Extracting kinetic information from literature with KineticRE

Ana Alão Freitas[1], Hugo Costa[2], Miguel Rocha[1] and Isabel Rocha[1,*]

[1]Centre of Biological Engineering, University of Minho, Braga, Portugal

[2]SilicoLife Lda, Rua do Canastreiro, 15, 4715-387 Braga, Portugal

Summary

To better understand the dynamic behavior of metabolic networks in a wide variety of conditions, the field of Systems Biology has increased its interest in the use of kinetic models. The different databases, available these days, do not contain enough data regarding this topic. Given that a significant part of the relevant information for the development of such models is still wide spread in the literature, it becomes essential to develop specific and powerful text mining tools to collect these data. In this context, this work has as main objective the development of a text mining tool to extract, from scientific literature, kinetic parameters, their respective values and their relations with enzymes and metabolites. The approach proposed integrates the development of a novel plug-in over the text mining framework *@Note2*. In the end, the pipeline developed was validated with a case study on *Kluyveromyces lactis*, spanning the analysis and results of 20 full text documents.

1 Introduction

The use of metabolic models in several areas of science and industry has taken an increasing interest, since they allow a variety of *in silico* simulations under different experimental conditions [1]. These simulations provide information to increase the production of compounds of interest, for strain optimization and other applications. In most cases, the simulations are performed under steady state conditions, using genome-scale stoichiometric metabolic models [2, 3, 4]. However, to better understand the dynamic behavior of metabolic systems in a wide variety of conditions, it is imperative to develop dynamic kinetic models of cellular metabolism [5, 6, 7].

Despite the existing large number of databases on metabolism (e.g. BRENDA [8], SABIO-RK [9], ExPASy [10], MetaCyc [11], etc), available data on the structure and parameters of kinetic equations are not sufficient for the development of such models and a large amount of relevant information still resides in the biomedical literature [3, 5].

In recent years, interest in biomedical text mining (BioTM) has increased, as a way to automatically extract meaningful knowledge from unstructured texts [12]. Typical BioTM tasks are divided in two main areas: Information Retrieval (IR) and Information Extraction (IE), which is further sub-divided in two major sub-areas: Named Entity Recognition (NER) and Relation/Events Extraction (RE) [13, 14] (Figure 1). To summarize, a set of relevant publications

*To whom correspondence should be addressed. Email: irocha@deb.uminho.pt

can be retrieved, searching in bibliographic repositories and form a document set (corpus). This can be submitted to an NER process, where biological entities are tagged with classes (e.g. genes, proteins, compounds). This annotated corpus can be further processed through RE processes to identify relations between entities [13, 15].

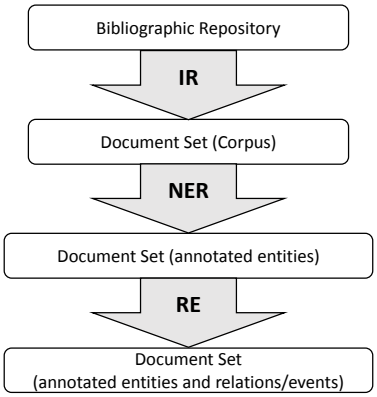


Figure 1: Generic pipeline for BioTM.

There are currently very few BioTM tools specifically designed to extract kinetic data from the scientific literature. The BRENDA database developed a new resource, KENDA (Kinetic ENzyme DAta), which uses rule and dictionary-based approaches to extract kinetic parameters and expressions from literature. However, it only analyzes abstracts and titles [16]. Using a similar approach, the KID algorithm extracts kinetic information for enzymes from abstracts and automatically generates a database [17]. Other approaches based on machine learning [18] have been focused on abstracts, therefore excluding the information contained in full text documents.

The main goal of this work is the development of a BioTM pipeline for the automatic identification and collection of kinetic data from full text documents in the literature, implemented as the KineticRE plug-in for the *@Note2* framework [19]. It works over annotated documents that result from an NER process. Based on those annotations, it looks for information on kinetic parameters, materialized by specific relations between the different annotated entities (kinetic parameters, values, enzymes and metabolites). The KineticRE algorithm is a rule-based RE process that returns a relation set ranked by score. The final objective is to obtain an ordered list of relations per document, which can be analyzed and may allow the user to select the parts of interest of the various documents, thereby decreasing the time necessary to collect relevant information.

