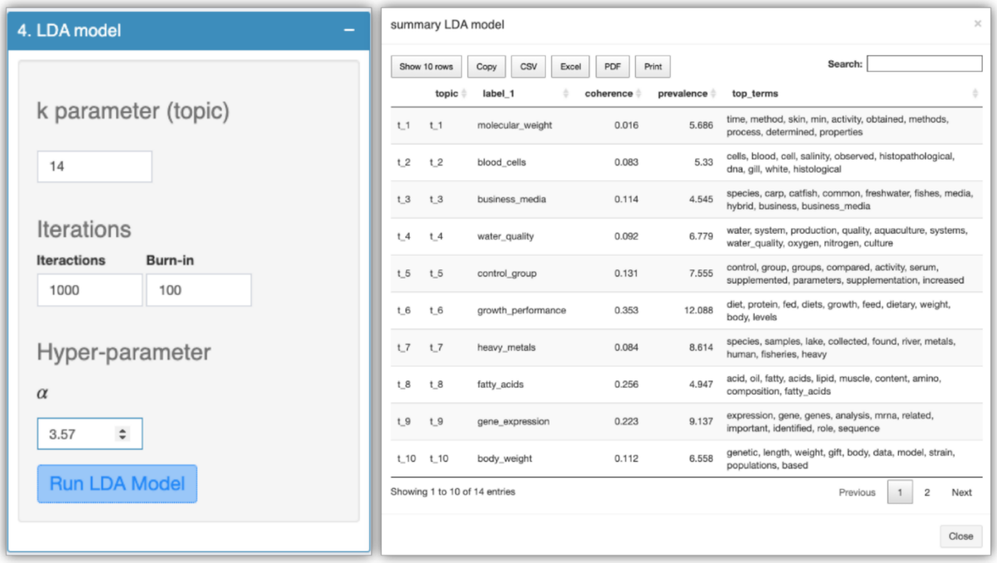
```
   user  system  elapsed
1891.525  35.557 2755.643
```

**Figure 7.** Configuration options used to calculate the number of topics (harmonic mean).

Because a general description of research on *O. niloticus* was required in our case, we preferred to use the smallest number of topics. However, determining what constitutes a small number of topics will differ from the model's input corpus. Nevertheless, visualizing the metric outputs can provide the appropriate guidance.

## 5.3. LDA Model

Once the number of topics has been defined, the LDA model is fitted. The parameters of inference should be used as a guide. However, some can be modified, such as the number of iterations, which may be higher. Also, the recommendation of Griffiths and Steyvers (2004) [23] could be used, setting a $\alpha$ value of 50/k. In this example, as input parameters, 1000 iterations and 100 burnin were used, and the $\alpha$ value was set to 3.57 (Figure 8).



k parameter (topic)

