BioHackEU24 report: Integrating Bioconductor packages with the ELIXIR Research Software Ecosystem using EDAM

Claire Rioualen[1], **Aurélien Barre**[2], **Vincent J Carey**[3], **Benjamin Dartigues**[2], **Matúš Kalaš**[4], **Sebastian Lobentanzer**[5], **Hervé Ménager**[1, 6], **Steffen Neumann**[7, 8], **Kozo Nishida**[9], **Veit Schwämmle**[10], **Anh Nguyet Vu**[11], **Egon Willighagen**[12], **and Maria A Doyle**[13]

1 Institut Français de Bioinformatique, CNRS UAR 3601, Évry, France **2** University of Bordeaux (CBiB): Bordeaux, Nouvelle-Aquitaine, France **3** Channing Division of Network Medicine, Mass General Brigham, Harvard Medical School, Boston MA, 02115 USA **4** ELIXIR Norway, and Department of Informatics, University of Bergen, Norway **5** Heidelberg University, Faculty of Medicine and Heidelberg University Hospital, Institute for Computational Biomedicine, Heidelberg, Germany **6** Institut Pasteur, Université Paris Cité, Bioinformatics and Biostatistics Hub, 75015, Paris, France **7** Leibniz Institute of Plant Biochemistry, MetaCom Center, Computational Plant Biochemistry, Halle (Saale), Germany **8** German Center for Diabetes Research, Munich, Germany **9** RIKEN Center for Biosystems Dynamics Research, Kobe, Japan **10** Department of Biochemistry and Molecular Biology, University of Southern Denmark, Odense, Denmark **11** Sage Bionetworks **12** Dept of Translational Genomics, NUTRIM, FHML, Maastricht University, Maastricht, The Netherlands **13** Limerick Digital Cancer Research Centre, School of Medicine, University of Limerick, V94 T9PX, Ireland

BioHackathon series:
BioHackathon Europe 2024
Barcelona, Spain, 2024
Project 27

Submitted: 07 Apr 2025

Published by BioHackrXiv.org

Abstract

This project seeks to enhance the ELIXIR Research Software Ecosystem (RSEc) by increasing the findability, accessibility, interoperability, and reusability (FAIR principles) of Bioconductor's extensive collection of over 2,000 bioinformatics packages. By aligning Bioconductor metadata with the EDAM ontology and integrating detailed package descriptions into the *bio.tools* registry, we aim to improve the discoverability and usability of bioinformatics analysis tools. Short-term goals include mapping Bioconductor's biocViews controlled vocabulary to EDAM concepts, developing a set of manually annotated "gold standard" packages, and evaluating tools for automated EDAM concept suggestions. Long-term, we intend to expand EDAM coverage across Bioconductor, phase out biocViews, and implement automated synchronisation with *bio.tools*. This initiative fosters collaboration between Bioconductor and ELIXIR, establishing a foundation for sustainable software management in European bioinformatics.

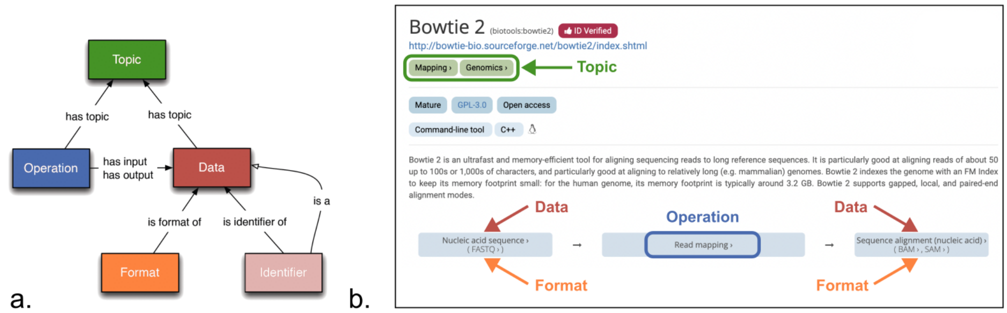
Key results from the ELIXIR BioHackathon 2024 week include substantial progress in mapping the biocViews vocabulary to EDAM concepts, initiating the curation of a reference set of packages with manual annotations, integrating Bioconductor metadata into the ELIXIR Research Software Ecosystem (RSEc) with automated updates, and prototyping a tool for automated EDAM concept suggestions. Together, these achievements establish a strong foundation for further integration and refinement.

Introduction

Background

Bioconductor (Gentleman et al., 2004; Huber et al., 2015) is a global open-source project that provides over 2,000 packages for biological data analysis within the R programming environment. Supporting a vast range of bioinformatics applications, these tools are widely used for gene expression analysis, visualisation and data integration. Now in its third decade, Bioconductor is thus a critical resource for life science research. However, the expanding bioinformatics landscape calls for enhanced methods to locate and apply appropriate tools, especially in complex workflows. All resources managed by the project are quality-controlled and revised every release. New resources are reviewed and added, and obsolete or unmaintained resources are dropped in a formal deprecation process with a six-month cadence.

The ELIXIR Research Software Ecosystem (RSEc) (Ienasescu et al., 2023) promotes the FAIR principles—Findability, Accessibility, Interoperability, and Reusability—, through the centralisation and curation of metadata for computational biology software tools. Some of its main components are the *bio.tools* registry (Ison et al., 2019) and the EDAM ontology (Black et al., 2022; Ison et al., 2013) for the description of software tools and services. EDAM is an ontology describing concepts and terms related to data analysis, modelling, and data management, in life sciences and beyond (Figure 1a). *bio.tools* holds a collection of more than 30,000 tools, curated and annotated using EDAM concepts. It also provides enhanced search capacities and navigation (Figure 1b), as well as an API for easy programmatic access to the database.

