DOI: 10.1002/minf.201600035

Biological Web Service Repositories Review

David Urdidiales-Nieto,[a] Ismael Navas-Delgado,[a] and José F. Aldana-Montes*[a]

Abstract: Web services play a key role in bioinformatics enabling the integration of database access and analysis of algorithms. However, Web service repositories do not usually publish information on the changes made to their registered Web services. Dynamism is directly related to the changes in the repositories (services registered or unregistered) and at service level (annotation changes). Thus, users, software clients or workflow based approaches lack enough relevant information to decide when they should review or re-execute a Web service or workflow to get updated or improved results. The dynamism of the repository could be a measure for workflow developers to re-check service availability and annotation changes in the services of interest to them. This paper presents a review on the most well-known Web service repositories in the life sciences including an analysis of their dynamism. Freshness is introduced in this paper, and has been used as the measure for the dynamism of these repositories.

Keywords: web services · dynamism · detection of changes · frequency of changes · freshness · curation

1 Introduction

Today Web services are widely used in the context of the life sciences and there are many platforms that use them to exchange data over the Internet. However, the main interest in Web services is their capacity for interconnection. In this domain, Web services are usually connected using workflows. There are many proposals for developing scientific workflows, but Taverna,[1] Triana[2] and Kepler[3] are the most popular in bioinformatics. These tools require the discovery of and access to different Web services. This was a research challenge that was solved with Web service repositories.

Thus, several repositories have appeared in recent years for bioinformatics Web services. The most well-known repositories are BioMOBY,[4] EMBRACE[5] and BioCatalogue.[6] However, ELIXIR[7] is positioning itself as the most promising repository for bioinformatics tools. These repositories provide human readable information about Web services (such as status and availability and monitoring). However, the information and functionalities provided in each repository may differ. In this paper we review these features for four different repositories: BioMOBY, EMBRACE, BioCatalogue and ELIXIR.

The features are analysed both qualitatively and quantitatively. A quantitative analysis focuses on dynamism, which is directly related to the changes produced in the repositories (service registered or unregistered) and at service level (annotation changes). This aspect is relevant because Web services in the life sciences are subject to changes due to improvements in their software or updates in the data used by the service. In this context, Web services are provided by independent entities, so it may be the case that a given service stops working for a while or even permanently. It is also common that a service is provided by different entities (for example, a blast service). Thus, application developers would want to know which of the different instances of the same service would be the most reliable.

This paper proposes a methodology to measure the Web service's dynamism. This measurement is based on the freshness concept that is explained further on.

This paper considers known bioinformatics Web service registries, but the measuring of dynamism could be useful for other kinds of registries, such as Workflow repositories (myExperiment[8]), Bio–ontologies repositories (BioPortal[9]), repositories of resources (Expasy[10]) or databanks (such as DDBJ,[11] GenBank[12] or UniProt[13]), etc. However, these registries are beyond the scope of this study. Although we have applied the study to existing and public repositories of bioinformatics Web services, we believe that the theoretical study is useful to the reader owning a private repository or for the analysis of an unpublished repository prior to its publication.

The rest of the paper is organised as follows. Section 2 provides the background. In Section 3 the Web service repository qualitative comparison is presented. The quantitative comparison based on the freshness is described in Section 4. Finally, in Section 5, a discussion of the results and conclusions are given.

2 Background

In 2002, BioMOBY (http://www.biomoby.org/) emerged as one of the first open source biological Web service proposals. It aimed to generate an architecture for the discovery and distribution of biological data through Web services. Web services were registered in a central location, MOBY Central. This proposal includes strongly typed services, which have been further categorised according to the BioMOBY ontology (http://www.biomoby.org/). A service monitoring component has been developed to remove unresponsive services.

