Extraction of data events from the computational biology literature

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Science and Engineering

2022

Manal M. Albahlal
Department of Computer Science

Contents

Word Count: 46,105

List of figures

List of tables

Acronyms

Att-BiLSTM Attention-based Bidirectional Long Short-term Memory. 52, 106, 120, 128

BERT Bidirectional Encoder Representations from Transformers. 42, 44, 103, 112, 116, 196

BiLSTM Bidirectional Long short-term memory. 14, 39, 119

BiLSTM-CRF Bidirectional Long short-term memory with a conditional Random Field. 112, 115

BioBERT Bidirectional Encoder Representations from Transformers for Biomedical Text Mining. 14, 42, 103, 112, 116, 117, 121, 129, 132, 196

CRF Conditional Random Fields. 39, 62, 115

CSV Comma Separated Values. 60, 121, 139, 144, 198

MeSH Medical Subject Headings. 3, 32, 45, 53, 61, 72, 77, 78, 97, 109, 137, 161

MIAME Minimum Information About a Microarray Experiment. 30, 94

NER Named Entity Recognition. 42, 44, 46, 47, 66, 196

NLP Natural Language Processing. 42, 53, 100, 103

PMCID PubMed Central Identifier. 123, 128, 144, 246

RST Rhetorical Structure Theory. 143, 154, 205

Typographical conventions

This thesis uses some typographical conventions to highlight elements of the data events components, normalisation classes and discourse concepts.

Typeface	Usage
Bold-italic	This typeface is used for the data event's components names and relations. We have four components and four relations. The four components are: ***operation, data, software*** and ***database***. The four relations are: ***input data, output data, by software,*** and ***from database***. We used this presentation to help the reader to differentiate between the component that we called ***data*** and the general meaning of data.
Italic	This typeface is used for the annotated text e.g. *The data are analysed by R*. It is also used for annotated entities as well e.g. *data* and for the examples that are quoted from other articles.
Teletype font	This typeface is used for EDAM classes e.g. `Visualisation, Data` and `Operation`. It is also used for representing data event components relations e.g. `Rel<operation:Design, SW:Primer3>`.
Teletype font-bold	This typeface is used for discourse relations or functions e.g. **`Elaboration, Result`** and **`Method`**.

Abstract

With the current rate of research activities, it is widely accepted that scientists face a challenge of keeping up-to-date with new findings, even within a sub-field of a discipline. This difficulty extends to methods that have been used in the research. Understanding reported methods gives us confidence that the findings have resulted from an appropriate, rigorous and sound scientific process. However, the modern dynamic of science is also characterised with ever-changing methods, so scientists need to be able to learn about new ones and identify the common or most appropriate methods to use in a given situation.

One of the best sources of information about methods is the scientific literature. In this thesis, we developed a computational model to automatically represent the text that describes reported methods as an abstract method workflow. We focus on computational sciences, which centre on data processing. Specifically, we consider *data events* as a representation of processes and changes that happen to data. A data event contains the main components of each step in computational experiments, such as input/output data, processes and operations on data, databases where the data is stored and software and tools that are used in these processes. An abstract method workflow then models relationships between data events, ordering them in a way that represents the methodology as reported in the literature.

This thesis introduces ODNoRFlow, a text mining method that extracts and represents an abstract method workflow from a Methods section of a publication. It relies on a hybrid text mining approach (ODNoR) that combines machine learning and a rule-based method to recognise data event components, normalise them to existing ontologies and identify the links and relations between them. Specifically, we fine-tuned a pre-trained transformer model (BioBERT) to extract mentions of data and operations, and used an existing named entity recognition system (bioNerDS) to extract software and database mentions. Mentions were normalised to the EDAM ontology. We used a combination of syntactic rules and a pre-trained attention-based BiLSTM model to identify relations and links between components, and considered whether an automated discourse analysis tool can be used to improve the outcomes.

We used the microarray analysis literature as a case study to demonstrate the feasibility of the proposed approaches. At the data event level, the approach achieved F-

scores for the identification and normalisation of components between 78% (for data) and 92% (for operations), whereas the relationship extraction F-scores were between 62% and 92.5%. At the workflow level, we manually analysed automatically reconstructed workflows from 25 papers, with the F-score between 61% and 93.5%. We also applied ODNoRFlow to a large corpus of the microarray analysis literature to identify and analyse the distribution of data events components, the differences in their usage and the associations between them.

Overall, the thesis provides a new computational framework that contributes to the automated extraction, representation and analysis of methods used in the computational biology literature.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Dedication

To my father soul for the values he taught me,
To my mother for her belief in education, and endless love and prayers,
To my husband for his support,
To my lovely children for their love and support.

Acknowledgements

I thank Allah, glorified and exalted be He, for the great favours and blessings that He has bestowed upon me. He has given me the strength to endure the tough times and keep going.

I thank my supervisors, Professor Robert Stevens and Professor Goran Nenadic, for all the support and guidance they have lent me throughout this journey. It has been such a privilege learning from them and an honour to be their student.

I cannot thank my husband, Khalid, enough. It wouldn't be possible without his support and patience. My thanks continued to my lovely daughter, Orjuwan, my lovely son, Abdullah, and my little daughter, Raghad, for their love and continued support. They experienced the good days as well as the stressful moments.

Words cannot express how grateful I am for my family in Saudi Arabia. My mother, Muneerah, brothers, and sisters, Majed, Amal, Bader, Rehab, Jaruallah and Abdullah.

I would like to thank the text mining and NLP group, Goerge, Ruth, Nikola, Maksim, Haifa, Ghadah, Ghadeer, Emma, Viktor, Rui, Filippos and Gavin. Special thanks go out to Geraint Duck for sharing his work and being supportive in the early days of my PhD.

I would like to acknowledge the country I love, Saudi Arabia, for the opportunity and financial support that allowed me to study abroad and further my academic career. I would like to thank the faculty and friends at King Saud University.

Chapter 1

Introduction

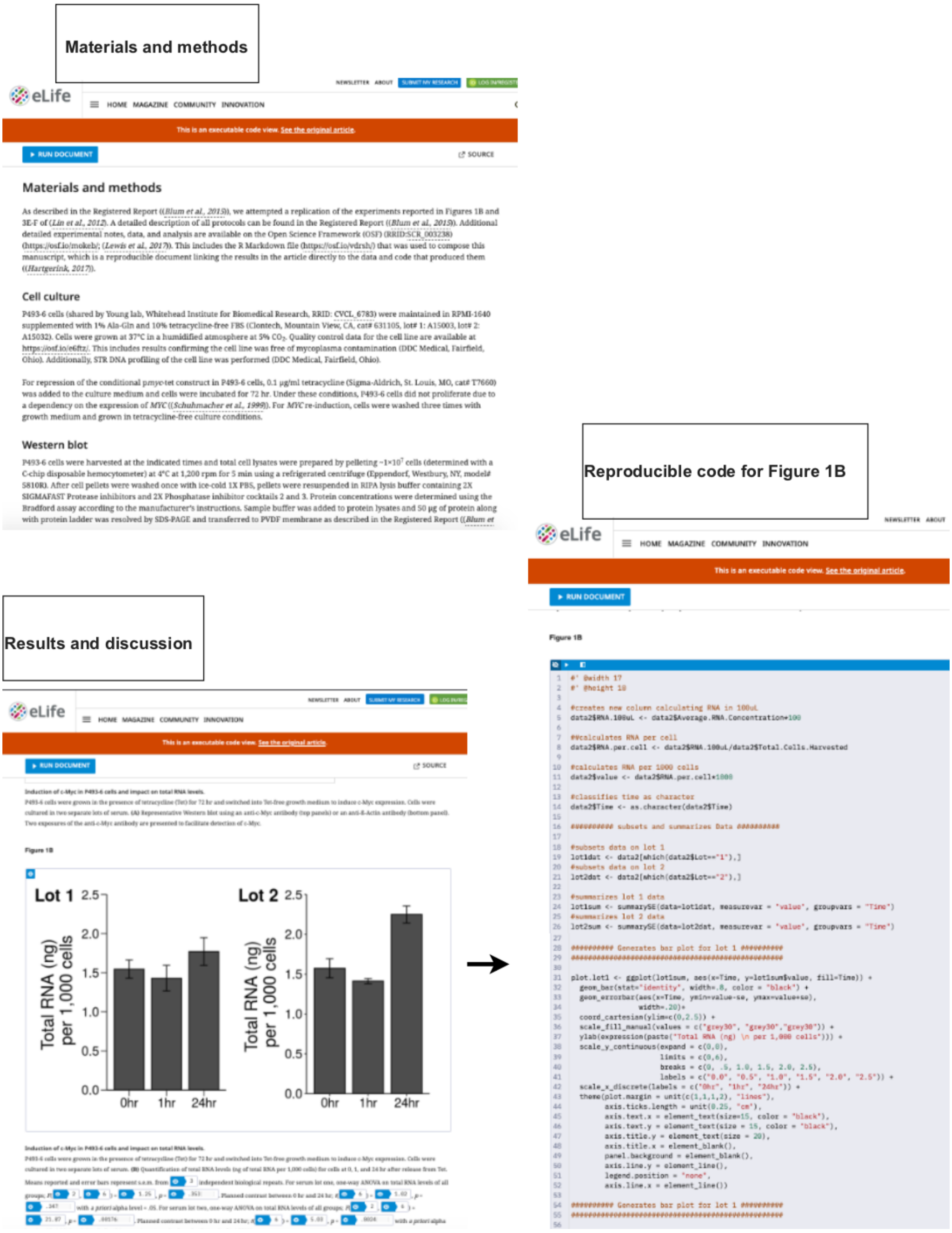
1.1 Motivation

The number of scientific publications has increased significantly over recent years [1]. This large volume of literature implies a large number of scientific methods that require a great effort to follow, understand and apply. This can be seen in particular in domains with rapid development, such as computational biology, in which experiments being reported are growing in quantity and complexity. The situation becomes even more challenging in interdisciplinary domains that burden users with a broad variety of, often complex, methods published in the literature in different domains.

Publications usually present details of used methodologies (which contain methods and data) and results. Most of research results (such as specific findings) can be presented with computational representations (e.g. lists of interacting proteins) and can be retrieved, analysed and compared with other results. Even when the results of scientific studies are not readily available, a number of text mining methods have been developed and applied to extract these from text. Indeed, the vast majority of text mining efforts so far has been about making results and findings available in a structured, computational representation. For example, there are several BioC implementations [2] to enable shared formats for annotations resulting from applying text mining tools on the biomedical literature. Several corpora [3, 4, 5] have been produced to share and make such annotation information available in standardised BioC XML or JSON formats.

Some of the more recent attempts aim at checking and monitoring the reproducibility of research (and findings) as reported in scientific papers. For example, the Reproducibility Project: Cancer Biology aimed at replicating 50 experiments from a selection of 23 high-profile papers in the field of cancer biology [6]. The replicated experiments were linked to specific figures and tables in the original papers. An example of the extracted results is shown in Figure 1.1, with associated code that could be used to reproduce the results.

Methods presented in research papers are, however, only rarely represented by or linked to executable workflows, but are rather described in prose as part of scientific publications, often as part of the Methods (or Materials and Methods) section (see Figure 1.1). This section describes the work conducted, what components are involved and why they were chosen [7]. It either cites well-known methodologies or should detail the important information related to the new proposed or combined methods [8]. Being in a free-text narrative, this means that the vast bulk of reported methods are inaccessible computationally, making it difficult to query, retrieve and identify specific methods in a large body of research. However, finding, understanding and accessing



Materials and methods

eLife
NEWSLETTER ABOUT SUBMIT MY RESEARCH LOG IN/REGISTER
HOME MAGAZINE COMMUNITY INNOVATION
This is an executable code view. See the original article.
▶ RUN DOCUMENT
SOURCE

Materials and methods

As described in the Registered Report ((*Blum et al., 2015*)), we attempted a replication of the experiments reported in Figures 1B and 3E-F of (*Lin et al., 2012*). A detailed description of all protocols can be found in the Registered Report ((*Blum et al., 2015*)). Additional detailed experimental notes, data, and analysis are available on the Open Science Framework (OSF) (RRID:SCR_003238) (https://osf.io/mokeb/; (*Lewis et al., 2017*)). This includes the R Markdown file (https://osf.io/vdrsh/) that was used to compose this manuscript, which is a reproducible document linking the results in the article directly to the data and code that produced them ((*Hartgerink, 2017*)).

Cell culture

P493-6 cells (shared by Young lab, Whitehead Institute for Biomedical Research, RRID: CVCL_6783) were maintained in RPMI-1640 supplemented with 1% Ala-Gln and 10% tetracycline-free FBS (Clontech, Mountain View, CA, cat# 631105, lot# 1: A15003, lot# 2: A15032). Cells were grown at 37°C in a humidified atmosphere at 5% CO_2. Quality control data for the cell line are available at https://osf.io/e6ftz/. This includes results confirming the cell line was free of mycoplasma contamination (DDC Medical, Fairfield, Ohio). Additionally, STR DNA profiling of the cell line was performed (DDC Medical, Fairfield, Ohio).

For repression of the conditional p*myc*-tet construct in P493-6 cells, 0.1 µg/ml tetracycline (Sigma-Aldrich, St. Louis, MO, cat# T7660) was added to the culture medium and cells were incubated for 72 hr. Under these conditions, P493-6 cells did not proliferate due to a dependency on the expression of *MYC* ((*Schuhmacher et al., 1999*)). For *MYC* re-induction, cells were washed three times with growth medium and grown in tetracycline-free culture conditions.

Western blot

P493-6 cells were harvested at the indicated times and total cell lysates were prepared by pelleting ~1×10^7 cells (determined with a C-chip disposable hemocytometer) at 4°C at 1,200 rpm for 5 min using a refrigerated centrifuge (Eppendorf, Westbury, NY, model# 5810R). After cell pellets were washed once with ice-cold 1X PBS, pellets were resuspended in RIPA lysis buffer containing 2X SIGMAFAST Protease inhibitors and 2X Phosphatase inhibitor cocktails 2 and 3. Protein concentrations were determined using the Bradford assay according to the manufacturer's instructions. Sample buffer was added to protein lysates and 50 µg of protein along with protein ladder was resolved by SDS-PAGE and transferred to PVDF membrane as described in the Registered Report ((*Blum et

Results and discussion

eLife
NEWSLETTER ABOUT SUBMIT MY RESEARCH LOG IN/REGISTER
HOME MAGAZINE COMMUNITY INNOVATION
This is an executable code view. See the original article.
▶ RUN DOCUMENT
SOURCE

Induction of c-Myc in P493-6 cells and impact on total RNA levels.
P493-6 cells were grown in the presence of tetracycline (Tet) for 72 hr and switched into Tet-free growth medium to induce c-Myc expression. Cells were cultured in two separate lots of serum. **(A)** Representative Western blot using an anti-c-Myc antibody (top panels) or an anti-ß-Actin antibody (bottom panel). Two exposures of the anti-c-Myc antibody are presented to facilitate detection of c-Myc.

Figure 1B

Induction of c-Myc in P493-6 cells and impact on total RNA levels.
P493-6 cells were grown in the presence of tetracycline (Tet) for 72 hr and switched into Tet-free growth medium to induce c-Myc expression. Cells were cultured in two separate lots of serum. **(B)** Quantification of total RNA levels (ng of total RNA per 1,000 cells) for cells at 0, 1, and 24 hr after release from Tet. Means reported and error bars represent s.e.m. from 3 independent biological repeats. For serum lot one, one-way ANOVA on total RNA levels of all groups; *F*(2, 6) = 1.26, *p* = .353. Planned contrast between 0 hr and 24 hr; *t*(6) = 1.62, *p* = .347 with a priori alpha level = .05. For serum lot two, one-way ANOVA on total RNA levels of all groups; *F*(2, 6) = 21.87, *p* = .00176. Planned contrast between 0 hr and 24 hr; *t*(6) = 5.03, *p* = .0024 with *a priori* alpha

Reproducible code for Figure 1B

eLife
NEWSLETTER ABOUT
HOME MAGAZINE COMMUNITY INNOVATION
This is an executable code view. See the original article.
▶ RUN DOCUMENT

Figure 1B

```
#' @width 17
#' @height 10

#creates new column calculating RNA in 100uL
data2$RNA.100uL <- data2$Average.RNA.Concentration*100

##calculates RNA per cell
data2$RNA.per.cell <- data2$RNA.100uL/data2$Total.Cells.Harvested

#calculates RNA per 1000 cells
data2$value <- data2$RNA.per.cell*1000

#classifies time as character
data2$Time <- as.character(data2$Time)

########## subsets and summarizes Data ##########

#subsets data on lot 1
lot1dat <- data2[which(data2$Lot=="1"),]
#subsets data on lot 2
lot2dat <- data2[which(data2$Lot=="2"),]

#summarizes lot 1 data
lot1sum <- summarySE(data=lot1dat, measurevar = "value", groupvars = "Time")
#summarizes lot 2 data
lot2sum <- summarySE(data=lot2dat, measurevar = "value", groupvars = "Time")

########## Generates bar plot for lot 1 ##########
##################################################

plot.lot1 <- ggplot(lot1sum, aes(x=Time, y=lot1sum$value, fill=Time)) +
  geom_bar(stat="identity", width=.8, color = "black") +
  geom_errorbar(aes(x=Time, ymin=value-se, ymax=value+se),
                width=.20)+
  coord_cartesian(ylim=c(0,2.5)) +
  scale_fill_manual(values = c("grey30", "grey30","grey30")) +
  ylab(expression(paste("Total RNA (ng) \n per 1,000 cells"))) +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0,6),
                     breaks = c(0, .5, 1.0, 1.5, 2.0, 2.5),
                     labels = c("0.0", "0.5", "1.0", "1.5", "2.0", "2.5")) +
  scale_x_discrete(labels = c("0hr", "1hr", "24hr")) +
  theme(plot.margin = unit(c(1,1,1,2), "lines"),
        axis.ticks.length = unit(0.25, "cm"),
        axis.text.x = element_text(size=15, color = "black"),
        axis.text.y = element_text(size = 15, color = "black"),
        axis.title.y = element_text(size = 20),
        axis.title.x = element_blank(),
        panel.background = element_blank(),
        axis.line.y = element_line(),
        legend.position = "none",
        axis.line.x = element_line())

########## Generates bar plot for lot 1 ##########
##################################################
```

Figure 1.1. Reproducible papers. An example of the Methods section and the Results section from a replication Study paper. Some parts of the results are extracted and reproduced by code e.g. Figure 1B. This paper is part of Reproducibility Project: Cancer Biology. The image contains screenshots of the paper reported in the website `https://elifesciences.org/articles/30274/executable`.

methods is a key for further research development - researchers need to be aware of the evolving methods ‘space’ so that they can confidently apply the appropriate, up-to-date methods to solve their scientific tasks. Therefore, there is a need to provide a medium of computational representation for methods reported in papers that are compatible with the FAIR (findable, accessible, interoperable and reusable) model [9, 10], so that scientists, including new domain users, could have the opportunity to better understand the methodological landscape in a given domain, including, for example:

- Finding appropriate methods for a given task;
- Finding trending and popular methods in a period of time for a given task;
- Understanding how these methods changed over time;
- Identifying by which new methods old ones are replaced;
- Finding those methods that are proposed but not used widely.

A computational FAIR representation of methods reported in a research paper could therefore reduce the time and effort required to identify papers of methodological interest, replicate the detailed information about the methods, and enhance the ability to keep track of existing and recently published methods. The main aim of this thesis is to explore how to automatically build such a methods representation from a research paper.

As a case study, the thesis uses computational biology, where computational tools are used to analyse biological data in order to provide new biological insights: it is the science of studying biology using computational techniques. Methods sections in computational biology therefore often involve data and tools used to achieve analytical goals [11]. We specifically focus on the microarray analysis literature, which includes the analysis of experiments on the expression of thousands of genes [12]. We chose this domain as it has had a sustained period of use in the last twenty years: a huge amount of biological data have been generated through research in microarray technology and methods have been continually developed to analyse these data in this changing landscape [13, 14].

There have been previous attempts to reconstruct methods from papers in computational biology. For example, Eales et al. [15] have demonstrated the ability to obtain phylogenetic methods from the literature and depicted best practice methods for a specified time frame. Additionally, Duck et al. [11] have shown that it is possible to extract methods’ components (database and software names) from the computational biology literature) and assess the usage over a period of time. Zhao et al. [16]

have focused on the resource citation hyperlinks mentioned in the scientific articles to extract the methods' components as (online-resource citation, resource role (code or data) and resource function (use or procedure)). Still, to the best of our knowledge, extraction of specific data "events" and how they are connected to other method components for an individual research article is not addressed in previous work.

In this thesis, we aim to build a computational model and represent the text from an individual paper that describes computational biology methods as an abstract workflow of data events. A *data event* is a representation of processes and changes that happen to data. It contains the main components of the experiments, such as input/output data, processes, databases and software. As an example, sentence "GSE35957 was downloaded from Gene Expression Omnibus (GEO) database" [PMC3735399] contains a data event where process *download* is applied to a data instance (*GSE35957*). An abstract workflow then models the relationships between the data events represented in a paper, and can be queried and patterns of data events for a given time can be extracted and analysed.

We note that some journals now accept "executable papers". An *executable paper* combines the paper's prose with embedded chunks of code that could be executed on provided datasets to (re)produce the results. This approach facilitates the reproducibility but only works for a very small number of papers where the authors submit the associated code. We also note that such chunks of code do not necessarily increase the understanding of the method, but rather provide an executable code. While these are useful, our aim and challenge is to use the existing literature to reconstruct computational data event workflows.

1.2 Research hypothesis and questions

The main hypothesis of this thesis is that by using text mining techniques we will be able to identify, extract and represent the individual computational biology methods that are reported in scientific articles. The consequent research questions for this study are:

1. How are methods sections written? What specific entities and discourse elements characterise method sections?

2. How are data and operations mentioned in papers? How can we use these mentions as a proxy to identify data events that are part of methods' workflows?

3. How can the answers to the previous questions be used to extract representations of computational biology workflows from text?

4. Can we use a corpus of methods in computational biology to identify common patterns of method over time?

1.3 Research aim and objectives

The main aim of this project is to design, develop and evaluate a methodology to extract and represent data events workflows from the computational biology literature, in particular for the microarray analysis case study. We focus on what is reported in the manuscript, rather than what really has been conducted. This will help to index the literature with specific data events and methods' workflows, and facilitate identification of well-known methods and new methodology patterns. The specific objectives of this project are:

1. To conduct a survey of how methods sections are written.
2. To investigate the use of text mining techniques, such as named entity recognition (NER), discourse analysis (DA) and domain relation extraction (RE) to identify data events.
3. To develop an approach for extracting data events mentioned in the Methods section by defining templates that contain information about processes conducted and reported in the text.
4. To develop a method for constructing workflows consisting of extracted data events, and evaluate the workflow construction in a case study on microarray analysis research.
5. To demonstrate how computational data event workflows can be used to identify common patterns of method over time.

1.4 Research contributions

This thesis provides the following research contributions:

1. Extracts and analyses the common writing patterns in Methods sections in the microarray analysis literature in terms of discourse.
2. Provides annotation guidelines for labelling data and operations mentioned in scientific text and provides a first manually annotated corpus of Methods sections.
3. Develops and evaluates the ODNoR system that uses text mining and machine learning to annotate, normalise and relate the operations and data used in conducting the work expressed in microarray analysis research papers.
4. Develops and evaluates the ODNoRFlow system that reconstructs an abstract workflow from a research paper.
5. Explores the common patterns of data events as reported in the microarray literature over the past 20 years.

1.5 Thesis structure

This thesis is submitted, with permission from the Faculty of Science and Engineering, in the journal format. As a result, the major chapters are organised into the structure of an individual research paper, and therefore necessarily have some repetitions that provide the context. Table 1.1 links the research questions specified above to Chapters that answered them. Specifically, the rest of this thesis is structured as follows:

- Chapter 2 introduces the background material for this thesis.
- Chapter 3 provides the analysis of discourse patterns on a complete set of open-access full-text PubMed Central literature.
- Chapter 4 introduces and uses ODNoR to annotate, normalise and relate operations and data used to express the work reported in the literature.
- Chapter 5 develops and uses ODNoRFlow to reconstruct and evaluate abstract method workflows of articles. It also uses the discourse analysis to improve the result obtained from ODNoR.

- Chapter 6 utilises the results of ODNoRFlow to automatically extract the patterns of data events of a large corpus of papers in microarray analysis research.
- Chapter 7 evaluates the primary questions and challenges that this thesis addresses, elaborates on the limitations of the work contained within, and states some ideas for further work. It also summarises the contributions of this work, and draws some conclusions.

Table 1.1. Matching the research questions with the objectives, contributions and Chapters.

Question	**Objective(s)**	**Contribution(s)**	**Chapter**
1	1	1	3
2	2	2, 3	4
3	3, 4	4	5
4	5	5	6

Chapter 2

Background

To fulfil the objectives, we searched the literature to construct background information about computational biology and how data are expressed in its methods (Obj_3). We investigated text mining and machine learning techniques to build the methods that extract data events $(Obj_{1,2,3})$. We explored evaluation techniques (Obj_4). We reviewed the available medium for representing, saving and exploring the workflow abstracts (Obj_4). Finally, we have shown how previous work attempted to extract related information of the methodologies mentioned in the papers.

2.1 Methodologies and scientific methods

The Merriam-Webster dictionary defines *methodology* as “a set of methods, rules, or ideas that are important in a science or art: a particular procedure or set of procedures” [17]. A *method* is defined as “a careful or organized plan that controls the way something is done” [18]. However, in the literature, the two terms are often used to refer to the same thing.

Scientific method is the core of science and research. It details the required steps to achieve a desired result. The steps should be described objectively and a sufficient level of information should be included to facilitate the understandability of the method used and the result obtained. This level of understandability leads to an ability to judge the method and results, or to go further and repeat the experiment, or reproduce it according to the data available.

To describe a method, two elements should be reported: participants and procedures. The participants can be defined as the subjects (animate or inanimate) of the experiment, and information should be provided on how they were chosen and any conditions (i.e. exclusion/inclusion criteria) that were applied to them. The procedures describe any steps carried out with the participants, any chronological changes that occur, the design of the methods applied, any measurement or well-known methods used, and why they were used; this provides evidence for the research’s validity [7].