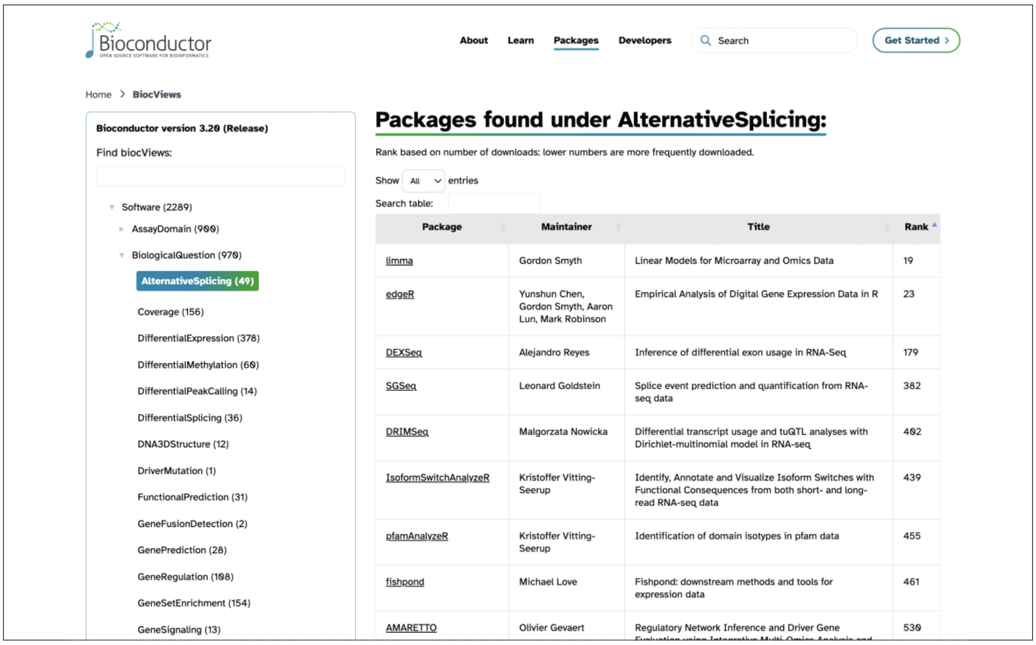


Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s or 1,000s of characters, and particularly good at aligning to relatively long (e.g. mammalian) genomes. Bowtie 2 indexes the genome with an FM Index to keep its memory footprint small: for the human genome, its memory footprint is typically around 3.2 GB. Bowtie 2 supports gapped, local, and paired-end alignment modes.

Figure 1: a. EDAM ontology structure. EDAM organises concepts into a hierarchical ontology with four main sections: Data, Format, Operation, and Topic. This formal structure facilitates interoperability by providing standardised, machine-readable annotations that enhance discoverability and integration across tools and platforms. **b.** *bio.tools* tool page. The *bio.tools* entry for Bowtie 2 exemplifies how EDAM concepts are applied to describe the scientific function of the tool, using EDAM data, operations, formats, and topics.

Objectives

Bioconductor uses an *ad hoc* vocabulary for the description of packages, called biocViews (Carey et al., 2024), which is structured as a graph with nearly 500 terms describing package attributes (Figure 2). However, it lacks the formal structure of an ontology, which can limit its utility for automated discovery and interoperability. EDAM, on the other hand, is an OWL-based ontology specifically designed for data analysis and data management concepts in biosciences, making it a better candidate for formalised, interoperable annotations. Aligning Bioconductor packages with this standard aims to enhance tool discoverability within a broader bioinformatics community.



Bioconductor version 3.20 (Release)

Find biocViews:

- Software (2289)
  - AssayDomain (900)
  - BiologicalQuestion (970)
    - AlternativeSplicing (49)
    - Coverage (156)
    - DifferentialExpression (378)
    - DifferentialMethylation (60)
    - DifferentialPeakCalling (14)
    - DifferentialSplicing (36)
    - DNA3DStructure (12)
    - DriverMutation (1)
    - FunctionalPrediction (31)
    - GeneFusionDetection (2)
    - GenePrediction (28)
    - GeneRegulation (108)
    - GeneSetEnrichment (154)
    - GeneSignaling (13)

Packages found under AlternativeSplicing:

Rank based on number of downloads: lower numbers are more frequently downloaded.

Show All entries

Search table:

Package	Maintainer	Title	Rank
limma	Gordon Smyth	Linear Models for Microarray and Omics Data	19
edgeR	Yunshun Chen, Gordon Smyth, Aaron Lun, Mark Robinson	Empirical Analysis of Digital Gene Expression Data in R	23
DEXSeq	Alejandro Reyes	Inference of differential exon usage in RNA-Seq	179
SGSeq	Leonard Goldstein	Splice event prediction and quantification from RNA-seq data	382
DRIMSeq	Malgorzata Nowicka	Differential transcript usage and tuQTL analyses with Dirichlet-multinomial model in RNA-seq	402
IsoformSwitchAnalyzeR	Kristoffer Vitting-Seerup	Identify, Annotate and Visualize Isoform Switches with Functional Consequences from both short- and long-read RNA-seq data	439
pfamAnalyzeR	Kristoffer Vitting-Seerup	Identification of domain isotypes in pfam data	455
fishpond	Michael Love	Fishpond: downstream methods and tools for expression data	461
AMARETTO	Olivier Gevaert	Regulatory Network Inference and Driver Gene	530

Figure 2: biocViews categories on the Bioconductor website. biocViews (shown on the left) is a non-ontological, hierarchical vocabulary of nearly 500 terms used to categorise Bioconductor packages based on their functionality. Packages are annotated and can be filtered using said vocabulary (shown on the right).

