COMINC 2nd Edition Volume 1

An approach towards the reconstruction of regulatory networks

Rafael Pereira[1], **Hugo Costa**[2], **Rui Mendes**[1]

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

[2]Silico Life Lda. – Rua do Canastreiro, 15 – Braga - Portugal

rafatp@di.uminho.pt, hcosta@silicolife.com, rcm@di.uminho.pt

Abstract. *Currently, one of the main issues addressed in the bioinformatics field is understanding the structure and behaviour of complex molecular interaction networks. Since most of the information available belongs to biomedical literature, a large part of this task entails selecting the relevant articles from a large body of papers. However, due to the rapidly increasing number of scientific papers, it is quite difficult to read all the papers that have been published about this subject. In order to accomplish this, this work is focused on developing methods for retrieving information from biological databases, gathering as much information as possible; to create an integrated repository, that is able to store and load this data and also to design a pipeline to allow the reconstruction of regulatory networks through using Biomedical Text Mining techniques.*

1. Introduction

In recent years, the exploration of life sciences, has been bolstered through the advent of whole genome sequencing. This new information potentiates the reconstruction of genome-scale metabolic networks [Barrett and Palsson 2006]. But only the reconstruction of metabolic networks is not sufficient to understand the principles about how organisms work.

After this step it is necessary to discover how a genetic machinery operates inside an organism, which includes the Transcriptional Regulatory Networks (TRNs). A TRN can be considered as a model of the biological process that occurs within the cells and provides links between genes and their products.

In this work, we introduce an integrative approach for building TRNs by retrieving relevant information concerning the target organism from both databases and literature and applying text mining techniques, provided by the @Note2 framework, in order to extract biological terms and using them to create a dictionary of names and synonyms for genes, proteins and transcription factors. Thus, one is able to gather all necessary information for building these networks.

2. Background

In the 1960's, genetic and biochemical experiments demonstrated the presence of regulatory sequences in the proximity of genes and the existence of proteins that are able to bind those elements and to control the activity of genes by either activation or repression of transcription [Schlitt and Brazma 2007].

According to *Leon et al* [Ben-Tabou de Leon and Davidson 2007], the regulation is composed by two complementary components; One component is the regulatory genes, e.g. transcription factors and signalling molecules.

Transcription factors bind to specific sequences in the DNA and activate or inhibit the transcription of a gene. Signalling molecules carry out the communication between cells and initiate the activation of certain transcription factors in the cells that receive the signal.

The complementary part of these components is the regulatory genome. Every gene contains regulatory sequences that control when and where it is expressed. The regulatory sequences are arranged in units that are termed *cis*-regulatory modules, that contains a cluster of different transcription factors binding sites.

The most common way to represent a TRN is the use of directed graphs, where the nodes represent the regulators (i.e., transcription factors) and targets, and the edges represent the regulatory interactions. Basically this representation can be divided into three levels. The first one includes the set of transcription factors, downstream target genes and the binding sites in the DNA (Figure 1*a*). At the second level, these basic units are arranged into common patterns of interconnections called network motifs (Figure 1*b*). At the third level, motifs are grouped into semi-independent transcriptional units named modules (Figure 1*c*)[Babu et al. 2004]. Thus at the last level, the regulatory network is composed of interconnecting interactions between the modules that comprise the entire network (Figure 1*d*)[Carrera et al. 2009].

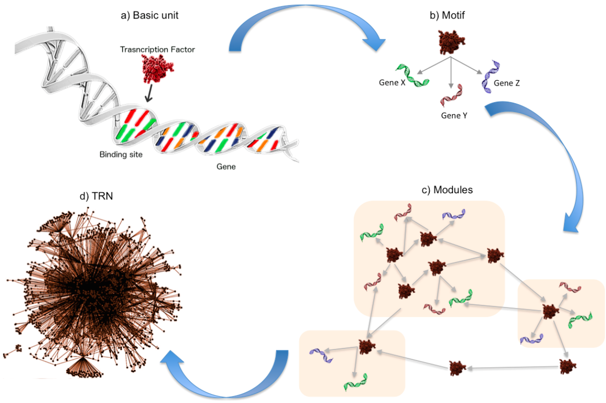

Figure 1. (a) Comprises the transcription factor, its target gene with DNA recognition site and the regulatory interaction between them. (b) The basic units are organized into networks motifs. (c) Network motifs are interconnected to form semi-independent modules. (d) The whole set of interactions that represent a transcription regulatory network

Nowadays, several approaches to model gene regulatory networks can be found, ranging from linear models [van Someren et al. 2000], Bayesian networks [Friedman et al. 2000] to Neural networks [Weaver 1999]. But, recent studies reinforce the idea that several biological processes may be modeled through the Boolean formalism.