The current publishing practices, however, do not guarantee that the methods will be reported in all necessary detail in the literature. A study by Ioannidis et al. [19] showed that more than a half of 18 chosen microarray experiment articles published in the Nature Genetics journal could not be reproduced due to the incomplete information provided about the dataset, insufficient details on the processing methods applied, or the unavailability of the software employed. Interestingly, the reason for being able to fully reproduce only 25% of the articles was said to be the sufficiency of the information given when describing the methods.

Recent years have brought in several reporting standards that have been devised for experiments in a number of domains. For example, in biology, Minimum Information About a Microarray Experiment (MIAME) [20] provides a set of guidelines for describing microarray experiments in a publication. It includes six mandatory elements: sufficient information about raw data, final processed data, essential sample annotation, experimental design, annotation of the array and, finally, laboratory and data processing protocols. In addition, there are available software that help researchers to produce their experiments in a format compatible with MIAME such as MAGE-TAB [21]. The method should be precise and explicit in order to ensure a high level of understandability and reproducibility.

2.2 Computational biology

Computational biology is the field where computational tools are used to analyse biological data in order to provide new insights. It is the science of studying biology using computational techniques. Huerta et al. [22] defined computational biology as "the development and application of data-analytical and theoretical methods, mathematical modelling and computational simulation techniques to the study of biological, behavioural, and social systems". Also, Yu et al. [23] pointed out that computational biology traditionally refers to the simulation of biological process. This simulation requires tools and databases generated by engineering to answer biological questions and gain biological insights.

Computational biology differs from bioinformatics in that the latter is more concerned with developing the tools that are used in the former. In other words, bioinformatics is more related to engineering, while computational biology is more related to science i.e. biology [24]. For example, BLAST (Basic Local Alignment Search Tool) is a bioinformatics algorithm developed to compare biological sequences. Using this tool in the literature is considered as a part of computational biology science, for example, using BLAST to compare amino-acid sequences in a dataset of proteins, getting results and analysing them.

Computational biology methods can be represented in executable scientific workflows. Merriam-Webster [25] defines workflow as "the set of relationships between all the activities in a project, from start to finish". Scientific workflow systems "compose and execute a series of computational or data manipulation steps, or workflow, in a scientific application" [26].

There have been several scientific workflow systems that represent computational bi-

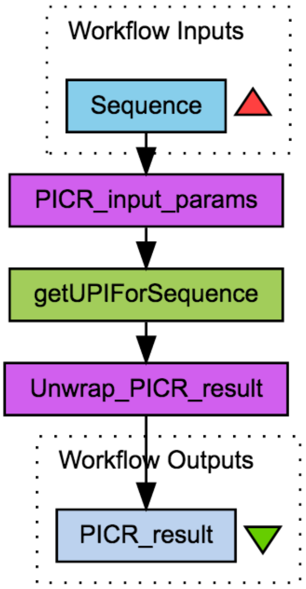


Figure 2.1. Sample workflow for mapping a protein sequence to identical sequences using EBI's PICR web service, the workflow found from myExperiment[3] by Hamish McWilliam, and which can be run on Taverna.

ology workflows, for example, Galaxy [27], Kepler [28] and Taverna [29] .

^{my}Grid team[1] have build Taverna as a workflow management system and myExperiment[2] as a repository of these workflows. Taverna has a graphical representation that enables researchers to design and execute workflows. In this context, the workflow represents an experiment's steps where each step can perform a specific task. Figure 2.1 shows an example of a workflow.

2.3 Data

2.3.1 Data definitions

Data are an important part of the methods since they represent the dataset that is being used in an experiment. In computational biology, extensive use of data is required in dry or wet lab experiments. It is important to define what the data means in general, as well as in computational biology, in order to define them in the scope of the project.

The definition of data in a number of dictionaries, encyclopaedias and research papers agrees on the point that data are facts, have a value, and can be manipulated and used as an input to a process or can be analysed as an output from a process.

[1] http://www.mygrid.org.uk/
[2] http://www.myexperiment.org/home

For example, in Merriam-Webster [30], data are either “factual information (as measurements or statistics) used as a basis for reasoning, discussion, or calculation”, ”information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful”, or ”information in numerical form that can be digitally transmitted or processed”.

Additionally, Wikipedia [31] defines data as “a set of values of qualitative or quantitative variables” and these values can be manipulated, “measured, collected, reported, and analysed” or “visualized using graphs or images”. They are also defined data at an abstract level by “the fact that some existing information or knowledge is represented or coded in some form suitable for better usage or processing”. Raw data are defined in [31] as “unprocessed data” which can be processed at one stage and considered as “raw data” for the following stage. Additionally, experimental data are “data that are generated within the context of a scientific investigation by observation and recording” [31].

Dinov et al. [32] define data by looking into its modality: as “observed biomedical (raw) data, which is typically fed as input in different computational tools; model data, processed data resulting as an output from various tools (e.g., atlases); and textual data, spread sheets, web-pages (e.g., clinical charts)”.

The Concise Encyclopaedia of Bioinformatics and Computational Biology [33] does not provide a direct definition of data, however it describes the data-related concepts such as data structure and data description language and data standards. It provides the definition of metadata by Robert Stevens as ”data about data”. He confirmed that metadata can be furthermore described by other metadata. Europe PubMed searching services describe all information related to publication as metadata. For example, the description of the publication itself (year of publication, authors’ names and page-length) and the description of the text of publication (related key words and MeSH terms, GO ontology and organisms’ terms)

2.3.2 Data in computational biology

It is of our interest to know how data are reported in computational biology. We will mention some examples of data format and data mentions. We were also interested to know whether there are comprehensive schemas or guidelines to organise the mentions and use of the data in the computational biology literature.

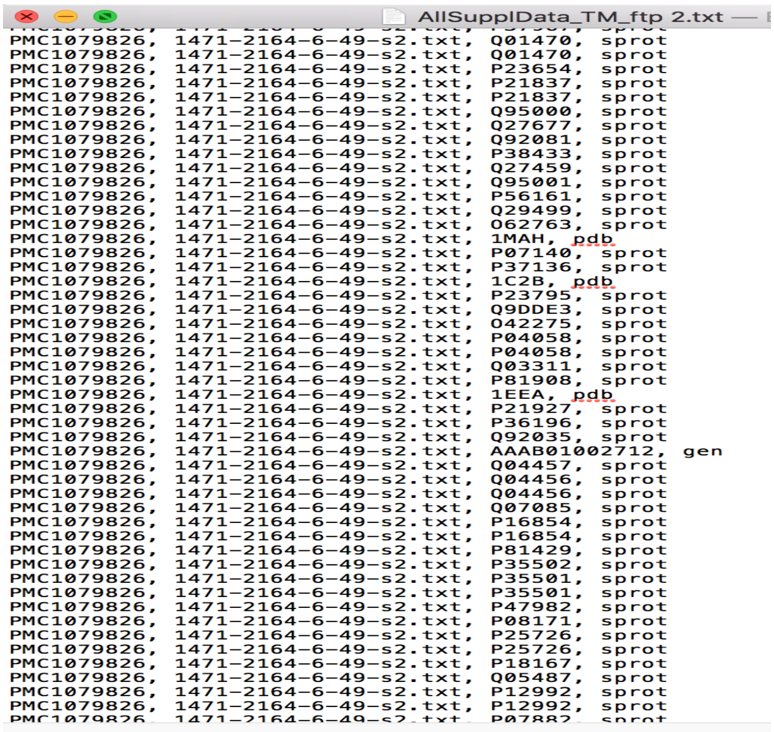
The data in computational biology can be found in different formats. For example, there are more than 300 types of formats grouped into 7 general formats in EDAM’s format ontology [34]. They can be binary format, HTML (Hypertext Markup Language), RDF (Resource Description Framework) format, Textual format, JSON (JavaScript Object Notation), XML (eXtensible Mark-up Language) and YAML (YAML Ain't Markup Language).

In the manuscripts, the data may be mentioned in a number of ways based on the purpose of that mention. For example, DOI is referred to in the text by reference or a link (*doi 10.5061/dryad.478g5*). The data may be reported by mentioning the file format where the data exist (.CLE in *raw intensity (.CEL) file*), mention of biological terms names (*Gene .04 protein*), short name (*Gp0.4*), mention of accession number as a reference to a database record (*A2BC19*), mention of sequences (*ACTATCTAGAGCGGCCGCTT*), reference to a biological concept that is used in the dataset (*cluster, transcripts*), metadata that refer to real data (*human and mouse genomes*) or parameters for named and unnamed software (for example blastn, e-value and DUST filter in *"BLASTed (blastn, $e-value = 10^6$, DUST filter off)"* or identity in *"identity >= 94%"*).

The above type of mention can be found in a supplementary file as well. Kafkas et al. [35] studied the percentage of mentions in the body of articles and supplementary files and found the supplementary files contain more mentions than the body. Figure 2.2 shows an example of accession numbers mentioned in supplementary files.

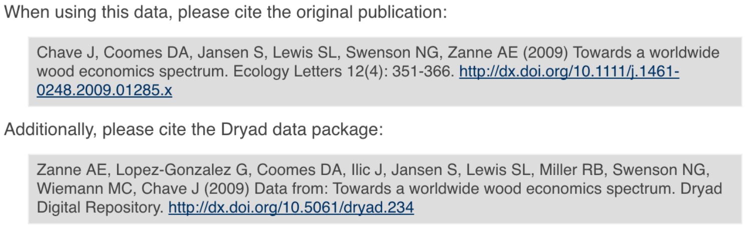
There are no comprehensive schemas for data in life science literature [37]. There are some general practices that are good to follow to facilitate the accessibility of the data used, for example, including identifiers or accession numbers for any usage of public access biological databases (e.g. Ensembl, UniProt). Another good practice is referencing the data in the References section. For example, if authors used published data deposited in Dryad [36], then they should reference the original papers that published the data and reference the data package in Dryad. Figure 2.3 shows an example.

There are some journals that have guidelines which organise the process of depositing then referencing the data in their journals. For example, PLOS journal requires the data and metadata to be deposited in a public repository to make them available and accessible. The public repositories can be subject-specific (e.g. GenBank and PDB) and can store specific types of data such as sequences and structures, or they can be general repositories that accept multiple data types. After the data are deposited, they should be referenced in the literature by their digital object identifier (DOIs) or database reference in the public repositories.



```
PMC1079826, 1471-2164-6-49-s2.txt, Q01470, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q01470, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P23654, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P21837, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P21837, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q95000, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q27677, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q92081, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P38433, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q27459, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q95001, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P56161, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q29499, sprot
PMC1079826, 1471-2164-6-49-s2.txt, O62763, sprot
PMC1079826, 1471-2164-6-49-s2.txt, 1MAH, pdb
PMC1079826, 1471-2164-6-49-s2.txt, P07140, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P37136, sprot
PMC1079826, 1471-2164-6-49-s2.txt, 1C2B, pdb
PMC1079826, 1471-2164-6-49-s2.txt, P23795, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q9DDE3, sprot
PMC1079826, 1471-2164-6-49-s2.txt, O42275, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P04058, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P04058, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q03311, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P81908, sprot
PMC1079826, 1471-2164-6-49-s2.txt, 1EEA, pdb
PMC1079826, 1471-2164-6-49-s2.txt, P21927, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P36196, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q92035, sprot
PMC1079826, 1471-2164-6-49-s2.txt, AAAB01002712, gen
PMC1079826, 1471-2164-6-49-s2.txt, Q04457, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q04456, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q04456, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q07085, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P16854, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P16854, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P81429, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P35502, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P35501, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P35501, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P47982, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P08171, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P25726, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P25726, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P18167, sprot
PMC1079826, 1471-2164-6-49-s2.txt, Q05487, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P12992, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P12992, sprot
PMC1079826, 1471-2164-6-49-s2.txt, P07882, sprot
```

Figure 2.2. An example of a supplementary file with accession number annotation. The first column refers to the article full text identifier(PMC), the second is the name of the supplementary file, the third is the annotated accession numbers and the fourth is the contextual cue to the database (e.g. sprot for UniProt). Image is taken from Kafkas et al. [35].



When using this data, please cite the original publication:

Chave J, Coomes DA, Jansen S, Lewis SL, Swenson NG, Zanne AE (2009) Towards a worldwide wood economics spectrum. Ecology Letters 12(4): 351-366. http://dx.doi.org/10.1111/j.1461-0248.2009.01285.x

Additionally, please cite the Dryad data package:

Zanne AE, Lopez-Gonzalez G, Coomes DA, Ilic J, Jansen S, Lewis SL, Miller RB, Swenson NG, Wiemann MC, Chave J (2009) Data from: Towards a worldwide wood economics spectrum. Dryad Digital Repository. http://dx.doi.org/10.5061/dryad.234

Figure 2.3. If authors used data stored in Dryad, they have to cite two references. The first one is the reference for the original paper that deposits the data in Dryad (top reference in the figure) and the second one is the reference to the data package that is stored in Dryad. Image is taken from Dryad [36].

2.3.3 Data repositories

Data instances can be found in lists, thesauruses, controlled variables or ontologies. In this context we are going to explore the use of ontologies as a resource of data.

An ontology is a machine-readable description of specific domain concepts, the properties of these concepts, and the constraints of these properties [38]. The ontology concepts are classes and subclasses that are linked to relations extracted from the domain. They contain the common domain terms so that the domain knowledge can be shared for analysis and annotation tasks. An ontology itself is only a definition

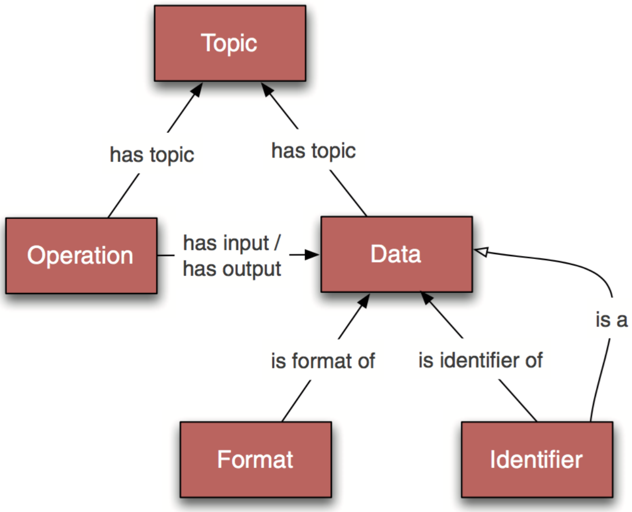


Figure 2.4. EDAM ontology concepts and relations. Data has a topic, is an output or input to an operation, identified by an identifier and has a format. Image is taken from Ison et al. [34].

of classes, and together with instances of classes, they comprise a knowledge base. Domain experts tend to produce standard ontologies to facilitate the reuse and annotation of domain terms [38]. In the following sections, examples of general data ontologies and computational biology ontologies are described.

EDAM

EDAM [34] is an ontology that describes bioinformatics and biomedical resources. It includes five sub-ontologies linked to five types of relations (see Figure 2.4). The five sub-ontologies are topic, data, operation, format and identifiers.

Topic declares the general concept's terms e.g. *computational biology* concept is a main class in topic ontology and the subclasses can be *nucleic acids, phylogeny, proteins, etc.*

Data are more specific than a topic; examples of biological data are *protein sequences* and *protein sequence alignment*.

Operation is where the concepts of processes are defined, with their relationships to possible inputs and outputs e.g. *alignment, mapping, clustering, etc.* (see Figures 2.5 and 2.6).

Format defines how the data are represented, e.g. *binary or XML* (format is detailed in Section 2.3.2), and the identifier is a subset of the data and refers to an entity e.g. *UniProt accession*.

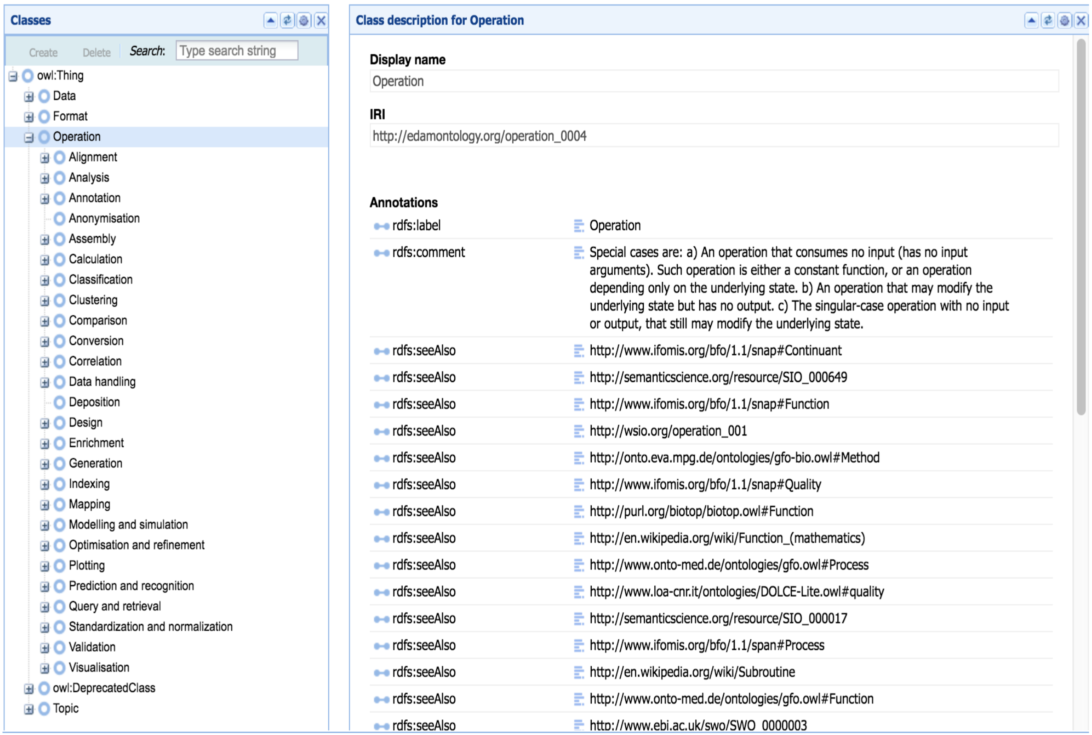
The current version of EDAM includes over 2,200 concepts that are defined by names (terms), synonyms, definitions and other properties. The definitions of concepts include useful terms that can be used as control variables for finding the biological terms in the text. They can also be used for semantic annotation of workflows and web services, and as a standard for exchanging data. EDAM also can be used for verifying files and exchange formats, as some useful information about regular expressions can be used to validate the exchanged identifier values.

Software Ontology

The Software Ontology (SWO) [39] is an ontology that defines computational biology software in terms of data, data format, algorithm, organisation, programming language and software license (see Figure 2.7). The data and data format terms are merged with the EDAM data and format terms that result (1,168 and 434 terms respectively, compared to 1,140 and 347 terms in EDAM).

Gene Ontology

Gene Ontology (GO) [40] contains three GO domains: cellular component, molecular function and biological process. There is no *is–a* relation between the sub-ontologies but other relations such as *part–of* and *regulates* can be maintained between them. The structure of a node in GO consists of essential and optional parts. The mandatory elements are GO ID (*GO:0005125*), namespace which refers to the sub-ontology, definition and relation to other GO elements. GO optionally defines database cross-references where Gene and protein nodes are linked to their databases such as SwissProt, GenBank, EMBL, DDBJ, PIR, MIPS, YPD and WormPD, Pfam, SCOP and ENZYME. Figure 2.8 shows a representation of GO ontology.



Class description for Operation

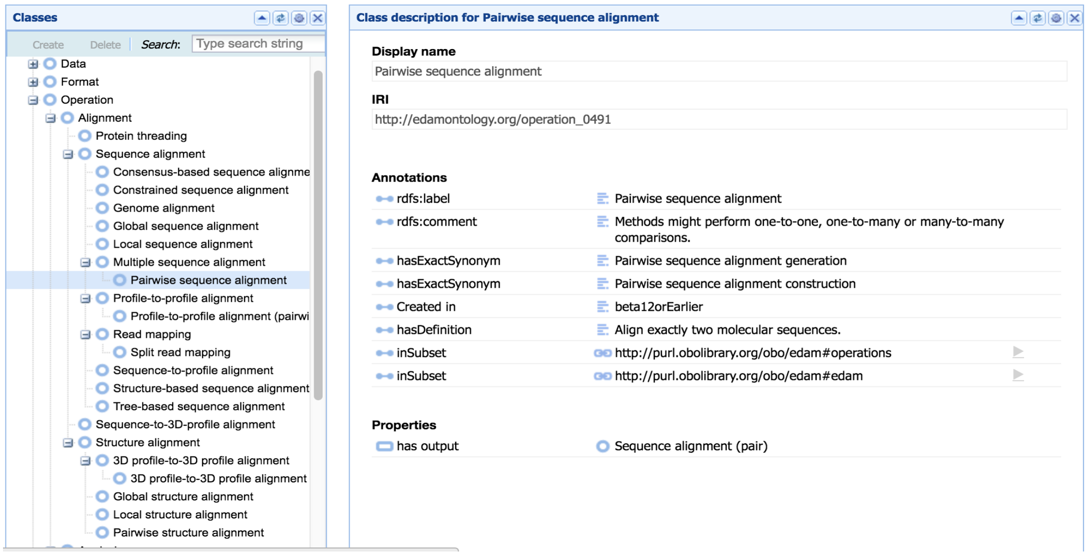
Display name
Operation

IRI
http://edamontology.org/operation_0004

Annotations

rdfs:label	Operation
rdfs:comment	Special cases are: a) An operation that consumes no input (has no input arguments). Such operation is either a constant function, or an operation depending only on the underlying state. b) An operation that may modify the underlying state but has no output. c) The singular-case operation with no input or output, that still may modify the underlying state.
rdfs:seeAlso	http://www.ifomis.org/bfo/1.1/snap#Continuant
rdfs:seeAlso	http://semanticscience.org/resource/SIO_000649
rdfs:seeAlso	http://www.ifomis.org/bfo/1.1/snap#Function
rdfs:seeAlso	http://wsio.org/operation_001
rdfs:seeAlso	http://onto.eva.mpg.de/ontologies/gfo-bio.owl#Method
rdfs:seeAlso	http://www.ifomis.org/bfo/1.1/snap#Quality
rdfs:seeAlso	http://purl.org/biotop/biotop.owl#Function
rdfs:seeAlso	http://en.wikipedia.org/wiki/Function_(mathematics)
rdfs:seeAlso	http://www.onto-med.de/ontologies/gfo.owl#Process
rdfs:seeAlso	http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl#quality
rdfs:seeAlso	http://semanticscience.org/resource/SIO_000017
rdfs:seeAlso	http://www.ifomis.org/bfo/1.1/span#Process
rdfs:seeAlso	http://en.wikipedia.org/wiki/Subroutine
rdfs:seeAlso	http://www.onto-med.de/ontologies/gfo.owl#Function
rdfs:seeAlso	http://www.ebi.ac.uk/swo/SWO_0000003

Figure 2.5. EDAM operation subclasses.


Classes
Create
Delete
Search:
Type search string
Data
Format
Operation
Alignment
Protein threading
Sequence alignment
Consensus-based sequence alignme
Constrained sequence alignment
Genome alignment
Global sequence alignment
Local sequence alignment
Multiple sequence alignment
Pairwise sequence alignment
Profile-to-profile alignment
Profile-to-profile alignment (pairwi
Read mapping
Split read mapping
Sequence-to-profile alignment
Structure-based sequence alignment
Tree-based sequence alignment
Sequence-to-3D-profile alignment
Structure alignment
3D profile-to-3D profile alignment
3D profile-to-3D profile alignment
Global structure alignment
Local structure alignment
Pairwise structure alignment

Class description for Pairwise sequence alignment

Display name
Pairwise sequence alignment

IRI
http://edamontology.org/operation_0491

Annotations

rdfs:label	Pairwise sequence alignment
rdfs:comment	Methods might perform one-to-one, one-to-many or many-to-many comparisons.
hasExactSynonym	Pairwise sequence alignment generation
hasExactSynonym	Pairwise sequence alignment construction
Created in	beta12orEarlier
hasDefinition	Align exactly two molecular sequences.
inSubset	http://purl.obolibrary.org/obo/edam#operations
inSubset	http://purl.obolibrary.org/obo/edam#edam

Properties

has output	Sequence alignment (pair)

Figure 2.6. As an example of how a class is defined in EDAM ontology, pairwise alignment is an operation under the sequence alignment operation class. It has a definition, synonym and properties that indicate this operation has an output data of type sequence alignment (pair).

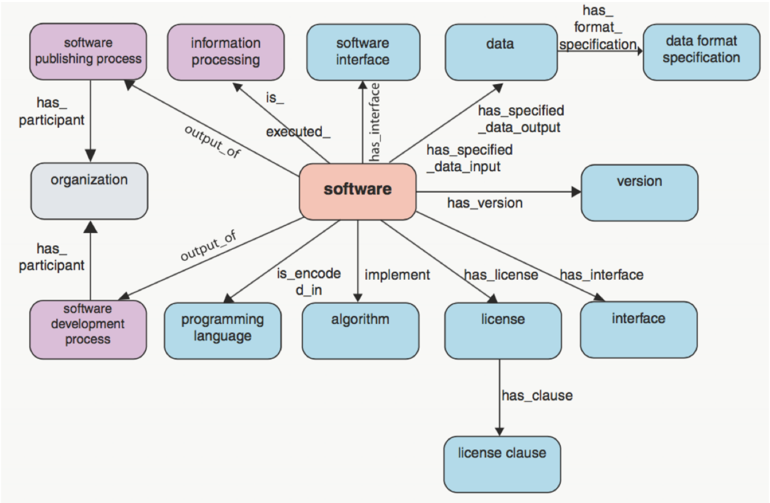


Figure 2.7. Software ontology concepts and relations. Image is taken from Malone et al. [39].

