Universidade do Minho

Escola de Engenharia

Ana Margarida Antunes Coelho Alão de Freitas

EXTRACTION OF KINETIC INFORMATION FROM LITERATURE

Braga, October 2014

Universidade do Minho

Escola de Engenharia

Ana Margarida Antunes Coelho Alão de Freitas

EXTRACTION OF KINETIC INFORMATION FROM LITERATURE

Master Thesis Report

Supervisors:

Prof. Isabel Rocha

Prof. Miguel Rocha

Braga, October 2014

DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA TESE

Universidade do Minho, __/__/____

Assinatura: ____

Acknowledgements

Não poderia iniciar este trabalho, sem prestar o meu agradecimento a um conjunto de pessoas, que de diversas formas tornaram possível a sua concretização.

À Doutora Isabel Rocha e ao Doutor Miguel Rocha, meus orientadores, pela oportunidade que me deram para trabalhar neste projecto. Pelos conhecimentos transmitidos, pela constante disponibilidade e apoio, pela motivação, incentivo e confiança depositada.

Ao Hugo Costa, por todo o tempo dispendido, pelo apoio, ajuda e sugestões.

À Brígida, "É para amanha!", à Sara e ao Zenha, pelas longas horas de trabalho e não só, por todo o apoio dado nos momentos bons e nos menos bons, pelos risos e lágrimas, e acima de tudo pela amizade.

À Sophia, à Daniela e à Sara, nada melhor que: "*Together in electric dreams, because the friendship that you gave, has taught me to be brave*".

Ao Liu, o "meu amor profissional", que sempre se mostrou disponível para ajudar quando precisei.

A todos os elementos do grupo BisBII, que tornaram esta etapa inesquecível.

Aos meus amigos, que mesmo longe nunca deixaram de me apoiar.

À madrinha adoptiva, por estares sempre lá quando preciso e até pelos "puxões de orelhas".

E por fim, porque os últimos são sempre os primeiros, obrigada pais e mano. A pessoa que sou devo-o a vocês, obrigada por nunca terem deixado de acreditar em mim.

Gostaria também de deixar um agradecimento, à Fundação Portuguesa para a Ciência e Tecnologia pela bolsa de iniciação à investigação que me foi atribuída, no âmbito do projecto financiado PTDC/QUI-BIQ/119657/2010 – "Finding the naturally evolved design principles of prevalent metabolic circuits".

Abstract

Several areas of science and industry are increasingly interested in the use of metabolic models, because they allow *in silico* simulation of the behavior of organisms under different experimental conditions, optimization and maximization of production of products of interest, the testing of new drugs and the effect of mutations or gene deletions, among others.

In most cases, stoichiometric models are used, allowing to perform accurate simulations. However, they are based on steady state assumptions.

In order to better understand the dynamic behavior of metabolic networks in a wider variety of conditions, it is imperative to develop kinetic models of cellular metabolism. These models describe, in a dynamic way, the mass balances for each metabolite in the network. However, despite the large number of databases, available data are not sufficient for the development of such models and a large amount of relevant information still resides in the literature.

Due to the significant increase in publications in digital format coupled with the growing need to glean information of interest and the difficulty of that task, it becomes essential to develop more specific, assertive and powerful text mining tools, able to handle with a large number of publications at the same time and extract from them the most important information.

Towards this purpose, this work has as main objective the development of a specific text mining tool to identify relationships between kinetic parameters, their respective values and between their enzymes and metabolites.

The pipeline proposed within that objective integrates the plug-in into an existing text mining tool, @Note2. This tool already allows the annotation of documents based on specific terms lists.

The results validate the whole process presented, as well as corroborate the initial idea that this method may be used to gather information of interest.

Keywords: Enzyme kinetics, metabolic models, text mining, name entity recognition, relation extraction, databases.

EXTRAÇÃO DE INFORMAÇÃO CINÉTICA DA LITERATURA

Resumo

Diversas áreas da ciência e indústria estão cada vez mais interessadas no uso de modelos metabólicos, pois estes permitem a simulação *in silico* do comportamento de organismos em diferentes condições experimentais, a otimização e maximização da produção de produtos de interesse, o teste de novas drogas e a análise do efeito de mutações ou deleção de genes, entre outros.

Na maioria dos casos são usados modelos estequiométricos que permitem realizar simulações bastante precisas, mas que se baseiam em hipóteses no estado estacionário.

Com o intuito de melhor compreender o comportamento dinâmico das redes metabólicas sob uma variadade mais ampla de condições, é imperativo desenvolver modelos cinéticos do metabolismo celular. Estes modelos descrevem, de uma forma dinâmica, os balanços de massa para cada metabolito na rede. No entanto, apesar do grande número de bases de dados, os dados disponíveis não são suficientes para o desenvolvimento deste tipo de modelos, pois uma parte sinificativa da informação relevante encontra-se dispersa pela literatura.

Devido ao aumento significativo de publicações em formato digital, aliado à necessidade crescente de retirar delas informação de interesse e à dificuldade da execução dessa tarefa, torna-se indispensável desenvolver ferramentas de mineração de texto mais específicas, assertivas e poderosas, capazes de lidar com um grande número de publicações ao mesmo tempo e extrair delas a informação mais importante.

Tendo em vista esse propósito, este trabalho tem como principal objectivo o desenvolvimento de uma ferramenta de mineração de dados específica para identificar relações entre parâmetros cinéticos, os seus respectivos valores e entre as suas enzimas e metabolitos.

A *pipeline* proposta para esse efeito integra o *plug-in* numa ferramenta de mineração de texto já existente, o @Note2, ferramenta esta que já permite a anotação de documentos com base em listas de termos específicos.

Os resultados obtidos validam todo o processo apresentado, assim como corroboram a ideia inicial de que esta poderá ser uma alternativa cada vez mais usada para a recolha de informação de interesse.

Table of Contents

Nomenclature

BioTM	Biomedical Text Mining
BRENDA	Braunschweig Enzyme Database
CM	Corpora Module
E	Enzyme
Ep_u	End position of unit
Ep_v	End position of value
ES	Enzyme-substrate complex
ExPASy	Expert Protein Analysis System
GUI	Graphical User Interface
IE	Information Extraction
IR	Information Retrieval
K_m	Michaelis constant
MVC	Model-View-Controller
NER	Name Entity Recognition
NLP	Natural Language Processing
P	Product
PMM	Publication Manager Module
RE	Relation Extraction
RM	Resources Module
S	Substrate
S_1	Inhibitor
SABIO-RK	System for the Analysis of Biochemical Pathways-Reaction Kinetics
Sp_u	Start position of unit
Sp_v	Start position of value
SBML	Systems Biology Markup Language
V	Velocity
V_0	Initial velocity
V_{max}	Maximum velocity

Index of Figures

Index of Tables

1 Introduction

1.1 Scope and Motivation

This thesis work is integrated within the project "*Finding naturally evolved design principles of prevalent metabolic circuits*", whose main goal is to find and validate a catalogue of circuit designs that identify the most frequent elementary circuits in metabolic networks and the regulatory patterns that are most frequently associated to those circuits [1]. For that project it is necessary to collect data about enzyme kinetics under different experimental conditions for six organisms (*Kluyveromyces lactis*, *Helicobacter pylori*, *Streptocuccus pneumonia*, *Enterococcus faecalis, Saccharomyces cerevisiae* and *Escherichia coli*) which will be integrated in the respective metabolic model. We intend to collect those data from databases and literature.

Table 1.1: Number of articles available in Pubmed Central and MedLine since 1900. Search done in @Note2.

Organism	Organism name	Organism name + keyword: kinetics
Kluyveromyces lactis	1284	90
Helicobacter pylori	35855	737
Enterococcus faecalis	12324	583
Streptococcus pneumoniae	27601	1509
Saccharomyces cerevisiae	107199	10664
Escherichia coli	334282	36839

Regarding databases, it is known that a considerable number of them are available and include a variety of information from different areas of life sciences.

Therefore, it is relatively easy to search and select which one/ones are best suited for the needs and objectives of the research and the best way to retrieve those data. However, concerning literature, the process is not so easy and straightforward.

Due to the large amount of literature available these days (Table 1.1), allied to the difficulty in finding on that literature the information of interest and the time consumed, we propose in this thesis a text mining software tool (plug-in) specific to identify and extract this kind of data.

The main requirements for the plug-in are that it should present the most important papers in order to decrease the amount of literature for the user to survey, and at the same time to identify and compile the relevant information. This software is integrated in the @Note2 (http://www.anote-project.org/) toolbox (in collaboration with SilicoLife), which is an open source platform for Biomedical Text Mining that handles with major information retrieval and information extraction tasks and promotes multi-disciplinary research [2].

1.2 Objectives of this study

The aim of this thesis is to develop a semi-automated way to identify and collect kinetic data from literature, and for that propose we will present the development and implementation of a plug-in specific for that identification. This plug-in will be integrated in @Note2 toolbox.

It will work with a set of annotated documents that result from a NER process that uses dictionary and rules to identify and annotate the biological entities. Based on the NER annotation, it will look for possible relations between the different annotated entities. Like the NER, it will use a rule-based system and the entity classes to identify the relations, which will be based on a value-unit pair

The final objective is to obtain a list of relations per document, ordered by importance.

1.3 Thesis Structure

This document is structured in five main chapters. In this first, a brief context to the area, the work motivation and the problems to confront are described. The different objectives proposed are also presented.

The second chapter presents a review on the state-of-the-art regarding the several topics related with this work. It starts with a brief and general approach to Systems Biology, followed by a presentation on enzyme kinetics, metabolic models and some databases. To finish this chapter we present the text mining approach, the main concepts, as well as the different tools already available.

The plug-in objective, concept and the approach used are presented in chapter three, as well as all the implementation steps, user interface and the integration with @Note2. Moreover, this chapter includes the structure of @Note2 and a description of the AIBench framework, in which the software and all the plug-ins have been built on.

The fourth chapter is devoted to the case study used to prospect and achieve the best approach to develop the proposed software. Along with the analysis and discussion of the results obtained, limitations that have been found and also steps that can be improved are described. In this chapter some modifications that have been done to the previous @Note2 based on the case study analysis are also explained.

To finalize, in the fifth chapter, an overview of all the work done is presented, together with several conclusions and work that can still be done in the near future.

2 State of the Art

More than fifty years ago, Watson and Crick discovered the double-helix structure of DNA. A new era started in molecular biology and ever since our knowledge of biological processes and structures has been growing immensely. However, several of these progresses would not have been possible without using computational methods. Thus, computer science plays an important role in the emerging and interdisciplinary field of systems biology. Due to the link between biological methods and informatics concepts, many projects, such as the Human Genome Project, were held successfully. With the end of this project, new challenges have emerged for bioinformatics and nowadays the main goal is to analyze and use the data collected. However, progress in molecular biology also influences the design and development of methods and concepts of computer science [3].

In the last years, the techniques associated with systems biology and known as omics (genomics, transcriptomics, proteomics, metabolomics, interactomics, fluxomics, etc.) emerged with the huge development of molecular biology and technology. Although these techniques generate a large amount of data every day, living systems are extremely complex and therefore it is difficult to predict their behavior over time and under various different conditions [4].

2.1 Systems Biology

Systems biology is an emerging, interdisciplinary and integrated study of complex interactions on all omics levels of biological systems. The aim of systems biology is to understand biological systems, studying the structure of the system, such as gene regulatory and biochemical networks; the dynamics of the system, performing qualitative and quantitative analyses, and also building models of the systems with high predictive ability. It also aims at understanding the control and design methods of the system, integrating theoretical and computational methods with experimental efforts [4].

Systems biology involves the use of computer simulations of cellular subsystems to analyze and visualize the complex connections of these cellular processes and makes heavy use of mathematical and computational models. It is an emerging approach applied to biomedical and biological scientific research [4].

Tools such as automated genome annotation, genome-scale metabolic reconstructions and regulatory network reconstructions using microarray data are being used to bring together new knowledge in this field [5].

One of the key challenges in systems biology is to provide mechanisms to collect and integrate the necessary data in order to meet the different requirements of the analysis [6].

Currently, an indispensable tool for the study of metabolism using a systems biology approach is using a metabolic network reconstruction [7]. These network reconstructions contain all known metabolic reactions of an organism and the genes that encode each enzyme and they are used to compute a variety of phenotypic states [6]. Metabolic network reconstructions are becoming available for an increasing number of organisms each year [8].

The construction of these models will allow further *in silico* simulations of the microorganisms behavior under different conditions [9], thus aiding multiple biological fields such as metabolic engineering, prediction of outcomes of gene deletions and drug-target identification, among others [10]. The rapid evolution of computational tools and software allow the construction and analysis of more biological models, with more reliability.

2.2 Enzyme Kinetics

Enzymes are usually protein molecules that manipulate other molecules, the enzymes substrates.

Enzyme kinetics is a field of science that deals with the many factors that can affect the rate of an enzyme-catalyzed reaction. The most important factors include the concentration of enzyme, reactants, products, and the concentration of any modifiers such as specific activators, inhibitors, pH, ionic strength, and temperature. When we study the action of these, it is possible to deduce the kinetic mechanism of the reaction. That is, the order in which substrates and products bind and unbind and the mechanism by which modifiers alter the reaction rate [11].

Brown and Henri suggested the first standard model for enzyme action, but in 1913 it was meticulously established by Michaelis and Menten, describing the binding of a free enzyme to the substrate, forming an enzyme-substrate complex. This complex suffers a transformation, releasing the product and free enzyme that is then available for binding again to a new substrate molecule [12].

Traditionally, the reactant molecule that binds to the enzyme is called the substrate (S), P is the product, E is the free enzyme, ES the enzyme-substrate complex and k_1; k_{-1} and k_2 are rate constants [13].

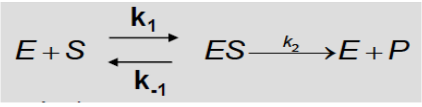


Figure 2.1: Standard model for an enzymatic reaction (Michaelis-Menten model).

The Michaelis-Menten model assumes that a fast equilibrium is established between the reactants (E + S) and the ES complex, followed by a slower conversion of the ES complex back into free enzyme and product, so the model (Figure 2.1) assumes that $k_2 << k_{-1}$ and the reaction velocity will be proportional to the concentration of the ES complex as [12]:

$$V = K_2 * [ES]$$

The Michaelis-Menten equation that describes the kinetic behavior of a large number of enzymes, is the rate equation for an irreversible one-substrate enzyme-catalyzed reaction [13]:

$$V_0 = \frac{V_{max} * [S]}{K_m + [S]}$$

It is a statement of the quantitative association between the initial velocity (V_0), the maximum velocity (V_{max}) and the initial substrate concentration ([S]), all related through the Michaelis constant (K_m). This equation can be algebraically changed into forms that are useful to practically determine K_m and V_{max} [14].

By definition K_m is the substrate concentration that results in half-maximal velocity for the enzymatic reaction. An equivalent way of affirming this is that the K_m represents the substrate concentration at which half of the enzyme active sites in the sample are occupied by substrate molecules in the steady state [15].

Some molecules are capable of slowing down or speeding up the rate of enzyme catalyzed reactions. Such molecules are respectively called enzyme inhibitors and activators. An enzyme inhibitor binds to enzymes and reduces or abolishes their activity, and since blocking enzyme activity can kill a pathogen, many drugs are enzyme inhibitors. On the other hand, activators are molecules that bind to enzymes and increase their enzymatic activity [14].

2.2.1 Inhibition

Inhibitors are classified in two large groups. The irreversible group includes the inhibitors that will covalently bind to the enzyme or modify the enzyme chemically, causing a permanent change to the enzyme. Inhibitors from the reversible group bind reversibly to the enzyme and different types of inhibition are produced. Conventionally, reversible enzyme inhibitors are classified as competitive, uncompetitive, or non-competitive, depending to their effects on V_{max} and K_m. These different effects result from the inhibitor binding to the enzyme E, to the enzyme-substrate-complex ES, or to both, respectively. Competitive and uncompetitive are the mechanisms that usually are more described in text books, but in practice, competitive inhibition is very common while the uncompetitive one is quite rare [15, 16].

Competitive inhibition refers to the case in which the inhibitor (S_1 in Figure 2.2) competes with the substrate to occupy the active site of an enzyme. And when this happens, the inhibitor is avoiding the binding of the substrate to the enzyme. The two ligands (inhibitor and substrate) compete for the same enzyme active site and generally bind in a mutually exclusive way; that is, the free enzyme binds either a molecule of inhibitor or a molecule of substrate, but not both at the same time [13, 15].

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \xrightarrow{k_2} E + P$$

And:

$$E + S_1 \underset{k_{-3}}{\overset{k_3}{\rightleftarrows}} ES_1$$

$$v = \frac{k_2 E_0 S}{S + K_{eq}\left(1 + \frac{S_1}{K_{eq1}}\right)}$$

Figure 2.2: Competitive Inhibition

Uncompetitive inhibitors only bind when the substrate is bound to the enzyme, meaning that this kind of inhibitor (S_1 in Figure 2.3) binds exclusively to the ES complex.

The substrate binding causes a conformational change on the enzyme that allows the subsequent binding of the inhibitor, which can bind to the substrate bound to the enzyme or in a completely separate site. However, this inhibitor does not directly compete with the substrate for the active site, so increasing the substrate concentration cannot beat an uncompetitive inhibitor [14, 15].

$$\begin{tikzcd}
E \arrow[r, shift left, "S"] & ES \arrow[l, shift left] \arrow[r] \arrow[d, shift right] & E + P \\
 & ESS_1 \arrow[u, shift right, "S_1"'] &
\end{tikzcd}$$

$$v = \frac{k_2 E_0 S}{S\left(1 + \frac{S_1}{K_{eq1}}\right) + K_{eq}}$$

Figure 2.3: Un-Competitive Inhibition

An uncompetitive inhibitor decreases the affinity of the enzyme for its substrate, increasing the value of K_m [15, 16]. Uncompetitive inhibition is frequently defined in terms of one substrate enzymes, but in practice is only observed with enzymes that have two or more substrates. In reality, uncompetitive inhibition is quite rare [14].

Noncompetitive inhibition refers to the case in which an inhibitor can either bind to the free enzyme or to the enzyme-substrate complex (ES in Figure 2.4).

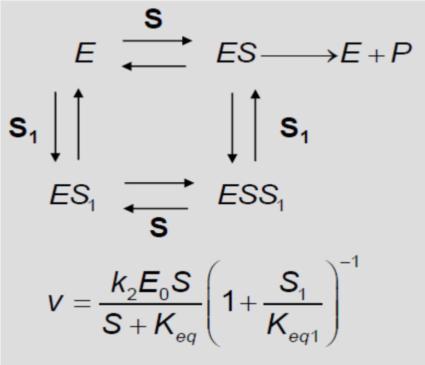

S
E
ES
E + P
S1
S1
ES1
ESS1
S
$v = \frac{k_2 E_0 S}{S + K_{eq}} \left(1 + \frac{S_1}{K_{eq1}}\right)^{-1}$

Figure 2.4: Non-Competitive Inhibition.