14

Iterations

Iteractions 1000

Burn-in 100

Hyper-parameter

$\alpha$

3.57

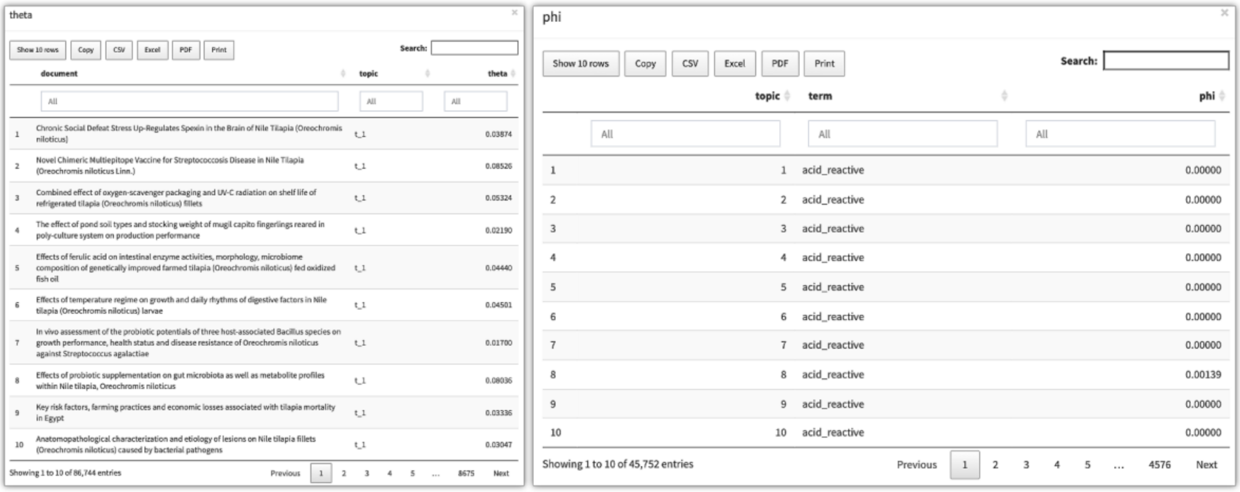
Run LDA Model

| | topic | label_1 | coherence | prevalence | top_terms |
|---|---|---|---|---|---|
| t_1 | t_1 | molecular_weight | 0.016 | 5.686 | time, method, skin, min, activity, obtained, methods, process, determined, properties |
| t_2 | t_2 | blood_cells | 0.083 | 5.33 | cells, blood, cell, salinity, observed, histopathological, dna, gill, white, histological |
| t_3 | t_3 | business_media | 0.114 | 4.545 | species, carp, catfish, common, freshwater, fishes, media, hybrid, business, business_media |
| t_4 | t_4 | water_quality | 0.092 | 6.779 | water, system, production, quality, aquaculture, systems, water_quality, oxygen, nitrogen, culture |
| t_5 | t_5 | control_group | 0.131 | 7.555 | control, group, groups, compared, activity, serum, supplemented, parameters, supplementation, increased |
| t_6 | t_6 | growth_performance | 0.353 | 12.088 | diet, protein, fed, diets, growth, feed, dietary, weight, body, levels |
| t_7 | t_7 | heavy_metals | 0.084 | 8.614 | species, samples, lake, collected, found, river, metals, human, fisheries, heavy |
| t_8 | t_8 | fatty_acids | 0.256 | 4.947 | acid, oil, fatty, acids, lipid, muscle, content, amino, composition, fatty_acids |
| t_9 | t_9 | gene_expression | 0.223 | 9.137 | expression, gene, genes, analysis, mrna, related, important, identified, role, sequence |
| t_10 | t_10 | body_weight | 0.112 | 6.558 | genetic, length, weight, gift, body, data, model, strain, populations, based |

Showing 1 to 10 of 14 entries

**Figure 8.** Configuration options used to calculate LDA model.

Within the tabular results of the model is the list of probabilities of each article for each topic (matrix theta) and the matrix that shows the most frequent words in each topic (phi) (Figure 9). The results of the estimations of the simple linear regression and their *p*-value (trends) (Figure 10, left). Also, they show the summary of the model where the label, coherence score, prevalence and the top term for each topic are included (summary) (Figure 10 right) and finally a table with the allocation of topics (Figure 11).



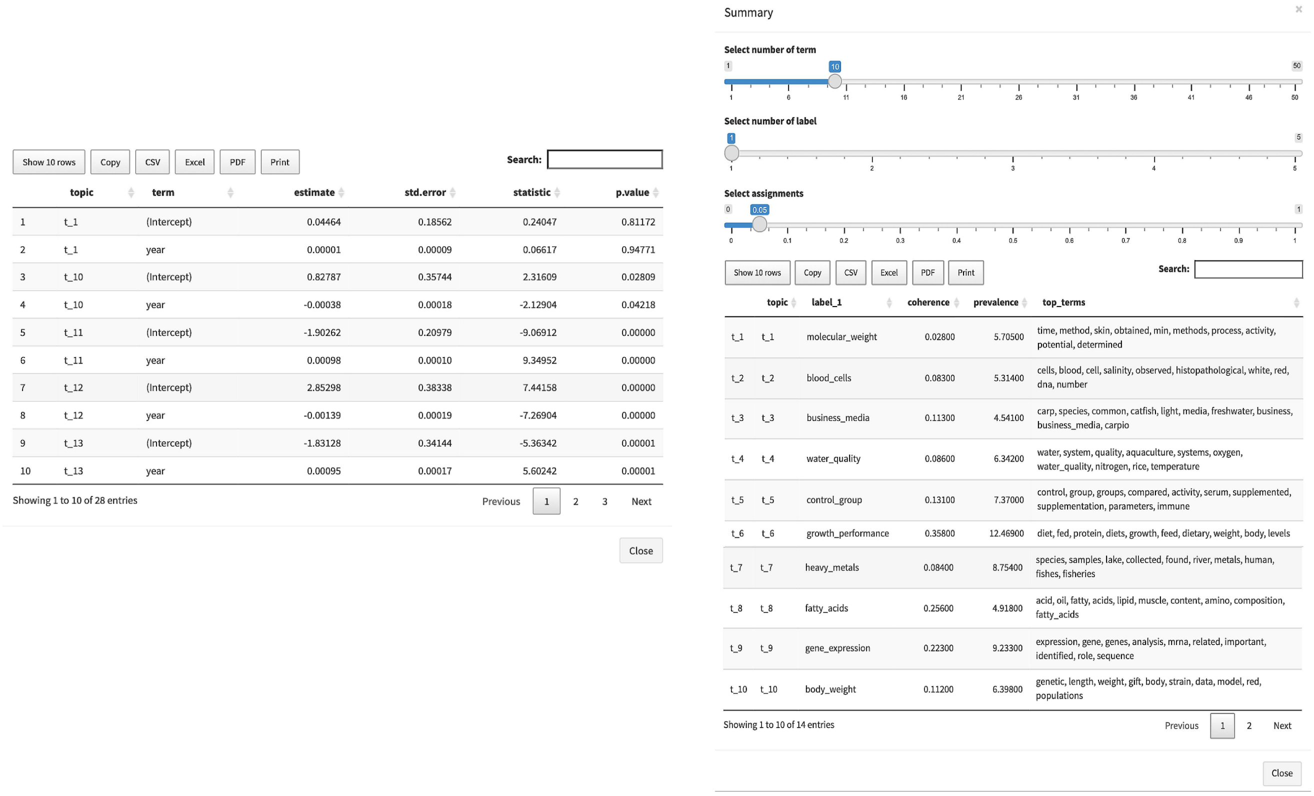
|  | document | topic | theta |
|---|---|---|---|
| 1 | Chronic Social Defeat Stress Up-Regulates Spexin in the Brain of Nile Tilapia (Oreochromis niloticus) | t_1 | 0.03874 |
| 2 | Novel Chimeric Multiepitope Vaccine for Streptococcosis Disease in Nile Tilapia (Oreochromis niloticus Linn.) | t_1 | 0.08526 |
| 3 | Combined effect of oxygen-scavenger packaging and UV-C radiation on shelf life of refrigerated tilapia (Oreochromis niloticus) fillets | t_1 | 0.05324 |
| 4 | The effect of pond soil types and stocking weight of mugil capito fingerlings reared in poly-culture system on production performance | t_1 | 0.02190 |
| 5 | Effects of ferulic acid on intestinal enzyme activities, morphology, microbiome composition of genetically improved farmed tilapia (Oreochromis niloticus) fed oxidized fish oil | t_1 | 0.04440 |
| 6 | Effects of temperature regime on growth and daily rhythms of digestive factors in Nile tilapia (Oreochromis niloticus) larvae | t_1 | 0.04501 |
| 7 | In vivo assessment of the probiotic potentials of three host-associated Bacillus species on growth performance, health status and disease resistance of Oreochromis niloticus against Streptococcus agalactiae | t_1 | 0.01700 |
| 8 | Effects of probiotic supplementation on gut microbiota as well as metabolite profiles within Nile tilapia, Oreochromis niloticus | t_1 | 0.08036 |
| 9 | Key risk factors, farming practices and economic losses associated with tilapia mortality in Egypt | t_1 | 0.03336 |
| 10 | Anatomopathological characterization and etiology of lesions on Nile tilapia fillets (Oreochromis niloticus) caused by bacterial pathogens | t_1 | 0.03047 |