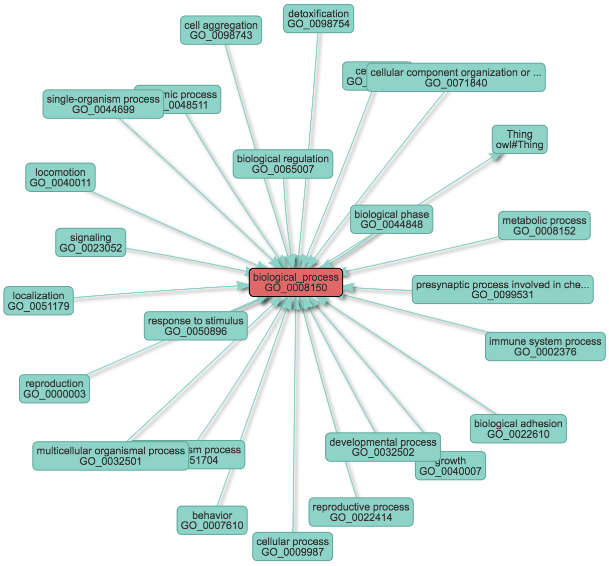


Figure 2.8. Graphical representation of subclasses of biological process class in Gene Ontology.

2.4 Machine Learning

Machine learning (ML) is a sub-field of artificial intelligence (AI), which uses computer science, statistics and data science to develop systems that automatically improved by experience [41]. The developments of novel learning algorithms and theory, as well as availability of online data and low-cost computation, have spurred recent advances in machine learning [41]. The available techniques (such as Support Vector Machine (SVM), Conditional Random Fields (CRF), Hidden Markov model (HMM), Random Forest, etc.) can be supervised or unsupervised. Supervised models need sufficient training data to learn from, however, unsupervised models do not need training data, but may suffer from over-fitting where noisy data are captured by the model.

- **Deep learning**

  Deep learning is a type of machine learning that uses a multi-layer neural network to learn complex features of the input data and improve the prediction output. It surpasses the traditional machine learning algorithms by automating the features extractions process and learning non-linear relations and hence boundaries between inputs [42, 43]. Deep learning is usually implemented using a neural network architecture. A Feedforward neural network (FNN) is an artificial neural network that has multi-layered architecture where all nodes in one layer are fully connected to the nodes in the next layer. It has an input layer that accepts the input features, and an output layer that produces the output of the network. The layers in between are known as hidden layers. They process the inputs based on weights and activation functions and pass the knowledge from layer to layer to learn complex features. Figure 2.9 shows the layout of the feedforward neural network. Bidirectional Long short-term memory (BiLSTM) is a model with two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. Knowing the preceding and following words of a word gives more level of context and hence improves the learning process and performance of the model. Figure 2.10 shows the layout of the BiLSTM neural network.

- **Transformer**

  Transformers are the evolution of encoder decoder architecture. The encoder reads the text input and the decoder produces a prediction for the task. Faster training with better performance is achieved due to utilising parallel processing and attentions. Transformer consists of multi-layers; each layer contains multi heads, each head is the weighted sum of the value vectors. The vectors corre-

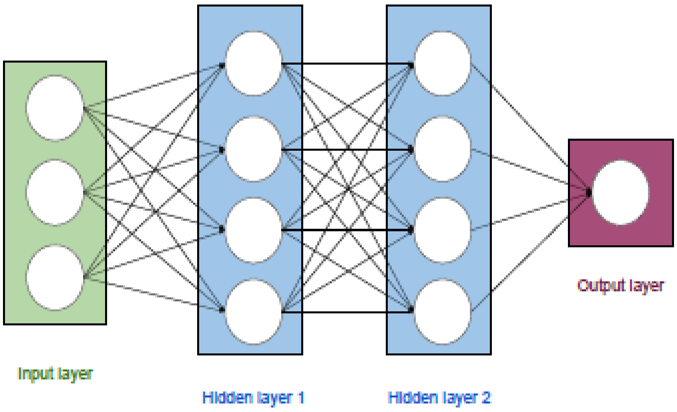


Figure 2.9. Feedforward neural network with two hidden layers. Information always moves forward, no feedback connections. The input layer accepts the input features, and the output layer is the final layer that produces the output of the network. The layers in between are known as hidden layers. Image is taken from Shridhar et al. [44].

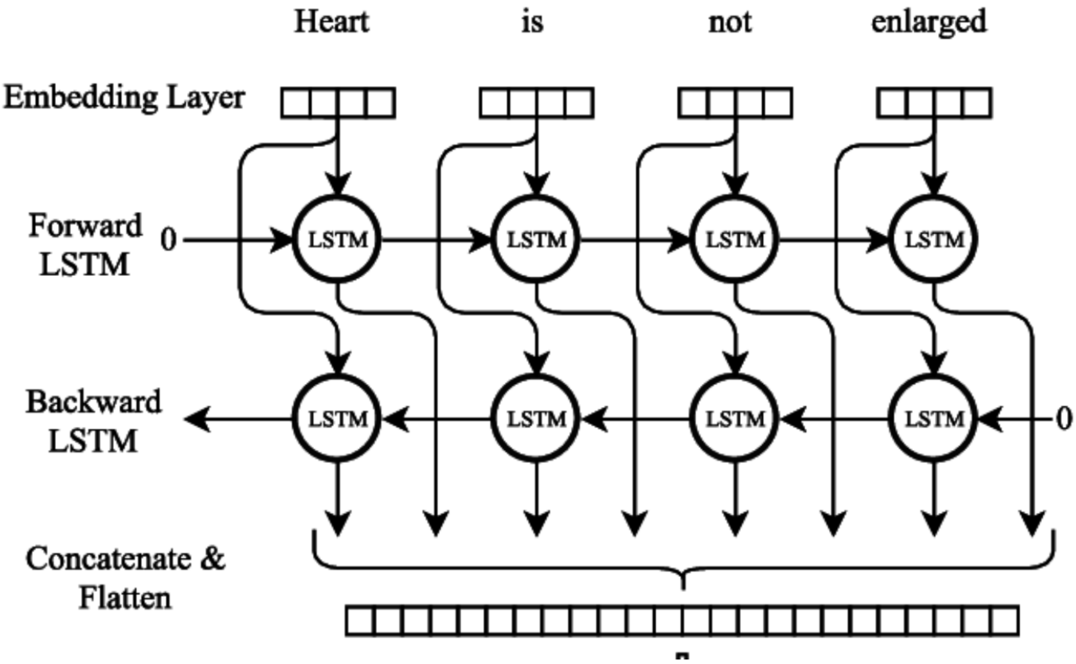


Figure 2.10. BiLSTM neural network architecture. Image is taken from Cornegruta et al. [45].

sponding to each token/word are transformed into query key value vectors, some calculations are done using softmax-normalization dot product. There are two deep learning frameworks: TensorFlow [46] and PyTorch [47]

Attention [48] is a well-known method used in modern deep-learning models. It provides the neural network with the ability to "focus" on features (select inputs). Transformers use attentions to boost the speed of training.

Self-attention is the mechanism the transformer uses to learn the value of the relevance between the current word and all words of the input sequence.

The formula that calculates the outputs of the self-attention layer:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where

$\sqrt{d_k}$ is the dimension of the key vector K and query vector Q.

The following steps (two through six) detail how to calculate the self-attention values:

1. For each word in the input sequence, calculate the query (Q), keys (K) and values (V) vectors. These vectors are calculated by multiplying the word embedding of the word by three weight matrices. The weight matrices are calculated during the training process.

2. Calculate a score for each word in a sequence against other words in the input sequence. The score is calculated by the dot product of the query vector of the word with the key vector of the other words. For example: scores for word1, against word1 = q1.k1, against word2 = q1.k2, ..

3. Divide the scores by the square root of the length of Q and K vectors.

4. Normalise the scores by softmax, so they are all positive and add up to 1. This softmax score determines the relation between the word and all words positions, either the same word position or other words' positions. The higher the scores, the more relevant the words. The word has the highest softmax score to itself. Softmax gets the value produced for each class and returns the probability of each class.

5. Multiply each softmax score by V value vector. This will lower the score of the irrelevant words.

6. For each word, sum up the weighted values vectors; this produces one vector that expresses the self-attention of the word position and forwards it to the feedforward neural network layer.

Self-attention produces one vector, and the value of the word's position is the highest. If we have multi-layers of self-attention, the model will be able to inspect different positions. **Multi-head attention** uses multiple sets of Query, Key, and Value weight matrices to produce more vectors for each word. The multiple attentions are concatenated and multiplied by a weight matrix (W^O) to produce one vector for the feedforward neural network layer.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

where

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

We will explore Bidirectional Encoder Representations from Transformers (BERT), the state-of-the-art language model. It used transformers, which utilise the attentions to build robust models that can be fine-tuned to different tasks or trained on more features. Figure 2.11 shows a graphical representation of BERT, transformers and attentions.

BERT [49] is a new open-source model released by Google. It has pre-trained models that are trained on textbooks and Wikipedia.

The BERT can be fine-tuned to do NLP tasks such as NER. It also can be trained on specific domains like BioBERT [50] which trained on PubMed and PMC articles and SciBERT [51] that is trained on papers from the corpus of semanticscholar.org .

The power of BERT is first, that it has multiple head attentions. Twelve attention heads in BERT base and 16 in BERT large. Second, it applies bidirectional training, which learns contextual relations of a text sequence from all surrounding words (left to right and right to left) at the same time. This overcomes the traditional way of combining two directional left-to-right and right-to-left training.

BERT can be used as a feature-based or fine-tuning model. Input features are extracted from word embedding. A word embedding is a natural language modelling technique that turns a word into a vector of continuous values that represent a lot of meaning/semantic information of the word. The vector could have length of 256, 512 or 1024. Instead of starting a model with initial embedding,

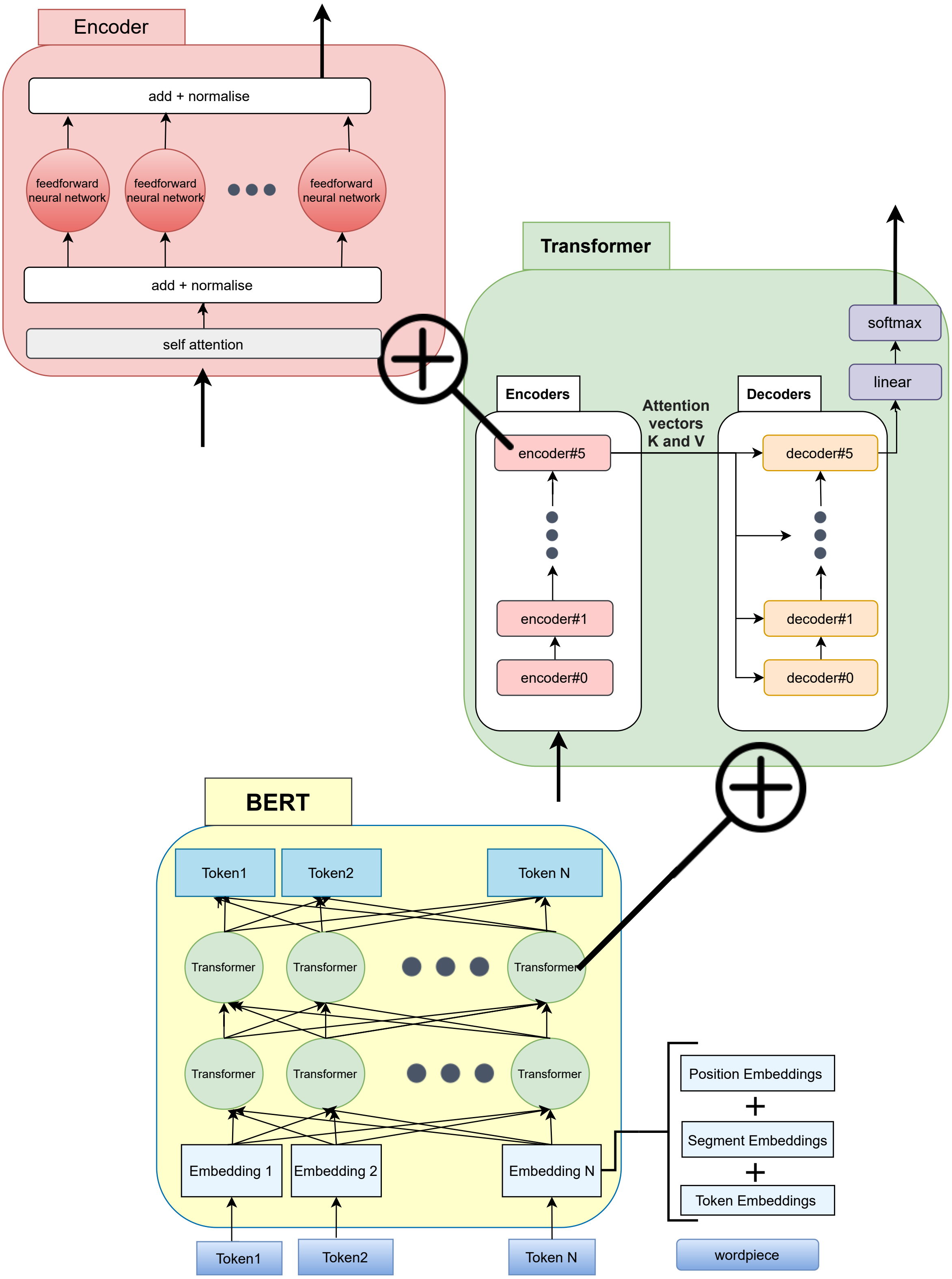


Figure 2.11. BERT uses a bidirectional Transformer. BERT input representation is the tokens by wordpiece and the input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings. Transformer is an encoder decoder that uses attentions. BERT uses only the encoder mechanism.

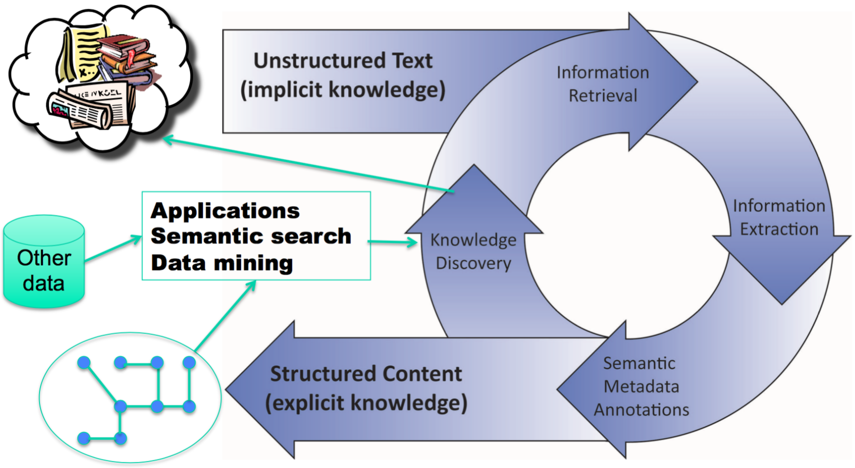


Figure 2.12. Text mining aims to convert unstructured text to structured data. The text mining pipeline typically contains information retrieval, information extraction, semantic metadata annotation followed by knowledge discovery. Image is taken from McNaught [52].

the word embeddings resulted from pre-trained models can be used as input features for other models.

A word is directly mapped to its word or subword as it appears in the BERT vocabulary words (wordpiece embeddings). If the word is absent from the vocabulary file, it is divided into pieces that are available. Any corpus can be represented by around 30k subword vocab. BERT vector is a context sensitive; the word order, the position and the neighbour contribute to the attention head matrices.

Fine-tuning is using the trained models as a ground framework of a new purpose-specific model. For example, with a classification layer that predicts the NER label, BERT can be trained to recognise the type of entities mentioned in a sequence of a text.

2.5 Text mining

Text mining or text data mining is the process of finding and extracting information from text [53]. Automating this process aims at handling the rapid development of domains that deliver a huge amount of data containing valuable information. Without these automated techniques, it would be impossible to follow, investigate and draw insights from the data. The key difference between data processing systems and language processing systems is the use of language-related knowledge [54, chap. 1].

Manning and Schütze [54, chap. 1] states six specific levels of language behaviour: The first one is phonetics and phonology, where sounds and their phonemes are the units of processing. The second is morphology where morphemes and other linguistic units, such as roots, affixes and parts of speech are identified and analysed. The third is the syntax, which relates to the order of the words and how they are combined to form the structure. The fourth is semantics, where the meanings of words are studied. The fifth is pragmatics, which addresses how the meaning is related to the goals and intentions of the writer/speaker. Finally, there is the discourse, which interprets the meaning of a whole paragraph or document.

2.5.1 Information retrieval

Text mining is divided into three phases: information retrieval, information extraction and data mining [55, chap. 2]. Information retrieval retrieves the text/audio from resources based on the terms of the query, while the information extraction is more precise and considers the syntax, and sometimes semantics, and the highest two language behaviours in the extraction process. In terms of information retrieval, there are a number of free accessible repositories for life science publications, for example:

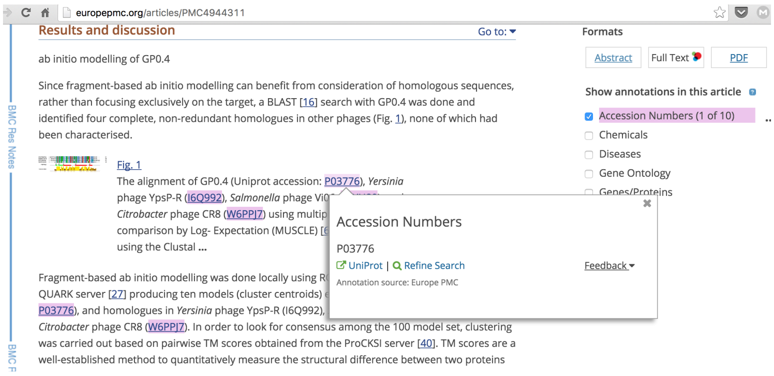
- **PubMed**[4] includes more than 26 million biomedical literature abstracts with links to full text. The links may include references to full-text content from PubMed Central[5] which contains 4 million free access articles, or a reference to where the paper is published in life science journals. The articles are identified by a PubMed reference number (PMID) and PubMed Central reference number (PMCID). PMID identifies the abstracts in PubMed, while PMCID identifies the full text in PubMed Central. The articles can be searched by multiple criteria such as MeSH terms.

- **Europe PMC**[6] (EPMC) contains 31.4 million abstracts and 3.8 million full texts. It provides searches over the abstract and full text at the same time. The EPMC offers search services through their website as well as a programming guideline for two kinds of web services Representational State Transfer (RESTful) and Simple Object Access Protocol (SOAP). The search criteria can be keywords searched at specific article parts (e.g. abstract, methods, results or supplementary data) and the search query can retrieve the articles that contain DOI or data mention of different databases (e.g. OMIM, Pfam or Ensembl) or different ontologies (GO or ChEBI). The search results can be sorted by “times cited”, which displays the highly cited publications first. The retrieved articles are annotated further by biomedical terms, GO terms and accession numbers. See Fig-

[4] http://www.ncbi.nlm.nih.gov/pubmed

[5] http://www.ncbi.nlm.nih.gov/pmc

[6] http://europepmc.org/



Results and discussion

ab initio modelling of GP0.4

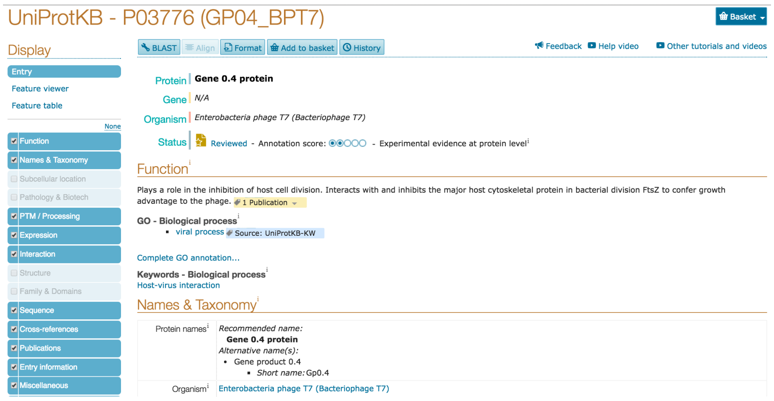
Since fragment-based ab initio modelling can benefit from consideration of homologous sequences, rather than focusing exclusively on the target, a BLAST [16] search with GP0.4 was done and identified four complete, non-redundant homologues in other phages (Fig. 1), none of which had been characterised.

Fig. 1
The alignment of GP0.4 (Uniprot accession: P03776), *Yersinia* phage YpsP-R (I6Q992), *Salmonella* phage Vi0
Citrobacter phage CR8 (W6PPJ7) using multip
comparison by Log- Expectation (MUSCLE) [6
using the Clustal ...

Fragment-based ab initio modelling was done locally using R
QUARK server [27] producing ten models (cluster centroids) e
P03776), and homologues in *Yersinia* phage YpsP-R (I6Q992),
Citrobacter phage CR8 (W6PPJ7). In order to look for consensus among the 100 model set, clustering was carried out based on pairwise TM scores obtained from the ProCKSI server [40]. TM scores are a well-established method to quantitatively measure the structural difference between two proteins

Accession Numbers
P03776
Annotation source: Europe PMC

Figure 2.13. The articles in EPMC are annotated by features shown on the right (e.g. accession numbers, GO, diseases, etc.). The annotated terms are linked to their definition in corresponding databases or ontologies.



UniProtKB - P03776 (GP04_BPT7)

Protein | **Gene 0.4 protein**
Gene | *N/A*
Organism | *Enterobacteria phage T7 (Bacteriophage T7)*
Status | Reviewed - Annotation score: ●●○○○ - Experimental evidence at protein level

Function

Plays a role in the inhibition of host cell division. Interacts with and inhibits the major host cytoskeletal protein in bacterial division FtsZ to confer growth advantage to the phage. 1 Publication

GO - Biological process
- viral process Source: UniProtKB-KW

Complete GO annotation...

Keywords - Biological process
Host-virus interaction

Names & Taxonomy

Protein names: *Recommended name:* **Gene 0.4 protein**
Alternative name(s):
- Gene product 0.4
  - *Short name:* Gp0.4

Organism: Enterobacteria phage T7 (Bacteriophage T7)

Figure 2.14. The definition of Gene 0.4 protein (Gp0.4) in UniProtKB. It is linked to the annotated accession number P03776 in Figure 2.13.

ures 2.13 and 2.14 that show an example of an annotated article that is linked to a related database. As an example of using EPMC RESTful API services, Bousfield et al. [56] studied the reusage of data in scientific articles and patents by tracking the data mention and whether it was cited by other publications.

2.5.2 Information extraction

For information extraction (IE), pre-processing text is usually performed before applying extraction techniques. It includes lexical level steps, such as tokenization, where text is split into tokens (word, space, punctuation, etc.), sentence splitting where the text is divided into sentences, stemming which returns the word to its root and part of speech tagging where the word is assigned to its classification as verb, noun, adjective or preposition. Also, it may include steps at the syntactic level such as chunking and dependency parsing. This report will explore three IE categories: Named Entity Recognition (NER), Entity linking (EL) and Relation extraction (RE).

Named-entity recognition

NER is the process of identifying a group of terms that belong to the same cluster. Krauthammer and Nenadic [57] stated three steps of NER: term recognition, where the terms are identified, term classification where the identified terms are linked to their class, and term mapping where these terms are normalised to unique identifiers. The third steps, term mapping, is considered as an important task in IE that also known as entity linking.

Entity linking

Entity linking, also known as entity normalisation in biomedical literature, simply aims to find a corresponding concept defined in a knowledge base (KB) or an ontology and link it to the annotated entities. There are three main challenges faced while doing this task, absence, if there is no corresponding concept, ambiguity, if the same entity linked by different concepts, variation, if the same concept linked by different entity mentions [58].

Relation extraction

Relation extraction (RE) is an important task in information extraction (IE). It aims to extract and classify relational between identified entity mentions in plain text. Generally speaking, relation extraction modules can be classified into two categories, rule-based approaches relying on predefined patterns and machine learning methods based on well-designed features.

Most relation extraction systems focus on extracting binary relations [59]. Examples of dataset are SemEval-2010 Task 8 [60], KBP37 [61] and TACRED [62].

SemEval-2010 Task 8 [60] containing 10,717 annotated examples covering nine relations: cause-effect, instrument-Agency, product-Producer, content-container, entity-Origin, entity-Destination, component-Whole, member-Collection and Communication-Topic. They added *other* relation to express the relations that are apart from the nine explicitly annotated relations. Examples are:

- sentence: *"He had chest pains and <e1>headaches</e1> from <e2>mold</e2> in the bedrooms."* has the relation *Cause-Effect(e2,e1)*.
- sentence: *"<e1>People</e1> have been moving back into <e2>downtown</e2>."* has the relation *Entity-Destination(e1,e2)*.

- Sentence: *"The <e2>farmer</e2> grows <e1>apples</e1>"* has the relation *Product-Producer(e2,e1).*

There are few dataset handles the relations between entities across sentences (i.e. two entities are mentioned in two different sentences) [63]. Examples of dataset that address the relations between entities at the sentence level and the document level are BioRED [64], DocOIE [65] and DocRED [66].

Information extraction approaches:

- **Dictionary-based:**

  The dictionary has a list of all possible terms that need to be annotated in the text. The advantage of this approach is its ease, which requires only a match between the text terms and the dictionary. The disadvantage is the creation of such a dictionary, especially with the evolving fields that generate new terms. It also increases the level of ambiguity since the matching can be with more than one entry in the dictionary, e.g. "Bad" can be linked to behaviour or to a protein name. Linnaeus [67] is a dictionary-based recogniser that identifies species names with an F-score of 95%. It can be configured to adapt other dictionaries from different domain.

  Dictionary lookup approach is widely used to enhance the entity linking [68, 58]. A dictionary contains term(s) and a mapping value, e.g. normalising gene name, reducing many synonyms and phrases representing the same concept to a single identifier for that gene. Cohen [69] built a dictionary based gene and protein NER from online genomics resources and achieved a level of performance comparable to state-of-the-art systems that require supervised learning and manual dictionary creation.

  For relation extraction, the dictionaries can be used to identify the relations if their indicators are fixed and can be easily enumerated, which is not commonly possible. For example, Zhou et al. [70] have a relation trigger words identification dictionary for identifying protein-protein interactions (PPIs) because the words describing interactions between proteins are more likely fixed. They have an example of a sentence "Leukotriene B4 stimulates c-fos and c-jun gene transcription and AP-1 binding activity in human monocytes" that contains three PPIs as:

  - *Stimulate (leukotriene B4, c-fos)*,
  - *Stimulate (leukotriene B4, c-jun)* and

  - *Stimulate (leukotriene B4, AP-1).*

  However, they need to use rules to extract biomolecular events, such as positive regulation and transportation, because the trigger words for these relations cannot be handled by only dictionaries.