Although the life sciences community embraced the use of Web services, in 2006, Hull et al.[14] identified many prob-

[a] *D. Urdidiales-Nieto, I. Navas-Delgado, J. F. Aldana-Montes*
Department of Computer Languages and Computing Science, Higher Technical School of Computer Science Engineering, University of Malaga, Malaga, 29071, Spain
**e-mail: jfam@lcc.uma.es*

lems with the use of these Web services. In an effort to address these problems and to provide the life sciences community with a collection of coherent and robust bioinformatics Web services, the European Commission provided funds to establish the EMBRACE project (European Model for Bioinformatics Research and Community Education). During the 5-year project, many Web services for the life sciences were developed, defining a standard for Web service technology, for semantically annotating both Web service functions and data exchanged, and a mechanism for discovering Web services. The EDAM ontology for describing life sciences' Web services and the BioXSD schema for exchanging data between Web Services were created.

With the publication of BioCatalogue (http://www.biocatalogue.org/), currently there are more than 2000 Web services available. This registry provides information on available Web services (EMBRACE Web services are going to be transferred to the BioCatalogue system). BioCatalogue provides an interface for registering, browsing and annotating Web services for the life science community, aggregating life science-specific content from other sources and classifying it according to an ontology. So, Web services can be described and searched in multiple ways, such as those based on their technical types, bioinformatics categories, user tags, Web service providers and data inputs and outputs. The Web services are also constantly monitored, and the system is accessible by any member of the scientific community, so they can make comments or annotations on any Web service, which are then verified by a curator.

Finally, in 2015 a recent registry appeared, ELIXIR (https://biotools/). The ELIXIR Tools and Data Services Registry is a gateway to databases and tools for life science data analysis. The registry is being built, in collaboration with the key resource providers, on a federated curation model which helps resource providers curate their own resources. This model decentralises the curation burden and should not only lead to a registry that is more durable, but one that is of higher quality because it leverages the knowledge of the resource providers. The registry provides a comprehensive, up-to-date catalogue of resources that are interactive, downloadable and offer programmatic access.

David Urdiales-Nieto is a PhD student of the University of Málaga. Spain. His research, in the Khaos Research group, focuses on Web Services in Life Sciences ad statistical analysis. Currently, he is working in Big Data approaches in Life Sciences.

Ismael Navas-Delgado received a PhD in Computer Science and a MSc in Celular and Molecular Biology, both at the University of Málaga. He is Assistant Professor at the University of Málaga, Spain. His research, at the Khaos Research Group, focuses on the use of Semantics and Big Data technologies in Life Sciences and Precision Medicine.

José F. Aldana-Montes received a PhD in Compouter Science and is Full Professor at the University of Málaga, Spain. He is the main researcher of the Khaos Research Group. His research focuses on management, integration and analysis of data.

3 Qualitative Review of Web Service Repositories

This section describes the set of features a good Web service repository should have, from a qualitative point of view. These features are analysed in four repositories: BioMOBY, EMBRACE, BioCatalogue and ELIXIR.

Presentation of general information. Repositories can provide general information on their registered services. This information is useful for end users and developers to understand how this service works. We have identified the following items of interest:

- General service information (name, location, description);
- Input and output requirements for the service's execution;
- Web service categories;
- Keywords;
- Provider information (name, location, etc.);
- Web service type;
- Web service monitoring (Web service status);
- Web service reliability, stability and consistency components;
- Event history and event type produced in each Web service.

Annotations provided or annotation tools. Web services can be annotated in different ways. Annotations can be associated with different entities: providers, other registers or named curators. These annotations can also be provided in different formats: structured data, free text, tags or ontology terms. For this reason, we have placed the annotations into five categories:

- Functional annotation: type of analysis;
- Data exchanged: schema for exchanging data between Web services);
- Operational: mechanisms and conditions necessary to execute a service;
- Profile: annotation capabilities depending on the user profile;
- Provenance: details of the provider and the location of the service.

Social networking and curation. This feature includes information based on the user experience with the repository. Thus, it includes Web service popularity based on view statistics, comments and the number and quality of the annotations.

Syntactic or semantic search capabilities. Web service repositories provide search capabilities that are usually based on annotations, tags or provider documentation. These search engines can be based on syntactic or semantic solutions.