Showing 1 to 10 of 86,744 entries Previous 1 2 3 4 5 … 8675 Next

|  | topic | term | phi |
|---|---|---|---|
| 1 | 1 | acid_reactive | 0.00000 |
| 2 | 2 | acid_reactive | 0.00000 |
| 3 | 3 | acid_reactive | 0.00000 |
| 4 | 4 | acid_reactive | 0.00000 |
| 5 | 5 | acid_reactive | 0.00000 |
| 6 | 6 | acid_reactive | 0.00000 |
| 7 | 7 | acid_reactive | 0.00000 |
| 8 | 8 | acid_reactive | 0.00139 |
| 9 | 9 | acid_reactive | 0.00000 |
| 10 | 10 | acid_reactive | 0.00000 |

Showing 1 to 10 of 45,752 entries Previous 1 2 3 4 5 … 4576 Next

**Figure 9.** Output tabular of the model (theta and phi matrix).



|  | topic | term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|---|---|
| 1 | t_1 | (Intercept) | 0.04464 | 0.18562 | 0.24047 | 0.81172 |
| 2 | t_1 | year | 0.00001 | 0.00009 | 0.06617 | 0.94771 |
| 3 | t_10 | (Intercept) | 0.82787 | 0.35744 | 2.31609 | 0.02809 |
| 4 | t_10 | year | -0.00038 | 0.00018 | -2.12904 | 0.04218 |
| 5 | t_11 | (Intercept) | -1.90262 | 0.20979 | -9.06912 | 0.00000 |
| 6 | t_11 | year | 0.00098 | 0.00010 | 9.34952 | 0.00000 |
| 7 | t_12 | (Intercept) | 2.85298 | 0.38338 | 7.44158 | 0.00000 |
| 8 | t_12 | year | -0.00139 | 0.00019 | -7.26904 | 0.00000 |
| 9 | t_13 | (Intercept) | -1.83128 | 0.34144 | -5.36342 | 0.00001 |
| 10 | t_13 | year | 0.00095 | 0.00017 | 5.60242 | 0.00001 |

Showing 1 to 10 of 28 entries Previous 1 2 3 Next

Select number of term: 10

Select number of label: 1

Select assignments: 0.05

|  | topic | label_1 | coherence | prevalence | top_terms |
|---|---|---|---|---|---|
| t_1 | t_1 | molecular_weight | 0.02800 | 5.70500 | time, method, skin, obtained, min, methods, process, activity, potential, determined |
| t_2 | t_2 | blood_cells | 0.08300 | 5.31400 | cells, blood, cell, salinity, observed, histopathological, white, red, dna, number |
| t_3 | t_3 | business_media | 0.11300 | 4.54100 | carp, species, common, catfish, light, media, freshwater, business, business_media, carpio |
| t_4 | t_4 | water_quality | 0.08600 | 6.34200 | water, system, quality, aquaculture, systems, oxygen, water_quality, nitrogen, rice, temperature |
| t_5 | t_5 | control_group | 0.13100 | 7.37000 | control, group, groups, compared, activity, serum, supplemented, supplementation, parameters, immune |
| t_6 | t_6 | growth_performance | 0.35800 | 12.46900 | diet, fed, protein, diets, growth, feed, dietary, weight, body, levels |
| t_7 | t_7 | heavy_metals | 0.08400 | 8.75400 | species, samples, lake, collected, found, river, metals, human, fishes, fisheries |
| t_8 | t_8 | fatty_acids | 0.25600 | 4.91800 | acid, oil, fatty, acids, lipid, muscle, content, amino, composition, fatty_acids |
| t_9 | t_9 | gene_expression | 0.22300 | 9.23300 | expression, gene, genes, analysis, mrna, related, important, identified, role, sequence |
| t_10 | t_10 | body_weight | 0.11200 | 6.39800 | genetic, length, weight, gift, body, strain, data, model, red, populations |