This kind of inhibitors do not compete with the substrate for binding to the enzyme active site, since they bind in a site distinct from the active one. Because of this, increasing substrate concentration does not overcome this type of inhibition, like it occurs in uncompetitive inhibition.

The apparent effect of a noncompetitive inhibitor is a decrease in the V_{max} value without affecting the K_m value for the substrate [14, 16].

Inhibitors are important for a number of reasons; for example, many pharmaceutical compounds act as inhibitors of enzymes or signaling proteins. It is therefore important to understand how the rate of enzymatic reactions answers to changes in the concentration of inhibitors. In basic research, inhibitors are also important because they allow understanding the active and other sites and the catalytic action. There are many naturally occurring enzyme inhibitors, the most famous being the antibiotic penicillin and vancomycin or the antibacterial sulfonamides [14, 15].

2.3 Genome-scale Metabolic Models

Usually, metabolic models used in systems biology can be grouped into two classes: stoichiometric models and kinetic models [17]. The information required to build stoichiometric models is much lower compared with kinetic models and thus these have become very popular in the last years, especially the metabolic reconstructions at a genome-scale.

In 1999, the first genome-scale metabolic network reconstruction was published [18] and every year since then the number of organisms with available genome-scale stoichiometric models has been increasing [8].

These metabolic network reconstructions represent the majority of metabolic reactions occurring in the organism and the genes encoding each of the enzymes by integration of mathematical models and biochemical data [17].

These models allows further *in silico* simulations of the microorganisms' behavior under different environmental and genetic conditions [9], thus aiding multiple biological fields such as drug design, metabolic engineering, prediction of outcomes of gene deletions, assignment of functions to unknown genes, drug-target identification, etc [10].

Despite being very important, reconstructing genome-scale metabolic models is a very complex and time consuming task, as evidenced by the protocol of 96 steps, published by Thiele and Palsson in 2010, to create these metabolic models [7].

For this reconstruction we have to consider several steps, such as the genome annotation, the determination of the stoichiometry of the reactions, the definition of compartmentation and assignment of reaction localizations, the determination of the biomass composition, the measurement of energy requirements and additional constraints [19, 20].

Nowadays, there are tools to create a genome-scale metabolic reconstruction and it is possible to make the model obtained public so that anyone can access and extract knowledge.

2.4 Kinetic models

Although stoichiometric models allow to perform accurate simulations, they are based on steady-state assumptions. Thus, in order to better understand the systemic behavior of metabolic networks under a wider variety of conditions, it is important to develop kinetic models of cellular metabolism [21].

Kinetic models aim to describe the mass balances for each metabolite in the network, in a dynamic way. To simulate that dynamic behavior of the system, they make use of systems of ordinary differential equations [22].

However, existing data are not sufficient to infer the kinetics of cellular systems at a large scale, and besides, the values of the kinetic parameters and the structure of the kinetic expressions show significant differences when comparing the results obtained *in vivo* with *in vitro* [22].

So, kinetic models are difficult to obtain due to the lack of information on the dynamics and regulation of metabolic reactions and on the understanding of kinetic processes [17].

2.4.1 SBML Format

Despite all the progress made in metabolic models construction, it remains the need for researchers to have a standard way to share that information. In order to facilitate this exchange of information, a platform has been developed for storing and exchanging model information – the Systems Biology Markup Language (SBML).

SBML is a software-independent language, based on XML, for storing biochemical reaction models and transferring them between software tools. It is a free and open standard with widespread software support and a community of users and developers (http://www.sbml.org) [23].

SBML can represent models from many different areas of computational biology, including metabolic networks, cell signaling pathways, regulatory networks, gene regulation, and many others. It is the standard for representing computational models in systems biology today. But it is not suited for representing experimental data or numerical results [23].

The use of SBML in software packages solves issues regarding compatibility, providing a standard format for publications and databases.

Nowadays more than 250 programs support SBML and around 460 curated models have been published in this format (BioModels database [24]).

2.5 Databases

The rapid sequencing of a large number of genomes and the consequent large amount of data generated has made it imperative to organize all the available data in ways that facilitate analysis, and lots of different databases were created with this purpose.

2.5.1 BRENDA

BRENDA (**BR**aunschweig **EN**zyme **DA**tabase) is the main collection of enzyme functional data available to the scientific community (http://www.brenda-enzymes.org). The development was started in 1987 at the German National Research Centre for Biotechnology in Braunschweig (GBF), continued at the University of Cologne, was made available via internet in 1998 and is continuously updated since then. Now it is curated and hosted at the Technical University of Braunschweig, Institute of Biochemistry and Bioinformatics [25].

Data on enzyme functions are extracted directly from the primary literature and critically evaluated by qualified scientists. Formal and consistency checks are done by computer programs and each data set on a classified enzyme is checked manually by at least one biologist and one chemist [26].

The database contains a large range of aspects of enzymology, such as functional data like kinetic data for catalysis and enzyme inhibition, enzyme-catalysed reactions, purification, enzyme stability, crystallization or mutations and the biggest collection of enzymes names and synonyms, fully stored in approximately 50 information fields. Each data entry is connected to the name of the source organism, to the literature reference and to the protein sequence identifier if this one is available [27].

The enzyme content of BRENDA can be accessed using the enzyme EC number (the enzymes are classified according to the Enzyme Commission list of enzymes [28]), the enzyme name and the organism name. Or the user can browse based on either the taxonomy tree of the organism, in which the enzyme is present or the EC tree of the enzyme. The last classification categorizes enzymes into six different classes based on their functions [29].

BRENDA also provides links to several other databases with a different focus on the enzyme (for example metabolic function, enzyme structure or ontological information on the corresponding gene of the enzyme in question), like BRENDA tissue ontology [30], ExPASy [31], NCBI [32], KEGG [33], PDB [34], PROSITE [35], Uniprot [36], etc.

2.5.2 SABIO-RK

SABIO-RK (System for the Analysis of Biochemical Pathways – Reaction Kinetics) is a curated and a web-accessible (http://sabio.h-its.org/) database which stores information about biochemical reactions and their kinetic properties. The standardized data offered in this database are manually extracted from literature or directly submitted from lab experiments. The manually extracted data is verified by curators, concerning standards, formats and controlled vocabularies, and this process is supported by automated consistency checks [37].

This database contains and merges information about reactions, such as their participants (substrates and products), modifiers, catalyst details, together with the corresponding rate equation, biological sources (organism, tissue and cellular location), environmental conditions (pH and temperature) and reference details. It also includes kinetic parameters in relation to biochemical reactions with no restriction on any particular set of organisms [38]. Data are also related to external sources including KEGG [33], Uniprot [36], ChEBI [39], PubChem [40], NCBI [32] and PubMed [41].

It can be accessed via web-based users interface or automatically via web services that allow a direct automated access by other tools. Both ways support the export of the data together with their annotations in SBML files, allowing users to employ the information as the basis for their simulations models [37].

2.5.3 EXPASY

ExPASy (Expert Protein Analysis System) is a bioinformatics resource portal hosted and operated by different Swiss Institute of Bioinformatics (SIB) groups and partner institutions and it was launched on August 1st 1993 (http://expasy.org/). It has a worldwide reputation as one of the main bioinformatics resources for proteomics but it has now evolved, becoming an extensible and integrative portal accessing many scientific resources, databases and software tools in different areas of life sciences. Scientists can access in a single web portal to a wide range of resources in many different domains, such as proteomics, genomics, phylogeny/evolution, systems biology, population genetics, transcriptomics, etc [42].

ExPASy databases include SWISS-PROT [43] and TrEMBL [44], SWISS-2DPAGE [45], PROSITE [35], ENZYME [46] and the SWISS-MODEL [47] repository and these databases and tools are tightly interlinked and extensively cross-referenced to other molecular biology databases or resources all over the world [31].

The data can be accessed by a variety of query options that are available from the home pages of each of the ExPASy databases. These options allow the users to display and retrieve specified subsets of the database [48].

2.5.4 METACYC

MetaCyc is a non-redundant database which contains extensive information on metabolic pathways and enzymes from many organisms (http://www.metacyc.org/). The pathways, involved in primary and secondary metabolism, as well as the associated compounds, enzymes, and genes are experimentally determined and are curated from the primary scientific literature. It Includes extensive data on individual enzymes, describing their subunit structure, cofactors, activators and inhibitors, substrate specificity, and, in some cases, kinetic constants and also provides commentary and literature references [49].

The goal of MetaCyc is to catalog the universe of metabolism by storing a representative sample of each experimentally elucidated pathway. It is used in a variety of scientific applications, such as providing a reference data set for computationally predicting the metabolic pathways of organisms from their sequenced genomes, supporting metabolic engineering, helping to compare biochemical networks, and serving as an encyclopedia of metabolism.

The data can be accessed searching for pathways, enzymes, reactions or metabolites through the web site or installing MetaCyc on the computer in conjunction with the Pathway Tools software which allows a faster access querying the data using Java or PERL programs [49].

2.6 Text Mining

Life sciences are characterized by the production of a vast and diverse amount of information. In recent years, there has been a significant increase in the rate of published digital information. The high number of scientific publications makes extremely difficult to process, analyze and collect relevant data [50, 51].

Most of this information is presented in the form of scientific literature, such as journal articles, reviews or thesis documents. Normally, these documents are written in natural languages mixed with domain-exclusive terminologies added to numerical data. Thus, they are rich with unstructured data that cannot be understood by machines, making their reuse not an easy task. Thus, the manual extraction of information from literature by humans has become a profitable business operated by information providers [50, 52, 53].

However, aside from expensive, extracting information from text is time consuming and often unreliable, which makes essential to develop tools that contribute directly to an automated process of extracting relevant information from scientific publications. Thereby, transforming the scattered information on natural language and unstructured text in a structured and systematic information is one of the aims of current research efforts [52, 53, 54].

Text mining refers to the automated process of obtaining high-quality information from a text. High-quality information is typically derived through the development of patterns and trends through means such as statistical pattern learning [55, 56]. There are at least as many motivations for doing text mining work as there are types of bio scientists [57].

Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and finally evaluating and interpreting the output [54, 51].

Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities) [54, 56, 57].

A typical application is to scan a set of documents written in a natural language and model the document set for predictive classification purposes or populate a database or search index with the information extracted [57, 58].

Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics [54, 56].

The overarching goal is, essentially, to turn text into data for analysis, via application of natural language processing (NLP) and analytical methods. NLP is a range of computational techniques for analyzing and representing naturally occurring texts for the purpose of reaching human-like language processing, and it can be divided in two levels: the lexical one that makes considerations about words, and the syntactic one that deals with the organization of words in groups within phrases or clauses [2, 51, 54, 59].

In general, biomedical text mining (BioTM) aims to identify and present relevant information in a scientific text, in order to meet the diverse needs of a large community of biomedical scientists [53].

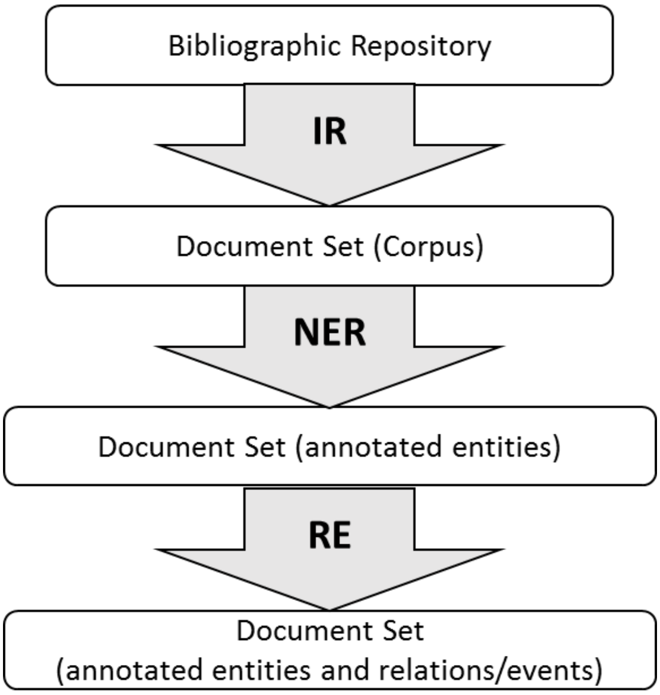

Bibliographic Repository
IR
Document Set (Corpus)
NER
Document Set (annotated entities)
RE
Document Set
(annotated entities and relations/events)

Figure 2.5: Generic pipeline for information extraction.

Commonly, a biomedical text mining tool is divided in two main methods: information retrieval (IR) and information extraction (IE). This in turn is divided in named entity recognition (NER) and relations extraction (RE). Together, these tasks form the pipeline to follow, in order to identify relations (Figure 2.5).

To summarize, relevant publications for a case study are collected (IR), submitted to an annotation of the biological entities (NER) and in the end the entities are associated (RE) in order to identify biological relationships of interest [51, 54, 56].

2.6.1 Information Retrieval (IR)

In its most basic form, information retrieval can be defined as the task of finding a set of relevant documents or specific parts like abstracts in a large text collection.

Usually, those documents are retrieved from bibliographic repositories according to specific queries (normally a Boolean combination of terms or words) defined by the user. Every day, most of us accomplish information retrieval, when we use search engines as Google or Pubmed [56].

The result of this IR step is a set of documents that can further form a Corpus (document set for a specific case study), that later can be submitted to the next step, the IE. Due to the deficiency of annotated corpora available for evaluation and training NER systems, most developers of the systems create their own corpora [53].

Initially, all these techniques have been applied only to abstracts, but text mining technology advanced fast, and these days, full text documents are also the focus. This change has increased the need for better PDF to text conversion, which is still one of the greatest limitations of information extraction. Publications do not have a standard format, which makes it a challenge to develop a conversion algorithm that can fit all needs [52, 56].

2.6.2 Name Entity Recognition (NER)

NER is used for biological entities identification and classification. The NER approaches generally fall into three categories: dictionary-based (lexicon-based), rules-based and machine learning-based [54, 56, 60].

The dictionary approach uses dictionaries to identify the entity occurrences in the text. A dictionary is a collection of vocabulary for a specific domain usually collected from repositories related to the domain. Dictionaries can be built manually or automatically from public sources, such as databases. The main limitation of this method is the dictionary itself because good dictionaries are not available, containing all the terms of interest, making it necessary to create and edit dictionaries according to the case study. A way to overcome this problem is using multi dictionaries [51, 52, 54].

The rule-based approach uses a set of hand-made regular expressions to identify names of entities. Those rules combine grammatical (e.g. parts of speech) and syntactic (e.g. word precedence) information to do the identification. A major disadvantage of this approach is the absence of a pattern of rules, because in different contexts, the rules to apply are different and must be reformulated [53, 54, 56].

Machine learning approaches require a training process to reach a model that will be used in new texts to find the terms. The common machine learning algorithms used in NER are Hidden Markov Models (HMMs) and Conditional Random Fields (CRF). The needs of having a set of annotated training data, considering that creating a set is a costly and time consuming task, is the main limitation of this method [53, 54, 56].

2.6.3 Relation Extraction (RE)

Three basic types of approaches to RE have been prevalent in the biomedical domain: co-occurrence, rule-based systems and statistical or machine-learning-based systems [57].

Co-occurrence looks for concepts that occur in the same part of the text (typically a sentence, but sometimes as large as an abstract) and for a positive relationship between them [56, 57].

Rule-based systems make use of some sort of knowledge, about how the language is structured, how biologically relevant facts are stated in the biomedical literature and knowledge about the sets of things that bioscientists talk about and the kinds of relationships that they can have with one another [54, 60].

In contrast, statistical or machine-learning–based systems operate by building classifiers that may operate at any level, from labelling parts of speech to choosing syntactic parse trees to classify full sentences or documents [2, 54, 57].

Rule-based and statistical systems each have their advantages and disadvantages. For example, rule systems are often assumed to take a significant amount of time to develop. Meanwhile, statistical systems typically require large amounts of expensive-to-get labelled training data. In practice, statistical and rule-based systems can be successfully combined [56, 57].

2.6.4 Tools

Biomedical text mining is not a homogeneous field because texts in different fields have different structures. For instance, the medical records are written differently from scientific articles or sequence annotations. As another example, laboratories create their own protein nomenclatures [51].

In practice, this diversity means that text mining applications are designed and developed to specific types of text and with that purpose, over the past 15 years, several tools (Table 2.1) have been developed [52].

In Table 2.1 a list with 27 tools and their main features, created between 2004 and 2014 is presented. About half of them are supported in the Java language and web services and only one uses Perl (*Textpresso*). Only four used text mining workflows, but almost all have a user interface, at least a web user interface.

In a huge number (24) it is possible to perform information extraction using NER, but only a few work with RE and from those not all of them allow the user to view the relationships found in a user interface.

Among them, only *@Note2* and *Textpresso* allow full text PDF retrieval but @Note2 is the only one that allows manual curation on entities and relations.

So, after the analysis of this table it is possible to conclude that @Note2 is one of the most complete tools available and thus it was selected to support this work.

Table 2.1: List of Text Mining tools and some of their features.

Short Name	**Link**	**Date**	**Tecnologies**	**Text Mining Workflows**	**Information Retrieval**	**PDF/Full Text Retrieval**	**Information Extraction**		**Relations View UI**	**User Interface**
					Pubmed/Medline Search		**NER**	**RE**		
@Note2	www.anote-project.org	2014	Java and Mysql	**x**	**x**	**x**	**x**	**x**	**x**	**x**
PIE	http://www.ncbi.nlm.nih.gov/CBBresearch/Wilbur/IRET/PIE/	2008	N/A		**x** (MEDLINE ONLY)		**x** (Just Proteins)	**x** (Just PPI)		**x** (web)
KLEIO	http://www.nactem.ac.uk/Kleio/	2008	N/A		**x**		**x**			**x** (Web)
FACTA+	http://www.nactem.ac.uk/facta/	2008	N/A		**x**		**x**			**x** (Web)
U-Compare	http://u-compare.org/	2009	Java	**x**			**x**	**x** (event Mine)	**x**	**x**
TermMine	http://www.nactem.ac.uk/software/termine/	N/A	N/A				**x** (without class classification)			
PLAN2L	http://zope.bioinfo.cnio.es/plan2l/plan2l.html	2009	N/A				**x**	**x**	**x**	**x** (Web)
MEDIE	http://www.nactem.ac.uk/tsujii/medie/	N/A	N/A				**x**	**x**	**x**	**x** (Web)
GENIA Tagger	http://www.nactem.ac.uk/tsujii/GENIA/tagger/	2006	unix				**x**			
Yeast MetaboliNER	http://nactem7.mib.man.ac.uk/metaboliner/	2010	web service				**x**			
Chilibot	http://www.chilibot.net/	2004	N/A					**x**	**x**	**x** (Web)
EBIMed	http://www.ebi.ac.uk/Rebholz-srv/ebimed/	2007	N/A		**x**		**x**			**x** (Web)
FABLE	http://fable.chop.edu/	2005	N/A		**x**		**x**			**x** (Web)
GoPubMed	http://www.gopubmed.org/web/gopubmed/	2005	N/A		**x**		**x**			**x** (Web)

Table 2.2: List of Text Mining tools and some of their features. (cont.)