These concepts are very important to start the reconstruction of regulatory networks that will allow researchers to understand how bacterium such as *E.coli* and *H.pylori*, can adapt to almost all environmental conditions and how they control the response to environmental changes.

3. Extending the @Note Framework for Building Transcriptional Regulatory Networks

Over the last few years, the BIOSYSTEMS[1] research group at the University of Minho and the SilicoLife[2] Company have worked together in the Biomedical Text Mining field. In this period a software platform called @Note was developed.

The @Note framework aims to help researchers establish Text Mining workflows, to facilitate the curation process and literature annotation and also to use developed models for automating tasks like text annotation and document retrieval. It is an open source project and allows developers to extend it by adding new capabilities.

Despite the many functionalities implemented therein, it was not possible to build regulatory networks using this tool. This section covers the work performed in order to extend the @Note framework for building TRNs, namely how the KREN [Pereira and Mendes 2014] repository is used for the information retrieval process, building the dictionary that will be used for recognizing the biological entities, the creation of corpora, the process of Named Entity Recognition and finally the extraction of relationships based on regulatory events.

Figure 2 depicts the workflow designed in this section for building TRNs that starts by using the KREN as the data source and the @Note to implement the Text Mining pipeline.

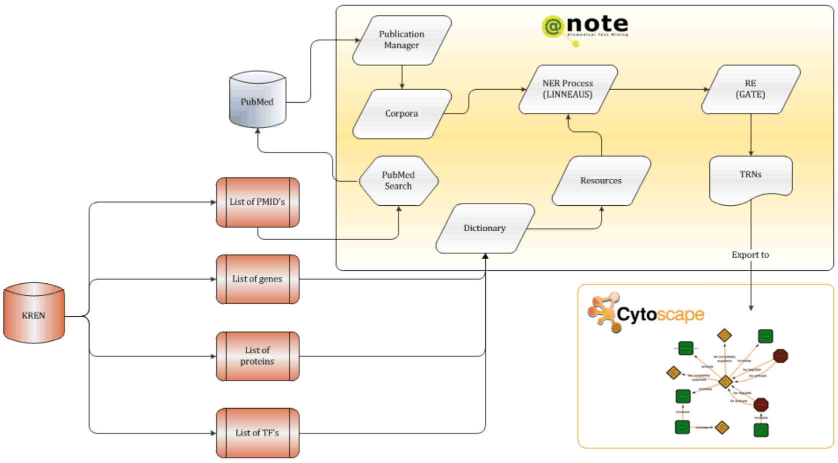

Figure 2. Workflow developed for building TRNs

3.1. Retrieving Relevant Information from PubMed

The aim of this task is to provide an efficient information retrieval process which is deemed relevant for a specific query. In this work, the query is related to the genes annotated for a given organism stored on the KREN repository. This section describes the process of retrieving a large amount of scientific papers related to the organisms studied and storing them on the @Note database.

[1] http://www.ceb.uminho.pt/biosystems

[2] http://www.silicolife.com/pt/

For this propose, the PubMed database was chosen since it is one of the most important sources of available information in the field of the Life Sciences. It provides information concerning several fields in the literature (e.g., titles, authors and abstracts) through the use of Web Services.

The Web Service implemented by PubMed provides a stable interface for information access through using a fixed URL[3] (Uniform Resource Locator) syntax that is able to interpret a set of input parameters into the values necessary for searching and retrieving the requested information [Sayers 2008]. It is implemented using the REpresentational State Transfer (REST) defined by Fielding [Fielding and Taylor 2000]. It is a type of architecture based on the client-server paradigm, commonly used by Web Services, that provides a uniform interface and makes this communication be as generic as possible [Shi 2006].

This section describes a method that was developed in order to perform a search for all scientific papers associated to each gene in the target organism through their PubMed identifiers. This method is able to search for all genes in the KREN repository and get the PubMed identifiers associated to these genes. Then, for each PubMed key, a query is performed in the @Note database in order to verify if this publication is already stored; otherwise this paper is downloaded from the source and stored into the @Note database.