Over the years, several initiatives have tried to undertake the task of connecting Bioconductor and components of the ELIXIR ecosystem, however no integrated and sustainable solution has been implemented so far.

The main goals of this project are to (1) annotate more than 2,000 Bioconductor packages using EDAM, and (2) automate their integration within the ELIXIR Research Software Ecosystem and the *bio.tools* registry. More specifically, our objectives for this BioHackathon included mapping the biocViews taxonomy to EDAM concepts, assessing biocViews-EDAM mappings to identify gaps and inconsistencies, curating a reference subset of Bioconductor packages with manual annotations, and developing tools for automated EDAM concept suggestions. Additionally, the implementation of automated synchronisation mechanisms between Bioconductor and *bio.tools* was initiated.

Beyond advancing the EDAM standard, this initiative builds a collaborative bridge between Bioconductor and ELIXIR. Through structured integration, community-driven development, and quality refinement, this project contributes to ongoing efforts towards a more accessible and interoperable bioinformatics ecosystem.

Results

Overview of the BioHackathon results

One of the main goals of the work initiated during the BioHackathon was to synchronise the Bioconductor ecosystem with the ELIXIR Research Software Ecosystem, so that all relevant information regarding resources maintained in Bioconductor are automatically updated on the schedule of Bioconductor's six-month release cadence.

Bioconductor itself maintains four types of packages for different purposes (Table 1).

Package type	Description
Software	packages for data processing and analysis
AnnotationData	packages related to genome and organism structure and function
ExperimentData	packages providing curated experiment data
Workflow	packages consisting of workflow demonstrations

Table 1. Types of packages provided and curated by the Bioconductor community.

The terms “Software”, “AnnotationData”, “ExperimentData” and “Workflow” are children of the root node of the biocViews vocabulary.

Since *bio.tools* is specifically intended for software and databases, it does not support data-focused packages such as annotation and experiment packages. While workflow packages (currently only around 30) may be considered for future inclusion in the RSEc, they are not within the scope of this initial phase, and only “Software”-tagged packages are therefore currently considered for synchronisation. This decision allows us to focus on software package integration, while laying the groundwork for potentially applying EDAM annotations across all four Bioconductor package types to improve metadata consistency and interoperability in the future.

Key results from the BioHackathon include (1) mapping the biocViews vocabulary to the EDAM ontology, identifying gaps in the ontology and suggesting new terms and concepts; (2) defining a set of reference software packages from Bioconductor and manually annotating them, in order to provide a “gold-standard” to evaluate automated annotations; (3) developing large language model-based tools to automate the annotations; (4) synchronising Bioconductor software packages with the ELIXIR Research Software Ecosystem and (5) developing a BioChatter module to leverage the *bio.tools* API, enabling users to query Bioconductor package information more intuitively.

Mapping biocViews terms to EDAM

The first step in order to shift from biocViews annotations to EDAM annotations consists in mapping the existing vocabulary with the ontology, and identifying potential gaps to be filled in the future.

Exploration of software package annotations. Bioconductor uses the biocViews vocabulary for package annotations. Leaving aside annotation, experiment, and workflow packages, there is a collection of 2,289 software packages to synchronise with the RSEc. Overall, those packages are annotated using 235 different terms, with high disparities in their respective usage (Figure 3a,c). Besides, the number of annotations per package also varies widely (Figure 3b).

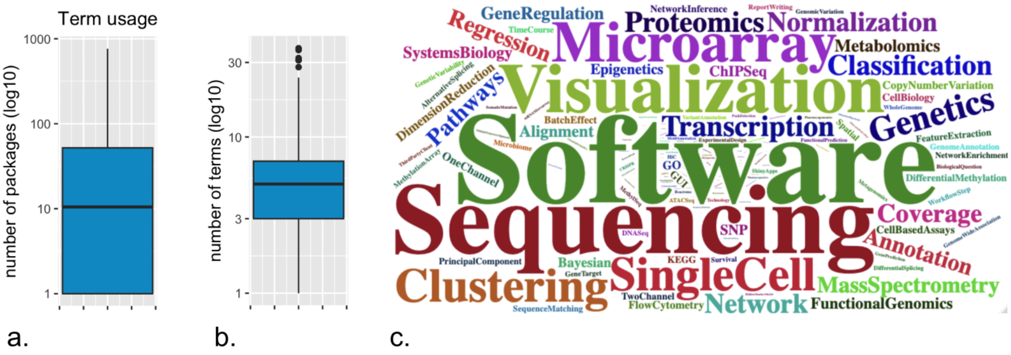


Figure 3: a. On average, the terms used for software package annotations are used 10 times, ranging from 0 to nearly 800 for the term "Software" itself. **b.** Packages are annotated with about 8 different terms on average, while overall these values range from 0 to 45. **c.** Wordcloud of terms usage.

```r
# [R] get software annotations
annotated_terms <- unique(
  (BiocPkgTools::biocPkgList(version = "3.20",
    addBiocViewParents = FALSE, repo = c("BioCsoft"))
  %>% unnest(biocViews))$biocViews
)

# [R] make some manual corrections after identifying a few bugs
annotated_terms <- annotated_terms[!annotated_terms %in%
  c("Scale\nsimulation","Genetics\nCellBiology", "3' end sequencing",
  "Differential Polyadenylation\nSite Usage", "", NA)]

annotated_terms <-
  c(annotated_terms, "Scale", "simulation", "3p end sequencing",
  "Differential Polyadenylation", "Site Usage")
```