Short Name	**Link**	**Date**	**Tecnologies**	**Text Mining Workflows**	**Information Retrieval**	**PDF/Full Text Retrieval**	**Information Extraction**		**Relations View UI**	**User Interface**
					Pubmed/Medline Search		**NER**	**RE**		
iHOP	http://www.ihop-net.org/UniPub/iHOP/	2004	N/A		**x**		**x**			**x (Web)**
Whatizit	http://www.ebi.ac.uk/webservices/whatizit/info.jsf	2008	Web Service				**x**	**x** (Few examples)		
Becas	http://bioinformatics.ua.pt/becas/	2013	Web Service		**x**		**x**			
PathBinder	http://metnet.vrac.iastate.edu/MetNet_PathBinder.htm	2009	Java		**x**				**x**	**x**
ABNER	http://pages.cs.wisc.edu/~bsettles/abner/	2004	Java				**x**			
Textpresso	http://www.textpresso.org/	2004	Perl		**x**	**x**	**x**			
Ali Baba	http://alibaba.informatik.hu-berlin.de/	2010	Java		**x**		**x**	**x** (Co-occurence)	**x**	**x**
BioIE	http://www.bioinf.manchester.ac.uk/dbbrowser/bioie/	2005	Web	**x**	**x**		**x**	**x** (specific)		**x** (web)
PolySearch	http://wishart.biology.ualberta.ca/polysearch/	2008	Web		**x**		**x**		**x**	**x** (web)
EventMine	http://www.nactem.ac.uk/EventMine/	2013	Web Service				**x**	**x**		
OpenDMAP	http://opendmap.sourceforge.net/	2008	Java				**x**	**x**		
BioContext	http://www.biocontext.org	2012	Java,Webservices,external tools	**x**	**x**		**x**	**x**		

3 Finding Kinetic Parameters with @Note2

3.1 @Note2

@Note2 (http://www.anote-project.org/) is a platform for biomedical text mining that is a result of the advances between three distinct classes of users: biologists, software developers and text miners. It is the result of a joint effort between the University of Minho and the company SilicoLife. It is implemented over the AIBench framework, a Java desktop application framework that follows a Model-View-Controller (MVC) design pattern [2, 61].

This framework facilitates connectivity, integration and execution of data operations since the inputs / outputs are well defined. The main idea is to provide a powerful programming model, easy to use and non-intrusive, for fast development of applications characterized by providing the following features:

- The logic can be decoupled from the user interface (UI).
- Decoupling interconnection of operations and thinking of their concepts.
- Increasing the reuse of code, forcing the user to "think before programming".

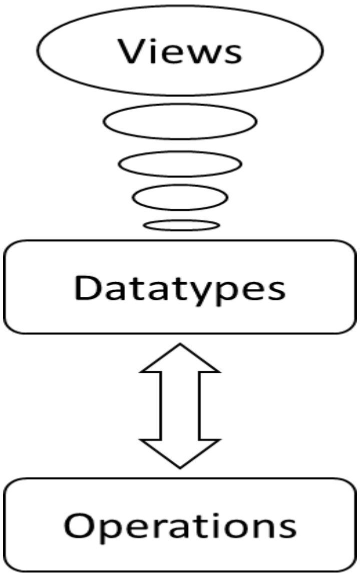


Figure 3.1: AIBench MVC model.

The AIBench platform was constructed to facilitate the development of a wide range of applications based on generic information with basic input-process-output cycles in which the framework is responsible for connecting these tasks. AIBench was developed as a joint effort between the Universities of Minho and Vigo. Following the definition of the MVC model, AIBench is based on three key concepts that are constant in all of its applications: operations, data types and display modes or views (Figure 3.1) [54, 61].

The programmer only needs to worry about how to separate and organize the specific problem on objects of those three entities. The AIBench performs the rest of the work, generating a final and completely executable application that includes:

- automatic generation of a graphical interface where the user can select and perform the functions implemented;
- automatic retrieval of user parameters in a given operation when necessary;
- execution of defined operations, gathering results and keeping them available for reuse in a shared area;
- showing the results through custom display modes.

Programming on AIBench platform is a light task, since it makes use of Java annotations, facilitating the implementation of MVC to users. Furthermore, applications can be developed separately and added as components, called plug-ins. Each of these components contains a set of AIBench objects that allows reuse and integration of any functionality built on this platform[2, 54, 61].

The interoperability, flexibility and modularity present in the @Note2 platform enables the application and development of new BioTM systems produced by researchers while keeping the same user-friendly interface for biomedical data research [2, 62].

The workbench is meant for both BioTM research and curation. It provides an intuitive Graphical User Interface (GUI), that does not require any knowledge about the workbench or implementation technique, which supports regular curation activities. At the same time, it is also meant for people with programming skills that might wish to extend the workbench capabilities. @Note sustains the general workflow of BioTM by fully covering all activities performed in manual curation, it supports the retrieval, processing and annotation of documents as well as their analysis at different levels. Its main features are: biomedical text mining pipelines, information retrieval algorithms, name entity recognition, lexical resources and corpora management, relationship/event extraction and manual curation of entities and relations/events [2, 54,6 2].

Due the fact that @Note2 was built based on the AIBench framework, it follows the same three key concepts: datatypes, operations and views. Operations and datatypes are used in problem modelling, while views display data in a "friendly" way [2, 62].

@Note2 can be separated in four main functional modules (Figure 3.2): Publication Manager Module (PMM), Corpora Module (CM), Resources Module (RM) and the Corpus processes module (includes NER and RE Schema and Curator). PMM, CM and RM are examples of datatypes that can be generated, updated or removed by operations, while NER and RE are operations that can be applied to the datatypes, and their results are datatypes that can suffer more operations [2, 54].

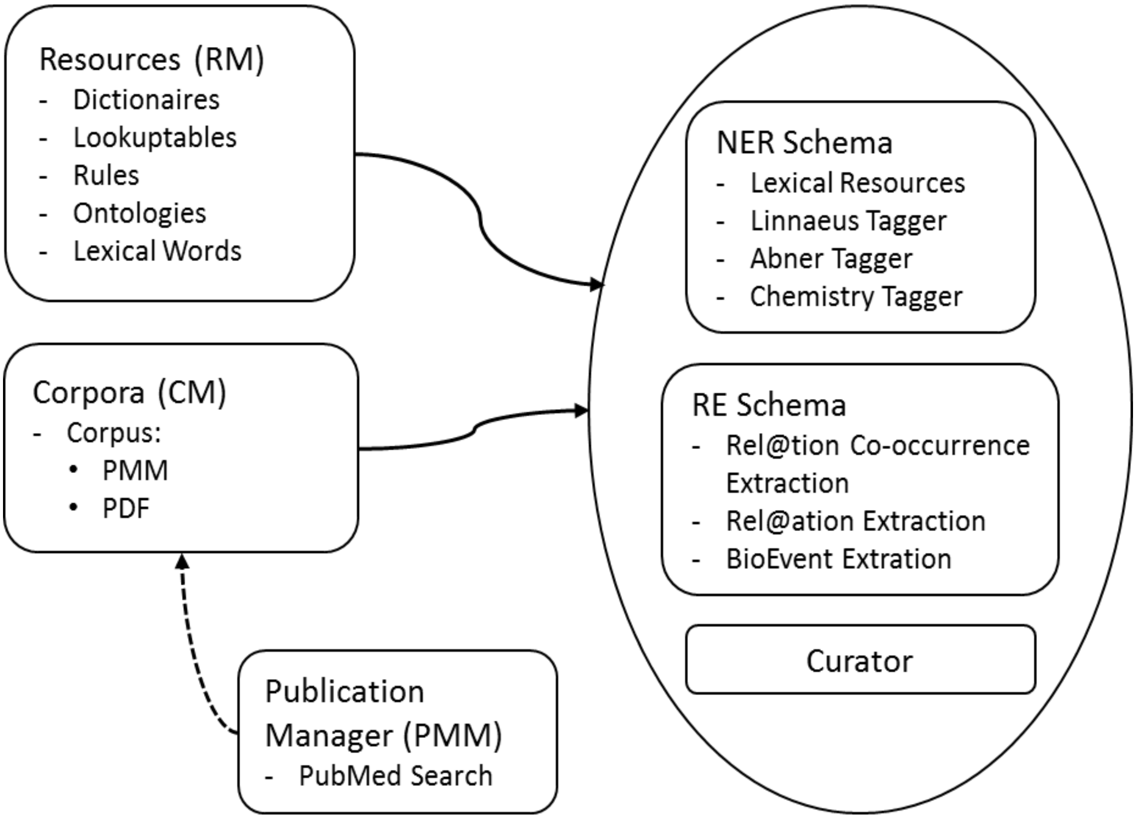


Figure 3.2: @Note2 structure.

The PMM, which is the one responsible for the IR step, has the functionality of searching and clustering documents from repositories based on queries made by the user. The CM receives those publications from PMM or imports the PDF files to form a Corpus. If the imported PDF files are listed in PubMed [41], this module has the capability of recognizing it and retrieving the different corresponding information, like PubMed ID, authors, abstract, etc [2, 54].

The RM is responsible for the management of the lexical resources that are used in IE processes. These lexical resources include the dictionaries, rules, ontologies, lookup tables and lexical words that can be dynamically created by the user. To each Corpus created, different IE processes can be applied, together with the use of some resources, to identify and extract bio entities and their relationships [2, 54].

The processes module can be divided in three main processes: the NER process, which contains a hybrid dictionary-based and rules-based denominated Lexical resources, the Linnaeus tagger, Chemistry tagger and ABNER tagger systems to identify bio-entities; the RE process that encompasses the co-occurrence extraction process, the Rel@tion extraction, an NLP relation extraction system and finally the BioEvent extraction, a machine learning event extraction of bio-entities. The Curator process allows to add/edit/remove entities and relations from the IE processes results and to start a manual curation process [2, 54].

3.2 Pipeline

Based on different features, properties and processes available on the @Note2 tool, we designed a pipeline to identify and extract kinetic data from text.

As shown in Figure 3.3, we will work with PDF files and, as a first step to create the Corpus, we used the @Note2 built-in functionalities to convert our full text PDFs to text. We chose the Lexical Resources as the NER process type. This type of NER process is dictionary and rule-based, and so, before running it, the different lexical resources, such as dictionary, lookuptables, rules and ontology (resources in Figure 3.3), had to be created taking into account the tool possibilities.

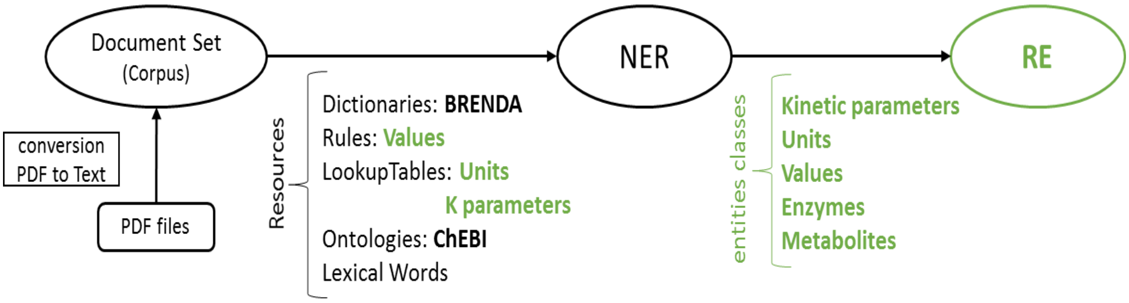


Figure 3.3: Main steps of the pipeline, shown in black the existing resources and in green the new ones that were created in @Note2 in the scope of this thesis.

In lookuptables it is possible to define a list of terms. According to the class that we attribute to each term, it will be grouped inside the list and later identified and annotated in the text.

For specific cases of characters set or, like our case, a set of values, we can take advantage of the rules resource, in which using regular expressions allows the identification in the text of specific patterns. For each rule created, a class has to be attributed, which can be the same class for all of the rules or not.

To create the dictionary resource, we have to select if it will be a protein or an enzyme dictionary, select the file that contains that information and select the database source (Uniprot , ChEBI, EntrezGene, BRENDA, KEGG, BioCyc, NCBI Taxonomy or Biowarehouse) of the information contained in the file.

For the ontology creation the process is similar: the file with the information regarding the type of ontology that will be used has to be upload to the tool.
The NER will annotate the entities based on the different classes defined within these resources. The details of their creation will be explained later, in section 4.2.

For the annotation, NER allows the user to choose some options regarding the way that entities will be recognized and identified in the text:

- case sensitive: true or false;
- pre-processing: no/stop words/pos-tagging;
- normalization: yes or no.

For example, if we choose “case sensitive false”, it will not matter if the term is listed in upper or lower case, so it will be recognized in the text in both forms. However, if the selection is “case sensitive true”, we have to be aware that in our resources lists the terms have to be listed in all forms needed to be found (for example, upper and lower case or the first letter upper and the others lower).

Explaining in a succinct way, the NER process begins by identifying the sentences in a document, assuming that in each period followed by a space a new sentence starts. Each sentence of the document is thus analyzed separately. For each sentence, the words are identified based on the existing spacing between them, and for each word it is verified whether there is a complete match with any of the terms listed in the different resources used.

The NER GUI holds several steps in which the user can select the different options mentioned above, as well as all the resources created.

In further analyses, we realized that the RE process already developed within the @Note2 did not have the most suitable options to extract the relations between values, kinetic parameters, enzymes and metabolites, so we preferred to develop a new specific RE process. This new process works based on the lexical resources and on rules defined by us. The basis for the RE process is the NER annotation results.

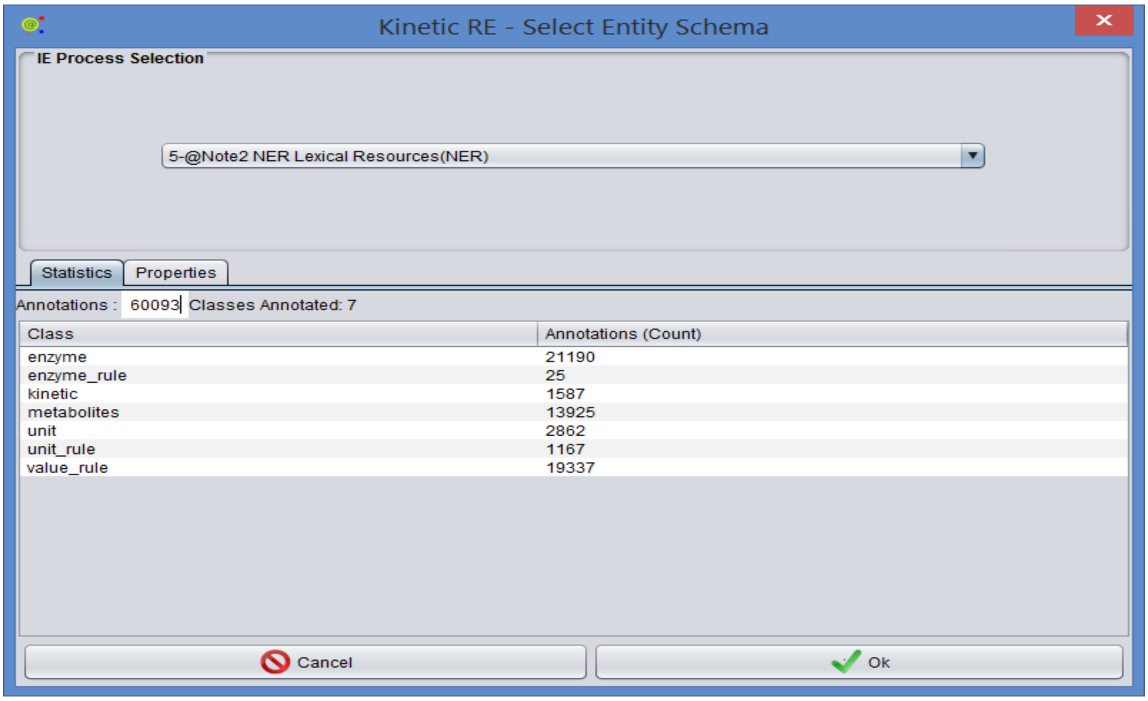
The algorithm of this new process is based on a set of rules that define the patterns to look up for and the relations between these pattern and entities previously annotated by the NER process.

3.3 Kinetic RE plug-in

A new package in the @Note2 source code was created for our Kinetic RE plug-in, in which a new class for the algorithm itself and some other classes to extend methods and global settings implemented in the tool are included.

3.3.1 User Input Interface

After the selection of “Kinetic RE” on @Note2 Clipboard, a first GUI will be launched (Figure 3.4), in which the user can choose the NER process that will be submitted to the RE and visualize some statistics about the annotation process of the selected NER (number of annotations to each class).



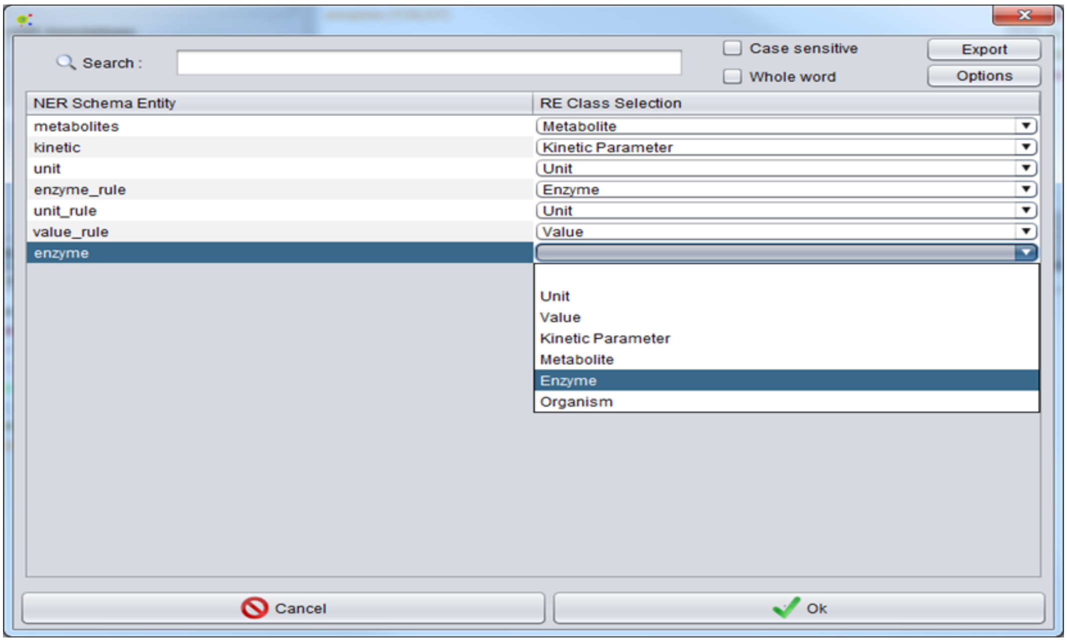
Kinetic RE - Select Entity Schema
IE Process Selection
5-@Note2 NER Lexical Resources(NER)
Statistics
Properties
Cancel
Ok

Annotations : 60093 Classes Annotated: 7

Class	Annotations (Count)
enzyme	21190
enzyme_rule	25
kinetic	1587
metabolites	13925
unit	2862
unit_rule	1167
value_rule	19337

Figure 3.4: Kinetic RE first input GUI

After confirming the selection, a second GUI appears (Figure 3.5). In this, the user should do the mapping between the resources that were used in the NER and the classes that are defined and available for the RE process. For example, the resource “unit” (lookuptable) and the resource “unit_rule” (set of rules), both used to identify and annotate units, should be mapped to the RE class “Unit”, as shown in Figure 3.5.