Web service execution and workflow design and execution. These repositories can include functionalities not only for discovering services, but also for using them:

- Instructions on how to test and run services.
- Execution of Web services.
- Dynamic connection of Web service executions.
- Design of workflows based on registered services.
- Execution of workflows, usually delegated to workflow execution engines, such as Taverna or Kepler.

Statistical information at repository level. Repositories can also provide aggregated information at the repository level, such as Last Activity, Last Monitoring, Last Services registered and Service modifications, such as updated or deleted services.

3.1 Qualitative Review

Most of these repositories lack at least some of the features described in the previous section. Tables 1 and 2 summarise just how many of these features they include.

The BioMOBY Web Service repository lacks most of the features provided by novel Web service repositories. This is because it was one of the first repositories to be developed, and has not been brought up to date.

The EMBRACE Web Service Repository does not have the sophisticated social curation, tagging, browsing and searching facilities that more recent Web service repositories have. This repository is not being updated, as its services are currently being migrated to BioCatalogue, and so will be soon deprecated.

BioCatalogue is one of the most recent repositories, with a high level of activity as we will show in the following sections. However, it lacks some important features such as event history (to know when and how often changes are produced) and a recommendation system based on data produced by service executions.

4 Quantitative Review

This section analyses a quantitative measure of the repository dynamism: the freshness. The freshness of a repository can be measured using the information about the changes made in the registered Web services. Thus, it can include

Table 1. Capabilities and features of each Web service of the Web service repositories in the life sciences.

Web Service Repository	General service information (name, location, description)	Input and output requisites for the service execution	Web service categories	Keywords	Provider information (name, location, etc.)	Web service type	Web service monitoring (Web service status)	Web service reliability, stability and consistency components	Event history and Event type produced in each Web service
BioMOBY	X	X	X	X		X			
EMBRACE	X	X	X	X		X	X		
BioCatalogue	X	X	X	X	X	X	X		

Table 2. Capabilities and features of the Web service repositories in the life sciences.

Web Service Repository	Information presentation	Annotations provided or annotation tools	Social networking capabilities	Syntactic or semantic search capabilities	The capability of the Web services repository to offer Web service execution and workflow design and execution
BioMOBY	X			X	
EMBRACE	X				
BioCatalogue	X	X	X	X	X

the information about inserted, modified or deleted services over a given period of time. However, this information is not usually provided by the repositories.

The use of existing repositories is not limited to end users (humans searching for services to be included in workflows or applications), as they usually provide programmatic interfaces for accessing the information. In the cases where such interfaces are not available, we can use crawlers to extract the information directly from the human-oriented interfaces. In this paper, a crawling approach has been developed to track the changes in the different repositories.

The types of changes that can be detected are:

- An insertion or elimination of a service in a repository;
- A change in the service description (metadata of WSDL document for SOAP Web Services and metadata of XML document of the URI for REST services).

This section describes how these measurements can be calculated in a general scenario, and the results of calculating them over the repositories being analysed: BioMOBY, EMBRACE, BioCatalogue and ELIXIR.

4.1 Crawling Web Service Repositories

This section describes the process of collecting information from the different repositories. Firstly, a crawling process is initiated, to get data from each repository, indexing the list of repositories and their services. Secondly, the crawling process, after a defined, scheduled time (once a week for example) obtains data from each repository and analyses the changes produced in this time interval. Thus, we have the number and types of modifications in periodic time intervals.

The crawling process is centralised in a software application that opens different threads to crawl the different repositories. The crawling of each repository is an ad hoc solution based on the interfaces provided by the repository. Generally, the repositories display API interfaces to allow external software integration. This crawler has been developed to address different access mechanisms: REST APIs and SOAP Web services. For each crawler execution, the list of services is provided by these services, and for each service the available metadata is retrieved. In this step the crawler can find different service statuses:

- **New service registered**: service that was not previously indexed.
- **Service unregistered**: service previously indexed but not in the current list.
- **Service unavailable**: a service is not currently accessible.
- **Updated service**: the service metadata has changed after the previous crawling process.
- **Service not modified**: The service is accessible and its metadata has not changed.