Exploration of the biocViews vocabulary. Currently, Bioconductor's biocViews vocabulary includes a total of 497 terms, of which 175 are meant for software annotation. In order to ensure the consistency of the annotations, an automated validation is performed by BiocCheck upon submission of a new package. This ensures that packages include valid biocViews terms and meet the minimum requirement of at least two non-top-level terms. Invalid terms trigger an error during package submission, and recommendations for valid terms are provided using the recommendBiocViews function from the biocViews package. However, the systematic comparison of this controlled vocabulary against the existing annotations shows a few issues to be considered. While 160 of the dedicated 175 terms are used for software package annotations (Figure 4, green bar), some packages are annotated with non-valid biocViews terms, likely submitted before the implementation of automated checks, amounting to a total of 51 terms (Figure 4, yellow bar). Besides, 24 biocViews terms that are not meant for software annotation are used as such nonetheless (Figure 4, blue bar); and 15 valid terms are not used at all (Figure 4, orange bar). Finally 298 biocViews terms that are not meant for software annotation are consistently not used as such (Figure 4, red bar). The latter are thus of minor importance in the current scope of our project.

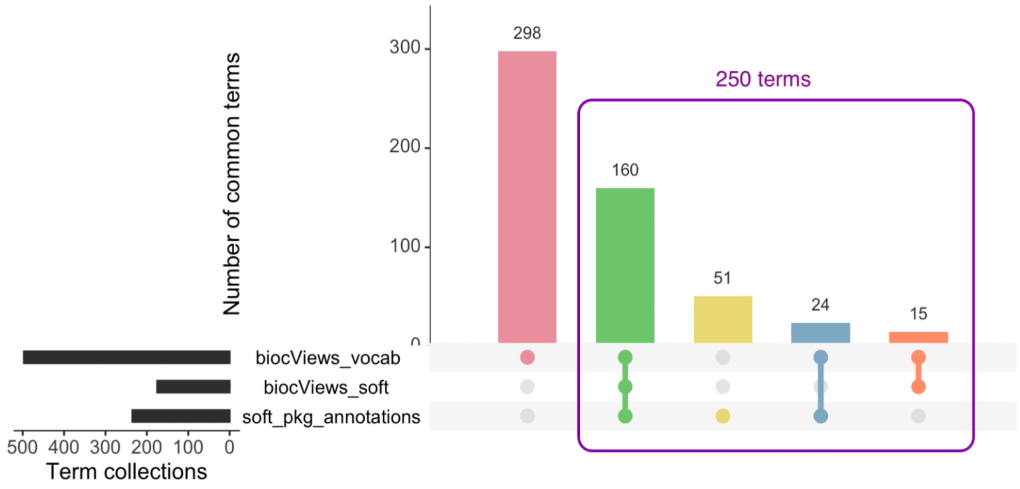


Figure 4: Upset plot showing the overlaps between 3 lists of terms: the biocViews vocabulary, the biocViews vocabulary for software, and the annotated terms from the current collection of 2,289 software packages. In total there are 548 terms used for annotation and/or proposed as part of the biocViews controlled vocabulary, including 250 terms either used or proposed for software package annotations.

```r
# [R] get biocViews vocabulary
data(biocViewsVocab)
biocviews_df <- biocViewsVocab %>% graph_from_graphnel() %>%
  as_data_frame(what = "edges")
biocViews_vocab <- unique(sort(c(biocviews_df$from, biocviews_df$to)))

# [R] get biocViews software vocabulary
reposPath <- system.file("doc", package="biocViews")
reposUrl <- paste("file://", reposPath, sep="")
biocViews_soft <- names(getBiocSubViews
  (reposUrl, biocViewsVocab, topTerm="Software"))
```

Mapping results. We mapped all of the vocabulary considered above against the EDAM ontology using the `text2term` Python library. This library proposes a variety of scoring methods based on string similarity, however it may underestimate the actual relevance of a mapped term, or not output a satisfying match when an actually relevant but distinct concept or term (synonym) is available in EDAM. Hence, though it provides a good basis for the "translation" of Bioconductor package annotations to EDAM concepts, it requires significant manual curation (see supplementary_tables, "Mapping_curated" tab - background color code follows the categories shown in Figure 4).

After curation, the mapped vocabulary was divided into five categories (Table 2).

Category	Description
Good	A perfect match or very close match with an EDAM concept's preferred label
Partial	A good enough match with an EDAM concept's preferred label
Term suggestion	There is no good match, but curation suggests another existing EDAM concept
Missing - to add	There is no good match, and there is no adequate term or concept currently available in EDAM
Out of scope	There is no good match, and the term is not in the scope of EDAM

Table 2. Types of matches distinguished after the mapping and manual curation of the mapping results.

While a total of 548 terms were mapped (497 biocViews terms + 51 non-valid terms) (Figure 5a), for the sake of the present work we will focus on the vocabulary that is either valid or actually used in current software package annotations (250 terms) (Figure 4, Figure 5b-c).