NER Schema Entity	RE Class Selection
metabolites	Metabolite
kinetic	Kinetic Parameter
unit	Unit
enzyme_rule	Enzyme
unit_rule	Unit
value_rule	Value
enzyme	

Figure 3.5: Kinetic RE second input GUI.

All these user-defined options will be received as input parameters to the algorithm.

3.3.2 RE process

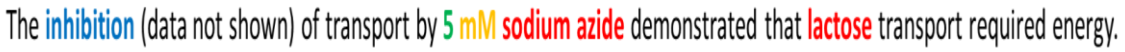
As stated above, the algorithm will receive as input parameters:

- the NER process, as well as the configurations chosen in the process and the Corpus in analysis;
- the mapping between NER resources and RE classes;
- and most importantly, the set of annotated documents.

The algorithm starts by mapping the NER resources to the RE classes (kinetic parameters, values, units, metabolites and enzymes), according to the user selection. These RE classes were created previously and a score set by default was assigned to each of them.

The result set of NER can have lots of documents, but the analysis will be performed separately for each one, therefore the algorithm will access the list of annotated entities per document.

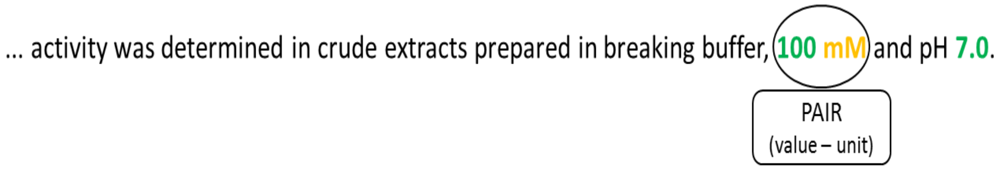
For each document, the analysis is done sentence by sentence. At first, the annotated entities are identified in the sentence. Information regarding the class where each entity belongs and the start and end position of the word in the sentence are already associated to each annotated entity. Then, based on that class classification, separated lists of enzymes, kinetic parameters and metabolites that were annotated in that sentence are created. For the example shown in Figure 3.6, the kinetic parameters (blue annotation) list will have one term, the metabolites (red annotation) list will have two entities and the enzymes list stays empty.



The inhibition (data not shown) of transport by 5 mM sodium azide demonstrated that lactose transport required energy.

Figure 3.6: Example of an annotated sentence.

For each sentence, the system starts recognizing the value-unit pairs. This is done by looking for entities of the classes "Value" and "Unit" annotated with a distance of between 0 and 3 spaces (this distance is set by default on the code). For example, as shown in Figure 3.7, two values (green annotation) were annotated as belonging to the "Value" class, but only the first one is followed by an entity of "Unit" class (yellow annotation), so, in this case, only one pair (value– unit) will be identified by the algorithm.



... activity was determined in crude extracts prepared in breaking buffer, 100 mM and pH 7.0.

Figure 3.7: Example of a sentence with only one value-unit pair.

After this first recognition, the algorithm, using the list of pairs that occurs in the sentence, will check for each of them if it is possible to find a relation.

If only a pair was identified, the whole sentence is considered the space for the possible relation. At this point, the algorithm analyses all the lists created previously (enzymes, metabolites, k parameters). If the lists are empty (like the case shown in Figure 3.7), this is considered as the simplest relation possible and it will receive the minimum score (set by default).

If the lists have terms, first the minimum score regarding the pair is added to the relation score and then the algorithm checks if the entities occur on the left (entities positions are lower than the pair/value start position – Sp_v in Figure 3.8), or on the right (entities positions are higher than pair/unit end position – Ep_u in Figure 3.8).

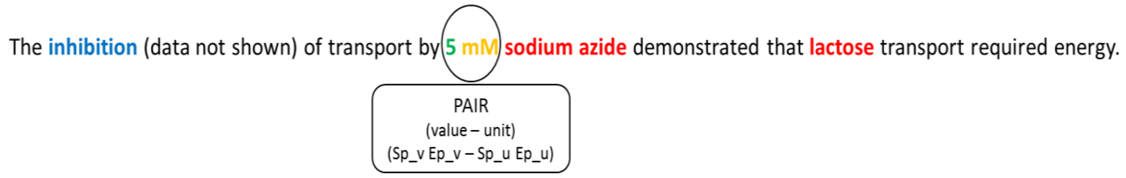

The inhibition (data not shown) of transport by 5 mM sodium azide demonstrated that lactose transport required energy.
PAIR
(value – unit)
(Sp_v Ep_v – Sp_u Ep_u)

Figure 3.8: Example of a sentence with only one value-unit pair and annotated entities from "metabolites" and "kinetic parameters" classes.

For each entity identified and added to the relation, the algorithm adds to the relation score the corresponding score of its class. In the case example of Figure 3.8, the relation score will get the value corresponding to the pair plus the corresponding score of the kinetic parameters class (blue annotation) plus twice the score defined to the metabolites class (red annotation). So, considering the following values for each of the classes: pair->10; metabolites->100; enzymes->1000 and kinetic parameters->10000, the example showed in Figure 3.8 will received a score of 10210 = 10 + 2*100 + 10000.

If the sentence has two value-unit pairs (Figure 3.9), for the first pair the space considered to find/extract a relation will be between the beginning of the sentence and the start position of the second pair (Sp_v2 in Figure 3.9). For the second pair the space is between the end position of the previous pair (Ep_u1 in Figure 3.9) and the end of the sentence. Based on this, the rest of the method, checking entities on the left and on the right of the pair, adding those entities to the relation and their corresponding scores to the relation score, occurs for each pair the same way that was described above for the one pair example.

Using the scores defined previously to each class, to the example presented in Figure 3.9, the relation with pair 1 will get a score of 20110 and the relation with pair 2 a score of 10110.

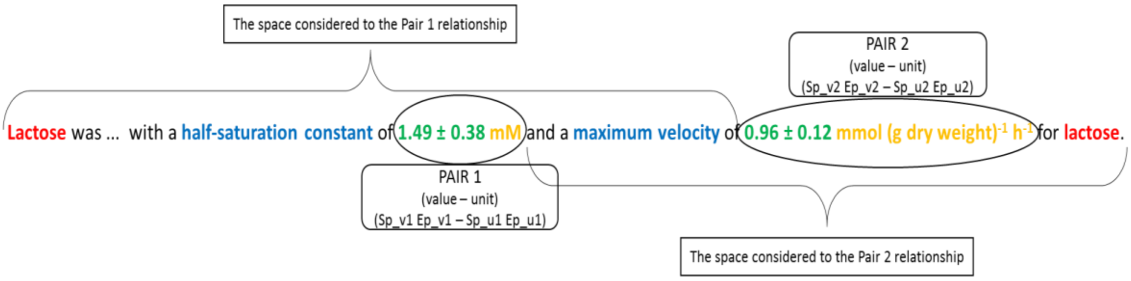

The space considered to the Pair 1 relationship
PAIR 2
(value – unit)
(Sp_v2 Ep_v2 – Sp_u2 Ep_u2)
Lactose was ... with a half-saturation constant of 1.49 ± 0.38 mM and a maximum velocity of 0.96 ± 0.12 mmol (g dry weight)-1 h-1 for lactose.
PAIR 1
(value – unit)
(Sp_v1 Ep_v1 – Sp_u1 Ep_u1)
The space considered to the Pair 2 relationship

Figure 3.9: Example of a sentences where two value-unit pairs were identify.

Observing the example shown in Figure 3.9, it is possible to realize that with this method, the information annotated between pairs 1 and 2 will be added to both relations, creating an increase of redundancy. In this case it is obvious that the kinetic parameter annotated in blue belongs to the pair 2, but since there is no straight way to code this kind of separation, it was decided to allocate the information in the middle to both pairs, with the intention of minimizing the probability of losing important information.

In a generic way, as shown in Figure 3.10, for sentences with more than one pair, the relation space for the first pair begins in the beginning of the sentence and ends before the first position of the second pair (Sp_v2). For the last pair (in this case is 3 but can be “n”), the relation starts after the last position of the previous pair (Ep_u2 or Ep_un-1) and ends in the end of the sentence. For the pairs in the middle, the relation goes from the end of the previous pair (Ep_u1 or Ep_upair-1) to the start position of the following pair (Sp_v3 or Sp_vpair+1).

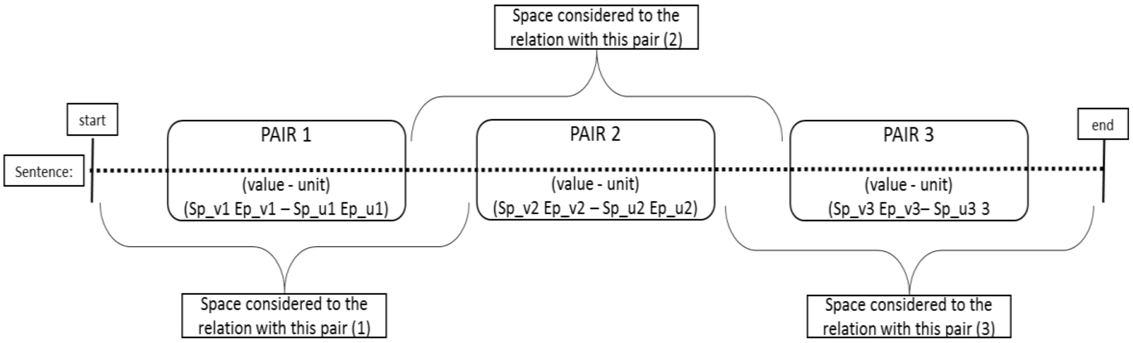

Space considered to the relation with this pair (2)
start
end
Sentence:
PAIR 1
(value - unit)
(Sp_v1 Ep_v1 – Sp_u1 Ep_u1)
PAIR 2
(value - unit)
(Sp_v2 Ep_v2 – Sp_u2 Ep_u2)
PAIR 3
(value - unit)
(Sp_v3 Ep_v3– Sp_u3 3
Space considered to the relation with this pair (1)
Space considered to the relation with this pair (3)

Figure 3.10: Generic scheme for a sentence with three value-unit pairs.

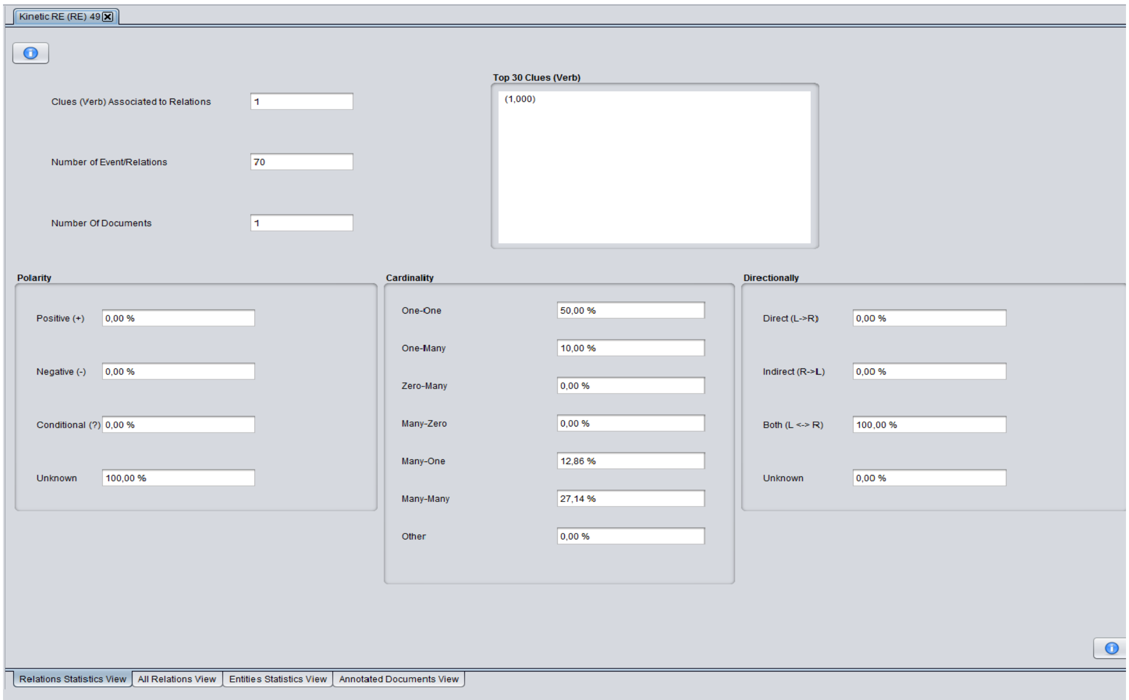
The algorithm was designed in order to treat the values presented in tables as specific cases, because usually after the PDF conversion to text, they appear together in lines and sometimes with the units or column/row legends mixed between them.

In each sentence, the algorithm looks for a set of values closely annotated, and if it occurs the whole set of values works like a pair and the method to construct the relation and calculate the score is the same as the described above for a single pair.

Per document the algorithm creates a data structure where all the relations found in each sentence and the corresponding scores are saved. So, in the end it will create a list of relations ordered by score and launching it in an output GUI.

3.3.3 Output Interface

The output of the RE process is a GUI with four separators/views. In the first one, the "Relations Statistics View" (showed in Figure 3.11) it is possible to see some RE statistics, like the number of documents analyzed and the number of relations extracted from those documents.



Clues (Verb) Associated to Relations: 1

Number of Event/Relations: 70

Number Of Documents: 1

Top 30 Clues (Verb)
(1,000)

Polarity

Positive (+)	0,00 %
Negative (-)	0,00 %
Conditional (?)	0,00 %
Unknown	100,00 %

Cardinality

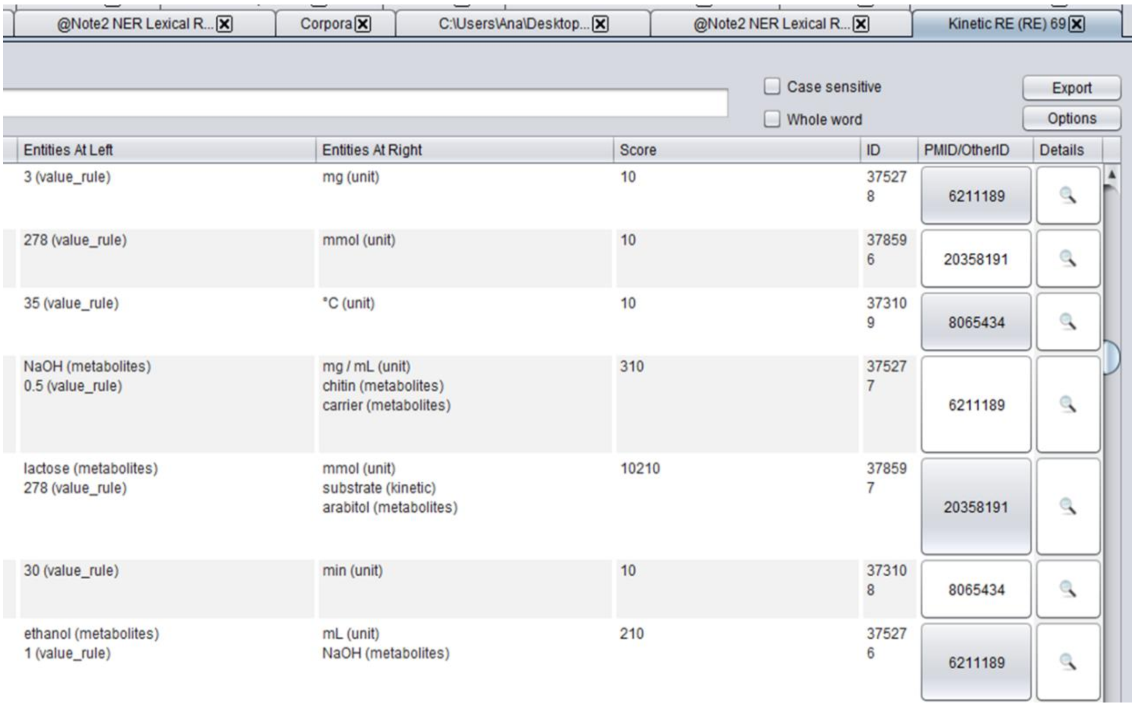
One-One	50,00 %
One-Many	10,00 %
Zero-Many	0,00 %
Many-Zero	0,00 %
Many-One	12,86 %
Many-Many	27,14 %
Other	0,00 %

Directionally

Direct (L->R)	0,00 %
Indirect (R->L)	0,00 %
Both (L <-> R)	100,00 %
Unknown	0,00 %

Figure 3.11: Kinetic RE output.

In the second separator (Figure 3.12), named "*All Relations View*", all the relations extracted are listed. The entities found on the left of the pair are shown in a different column from the entities found on the right.