The crawler is executed using a Pentium Dual-Core, 2.8 GHz CPU, 3 GB of main memory, a SATA 160 GB disk and 100 Mbps of internet connection. The complete crawling process usually takes around 3 hours.

4.2 Web Service Freshness

This section describes how to measure the changes produced in Web service repositories.[15] defines freshness as the fraction of the local database that is up-to-date. Based on this concept, we have defined the freshness for Web services. The higher its freshness the more dynamic a repository is. We define the freshness as: *the percentage of a repository that has changed in a given period of time.*

Given a number of services changed in a time interval (Xi) in a repository, the individual freshness of a repository with N services is calculated as Xi/N. However, we aim to calculate the repository freshness over time. Thus, we can calculate frequency of change[16][17][18] (λ) in a set of measurements in a temporal interval. In this case the freshness is calculated as: $freshness = \gamma/N$. In this way, we can calculate the freshness (using the frequency of changes) and estimate the time needed to detect (in days for example) a percentage of changes in each repository, based on the freshness value.

We define λ as $(X/n{*}I)$, being X the number of changes, I a defined interval of time (for example one week) and n the number of intervals. For example, if we set up an experiment measuring the number of changes and we get 140 changes (X) produced over 30 weeks (n), the *frequency of changes* is $\lambda = (140/30{*}\ 1) = 140/30 = 4.66$ *changes/week.* This is an absolute value, and because the total number of services in each repository is different, the *freshness (relative value)* is defined as $freshness = \lambda/N$, where N is the total number of services in the repository.

4.3 Experimental Results

In this section we introduce a set of experiments to test the freshness measurement in the repositories indicated. Apart from the freshness calculation we have included the use of the typical deviation (s). The typical deviation could have a high value if there are a high number of inserted, modified or deleted events at any given time. The changes produced in a Web service repository are generally lower than the calculated frequency of change values λ, and therefore, lower than their estimated freshness value, if these isolated events had not occurred.

Once the freshness of a Web service repository has been estimated, we can use it to decide whether the frequency of measurements or revisits of the repository would find changes.