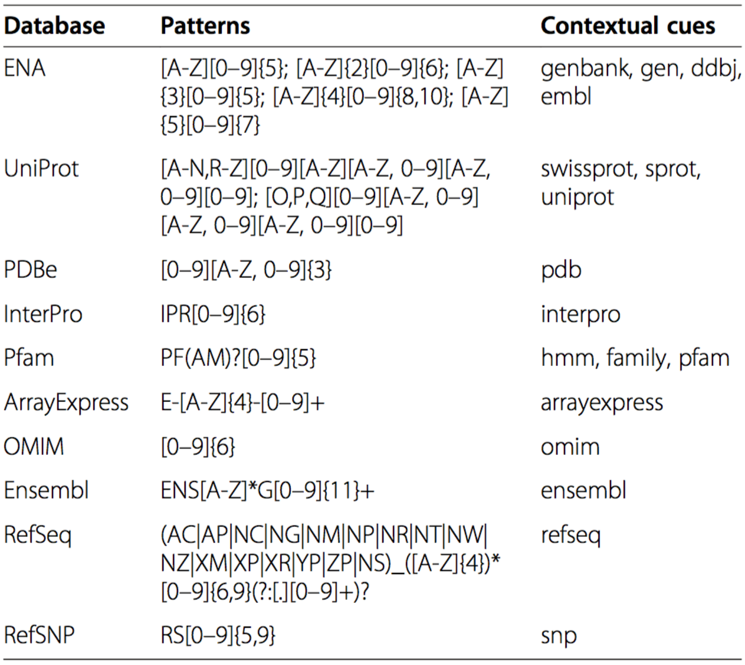
- **Rule-based:**

  This approach uses lexical and syntactical features in rules to match the terms, e.g. the task that returns the name of cities that followed by the word "university" to annotate potential names of universities. The advantage of this approach is that it can be used as a second step to filter the results created by dictionary based systems and give more flexibility in matching the terms. The disadvantage is it should be manually identified and created.

  BioNerDS [71] is a named entity recognition system for bioinformatics software and databases. It has an F-score at mention level of 63-91% and 63-78% at document level. The mention level F-score is calculated for all the mentions over the corpus. The document level F-score is calculated for the mentions over the document. It has a case-sensitive dictionary with 8,214 entries collected from online resources to annotate known names and 17 rules to recognise unknown names. The result is filtered by a machine-learning classifier to eliminate the terms incorrectly annotated as resource names.

  Whatizit was initially developed by Rebholz-Schuhmann et al. [72] and identified names that refer to biomedical terms and linked them to their definition in a public database. Kafkas et al. [73] expanded it to include identification of accession numbers in three public databases. Kafkas et al. [35] improved the two previous works and obtain an F-score 77%-96% in a total of ten databases. The patterns that are used can be seen in Figure 2.15. The service is now integrated with the EPMC search service and the identified terms or accession numbers are tagged by XML then wrapped to HTML to facilitate navigation to their definition in the public access database.

  For entity linking, most of the biomedical entity normalisation studies in the last decades use the morphological information to normalise the entities [74]. The state-of-art rule-based system proposed by D'Souza and Ng [75] on two datasets, ShARe/CLEF eHealth Challenge corpus [76] and NCBI disease corpus [77]. It achieved F-score 90.75% and 84.65% on the two datasets. It defined ten kinds of rules at different priority levels, such as abbreviation expansion, to measure morphological similarities between disorder mentions and concepts in two Knowledge base: SNOMED-CT resource of the UMLS Metathesaurus [78] and MEDIC lexicon [79].



Database	Patterns	Contextual cues
ENA	[A-Z][0–9]{5}; [A-Z]{2}[0–9]{6}; [A-Z]{3}[0–9]{5}; [A-Z]{4}[0–9]{8,10}; [A-Z]{5}[0–9]{7}	genbank, gen, ddbj, embl
UniProt	[A-N,R-Z][0–9][A-Z][A-Z, 0–9][A-Z, 0–9][0–9]; [O,P,Q][0–9][A-Z, 0–9][A-Z, 0–9][A-Z, 0–9][0–9]	swissprot, sprot, uniprot
PDBe	[0–9][A-Z, 0–9]{3}	pdb
InterPro	IPR[0–9]{6}	interpro
Pfam	PF(AM)?[0–9]{5}	hmm, family, pfam
ArrayExpress	E-[A-Z]{4}-[0–9]+	arrayexpress
OMIM	[0–9]{6}	omim
Ensembl	ENS[A-Z]*G[0–9]{11}+	ensembl
RefSeq	(AC|AP|NC|NG|NM|NP|NR|NT|NW|NZ|XM|XP|XR|YP|ZP|NS)_([A-Z]{4})*[0–9]{6,9}(?:[.][0–9]+)?	refseq
RefSNP	RS[0–9]{5,9}	snp

Figure 2.15. Data accession reference patterns and contextual clues used for ten databases used to extract the accession numbers. Image is taken from Kafkas et al. [35]

For relation extractions, pattern rules are constructed from syntactic and semantic features in text. Recent work involved larger corpora for extracting better patterns [63].

PATTY system [80] is based on mining algorithm that computes the n-gram combinations with large co-occurrence support. It processed two different corpora: the New York Times archive and the English edition of Wikipedia and achieved an accuracy between 75%-84.7%. It produced a large resource of relational patterns that are arranged in a semantically meaningful taxonomy, along with entity-pair instances. Examples of extracted patterns are:

- *<person> criticized by <organization>* for the relation *critizedByMedia*
- *<person> successfully sued <person>* for the relation *suedBy*
- *<musician> PRP idol <musician>* for the relation *hasMusicalIdol*
- *<musician> wrote hits for <musician>* for the relation *wroteHitsFor*

RelEx [81] is a rule-based biomedical relationship extraction system. It extracted 150,000 relations between genes and proteins from set of one million MEDLINE abstracts with an F-score of 78%. It used the dependency parse trees in comination with noun-phrase chunk that contains entities to build three extraction rules:

  - *effector-relation-effectee ('A activates B')*
  - *relation-of-effectee-by-effector ('Activation of A by B')*
  - *relation-between-effector-and-effectee ('Interaction between A and B').*

- **Machine learning:**

  In machine learning (ML) techniques, the patterns are extracted by training models on pre-processed data representation. This approach better result are achieved with less human efforts.

  There are many NER systems that have been implemented using ML. BERT model [49] can be fine-tuned for NER task. It achieved the state-of-the-art performance, F-score of 94.45%, on the English version of the standard CoNLL-2003 Named Entity Recognition dataset [82]. CoNLL-2003 dataset consists of 1393 English news articles and is annotated with four entity types: location (LOC), organizations (ORG), person (PER) and Miscellaneous (MISC).

  The current state-of-the-art in entity linking is EntQA[83]. They followed the trend of formulating language tasks as QA problems. They proposed inverted open-domain QA by performing entity linking before entity extraction. Given a passage, the retrieval model retrieves candidate entities that might be mentioned in the passage (questions), by studying the description of the passage, then the reader model predict potentially mentions (answers). They only used standard pre-trained Transformers for initialisation and is directly fine-tuned on AIDA dataset. EntQA achieved strong performance on the GERBIL benchmarking platform [84] with an F1 score of 85.8% (2.1 absolute improvement) on the test portion of the AIDA-CoNLL dataset [85].

  In the biomedical text, the state-of-the-art is achieved by Ji et al. [86] which increased the accuracy of biomedical entity normalisation by 1.17%. They compare the performance of three transformers models (BERT [49], bioBERT [50] and ClinicalBERTB [87]) on three different datasets (ShARe/CLEF [76], NCBI [77] and TAC2017ADR [88]) and achieved the best F-score as 91.10% 89.06% 93.22%, respectively.

  For relation extraction, Figure 2.16 shows how the performance of the state of art relation extractions models on SemEval-2010 Task 8 dataset are increased especially after using the neural networks in extracting the relations in year 2013. Examples of models that achieved the state-of-the-art by Zhou et al. [89] and Soares et al. [90]. Zhou et al. [89] achieved state-of-the-art in relation extraction from a plain text without the need to get high-level features from lexical resources such as WordNet or NLP systems like dependency parser and named

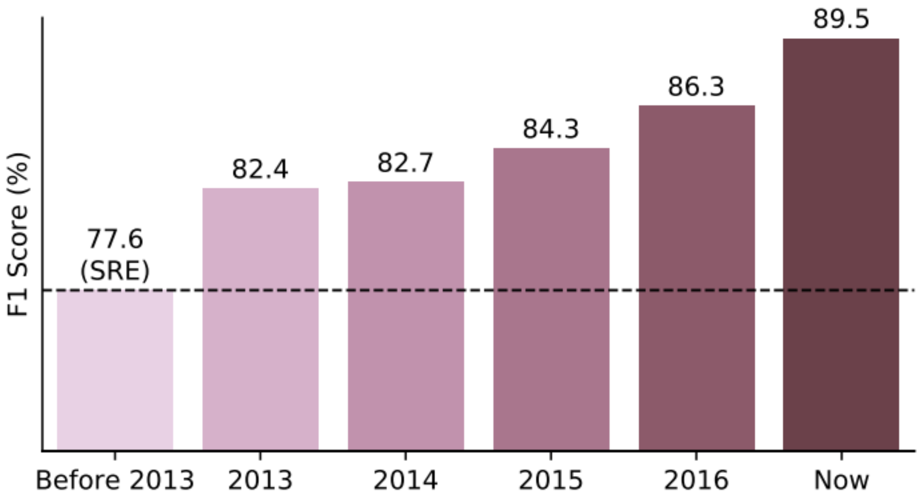


Figure 2.16. The F-score of the state-of-the-art relation extraction models increased over years on SemEval- 2010 Task 8 dataset. Now in the image is 2020 and image is taken from Han et al. [63].

entity recognizers (NER). They used Att-BiLSTM, on SemEval-2010 Task 8 dataset, to capture the most important semantic information in a sentence and achieved F-score of 84%. Soares et al. [90] introduced another state-of-the-art RE system using BERT. The learning method based on matching the blanks (MTB) that learns relation representations from entity resolution annotations without any further tuning for relation extraction. It achieved F-score of 89.5% on SemEval-2010 Task 8 dataset. Cohen et al. [91] reported the current state-of-the-art, on SemEval-2010 Task 8 dataset, which achieved 91.9% using BERT model. They used supervised span-prediction based system, similar to question answering (QA), for relation classification (RC). RC task is classifying the relation of two given entities into one of the predefined relations, or to a null "no-relation" class if there is no match to one of the relations. It achieved significantly better then the standard classification based that uses a single embedding to represent the relation between a pair of entities.

Example of text mining systems

GATE [92] is an open source framework that provides a graphical interface as well as Java plug-ins for text mining techniques. It includes embedded solutions for text pre-processing, information extraction and evaluation. For pre-processing and simple name entity recognition (for known person, places and time), it includes ANNIE. The ANNIE pipeline pre-processes the text by the tokenizer, sentence splitter and POS Tagger. It uses a gazetteer as a dictionary of domain terms and uses semantic rules in the ANNIE transducer to match pre-processed text with the gazetteer terms to identify target entities. Orthographic coreference (OrthoMatcher) does not identify new entities, but it assigns a classification to a proper name that is annotated with an unknown type. GATE reuses the existing solution instead of implementing a new one. E.g. for biomedical NER, they integrated ABNER [93] and GENIA tagger [94]. Jape rules in GATE are one of the important features that facilitate creating and annotating rules with Java programming.

GENIA project[7] annotated GENIA Corpus at different levels. GENIA Corpus includes 2000 MEDLINE abstracts that were retrieved by the MeSH terms "human", "blood cells", and "transcription factors". The annotation format is XML and the annotations can be found added for Part-of-speech, molecular biology terms such as proteins, genes and cell types taking into account the coreference annotation, the protein-protein interactions or gene regulatory relations, bio-events that describe the changes and molecular biology processes that happen to annotated biomedical terms.

2.5.3 Discourse analysis

Discourse is defined as the "use of spoken or written language in a social context" [95].

Discourse analysis was developed in the 1970s and it differs from common language analysis that isolates the text from surrounding conditions by concerning itself "with the use of language in a running discourse, continued over a sequence of sentences, and involving the interaction of speaker (or writer) and auditor (or reader) in a specific situational context, and within a framework of social and cultural conventions." [96]

The analysis of discourse involves interpreting the meaning behind its units (document, paragraph, sentences, phrases, clauses or words). The importance of discourse, as the top of NLP processes, is that it enables problems to be solved that underlying techniques cannot solve. Discourse has four structures that differ in complexity, coverage and symmetry [97]. Complexity refers to the level of extraction that is required from the text - are segments enough, do we need to extract chunks or do we need to use a parser? Coverage is how much the extracted discourse spans the text. Symmetry or asymmetry refers to whether the discourse units have the same level of importance or not.

[7] http://www.geniaproject.org/

- **Topic**

  Topic structure defines the main entities in a text and how they have been described. It is commonly seen in textbooks and encyclopaedias where one subject can be identified, and then subsequently related information can be linked to it [97]. It is used to explain a topic or an area of interest e.g. if a book talks about computer hardware, the topic structure then will include the definition, a list of these hardware components with their functions, and whether they have synonyms. Topic structure partly covers the text with segments that contain lexicon related to the topic. This can be clearly identified if the same word is repeated in the text or by looking for semantically related words. The semantic-related words can be hypernyms (more general terms), hyponyms (more specific terms), synonyms (terms with a similar sense), and meronyms (terms that refer to a part of a given whole).

- **Functions**

  Functions structure defines the elements by the functions and roles they play in the text [97]. One of the significant functional discourse examples is extracting the functions of text in scientific papers. In this case, the sentences or clauses are classified as zones, where each zone represents a high-level rhetorical status such as background, hypothesis, observation, conclusion, etc. As topical structures, the functional segments partially cover the text. The lexicon can be a list of cues with possible synonyms, e.g. to segment the results section, the words that are looked for are result, finds, conclude, etc. Functional structure is helpful to determine which sentence or paragraph belongs to which section especially in the absence of xml tags that are utilised by some journals.

  Functional discourse can also be beneficial for defining the advantage and disadvantage of using proposed methodologies, or in assessing the quality of writing where a piece of work is tested regarding specific requirements, such as a thesis statement in the introduction part of a document [98]. Another application is the comparison between an author's own work and the methods of others included in the publication [99], for example 20 online journal articles annotated manually. The annotation schema are presented as zones that are grouped into three. The first group includes the main classes: background, problem-setting, and personal work. The personal work, which refers to the authors work, contains methods, results, insight, implications and miscellaneous. The second group is the comparison between the author's work and other work: connections and differences. The last group includes the outline, which includes the summary of the work presented in the article.

The ART Corpus [100] contains 225 physical chemistry and biochemistry articles that are annotated by the functionality of each sentence (e.g. motivation, goal, method, experiment, observation, conclusion, hypothesis, etc.).

- **Event**

  Event discourse, or eventualities, covers the description of events that happen to an item and cause a change over time and/or place [97]. It is commonly used in narrative and reporting news and accidents identifying the events that can make a change in the item [101] e.g. "*x arrested*", *arrested* is an event where the item x is changed by. The order of events is important and for that the identification of temporal relations in an event structure is crucial [102].

  Chambers and Jurafsky [102] demonstrated that the event structure can be described with templates that represent that event type and semantic roles of surrounding components. They suggested a system that learned to build and fill a template for a domain that has no templates, by learning from a domain that has. They suggested three steps in order to build such a template schema. The first step involves identifying event types and event words e.g. the event type *attack* has event words like *kill, shoot down, down, etc.* The event words can be in the form of a verb or noun. To know if two words belong to the same group, the cosine similarity function is used to identify the distance between them. The second step is retrieving the documents that contain such kinds of events and infer the semantic roles that surround the events. The third step is to fill the created templates with information extraction tools.

  There are a number of corpora that are event annotated based on domain. For example, ACE 2005 [103] contains news articles that are annotated with 33 different event types. In addition, there are guidelines of how to annotate events, e.g. for timely events, there is a guideline written by Saurı et al. [104].

  Others examine the meta-knowledge annotation schemas (see Figure 2.17). In biomedical texts, there are some works in events like in bio-events which can be described by six meta-knowledge categories. Nawaz et al. [105] focused on two aspects of meta-knowledge; the knowledge type that includes the functional roles such as fact, observation, investigation, introduction, methods, etc. and on certainty levels that handle how much the mentioned information is certain (L1 for neutral opinion, L2 for high confidence and L3 for low confidence) e.g. the word *"may"* indicates a low level of certainty. They identified the changes of knowledge type inside a manuscript's sections e.g. the mention of a method is sometimes followed by observation, which might be followed by analysis.

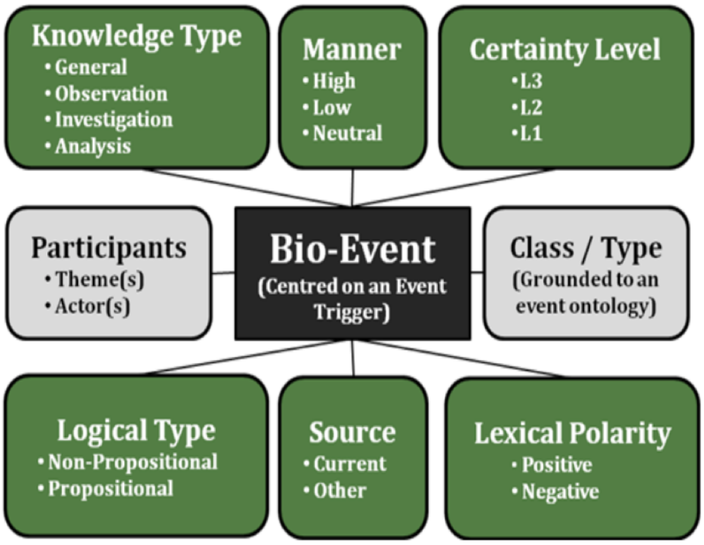


Figure 2.17. Meta-knowledge annotation scheme for bio-event. Image is taken from Nawaz et al. [105].

Table 2.1. RST relations examples from [106].

Relation Name	**Nucleus**	**Satellite**
Background	text whose understanding is being facilitated	text for facilitating understanding
Elaboration	basic information	additional information
Preparation	text to be presented	text which prepares the reader to expect and interpret the text to be presented.
Condition	action or situation whose occurrence results from the occurrence of the conditioning situation	conditioning situation
Purpose	an intended situation	the intent behind the situation
Restatement	a situation	a re-expression of the situation
Solutionhood	a situation or method supporting full or partial satisfaction of the need	a question, request, problem, or other expressed need
Summary	text	a short summary of that text

- **Relational Discourse**

  Also known as coherence relations discourse or rhetorical relations discourse. There are a number of discourse schemas in this area and this report will provide three of them.

  - **Rhetorical Structure Theory** RST [108] maintains the relationship between two text spans. It fully analyses the text to build a tree of units that are linked by relations. It is asymmetry discourse that gives some units (nuclei) more weight than other units (satellites). This is because the nuclei contain the main information and satellites contain additional information.

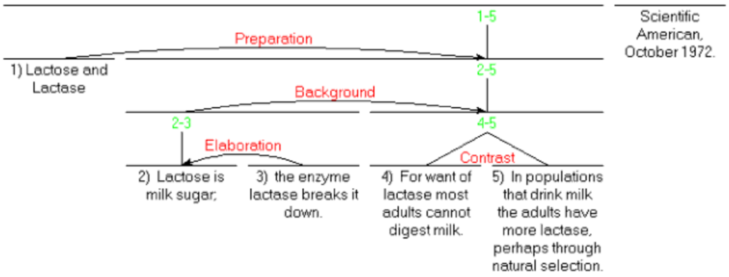
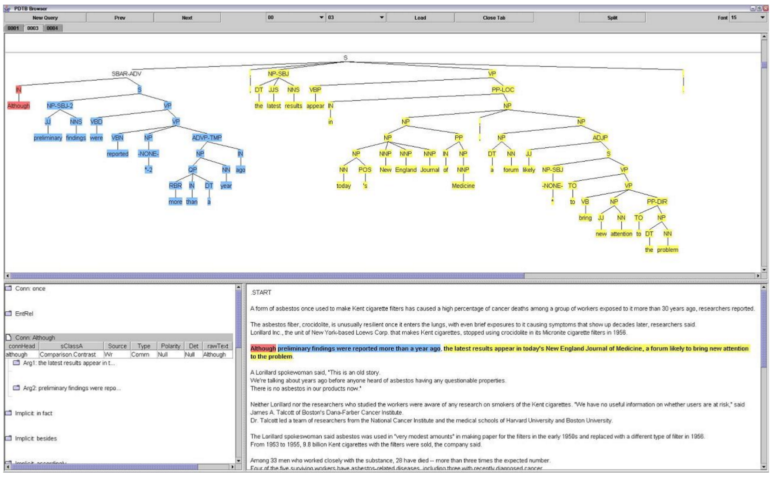

1-5
Scientific American, October 1972.
Preparation
1) Lactose and Lactase
2-5
Background
2-3
4-5
Elaboration
Contrast
2) Lactose is milk sugar;
3) the enzyme lactase breaks it down.
4) For want of lactase most adults cannot digest milk.
5) In populations that drink milk the adults have more lactase, perhaps through natural selection.

Figure 2.18. The abstract extracted from Scientific American magazine “*Lactose and Lactase. Lactose is milk sugar, the enzyme lactase breaks it down. For want of lactase most adults cannot digest milk. In populations that drink milk the adults have more lactase, perhaps through natural selection.*” is analysed by RST. The first sentence is a preparation for the remaining sentences. The second sentence is a definition of the two terms which in turn play a role as a background for the remaining two sentences. Image is taken from Mann and Taboada [106].



connHead	sClassA	Source	Type	Polarity	Det	rawText
although	Comparison.Contrast	Wr	Comm	Null	Null	Although

.START

A form of asbestos once used to make Kent cigarette filters has caused a high percentage of cancer deaths among a group of workers exposed to it more than 30 years ago, researchers reported.

The asbestos fiber, crocidolite, is unusually resilient once it enters the lungs, with even brief exposures to it causing symptoms that show up decades later, researchers said.
Lorillard Inc., the unit of New York-based Loews Corp. that makes Kent cigarettes, stopped using crocidolite in its Micronite cigarette filters in 1956.

Although preliminary findings were reported more than a year ago, the latest results appear in today's New England Journal of Medicine, a forum likely to bring new attention to the problem.

A Lorillard spokewoman said, "This is an old story.
We're talking about years ago before anyone heard of asbestos having any questionable properties.
There is no asbestos in our products now."

Neither Lorillard nor the researchers who studied the workers were aware of any research on smokers of the Kent cigarettes. "We have no useful information on whether users are at risk," said James A. Talcott of Boston's Dana-Farber Cancer Institute.
Dr. Talcott led a team of researchers from the National Cancer Institute and the medical schools of Harvard University and Boston University.

The Lorillard spokeswoman said asbestos was used in "very modest amounts" in making paper for the filters in the early 1950s and replaced with a different type of filter in 1956.
From 1953 to 1955, 9.8 billion Kent cigarettes with the filters were sold, the company said.

Among 33 men who worked closely with the substance, 28 have died -- more than three times the expected number.
Four of the five surviving workers have asbestos-related diseases, including three with recently diagnosed cancer.

Figure 2.19. The parse tree of sentence “*Although preliminary findings were reported more than a year ago, the latest results appear in today’s New England Journal of Medicine, a forum likely to bring new attention to the problem.*” is shown on the top. The explicit connective *although* gives a contrast relation between the two arguments highlighted by yellow and blue. Image is taken from Pennsylvania [107].

It is important e.g. in text summarisation and generation. The relations that are defined under this type of discourse differ slightly between the available annotation guidelines, although they share the same broad categories. Examples of relations used are elaboration, preparation, condition, etc. Table 2.1 shows some examples of RST relations. Figure 2.18 shows a paragraph discoursed by RST.

– **Penn Discourse TreeBank** PDTB [109] contains open-domain articles collected from the Wall Street Journal. The PDTB corpus maintains discourse relations between two arguments that are partially covered the text and each argument has the same weight of importance. The relations can be explicit and implicit based on the presence of the discourse connectives. The relation could be temporal, comparison, causal or expansion. Figure

Table 2.2. Contingency Table

	Correct (C)	**Not Correct (NC)**
Retrieved (R)	True Positive (TP)	False Positive (FP)
Not Retrieved (NR)	False Negative (FN)	True Negative (TN)

2.19 shows a sentence discoursed by PDTB.

– **Biomedical Discourse Relation Bank**

BioRDB [110] follows the PDTB annotation schema but is used in biomedical text. The Bio-DRB corpus contains 24 biomedical articles from the GENIA corpus. It is manually annotated by explicit and implicit discourse relations with 80% inter-annotator agreement. The BioRDB team developed a machine-learning classifier to identify the explicit connectives with a 0.89 F score. The interesting conclusion they discovered was that training the classifier in biomedical text and testing it in biomedical text produced a better result than when training it on a general domain corpus and then testing it on a biomedical text. Another interesting result is that more than 50% of the annotated temporal relations occur in the Methods and Materials section, which means that temporal relations deserve further attention and study.

2.6 Evaluation

2.6.1 Precision, recall, F-score

Rijsbergen [111] stated that three important questions should be addressed at the time of evaluation: why do we need to evaluate?, what are the things that we are going to evaluate?, and how we are going to evaluate them? The evaluation process is important for measuring the benefits or cost caused by a proposed solution, or to know if it performs better or worse than a standard. In order to do such an evaluation, certain measures should be calculated and analysed.

The two most frequent measures for information retrieval and extraction are precision and recall Rijsbergen [111, chap. 7]. Their elements can be defined by the contingency table.