The organisms chosen to perform this task were *E. coli K-12* and *Bacillus subtilis 168*. All the corresponding papers are then stored in a corpus on the @Note database, indexed by gene identifier (*locus tag*), thereafter creating the corpora. 33702 papers were retrieved concerning the *E.coli* bacteria and 6715 for the *B. subtilis*.

Due to the large amount of papers published about these organisms, only the abstracts of the papers were retrieved from the PubMed database. This decision was taken due to the time-consuming nature of the Text Mining tasks if they were to be applied to the full documents.

3.2. Applying the Named Entity Recognition (NER) Process

Before applying the NER process, a set of preprocessing steps may be used to facilitate this task, such as removing stop-words or using a part of speech tagging mechanism (Pos-tagging) in order to label the words according to their grammatical features, e.g. verbs, nouns and adjectives, in order to facilitate the process thus avoiding labeling possible biological entity as verbs.

For the NER processing, the @Note framework implements a short version of the LINNAEUS [Gerner et al. 2010] algorithm, that was developed mainly to recognize organism names; nowadays it has evolved to an algorithm for general search (e.g., genes and proteins). Listing 1 represents a pseudocode for the dictionary matcher method implemented on LINNAEUS.

The algorithm starts by creating a sorted list containing all the terms that were retrieved from the dictionary. The next step loops through each document in the corpus (cf. lines 3 through 17). Inside this loop, the document is broken into a list of tokens and their positions.

[3] https://www.w3.org/TR/url-1/

While there are still tokens to process, the dictionary is searched for the current term increasing it by one token every time until either the term is found or the end of the dictionary is reached.

The loop from lines 10 through 20 handles the search of terms in the dictionary. Initially, the search for the term starts at the beginning of the dictionary but, if the beginning of a term is not found, the search continues with a bigger term (i.e., with more tokens) from the position where the current term would be if it was found in the dictionary. This happens because the `binarySearch` function either returns a positive integer with the position where the term was found or a negative value with the position where the term would have been found if it was in the dictionary.

Listing 1. Pseudocode for representing a short version of Linnaeus algorithm

```
1  result = []
2  dic_terms = sorted(dictionary)
3  for text in documents:
4      positions, tokens = enumerate(text.split())
5      pos = 0
6      while pos < len(tokens):
7          lst = pos
8          dic_pos = 0
9          found = False
10         while not found and dic_pos < len(dic_terms):
11             txt_term = tokens[pos : lst]
12             dic = dic_terms[dic_pos:]
13             index = binarySearch(dic, txt_term)
14             if index >= 0:
15                 result.append(txt_term)
16                 found = True
17                 pos = lst + 1
18             else:
19                 dic_pos = - index - 1
20                 lst += 1
21 return result
```

An advantage of this approach is the use of a lexicon resource such as the dictionary for entity identification thus providing a convenient way of finding and identifying entities. Meanwhile, a list of PubMed identifiers for each gene was recovered from the repository, and subsequently, a search on PubMed was performed, getting 33702 papers concerning 4489 genes from *E. coli*, and 6715 papers related with 4421 genes from *B. subtilis*.

This step yielded a corpora, where each gene is associated with a list of publications. Figures 3 and 4 show the amount of biological entities that were identified for each of these organisms according to their relevance.

Using this information, the following step will be able to create a dictionary that will store the biological entities.

3.3. Using the KREN Repository for building the dictionary in the @Note Framework

Creating a complete dictionary is an essential task because it will allow the @Note framework to recognize every biological entity necessary for building TRNs. However, it is necessary to build a dictionary for each case study, because some of the genes, proteins and transcription factors are specific to each organism. The main idea in this task is to use

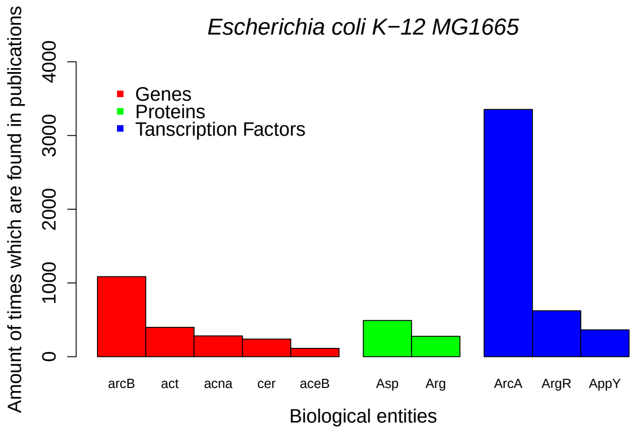

Figure 3. Number of times which entities were identified by NER process for *E. coli* organism according to their relevance