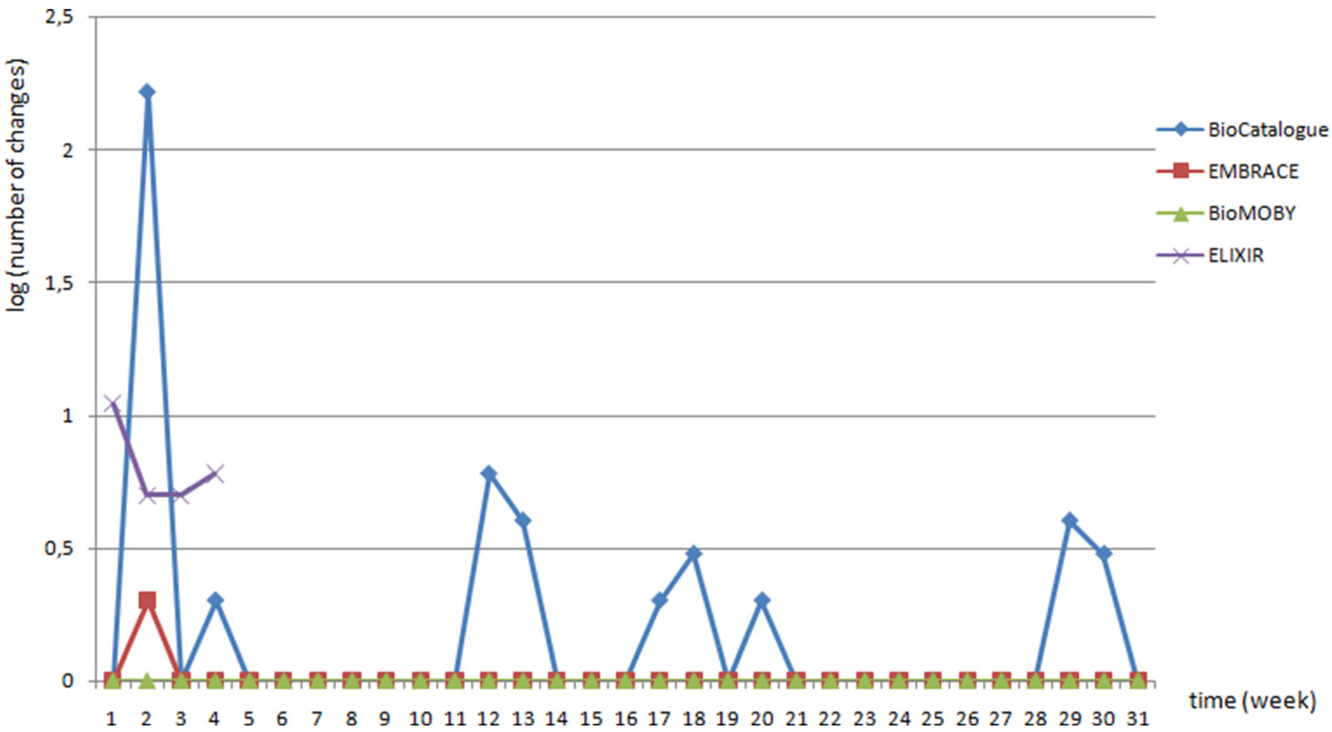

Figure 1. Number of changes after each experiment. It is clear that BioCatalogue had an isolated event when most of the services were registered and after that the number of changes remains quite low. In the case of ELIXIR the changes until now have remained quite constant. The Y axis uses a logarithmic scale as the changes are of different magnitude orders for BioCatalogue and ELIXIR.

Figure 1 shows the results of the experimental tests over BioMOBY, EMBRACE, and BioCatalogue. Each experiment was carried out with a fixed interval time between each use of the update process of one week (I = one week). An analysis of changes was carried out once a week for eight months.

For the BioMOBY Web Service repository (with N = 1443 Web service files) the frequency of changes λ is approximately 0.03 changes per week, and it has a freshness value of $2.08*10^{-3}$ % changes per week. The typical deviation is quite low (0.18) which means that there are no large update processes.

For the EMBRACE Web Service repository (with N = 808 Web service files) the frequency of changes is approximately 0.06 changes per week, and it has a freshness value of $7.42*10^{-3}$ % changes per week. The typical deviation is quite low (0.35) which means that there are no large update processes.

For the BioCatalogue Web service repository the frequency of changes is approximately six changes per week. The BioCatalogue Web service repository has N = 2165 SOAP and REST Web service files, so the estimation of its Web service repository is 0.291 % changes per week. In this case the typical deviation has a high value (29.31), which means that a large number of isolated modifications were produced. Therefore, a significance test, a t-test, has been carried out to check whether these calculated frequency of changes (λ), and therefore, their estimated freshness values, would have had a lower value than that presented in this table, if these events had not occurred.

Finally, due to the newness of ELIXIR the number of experiments is lower than for the other three repositories. An analysis of changes was carried out once a week for a month. For the ELIXIR repository (with N = 2532 Web service files) the frequency of changes λ is approximately 6.75 changes per week, and it has a freshness value of 0.26 % changes per week. In this case the typical deviation has an average value of 2.59. This is due to the fact that the number of experiments is low, rather than a large number of isolated modifications.

The Web services of the BioMOBY and EMBRACE repositories are not classified in categories; however, currently many Web services of BioCatalogue (622 of 2165 Web services) are classified in categories. Therefore, BioCatalogue has been tracked to detect these individual changes. Table 3 shows an extended study, which reveals the evolution of inserts and changes (changes in metadata-operations, port, etc.) produced over the lifetime of BioCatalogue, divided into semantic categories. It shows that each category of BioCatalogue has had inserts of new Web services and changes in Web services over the last 3 years. It also shows the freshness of the experimental test over BioCatalogue divided into categories, where the highest values are found in the text mining, ontology, cheminformatics, genomics, system biology and sequence analysis categories.