The correct column (C) defines all relevant items that the solution should address that include true positive (TP) and false negative (FN). TP is the correct retrieved items. FN is the correct items that are not retrieved by the system or are in another way incorrectly missing items. The incorrect column (NC) defines the things that are not relevant and should not be considered as a part of the solution - false positive (FP) and true negative (TN). However, according to the system behaviour, it can retrieve FP; the items that are not correct or relevant to the answer, or it can correctly avoid extracting non-relevant answers and that is called TN. The retrieve (R) indicates the real system output and shows all the terms retrieved (TP and FP). The row (NR) shows the items that are not retrieved (FN and TN). The system result should be ideally compared to a "gold standard" that contains the correct terms. Precision declares the percentage of the retrieved items that are correct and recall declares the percentage of correct items that are retrieved.

$$precision = \frac{TP}{R}$$

$$recall = \frac{TP}{C}$$

There is a trade-off between precision and recall, e.g. to increase the recall, some of the retrieved items may be incorrect which will affect precision. This introduces a combined measure called the F-measure, which is a weighted harmonic which means that it gives weight to the most important measures [112].

$$F = \frac{1}{\alpha\frac{1}{Precision} + (1-\alpha)\frac{1}{Recall}} = \frac{(\beta^2+1)Precision \times Recall}{\beta^2 \times Precision + Recall}$$

where

$$\beta^2 = \frac{\alpha}{1-\alpha}$$

Values of $\beta < 1$ emphasize precision, while values of $\beta > 1$ emphasize recall. F measure is the one which gives equivalent importance to both precision and recall where $\alpha = 0.5$ i.e. $\beta = 1$

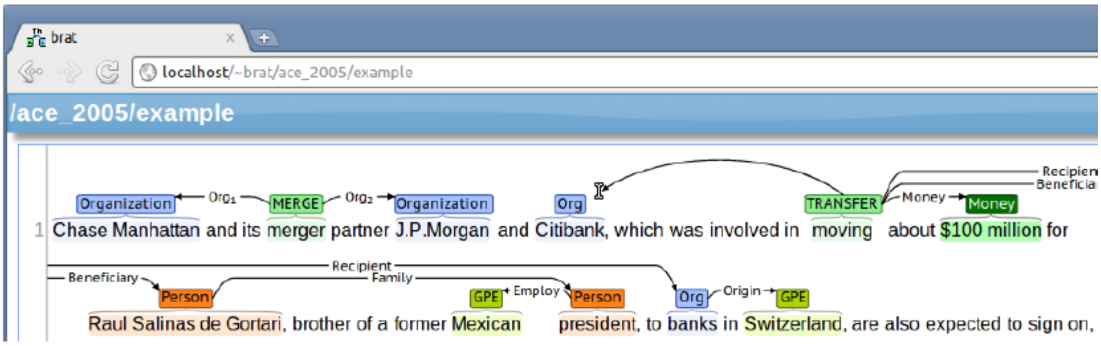
$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

2.7 Workflow representation, exploration and saving

There are three ways to save, represent and explore the data events in workflows:

- BRAT format

  Brat format [113] provides a medium for storing, presenting and processing the



1 Chase Manhattan and its merger partner J.P.Morgan and Citibank, which was involved in moving about $100 million for Raul Salinas de Gortari, brother of a former Mexican president, to banks in Switzerland, are also expected to sign on,

Figure 2.20. An example of a text file annotated with BRAT and viewed in user-interface. Image is taken from Stenetorp et al. [113].

annotation files in order to explore them and construct further information. The annotation file includes the annotation of entities, relations, normalisation and events. An example of BRAT annotation is in Figure 2.20 .

The files in BRAT annotation format can be viewed online through the brat rapid annotation tool[8]. The colourful user interface enables a viewing and editing platform for the annotations. This service can be installed in local devices as well.

There are configuration files that define the annotation structure and format. The annotation structure includes the defined names of all annotation types, the structure of the annotation types and any restrictions. The links to other websites can be defined in case an entity is going to be linked to its definition or normalisation. The user can define the viewing format for each defined annotation type as well.

The annotation file is stored with the same folder and same name as the original text file but with the .ann suffix. The file contains a line for each annotation. It must start with an ID of a single upper-case character identifying the annotation type and a number. The characters are: T for entities annotation, R for relations, N for normalisation and E for events. It is followed by the defined name of the annotation type (that previously defined and stored in a configuration file).

The annotation files can be processed with programming languages such as Java and Python. Example of Java library is https://github.com/yfpeng/pengyifan-brat.

- Comma-Separated Values (CSV)

  CSV are widely used and easily imported and processed by multiple environments, including programming languages. There are different format for how the data could be stored and interpreted. There could be an optional header line in

[8] `http://brat.nlplab.org`

the first line of the file with the same format as normal record lines. The records could be double quoted. The data can be viewed as tables which makes for easier capturing and understanding of the information.

- Directed network graph

  A directed network graph has nodes that are connected by arrows that show the flow direction between the nodes. Gephi [114] is software to represent graphs. It accepted the information of the nodes and edges in spreadsheets and converts them to graphs. The first spreadsheet contains the id of nodes' labels that are going to be represented in the graph. The second spreadsheet defines the relations between the nodes. Each line contains two ids from the first file to express a from/to relation between these two nodes. Additional information can be added to describe the name or the weight of the relationship. Gephi provides an interactive environment to get insights from the graphs. It gives the ability of formatting, partitioning, and calculating the value of some statistical matrices.

2.8 Related work in the literature

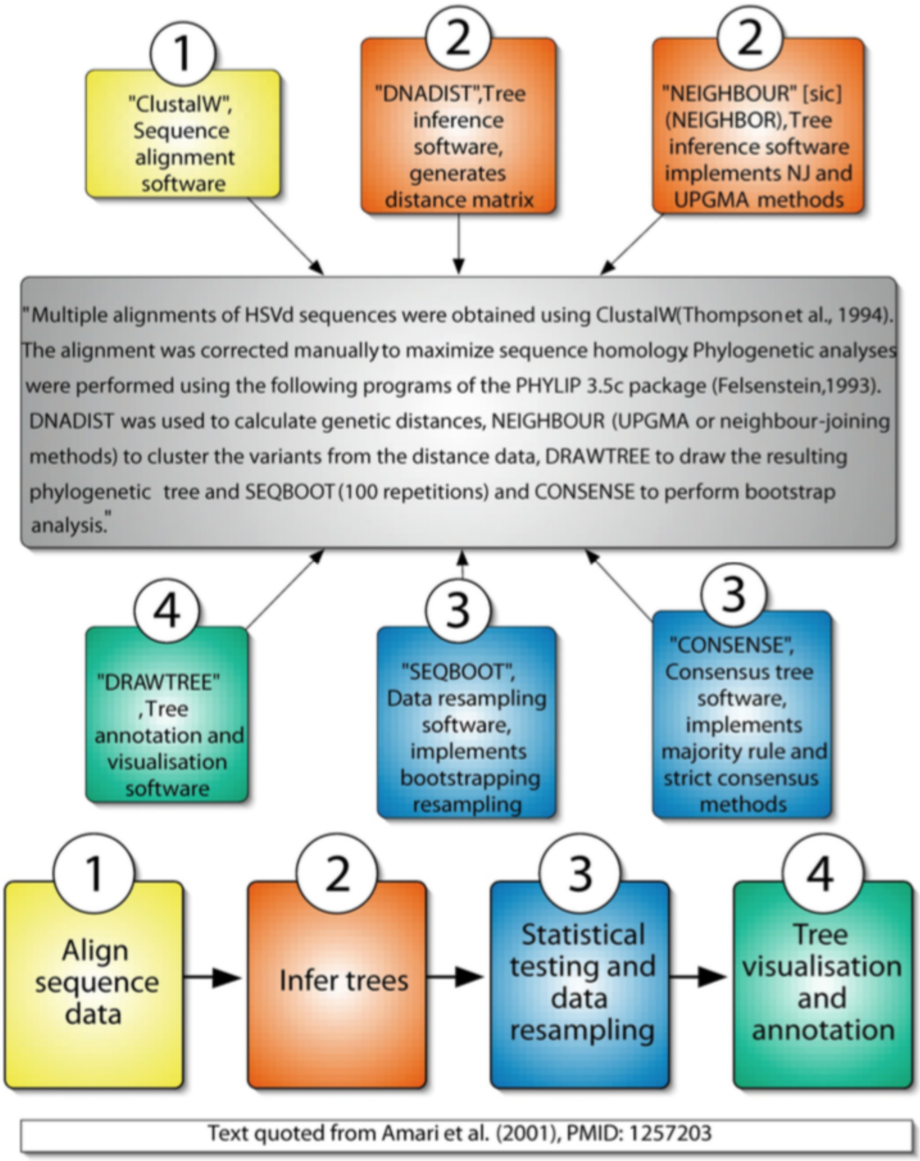
Eales et al. [15] defined the best practice methods of the phylogenetic field. They first identified a method model for molecular phylogenetics. This model contains four main steps: sequence alignment; tree inference; statistical testing and data resampling; and finally tree visualisation and annotation. Then, from a journals' articles that implemented phylogenetic methods, they extracted terms, relating to the above mentioned four stages, to model the individual methods, or phylogenetic protocols that were used. The extraction process mainly depends on a controlled vocabulary, populated manually by important phylogenetic terms and software names. The matching between extracted results and corresponding occurrence in the articles gave an F-score of 78.7%. They found that the total of the most used protocols was 10 protocols out of 847 uniquely extracted protocols. They studied the usage of the most used protocols by sub-discipline, authors, and time span. Figure 2.21 shows the connected model and terms extracted from the text in order to form the experiments used by mapping to a model protocol.

Duck et al. [115] extracted the patterns of software and databases used in bioinformatics literature. They built a network with directed edges between databases and software and another one that linked software only. The dataset was a method section of PubMed articles that contained the MeSH term "bioinformatics" and between the years 2004-2013. They extracted the named database and software by using their published named entity recognition tool bioNerDS [71]. Then they filtered the results to include only resources that occurred more than once to focus on commonly used resources. The patterns formed were based on the co-occurrence in the literature, the co-occurrence of the database and software, and the co-occurrence of the database. The binominal test was used to include only links that were above confidence levels to ensure only links that were common were included. The software links reported that common tasks (with % confidence level) were present in the sequence search followed by sequence alignment. For database-software links, the GO database was the most commonly used. This is interesting as it was used as a 'data sink' where software deposit the data in instead of retrieving it. The automatic extraction was compared to the manually created steps by the previously mentioned study [15] and it was found that the resources extracted were compatible with the manual protocol (see Figure 2.22). They extracted the patterns that report the commonly used resources in the literature, which gave a view on what is currently used and the new resources that are turning out to be popular. Individual methods were not extracted and the order of extracting depended only on the occurrence of the text. Still, extraction of specific data and how they are connected to other method components for a given research article is not addressed in previous work.

Additionally, Kovačević et al. [116] extracted the Automatic Term Recognition (ATR) methodologies. They first analysed the text to annotate the statements that involved a mention of methodology. Then the sentences were segmented and mapped to one of the four functional categories (task, method, implementation and resource/features). The segments were detected and classified automatically using a Conditional Random Fields (CRF) model for each category. Their approach produced a good performance; F-scores at document level were 72% for Tasks, 60% for Method mentions, 74% for the Resources/Features and 78% for the Implementation mentions. The interesting findings were that the most mentioned methods in ATR were part-of-speech tagging, morphological analysis and syntactic parsing, while the most common tasks were term recognition, classification, pattern matching and similarity calculation.

Kappeler et al. [117] extracted methods being provided for each paper from biomedical science research. The domain was scoped to involve empirical procedures that revealed protein interactions. Pattern matching of lexical clues was the basis of the procedure adopted in order to identify empirical methods with the assistance of statistical methods. The results showed that 67.3% were composed of the top five most commonly used methods. Thus, for manual rules design, they concentrated on these five methods. They were able to identify the experimental methods of protein interaction in publications with an F-score of 45.9%.

Recently there are few detests that handles scientific method entities and relations.



"Multiple alignments of HSVd sequences were obtained using ClustalW(Thompsonet al., 1994). The alignment was corrected manually to maximize sequence homology. Phylogenetic analyses were performed using the following programs of the PHYLIP 3.5c package (Felsenstein,1993). DNADIST was used to calculate genetic distances, NEIGHBOUR (UPGMA or neighbour-joining methods) to cluster the variants from the distance data, DRAWTREE to draw the resulting phylogenetic tree and SEQBOOT (100 repetitions) and CONSENSE to perform bootstrap analysis."

1
"ClustalW", Sequence alignment software
2
"DNADIST",Tree inference software, generates distance matrix
2
"NEIGHBOUR" [sic] (NEIGHBOR),Tree inference software implements NJ and UPGMA methods
4
"DRAWTREE" ,Tree annotation and visualisation software
3
"SEQBOOT", Data resampling software, implements bootstrapping resampling
3
"CONSENSE", Consensus tree software, implements majority rule and strict consensus methods
1
Align sequence data
2
Infer trees
3
Statistical testing and data resampling
4
Tree visualisation and annotation
Text quoted from Amari et al. (2001), PMID: 1257203

Figure 2.21. Extracting phylogenetic models that correspond to a protocol represented as a sequence from 1 to 4 from a text mentioned in (Amari paper). Image is taken from Eales et al. [15].

SienceIE [119] is a corpus for the SEMEval 2017 task 10. It contains 500 scientific articles of the Computer Science, Material Sciences and Physics domains. It includes three types of entities (called keyphrases): Tasks, Methods, and Materials and two relation types: hyponym-of and synonym-of. Augenstein et al. [119] and Luan et al. [120] introduced the state-of-the-art systems that extract entities and relations from this dataset.

SciERC [121] extended previous dataset, SienceIE, to include 500 annotated scientific abstracts, from AI domain, containing annotations for scientific entities, their relations, and coreference information between mentions. They defined six types for annotating scientific entities (Task, Method, Metric, Material, Other-ScientificTerm

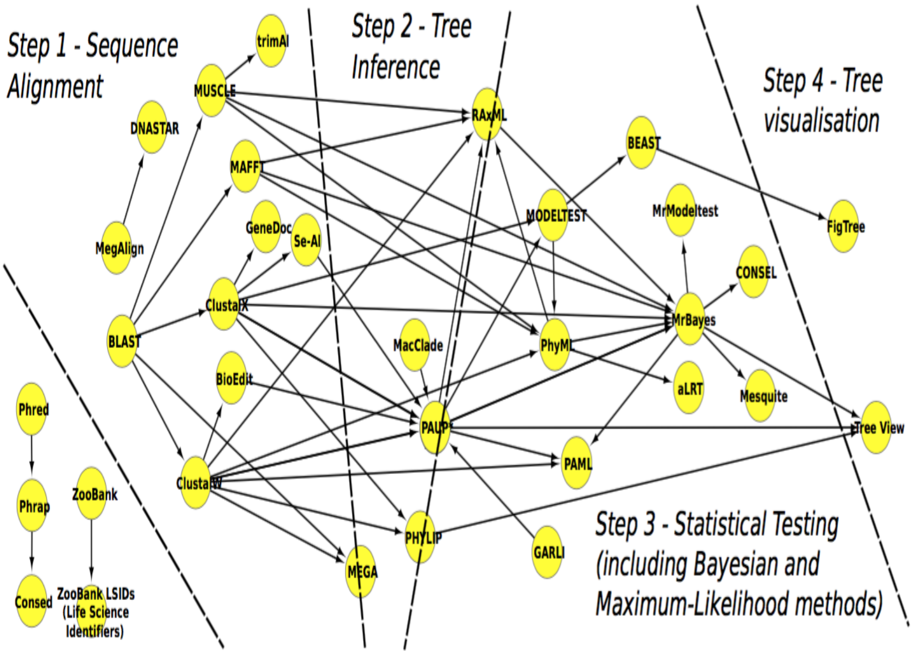
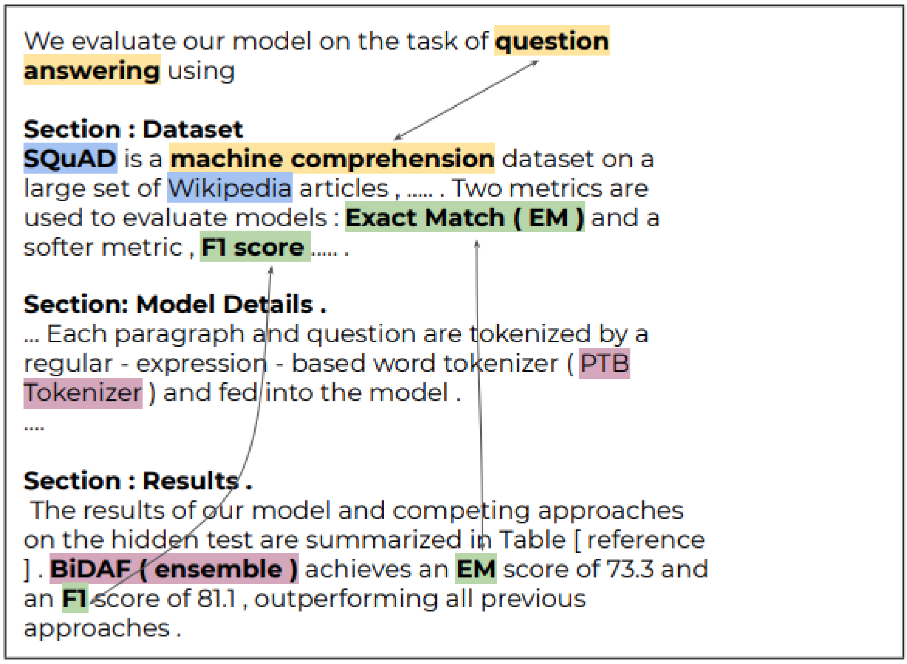


Figure 2.22. Pattern of software and database usage within phylogenetics papers extracted by [115]. It was consistent with the model provided by [15] in Figure 2.21. Image is taken from Duck et al. [115].



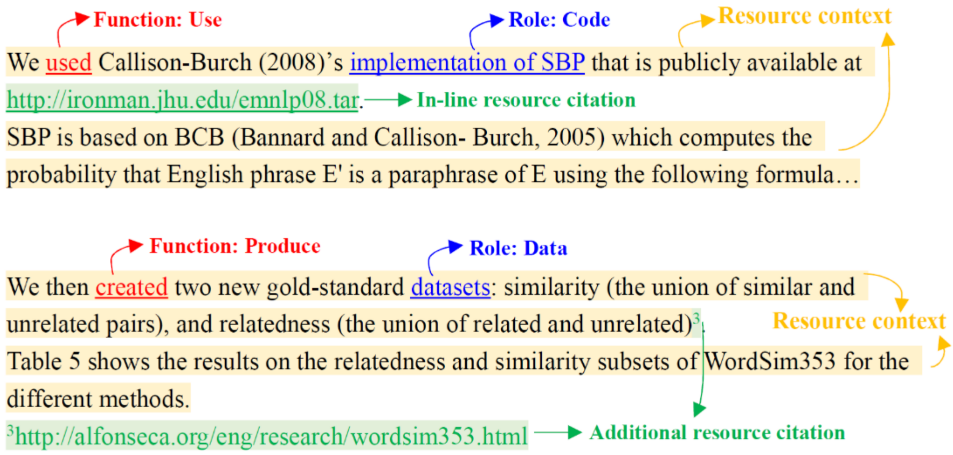
We evaluate our model on the task of **question answering** using

Section : Dataset
SQuAD is a **machine comprehension** dataset on a large set of Wikipedia articles , ... Two metrics are used to evaluate models : **Exact Match (EM)** and a softer metric , **F1 score** ..... .

Section: Model Details .
... Each paragraph and question are tokenized by a regular - expression - based word tokenizer (PTB Tokenizer) and fed into the model .
....

Section : Results .
The results of our model and competing approaches on the hidden test are summarized in Table [reference] . **BiDAF (ensemble)** achieves an **EM** score of 73.3 and an **F1** score of 81.1 , outperforming all previous approaches .

Figure 2.23. An example of SciREX annotation. Image is taken from Jain et al. [118].



We used Callison-Burch (2008)'s implementation of SBP that is publicly available at http://ironman.jhu.edu/emnlp08.tar. SBP is based on BCB (Bannard and Callison- Burch, 2005) which computes the probability that English phrase E' is a paraphrase of E using the following formula…

We then created two new gold-standard datasets: similarity (the union of similar and unrelated pairs), and relatedness (the union of related and unrelated)[3]. Table 5 shows the results on the relatedness and similarity subsets of WordSim353 for the different methods.

[3] http://alfonseca.org/eng/research/wordsim353.html

Figure 2.24. An example of SciRes annotation. Image is taken from Zhao et al. [16].

and Generic) and seven relation types (Compare, Part-of, Conjunction, Evaluate-for, Feature-of, Used-for, HyponymOf). They developed *SciIE*, BiLSTM model, to extract entities, relations and coreference resolution across sentences and achieved F-score of 64.2%, 39.3% and 48.2%, respectively.

SciREX [118] extended the previous dataset, SciERC, to include 438 fully annotated documents from ML articles available in Papers with Code website. It contains annotations for entity mentions (Dataset, Metric, Task, Method), coreferences, salient entities, and N-ary relation over the document (between clustered salient entities). They introduced an end-to-end neural model, using BERT-BiLSTM-CRF, to perform document level IE task. The model to outperform existing state-of-the-art models on subtasks and achieved F-score between 67%-27% for the end-to-end tasks. An example of a relation with extracted entities: *(Dataset: SQuAD, Metric: EM, Method: BiDAF, Task:machine comprehension)*. Figure 2.23 shows an example of the annotation.

SciRes [16] is a dataset of 3,088 manually annotated resource contexts, i.e. a sequence of words that appear around a particular citation. They used full articles from three sources: the ACL Anthology Reference Corpus (ARC), a corpus of scientific publications about computational linguistics; the NeurlPS Proceedings (NeurlPS), a corpus of conference proceedings about neural information; and the PubMed, an archive of biomedical and life sciences journal literature. They have focused on the resource citation hyperlinks mentioned in the scientific articles to extract the methods' components as (online-resource citation, resource role and resource function). They defined three general role types with 9 subtypes: Material (Data), Method (Tool, Code, Algorithm) and Supplement (Document, Website, Paper, License, Media). The function types are: Use, Produce, Introduce, Extend, Compare, and Other. Figure 2.24

shows an example of the annotation. They implemented a BERT-based classifier, *SciResCLF*, to classify the roles and functions of a resource based on the word sequences of resource contexts. Then *SciResREC*, another BERT-Based model, uses these classification labels as features for predicting the resource hyperlinks. SciResCLF classifier achieved an F-score between 90%-35% and SciResREC achieved an accuracy of 50% in predicting resources.

2.9 Conclusion

We have looked at different topics to build a knowledge of the available methodologies that could help in answering our research questions.

We investigated the NER techniques to help in annotating the terms that represent the key aspects of the work conducted in the papers. We also searched for available knowledge bases to normalise the annotated concepts to their classes. The possible ways of presenting and storing this information for further search and processing are also searched.

The previous research papers either focused on extracting methods of one topic or looking at some key mentions of the Methods section for the whole corpus. We now need to extract more key mentions from an individual paper and represent the flow of the work conducted.

Chapter 3

Exploring the Writing Patterns of the Methods Sections in the Microarray Analysis Literature using Discourse Analysis

Abstract

Background A Methods section is a crucial part of a research article in terms of understanding and validating the work conducted. In order to handle method descriptions computationally, it is of interest to understand how the Methods section is written, find any patterns in how methods are written and whether they are consistent with general writing guidelines for methods. We are interested in studying the writing style in terms of the sentences' roles, or functional discourse. For example, are the sentences concerned with background information, explaining the experiment conducted, the operations of the method or showing the result obtained? **Objectives** We want to know the answers to the following questions: What is the size and shape of a Methods section? What are the discourse styles of Methods sections? Does discourse style change over time and publication venue? Can we use functional discourse annotations to focus on the parts of a Methods section that talk only of methods?

Methods In this work, the writing style of the Methods section in a corpus of 13,559 papers from 1999 to 2017 from the microarray analysis literature are analysed in terms of functional discourse using the SAPIENTA tool. The functional discourse patterns are produced across the whole corpus, across time and for publication venues.

Results The Methods section mainly contains the sentences describing experiments, methods or models. Since there is no restriction on the articles' length, the size of the Methods section varies. We found that 75% of the articles in the corpus have fewer than 70 sentences in their methods section. Most of the patterns obtained contain methods and experiments but of different lengths, and few of them express the outlier behaviours of having additional functions. The functional discourse patterns varied over the time and different journals.

Conclusions We demonstrate the feasibility of identifying writing patterns over the Methods section. Our results help to investigate the writing structure patterns, the probability of having these patterns, or a sequence of functions and linked those to the journals. We speculate that discourse analysis will be a useful methodology in computationally manipulating descriptions of methods in text.

3.1 Introduction

Scientific research is typically published in the form of an article structured as an Introduction, Materials and Methods, Results and Conclusion [122]. The number of scientific publications has increased dramatically over recent years [1]. Consequently, it is broadly accepted that scientists cannot keep up-to-date with the plethora of new findings, even within a sub-field of a discipline [123]. This difficulty extends to methods; there are many methods and scientists need to know about new methods and they need to know the best method to use in a given situation.

Methods are typically described with prose in scientific publications. The description illustrates how the work is conducted, what components are involved and why they were chosen for use [7]. There is a considerable body of work showing that methods are poorly described in the scientific literature, in terms of incomplete and non-transparent reporting of how trials were designed and conducted [124, 125, 126]. This results in findings not being able to be reproduced and results being unreliably compared across findings [126]. One consequence of not being able to reproduce methods is a poorer dissemination of knowledge about methods that can be applied in studies and the consequent diminution of the efficiency and effectiveness of science.

Given the number of publications, representing the current prose descriptions of methodologies in a structured, searchable representation is essential for providing information about which methods are available for a given task. This will enhance the ability of researchers to keep track of existing and recently published methods. It will also allow a community to assess trends in the uptake and deprecation of a particular method.

Thus, our broad aim is to extract a structured representation of computational methods from the biology literature. We have previously developed bioNerDS [71], a named entity recogniser for bioinformatics software and databases in the literature [127]. We have also used results of a survey of the literature using bioNerDS to show the networks of tools and databases used at the population level for computational biology [11]. The next step is to organise the tools and data identified in a method workflow for an individual scientific article.

A prerequisite for tackling this challenge is to understand how a Methods section is written. That is, what is the articulation or organisational presentation of methods in semi-structured or unstructured prose? If we have an understanding of the patterns of how prose describing methods is organised, then we can direct our text analytics methods to recover the data inputs and outputs, the software and the steps that embody the method informed by those patterns.