2 @Note2

@Note2 (http://www.anote-project.org/) is a multi-platform BioTM workbench, fully written in Java and that uses a MySQL database, which copes with the most important IR, NER and RE tasks. The framework is implemented over the AI-Bench framework, thus following the Model-View-Controller (MVC) paradigm [19]. *@Note2* is organized into four main functional modules ((Figure 2): Publication Manager Module (PMM), Corpora Module (CM), Resources

Module (RM) and the Corpus processes module (includes NER and RE processes). PMM, CM and RM are examples of datatypes that can be generated, updated or removed by operations; while NER and RE are operations/prrocesses that can be applied to the datatypes, and their results are new datatypes that can suffer more operations [19].

The PMM is responsible for the IR step and has the functionality of searching and retrieving publications from repositories like PubMed, based on queries. The CM can receive those publications or import PDF files to form a Corpus. The RM is responsible for the management of lexical resources used in some IE processes, which include dictionaries, rules, ontologies, lookup tables and word lists.

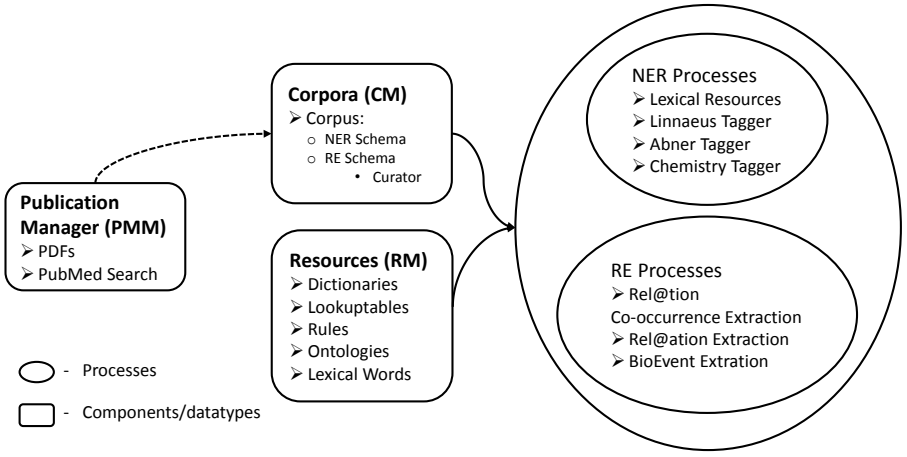


Figure 2: @*Note2* structure.

Different IE processes can be applied to a corpus, using some of the resources, to identify and extract bio-entities and their relationships: the NER process pool contains the Lexical resources and Linnaeus tagger [20], a hybrid dictionary and rule-based algorithm, the rule-based Chemistry tagger and the Machine Learning based ABNER tagger; the RE process pool contains a co-occurrence extraction process, a linguistic-based algorithm (Rel@tion) and a Machine learning approach [19].

3 Kinetic RE Pipeline

Since the RE processes previously available on @*Note2* are not adequate to extract kinetic information, we developed new specific RE algorithms, taking advantage on some of the @*Note2* existing features. This new process works based on kinetic parameter values and names, enzymes and metabolites that are annotated using NER dictionaries and a rule-based process.

In the context of this work, a relation is defined as a set of entities from different classes, co-occurring in a sentence, which has a score. A basic relation occurs when an entity from the class "values" is found, followed by an entity from the class "units". This basic relation, named as a pair, is the basis to look for more complex relations. To form more complex relations, entities from the classes "kinetic parameters", "enzymes" or "metabolites" have to be found

before or after the pair. Identified relations receive a score according to the different classes of the entities they contain.

3.1 NER Process

The NER process chosen was the Linnaeus Tagger [20], a dictionary and rule-based process. The dictionary regarding enzymes was generated by uploading a BRENDA [8] file created on May 20th 2015, using BRENDA SOAP web service, containing 6676 enzymes terms and 81985 synonyms. Two other dictionaries were created: "kinetic parameters" (21 terms and 124 synonyms) and "units" (52 terms and 144 synonyms). The first was filled with terms related to enzyme kinetics and the second with units terms (some examples are presented in Table 1).

Table 1: Examples of terms listed in the dictionaries "kinetic parameters" and "units".

Kinetic Parameters Terms		**Units Terms**	
Term	*Synonyms*	*Term*	*Synonyms*
Km	KM km kM Michaelis constant michaelis constant Michaelis constants michaelis constants	**mol/L**	mol / L mol L-1 mol L -1 mol L- 1 mol L - 1
Vmax	vmax V max v max Maximal velocity maximal velocity Maximum velocity maximum velocity	**mol g-1 h-1**	mol/g.h mol / g.h mol g- 1 h- 1 mol g - 1 h - 1 mol g -1 h -1
Inhibitor	inhibitor Inhibitors inhibitors	**g mL-1**	g mL- 1 g mL -1 g mL - 1

Rules make use of Java regular expressions to identify specific patterns. From Table 2, it is possible to analyze some of the rules from a total of 19 that were created.