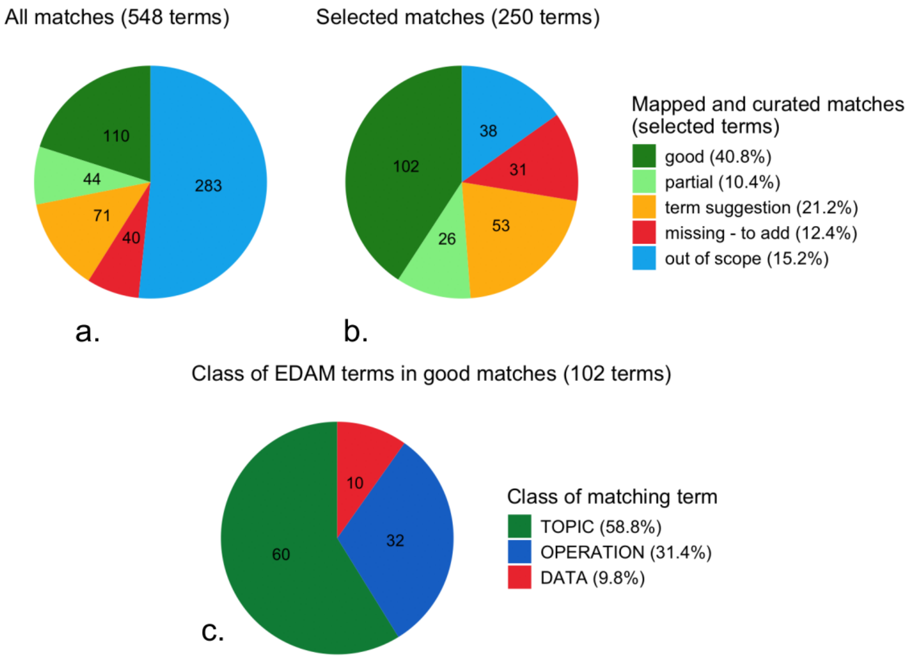


Figure 5: a. A total of 548 terms were mapped to EDAM using the text2term library, of which about half are out of the scope of EDAM. **b.** Setting aside the less relevant terms (Figure 4), 250 curated terms were further considered. 128 terms (51.2%) are mapped correctly; 53 terms (21.2%) do not have good matches but other concepts were suggested through manual curation; 31 terms (12.4%) are missing from the ontology, and 38 terms (15.2%) were considered as out of the scope of EDAM. **c.** Among the 102 "good" matches, terms were mapped to 60 EDAM topics (58.8%), 32 EDAM operations (31.4%), and 10 EDAM data types (9.8%).

A list of 29 concepts deemed as missing from EDAM was proposed from the 31 curated terms missing a match (see supplementary_tables, "Missing_terms" tab - color code follows Figure 4). Among those, some terms are related to high-throughput technologies and should be considered for addition; a few terms are related to microarray technologies, which may rise questions about their relevance nowadays; a few terms are currently part of a separate extension of the EDAM ontology and thus not available for synchronisation with *bio.tools*; and 3 terms were once part of EDAM before being deprecated, and should be reinstated.

```
# [R] save list of all above terms to file
write.table(unique(c(annotated_terms, biocViews_vocab, biocViews_soft)),
  file = "bioc_all_terms_used.tsv", col.names = F, row.names = F,
  quote = F, sep = "\t")
```

```
# [python] map all terms to EDAM using text2term
edam_dev_owl =
  "https://raw.githubusercontent.com/edamontology/edamontology/\
  refs/heads/main/EDAM_dev.owl"
```

```
text2term.map_terms(source_terms="bioc_all_terms_used.tsv",
  target_ontology=edam_dev_owl, min_score=0, save_mappings=True,
  output_file="mapping_tests_claire/bioc_all_terms_used_mapped.csv",
  term_type="class", incl_unmapped = True)
```

Defining a reference set of packages

While the translation of the biocViews vocabulary to EDAM concepts is a first step towards the standardisation of Bioconductor software package metadata, we could go further and take full advantage of the terminology available in EDAM, including topics, operations, formats and types of data. This is particularly relevant for their synchronisation with the *bio.tools* registry, and their potential future integration in platforms such as WorkflowHub (Gustafsson et al., 2024) or Galaxy (The Galaxy Community, 2024).

Since doing so manually would require a significant amount of time and expertise, we started to explore semi-automated AI-based methods. For this purpose, we decided to create a small list of packages to be curated manually, with two main purposes: serve as a basis for future annotation guidelines, and provide a gold standard to use as a reference in order to evaluate automated annotation strategies.

We created a reference set of 45 Bioconductor packages (see supplementary_tables, "Reference_packages" tab), featuring well-known, heavily downloaded packages, as well as those suggested by project contributors and developers, covering a large variety of topics. We initiated their curation (see supplementary_tables, "Package_curation" tab) which includes the extraction of existing annotations in *bio.tools* using the API, revising said annotations, and suggesting new annotations where relevant. Some package developers reviewed and updated EDAM annotations in *bio.tools* for their packages, providing detailed curated examples for future reference (`xcms`, `BridgeDbR`, `rWikiPathways`).