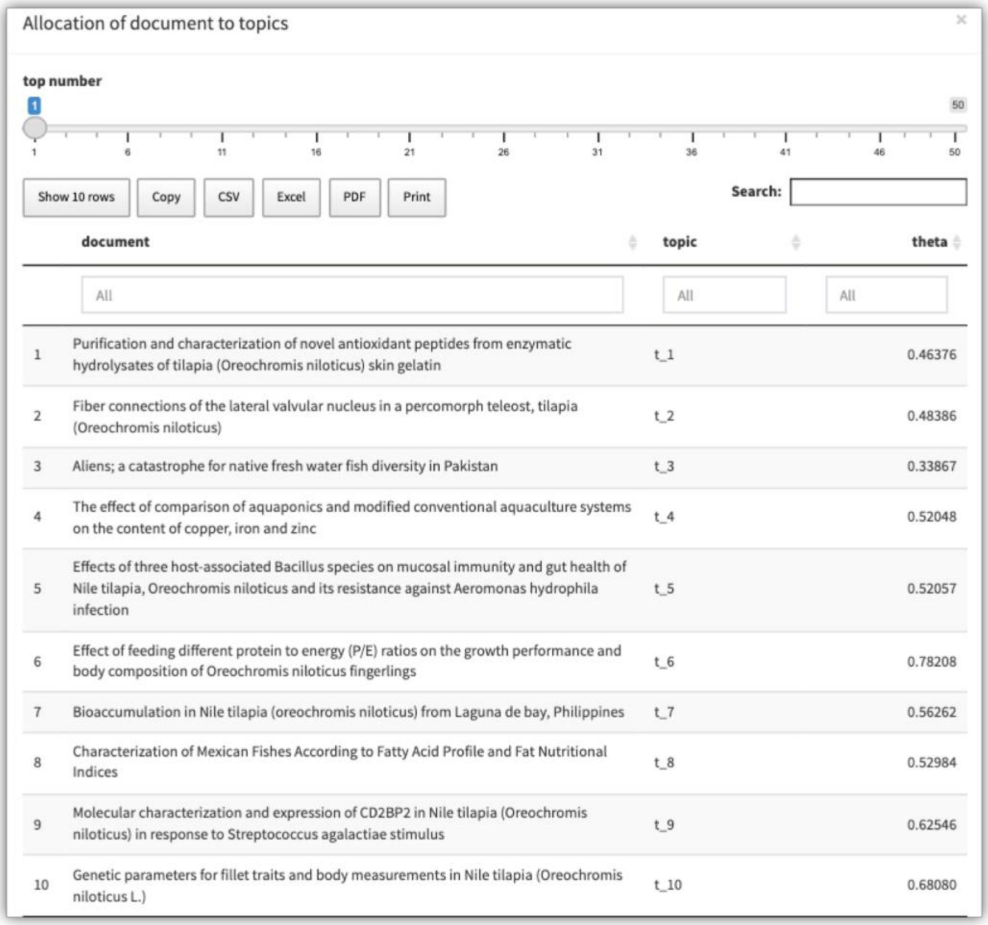
Showing 1 to 10 of 14 entries Previous 1 2 Next

**Figure 10.** Output tabular of the model Trends (**left**) summary (**right**).

## 5.4. Postprocessing

Among the main outputs of the topic modelling algorithm are the collection of terms in relation to the frequencies of occurrence that characterize a topic and the composition, in percentage terms, for each document that has been analyzed. The distribution of topic terms does not come with a semantic interpretation. However, the topics can be properly labeled in most cases, inferring from the word frequency.

LDAShiny provides a topic labeling using a naive n-gram based topic algorithm from the textmineR [14] Package. However, as indicated above, these algorithms have limited capacity, so it is recommended that the labeling be validated by an expert in the research area. If a domain expert is not available, it could generate incorrectly labeled topics and an invalid result [4].



Allocation of document to topics

top number

| | document | topic | theta |
|---|---|---|---|
| 1 | Purification and characterization of novel antioxidant peptides from enzymatic hydrolysates of tilapia (Oreochromis niloticus) skin gelatin | t_1 | 0.46376 |
| 2 | Fiber connections of the lateral valvular nucleus in a percomorph teleost, tilapia (Oreochromis niloticus) | t_2 | 0.48386 |
| 3 | Aliens; a catastrophe for native fresh water fish diversity in Pakistan | t_3 | 0.33867 |
| 4 | The effect of comparison of aquaponics and modified conventional aquaculture systems on the content of copper, iron and zinc | t_4 | 0.52048 |
| 5 | Effects of three host-associated Bacillus species on mucosal immunity and gut health of Nile tilapia, Oreochromis niloticus and its resistance against Aeromonas hydrophila infection | t_5 | 0.52057 |
| 6 | Effect of feeding different protein to energy (P/E) ratios on the growth performance and body composition of Oreochromis niloticus fingerlings | t_6 | 0.78208 |
| 7 | Bioaccumulation in Nile tilapia (oreochromis niloticus) from Laguna de bay, Philippines | t_7 | 0.56262 |
| 8 | Characterization of Mexican Fishes According to Fatty Acid Profile and Fat Nutritional Indices | t_8 | 0.52984 |
| 9 | Molecular characterization and expression of CD2BP2 in Nile tilapia (Oreochromis niloticus) in response to Streptococcus agalactiae stimulus | t_9 | 0.62546 |
| 10 | Genetic parameters for fillet traits and body measurements in Nile tilapia (Oreochromis niloticus L.) | t_10 | 0.68080 |

**Figure 11.** Output tabular allocations.

As a result, the 14 topics found reflect an overview of the research on the species *O. niloticus*. This shows one of the main benefits of the application, by providing information on a large collection of documents with relatively little effort on the part of the researcher.