An ontology was created, where the process is identical to the development of a dictionary. In this case, a list of metabolites was uploaded originating from the ChEBI database. An OBO file, with release 114 was downloaded (April 3rd 2014) containing 45436 terms and 266644 synonyms. To each term, an id, a name and synonyms are associated.

Lastly, a "stop words" list was generated, with 459 common English words (e.g.: all, and, any, as, her, off, the, ...) to be used in the NER pre-processing step, for the purpose of removing

Table 2: Some regular expressions of the rules created and examples of instances.

Nr	Regular Expressions (Java)	Examples
1	?=\s+(V)\s+	V
2	?=\s+(-{0,1}\d+\.{0,1}\d*\s{0,1}±\s{0,1}\d+\.{0,1}\d*)\s+	7.8 ± 23 ; -45±1.23
3	\s(-{0,1}\d+\.{0,1}\d*\s{0,1}x\s{0,1}10exp-{0,1}\d+)	9.99 x 10exp9 ; 7x10exp-6
4	?=(\s+(-{0,1}\d+\.\d+)\s+)	]-99.99999; 99.99999[
5	?=(\s+(-{0,1}\d+)\s+)	]-99999; 9999[

from the annotation process ambiguous terms that can be listed in the metabolites resource. Even though, 19 terms were found (listed in Table 3) that were contained in both BRENDA dictionary and ChEBI ontology. From the analysis of the terms we realized that most of them had been incorrectly associated to enzymes.

Table 3: Common terms between the BRENDA dictionary and ChEBI ontology.

6-deoxyerythronolide B	sulfite oxidase	GDP
angiogenesis inhibitor	Luciferin	IDP
Ebselen	ADA	IMP
Erlotinib	CDP	PAL
formate dehydrogenase	DDT	PAP
Glutaredoxin	G12	PP2
Hemoglobin		

3.2 Identification of Relations

The lexical resources used in the NER process need to be mapped by the user to the corresponding class (values, units, kinetic parameters, enzymes and metabolites), being the score values defined for each class.

The RE algorithm is depicted in Figure 3. The search for a relation starts with the identification of values and units annotated together, where each occurrence found is considered a pair with a minimum score assigned. For example, as shown in Figure 4, two occurrences (green) were annotated as belonging to the "value" class, but only the first is followed by an entity of "Unit" class (purple), so only one pair will be identified.

If only one pair is identified in the sentence, the whole sentence is considered the scope for possible relations. If the algorithm does not find entities from the other classes in the sentence, the pair is considered as the simplest relation and given the minimum score. Otherwise, when entities from the classes "kinetic parameters", "enzymes" or "metabolites" are found either on the left (positions are lower than the pair/value start position Sp_v in Figure 4), or on the right (positions are higher than pair/unit end position Ep_u) the score is increased. For each entity, the score to add is the one assigned to its class in the configuration of the algorithm.

Input: Annotated corpus (NER result), entities classes and corresponding scores.

Algorithm:

FOR each doc from the corpus:

Split in sentences;

FOR each sentence:

Search for values annotated close to annotated units (distance between 1 and 3 spaces);

FOR each pair (value/unit):

Consider a basic relation and assign the minimum score;

Search for other entities annotated (metabolites, enzymes and kinetic parameters classes) on the left or right of the pair;

Create a new relation (pair + new entities found);

Calculate the relation score.

Output: List of complex relations with the corresponding score.

Figure 3: Algorithm for relation extraction in KineticRE.

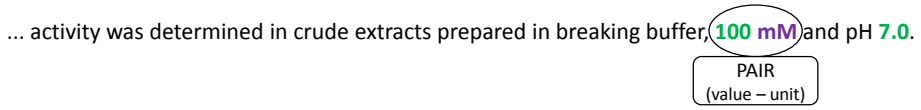


Figure 4: Example of a sentence with only one value-unit pair.

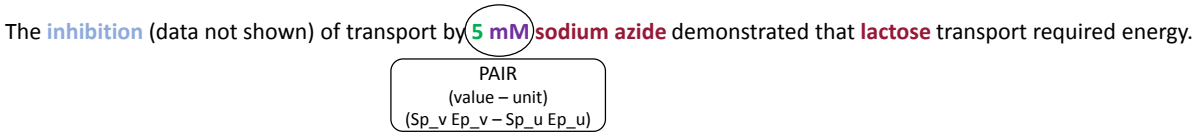


Figure 5: Example of a sentence with one value-unit pair. Annotated entities from "metabolites" (dark red) and "kinetic parameters" (light blue) classes. SP_v and Ep_v are respectively start and end position of the value in the sentence; Sp_u and Ep_u represents the same for the unit parameter.