In this study, we seek to understand how Method sections are written by investigating the discourse patterns of method descriptions. Discourse analysis refers to the "study of the organisation of language above the sentence or above the clause" [128]. The aim of studying discourse is to determine the patterns of the important discourse elements within the text, such as clarifying an aim, explaining a method or justifying a result.

Multiple studies have been conducted on identifying functional discourse in scientific literature.

- Tuefel et al. [129] defined a rhetorical discourse annotation schema known as argumentative zoning (AZ). It contains seven concepts (`Aim`, `Textual`, `Own`, `Background`, `Contrast`, `Basis`, `Other`) and is concerned more in classifying whether the mentioned scientific concepts belong to the paper's author or are cited from others' work.
- Guo and Korhonen developed weakly supervised [130] and then unsupervised classifiers [131] to extract argumentative zoning in abstracts of biomedical articles. They extracted six functional concepts (`Background`, `Objective`, `Method`, `Result`, `Conclusion`, and `Related work`). These categories were originally formed and illustrated by Mizuta [99].
- Liakata et al. [132] developed an automatic annotation tool to extract the core scientific concepts (CoreSC) in life science articles. The eleven core concepts (`Hypothesis`, `Motivation`, `Goal`, `Object`, `Background`, `Method`, `Experiment`, `Model`, `Observation`, `Result` and `Conclusion`) are described in [133]. Short definitions for these functions are listed in Table 5.2.

While AZ focused on identifying the author's own text and cited text, CoreSC is more concerned with the ultimate role of a sentence in the text [134].

Example 1 (PMC2582621)
The Affymetrix Latin Square spike-in data U133A were retrieved from (12). They contained 14×3 hybridizations where spike-in targets were added at various concentrations from 0 pM to 512 pM. Probe information was obtained through (20), where only 30 of the 42 probesets were found. A total of 365 probes matched to target sequences. Among them, 10 probes with very low signal intensities (under 900 at highest target concentration) were removed. In total, 355 probes are included in this study.

An example of a functional discourse analysis using the general scientific concepts

Table 3.1. Functional discourse concepts definitions from [133].

Category	Function	Definition
Background	`Background`	Pre-existing facts and known information. It is not an old method.
Approach	`Method`	Pre-existing or new methods. It is a general description of the procedures conducted.
	`Model`	Theoretical model that contains equations, assumptions and the design of objects used.
	`Experiment`	Physical procedures that contains instruments, measurements, any special conditions and detailed steps of the processes.
Outcome	`Observation`	Simple, clear description of the experiment output. This includes the reference to the data presented in tables or figures.
	`Result`	Factual description of the experiment output. It is a further level of output description that could interpret, summarise, or compare a group of `Observation`.
	`Conclusion`	A general summary of what is found by the research and whether it supports or violates the research hypothesis.
Objective	`Hypothesis`	The research prediction that needs to be tested.
	`Goal`	The aim of the work conducted.
	`Motivation`	The importance of the work conducted.
	`Object`	The entity being studied, investigated, reported or discussed in the work conducted. It could be a physical or abstract entity.

Table 3.2. Functional concepts of sentences in Example 1.

Sentence Number	Function
1	`Method`
2	`Background`
3	`Method`
4	`Method`
5	`Observation`
6	`Result`

annotation guideline [133] on paragraphs of a Methods section in Example 1, is shown in Table 3.2.

The first sentence was a `Method` since it expressed retrieving data which is part of the procedures conducted. The second sentence reported factual information about the retrieved data, and hence it was classified as `Background`. The last sentence described the `Result` of the filtering performed in the previous sentences.

This study is undertaken in the context of computational biology, focusing on the microarray analysis literature. We chose this domain as it has had a sustained period of use over the last twenty years. Huge amount of biological data have been generated through research in microarray technology and methods have been continually developed to analyse these data [13, 14].

3.2 Materials and Methods

Our aim is to look for patterns of functional discourse elements in how methods are described in a corpus of computational biology papers. A workflow of our method is shown in Figure 3.1. An outline of our approach is as follows:

1. Collect a corpus of literature featuring microarray data analysis
2. Automatically annotate these papers with functional discourse tags
3. Extract the Methods section from these annotated papers
4. Analyse the discourse annotations for the patterns of discourse.

3.2.1 Corpus creation

PubMed central [135] was searched in April 2017 for articles that were published between 1^{st} January 1998 and 31^{st} March 2017 using the MeSH term *microarray analysis*. In addition, to obtain full text articles, we restricted the PubMed search to include only those in the PubMed Central Open Access subset. The query used was:

```
``microarray analysis''[MeSH Terms] AND (``1998/01/01''[PubDate] :
``2017/03/31''[PubDate]) AND ``open access''[filter]
```

Using the retrieved PMC IDs, the full text in XML format is downloaded via the Europe PMC's SOAP web [136].

PMC open access articles are archived by a common XML tag suite that is provided by NCBI/NLM [137]. Apart from archiving the actual article contents (text, graphs and tables), the article's metadata included in the PMC articles contain the year of publication, the journal in which the article is published, the type of the article and ten to fifteen associated MeSH terms for each article. To consider a document as a microarray analysis article, the document should be indexed by at least one of the following MeSH terms: Microarray Analysis (D046228), Oligonucleotide Array Sequence Analysis (D020411), Protein Array Analysis (D040081) and Tissue Array Analysis (D046888).

These four terms are the main MeSH terms for microarray analysis articles.

The article type can be a *research article*, where an original study is conducted, a *review article*, where a review of literature is reported, a *product review*, where software is reviewed or analysed, or it can be *other* if the article does not fit to any of the 24 types that are defined by NCBI/NLM [137].

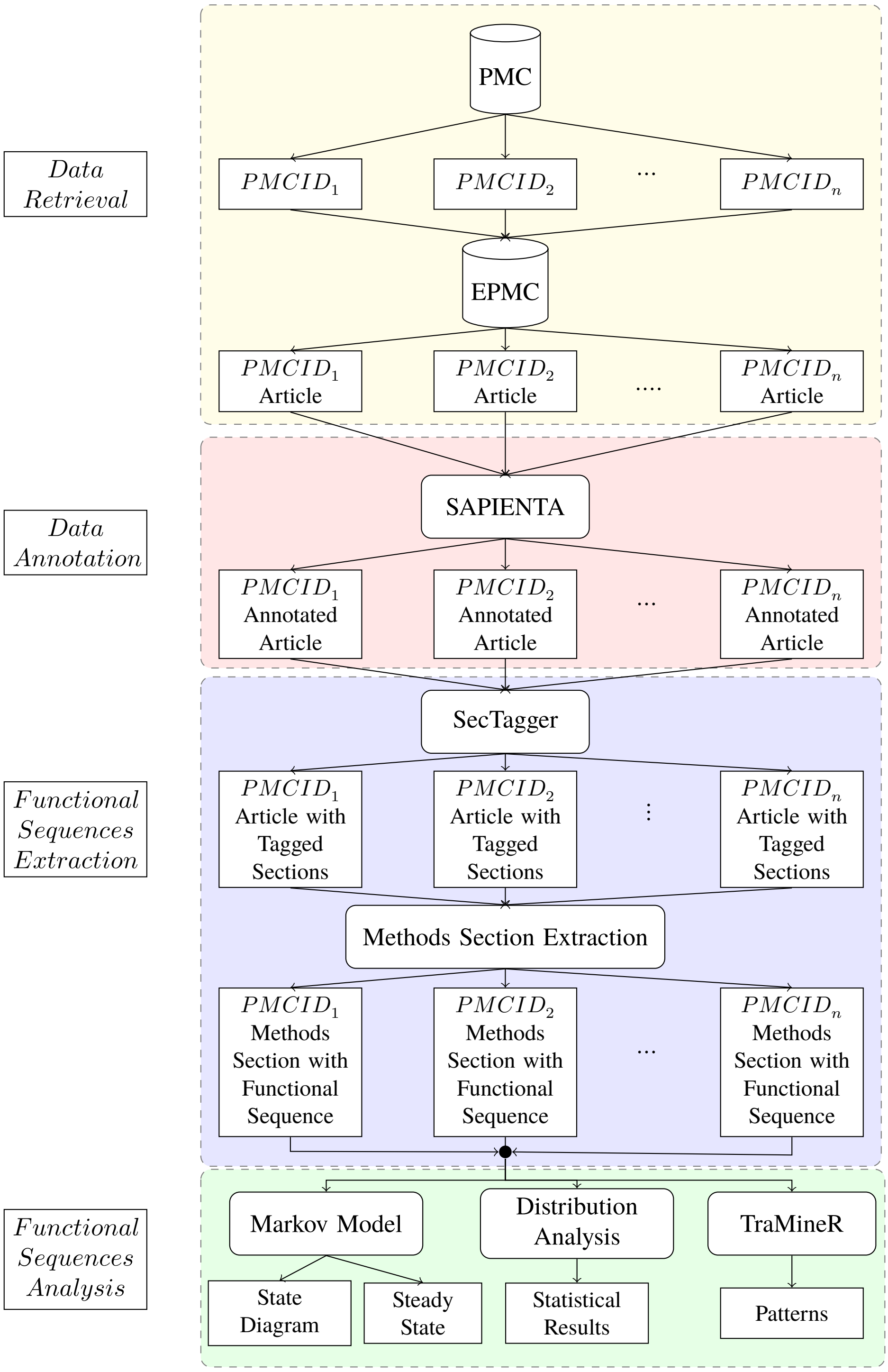

Data Retrieval
PMC
$PMCID_1$
$PMCID_2$
...
$PMCID_n$
EPMC
$PMCID_1$ Article
$PMCID_2$ Article
....
$PMCID_n$ Article
Data Annotation
SAPIENTA
$PMCID_1$ Annotated Article
$PMCID_2$ Annotated Article
...
$PMCID_n$ Annotated Article
Functional Sequences Extraction
SecTagger
$PMCID_1$ Article with Tagged Sections
$PMCID_2$ Article with Tagged Sections
⋮
$PMCID_n$ Article with Tagged Sections
Methods Section Extraction
$PMCID_1$ Methods Section with Functional Sequence
$PMCID_2$ Methods Section with Functional Sequence
...
$PMCID_n$ Methods Section with Functional Sequence
Functional Sequences Analysis
Markov Model
Distribution Analysis
TraMineR
State Diagram
Steady State
Statistical Results
Patterns

Figure 3.1. **System workflow.** The articles are first retrieved from Europe PubMed Central (EPMC) in XML formats based on PMC IDs. The articles are functionally discourse annotated by SAPIENTA then the paper sections are tagged by Section Tagger. The functional discourse sequences of the Methods section is extracted then analysed by the Markov model and TraMineR package.

3.2.2 Functional discourse annotation

We used SAPIENTA [132] for functional discourse annotation because it was the only online available tool for functional discourse at the time of doing the research. SAPIENTA automatically annotates the sentences with a scientific function with 51% average accuracy for all eleven CoreSC concepts on a dataset of 256 chemistry articles that include biochemistry and theoretical work [100]. The accuracy of identifying `Experiment`, `Method` and `Model` are 76%, 30% and 53% respectively. Merging the CoreSC concepts into the four broader categories gives a much better F-score as Background: 59%, Approach (`Experiment`, `Method` and `Model`): 72%, Outcome (`Observation`, `Result` and `Conclusion`): 81%, Objective (`Hypothesis`, `Motivation`, `Goal` and `Object`): 38%.

Guo et al. [138] proves that using automatic functional discourse classifiers, one of which is SAPIENTA, is useful in identifying the functional structure of the scientific documents of the biomedical field. It gets 81% F-score at the Abstract level on a dataset of 1000 cancer risk assessment abstracts.

We performed manual annotation for samples of 25 microarray analysis Methods sections and it gave an F-score of 62%. `Method`, `Experiment` and `Model` were extracted with F-scores of 74.26%, 44.62% and 60.44% respectively. Appendix D shows the details of evaluation scores.

3.2.3 Methods section extraction

Since the focus is on the Methods section, the paper sections are first tagged using the Section Tagger [139] then the Methods section is extracted. Section Tagger is a rule based tagger that extracts section categories of an article body with an F-score of 98.02% on a dataset of 100 full text articles randomly selected from PMC. We used the Section Tagger because it is the one used for the advanced search within Europe PMC. It extracts 17 predefined categories (Introduction - Background - Materials and Methods - Discussion - Conclusion - Future Work - Case Study - Acknowledgement and Funding-Author Contribution - Competing Interest - Supplementary Data - Abbreviations - Key words - References -Appendix - Figures - Tables - Other). For identifying the *Materials and Methods section*, the target section title should not contain the word ‘supplement’ and may contain words such as methods, materials, experimental, implementation, the study, theory…etc.

3.2.4 Functional discourse analysis

We wish to know how the discourse functions are distributed over the article sections, and over the Methods section:

1. Is there any relation between the functions to the article context?
2. What is the probability of having a specific function?
3. What are the possible patterns that could be extracted from the functional discourse sequences?

These questions could be answered by four types of analysis:

Functional distribution The frequency of the functions, the averages and correlations are calculated to show how the functions are distributed over the paper sections and over the sentences of the Methods section.

Sequential rules The general ordering of the functional discourse elements is calculated by sequential rules. A rule states that an item X is probably followed by another item $Y(X \rightarrow Y)$ based on the number of items that have this relation among the studied sequences. Most of the rules have a support measurement and confidence.

$$\text{Support}(X \longrightarrow Y) = \frac{\text{Number of transactions that contain } (X \rightarrow Y)}{\text{Total number of transactions}}$$

$$\text{Confidence}(X \longrightarrow Y) = \frac{\text{Number of transactions that contain } (X \rightarrow Y)}{\text{Total number of transactions that contain } X}$$

The top-k association rule methodology [140] was used in this paper to find the k most frequent rules and their minimum support values for a given confidence value. We used the tool implemented by Fournier-Viger et al. [141] to extract these rules.

The Functional discourse probability The annotated Methods sections are represented as a Markov model in order to analyse the behaviour of the functional discourse and then extract possible patterns. Markov models are effective in modelling categorical data sequences [142]. We chose District Time Markov Chains (DTMC) [143] to represent the functional discourse sequences. The state space is the eleven functions $S = \{Bac, Met, Mod, Exp, Mot, Goa, Hyp, Obj, Obs, Res, Con\}$ and the Markov chain is the sequence of discrete random variables $X_1, X_2, X_3, \dots, Xt, \dots X_n$, where X_t is the functional discourse state at sentence t ,and X_n is the last sentence's state in the observed sequence. The observed sequences need not have an equal length.

The Markov property assumes that the probability of the next state X_{t+1} depends solely on the current state X_t. This is in line with the assumption that within in the same paragraph the next sentence is affected by the previous one.

The probability distribution of transitions from one state to another can be represented in a transition matrix where each element gives the probability that starts with state i and ends with state j. It is obtained through the markovchain package [144] where the maximum likelihood estimator (MLE) method was used to calculate the conditional probability for p_{ij}.

$p_{ij}^{MLE} = (n_{ij}) / \sum(n_i)$

where n_{ij} is the number of sequences that has state i at sentence number t followed in the next sentence by state j. n_i is the number of sequences that is longer than t and has state i at sentence t.

From the probability transition matrix, we can deduce some of the states' behaviour and classifications, such as whether there is an absorbing state where reaching it means terminating the sequence, or if the studied states are irreducible and all the states can be followed by other states. It also can find if the state is periodic, which means the state reappears at regular intervals; otherwise it is aperiodic.

Also, having the probability transition matrix over a long period of time, the number of sentences in our case, gives a unique probability distribution on the state space S, known as steady state, where the behaviour of states remains in this distribution forever regardless of the starting state. The states should be aperiodic and irreducible to have a steady state.

Patterns We try first to find the patterns of Methods sections, regardless of year or journals. Then we find the patterns for each year and the top journals.

Every annotated Methods section is represented in a sequence of functions. One way of finding a representative for the sequences is to select one that is nearly similar to others sequences. The similarity is measured by the cost of converting a sequence to another. The conversion process is accomplished either by inserting the missed state or deleting an extra state or by substituting a state with the corresponding one. In this study, the cost is derived by the Long Common Sub-sequence method (LCS) [145]. It is a position-shifted method that looks for the longest common sub-sequence between

two sequences in order to convert one to another. The final conversion cost is calculated by the formula:
$|Seq_1| + |Seq_2| - 2 * Max(|LCS(Seq_1, Seq_2)|)$
This gave a nxn symmetric matrix that shows the distance between each sequence and another, n being the number of the sequences, which is the number of the articles in our study.

Since the distance between each two sequences is known, we looked for the ones that have the lowest distance. We used TraMineR R package [146] to extract the representative sequences. The package implemented by Gabadinho et al. [147] where the neighbourhood density method was used. It uses a neighbourhood radius to determine the number of neighbour sequences. The whole set of sequences are then put in descending order. The selected representatives should be far enough from each other by a threshold. This also ensures that outlier patterns could be found. We control the number of patterns by the ones which should at least have 88% of the sequences as their neighbours.

3.3 Results

The PMC search query returned 14,454 articles. A glance at Figure 3.2 reveals that the first article was published in 1999. The number of publications increased dramatically in the following ten years. The level of publication remained stable until 2016, which had a reduction of 68.67% in the number of the articles. The reason behind this drop is that, although most of the journals have full open access and the articles are immediately archived in PMC, the new articles are not necessarily indexed yet by MeSH terms.

As a general trend in Figure 3.3, PLoS ONE, BMC Genomics and BMC Bioinformatics together contribute 41.5% of the overall publications, where BMC Medical Genomics, BMC Genomics and BMC Bioinformatics contain the highest percentages of published articles related to microarray topics across all their publications (17%, 15% and 14% respectively). For more details see additional file 1.

The most common article types are *research articles* (94.42%) while the remaining types are distributed between *review articles, product review* and other article types (1.65%, 1.38% and 2.55% respectively). For more details see additional file 2.

The total number of associated MeSH terms is 9,732. The most frequent MeSH term over the four main MeSH terms is *Oligonucleotide Array Sequence Analysis* (77.22%). The other three MeSH terms are *Microarray Analysis*, *Tissue Array Analysis* and

Table 3.3. Minimum, maximum, mean and median of the Methods section length in terms of the number of sentences.

Min	Max	Mean	Median
1	388	57	52

Protein Array Analysis (10.99%,6.99% and 4.79% respectively). The majority of other associated MeSH terms are general terms such as *human ,animals, female, male and mice* (4.42%,2.64%,2.06%,1.64% and 1.36% respectively). For more details see Additional file 3.

The length of the Methods section varies between papers. Figure 3.4 shows the length distribution over the corpus. Table 3.3 shows the minimum, maximum, mean and median length of the Methods section in terms of the number of sentences.

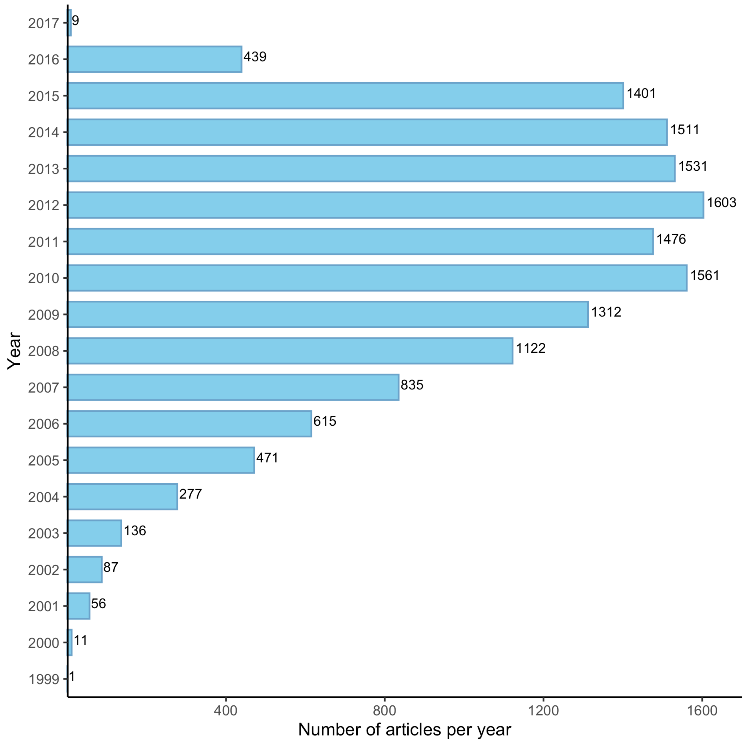


Figure 3.2. The number of articles published per year over the corpus.

Using the Section Tagger [139], 94% of the articles clearly express the Methods section within the paper sections (total articles 13,589). For the remaining 865 articles, 442 are not research articles (e.g. letters, meeting reports, case reports, etc.). That leaves 423 research articles (3% of research articles) without an extracted Methods section. The reasons behind this are discussed in Additional file 7.

During the investigation of the paper structure, some of the papers labelled as research articles are not really research articles; they might be research news or reviews but archived as research articles. By removing these various articles that do not match

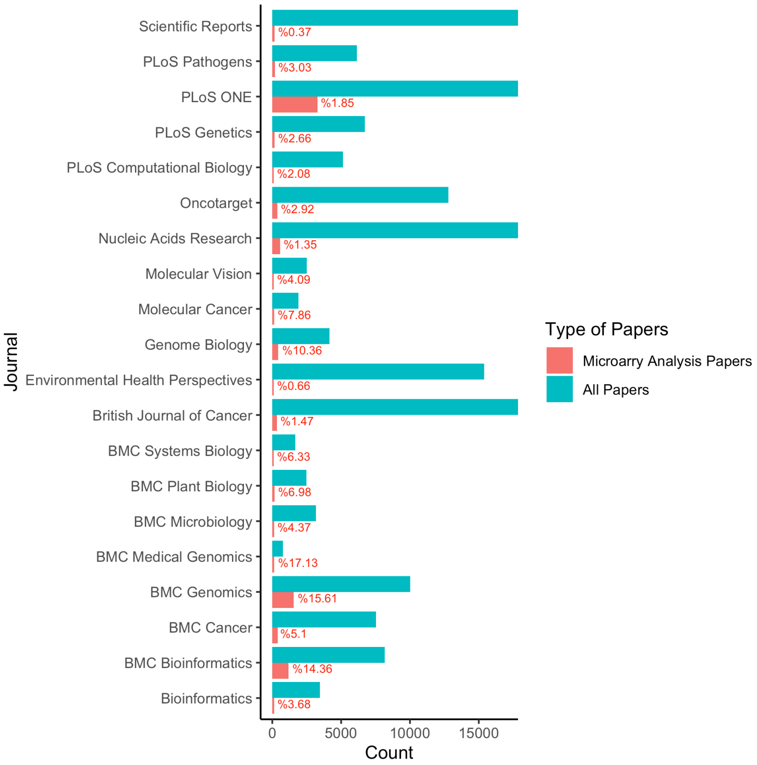


Figure 3.3. The percentage of microarray analysis papers. The blue bar shows the number of all publications per journal while the orange bar above it shows the number of microarray analysis papers. The percent is written on each bar. The journals are in descending order with respect to the number of microarray analysis papers. The top 20 journals are displayed.

the 'research article' profile, we removed 30 papers, resulting in a corpus of 13,559 articles. All the articles' details are in Additional file 4.

3.3.1 Functional discourse analysis

The most frequent functions over all the sections of the articles are `Result, Method` then `Background` (22.71%, 22.24% and 17.24%). Figure 3.5 shows the distribution of the functional discourse concepts over the papers' sections.

77.33% of `Method` mentions are in the Methods and Results sections while 87.49% of `Experiment` mentions are only in the Methods section. 56.71% of `Result` mentions are in the Results section while 66.65% of `Conclusion` mentions are in the Discussion section. 35.77% of `Background` mentions are found in the Introduction. It is also frequent in the Abstract, Result and Discussion sections (21.97%, 19.5% and 16.37%).

The Methods section contains all the eleven functions, divided into `Experiment` and `Method` (44.34%, 35.82%) followed by `Result`, `Background` and `Model` (8.33%, 3.22% and 3.21%). Although 53.36% of the `Model` mentions are in the Methods sec-

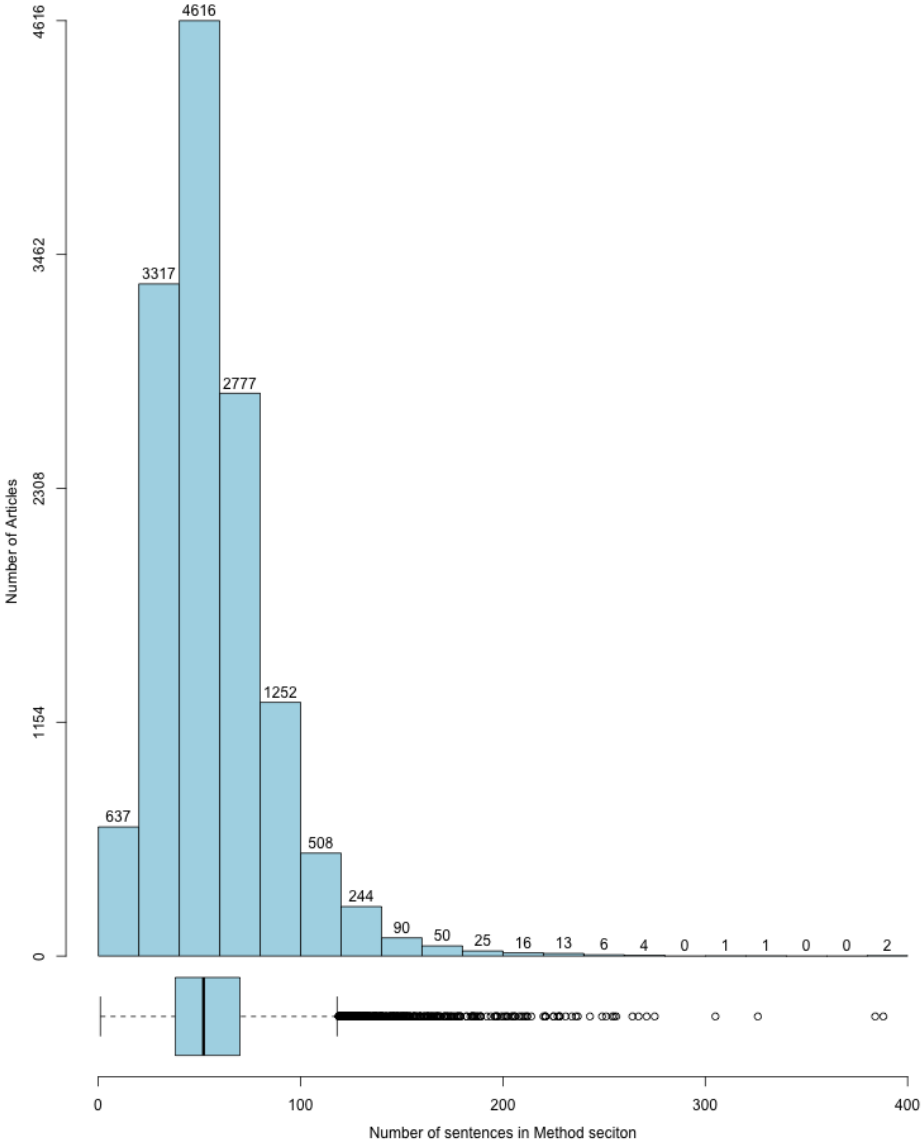


Figure 3.4. Methods section length analysis. The histogram shows the frequency of articles based on the Methods section length and the box plot diagram shows the median, first and third quartiles and outliers of the Methods section length based on Tukey's method [148].

tion, these mentions contribute only 3.21% of the mentions in the Methods section. This could be due to the nature of the microarray analysis papers which tends to have more experimental and theoretical methods than modelling or designing a theoretical solution. The presence of `Background` shows the propensity to mention some preliminary information either about the methods or data used. `Result` occurs more than `Background` and `Model`, where we can see the authors mention some factual result of using the methods inside the Methods section. Details of functional discourse distribution are in Additional file 5.