Table 3. Evolution of *Inserts + Changes* (*freshness*) over the lifetime of BioCatalogue, divided into categories.

Categories (Web Services)	Inserts + Changes (freshness) produced < 1 year ago	Inserts + Changes produced over 1 year ago	Inserts + Changes produced over 2 years ago	Inserts produced over 3 years ago
Sequence Analysis (229)	4 + 5 (*freshness* = 0.126 % per week) %)	59 + 3	36 + 101	110
Text Mining (35)	2 + 4 (*freshness* = 0.553 % per week)	12 + 3	11 + 6	10
Ontology (29)	1 + 1 (*freshness* = 0.222 % per week)	10 + 1	7 + 1	11
Phylogeny (8)	0 + 0 (*freshness* = 0 % per week)	3 + 0	1 + 2	4
Microarrays (44)	0 + 0 (*freshness* = 0 % per week)	2 + 0	2 + 22	40
Data Retrieval (164)	2 + 0 (*freshness* = 0.039 % per week)	12 + 3	74 + 102	76
Genomics (21)	0 + 1 (*freshness* = 0.154 % per week)	4 + 0	8 + 5	9
Proteomics (36)	0 + 0 (*freshness* = 0 % per week)	6 + 0	19 + 15	11
System Biology (24)	0 + 1 (*freshness* = 0.134 % per week)	10 + 1	7 + 2	7
Biostatistics (15)	0 + 0 (*freshness* = 0 % per week)	1 + 0	2 + 9	12
Chemoinformatics (17)	1 + 0 (*freshness* = 0.189 % per week)	4 + 0	10 + 2	2

5 Discussion

The dynamism of the analysed Web service repositories is quite low in three of them and quite high for ELIXIR. Thus, there is a great difference between the original repositories which now have less activity and the most recent ones where the activity is clearly higher. This is also notable in the qualitative features, where the most recent repositories tend to introduce more user oriented tools to help not only in discovering the registered services but also using them.

This study also reflects isolated events with a relatively large number of changes. This issue is relevant for those who wish to discover the behaviour of these repositories and so individual changes are also analysed. This also reflects that the changes are usually produced by providers with a large number of services, and that only a small number of providers have few services.

The individual statistics of the service are also relevant, and they have also been taken into account in this proposal. An experimental study divided into categories has been done, and can be used in the decision making process. This