After the label of the themes has been verified, the researcher can choose the articles that are relevant to the literature review. For example, if their main interest is in genetic expression, a specific number of articles on that topic can be selected by using the tabular output "allocation of document to topic."

LDAShiny allows the analysis of the dynamics of the topics over time in terms of their proportions, making it easier to understand the general trend of research. The increase in the proportion of some topics indicates that these are emerging fields of research, while their decrease shows a trend of less research interest. In addition, the high frequency pattern found at the beginning in some topics, which was followed by a negative trend during the period of study, has indicated a possible decrease in their popularity within the scientific community. This facilitates researchers not only to identify emerging research topics but also to visualize changes in the research focus.

The results obtained for the distribution of the topics by year, are also represented by a heatmap (Figure 12). In it, the color of the pixel represents the probability that a certain topic will be mentioned in a particular year.

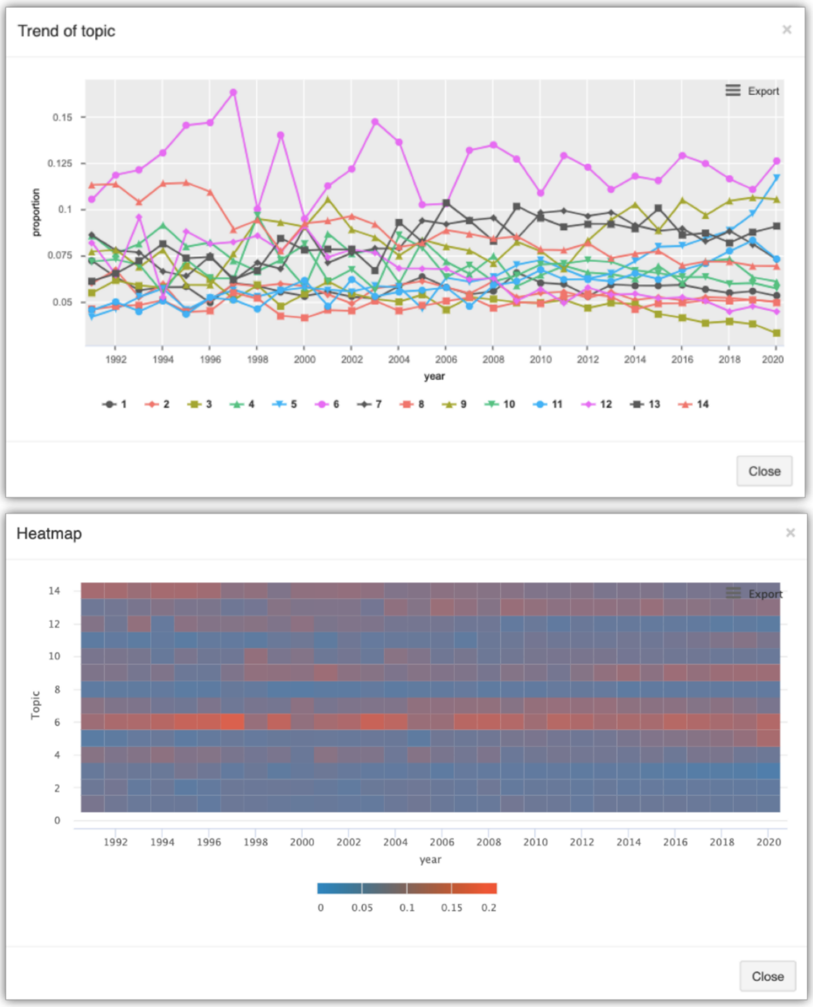


**Figure 12.** Graphical output of trend model.

## 6. Discussion

In 2004, Blei implemented the lda-c Software which was the first software that performs variational inference [6].

Commonly, available specialized open source software tools focus on stages or steps of text mining. Thus, they only focus for example on the preprocessing phase or in the inference phase. Some of these packages allow academics and researchers with a medium knowledge of a programming language (such as R) to follow the workflow required for an exploratory review of scientific literature. However, the available packages do not provide a GUI. In order to solve this problem, a R package with web-based GUI was developed in shiny, facilitating the execution of the exploratory review of scientific literature. Thus, LDAShiny facilitates the integral aspects of a review through LDA from preprocessing, inference (choosing between a set of models) and postprocessing (identifying trends in research). In addition, the information generated can be downloaded in various formats both in tabular and graph forms.

An additional benefit of LDAShiny is that it allows reproducibility, since all the steps of the exploratory review process can be reviewed and evaluated by other researchers in an agile and transparent way compared to a traditional review. In addition, the proposed application could be used to monitor the research trend. For instance, in the case of the example used, when more articles are published on the species under analysis, the review could be easily updated, since these new publications will be classified in related topics.