Considering the following score values: pair→10; metabolites→100; enzymes→1000 and kinetic parameters→10000, the example in Figure 5 will receive a score of 10210 (10 + 2*100 + 10000). Clearly, for the computation of the scores, the greater the number of annotated entities of different classes, the higher will be the score assigned to the relation.

If the sentence has two pairs (Figure 6), for the first pair the scope to find a relation will be between the beginning of the sentence and the start position of the second pair (Sp_v2), while for the second it will be between the end position of the first pair (Ep_u1) and the end of the sentence. Using the scores defined above, the first relation gets a score of 20110 and the second a score of 10110. It is possible to infer that the information annotated between pairs will be added to both relations, creating redundancy. In this case, it is obvious that the second kinetic parameter (in light blue) belongs to pair 2, but since there is no straight way to perform this kind of separation, it was decided to allocate the information in the middle to both pairs, thus minimizing the probability of losing important information.

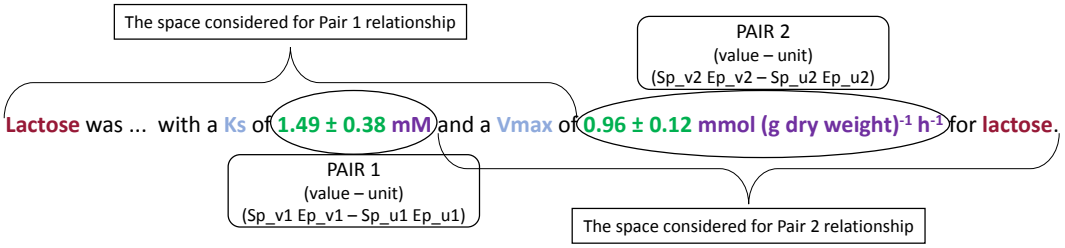


Figure 6: Example of a sentence where two value-unit pairs were identified. Annotated entities from "metabolites" (dark red) and "kinetic parameters" (light blue) classes are shown. SP_v and Ep_v are, respectively, start and end positions of the value in the sentence; Sp_u and Ep_u represent the same for the unit parameter.

4 Implementation

The algorithm developed was integrated in *@Note2*, using functionalities already implemented, like the conversion from PDF to text and the overall NER process. As shown in Figure 7, the algorithm receives an annotated corpus generated by an NER process, maps the lexical resources to the different classes and the result is a list of relations with scores, as well as a corpus with annotated entities and relations.

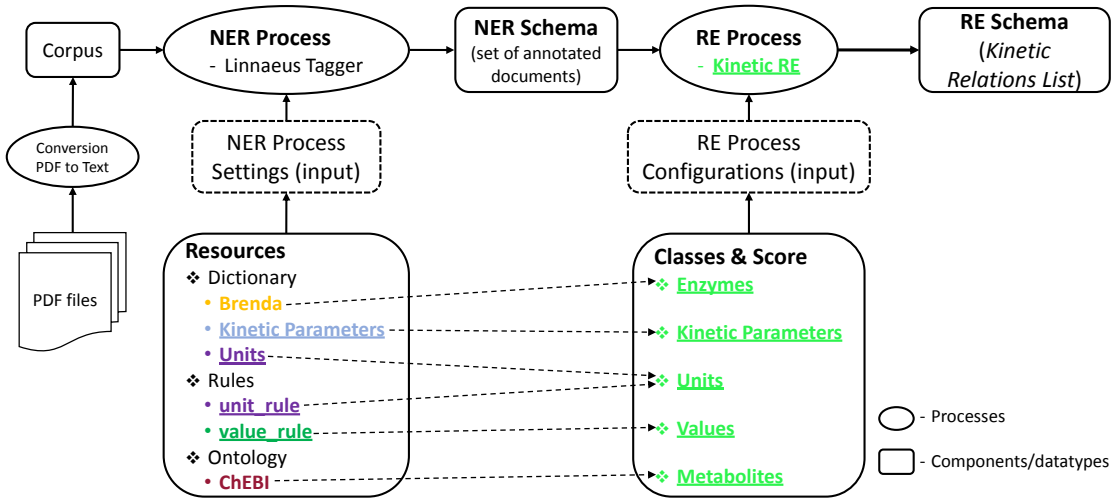


Figure 7: Main steps of the pipeline. Shown in light green and underlined are the new resources that were created in *@Note2* in the scope of this work.