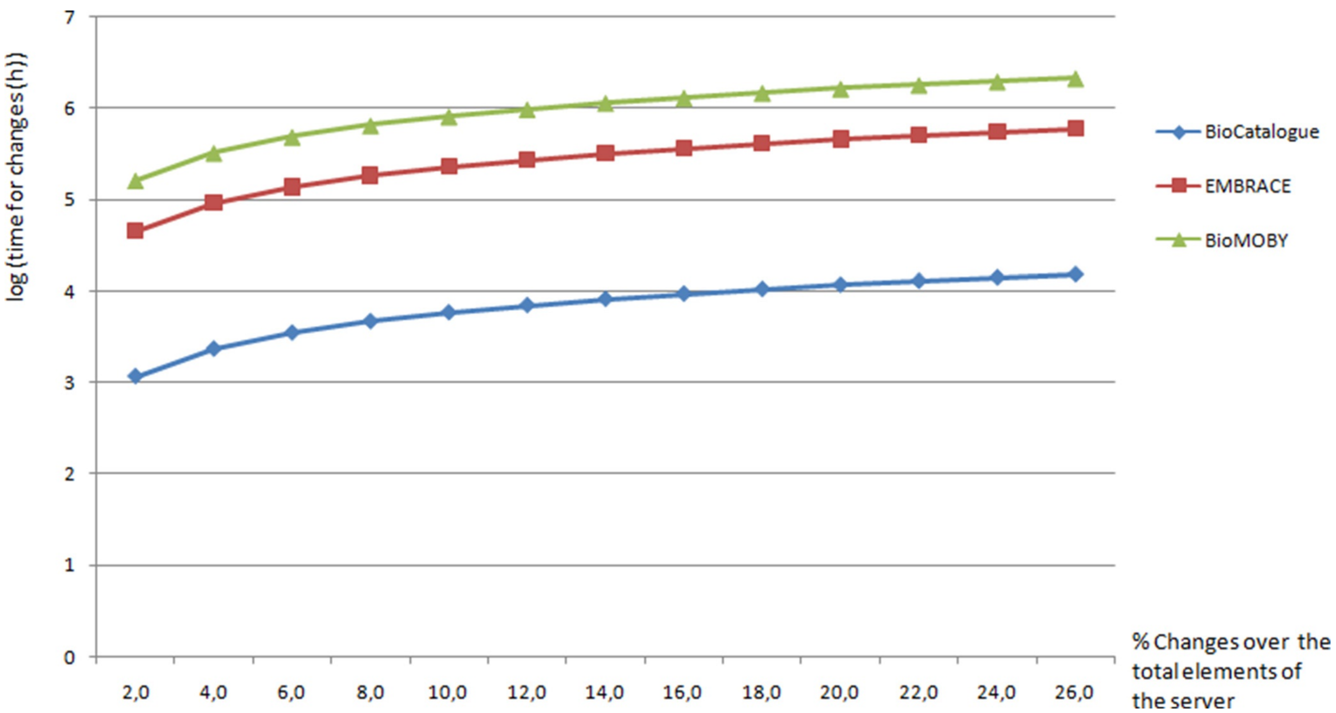

Figure 2. Time spent (hours) to detect a % of changes over total number of elements of each Web service repository. This figure plots the time spent (in a logarithmic scale) to detect a percentage of changes over total elements of each Web service repository. The Y axis uses a logarithmic scale as the changes are of different magnitude orders for BioCatalogue and ELIXIR. So, it shows that, for example, if we wanted to detect 26 % of changes over the total elements of BioCatalogue, the crawler process could be postponed more than $1*10^4$ hours, more than $5*10^5$ hours for EMBRACE and more than $2*10^6$ hours for BioMOBY.

study, divided into categories, shows that some categories are more dynamic than others. Thus, text mining services are quite dynamic which would reflect that current research is producing new services or improvements over existing ones.

The freshness estimation can be also used in the crawler to provide dynamic scheduling for the analysis process. Therefore, it could launch the crawler to detect changes produced at the moment they are more likely to occur. Considering the freshness of a Web service repository, Figure 2 plots the time spent to detect a percentage of changes over total elements of each Web service repository. Therefore, the crawler process can be postponed to ensure the detection of the change of a percentage of the repository. Thus, for a smaller freshness, the crawling can be delayed for longer. Table 4 shows the time spent to detect the change of a percentage of the total elements of each Web service repository. So, for example, if we wanted to detect 2% of changes in the total elements of BioCatalogue, the crawler process could be postponed 1154.64 hours (48.11 days).

Table 4. Time that has to be spent (in days) to detect the percentage of changes in each repository.

Changes (% over Total)	BioMOBY Web Service repository	EMBRACE Web Service repository	BioCatalogue Web Service repository
2.0	6730.77	1886.79	48.11
4.0	13461.53	3773.58	96.22
6.0	20192.31	5660.38	144.33
8.0	26923.08	7547.17	192.44
10.0	33653.84	9433.96	240.55
12.0	40384.62	11320.75	288.66
14.0	47115.38	13207.55	336.77
16.0	53846.16	15094.34	384.88
18.0	60576.92	16981.13	432.99
20.0	67307.70	18867.92	481.10
22.0	74038.46	20754.72	529.21
24.0	80769.23	22641.51	577.32
26.0	87500.00	24528.30	625.43

6 Conclusions

Web service repositories have evolved towards greater accessibility via a human-readable 'Web 2.0'-style interface and other capabilities, such as new characteristics to analyse them. In this paper, we have reviewed the repository features from both qualitative and quantitative points of view.