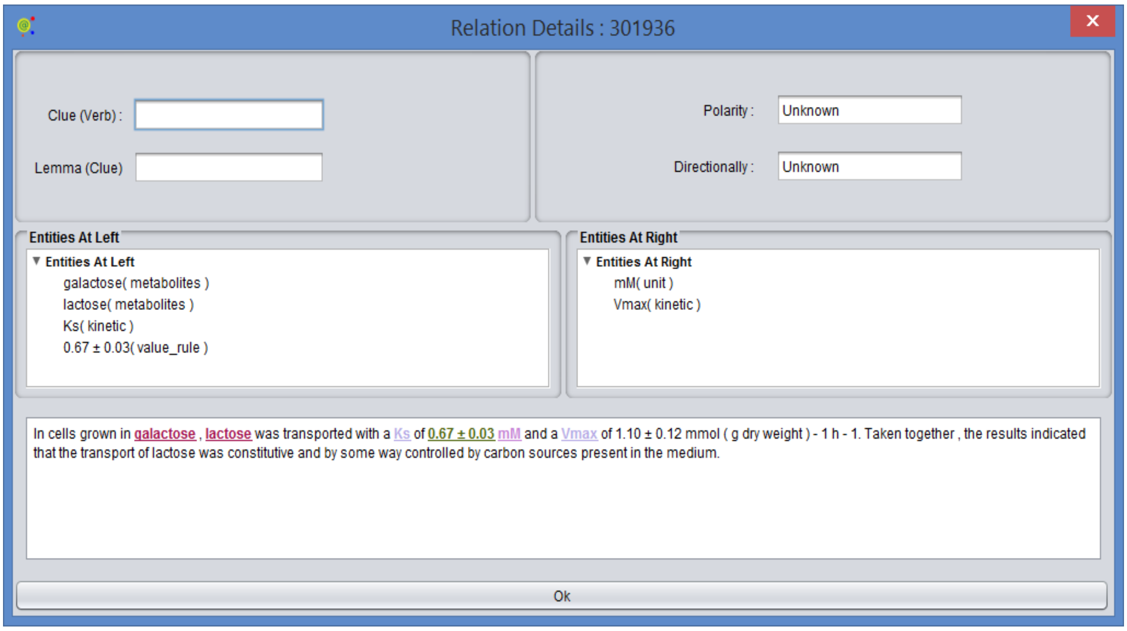
□ Case sensitive

□ Whole word

Entities At Left	Entities At Right	Score	ID	PMID/OtherID	Details
3 (value_rule)	mg (unit)	10	375278	6211189	
278 (value_rule)	mmol (unit)	10	378596	20358191	
35 (value_rule)	°C (unit)	10	373109	8065434	
NaOH (metabolites) 0.5 (value_rule)	mg / mL (unit) chitin (metabolites) carrier (metabolites)	310	375277	6211189	
lactose (metabolites) 278 (value_rule)	mmol (unit) substrate (kinetic) arabitol (metabolites)	10210	378597	20358191	
30 (value_rule)	min (unit)	10	373108	8065434	
ethanol (metabolites) 1 (value_rule)	mL (unit) NaOH (metabolites)	210	375276	6211189	

*Figure 3.12: Part of the "*All Relations View*" output.*

In this view it is possible to choose to see a specific relation in detail (details column/option). A new GUI will appears on top of the current one, showing the relation and its entities (Figure 3.13).



Clue (Verb) :

Lemma (Clue)

Polarity : Unknown

Directionally : Unknown

Entities At Left

- Entities At Left
  - galactose(metabolites)
  - lactose(metabolites)
  - Ks(kinetic)
  - 0.67 ± 0.03(value_rule)

Entities At Right

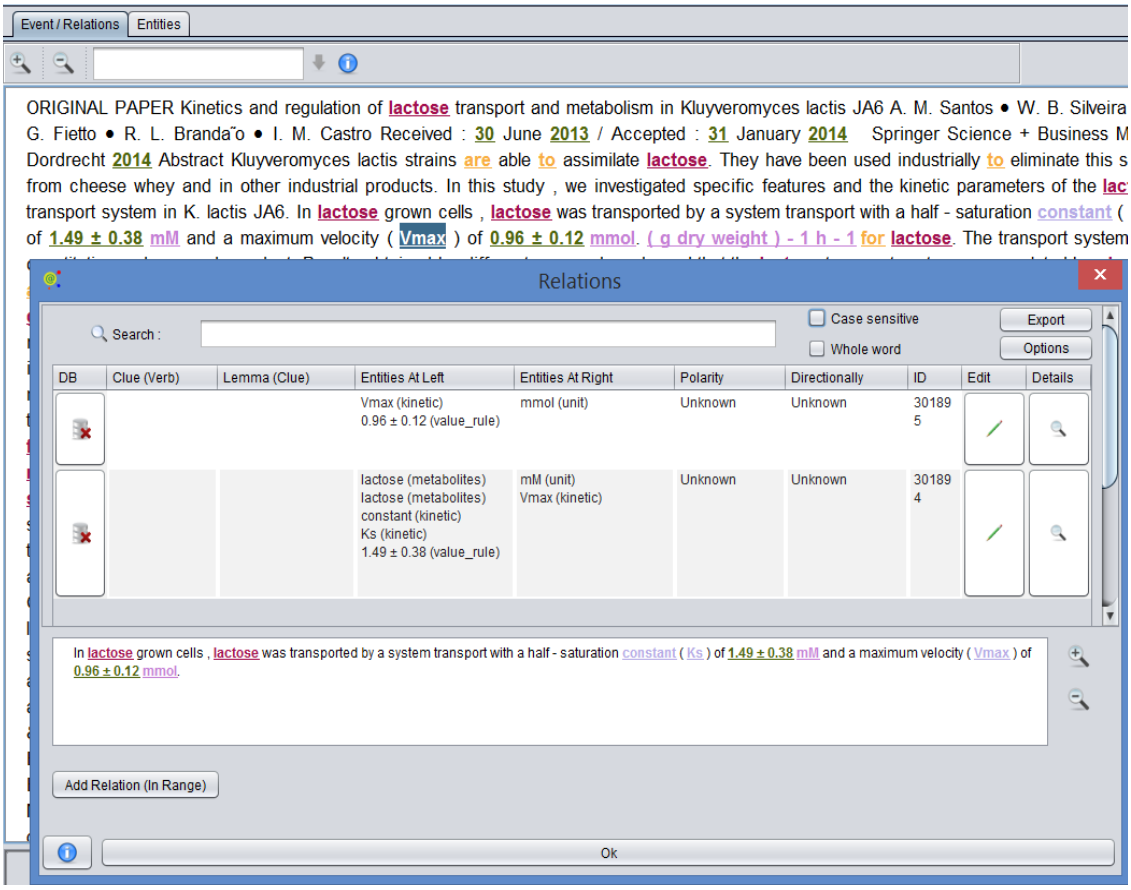
- Entities At Right
  - mM(unit)
  - Vmax(kinetic)

In cells grown in galactose , lactose was transported with a Ks of 0.67 ± 0.03 mM and a Vmax of 1.10 ± 0.12 mmol (g dry weight) - 1 h - 1. Taken together , the results indicated that the transport of lactose was constitutive and by some way controlled by carbon sources present in the medium.

Figure 3.13: View of the details for one relation.

On the third view, "*Entities Statistics View*", some statistics regarding the NER process are displayed, like the number of entities annotated to each class and the top terms of the annotation.

The last view, the "*Annotated Documents View*" is one of the most interesting, as here the user can choose which document he/she wants to see in detail and it will appear in a new GUI, the "*Curator View*". Here it is possible to see the entire converted text with all the annotated entities and relations identified. It also allows the user to create new relations between the annotated entities and edit the relations extracted, i. e., it is possible to add or remove entities from a specific relation.



ORIGINAL PAPER Kinetics and regulation of lactose transport and metabolism in Kluyveromyces lactis JA6 A. M. Santos • W. B. Silveira
G. Fietto • R. L. Branda˜o • I. M. Castro Received : 30 June 2013 / Accepted : 31 January 2014 Springer Science + Business M
Dordrecht 2014 Abstract Kluyveromyces lactis strains are able to assimilate lactose. They have been used industrially to eliminate this s
from cheese whey and in other industrial products. In this study , we investigated specific features and the kinetic parameters of the lac
transport system in K. lactis JA6. In lactose grown cells , lactose was transported by a system transport with a half - saturation constant (
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. The transport system

DB	Clue (Verb)	Lemma (Clue)	Entities At Left	Entities At Right	Polarity	Directionally	ID	Edit	Details
			Vmax (kinetic) 0.96 ± 0.12 (value_rule)	mmol (unit)	Unknown	Unknown	30189 5		
			lactose (metabolites) lactose (metabolites) constant (kinetic) Ks (kinetic) 1.49 ± 0.38 (value_rule)	mM (unit) Vmax (kinetic)	Unknown	Unknown	30189 4		

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.

Figure 3.14: Part of the Curator View, together with the specific view to the “Vmax” term, selected in blue.

Also in this view, if the user clicks on an entity that is contained in one or more relationships, another GUI will open, showing all the relations where the selected entity was added, as showed in Figure 3.14. Here, it is possible to edit the relation or create a new relation.

4 Case Study: kinetic parameters of *Kluyveromyces lactis*

As mentioned earlier, this work is integrated into a larger project in which six organisms are considered. However, for this case study we decided to work with *Kluyveromyces lactis*, because it is not so well characterized and consequently does not have so much information associated when compared to organisms like *S. cerevisiae* or *E. coli* (see Table 1.1). Before applying the Text Mining pipeline, the information available in databases was evaluated and extracted. For this process, having less information available facilitates the analysis of the data taken from the database.

For the text mining method, we can use a small sample (10 papers) as a significant sample of the total number of papers available to get relevant information to proceed with the pipeline, validating the NER identification and test the plug-in.

4.1 Retrieval of kinetic information from databases

After an extensive analysis of several databases (DB) that contain information regarding enzymes and their kinetic parameters, the BRENDA database was chosen, since it provides most information of interest for this work in a way that is easy to access and extract. For each enzyme it is possible to get information concerning reactions, inhibitors and activators, K_i and K_m values, substrates and products. And, most importantly for our objective, all the different searches can be performed specifically per organism.

In order to extract from the database all the possible information, we follow a pipeline (Figure 4.1), in which a Perl script (uses SOAP, API available on BRENDA database) was written to access the data on the DB.

SOAP offers a variety of specific methods to access all kind of information available in BRENDA, like Ligand structure Id, Reference by Id, Reference by Pubmed Id and Activating Compound, among others. To use those methods, the user only needs to write a script in which those have been incorporated.

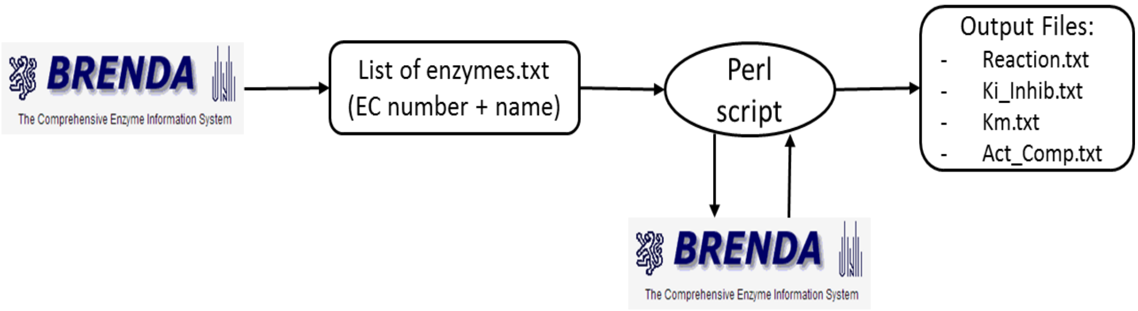


Figure 4.1: Pipeline used to retrieve data from the BRENDA database.

For the organism *Kluyveromyces lactis*, the BRENDA database was queried and the result was a list with 45 enzymes (Figure A 1). That list, containing the 45 EC numbers, the corresponding enzyme names and the organism/strain, was converted into a tab delimited txt file (input file) to be submitted to the script.

The script reads the input file and saves the enzymes list into a hash table (Figure A 2). For each enzyme specific functions are used to access BRENDA, according to the type of information that we wish to recover:

- *getReaction Function (*Figure A 3*);*
- *getInhibitors Function (*Figure A 4*);*
- *getActivatingCompound Function (*Figure A 5*);*
- *getKmValue Function (*Figure A 6*);*
- *getKiValue Function (*Figure A 7*);*
- *getReference Function (*Figure A 8),

Using the EC number and the organism name as input, each function accesses the database separately and retrieves as an output a string containing all entries regarding the method. The script parses the string received, according to the different fields and saves the information, while going through the list of enzymes. In the end, it generates the corresponding output files (tab delimited text file, Table A 1 - Table A 4) mentioned in the pipeline.

Table 4.1: Information obtained from BRENDA for the organism Kluyveromyces lactis.

	Reactions (organism specific)	**Number of enzymes with:**				**Total number of enzymes (listed in BRENDA)**
		Ki values	Inhibitors	Km values	Activators	
K. lactis	7	**3**	**9**	**7**	**6**	45

As we can see, from the analysis of Table 4.1, despite the number of enzymes listed in the database for this organism, only for a few of them it was possible to obtain information on their kinetics (the same lack of information was observed for the other organisms, Table A 5).

Furthermore, during the search and analysis done in the database we detected that sometimes some values are attributed to an organism by homology/similarity with another one, because if we check the paper associated to the values we realize that the work and the results were obtained for another organism. Therefore, despite this database being manually curated, it is important to be aware that not all available data are reliable.

Thus, the results obtained with this method and its limitations, have confirmed and highlighted the importance of the use of text mining for kinetic data collection in the literature.

4.2 Collection of kinetic data directly from the literature

Based on the previous results, we selected some papers and others were randomly picked from the total related to *Kluyveromyces lactis.* Since @Note2 allows uploading PDF files, a corpus with 10 papers was created.

The analysis of these articles helped in the creation of our resources (Figure 4.2). The type of resources available are listed in the @Note2 Clipboard, as showed in Figure A 9, but we still need to create them.

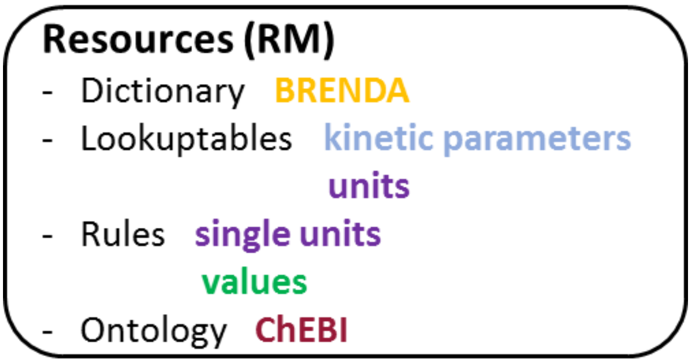


Figure 4.2: Scheme of resources created in @Note2.

So, to create the dictionary regarding enzymes names, which will be used during the NER process to identify enzymes in the documents, we uploaded a BRENDA file that was downloaded directly from the database on August 31st of 2012. This file contains an extensive list with 4682 enzymes terms and 55743 synonyms.

After the creation of this resource, it is possible to search terms in the dictionary. For example, if we perform a search using the term "*glucose-6-phosphate dehydrogenase*" as query, the result is that one term appears as the exact match, several terms appears listed as synonyms (Table 4.2) and it is also possible to see the external ID (source), in this case the EC number (1.1.1.49) from BRENDA. The searches can be performed looking for an exact match of the word/words in the query, partial match or only looking for synonyms terms.

Table 4.2: Some of the results of a search performed in the resource dictionary in @Note2.

Enzyme	Synonyms
glucose-6-phosphate dehydrogenase	D-glucose-6-phosphate:NADP oxidoreductase
	G6PDH5
	glucose 6-phosphate dehydrogenase (NADP)
	Glc6PDH
	Glc6PDH
	GPD
	G-6-PDH
	D-glucose 6-phosphate dehydrogenase

Besides the dictionary, two lookuptables were also created: "kinetic parameters" and "units". The first one was filled with terms related to enzyme kinetics, for which some examples are presented in Table 4.3.

This list can be considered one of the most important resources, taking into account that the purpose of the developed pipeline is to collect kinetic data based on relations extracted from documents.

Table 4.3: Twelve examples out of thirty terms listed in the "Kinetic Parameters" lookuptable.

Term Nr	Term	Class
1	km	kinetic
2	Km	kinetic
3	KM	kinetic
4	kM	kinetic
5	ki	kinetic
6	Ki	kinetic
7	Vmax	kinetic
8	vmax	kinetic
9	Michaelis constant	kinetic
10	Michaelis-Menten constant	kinetic
11	Michaelis-Menten	kinetic
12	Michaelis-Menten kinetics	kinetic

As it can be observed by comparing term numbers 1, 2, 3 and 4 presented in Table 4.3, for each term we intend to introduce in the lookuptable, we must consider all possible letters formatting, like one letter uppercase and the other lowercase, both lowercase or both uppercase, etc. The more terms are entered in the list, as well as more of their possible variations, the lookuptable will be more complete and thus, the annotation process will be more assertive.

To the other lookuptable, the "Units", the creation process is more a less the same. But, in this case, besides the problem of letters formatting, there is an issue regarding spaces between the words, as can be observed in some of the terms presented in Table 4.4. During the terms insertion, we realize that lookuptables do not allow terms with only one letter, as in the case of units for gram (g) and velocity (v). To overcome this limitation, we will use the rules resources to catch in the text specific cases, like the two examples mentioned above.

Table 4.4: Twelve examples out of ninety two terms listed in the "Units" lookuptable.

Term Nr	Term	Class
1	mM	unit
2	mmol	unit
3	mL-1	unit
4	mg protein -1 h -1	unit
5	mg protein-1 h-1	unit
6	nmol min-1	unit
7	nmol min -1	unit
8	mol -1	unit
9	mM s-1	unit
10	mM s -1	unit
11	mmol l-1	unit
12	mmol l -1	unit

Rules resources make use of Java regular expressions to identify specific patterns defined for each rule. As one of the goals of the annotation is to detect values that may be related to kinetic parameters, using these rules is the only way to identify them.

In Table 4.5, it is possible to analyze some of the rules that have been created. The first three are specific to catch the single letter v and the others listed below identify the values.

For example, the rule number 9 recognizes positive or negative integers while rule number 8 recognizes positive or negative decimal values.

Table 4.5: Nine examples out of sixteen rules created and examples of what they should identify.

Nr	Regular Expressions (Java)	Examples
1	?=\s+(V)\s+	V
2	?=\s+(v)\s+	v
3	?=\s+(v=)\s+	v=
4	\s(\(-{0,1}\d+\.{0,1}\d*\s±\s\d+\.{0,1}\d*\)\s{0,1}x\s{0,1}10exp-{0,1}\d+)\s	(-67 ± 8.23) x 10exp-5 (6.89 ± 46) x 10exp18
5	?=\s+(-{0,1}\d+\.{0,1}\d*\s{0,1}±\s{0,1}\d+\.{0,1}\d*)\s+	7.8 ± 23 ; -45 ± 1.23
6	\s(\d+\.{0,1}\d*\s{0,1}-{1}\s{0,1}\d+\.{0,1}\d*)\s	56.35 – 60; 96 – 98.678
7	\s(-{0,1}\d+\.{0,1}\d*\s{0,1}x\s{0,1}10exp-{0,1}\d+)	99.99x10exp9; 7x10exp-6
8	?=(\s+(-{0,1}\d+\.\d+)\s+)	]-99.99999; 99.99999[
9	?=(\s+(-{0,1}\d+)\s+)	]-99999; 9999[

Rule number 5 is a bit more complex, because it recognizes an "expression", in which the first member can be a positive or negative, decimal or integer value, followed by the plus-minus sign, and in the second member a positive integer or decimal value.