This plug-in can be selected within the RE options of *@Note2* on the clipboard. Two input GUIs are then launched, in which the user has to choose the NER process and do the mapping between the resources used in NER and the classes available for the RE. For example, the resource "Units" (dictionary) and the resource "unit_rule" (set of rules) should be mapped to the RE class "Units", as shown in Figure 8.a.

After running the process, an output GUI with five separators/views will appear. In the first, it is possible to check some RE statistics, like the number of relations extracted from the documents. In the second separator, all the relations extracted are listed. On the third view, some statistics regarding the NER process are displayed, like the number of entities annotated to each class.

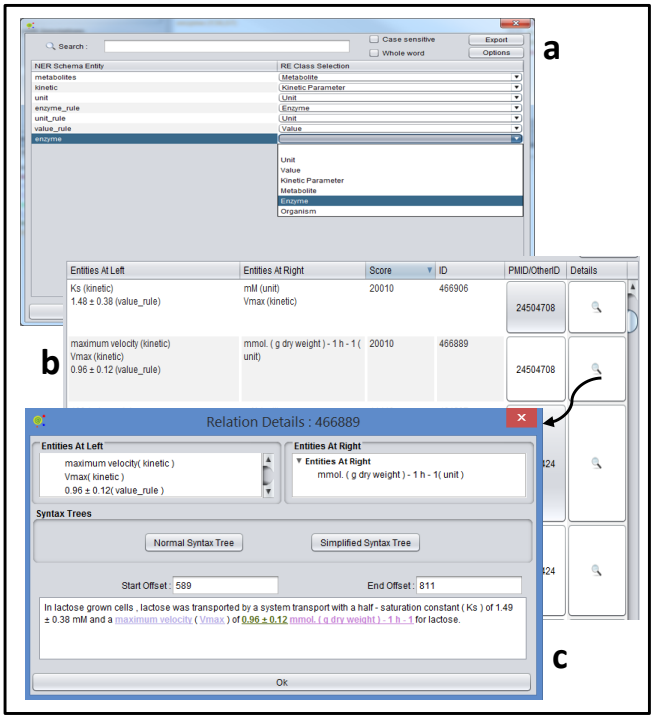


Figure 8: Kinetic RE GUIs: second Input (a) and two Output Views (b and c).

In the fourth, the user can choose a document to see in detail, which will appear in a new GUI where it is possible to see the entire text with all the annotated entities and relations identified. It also allows the user to create and edit relations. Also in this view, if the user clicks on an entity, another GUI will show all the relations where the selected entity was included. In the last view, all the relations are listed with the corresponding score (Figure 8.b) and a specific relation can be chosen (details column/option) to check in detail (Figure 8.c).

5 Results and Discussion

Initially, two different sets of 10 articles on *Kluyveromyces lactis* were selected to guide the creation of the different lexical resources, to validate the NER process and to test and validate the algorithm developed. During the validation of the NER process, it was detected that some terms of interest, that have been identified manually, were not annotated. Based on these issues, we proceeded with the improvement of the lexical resources, adding missing terms and some progress was also done in the conversion from PDF to text.

In a second stage, a new set of 20 documents was collected and submitted to the pipeline. Previously, the manual curation of the expected relations in each paper was done directly in the framework *@Note2*. This allows to compare automatically the results obtained from the RE with the expected ones, using the "Evaluate method", already implemented in the tool. The results obtained, as well as the results expected from the manual curation, are shown in Table 4. Analyzing this table, it is possible to verify that the number of expected relations is higher than the number of relations agreed, extracted with the algorithm.

Table 4: Results for the 20 papers submitted to the pipeline (manual curation vs KineticRE).