We found that the default parameters in the application example in the preprocessing steps offered a valid and usable result for the exploratory analysis of the literature on

*O. niloticus*. The execution time of the analysis did not take long, which is beneficial for the researcher. Usually, this time is mainly computer time and, although it is necessary to validate this verification, it requires less time than if a manual review were performed.

LDAShiny includes tools for undertaking an exploratory examination of scientific literature, as well as preprocessing features such as generating a corpus and removing stopwords, numbers and constructing ngrams. The tool also allows a document-term matrix to be created from a collection of documents, in a flexible manner, with a rudimentary understanding of the R programming language. Moreover, it facilitates researchers who are unfamiliar with R language to employ machine learning techniques. Users can point and click to generate a graphical or tabular of representation of the DTM matrix that can be downloaded in a variety of forms and saved and/or exported.

It is important to note that the preprocessing phase is an iterative process, as identifying stopwords, which might be difficult at initially [51,52] find that the preprocessing stages, in particular, can have a significant impact on the validity of the results, emphasizing the necessity of choosing the model parameters. However, for an exploratory study of the scientific literature, the default parameters and cleanup methods established in LDAShiny provide a legitimate and usable result.

In terms of inference, the app includes different metrics. While they are already available in R's CRAN in packages like topicmodel, ldatunning, and texmineR, the tool makes them easier to set by allowing them to be adjusted through easy-to-use interactive menus.

Although LDAShiny includes an algorithm for labeling, the identification of the topics is an important component of the post-processing phase. Because a mislabeled topic could lead to invalid results it is best if an expert reviews the labeling.

We might remark, for example, that one of the benefits of utilizing LDAShiny for a literature review is that the decision to include or delete articles can be postponed until a later stage when additional information is available, resulting in a decision-making process. Because all elements of the exploratory review process are reproducible, LDAShiny provides more reproducibility and transparency, allowing other researchers to analyze the entire review process in detail.

Although LDAShiny was evaluated in a study of academic scientific literature on the species *O. niloticus*, it is expected that researchers from various fields will put the tool to the test, as there is no technological reason why other types of documents cannot be included.

This is the first edition of the program. It is planned to add more features in future editions, such as the ability to read whole articles rather than just abstracts. This can improve the quality of the topics and provide more detail on latent themes [24].

## 7. Conclusions

In any scientific area, reviewing the scientific literature is a necessary step of the research process. As the number of publications increases over time, the task of acquiring knowledge becomes increasingly difficult.

This work aimed to present a tool, the LDAShiny package, that allow researchers to use topic modeling based on the use of the latent Dirichlet allocation. Thus, it is possible to perform an exploratory review of the literature, reducing the need to read articles manually and allowing the possibility to analyze a greater number of articles. The LDAShiny package was designed to be easily used by any researcher, as it requires less technical knowledge than using a normal topic model would imply.

LDAShiny development can also be addressed to the developer community, since the sources are published on GitHub (https://github.com/cran/LDAShiny), which allows the creation of shared development. The application can be run on a computer locally. Nonetheless, shiny can also be hosted on a server and deployed online.

There are options for preprocessing, inference, topic modeling and postprocessing in the application. The papers are loaded, cleaned, and authenticated during the preprocessing stage. The LDA approach is utilized in the inference step to estimate the number of topics that were used in the topic modeling phase. The post-processing step generates topic model results.

LDAShiny was designed with a step-by-step approach, and with a friendly interface allowing accessibility. However, researchers from various fields are expected to test it and provide valuable evaluations to improve its use.

The application was tested with 6196 scientific publications on the species *O. niloticus.* This data was processed in a short amount of time, taking roughly three days on a five-core laptop. The data were divided into 14 categories.

We consider LDAShiny to be especially relevant for researchers in various areas, as the literature review is essential for gaining an overview of the different research fields, where a shiny-based graphical user interface can allow more documents to be reviewed, more frequently. The LDAShiny package provides an interface that allows users to use the features interactively and in a friendly way, which can be used not only by statisticians but also by analysts who are unfamiliar with the R environment.

