Project Acronym:	**IDPfun**
Project Full Title:	**Driving functional characterization of intrinsically disordered proteins**
Grant Agreement:	**778247**
Project Duration:	**48 months (01/03/2018 - 28/02/2022)**

Deliverable D1.3

A software tool for the automatic extraction of IDR relevant information from literature

Work Package:	**WP1**
Lead Beneficiary:	**EMBL**
Due Date:	**28 Feb 2019 (M12)**
Submission Date:	**28 Feb 2019 (M12)**
Deliverable Type:	**D**
Dissemination Level:	**P**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 778247

Table of Content

Executive summary

This document describes the *"D1.3 A software tool for the automatic extraction of IDR relevant information from literatures"* for the IDPfun project. The deliverable was a software tool for the automatic extraction of IDR relevant information from literature. The pipeline was completed on time and as described.

List of Acronyms

Acronym	Definition
IDP	Intrinsically Disordered Protein
PPI	Protein-Protein Interaction
SLiM	Short Linear Motif
PMID	Pubmed Identifier (Pubmed ID)
TF-IDF	Term-Frequency times Inverse-Document-Frequency

Project overview

Introduction on the aims of the project

The goal of this project is to employ text-mining to retrieve relevant information from scientific publications. It provides a mechanism to recover valuable biological insights sequestered within scientific articles and offer them to the community in a standardized manner that facilitates its use. This can be of particular interest for annotators and curators of biological databases, who may struggle to find publications of interest within PubMed. The project was developed by the Intrinsically Disordered Proteins research community and is focused on the curation of protein-protein interactions (PPI) mediated by Short Linear Motifs (SLiMs). However, the developed tool, *Curation Helper - Classifier tool,* for text-mining based classification of scientific articles has been created to allow general usage by any protein curation project.

Extracting relevant terms from literature

Text-mining is a widely used approach for identifying and retrieving data of interest from a noisy corpus. In this project we have developed the *Curation Helper - Classifier tool*, a computational tool that provides a system for annotators and curators of biological data to prioritize the available literature. Publications are retrieved from PubMed and relevant features are extracted from the article title and abstract. These fields are split into smaller parts (such as individual words), cleaned from punctuation and non-descriptive characters, and labelled to recognize nouns and proper nouns. The list of extracted terms serves as a summary of the publication's content. Using machine learning models trained on a previously curated dataset, an article can be classified based on relevant terms to (i) make suggestions that will simplify the curation process and (ii) flag potential inaccuracies in the curation.

Curation Helper - Classifier tool is built entirely on Python and is currently provided as a standalone command-line tool. The tool uses:

- Natural Language Toolkit (nltk) to tokenize the article text and apply Part-Of-Speech tagging.
- Scikit-Learn (sklearn) is used for supervised classification, with several methods available (see Table 1).

Description of results

Text-mining and classification - *Curation Helper - Classifier tool*

Methodology

The *Curation Helper - Classifier tool* allows a user to take a set of labeled (classified) publications and use them to predict the class of unlabeled (unclassified) publications. *Curation Helper - Classifier tool* has 5 major parts:

Data preparation: The user provides a set to publications to create article classifiers. The training set is given as a standalone tab-separated file of publications (defined by PubMed Identifiers) - classification pairs. The tool gathers the data required to train the machine learning classifier from PubMed and UniProt REST services and stores the data locally in XML format. The default information used as training data for the article classifiers is the article title and abstract. This data can be augmented using MESH terms, SciLite data or UniProt-derived gene and protein names.

TF-IDF document scoring: The training data from each publication is converted to a matrix of TF-IDF (term-frequency times inverse-document-frequency) word occurrences. The TF-IDF matrix encodes the enrichment of a given term in a given document relative to the whole set of documents. For example, TF-IDF scores are high when a term occurs frequently in a document but not in the extended collection of documents. After TF-IDF transformation, common words are filtered and unigrams (terms made up of only one word) extracted for further consideration. At this point, non-nouns can also be removed from consideration.

Classifier model construction: The TF-IDF vector of each article are used to create a classifier based on the publication - classification pairs in the input training set. The classifier defines hyperplane and allows the class of a novel document to be distinguished by calculating a metric analogous to a similarity. The article similarity is quantified as a distance from the hyperplane for the class. The *Curation Helper - Classifier tool* includes 16 different methods to create a multi class classifier. Furthermore, the tool has an inbuilt option to benchmark the available methods.

Classifier significance distribution construction: A random article decision function distance distribution is calculated for each class in the classifier by calculating the distance of each article in the training set, excluding the articles that are members of the tested class, against a given class classifier. The assumption that these articles are motif articles which are not describing the given class and therefore will provide a conservative representation of the likelihood of seeing a given distance by chance. The article decision function distance distribution is converted to a cumulative probability and applied to each article during

classification to provide an intuitive probabilistic classification metric to complement the more abstract decision function distances.

Classification using the model: The goal of the classifier is to correctly identify the correct class of an unseen articles. *Curation Helper - Classifier tool* accepts articles for classification as either one or a list of PubMed IDs. The article information is retrieved and processed as described for the construction of the classifier. The distance from the vectorised input article from the hyperplane for each class in the model is then calculated. This distance is related to the similarity of the input article to the set of articles of a given class. All class distances are returned and the closest class is returned as the most likely classification for the article. A probability of the distance is also returned by the tool as defined by the random article decision function distance distribution calculated during *Classifier model construction.* The classification tool provides the data as either a JSON- or TDT-formatted output.