The quantitative review has been based on an approach to measure the freshness of a repository, which gives us a measure of the dynamism of a given Web service repository.

With the increase of the use of Semantic Web technologies, it is expected that Web service repositories in the life sciences will be upgraded to include the semantic annotation of Web services. Thus, ontologies are being used to describe and enable the semantic search of Web services in multiple ways.

Web services should also be constantly monitored to ensure that the services being used in the applications are the most up-to-date versions or even whether they are in fact still working. However, at the moment Web service repositories do not offer information on the freshness (dynamism). This information can be useful for crawlers, expert curators or for any kind of tool or workflow based approach, to decide how often a Web service repository needs to be revisited. The dynamism of the repository could also be a way for workflow developers to re-check service availability and annotation of the services of interest to them. The dynamism of the repository will also affect the trust placed in it to choose services for a workflow. Thus, for a given set of repositories, a workflow developer may choose the least dynamic one to ensure a more enduring workflow.

Conflict of Interest

None declared.

Acknowledgements

This work was funded in part by Grants TIN2014-58304-R (Ministerio de Ciencia e Innovación) and P11-TIC-7529 and P12-TIC-1519 (Plan Andaluz de Investigación, Desarrollo e Innovación).

References

[1] T. Oinn, M. Greenwood, M. Addis, et al. *Concurrency and Computation: Practice and Experience* **2006**, *18*, 1067–1100.
[2] D. Churches, G. Gombas, A. Harrison, et al. *Concurrency and Computation: Practice and Experience* **2006**, *18*, 1021–1037.
[3] B. Ludaescher, I. Altintas, C. Berkley, et al. *Concurrency and Computation: Practice and Experience* **2006**, *18*, 1039–1065.
[4] M. D. Wilkinson and M. Links, *Brief. Bioinformatics* **2002**, *3*, 331–341.
[5] S. Pettifer, J. Ison, M. Kalas, et al. *Nucleic Acids Research* **2010**, *38*, 683–688.
[6] J. Bhagat, F. Tanoh, E. Nzuobontane, et al. *Nucleic Acids Research* **2010**, *38*, 689–694.
[7] H. Ménager et al. *International Journal on Software Tools for Technology Transfer* **2015**, *17*, 1–6.
[8] C. A. Goble, J. Bhagat, S. Aleksejevs, et al. *Nucleic Acids Research* **2010**, *38*, 677–82
[9] D. L. Rubin, D. A. Moreira, P. P. Kanjamala, BioPortal: A Web Portal to Biomedical Ontologies. Association for the Advancement of Artificial Intelligence, **2007**.
[10] P. Artimo, M. Jonnalagedda, K. Arnold, et al. *Nucleic Acids Research* **2012**, *40*, 597–603.
[11] Y. Tateno, T. Imanishi, S. Miyazaki, et al. *Nucleic Acids Research* **2002**, *30 (1)*, 27–30.

[12] D. Benton et al. *Nucleic Acids Research* **2006**, *34*, 16-D20.
[13] UniProt Consortium. *Nucleic Acids Research* **2012**, *40*, D71–D75.
[14] D. Hull, K. Wolstencroft, R. Stevens, et al. *Nucleic Acids Research* **2006**, *34*, 729–732.
[15] C. Junghoo, H. Garcia-Molina, Synchronizing a database to improve freshness. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00)* **2000**. ACM, New York, NY, USA, 117–128.
[16] J. Cho, H. Garcia Molina, *Internet Technol.* **2003**, *3*, 256–290.
[17] J. Cho, H. Garcia Molina, The evolution of the web and implications for an incremental crawler. In: *Proceedings of the 26th VLDB Conference*, Cairo, Egypt, **2000**.
[18] J. Umbrich, M. Hausenblas, A. Hogan, et al. Toward Dataset Dynamics: Change Frequency of Linked Open Data Sources. In *LDOW* **2010** Raleigh, USA.

Received: March 7, 2016
Accepted: September 27, 2016
Published online: October 26, 2016