Lastly, the ontology resource was created, in which the process is identical to the dictionary creation, but instead of having a list of enzymes, in this case a list of metabolites will be created. From the ChEBI database an "obo" file, the 114 release version, was downloaded (on April 3rd of 2014). And then, the file containing 45436 terms and 266644 synonyms was uploaded to the ontology resource. To each term entree, an id, a name and synonyms are associated and, in some cases, a definition is also associated.

After this creation step, we submit our Corpus to the NER process along with all the resources. A first set of annotated documents was obtained and analyzed.

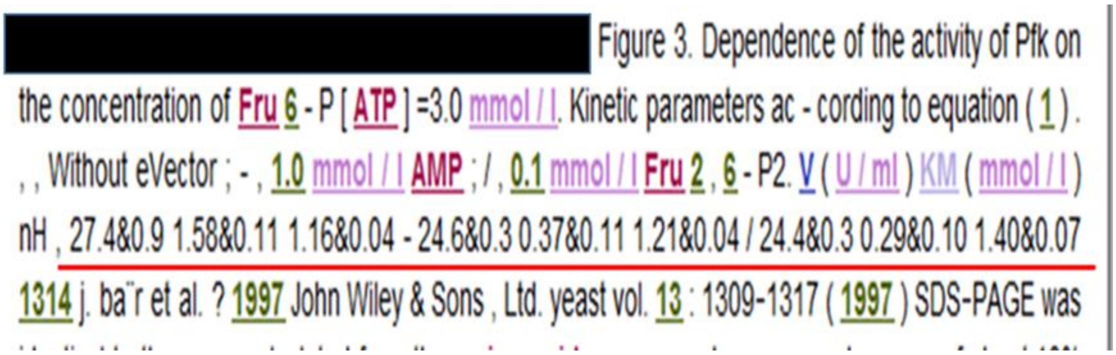

[redacted] Figure 3. Dependence of the activity of Pfk on
the concentration of Fru 6 - P [ATP] =3.0 mmol / l. Kinetic parameters ac - cording to equation (1) .
, , Without eVector ; - , 1.0 mmol / l AMP ; / , 0.1 mmol / l Fru 2 , 6 - P2. V (U / ml) KM (mmol / l)
nH , 27.4&0.9 1.58&0.11 1.16&0.04 - 24.6&0.3 0.37&0.11 1.21&0.04 / 24.4&0.3 0.29&0.10 1.40&0.07
1314 j. ba¨r et al. ? 1997 John Wiley & Sons , Ltd. yeast vol. 13 : 1309-1317 (1997) SDS-PAGE was

Figure 4.3: The result of the annotation of the text of Figure 4.4.

In Figure 4.3, we can see an example of an annotated part of a document, and when comparing to the original publication (Figure 4.4) we see that values were not annotated, as well as some metabolites. We realized that there is a conversion problem that is interfering with the annotation process. In this specific case, the values are not presented in a standard format: as decimal values, the numbers should have a comma/dot on the bottom of the number; however, it appears centered in the middle of the numbers. So, in the converted text, showed in Figure 4.3, it is not possible to understand the values.

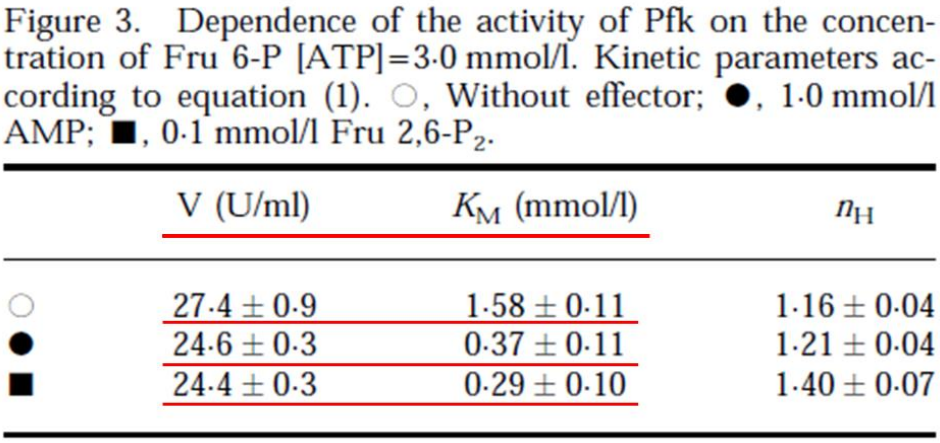


Figure 3. Dependence of the activity of Pfk on the concentration of Fru 6-P [ATP]=3·0 mmol/l. Kinetic parameters according to equation (1). ○, Without effector; ●, 1·0 mmol/l AMP; ■, 0·1 mmol/l Fru 2,6-P_2.

	V (U/ml)	K_M (mmol/l)	n_H
○	27·4 ± 0·9	1·58 ± 0·11	1·16 ± 0·04
●	24·6 ± 0·3	0·37 ± 0·11	1·21 ± 0·04
■	24·4 ± 0·3	0·29 ± 0·10	1·40 ± 0·07

Figure 4.4: Screenshot of a part of interest from a publication (PMID 9392075).

We also detected that many small words, as for example “to”, “are”, “as”, “et”, “for”, were being annotated as enzymes, which is not correct. To solve this problem we would need to parse and edit the BRENDA file that was used to create the dictionary in order to improve the annotation result, but was concluded that this task is out of the scope, at least in this step.

In order to improve the NER annotation, a manual curation of the resources was performed based on other wrong or failed identifications from the NER result, by adding missing terms to the units lookuptable, such as "mol-1" and "mL -1". If we check Table 4.4, we will find these units (terms number 3 and 8), but the terms added vary on the format. The analysis of preliminary annotation results also allowed to add terms to the kinetic parameters lookuptable, such as inhibition constant, the variation/definition for terms number 5 e 6 in Table 4.3 and adjusting or creating rules for values that were not properly identified.

These tasks were performed several times (Figure 4.5), and afterwards some statistics were made.

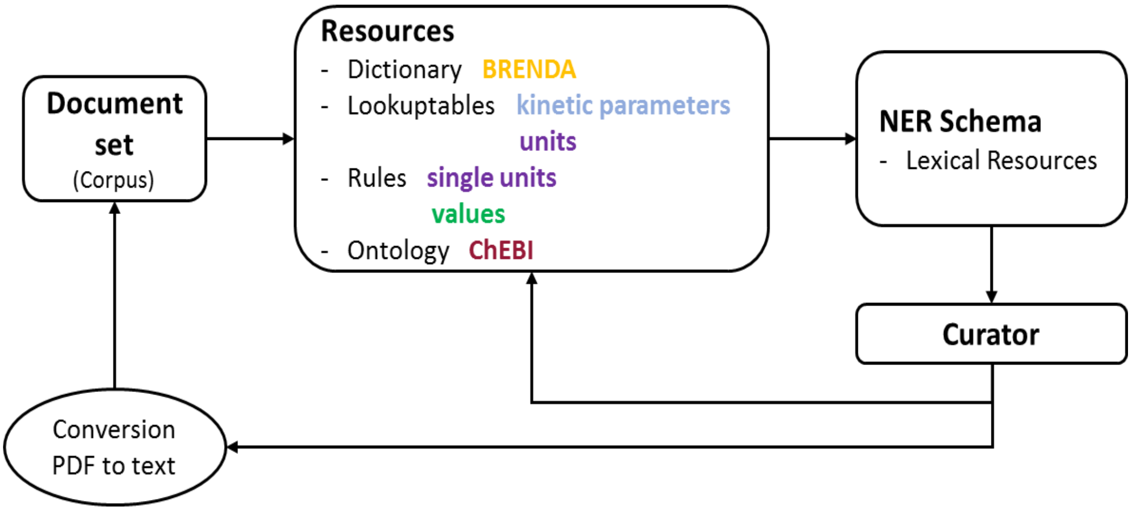


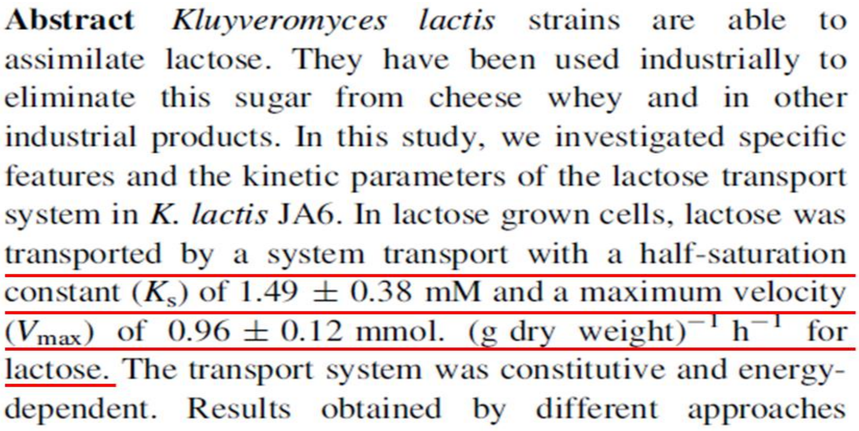
Figure 4.5: Scheme of the manual curation pipeline.

During the selection of the publications, the parts with interesting information were identified, and therefore the results of the NER annotation were compared with this manual identification and the percentage of correctly annotated entities was calculated.

Table 4.6: Results of the NER process (entities annotated vs. entities identified).

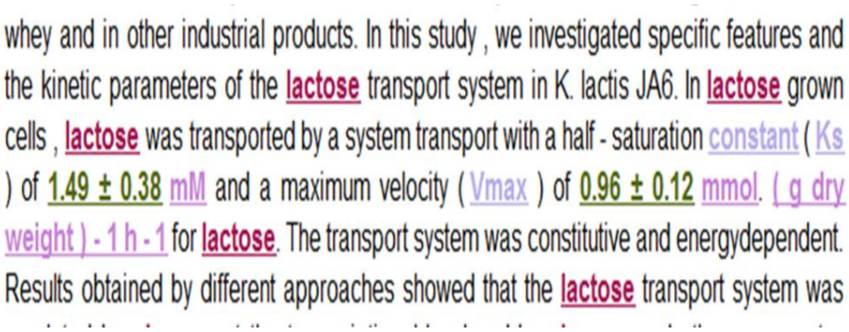
PubMed ID	Nr Relevant parts	Mets Accur	Mets Expect	Kpara Accur	Kpara Expect	Units Accur	Units Expect	Values Accur	Values Expect	Enzs Accur	Enzs Expect
6211189	6	**17**	28	**4**	16	**15**	25	**15**	23	**5**	8
9392075	8	**19**	44	**12**	21	**18**	18	**4**	36	**1**	8
12084066	3	**3**	3	**8**	10	**11**	13	**11**	12	**5**	5
12882981	3	**18**	20	**6**	7	**2**	9	**5**	27	**4**	7
15033461	5	**26**	27	**6**	10	**0**	10	**0**	10	**0**	2
15556281	2	**2**	4	**20**	26	**4**	6	**72**	72	**0**	0
15920622	0	**-**	-	**-**	-	**-**	-	**-**	-	**-**	-
24381144	0	**-**	-	**-**	-	**-**	-	**-**	-	**-**	-
24504708	10	**53**	61	**8**	15	**17**	23	**26**	28	**0**	5
24633311	0	**-**	-	**-**	-	**-**	-	**-**	-	**-**	-

The results were quite good, as globally the percentage was around 97% and, in some cases, as we can see when comparing Figure 4.6 and Figure 4.7, the annotation was 100% correct.



Abstract *Kluyveromyces lactis* strains are able to assimilate lactose. They have been used industrially to eliminate this sugar from cheese whey and in other industrial products. In this study, we investigated specific features and the kinetic parameters of the lactose transport system in *K. lactis* JA6. 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. The transport system was constitutive and energy-dependent. Results obtained by different approaches

Figure 4.6: Screenshot of a part of interest from a publication (PMID 24504708).



whey and in other industrial products. In this study , we investigated specific features and the kinetic parameters of the lactose transport system in K. lactis JA6. 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. The transport system was constitutive and energydependent. Results obtained by different approaches showed that the lactose transport system was

Figure 4.7: The result of the annotation of Figure 4.6.

The missing identifications were related to the PDF conversion to text. Some specify cases were inspected once again but unfortunately, at this point it was considered that there were no further improvements possible in the automatic identification. Most of the problems were related to numbers formatting or wrong conversion of symbols.

Taking into account our Corpus, the NER results, the analysis of the annotated documents and our objective of collecting kinetic data using extraction relations, we developed the **Kinetic RE** plug-in.

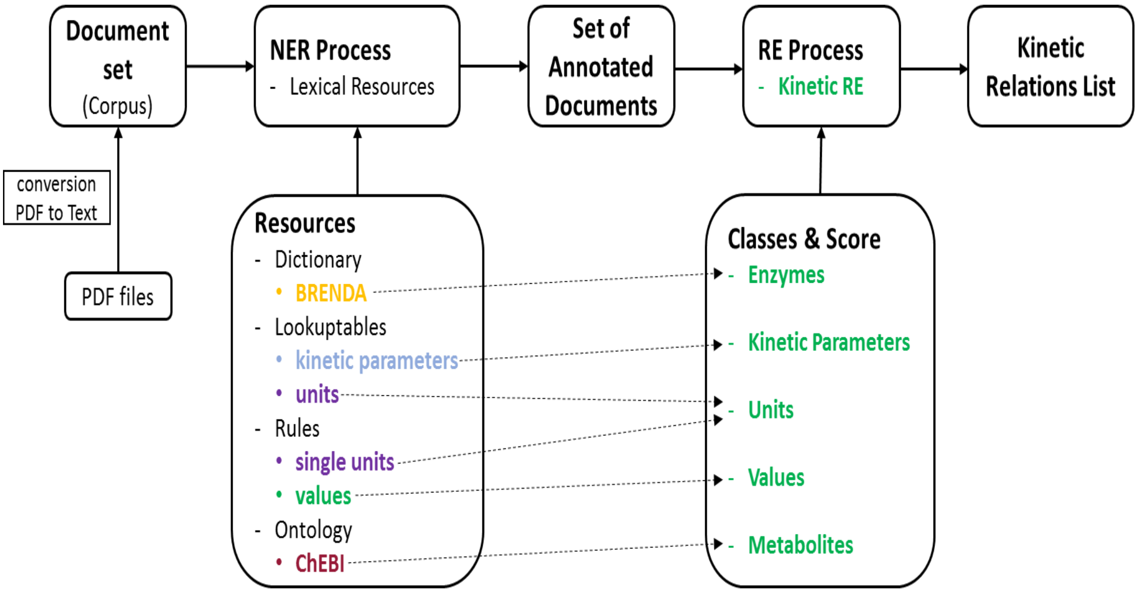


Figure 4.8: Final pipeline used in @Note2, to extract kinetic relations.

To test the plug-in, we picked randomly another set with 10 papers of *Kluyveromyces lactis* and the pipeline (Figure 4.8) was applied since the beginning.

The results obtained are showed in the following table as well as the results that were expected from the manual identification which was also previously made in this set of documents.

Table 4.7: Results of 5 papers out of 10 submitted to the pipeline.

PudMed ID	Nr Relevant parts	Nr relations Expected	Nr Relations Extracted	Comments
8065434	5	0	**0**	*13 kinetic parameters annotated **inhibitors in BRENDA
12353467	3	0	**46**	*5 kinetic parameters annotated **inhibitors in BRENDA
14626424	4	15	**130**	The Km and Vmax values of the purified enzyme for oNPG were 1.5 mM and 560 µmol min−1 mg−1 , and for lactose 20 mM and 570 µmol min−1 mg−1 , respectively. The kinetic parameters of the purified enzyme at 37 ◦C , Km and Vmax , for oNPG were 1.5 mM and 560 µmol min−1 mg−1 , and for lactose 20 mM and 570 µmol min−1 mg−1 , respectively .
17976174	2	1	**26**	*6 kinetic parameters annotated
20358191	5	7	**124**	It was confirmed that D - arabitol production was triggered when an initial lactose concentration was above 278 mmol L−1. The highest D - arabitol concentration of 79.5 mmol L−1 was achieved in the cultivation of K. lactis NBRC 1903 in a medium containing 555 mmol L−1 lactose and 40 g L−1 yeast extract.

For the Table 4.7 analysis is possible to verify that the number of extracted relationship is much higher than expected number. This difference can be explained due to the high number of simple relations (par value unit) that the algorithm detects, but that are not important because they do not contain relevant information in its neighborhood.

In cases like the third and fifth paper all the relations expected were correctly identified as is possible to see in the sentences on the column “Comments”.

In the case of the fourth paper the relation expected was not identified, but comparing the original publication with the annotation result allowed us to understand that the pair value-unit was not recognized due to a bad conversion from the PDF to text.

We conclude that relations that have not been identified/extracted, was because one or more entities were not properly annotated and that failure was result of problems in converting PDF to text, such as non-recognition of certain symbols in the case of units or bad formatting conversion in the case of digits.

5 Conclusions and Future Work

5.1 Conclusions

This work began by showing that the collection of kinetic data from databases falls well short from expected due to the lack of information and the possibility of less reliable information. This shows that the use of text mining tools can be the solution to close gaps existing in the databases and get the data directly from the different sources.

It is essential that efforts to develop text mining tools and improve the existing ones continue.

In a second phase of this work, we have developed a specific software tool that met our needs, for the existing text mining tool @Note2.

During the process of pipeline construction and algorithm design, we analyzed in detail the features and the processes contained in @Note2.

Using our case study we were able to improve the PDF conversion to text, one of the first steps in the IE process and maybe one of the most important but at same time the more challenging. We also created resources (rules, lookuptables, dictionary and ontology) to annotate kinetic parameters with NER lexical resources and developed an RE process that uses rule-based systems to extract relations from a set of annotated documents, the NER result.

In the end we propose a validated pipeline to collect kinetic data from literature using the text mining tool @Note2.

As stated previously, the PDF conversion to text is one of the more problematic tasks in text mining. Despite the improvements done during this work, we are aware that it is always possible to improve the process, as this is a continuous and time consuming process that alternates between changes, testing and validation. The same happens with the resources used, that can always be altered in order to improve the identification of entities. For example, this can be achieved by adding terms to the units and kinetic parameters lookuptables, or fitting the rules for a better detection of values.

5.2 Future Work

Based on the conclusions, we intend to improve the different resources, especially the BRENDA dictionary, used in NER and consequently in RE, as well as the PDF conversion to text, particularly in the case of values/symbols and tables.