**Author Contributions:** Conceptualization, J.D.l.H.-M. and M.J.F.-G.; methodology, J.D.l.H.-M.; software, J.D.l.H.-M.; validation, S.M.; formal analysis, M.J.F.-G.; investigation, J.D.l.H.-M.; resources, S.M.; data curation, J.D.l.H.-M.; writing—original draft preparation, J.D.l.H.-M.; writing—review and editing, S.M. and M.J.F.-G.; visualization, J.D.l.H.-M.; supervision, M.J.F.-G.; project administration, J.D.l.H.-M.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially funded by FCT (Fundação para a Ciência e a Tecnologia) for the financial support of this work through the project UID/MAR/04292/2020, attributed to MARE-Marine and Environmental Sciences Centre, Portugal, and by the Integrated Programme of SR&TD "SmartBioR" (reference Cen-tro-01-0145-FEDER-000018), cofounded by the Centro 2020 program, Portugal2020, European Union, through the European Regional Development Fund.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The software is distributed through CRAN https://cran.r-project.org, and data and code will be made available on Github at https://github.com/cran/LDAShiny.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Brocke, J.; Simons., A.; Niehaves, B.; Niehaves, B.; Reimer, K.; Plattfaut, R.; Cleven, A. Reconstructing the giant: On the importance of rigour in documenting the literature search process. In Proceedings of the 17th European Conference on Information Systems, Verona, Italy, 7–9 June 2009; p. 161. Available online: http://aisel.aisnet.org/ecis2009/161 (accessed on 16 April 2021).
2. Harzing, A.W.; Alakangas, S. Google Scholar, Scopus and the Web of Science: A longitudinal and cross-disciplinary comparison. *Scientometrics* **2016**, *106*, 787–804. [CrossRef]
3. DiMaggio, P.; Nag, M.; Blei, D. Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of US government arts funding. *Poetics* **2013**, *41*, 570–606. [CrossRef]
4. Asmussen, C.B.; Muller, C. Smart literature review: A practical topic modelling approach to exploratory literature review. *J. Big Data.* **2019**, *6*, 93. [CrossRef]
5. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2020. Available online: https://www.R-project.org/ (accessed on 16 April 2021).
6. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet allocation. *J. Mach. Learn Res.* **2003**, *3*, 993–1022.
7. Chang, J. *lda: Collapsed Gibbs Sampling Methods for Topic Models*, R package version 1.4.2; R Foundation for Statistical Computing: Vienna, Austria, 2015. Available online: https://CRAN.R-project.org/package=lda (accessed on 16 April 2021).
8. Erskine, N. *lda.svi: Fit Latent Dirichlet Allocation Models Using Stochastic Variational Inference*, R package version 0.1.0.; R Foundation for Statistical Computing: Vienna, Austria, 2015. Available online: https://CRAN.Rproject.org/package=lda.svi (accessed on 16 April 2021).
9. Rieger, J. *ldaPrototype: Prototype of Multiple Latent Dirichlet Allocation Runs*, R package version 0.1.1; R Foundation for Statistical Computing: Vienna, Austria, 2015. Available online: https://CRAN.R-project.org/package=ldaPrototype (accessed on 16 April 2021).
10. Nikita, M. *ldatuning: Tuning of the Latent Dirichlet Allocation Models Parameters*, R package version 1.0.2.; R Foundation for Statistical Computing: Vienna, Austria. Available online: https://CRAN.R-project.org/package=ldatuning (accessed on 16 April 2021).

11. Sievert, C.; Shirley, K. *LDAvis: Interactive Visualization of Topic Models*, R package version 0.3.2.; R Foundation for Statistical Computing: Vienna, Austria. Available online: https://CRAN.R-project.org/package=LDAvis (accessed on 16 April 2021).
12. Friedman, D. *topicdoc: Topic-Specific Diagnostics for LDA and CTM Topic Models*, R package version 0.1.0.; R Foundation for Statistical Computing: Vienna, Austria. Available online: https://CRAN.R-project.org/package=topicdoc (accessed on 16 April 2021).
13. Grun, B.; Hornik, K. topicmodels: An R Package for Fitting Topic Models. *J. Stat. Softw.* **2011**, *40*, 1–30. [CrossRef]
14. Jones, T. *textmineR: Functions for Text Mining and Topic Modeling*, R package version 3.0.4.; R Foundation for Statistical Computing: Vienna, Austria, 2019. Available online: https://CRAN.R-project.org/package=textmineR (accessed on 16 April 2021).
15. Blei, D.M. Probabilistic topic models. *Commun. ACM* **2012**, *55*, 77–84. [CrossRef]
16. Kao, A.; Poteet, S.R. *Natural Language Processing and Text Mining*, 1st ed.; Springer Science & Business Media: London, UK, 2007; p. 265.
17. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by latent semantic analysis. *J. Assoc. Inf. Sci. Technol.* **1990**, *41*, 391–407. [CrossRef]
18. Hofmann, T. Probabilistic latent semantic indexing. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, 15–19 August 1999; pp. 50–57. [CrossRef]
19. Grimmer, J. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases. *Political Anal.* **2010**, *18*, 1–35. [CrossRef]
20. Jacobi, C.; Van Atteveldt, W.; Welbers, K. Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digit. J.* **2016**, *4*, 89–106. [CrossRef]
21. Iwata, T.; Saito, K.; Ueda, N.; Stromsten, S.; Griffiths, T.L.; Tenenbaum, J.B. Parametric embedding for class visualization. *Neural. Comput.* **2007**, *19*, 2536–2556. [CrossRef] [PubMed]
22. Wang, Y.; Sabzmeydan, P.; Mori, G. Semi-latent Dirichlet allocation: A hierarchical model for human action recognition. In Proceedings of the Second Workshop, Human Motion—Understanding, Modeling, Capture and Animation, Rio de Janeiro, Brazil, 20 October 2007; Elgammal, A., Rosenhahn, B., Klette, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2007. [CrossRef]
23. Griffths, T.L.; Steyvers, M. Finding scientific topics. *Proc. Natl. Acad. Sci. USA* **2004**, *101* (Suppl. 1), 5228–5235. [CrossRef]
24. Syed, S.; Spruit, M. Full-text or abstract? Examining topic coherence scores using latent dirichlet allocation. In Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 19–21 October 2017; pp. 165–174. [CrossRef]
25. Newman, D.; Smyth, P.; Welling, M.; Asuncion, A.U. Distributed inference for latent Dirichlet allocation. In Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 1081–1088.
26. Porteous, I.; Newman, D.; Ihler, A.; Asuncion, A.; Smyth, P.; Welling, M. Fast collapsed gibbs sampling for latent Dirichlet allocation. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 569–577.
27. Blei, D.M.; Jordan, M.I. Variational inference for Dirichlet process mixtures. *Bayesian Anal.* **2006**, *1*, 121–143. [CrossRef]
28. Teh, Y.W.; Newman, D.; Welling, M. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In Proceedings of the 20th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2007; pp. 1353–1360.
29. Wang, C.; Paisley, J.; Blei, D.M. Online variational inference for the hierarchical Dirichlet process. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Machine Learning Research, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 752–760. Available online: http://proceedings.mlr.press/v15/wang11a.html (accessed on 17 April 2021).
30. Vijayarani, S.; Ilamathi, M.J.; Nithya, M. Preprocessing techniques for text mining-an overview. *Int. J. Comput. Sci. Commun. Netw.* **2015**, *5*, 7–16.
31. Jurafsky, D.; Martin, J.H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*; Pearson: Hoboken, NJ, USA, 2008.
32. Manning, C.D.; Manning, C.D.; Schutze, H. *Foundations of Statistical Natural Language Processing*, 2nd ed.; MIT Press: Cambridge, MA, USA, 1999.
33. Luhn, H.P. The automatic creation of literature abstracts. *IBM J. Res. Dev.* **1958**, *2*, 159–165. [CrossRef]
34. Benoit, K.; Muhr, D.; Watanabe, K. *Stopwords: Multilingual Stopword Lists*, R package version 0.9.0; R Foundation for Statistical Computing: Vienna, Austria, 2017. Available online: https://CRAN.R-project.org/package=stopwords (accessed on 17 April 2021).
35. Porter, M.F. An algorithm for suffix stripping. *Programming* **1980**, *14*, 130–137. [CrossRef]
36. Yano, T.; Smith, N.A.; Wilkerson, J.D. Textual predictors of bill survival in congressional committees. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Montreal, QC, Canada, 3–8 June 2012; pp. 793–802.
37. Grimmer, J.; Stewart, B.M. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Anal.* **2013**, *21*, 267–297. [CrossRef]
38. Blei, D.M.; Lafferty, J.D. A correlated topic model of science. *Ann. Appl. Stat.* **2007**, *1*, 17–35. [CrossRef]
39. Sbalchiero, S.; Eder, M. Topic modeling, long texts and the best number of topics. Some problems and solutions. *Qual Quant.* **2020**, *54*, 1095–1108. [CrossRef]