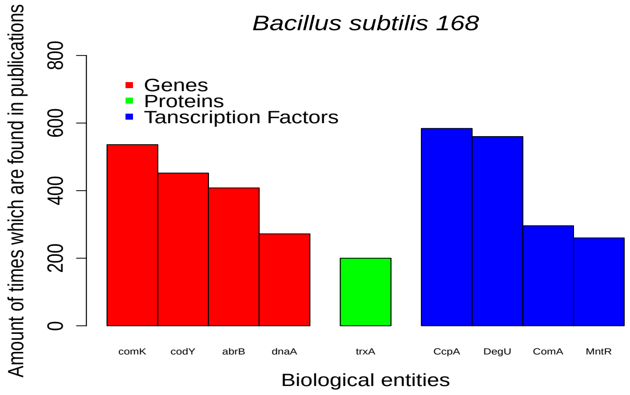

Figure 4. Graphic representation of the frequency each entity is recognized by the NER process for *B. subtilis* organism

the KREN repository to provide a way of retrieving all the needed information into a data source and export it onto @Note.

In Figure 5 it is possible to see a diagram composed by the most important information used from this data source. All information inside this data source is related with this entity and all genes are identified by an unique *B-number* that is common among the databases.

Using this repository, it is possible to retrieve all PubMed identifiers, names, synonyms and proteins associated with any given gene.

This information will be useful for the creation of a Biomedical Text Mining (BioTM) corpus and lexical resources (dictionaries) that will be needed to accomplish the goal of reconstructing regulatory networks.

In order for @Note to recognize the new biological entities, it was necessary to create three new types of resource elements (gene, protein and transcription factor). The main element is the gene, whose unique identifier serves as a key that is associated with the rest of the information.

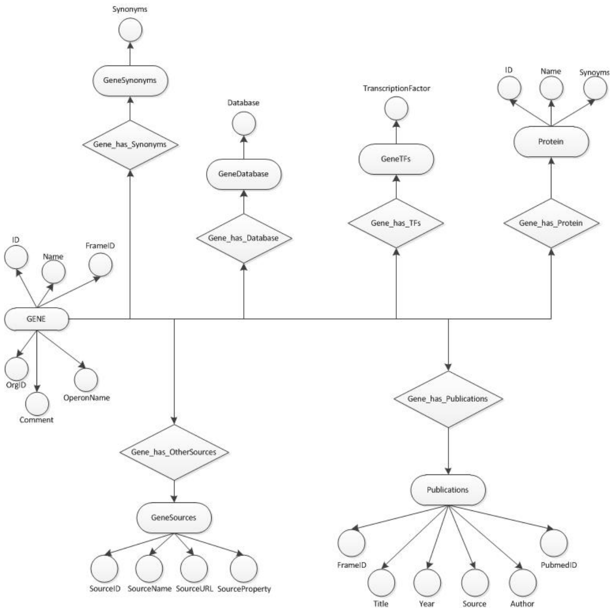

Figure 5. This diagram shows the information used for building the TRNs: synonyms, databases, proteins, transcription factors, other sources of information and publications. It is important to highlight that this information is related directly to each gene.

A method to access the data stored in the KREN repository and load in onto @Note was developed. This method performs a search for terms such as gene names, synonyms for each gene name, protein names, synonyms for each protein name, names of transcription factors and PubMed database identifiers. However, before this information can be loaded on @Note, it is necessary to find all the duplications concerning terms found among the databases that compose the KREN and exclude them.

After this task was complete, two different organisms *(Escherichia coli K-12 MG1665* and *Bacillus subtilis*) were chosen to extract information from KREN and create the @Note dictionary. The results obtained in this task can be seen in Tables 1 and 2.

Table 1. Statistics concerning the number of *E. coli* names and synonyms for genes proteins and transcription factors retrieved from KREN

Organism	Terms	Quantity
Escherichia coli K-12 MG-1665	Gene names	4586
	Gene synonyms	16747
	Protein names	6068
	Protein synonyms	6201
	Transcription Factors	181

After the dictionary creation, the next step will be the process of extracting regulatory interactions among the biological entities involved in the TRNs.