Nr	PubMed Id	Nr Relations Expected	Nr Relations Extracted	Nr Relations Agreed	Nr Sentences Curated	Precision %	Recall %
1	6211189	**11**	**17**	**6**	543	**35,29**	**54,55**
2	6853445	**12**	**13**	**9**	250	**69,23**	**75,00**
3	8065434	**4**	**6**	**4**	309	**66,67**	**100**
4	9165108	**0**	**8**	**0**	392	**0**	**0**
5	9392075	**1**	**6**	**0**	293	**0**	**0**
6	12084066	**4**	**12**	**4**	317	**33,33**	**100**
7	12142408	**9**	**14**	**9**	509	**64,29**	**100**
8	12353647	**0**	**4**	**0**	127	**0**	**0**
9	12451549	**1**	**7**	**1**	196	**14,29**	**100**
10	12677461	**6**	**8**	**6**	238	**75,00**	**100**
11	12882981	**2**	**5**	**0**	322	**0**	**0**
12	14626424	**10**	**12**	**10**	155	**83,33**	**100**
13	15033461	**6**	**0**	**0**	151	**0**	**0**
14	15556281	**0**	**7**	**0**	238	**0**	**0**
15	17307293	**0**	**5**	**0**	398	**0**	**0**
16	17976174	**2**	**11**	**2**	152	**18,18**	**100**
17	19943044	**3**	**6**	**1**	157	**16,67**	**33,33**
18	20358191	**4**	**7**	**4**	284	**57,14**	**100**
19	24504708	**15**	**17**	**15**	186	**88,24**	**100**
20	25204685	**7**	**10**	**6**	518	**60,00**	**85,71**
Total		**97**	**175**	**77**	**5735**	**44,00**	**79,38**

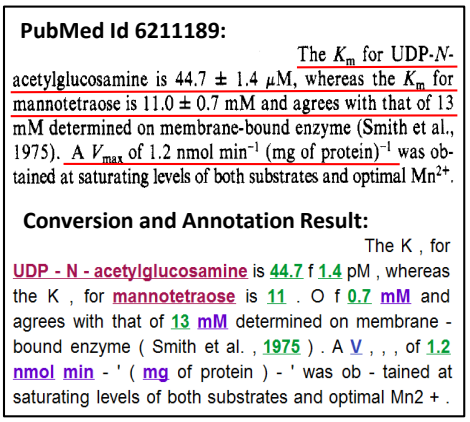
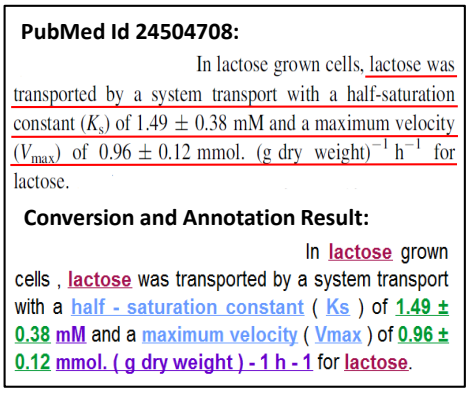


Figure 9: Comparison between original PDF and converted/annotated result showing some disagreements.

In cases like the first, fourth, thirteenth and seventeenth papers, not all the relations expected were identified, but comparing the original publication of the first paper, with the annotation

results (shown in Figure 9) allowed us to conclude that the pair value-unit was not recognized due to a bad conversion from the PDF to text, which led to one or more entities not being properly annotated. Common problems in that step include non-recognition of certain symbols in the case of units or bad formatting conversion in the case of digits.

In the case of the third, twelfth, eighteenth and nineteenth papers, all the relations expected were correctly identified, illustrated by the examples presented in Figure 10.



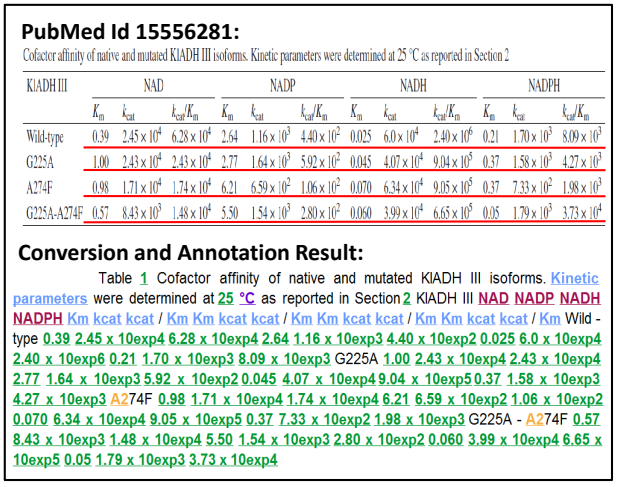
PubMed Id 24504708:

In lactose grown cells, lactose was transported by a system transport with a half-saturation constant (K_s) of 1.49 ± 0.38 mM and a maximum velocity (V_{max}) of 0.96 ± 0.12 mmol. (g dry weight)$^{-1}$ h^{-1} for lactose.

Conversion and Annotation Result:

In lactose grown cells , lactose was transported by a system transport with a half - saturation constant (Ks) of 1.49 ± 0.38 mM and a maximum velocity (Vmax) of 0.96 ± 0.12 mmol. (g dry weight) - 1 h - 1 for lactose.