Automating EDAM annotations for Bioconductor packages

The `biocEDAM` package (in development in github) includes functions that operate on content provided in Bioconductor packages to infer EDAM concepts appropriate for indexing the package. Briefly, the URL of a PDF or HTML vignette is supplied to the function `vig2data`. Facilities in the `pdftools` or `rvest` packages are used to extract text for analysis by GPT-4o. The analysis employs prompts defined in the `ellmer` package to produce data on vignette authorship, "topics" identified *ad libitum* by GPT-4o, and a textual summary, called "focus", of no more than 450 words. The "focus" result of `vig2data` is then passed to the `edamize` function, which employs further prompting in connection with data in the EDAM ontology provided in the form of JSON documents. The specific prompt is "Given content about a bioinformatics tool, represent it as a JSON object compliant with the provided schema". Results of this process applied to vignettes from 7 Bioconductor packages are available in the supplementary_tables, in the "Automated_annotation" tab.

The term-to-package assignments achieved through this process seem reasonable. Vignette summaries from two packages, `ChemmineOB` and `phyloseq`, could not be processed by `edamize`. Investigation of these failures is underway.

Synchronising Bioconductor packages with the ELIXIR RSEc and *bio.tools*

The synchronisation between Bioconductor metadata with both *bio.tools* and the ELIXIR Research Software Ecosystem has been successfully established, with automated imports from Bioconductor to the RSEc now occurring on a weekly basis (see https://github.com/research-software-ecosystem/content/tree/master/imports/bioconductor for imported contents from Bioconductor, and https://github.com/research-software-ecosystem/utils/tree/main/bioconductor-import for the scripts that perform it).

These metadata files consist of:

- JSON metadata retrieved from the Bioconductor package release API, available *e.g.* at https://bioconductor.org/packages/json/3.20/bioc/packages.json.

- Citation information published in an HTML format on the Bioconductor website, *e.g.* at https://www.bioconductor.org/packages/release/bioc/citations/DESeq2/citation.html.

Work is currently underway to automate the update of *bio.tools* metadata from Bioconductor metadata files (scripts that will perform this task are under development at https://github.com/research-software-ecosystem/utils/tree/main/bioconductor-to-biotools).

The main challenges for this update are:

- to avoid duplication of *bio.tools* entries for a given Bioconductor package. To reduce this risk, newly created entries in *bio.tools* will follow a naming convention based on bioconductor package names (*i.e.* *bio.tools* IDs will be named `bioconductor-{bioconductor package name}`. Existing Bioconductor entries, created before the setup of this mechanism, will be automatically detected (based on tool identifier/name or citation information, see Figure 6), and retain their original *bio.tools* identifiers.

- to guarantee that for all packages, metadata available from Bioconductor (*e.g.*, current version, reference citation, *etc.*) are updated from this source, while information which is only available from *bio.tools* (*e.g.* EDAM annotations) is not overwritten. The workflow currently developed will therefore eventually, once the Bioconductor raw files have been imported from Bioconductor, either create new *bio.tools* entries, or for Bioconductor packages already existing in *bio.tools*, merge the metadata from Bioconductor and *bio.tools*. The full workflow as currently envisioned is illustrated in Figure 7.

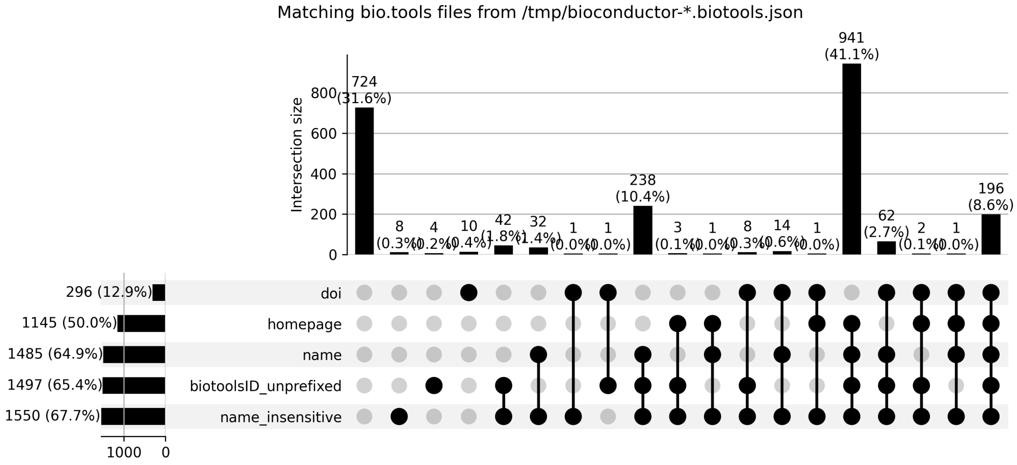


Figure 6: Upset plot summarising matching properties between *bio.tools* and Bioconductor files the in RSEc. Entry matches have been tested for equality in name, biotools ID, and homepage, as well as reference publication (through their DOI).