3.4. Relation Extraction Based on Regulatory Events

Automating the process of extracting relations concerning regulatory events has been a challenge in the Biomedical Text Mining area [Friedman et al. 2001]. Even though there are several attempts focusing mainly in automatic recognition, normalization, and on

Table 2. Statistics concerning the number of names and synonyms for genes proteins and transcription factors retrieved from *B. subtilis* retrieved from KREN

Organism	Terms	Quantity
Bacillus subtilis 168	Gene names	4435
	Gene synonyms	4314
	Protein names	2449
	Protein synonyms	0
	Transcription Factors	157

mapping these biological entities [Temkin and Gilder 2003], some advanced approaches must be implemented in order to be able to fully perform this task.

The main goal of this step is to use an approach to identify these interactions by using some Natural Language Processing (NLP) method [Spyns 1996] such as Shallow [Neumann and Piskorski 2002] and Deep processing [Crysmann et al. 2002] or Dependency parsing [Kubler et al. 2009].

Even tough some of these approaches for identifying these events were already implemented on @Note, it was necessary to adapt these tools in order to perform the necessary relation extraction. The starting point was to make a list of possible triggers (verbs that will be essential to determine an event, e.g.: regulation, inhibition, activation). The first step is to break the corpora from the previous step into phrases creating the syntactic and semantic layers (cf. Figure 6). The syntactic layer is responsible for categorizing the words in the sentence. The semantic layer conveys meaning by characterizing an identification with the biological entities.

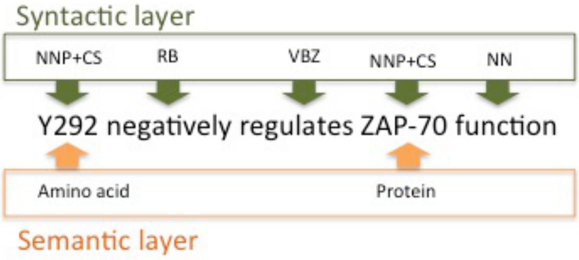

Figure 6. Illustration of the syntactic and semantic layers.

The next step, shown in Figure 7, involves the extraction and characterization of the biological relationships. It is composed of three main steps:

1. to perform a grammatical tagging based on the syntactic layer;
2. to match the syntactic layer with the morphological analysis; and
3. to extract and characterize the relationships using the verb to identify an interaction.

The relationship is delimited upstream by the previous verbal grouping (VG) or by the beginning of the phrase and downstream by the verbal grouping immediately following to it or by the ending of the phrase, cf. Figure 8.

In order to implement this pipeline, a short version of a tool for Natural Language Processing (NLP) that provides several resources to help perform the Relation Extrac-

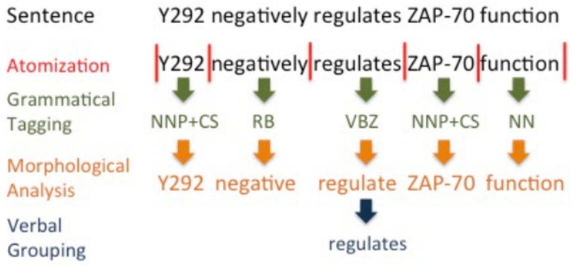
Sentence
Y292 negatively regulates ZAP-70 function
Atomization
Y292 negatively regulates ZAP-70 function
Grammatical Tagging
NNP+CS RB VBZ NNP+CS NN
Morphological Analysis
Y292 negative regulate ZAP-70 function
Verbal Grouping
regulates

Figure 7. Atomization process for extracting possible relationships between biological entities

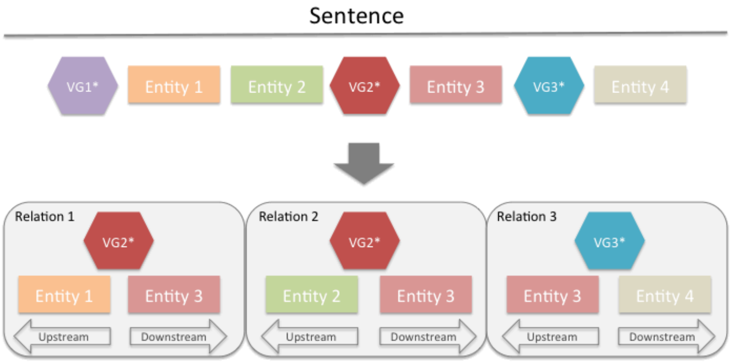

Figure 8. Approach to identify relationships in a phrase

tion (RE) processing called GATE [Cunningham 2002] was used. This tool allows the extraction of this type of relationships automatically.