Figure 10: Comparison between original PDF and converted/annotated result showing a successful conversion and annotation.



PubMed Id 15556281:

Cofactor affinity of native and mutated KlADH III isoforms. Kinetic parameters were determined at 25 °C as reported in Section 2

KlADH III	NAD			NADP			NADH			NADPH		
	K_m	k_{cat}	k_{cat}/K_m	K_m	k_{cat}	k_{cat}/K_m	K_m	k_{cat}	k_{cat}/K_m	K_m	k_{cat}	k_{cat}/K_m
Wild-type	0.39	2.45×10^4	6.28×10^4	2.64	1.16×10^3	4.40×10^2	0.025	6.0×10^4	2.40×10^6	0.21	1.70×10^3	8.09×10^3
G225A	1.00	2.43×10^4	2.43×10^4	2.77	1.64×10^3	5.92×10^2	0.045	4.07×10^4	9.04×10^5	0.37	1.58×10^3	4.27×10^3
A274F	0.98	1.71×10^4	1.74×10^4	6.21	6.59×10^2	1.06×10^2	0.070	6.34×10^4	9.05×10^5	0.37	7.33×10^2	1.98×10^3
G225A-A274F	0.57	8.43×10^3	1.48×10^4	5.50	1.54×10^3	2.80×10^2	0.060	3.99×10^4	6.65×10^5	0.05	1.79×10^3	3.73×10^4

Conversion and Annotation Result:

Table 1 Cofactor affinity of native and mutated KlADH III isoforms. Kinetic parameters were determined at 25 °C as reported in Section 2 KlADH III NAD NADP NADH NADPH Km kcat kcat / Km Km kcat kcat / Km Km kcat kcat / Km Km kcat kcat / Km Wild - type 0.39 2.45 x 10exp4 6.28 x 10exp4 2.64 1.16 x 10exp3 4.40 x 10exp2 0.025 6.0 x 10exp4 2.40 x 10exp6 0.21 1.70 x 10exp3 8.09 x 10exp3 G225A 1.00 2.43 x 10exp4 2.43 x 10exp4 2.77 1.64 x 10exp3 5.92 x 10exp2 0.045 4.07 x 10exp4 9.04 x 10exp5 0.37 1.58 x 10exp3 4.27 x 10exp3 A274F 0.98 1.71 x 10exp4 1.74 x 10exp4 6.21 6.59 x 10exp2 1.06 x 10exp2 0.070 6.34 x 10exp4 9.05 x 10exp5 0.37 7.33 x 10exp2 1.98 x 10exp3 G225A - A274F 0.57 8.43 x 10exp3 1.48 x 10exp4 5.50 1.54 x 10exp3 2.80 x 10exp2 0.060 3.99 x 10exp4 6.65 x 10exp5 0.05 1.79 x 10exp3 3.73 x 10exp4

Figure 11: Comparison between original PDF and converted/annotated result showing a successful conversion.

Other cases, like the fifth, fourteenth and fifteenth papers only one or zero relations were expected to be extracted, since most relevant data are presented in tables in the original papers. As it is possible to analyze in Figure 11 no problems occurred during the conversion and annotation process but the RE still does not deal with all kind of table possibilities to extract a series of values that appear together. Also, during the conversion process some tables are recognized as images, text or even graphics.

Globally, the system has shown acceptable performance. Indeed, it was possible to obtain a recall of 79,38% and a precision of 44,00%.

6 Conclusions and Future Work

In this work, we have proposed and validated a pipeline to collect kinetic data from literature, working over the text mining tool *@Note2*, for which we developed a plug-in that is made available for the community. Overall, the preliminary results are quite promising. Although, there is some work in related topics [17, 18], it was not possible to directly compare the performance of our algorithm with other tools. Indeed, in both mentioned studies, the analysis is performed using only abstracts. Furthermore, one of the studies applies Machine learning techniques to classify the documents and, while the other uses rule and dictionary based approach, the last update to the online database was in 2011.

In our tool, the PDF conversion to text is a core task, but also one of the most problematic. Despite the improvements done during this work, the results are not perfect and leave room for further improvements. The same can be said for the resources used, which can be altered to improve the identification of entities. For example, this can be achieved adding terms to the units and kinetic parameters dictionaries, fitting the rules for a better detection of values or making regular updates of the metabolites and enzymes lists.

We also intend to present a workflow where both processes, NER and Kinetic RE, will run sequentially without user interference. This will allow the user to set the classes score values and choose which resources to map to the RE classes. In the end, we aim to apply the approach developed to other organisms and corpora. The development and validation of this tool will continue along these lines, considering all other features that can improve the results.