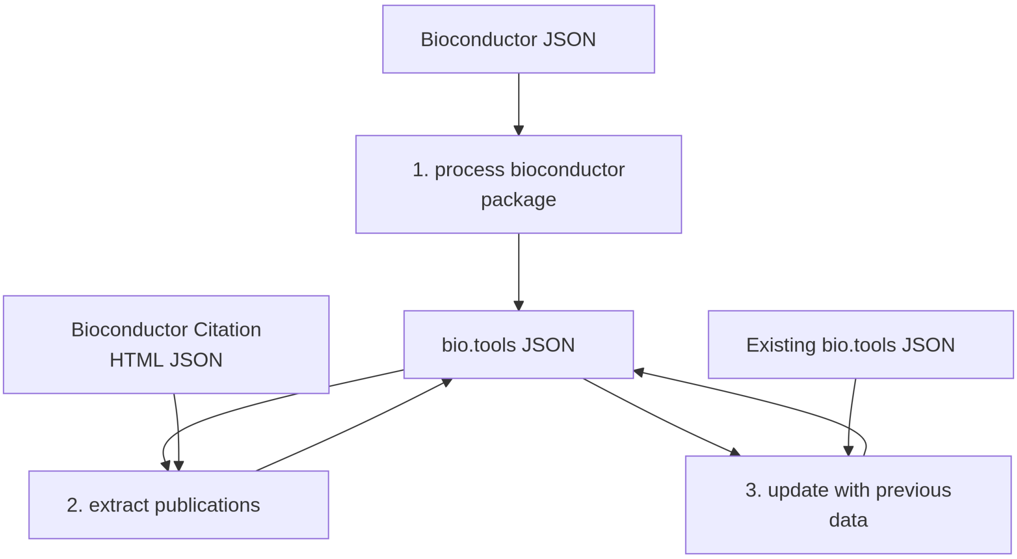

Bioconductor JSON
1. process bioconductor package
Bioconductor Citation HTML JSON
bio.tools JSON
Existing bio.tools JSON
2. extract publications
3. update with previous data

Figure 7: Workflow developed for the automated synchronisation of Bioconductor packages and their metadata in *bio.tools*. Step 1 converts the Bioconductor JSON file to the *bio.tools* format; Step 2 enriches it with the recommended citation data as defined by Bioconductor; Step 3 merges the previous description of the package, coming from *bio.tools*, so that information which is not available in Bioconductor metadata isn't removed through the update process.

Upon finalisation, package information from the 2,289 Bioconductor software packages will be available and automatically updated not only in the RSEc but also in *bio.tools*, with 1,565 updated entries and 724 new entries, representing 7.4% of the resulting *bio.tools* entries (numbers upon publication of this report).

Enhancing user querying of tools with an AI-based conversational agent

BioChatter is an open-source framework for the customisation of LLM-driven systems for applications in biomedical research (Lobentanzer et al., 2025). In addition to introducing transparency, flexibility, and open-source principles into the interaction with LLMs at a scientific level, one focus is on allowing tool use by LLMs, by implementing dedicated modules that characterise the tool. For instance, by describing a web API, the programmatic use of this API can be facilitated via the LLM.

A prototype of a `BioChatter` module was initiated to leverage the *bio.tools* API, enabling users to query Bioconductor package information more intuitively. The module interprets natural language questions, translates them into *bio.tools* API calls, and retrieves relevant package details based on EDAM concepts and other metadata. This approach is intended to support complex, context-specific queries, enhancing users' ability to identify suitable Bioconductor tools for particular bioinformatics applications.

To advance BioChatter, we are seeking specific user questions and expected outcomes to develop well-defined use cases. These examples will guide the API's natural language processing capabilities and refine responses, ensuring alignment with user needs.

Perspectives

Further collaboration with the Bioconductor community will be necessary to enable the direct maintenance of *bio.tools* metadata from Bioconductor packages. This will require integrating EDAM-based annotations to describe the scientific functions of the packages, necessitating extensions to the build infrastructure. Several technical approaches are being evaluated. One option is to add custom fields to the DESCRIPTION file, to accommodate one or more EDAM topics. Alternatively, a new annotation file, similar to the CITATION file for bibliographic information, could be used to avoid overloading the DESCRIPTION file or to include information that does not fit the key schema. Formats such as JSON-LD or RDF may be more suitable for this purpose due to their ability to handle ontology concepts and relations. This file could include information about operations and their input and output data and formats, similar to the data available from *bio.tools*. The format could be modeled after Bioschemas ComputationalTool (Beard et al., 2020) metadata profile. Additionally, to streamline maintenance, the Roxygen2 infrastructure and custom roclets could be employed to extract annotations directly from the R source code, similar to the current process for generating DESCRIPTION, NAMESPACE, and manpages.

The automated mapping of biocViews to EDAM offers a promising approach to facilitate this annotation and further metadata standardisation in the scientific description of research software. However, manual review of this mapping will be necessary to address structural differences and gaps in EDAM, thereby ensuring optimal semantic coverage of the scientific capabilities.

The entire corpus of *bio.tools* and the RSEc will also benefit from the natural language querying capabilities provided by the BioChatter module, initiated during the Biohackathon, and pending further development. To enhance BioChatter, we will collect specific user questions and expected outcomes to develop well-defined use cases. These examples will guide the API's natural language processing capabilities and refine responses, ensuring alignment with user needs.

Conclusion

Our integration of Bioconductor software packages within the ELIXIR Research Software Ecosystem through this work contributes to ongoing efforts toward enhanced discoverability and interoperability of bioinformatics tools. Mapping biocViews to EDAM, developing annotation prototypes, and exploring sustainable metadata practices have laid the groundwork for a cohesive bioinformatics ecosystem. Continued refinement and automation efforts will ensure Bioconductor resources are accessible and interoperable within ELIXIR's infrastructure.

Community and contributions

A roadmap through 2026 will guide future developments of this project, ensuring ongoing collaboration with Bioconductor and ELIXIR communities.

We welcome anyone interested to join our Bioconductor Slack #edam-collaboration channel or visit our working group page. Participation is flexible—members are encouraged to follow updates, drop into discussions, or join our meetings as often or as little as they'd like.