In order to evaluate this task, two test cases using a large set of abstracts were performed, one using 33702 related to *E. coli* and the other using 6715 abstracts related with *B. subtilis*. The results obtained in this task can be seen in Table 3.

Organism	Amount of verbs(clues)	Number of relations
E. coli	6715	3326
B. subtilis	342	924

Table 3. Results from Relation Extraction task performed over *E. coli* and *B. subtilis*

4. Conclusion

Nowadays, Systems Biology is a field that is attracting much interest due to the interest in the process of biological simulation, whose aim is to perform a reconstruction, *in silico* and *in vivo*, of all processes that occur inside the cells, both metabolic or regulatory. In this field, Transcriptional Regulatory Networks (TRNs) are powerful tools for representing interactions between biological entities within a cell and their study helps understand the process of regulatory interactions that link the Transcription Factors (TFs) to their target genes. This context has motivated the present research work, whose main objective is to provide an approach for discovering new knowledge using information from different data sources and scientific literature in order to build regulatory models.

The present work was mainly focused in proposing a solution for integrating biological data related to Transcriptional Regulatory Networks and to extend an already developed software system (@Note) with methods which allow the creation of these networks for any type of prokaryotic organism. Moreover, it suggests an approach that can be used as a reference to improve models and use it to perform *in silico* strain optimization. Although there is a lack of tools for building these networks, the reviewed literature and the outcomes of this work clearly show that this is a promising area of study

References

Babu, M. M., Luscombe, N. M., Aravind, L., Gerstein, M., and Teichmann, S. A. (2004). Structure and evolution of transcriptional regulatory networks. *Current opinion in structural biology*, 14(3):283–291.

Barrett, C. L. and Palsson, B. O. (2006). Iterative reconstruction of transcriptional regulatory networks: an algorithmic approach. *PLoS computational biology*, 2(5):e52.

Ben-Tabou de Leon, S. and Davidson, E. H. (2007). Gene regulation: gene control network in development. *Annual review of biophysics and biomolecular structure*, 36:191.

Carrera, J., Rodrigo, G., Jaramillo, A., and Elena, S. F. (2009). Reverse-engineering the Arabidopsis thaliana transcriptional network under changing environmental conditions. *Genome biology*, 10(9):R96.

Crysmann, B., Frank, A., Kiefer, B., Müller, S., Neumann, G., Piskorski, J., Schäfer, U., Siegel, M., Uszkoreit, H., Xu, F., et al. (2002). An integrated architecture for shallow and deep processing. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 441–448. Association for Computational Linguistics.

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Fielding, R. and Taylor, R. (2000). Principled design of the modern Web architecture. *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*, 2(2):115–150.

Friedman, C., Kra, P., Yu, H., and Rzhetsky, A. (2001). GENIES : a natural-language processing system journal articles. *Bioinformatics*, 17.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, 7(3-4):601–620.

Gerner, M., Nenadic, G., and Bergman, C. M. (2010). LINNAEUS: a species name identification system for biomedical literature. *BMC bioinformatics*, 11:85.

Kubler, S., McDonald, R., Nivre, J., and Hirst, G. (2009). *Dependency Parsing*. Morgan and Claypool Publishers.

Neumann, G. and Piskorski, J. (2002). A shallow text processing core engine. *Computational Intelligence*, 18:451–476.

Pereira, R. and Mendes, R. (2014). Integrating Biological Databases in the Context of Transcriptional Regulatory Networks. *International Journal of Bioscience, Biochemistry and Bioinformatics*, 4:345–350.

Sayers, E. (2008). *Entrez Programming Utilities Help*. National Center for Biotechnology Information (US).

Schlitt, T. and Brazma, A. (2007). Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8 Suppl 6:S9.

Shi, X. (2006). Sharing service semantics using SOAP-based and REST Web services. *IT Professional*, 8(2):18–24.

Spyns, P. (1996). Natural language processing in medicine: An overview. *Methods of Information in Medicine*, 35(4-5):285–301.

Temkin, J. M. and Gilder, M. R. (2003). Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, 19(16):2046–2053.

van Someren, E. P., Wessels, L. F., and Reinders, M. J. (2000). Linear modeling of genetic networks from experimental data. *Proceedings of International Conference on Intelligent Systems for Molecular Biology*, 8:355–366.

Weaver, D. C. (1999). Modeling regulatory networks with weight matrices. In *Pacific Symposium on Biocomputing*, volume 4, pages 112–123.