Example

As the current project is focused on the *automatic extraction of IDR relevant information from literatures*, we use a custom compendium of literature about functional modules in *IDP*. Each article covers one or more of the SLiM classes in the Eukaryotic Linear Motif database (ELM). Supervised learning algorithms can learn from these pairs of papers and manually assigned classes, and in return, provide a way to map an input (i.e., an unknown publication) to an output (a certain label, such as an ELM class here). The output of this procedure is its classification as referring (or not) to a certain ELM class and list of high-scoring terms in the article that are strong discriminators for that class. As an example, we trained a linear model classifier with stochastic gradient descent (SGDClassifier), regularized by the L2 penalty based a ELM training dataset of 2115 publications and 250 assigned ELM classes of publication-ELM motif class pairs. *Table 2* shows the classification of the publication "*A Conserved Motif Provides Binding Specificity to the PP2A-B56 Phosphatase*" (*Pubmed Identifier:* 27453045) and "*The Mitotic Checkpoint Complex Requires an Evolutionary Conserved Cassette to Bind and Inhibit Active APC/C*" (*Pubmed Identifier:* 27939943). The results show that "*A Conserved Motif Provides Binding Specificity to the PP2A-B56 Phosphatase*" is correctly recognized as referring to the PP2A holoenzyme B56-docking site (ELM class DOC_PP2A_B56_1), with relevant keywords such as "b56" and "lxxixe" (the regular expression of the consensus motif) automatically recognized. Likewise, several keywords in "*The Mitotic Checkpoint Complex Requires an Evolutionary Conserved Cassette to Bind and Inhibit Active APC/C*" (which is not part of the training dataset) receive high-scores, pointing out the relevance of this publication as bibliographic source for two different, although highly related, motifs.

Table 1 - Results from classification of "A Conserved Motif Provides Binding Specificity to the PP2A-B56 Phosphatase" (PMID: 27453045) and "The Mitotic Checkpoint Complex Requires an Evolutionary Conserved Cassette to Bind and Inhibit Active APC/C" (PMID: 27939943) using a custom classifier model.

PMID	Bonferroni -corrected probability	Decision function (distance)	Top keywords : score	ELM class

27453045	0	0.396	**b56**:0.86 **pp2a**:0.85 **lxxixe**:0.52 **phosphatase**:0.22 **provides**:0.19	DOC_PP2A_B56_1
27939943	0.0025	-0.398	**bubr1**:0.54 **abba**:0.48 **cdc20**:0.44 **mcc**:0.27 **sac**:0.24	LIG_APCC_ABBA_1
27939943	0	-0.077	**ken**:2.08 **box**:0.80 **chromatid**:0.37 **chromosomes**:0.32 **spindle**:0.30	DEG_APCC_KENBOX_2

Benchmarking

The ability of the *Curation Helper - Classifier tool* to correctly classify SLiM articles was tested using a 5-fold cross validation benchmark protocol based on the ELM training dataset. The assessment of 16 different methods available in Annotator Helper was performed by comparing values of precision, recall, F1 score and accuracy (only accuracy is shown in *Table 2*). In the following example for simplicity all models were trained on title and abstract, but classification can also be performed by augmenting the test article with SciLite annotations only (+*SciLite*), MeSH terms only (+*MeSH)* or the combination of both (*all*).

Our results indicate that injecting extra information including gene, protein names or MeSH terms into the classification provides a light but consistent jump in performance for all models. Linear Support Vector Classification with the standard l2 penalty and all available terms provides the best accuracy overall, although several classifiers perform similarly.

Table 2 - Benchmark of included classification methods. See main text for details.

Method	**Accuracy**			
	title + abstract	**t + a + SciLite**	**t + a + MeSH**	**all**
Perceptron	0.509	0.513	0.518	**0.524**
Ridge Classifier	0.643	0.651	0.652	**0.657**
K-Nearest Neighbors	0.572	0.552	**0.579**	0.560
Random Forest	0.559	0.579	0.585	**0.598**
Nearest Centroid (Rocchio	0.587	0.603	0.611	**0.618**

Classifier)				
Passive- Aggressive	0.649	0.655	**0.662**	0.660
RBF SVC	0.272	0.332	0.316	**0.349**
LinearSVC (L1 penalty)	0.590	0.604	0.597	**0.605**
LinearSVC (L2 penalty)	0.651	0.654	0.661	**0.667**
LinearSVC with L1-based feature selection	0.613	0.629	**0.633**	0.632
LinearSVC with L2-based feature selection	0.638	0.656	0.645	**0.665**
SGDClassifier (L1 penalty)	0.621	0.627	0.626	**0.631**
SGDClassifier (L2 penalty)	0.651	0.652	**0.664**	0.656
Elastic-Net penalty	0.649	0.657	0.663	**0.664**
BernoulliNB Naive Bayes	0.454	0.513	0.479	**0.530**
MultinomialNB Naive Bayes	0.517	**0.548**	0.511	0.542

Availability

The project source code is available at the IDPfun GitLab repository: *https://gitlab.com/idpfun/annotator-helper*