We still intend to analyze how much the case sensitiveness influences the annotation. It would also be important to take better advantage of the use of "stop words", primarily to exclude words that are wrongly described in the ChEBI ontology and which cannot be edited.

We would like to allow the user to change the default score defined to each entity class, as well as, the value of the minimum distance allowed between the value and unit, when the algorithm is looking for the pairs in the sentence.

Finally, it would be important to present a workflow with both NER lexical resources and Kinetic RE, in order to minimize user errors and optimize the whole process presented in this work.

References

1. Area, M. Concursos de Projectos de I&D Proposals for R&D Projects. *miguelprudencio.com* 1–19 (2012). at <http://www.miguelprudencio.com/ptdc-bia-bcm-71920-2006.pdf>

2. Lourenço, A. *et al.* @Note: a workbench for biomedical text mining. *J. Biomed. Inform.* **42,** 710–20 (2009).

3. Böckenhauer, H.-J. & Bongartz, D. *Algorithmic aspects of bioinformatics*. (Springer, 2007).

4. Kitano, H. Systems biology: a brief overview. *Science* **295,** 1662–4 (2002).

5. Edwards, J. S., Covert, M. & Palsson, B. Metabolic modelling of microbes: the flux-balance approach. *Environ. Microbiol.* **4,** 133–140 (2002).

6. Lourenço, A., Carneiro, S., Rocha, M., Ferreira, E. C. & Rocha, I. Challenges in integrating Escherichia coli molecular biology data. *Brief. Bioinform.* **12,** 91–103 (2011).

7. Thiele, I. & Palsson, B. Ø. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat. Protoc.* **5,** 93–121 (2010).

8. Gianchandani, E. P., Chavali, A. K. & Papin, J. A. The application of flux balance analysis in systems biology. *Wiley Interdiscip. Rev. Syst. Biol. Med.* **2,** 372–82

9. Durot, M., Bourguignon, P.-Y. & Schachter, V. Genome-scale models of bacterial metabolism: reconstruction and applications. *FEMS Microbiol. Rev.* **33,** 164–90 (2009).

10. Raman, K. & Chandra, N. Flux balance analysis of biological systems: applications and challenges. *Brief. Bioinform.* **10,** 435–49 (2009).

11. Cornish-Bowden, A. & Bowden, A. C. *Principles of enzyme kinetics*. (Butterworths London, 1976).

12. Laidler, K. J. A brief history of enzyme kinetics. *New Beer an Old Bottle Eduard Buchner Growth Biochem. Knowledge, Val. Spain Univ. Val.* 127–133 (1997).

13. Berg, J. M., Tymoczko, J. L. & Stryer, L. Biochemistry, ; W. H. (2002).

14. Nelson, D. L., Lehninger, A. L. & Cox, M. M. *Lehninger principles of biochemistry*. (Macmillan, 2008).

15. Copeland, R. A. *Enzymes: A Practical Introduction to Structure, Mechanism, and Data Analysis*. (2004). at <http://www.google.pt/books?hl=pt-PT&lr=&id=14nqceIs_ywC&pgis=1>

16. Sauro, H. M. *Enzyme Kinetics for Systems Biology*. (2012). at <http://www.google.pt/books?hl=pt-PT&lr=&id=wiWPiHeuWuwC&pgis=1>

17. Patil, K. R., Åkesson, M. & Nielsen, J. Use of genome-scale microbial models for metabolic engineering. *Curr. Opin. Biotechnol.* **15,** 64–69 (2004).

18. Edwards, J. S. & Palsson, B. O. Systems Properties of the Haemophilus influenzaeRd Metabolic Genotype. *J. Biol. Chem.* **274,** 17410–17416 (1999).

19. Dias, O., Rocha, M., Ferreira, E. C. & Rocha, I. Merlin : metabolic models reconstruction using genome-scale information. (2010). at <http://repositorium.sdum.uminho.pt/handle/1822/22314>

20. Rocha, I., Förster, J. & Nielsen, J. in *Microbial Gene Essentiality: Protocols and Bioinformatics* 409–431 (Springer, 2008).

21. Edwards, J. S. & Palsson, B. O. Robustness analysis of the Escherichia coli metabolic network. *Biotechnol. Prog.* **16,** 927–39 (2000).

22. Chassagnole, C., Noisommit-Rizzi, N., Schmid, J. W., Mauch, K. & Reuss, M. Dynamic modeling of the central carbon metabolism ofEscherichia coli. *Biotechnol. Bioeng.* **79,** 53–73 (2002).

23. Hucka, M. *et al.* The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19,** 524–531 (2003).

24. Li, C. *et al.* BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst. Biol.* **4,** 92 (2010).

25. Schomburg, I. *et al.* BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA. *Nucleic Acids Res.* **41,** D764–72 (2013).

26. Dis, G. F., Schomburg, I., Hofmann, O. & Baensch, C. Enzyme data and metabolic information : BRENDA , a resource for research in biology , biochemistry , and medicine. 3–4 (2000).

27. Barthelmes, J., Ebeling, C., Chang, A., Schomburg, I. & Schomburg, D. BRENDA, AMENDA and FRENDA: the enzyme information system in 2007. *Nucleic Acids Res.* **35,** D511–4 (2007).

28. Moss, G. P. Nomenclature Committee of the International Union of Biochemistry and Molecular Biology. at <http://www.chem.qmul.ac.uk/iubmb/enzyme/>

29. Pharkya, P., Nikolaev, E. V. & Maranas, C. D. Review of the BRENDA Database. *Metab. Eng.* **5,** 71–73 (2003).

30. Gremse, M. *et al.* The BRENDA Tissue Ontology (BTO): the first all-integrating ontology of all organisms for enzyme sources. *Nucleic Acids Res.* **39,** D507–13 (2011).

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

32. Sayers, E. W. *et al.* Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **37,** D5–15 (2009).

33. Kanehisa, M. *et al.* Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Res.* **42,** D199–205 (2014).

34. Berman, H. M. *et al.* The Protein Data Bank. **28,** 235–242 (2000).

35. Sigrist, C. J. a *et al.* New and continuing developments at PROSITE. *Nucleic Acids Res.* **41,** D344–7 (2013).

36. Consortium, T. U. Activities at the Universal Protein Resource (UniProt). *Nucleic Acids Res.* **42,** D191–8 (2014).

37. Wittig, U. *et al.* SABIO-RK : Integration and Curation of Reaction Kinetics Data. 94–103 (2006).

38. Wittig, U. *et al.* SABIO-RK--database for biochemical reaction kinetics. *Nucleic Acids Res.* **40,** D790–6 (2012).

39. Hastings, J. *et al.* The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Res.* **41,** D456–63 (2013).

40. Bolton, E. E. PubChem: Integrated Platform of Small Molecules and Biological Activities. **4,** 217–241 (2008).

41. PubMed. at <http://www.ncbi.nlm.nih.gov/pubmed>

42. Artimo, P. *et al.* ExPASy: SIB bioinformatics resource portal. *Nucleic Acids Res.* **40,** W597–603 (2012).

43. Biol, C. R. *et al.* Protein variety and functional diversity : UniProtKB / Swiss-Prot annotation in its biological context. **899,** 882–899 (2008).

44. Bairoch, A. & Apweiler, R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. **28,** 45–48 (2000).

45. Hoogland, C., Mostaguir, K., Sanchez, J.-C., Hochstrasser, D. F. & Appel, R. D. SWISS-2DPAGE, ten years later. *Proteomics* **4,** 2352–6 (2004).

46. Bairoch, A., Universitaire, C. M. & Servet, M. The ENZYME database in 2000. **28,** 304–305 (2000).

47. SWISS-MODEL. at <http://swissmodel.expasy.org>

48. Arnold, K. *et al.* http://www.expasy.org. 5

49. Caspi, R. *et al.* The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.* **40,** D742–53 (2012).

50. Cohen, A. M. & Hersh, W. R. A survey of current work in biomedical text mining. *Brief. Bioinform.* **6,** 57–71 (2005).

51. Rodriguez-Esteban, R. Biomedical text mining and its applications. *PLoS Comput. Biol.* **5,** e1000597 (2009).

52. Fluck, J. & Hofmann-Apitius, M. Text mining for systems biology. *Drug Discov. Today* **19,** 140–4 (2014).

53. Eltyeb, S. & Salim, N. Chemical named entities recognition: a review on approaches and applications. *J. Cheminform.* **6,** 17 (2014).

54. Costa, H. S. O. Extração e Classificação de Relações Biologicamente Relevantes da Literatura Biomédica Hugo Samuel Oliveira Costa Extração e Classificação de Relações Biologicamente Relevantes da Literatura Biomédica. (2012).

55. Chun, H.-W. *et al.* Extraction of gene-disease relations from Medline using domain dictionaries and machine learning. in *Pacific Symposium on Biocomputing* **11,** 4–15 (2006).

56. Shatkay, H. & Craven, M. *Mining the biomedical literature*. (MIT Press, 2012).

57. Cohen, K. B. & Hunter, L. Getting started in text mining. *PLoS Comput. Biol.* **4,** e20 (2008).

58. Lourenço, A. *et al.* Combining Syntactic and Ontological Knowledge to Extract Biologically Relevant Relations from Scientific Papers. 237–240 (2009).

59. Liddy, E. D. Natural language processing. (2001).

60. Ye, N. & others. *The handbook of data mining*. **24,** (Lawrence Erlbaum Associates Mahwah, NJ, 2003).

61. Glez-Peña, D. *et al.* AIBench: a rapid application development framework for translational research in biomedicine. *Comput. Methods Programs Biomed.* **98,** 191–203 (2010).

62. @Note2. at <http://anote-project.org/>

Annexes

"1.1.1.1 alcohol dehydrogenase Kluyveromyces lactis "

"1.1.1.2 alcohol dehydrogenase (NADP+) Kluyveromyces lactis "

"1.1.1.49 glucose-6-phosphate dehydrogenase Kluyveromyces lactis "

"1.1.1.184 carbonyl reductase (NADPH) Kluyveromyces lactis "

"1.1.2.4 D-lactate dehydrogenase (cytochrome) Kluyveromyces lactis "

"1.2.1.13 glyceraldehyde-3-phosphate dehydrogenase (NADP+) (phosphorylating) Kluyveromyces lactis "

"1.2.1.31 L-aminoadipate-semialdehyde dehydrogenase Kluyveromyces lactis "

"1.3.1.31 2-enoate reductase Kluyveromyces lactis "

"1.3.3.3 coproporphyrinogen oxidase Kluyveromyces lactis "

"1.4.1.15 lysine dehydrogenase no activity in Kluyveromyces lactis "

"1.4.1.15 lysine dehydrogenase no activity in Kluyveromyces lactis IFO 1090 "

"1.6.2.4 NADPH-hemoprotein reductaseKluyveromyces lactis "

"1.8.1.7 glutathione-disulfide reductase Kluyveromyces lactis "

"1.8.1.7 glutathione-disulfide reductase Kluyveromyces lactis NRRL-Y1140 "

"1.14.19.4 DELTA8-fatty-acid desaturase Kluyveromyces lactis "

"1.14.19.6 DELTA12-fatty-acid desaturase Kluyveromyces lactis "

"1.14.19.6 DELTA12-fatty-acid desaturase Kluyveromyces lactis NK1 "

"2.1.1.41 sterol 24-C-methyltransferase Kluyveromyces lactis "

"2.1.1.183 18S rRNA (adenine1779-N6/adenine1780-N6)-dimethyltransferase Kluyveromyces lactis "

"2.3.1.84 alcohol O-acetyltransferase Kluyveromyces lactis "

"2.4.1.138 mannotetraose 2-alpha-N-acetylglucosaminyltransferase Kluyveromyces lactis "

"2.4.1.138 mannotetraose 2-alpha-N-acetylglucosaminyltransferase Kluyveromyces lactis KL8 "

"2.4.1.232 initiation-specific alpha-1,6-mannosyltransferase Kluyveromyces lactis "

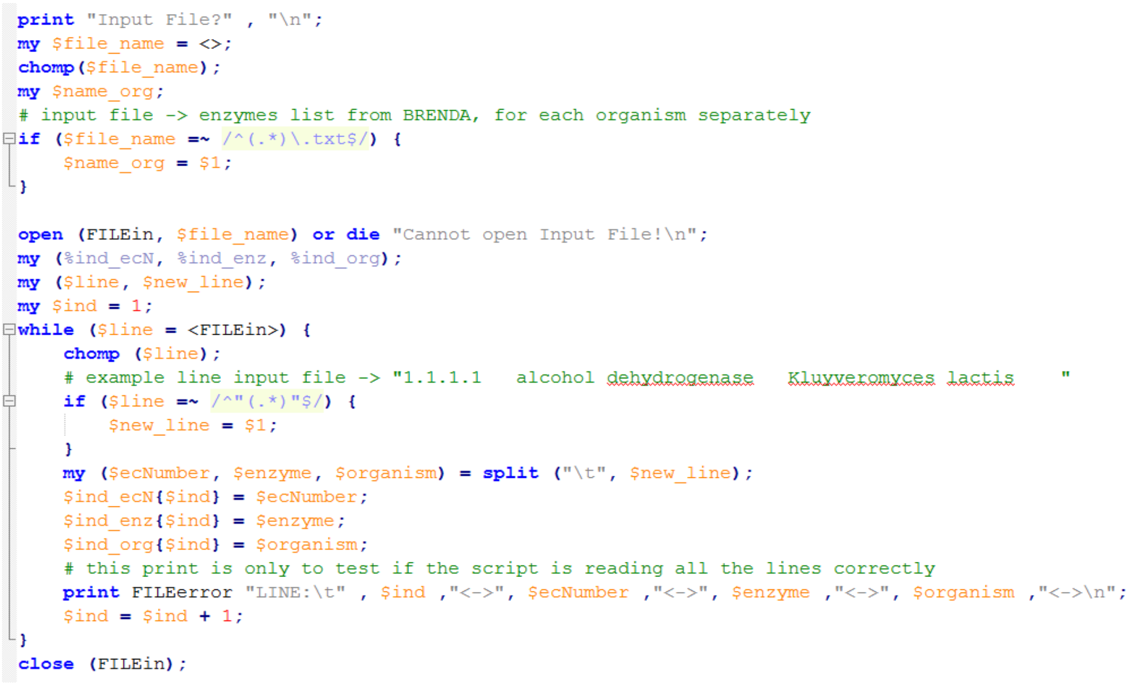
"2.7.1.1 hexokinase Kluyveromyces lactis "

"2.7.1.11 6-phosphofructokinase Kluyveromyces lactis "

"2.7.1.69 protein-Npi-phosphohistidine-sugar phosphotransferase Kluyveromyces lactis "

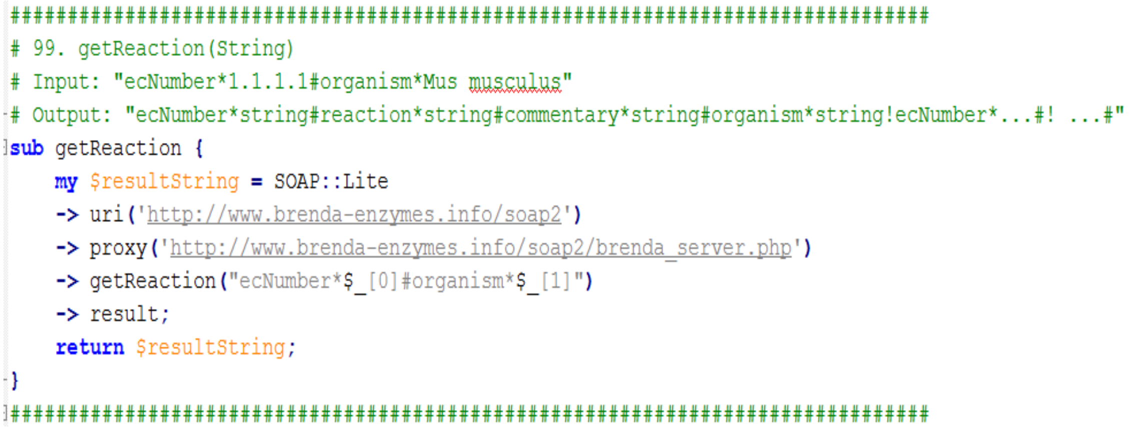
```
"2.7.7.13	mannose-1-phosphate guanylyltransferase	Kluyveromyces lactis	"
"2.7.7.50	mRNA guanylyltransferase	Kluyveromyces lactis	"
"2.7.8.5	CDP-diacylglycerol-glycerol-3-phosphate	3-phosphatidyltransferase	Kluyveromyces lactis	"
"3.1.1.5	lysophospholipase	Kluyveromyces lactis	"
"3.1.3.8	3-phytase	Kluyveromyces lactis	"
"3.1.3.26	4-phytase	Kluyveromyces lactis	"
"3.1.14.1	yeast ribonuclease	Kluyveromyces lactis	"
"3.1.26.3	ribonuclease III Kluyveromyces lactis	"
"3.2.1.23	beta-galactosidase	Kluyveromyces lactis	"
"3.2.1.23	beta-galactosidase	Kluyveromyces lactis MW 190-9B	"
"3.2.1.52	beta-N-acetylhexosaminidase	Kluyveromyces lactis	"
"3.4.16.6	carboxypeptidase D	Kluyveromyces lactis	"
"3.6.1.42	guanosine-diphosphatase	Kluyveromyces lactis	"
"3.6.3.6	H+-exporting ATPase	Kluyveromyces lactis	"
"4.1.1.1	Pyruvate decarboxylaseKluyveromyces lactis	"
"4.1.1.1	Pyruvate decarboxylaseKluyveromyces lactis CBS 2359	"
"4.1.1.1	Pyruvate decarboxylaseKluyveromyces lactis JA-6	"
"4.1.3.1	isocitrate lyase	Kluyveromyces lactis	"
"6.3.4.6	Urea carboxylase	Kluyveromyces lactis	"
```

Figure A 1: List of the 45 enzymes used as input file for the Perl script.