The average number of sentences for each function in the Methods section is shown in Figure 3.8. It shows that `Experiment` and `Method` have the highest averages.

We used a Pearson correlation coefficient to test the relationship between the functions over the papers' sections. We found 12 statistically significant relations that show there is a high chance when a number of sentences of one function is increased in a section, the other function is increased as well (see Figure 3.6). There is a strong positive relationship between (`Model` and `Method`), (`Goal` and `Object`) and finally between (`Result` and `Observation`). There is also a fairly strong positive relationship between (`Method` and `Experiment`), (`Background` and `Motivation`, (`Conclusion` and `Hypothesis`), (`Hypothesis` and `Result`), (`Observation` and `Object`), (`Observation` and `Goal`), (`Object` and `Result`) and finally between (`Goal` and `Result`).

Figure 3.7 shows the distribution of functional discourse concepts over the Methods sections' sentences. It can be noticed that `Background` is concentrated at the beginning of the section. `Result` is present in all sentences with almost the same level of usage. Despite fewer articles that have too long a Methods section, the usage did not decrease over the course of the Methods section but overtook the usage of the `Method` and `Experiment` functions.

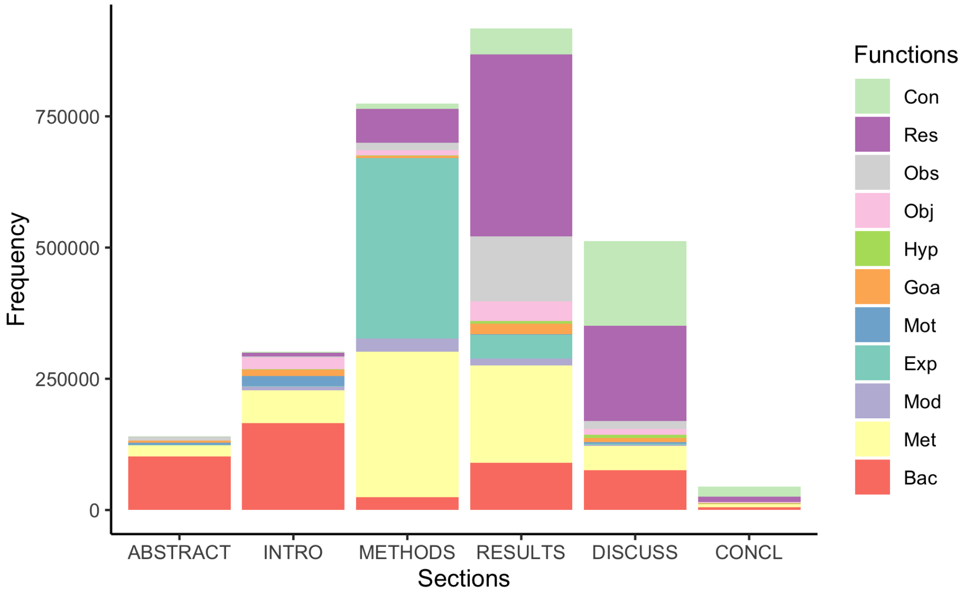


Figure 3.5. **Functional discourse concepts distribution over the article sections.** It is easy to notice that each section has all of the eleven functions, but with different percentages. 72.49% of the Abstract section is `Background` and the second most frequent function is `Method`, which constitutes only 14.85%. 55.01% of the Introduction section is `Background` and 20.69% is `Method`. Functions in the Methods section are distributed over `Experiment, Method` and `Result` (44.35%, 35.82% and 8.34%, respectively). 37.78%, 20.23% and 13.53% of the Result section are `Result, Method` and `Observation`, respectively. 35.48% of the Discussion section functions are `Result` and 31.47% are `Conclusion`. All numbers and percentages are in Additional file 5.

Using the top-k rule, the top 8 rules ordered by frequency then accuracy, are shown in

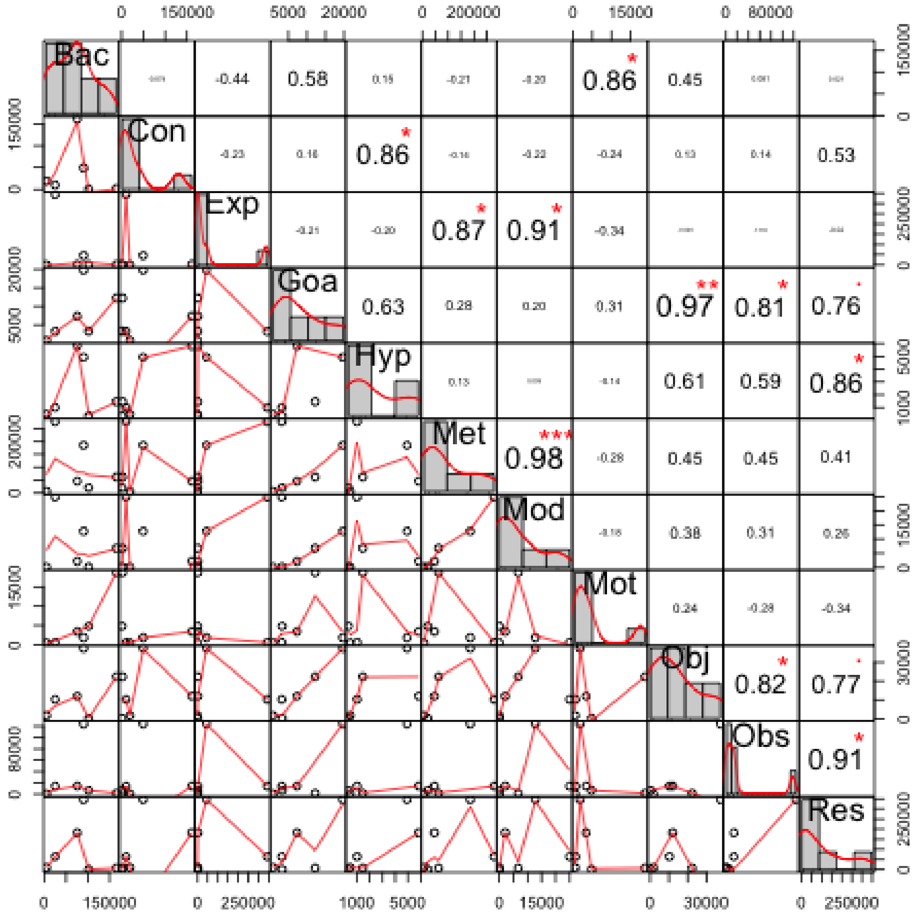


Figure 3.6. The correlation between functions over the sections. There are 12 statistically significant relations between the functions (out of 55). The Histogram with kernel density estimation and rug plot for each function is shown on the diagonal. On the top of the diagonal : the value of the correlation written with a font size proportional to the strength of the correlation and have the significance level as stars. Each significance level is associated to a symbol: p-values(0, 0.001, 0.01, 0.05, 0.1, 1) symbolised with (“***”, “**”, “*”, “.”, “ ”) respectively. The scatter plots with a fitted line are on the bottom of the diagonal.

Table 3.4. The most two frequent and accurate rules are that `Experiment` is followed by `Method` or vice versa. It also shows that a sentence with a `Result` is more likely to follow sentences with a `Experiment` function and `Method` function than preceding them. 2cm

Using the Markov Chain Model, the transition probabilities matrix is represented in the state diagram in Figure 3.9. The transition probabilities show that `Experiment` and `Method` are the most stable functions. This means that if the current sentence function is one of them, then this will remain for the next sentence. If it is going to be changed, then it is likely to be converted to the other, which is also supported by the first two rules in Table 3.4. `Model` is also stable with a 0.74 chance; we could say that in the Methods section the functions that belong to the approach category

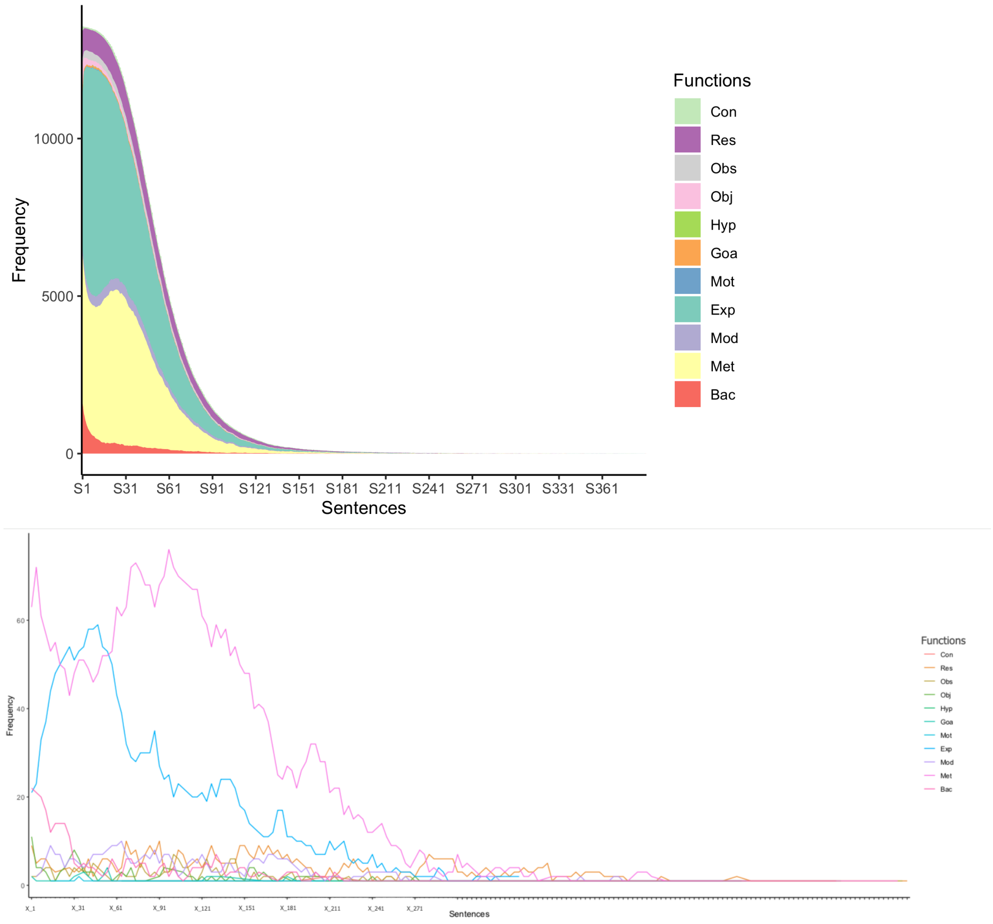


Figure 3.7. Functional discourse concepts distribution over sentences of the Methods section. It is more common to start with `Method` than `Experiment`. As a general trend, `Method` usage is more frequent than `Experiment`, the one exception is in the interval between sentence 20 and sentence 60. The usage of `Method` increased after that, reaching a peak around sentence 100. The `Background` mention is more frequent at the beginning of the section, then the usage is decreased steadily until sentence number 30 where the lower usage of `Background` is maintained through the section. In contrast, `Result` increased with the later sentences with small alternating increases and decreases, the most being between sentences 60 and 190. It overtook the usage of `Method` after 280 sentences.

(`Experiment`, `Method` and `Model`) are highly probable to come in a successive pattern. It is also noticeable from the state diagram that the transition for any function has a high chance of being to `Method`. The only exception is `Observation` with a 0.24 chance of moving to a `Result` in the next sentence and a chance of 0.15 of moving to `Method`. We also derived the steady state vector in Table 3.5. It gives a picture of what a function of a sentence could be after a long run of transition probability and almost reaching the end of the Methods section. It shows that it is expected to have `Experiment` by 0.43 and `Method` by 0.37. It is interesting that the probability of ending with `Experiment` is more than `Method` although the level of `Method` usage was the higher after sentence number 60. Besides the probability depending on the stability of a function, Figure 3.4 shows that 75% of articles finish at sentence 70 or before. The probability of ending with `Result` is 0.09 and other functions in the outcome

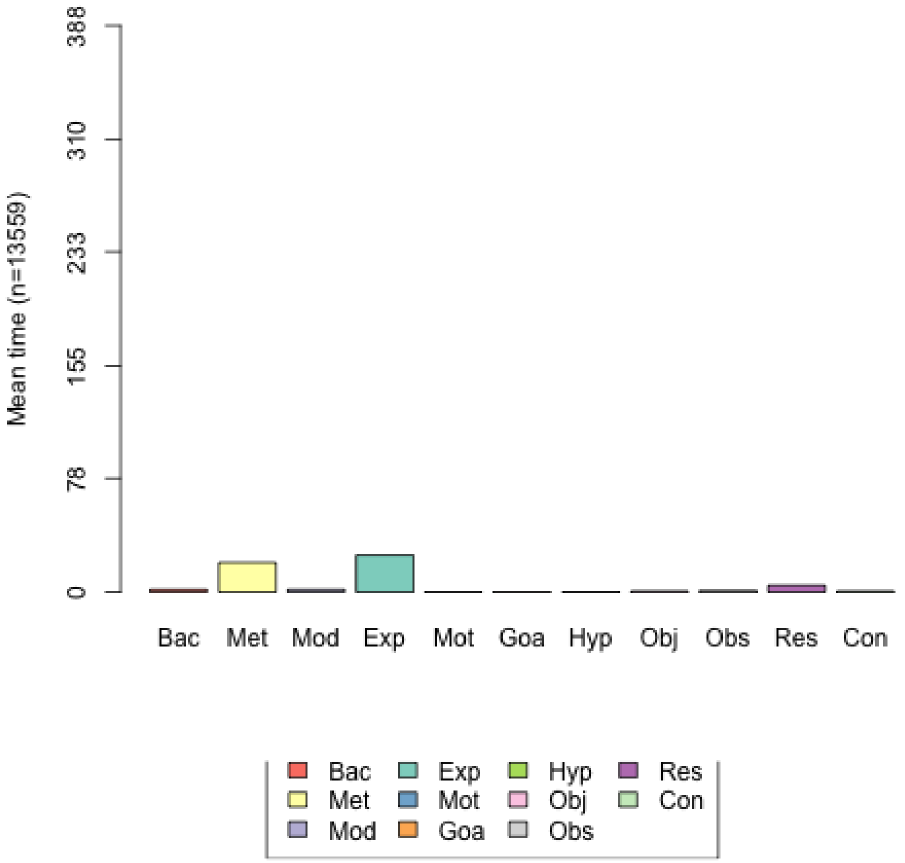


Figure 3.8. The average number of sentences for each function in the Methods section. Average number of sentences for `Method`, `Experiment` and `Result` are 20.449, 25.316 and 4.756 respectively.

Table 3.4. Top 8 rules show that an `Experiment` function is followed by a `Method` function with 91% confidence and a `Method` function is followed by an `Experiment` function with 78% confidence. The other rules relate `Method` functions with remaining functions which show `Background`, `Observation` and `Object` occur with high confidence before a `Method`, and where `Result` comes after `Method` and `Experiment`.

Rule	**Support**	**Confidence (accuracy)**
	occurrence of x → y / all sequences	Support / occurrence of x
`Experiment` → `Method`	79%	91%
`Method` → `Experiment`	75%	78%
`Method` → `Result`	56%	58%
`Method` , `Experiment` → `Result`	52%	90%
`Result` → `Method`	52%	84%
`Background` → `Method`	37%	90%
`Observation` → `Method`	35%	82%
`Object` → `Method`	26%	86%

category have 0.01 each. The probability of ending with `Background` is 0.03. The probability of having `Goal`, `Hypothesis` or `Motivation` at the end of the Methods

Table 3.5. The steady state vector means that after a long run of transition probability over a Methods section, it is highly likely to have `Experiment` or `Method` with a proportionality of 0.43 and 0.36 respectively.

Function	Transition probability
Bac	0.0285
Con	0.0144
Exp	0.4341
Goa	0.0041
Hyp	0.0013
Met	0.3648
Mod	0.0327
Mot	0.0007
Obj	0.0124
Obs	0.0189
Res	0.0879

section is zero.

Using the TraMineR package [146], the patterns are extracted as the sequences that are mostly similar to other sequences. Those patterns altogether should at least be similar to 88% of the whole sequence, as we stated in the Methods section 3.2.4. It means 88% of the whole set should be far less than a threshold from the obtained patterns set. The threshold is set to 55.7 (10% of the maximum distance between sequences). That means it only costs less than 55.7 to convert a sequence to its pattern. Maximum distance (557) is obtained from the distance matrix that has all the distances between the sequences. The pattern is written in state-permanence-sequence (SPS) format where a function is represented by an interval that shows how many successive sentences are holding the same function.

Applying the density method gives total coverage (88%) with a reasonable number of patterns (6) (see Figure 3.10). Some patterns show the behaviour of `Experiment` and `Method` while others show the behaviour of other functions that do not belong to the approach category. The patterns are as follows:

Pattern P1 (Met,1)-(Exp,19)-(Met,13)-(Exp,4)-(Met,2). It basically starts with a long interval of `Experiment` followed by `Method` and ends with some spells of both. 40.36% of the whole set belong to the cluster represented by P1. This pattern represents a balance between `Method` and `Experiment` mentions.

Pattern P2 (Met,13)-(Bac,1)-(Met,6)-(Res,3)-(Met,14)-(Res,1)-(Exp,14)-(Obs,1). This pattern has long spells of `Method` and `Experiment` with some intervention of `Background`, `Result` and `Observation`. 14.40% of the whole set belongs to the cluster represented by P2. This pattern has more `Method` mentions in total compared

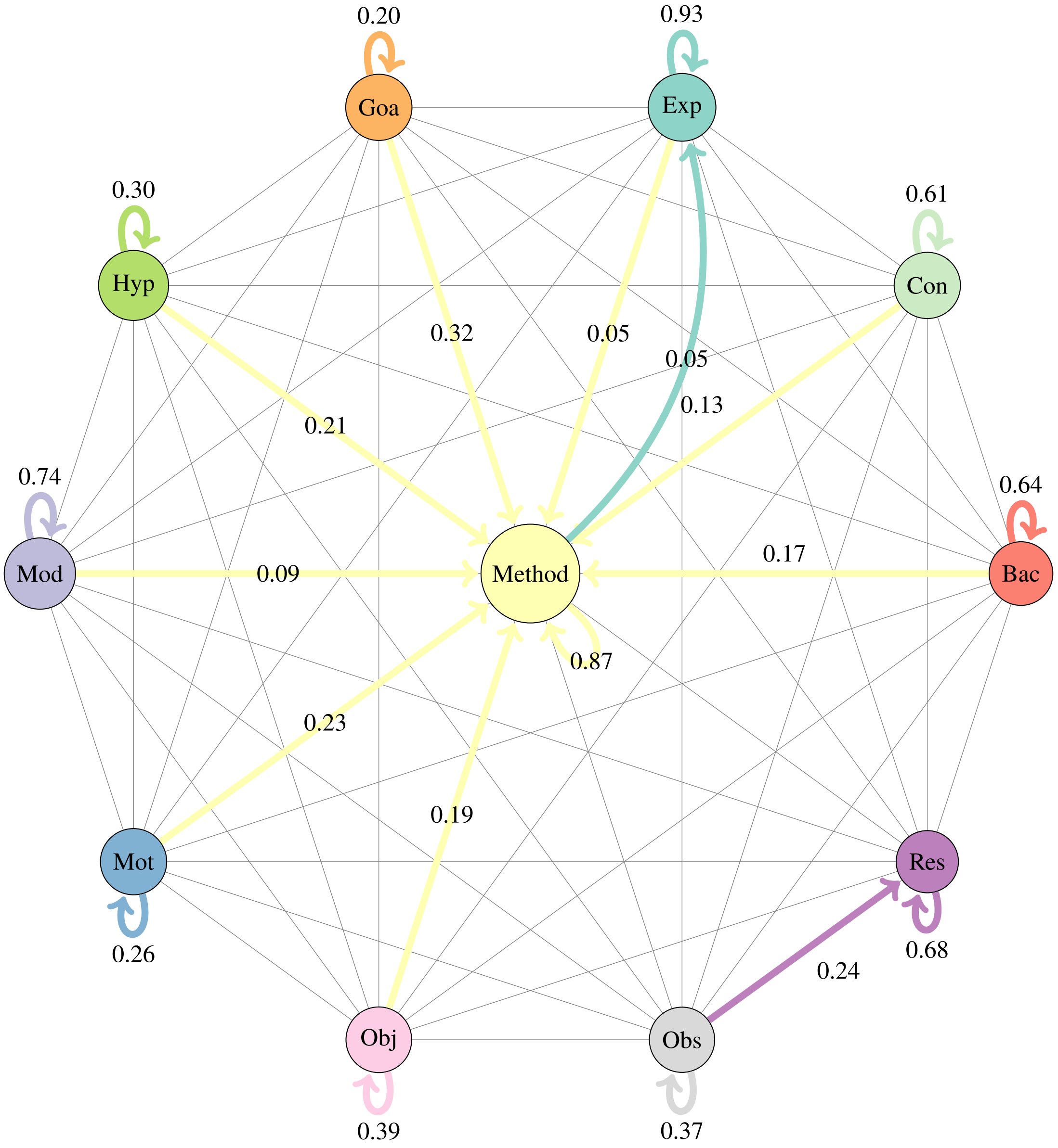


Figure 3.9. A state diagram. This shows the probability between state transitions. The loop arrow shows the stability of a function; if the current sentence is in state A, how how likely is it to be in state A in the next sentence? The most stable one is `Experiment` (0.93) and the most unstable one is `Goal` (0.20). `Method` is also a highly stable function and if it could be substituted with one of the 10 functions then this is going to be `Experiment`. We can see that most of the functions tend to be followed by `Method` except `Observation`, which is probably followed by a `Result`. All transition probabilities' data are in Additional file 6.

to the `Experiment` mentions.

Pattern P3 (Obj,1)-(Met,1)-(Res,2)-(Exp,13)-(Res,1)-(Met,2)-(Res,1)-(Met,4)-(Exp,23)-(Res,3)-(Exp,13)-(Bac,1). P3 expresses chunks of repeated patterns of `Method` and/or `Experiment` followed by `Result`. It has long successive mentions of `Experiment`. The pattern starts with an `Object` and ends with a `Background` and contains few mentions of `Method`. 19.21% of the whole set belongs to the cluster represented by

P3. The dominant function in this pattern is the `Experiment`.

Pattern P4 (Exp,2)-(Res,5)-(Met,7)-(Exp,1)-(Res,1)-(Exp,6)-(Met,9)-(Res,4)-(Exp,17)-(Res,2)-(Mod,6)-(Exp,5)-(Met,7). The forth cluster contains less than 1% of the sequences and generally expresses chunks of repeated patterns of `Method` and/or `Experiment` followed by `Result`.

Pattern P5 (Res,3)-(Con,2)-(Bac,4)-(Mot,1)-(Res,3)-(Con,1)-(Res,5)-(Met,2). This pattern represents 9.76% of the sequences. It has few spells of `Method` and represents the articles that focus more on explaining the background and result obtained using the methods more than the method itself.

Pattern P6 (Bac,7)-(Met,22)-(Exp,11)-(Met,29)-(Res,2). This pattern represents 6.67% of the sequences. It starts with `Background`, elaborate in `Method` and/or `Experiment` and ends with `Result`. It has more `Method` mentions than `Experiment`.

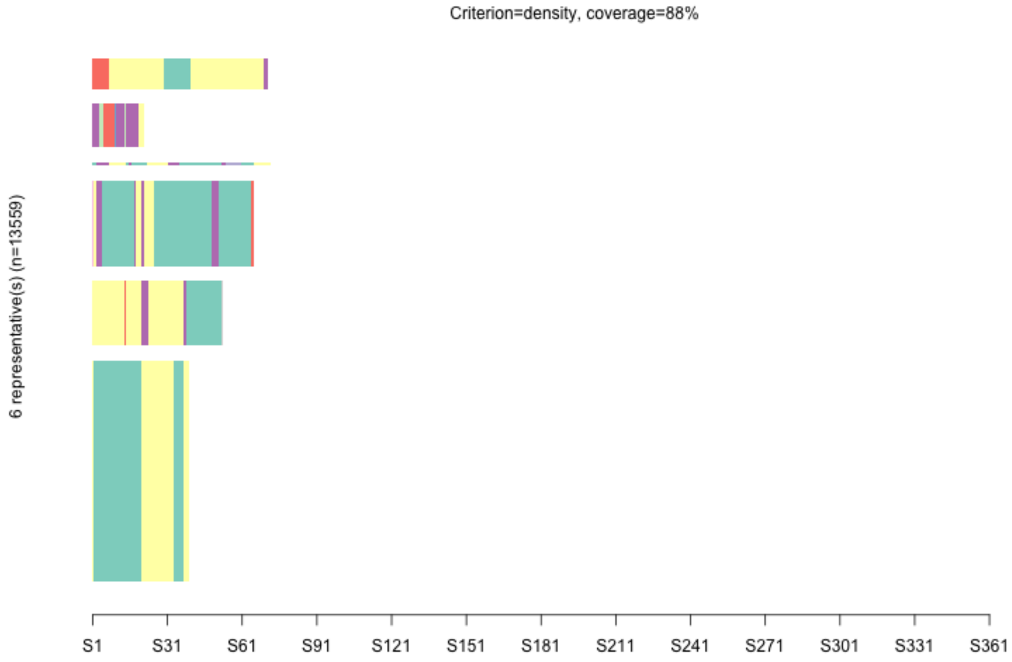


Figure 3.10. Patterns in terms of functional discourse. n= 13,559 is the number of Methods sections where the functional sequences have been analysed. The six patterns are the sequences chosen to represent all sequences. The width of a pattern is proportional to the number of sequences it represents. The criterion applied is the neighbourhood density with 88% total coverage. The bottom line is graded by the sentence number (1 to 388). The patterns are plotted for each sentence and coloured based on the function.