Data availability

During this project, several resources were developed to support the integration of Bioconductor packages within the ELIXIR Research Software Ecosystem.

- The biocEDAM GitHub repository served as the primary workspace for EDAM concept mapping and developing the `edamize()` function for automated annotations.
- Bioconductor metadata imports were added to the ELIXIR RSEc GitHub data repository. Code for the automation of these imports and the analysis/transformation of the data is available on the ELIXIR RSEc GitHub utils repository
- A BioChatter prototype module has been developed to support natural language querying via the *bio.tools* API, to be refined with additional use cases.

Acknowledgements

This work was performed during the ELIXIR BioHackathon Europe 2024 organised by ELIXIR in November 2024. This work was supported by ELIXIR, the research infrastructure for life science data. CR is part of the Institut Français de Bioinformatique (IFB, UAR 3601), funded by the Programme d'Investissements d'Avenir subsidised by the Agence Nationale de la Recherche, number ANR-11-INBS-0013. This project has been made possible in part by grants "Software for Science (Cycle 6): Ontological resource tagging and discovery for Bioconductor" ID: EOSS6-0000000067), 2024-342819 and 2024-342820 from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation. SN acknowledges funding from the German Federal Ministry of Education and Research in the frame of de.NBI/ELIXIR-DE (W-de.NBI-11).

References

Beard, N., Bacall, F., Nenadic, A., Thurston, M., Goble, C. A., Sansone, S.-A., & Attwood, T. K. (2020). TeSS: A platform for discovering life-science training opportunities. *Bioinformatics*, *36*(10), 3290–3291. https://doi.org/10.1093/bioinformatics/btaa047 **[cito:citesAsPotentialSolution]**

Black, M., Lamothe, L., Eldakroury, H., Kierkegaard, M., Priya, A., Machinda, A., Singh Khanduja, U., Patoliya, D., Rathi, R., Che Nico, T. P.others. (2022). EDAM: The bioscientific data analysis ontology (update 2021)[version 1; not peer reviewed]. *F1000Research*, *11*(ISCB Comm J), 1. https://doi.org/10.7490/f1000research.1118900.1 **[cito:citesForInformation]**

Carey, V. J., Harshfield, B., Falcon, S., Arora, S., & Shepherd, L. (2024). *biocViews: Categorized views of R package repositories*. https://doi.org/doi:10.18129/B9.bioc.biocViews. **[cito:citesAsDataSource]**

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., . . . Zhang, J. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, *5*(10), R80. https://doi.org/10.1186/gb-2004-5-10-r80 **[cito:usesDataFrom]**

Gustafsson, O. J. R., Wilkinson, S. R., Bacall, F., Pireddu, L., Soiland-Reyes, S., Leo, S., Owen, S., Juty, N., Fernández, J. M., Grüning, B., Brown, T., Ménager, H., Capella-Gutierrez, S., Coppens, F., & Goble, C. (2024). *WorkflowHub: A registry for computational workflows*. https://doi.org/10.48550/arXiv.2410.06941 **[cito:citesAsAuthority]**

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K. D., Irizarry, R. A., Lawrence, M., Love, M. I., MacDonald, J., Obenchain, V., Oleś, A. K., . . . Morgan, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, *12*(2), 115–121. https://doi.org/10.1038/nmeth.3252 **[cito:usesDataFrom]**

Ienasescu, H., Capella-Gutiérrez, S., Coppens, F., Fernández, J. M., Gaignard, A., Goble, C., Gruening, B., Gustafsson, J., Gelpi, J. L., Harrow, J., Manos, S., Miura, K., Möller, S., Owen, S., Paul-Gilloteaux, P., Peterson, H., Pitoulias, M., Repchevski, D., Tedds, J., . . . Ménager, H. (2023). The ELIXIR research software ecosystem. *F1000Research*, *12*. https://doi.org/10.7490/f1000research.1119585.1 **[cito:extends]**

Ison, J., Ienasescu, H., Chmura, P., Rydza, E., Ménager, H., Kalaš, M., Schwämmle, V., Grüning, B., Beard, N., Lopez, R., Duvaud, S., Stockinger, H., Persson, B., Vařeková, R. S., Raček, T., Vondrášek, J., Peterson, H., Salumets, A., Jonassen, I., . . . Brunak, S. (2019). The bio.tools registry of software tools and data resources for the life sciences. *Genome Biology*, *20*(1), 164. https://doi.org/10.1186/s13059-019-1772-6 **[cito:extends]**

Ison, J., Kalaš, M., Jonassen, I., Bolser, D., Uludag, M., McWilliam, H., Malone, J., Lopez, R., Pettifer, S., & Rice, P. (2013). EDAM: An ontology of bioinformatics operations, types of data and identifiers, topics and formats. *Bioinformatics*, *29*(10), 1325–1332. https://doi.org/10.1093/bioinformatics/btt113 **[cito:citesAsSourceDocument]**

Lobentanzer, S., Feng, S., Bruderer, N., Maier, A., Wang, C., Baumbach, J., Abreu-Vicente, J., Krehl, N., Ma, Q., Lemberger, T., & Saez-Rodriguez, J. (2025). A platform for the biomedical application of large language models. *Nature Biotechnology*, 1–4. https://doi.org/10.1038/s41587-024-02534-3 **[cito:usesMethodIn]**

The Galaxy Community. (2024). The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update. *Nucleic Acids Research*, *52*(W1), W83–W94. https://doi.org/10.1093/nar/gkae410 **[cito:citesAsAuthority]**