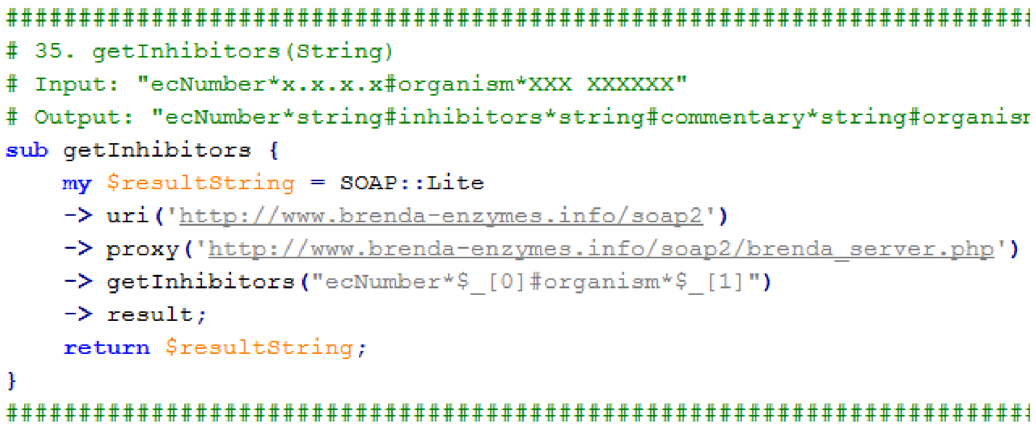
```perl
print "Input File?" , "\n";
my $file_name = <>;
chomp($file_name);
my $name_org;
# input file -> enzymes list from BRENDA, for each organism separately
if ($file_name =~ /^(.*)\.txt$/) {
    $name_org = $1;
}

open (FILEin, $file_name) or die "Cannot open Input File!\n";
my (%ind_ecN, %ind_enz, %ind_org);
my ($line, $new_line);
my $ind = 1;
while ($line = <FILEin>) {
    chomp ($line);
    # example line input file -> "1.1.1.1   alcohol dehydrogenase   Kluyveromyces lactis    "
    if ($line =~ /^"(.*)"$/) {
        $new_line = $1;
    }
    my ($ecNumber, $enzyme, $organism) = split ("\t", $new_line);
    $ind_ecN{$ind} = $ecNumber;
    $ind_enz{$ind} = $enzyme;
    $ind_org{$ind} = $organism;
    # this print is only to test if the script is reading all the lines correctly
    print FILEerror "LINE:\t" , $ind ,"<->", $ecNumber ,"<->", $enzyme ,"<->", $organism ,"<->\n";
    $ind = $ind + 1;
}
close (FILEin);
```

Figure A 2: First part of the script, in which it reads the input file and saves the information.



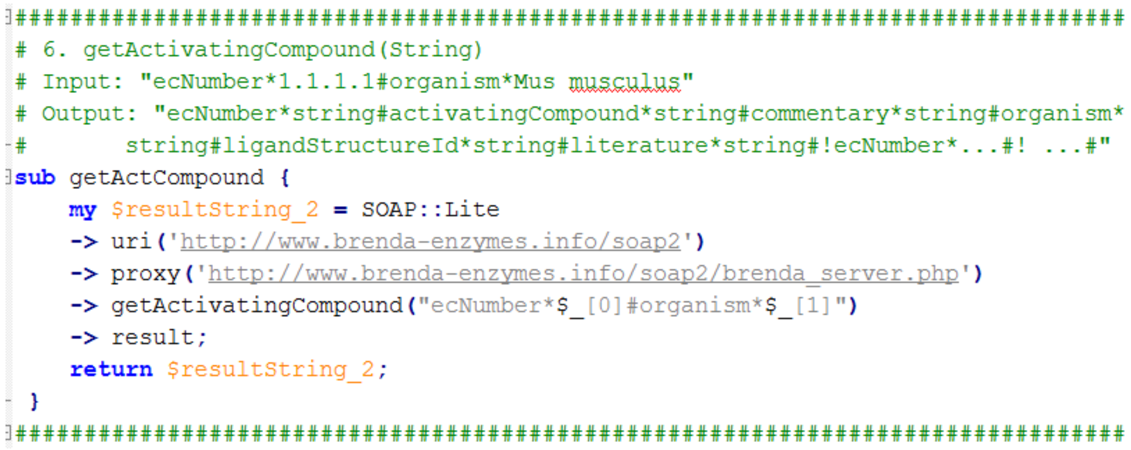
```perl
###############################################################################
# 99. getReaction(String)
# Input: "ecNumber*1.1.1.1#organism*Mus musculus"
# Output: "ecNumber*string#reaction*string#commentary*string#organism*string!ecNumber*...#! ...#"
sub getReaction {
    my $resultString = SOAP::Lite
    -> uri('http://www.brenda-enzymes.info/soap2')
    -> proxy('http://www.brenda-enzymes.info/soap2/brenda_server.php')
    -> getReaction("ecNumber*$_[0]#organism*$_[1]")
    -> result;
    return $resultString;
}
###############################################################################
```

Figure A 3: Function “getReaction” used in the Perl script.



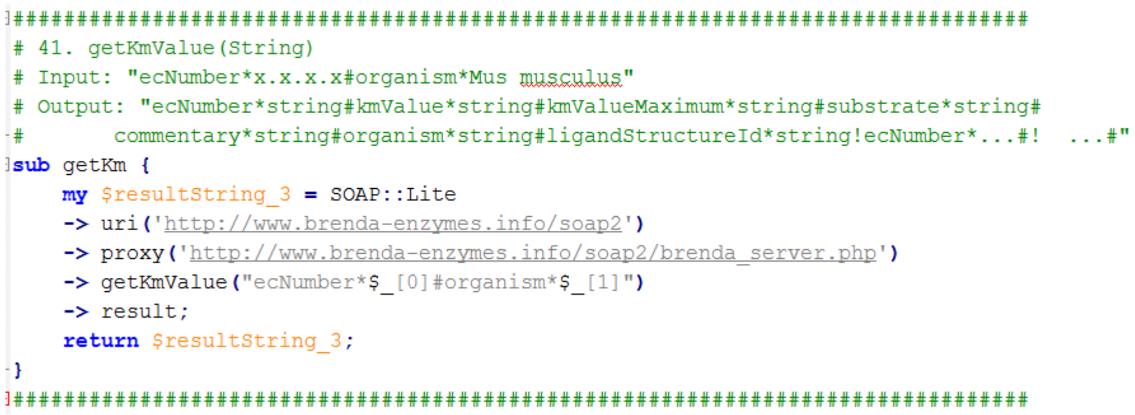
```
#######################################################################
# 35. getInhibitors(String)
# Input: "ecNumber*x.x.x.x#organism*XXX XXXXXX"
# Output: "ecNumber*string#inhibitors*string#commentary*string#organis
sub getInhibitors {
    my $resultString = SOAP::Lite
    -> uri('http://www.brenda-enzymes.info/soap2')
    -> proxy('http://www.brenda-enzymes.info/soap2/brenda_server.php')
    -> getInhibitors("ecNumber*$_[0]#organism*$_[1]")
    -> result;
    return $resultString;
}
#######################################################################
```

Figure A 4: Function “getInhibitors” used in the Perl script.



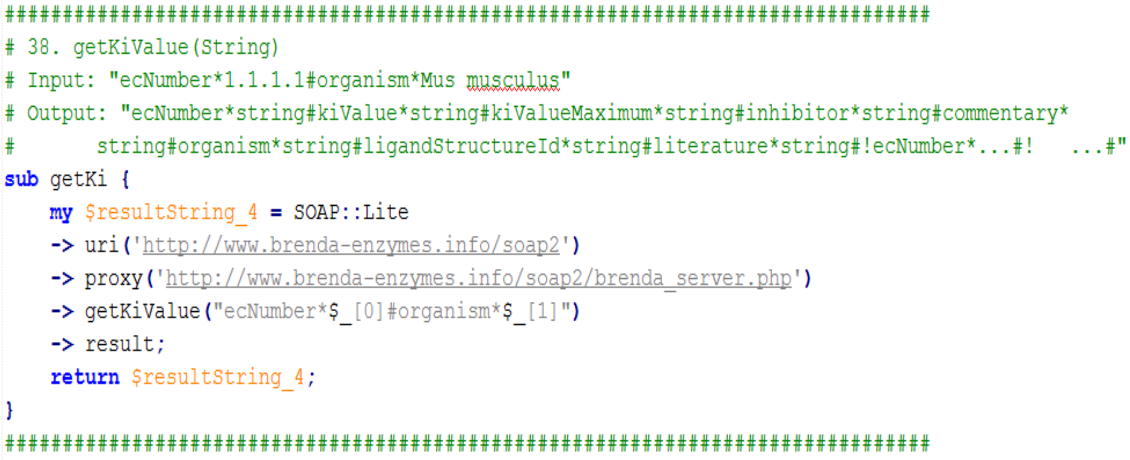
```
################################################################################
# 6. getActivatingCompound(String)
# Input: "ecNumber*1.1.1.1#organism*Mus musculus"
# Output: "ecNumber*string#activatingCompound*string#commentary*string#organism*
#        string#ligandStructureId*string#literature*string#!ecNumber*...#! ...#"
sub getActCompound {
    my $resultString_2 = SOAP::Lite
    -> uri('http://www.brenda-enzymes.info/soap2')
    -> proxy('http://www.brenda-enzymes.info/soap2/brenda_server.php')
    -> getActivatingCompound("ecNumber*$_[0]#organism*$_[1]")
    -> result;
    return $resultString_2;
 }
################################################################################
```

Figure A 5: Function “getActivatingCompound” used in the Perl script.



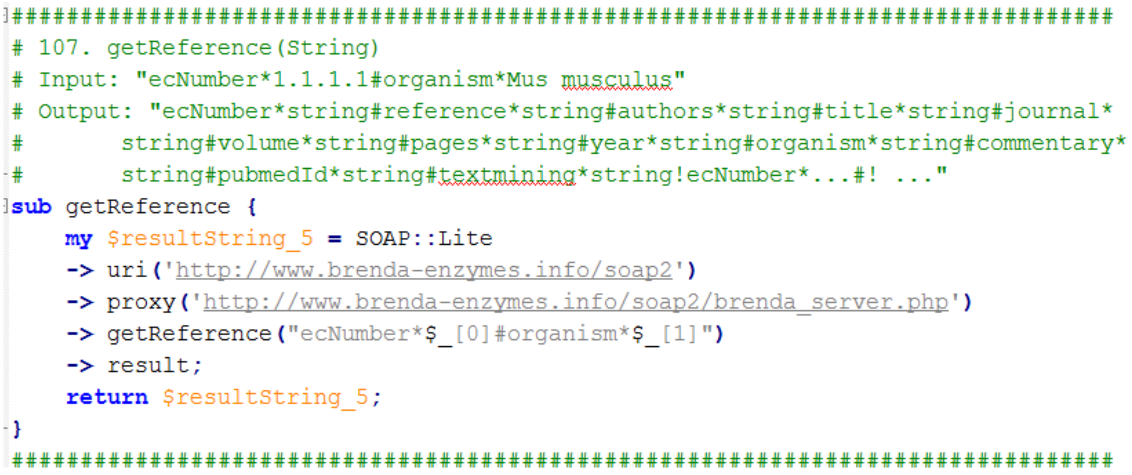
```
################################################################################
# 41. getKmValue(String)
# Input: "ecNumber*x.x.x.x#organism*Mus musculus"
# Output: "ecNumber*string#kmValue*string#kmValueMaximum*string#substrate*string#
#        commentary*string#organism*string#ligandStructureId*string!ecNumber*...#!  ...#"
sub getKm {
    my $resultString_3 = SOAP::Lite
    -> uri('http://www.brenda-enzymes.info/soap2')
    -> proxy('http://www.brenda-enzymes.info/soap2/brenda_server.php')
    -> getKmValue("ecNumber*$_[0]#organism*$_[1]")
    -> result;
    return $resultString_3;
}
################################################################################
```

Figure A 6: Function “getKmValue” used in the Perl script.



```
################################################################################
# 38. getKiValue(String)
# Input: "ecNumber*1.1.1.1#organism*Mus musculus"
# Output: "ecNumber*string#kiValue*string#kiValueMaximum*string#inhibitor*string#commentary*
#        string#organism*string#ligandStructureId*string#literature*string#!ecNumber*...#!   ...#"
sub getKi {
    my $resultString_4 = SOAP::Lite
    -> uri('http://www.brenda-enzymes.info/soap2')
    -> proxy('http://www.brenda-enzymes.info/soap2/brenda_server.php')
    -> getKiValue("ecNumber*$_[0]#organism*$_[1]")
    -> result;
    return $resultString_4;
}
################################################################################
```

Figure A 7: Function “getKiValue” used in the Perl script.



```
###################################################################################
# 107. getReference(String)
# Input: "ecNumber*1.1.1.1#organism*Mus musculus"
# Output: "ecNumber*string#reference*string#authors*string#title*string#journal*
#        string#volume*string#pages*string#year*string#organism*string#commentary*
#        string#pubmedId*string#textmining*string!ecNumber*...#! ..."
sub getReference {
    my $resultString_5 = SOAP::Lite
    -> uri('http://www.brenda-enzymes.info/soap2')
    -> proxy('http://www.brenda-enzymes.info/soap2/brenda_server.php')
    -> getReference("ecNumber*$_[0]#organism*$_[1]")
    -> result;
    return $resultString_5;
}
###################################################################################
```

Figure A 8: Function “getReference” used in the Perl script.

Table A 1: Part of Reactions output file for Kluyveromyces lactis.

ecNumber	enzyme	organism	reaction	commentary	pubmedId
3.1.14.1	yeast ribonuclease	-	exonucleolytic cleavage to nucleoside 3'-phosphates	general reaction	
3.2.1.23	beta-galactosidase	K. l.	hydrolysis of terminal non-reducing beta-D-galactose residues in beta-D-galactosides	the conserved residues E482, M522, Y523 and E551 are important in catalysis	17176477
3.2.1.23	beta-galactosidase	K. l. MW 190-9B	hydrolysis of terminal non-reducing beta-D-galactose residues in beta-D-galactosides	the conserved residues E482, M522, Y523 and E551 are important in catalysis	17176477
3.2.1.52	beta-N-acetylhexosaminidase	-	hydrolysis of terminal non-reducing N-acetyl-D-hexosamine residues in N-acetyl-beta-D-hexosaminides	general reaction	
3.4.16.6	carboxypeptidase D	-	preferential release of a C-terminal arginine or lysine residue	general reaction	

Table A 2: Part of Inhibitors output file for Kluyveromyces lactis.

ecNumber	enzyme	organism	inhibitors	commentary	pubmedId	Ki value	Ki value max
3.2.1.23	beta-galactosidase	K.l. MW 190-9B	Ba2+	1 mM, 56% inhibition of activity with o-nitrophenyl beta-D-galactopyranoside			
	beta-galactosidase	K.l. MW 190-9B	D-galactose	competitive inhibition			
	beta-galactosidase	K.l. MW 190-9B	D-glucose	noncompetitive inhibition			
3.2.1.23	beta-galactosidase	K.l. MW 190-9B	D-galactose	at 25ºC and pH 6.5, using 2-nitrophenyl-beta-D-galactopyranoside as substrate		45	
3.6.3.6	H+-exporting ATPase	K.l.	ADP	5 mM, 60% inhibition, Kd value 0.8 mM	17439159		
4.1.1.1	Pyruvate decarboxylase	K.l.	Pyruvamide	mixed type inhibitor	12084066		

Table A 3: Part of Km's output file for Kluyveromyces lactis.

ecNumber	enzyme	organism	substrate	commentary	pubmedId	Km value	Km vakue max
1.1.1.2	alcohol dehydrogenase (NADP+)	K.l.	NADPH	pH 8.0, 25ºC, mutant A274F; pH 8.0, 25 ºC, mutant G225A	15556281	0.37	
	alcohol dehydrogenase (NADP+)	K.l.	acetaldehyde	pH 8.0, 25ºC, mutant A274F	15556281	1.58	
1.2.1.13	glyceraldehyde-3-phosphate dehydrogenase (NADP+) (phosphorylating)	K.l.	D-glyceraldehyde 3-phosphate	pH 9.2, 30ºC, reaction with NAD+ or NADP+	12427047	0.75	
2.4.1.138	mannotetraose 2-alpha-N-acetylglucosaminyltransferase	K.l.	mannotetraose acceptor	membrane-bound enzyme	6211189	13	
2.7.1.1	hexokinase	K.l.	D-glucose	pH 7.4, 25ºC, recombinant enzyme at 0.001 mg/ml	12882981	0.196	
3.2.1.23	beta-galactosidase	K.l.	2-nitrophenyl-beta-D-galactopyranoside	pH 6.5, 40ºC, wild-type enzyme	17176477	1.5	

Table A 4: Part of Activator Compound output file for Kluyveromyces lactis.

ecNumber	enzyme	organism	Activating Compound	commentary	pubmedId
1.3.3.3	coproporphyrinogen oxidase	K.l.	oxygen		15920622
2.1.1.41	sterol 24-C-methyltransferase	K.l.	amphotericine B	activates at concentrations below 5.4 nM, optimal at 2.1 nM, wild-type and mutant	8065434
	sterol 24-C-methyltransferase	K.l.	candicidin	stimulates, best at low concentration	8065434
	sterol 24-C-methyltransferase	K.l.	Filipin	slightly activating at high concentration	8065434
	sterol 24-C-methyltransferase	K.l.	Nystatin	slightly activating at high concentration	8065434
	sterol 24-C-methyltransferase	K.l.	pimaricin	strong activation, activation level decreases with increasing concentration of pimaricin	8065434

Table A 5: Number of enzymes regarding each field considered

Organism	Number of enzymes with Reactions (organism specific)	Number of enzymes with				Total number of enzymes (listed BRENDA)
		Ki Values	Inhibitors	Km Values	Activators	
Kluyveromyces lactis	6	2	8	7	6	45
Helicobacter pylori	20	16	63	52	13	197
Enterococcus faecalis	15	15	53	37	17	163
Streptococcus pneumoniae	15	16	60	48	13	166
Saccharomyces cerevisiae	290	170	350	270	240	1206

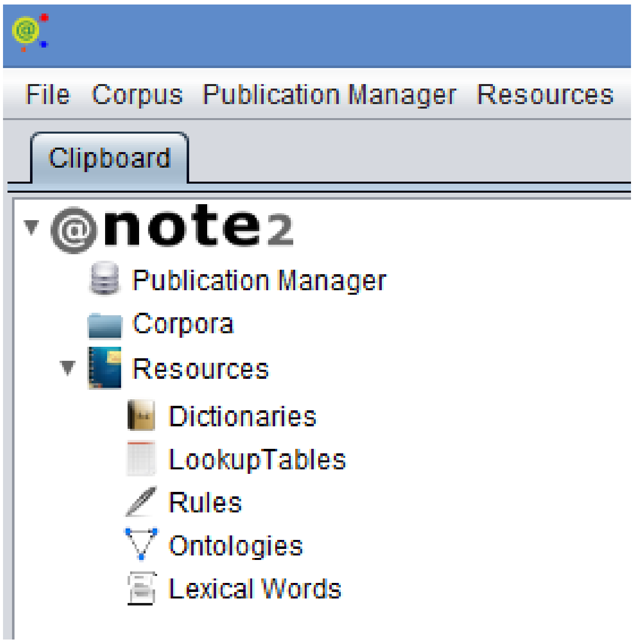


Figure A 9: Clipboard view with the list of Resources available in @Note2.