40. Newton, M.A.; Raftery, A.E. Approximate Bayesian inference with the weighted likelihood bootstrap. *J. R. Stat. Soc. Series B. Stat. Methodol.* **1994**, *56*, 3–26. [CrossRef]
41. Roder, M.; Both, A.; Hinneburg, A. Exploring the space of topic coherence measures. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 31 January–6 February 2015; pp. 399–408.
42. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162. [CrossRef]
43. Arun, R.; Suresh, V.; Veni Madhavan, C.E.; Narasimha Murthy, M.N. On finding the natural number of topics with latent dirichlet allocation: Some observations. In *Advances in Knowledge Discovery and Data Mining. PAKDD 2010. Lecture Notes in Computer Science*; Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6118. [CrossRef]
44. Cao, J.; Xia, T.; Li, J.; Zhang, Y.; Tang, S. A density-based method for adaptive LDA model selection. *Neurocomputing* **2009**, *72*, 1775–1781. [CrossRef]
45. Deveaud, R.; SanJuan, E.; Bellot, P. Accurate and effective latent concept modeling for ad hoc information retrieval. *Doc. Numer.* **2014**, *17*, 61–84. [CrossRef]
46. Lewis, S.C.; Zamith, R.; Hermida, A. Content analysis in an era of big data: A hybrid approach to computational and manual methods. *J. Broadcast. Electron. Media.* **2013**, *57*, 34–52. [CrossRef]
47. Lau, J.H.; Grieser, K.; Newman, D.; Baldwin, T. Automatic labelling of topic models. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA; 2011; pp. 1536–1545.
48. Xiong, H.; Cheng, Y.; Zhao, W.; Liu, J. Analyzing scientific research topics in manufacturing field using a topic model. *Comput. Ind. Eng.* **2019**, *135*, 333–347. [CrossRef]
49. Chang, W.; Cheng, J.; Allaire, J.; Xie, Y.; McPherson, J. *Shiny: Web Application Framework for R*, R package version 1.4.0.2; R Foundation for Statistical Computing: Vienna, Austria. Available online: https://CRAN.R-project.org/package=shiny (accessed on 17 April 2021).
50. Chang, J.; Gerrish, S.; Wang, C.; Boyd-Graber, J.L.; Blei, D.M. *Reading tea leaves: How humans interpret topic models. Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2009; pp. 288–296.
51. Denny, M.J.; Spirling, A. Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. *Political Anal.* **2018**, *26*, 168–189. [CrossRef]
52. Maier, D.; Waldherr, A.; Miltner, P.; Wiedemann, G.; Niekler, A.; Keinert, A.; Adam, S. Applying LDA topic modeling in communication research: Toward a valid and reliable methodology. *Commun. Methods Meas.* **2018**, *12*, 93–118. [CrossRef]