Acknowledgements

The work was funded by National Funds through the FCT (Portuguese Foundation for Science and Technology) within project ref. PTDC/QUI-BIQ/119657/2010 “Finding the naturally evolved design principles of prevalent metabolic circuits”. The authors would like to thank the FCT Strategic Project PEst-OE/EQB/ LA0023/2013 and the Projects “BioInd - Biotechnology and Bioengineering for improved Industrial and Agro-Food processes”, REF. NORTE-07-0124-FEDER-000028 and “PEM Metabolic Engineering Platform”, project number 23060, both co-funded by the Programa Operacional Regional do Norte (ON.2 O Novo Norte), QREN, FEDER.

References

[1] M. Durot, P.-Y. Bourguignon and V. Schachter. Genome-scale models of bacterial metabolism: reconstruction and applications. *FEMS microbiology reviews*, 33(1):164–90, 2009.

[2] J. S. Edwards and B. O. Palsson. Robustness analysis of the *Escherichia coli* metabolic network. *Biotechnology Progress*, 16(6):927–39, 2000.

[3] K. R. Patil, M. Å kesson and J. Nielsen. Use of genome-scale microbial models for metabolic engineering. *Current Opinion in Biotechnology*, 15(1):64–69, 2004.

[4] K. Raman and N. Chandra. Flux balance analysis of biological systems: applications and challenges. *Briefings in bioinformatics*, 10(4):435–449, 2009.

[5] C. Chassagnole, N. Noisommit-Rizzi, J. W. Schmid, K. Mauch and M. Reuss. Dynamic modeling of the central carbon metabolism of *Escherichia coli*. *Biotechnology and Bioengineering*, 79(1):53–73, 2002.

[6] A. Buchholz, J. Hurlebaus, C. Wandrey and R. Takors. Metabolomics: quantification of intracellular metabolite dynamics. *Biomolecular engineering*, 19(1):5–15, 2002.

[7] T. Kadir, A. A. Mannan, A. M. Kierzek, J. McFadden and K. Shimizu. Modeling and simulation of the main metabolism in *Escherichia coli* and its several single-gene knockout mutants with experimental verification. *Microbial Cell Factories*, 9:88, 2010.

[8] I. Schomburg, O. Hofmann, C. Baensch, A. Chang and D. Schomburg. Enzyme data and metabolic information: BRENDA, a resource for research in biology, biochemistry, and medicine. *Gene Function & Disease*, 1(3-4):109–118, 2000.

[9] U. Wittig, R. Kania, M. Golebiewski et al. SABIO-RK-database for biochemical reaction kinetics. *Nucleic Acids Research*, 40(D1):D790–D796, 2012.

[10] E. Gasteiger. ExPASy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Research*, 31(13):3784–3788, 2003.

[11] R. Caspi, T. Altman, K. Dreher et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research*, 40(D1):D742–D753, 2012.

[12] S. Ananiadou, D. B. Kell and J.-i. Tsujii. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12):571–579, 2006.

[13] H. Shatkay and M. Craven. *Mining the Biomedical Literature*. MIT Press, 2012.

[14] K. B. Cohen and L. Hunter. Getting started in text mining. *PLoS Computational Biology*, 4(1):e20, 2008.

[15] R. Rodriguez-Esteban. Biomedical text mining and its applications. *PLoS Computational Biology*, 5(12):e1000597, 2009.

[16] I. Schomburg, A. Chang, S. Placzek et al. BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA. *Nucleic Acids Research*, 41(Database issue):D764–72, 2013.

[17] S. Heinen, B. Thielen and D. Schomburg. KID - an algorithm for fast and efficient text mining used to automatically generate a database containing kinetic information of enzymes. *BMC Bioinformatics*, 11(1):375, 2010.

[18] J. Hakenberg, S. Schmeier, A. Kowald, E. Klipp and U. Leser. Finding kinetic parameters using text mining. *Omics: a Journal of Integrative Biology*, 8(2):131–152, 2004.

[19] A. Lourenço, R. Carreira, S. Carneiro, P. Maia, D. Glez-Peña, F. Fdez-Riverola, E. C. Ferreira, I. Rocha and M. Rocha. @Note: a workbench for biomedical text mining. *Journal of Biomedical Informatics*, 42(4):710–20, 2009.

[20] M. Gerner, G. Nenadic and C. M. Bergman. Linnaeus: a species name identification system for biomedical literature. *BMC Bioinformatics*, 11(1):85, 2010.