Table 3.6. The steady state for the four journals. It can be seen that BMC Bioinformatics, after a long run of transition probability over the Methods section, has the lowest probability in `Experiment` but the highest in `Model` and `Result`.

	PLoS One	BMC Genomics	BMC Bioinformatics	BMC Medical Genomics
Bac	0.0235	0.0234	0.061	0.041
Con	0.0077	0.0134	0.0435	0.0035
Exp	0.4898	0.3808	0.0467	0.2612
Goa	0.0032	0.0039	0.0106	0.0066
Hyp	0.0008	0.0012	0.0037	0.002
Met	0.3784	0.4146	0.3921	0.4988
Mod	0.0137	0.0213	0.1255	0.0567
Mot	0.0004	0.0005	0.0025	0.0015
Obj	0.0106	0.0124	0.0301	0.019
Obs	0.0134	0.0163	0.0299	0.0304
Res	0.0584	0.1123	0.2545	0.0793

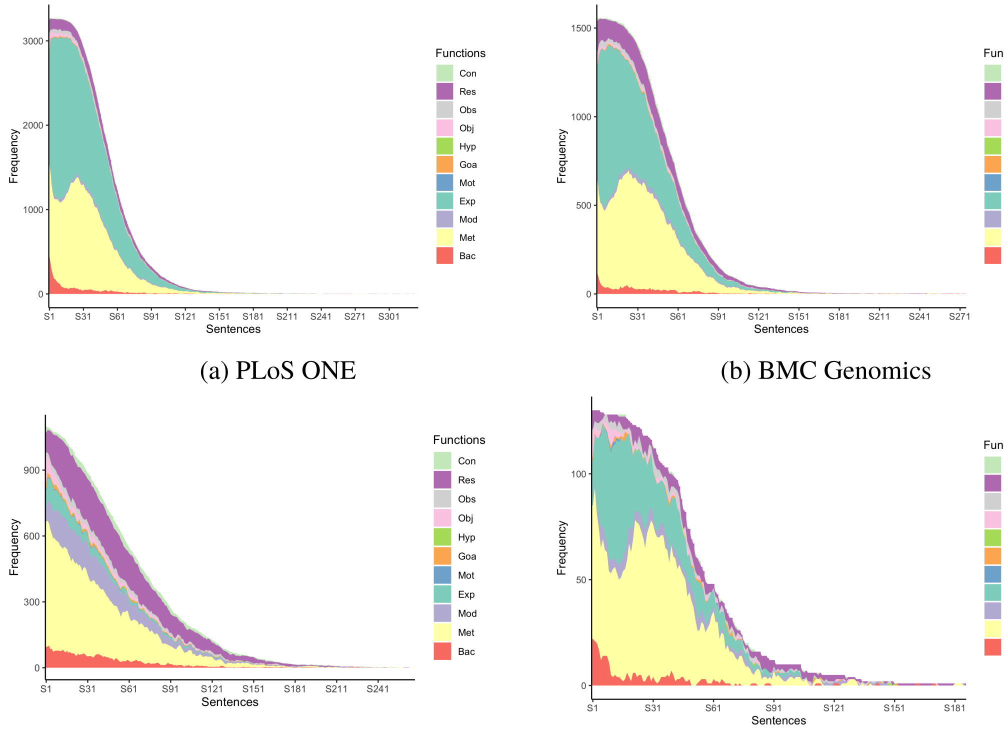
3.3.2 Journal patterns

There are 622 different journals in the corpus. We limit our analysis to the top three journals that have the largest number of microarray analysis papers and the top three journals that have a higher percent in publishing microarray analysis papers among their publications in the period covered by the corpus.

The top three publishing journals in our corpus are: PLoS ONE, BMC Genomics and BMC Bioinformatics and the top three journals with the highest percent of microarray articles are: BMC Medical Genomics, BMC Genomics and BMC Bioinformatics. In total, we have four journals to compare their writing behaviour in the Methods section in terms of the functional discourse.

Figure 3.11 shows the distribution of functional discourse concepts over the Methods sections' sentences. The distribution of functional discourse in four journals reflects the global behaviour of all the journals shown in Figure 3.7. However, BMC Bioinformatics, a non experimental journal, is slightly different in having far fewer `Experiment` and more `Model` functions. Also, the usage of the `Result` function did not decrease over the course of the long Methods section (within the inner outlier length of BMC Bioinformatics articles of 180 as in Figure 3.12) which means the usage of `Result` at the end of the section is frequent. This is also supported by steady state probability in Table 3.6 that the chances of ending with a `Result` is 26% compared to the other journals 11%, 6% and 8% and the average number of sentences with the `Result` function is more than the average number of sentences with the `Result` function in the other journals (see Figure 3.13).

Finding the patterns using TraMineR with the density neighbour method with 80% coverage and the threshold distance of 10% of the maximum length of the Methods



(a) PLoS ONE

(b) BMC Genomics

(c) BMC Bioinformatics

(d) BMC Medical Genomics

Figure 3.11. Functional discourse concepts distribution over the Methods sections' sentences in the top four journals. The x axis shows the sentences, the y axis shows how much the frequency of a function in a sentence over all the journal publications. PLoS ONE, BMC Genomics and BMC Medical Genomics have a similar distribution over their Methods section in having `Method` and `Experiment` functions. BMC Bioinformatics differs from the general behaviour by having much fewer `Experiment` functions and more `Model` and `Result` ones.

section for each journal produces a large number of patterns that are hard to be presented and analysed. For example, BMC Bioinformatics had 315 patterns and BMC Medical Genomic needed 53 patterns to represent the behaviour of the sequences. This could be because smaller corpora have smaller distances between functional sequences that means there are a limited number of operations to convert to the nearest pattern. If we have diversity between the functional sequences, then more patterns are expected. The maximum distance between sequences in the corpus was 557 while it is 267 in BMC Bioinformatics. It is more flexible to do up to 55.7 operations to convert a sequence to a pattern than limiting the operations to 26.7. This explains why these patterns are hidden by the longer more sequences in the whole corpus.

For simplicity, and the purpose of presentation, Figure 3.14 shows the top pattern for each journal and Table 3.7 shows the spells. Although the pattern did not cover the whole sequences, its behaviour agrees with the result obtained by the steady state (Table 3.6) and the average number of sentences (Figure 3.13). It is obvious that BMC Bioinformatics did not include `Experiment` in the pattern while more `Method` and `Result` are shown. This pattern is better linked to general pattern P2 that has more

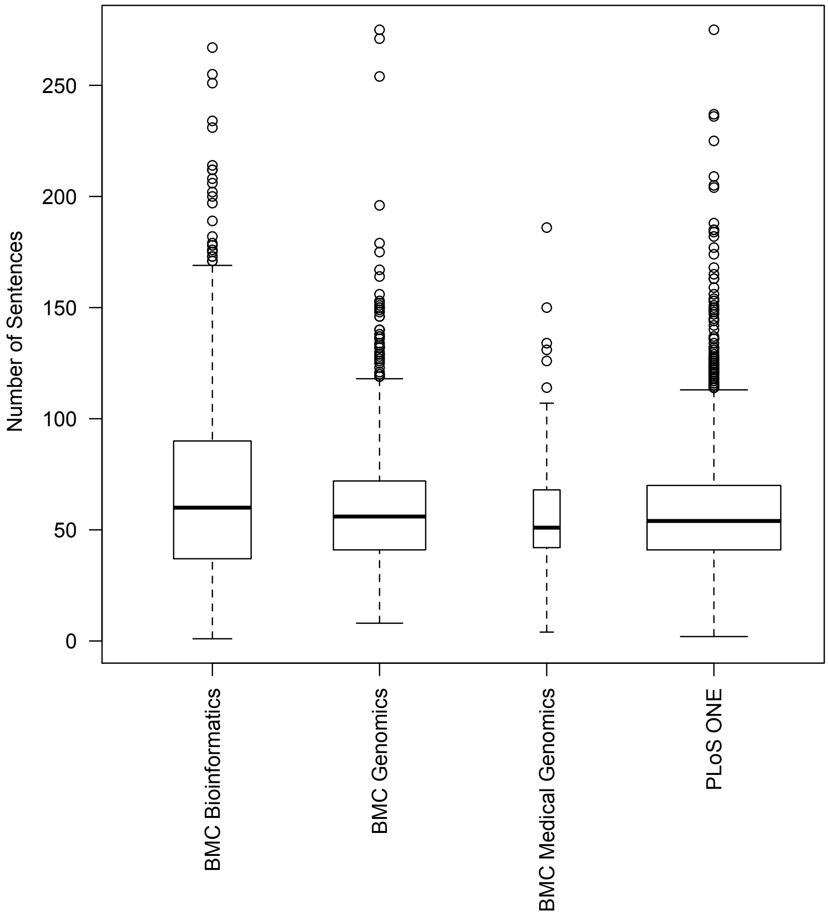


Figure 3.12. The length of the Methods section over the four journals. The width is proportional to the number of publications. The BMC Bioinformatics tends to have the longest length over the four journals.

Table 3.7. The top patterns of each journal.

Journal	Pattern
PLoS ONE	(Met,3)-(Exp,20)-(Met,14)-(Exp,7)
BMC Bioinformatics	(Met,9)-(Res,2)-(Met,11)-(Res,2)-(Met,4)
BMC Genomics	(Met,1)-(Exp,13)-(Met,16)-(Exp,1)-(Res,2)-(Met,2)
BMC Medical Genomic	(Met,28)

method in its sequence. P2 is also the nearest pattern to BMC Medical Genomics that has only spells of `Method`. PLoS ONE has a pattern that focuses on `Method` and `Experiment` whereas BMC Genomics has some `Result` functions in its pattern. Both of these two journals' patterns are mapped to the P1, the pattern that shows the balance appearance of `Method` and `Experiment`. Table 3.8 shows the exact cost between general patterns and journal patterns.

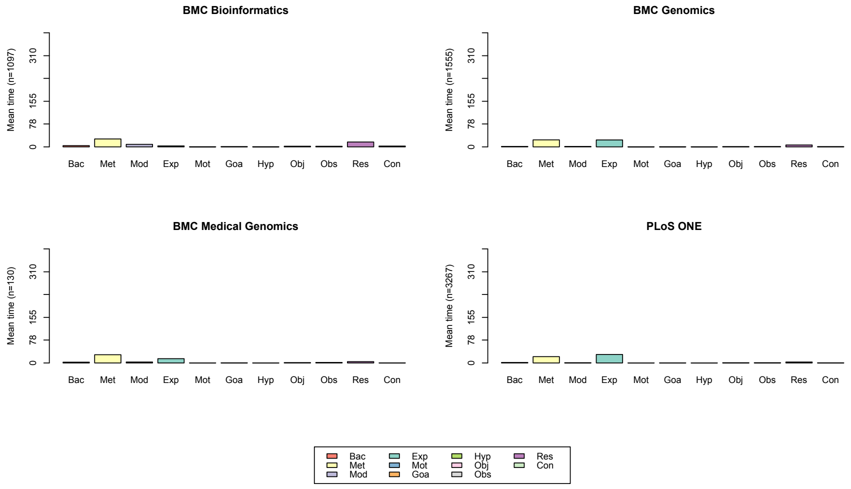


Figure 3.13. The average number of sentences with functional discourse concepts over the four journals. The average number of sentences with `Method` and `Experiment` are the most compared to the average number of sentences with other functions. However, BMC Bioinformatics has a greater mean for `Method`, `Model` and `Result` more than `Experiment`.

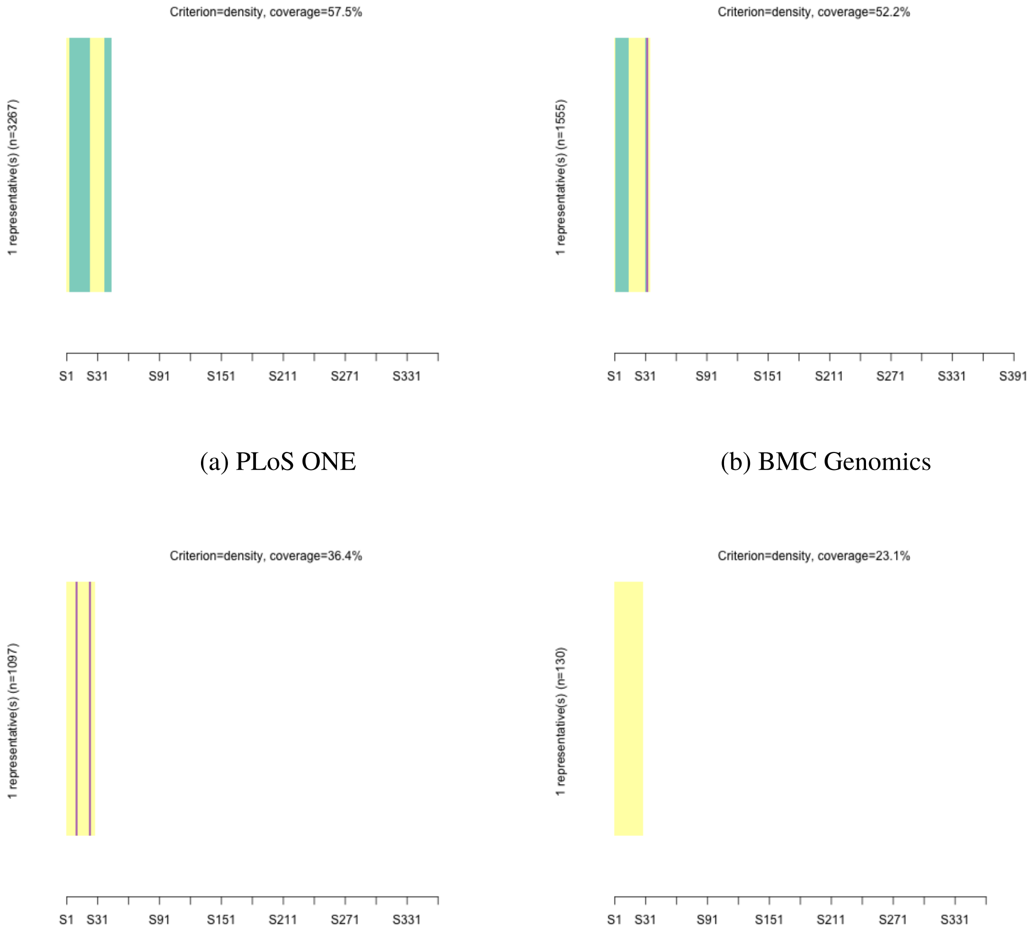
Table 3.8. Cost of mapping the journal patterns to one of the six patterns.

	P1	P2	P3	P4	P5	P6
BMC Bioinformatics	35	**31**	71	54	37	51
PLoS ONE	**9**	49	49	56	61	59
BMC Genomics	**14**	46	54	57	48	46
BMC Medical Genomic	35	**25**	79	54	45	43

3.3.3 Year patterns

The dominant patterns of the Methods section over the years from 2000 to 2017 have `Method` and `Experiment` intervals. Figure 3.15 and Table 3.9 show the best sequence that could represent the articles over the years. Figure 3.16 shows the usage of the six patterns over the years. We speculate the usage of the patterns over the years by calculating the mean of the distances between the patterns and the articles over the years; the smaller the mean, the higher chance that the pattern represents the articles over the year and hence the higher usage.

For patterns P1, P3 and P4, we can split the usage into two periods, from 2000 to 2007, the usage is decreasing and from 2007 to 2017, the usage is increasing. Pattern P2 has stable use since 2008. Pattern P5, the pattern which has few spells of methods and experiments, its chance to represent the articles decreased over the time. The usage of pattern P6 increased over the time since year 2004. Years 2000 and 20017 are not included since they have a small number of articles (2 and 7 respectively).



(a) PLoS ONE

(b) BMC Genomics

(c) BMC Bioinformatics

(d) BMC Medical Genomics

Figure 3.14. The patterns of Methods section of the top four journals. The PLoS ONE and BMC Medical Genomics have similar patterns as they completely focus on `Method` and `Experiment` whereas BMC Genomics and BMC Bioinformatics have some `Result` in the patterns.

3.4 Discussion

Most of the research articles have Methods sections that are over represented with sentences explaining methods, experiments and the result obtained by applying them. This article type goes with the observed rise of the experimental research articles in the 20th century that, leaving behind the observational articles, mainly reported the facts, that were dominant in the previous three centuries [122]. Generally, the articles used almost the same number of sentences in explaining the methods and reporting the results. The approach category that includes `Method`, `Experiment` and `Model` constitutes 38.56% of the functions of the articles. Almost the same percent of mentions (37.70%) applied to outcome category that includes `Result`, `Observation` and `Conclusion`. Background and objectives (that includes `Hypothesis`, `Goal`,

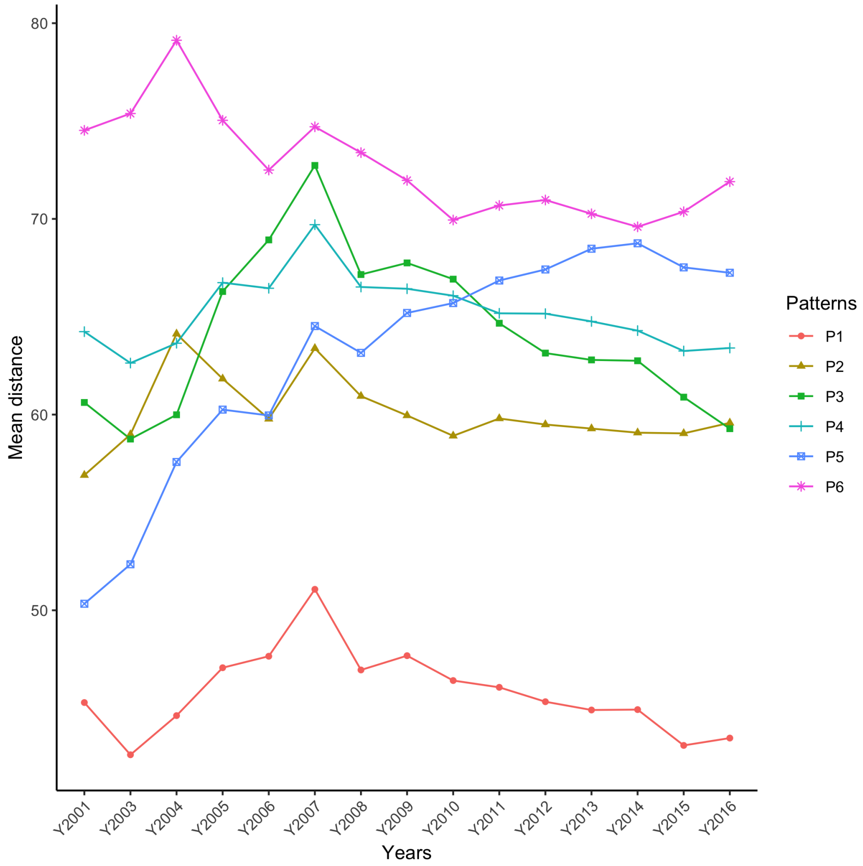


Figure 3.15. Usage of patterns over the years.

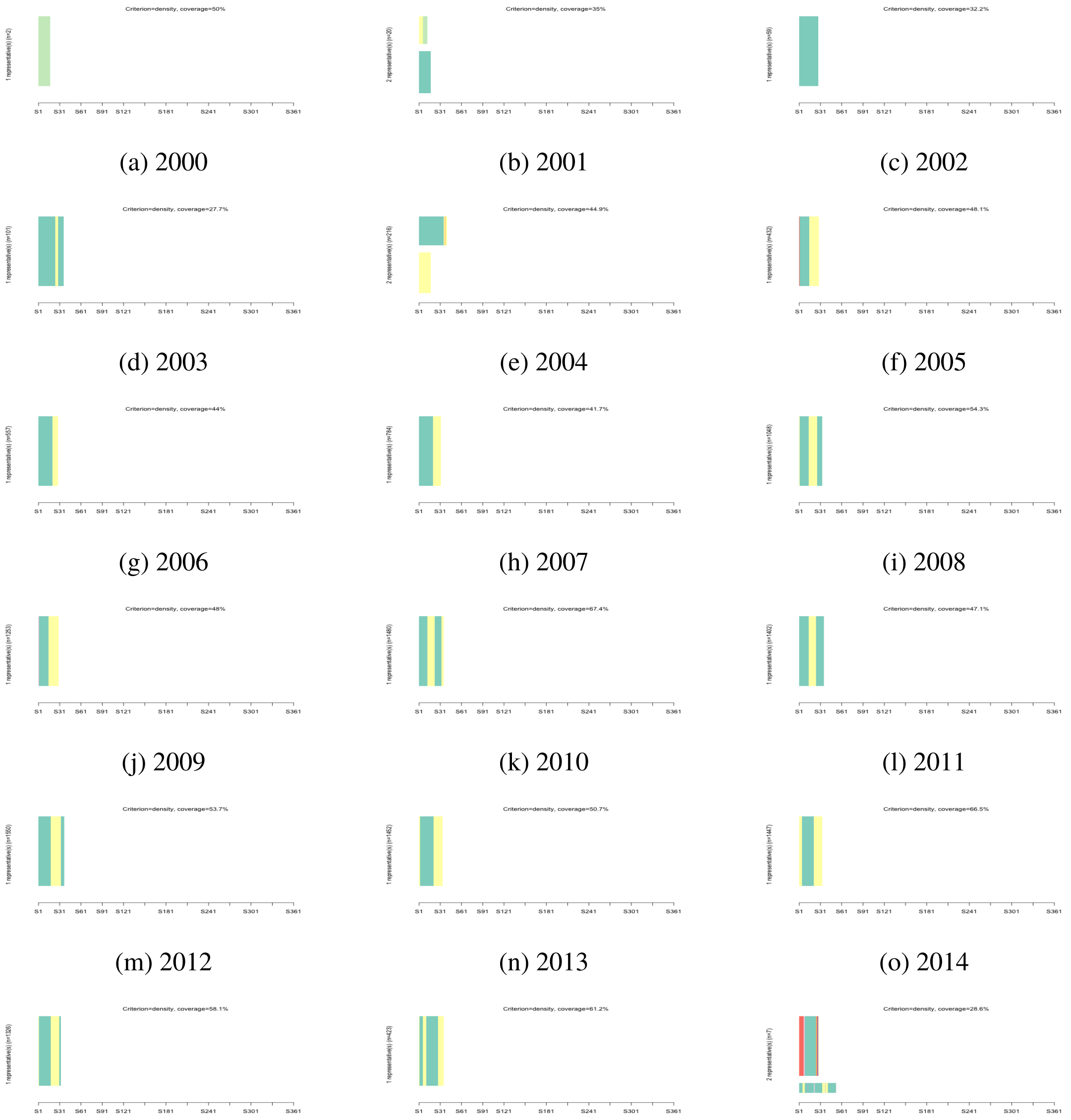
`Motivation` and `Object`) covered 17.24% and 6.50% respectively.

Methods are mostly written in the Methods section so it is legitimate to focus our attention on that section. Of course, we should check on the nature of the appropriate functions in the rest of the paper, but purely in terms of numbers, it looks like it is a reasonable thing to do.

The microarray analysis articles mostly include sentences that expressing experiments and methods in their Methods section. 80.17% of the Methods section's sentences are mentions of `Experiment` and `Method` functions.

Our analysis shows that the use of the `Experiment` function is highly restricted to the Methods section (87.49% of the mentions was in the Methods section). Almost half of the `Method` mentions are in the Methods section, the other half is distributed throughout the paper's sections.

Most of the articles follow the general writing guidelines in dedicating the Methods section for explaining the methods and experiments. Best practice guidelines suggest that the Methods section should focus only on the methods and procedures that are done, how the materials are prepared, inclusion and exclusion criteria and the analy-



(a) 2000 (b) 2001 (c) 2002

(d) 2003 (e) 2004 (f) 2005

(g) 2006 (h) 2007 (i) 2008

(j) 2009 (k) 2010 (l) 2011

(m) 2012 (n) 2013 (o) 2014

(p) 2015 (q) 2016 (r) 2017

Figure 3.16. The patterns of Methods section for each year in the corpus. The dominant patterns of Methods section over the years from 2001 to 2017 have `Method` and `Experiment` intervals. In 2017, only 7 documents are included in the study, combination of functions appeared in the patterns. Table 3.9 shows the spells of the patterns and the coverage percent.

sis tests that are used [149, 7, 150]. Besides these generally accepted writing styles, journals have their own instructions on how to write the Methods section (usually in submission guidelines). While some strictly prohibit adding any unimportant details, background information or found result, others are flexible and specify few or no instructions. For example, the aim of the study should be included in the Methods section in BMC Bioinformatics [151]. In PLoS ONE, if the study is about the microarray experiments, then Minimum Information About a Microarray Experiment should be followed as a guideline [152]. MIAME [20] requires the authors to mention the data as raw and after they have been processed. Due to this variation of guidelines and the

Table 3.9. Years patterns with the coverage percent.

Year	Number of articles	Coverage	Pattern
2000	2	50%	(Con,17)
2001	20	35%	(Exp,17) (Met,6)-(Con,6)
2002	59	32.2%	(Exp,27)
2003	101	27.7%	(Exp,24)-(Met,4)-(Exp,8)
2004	216	44.9%	(Met,17) (Exp,35)-(Met,2)-(Goa,1)-(Met,2)
2005	432	48.1%	(Bac,1)-(Exp,14)-(Met,13)
2006	557	44%	(Exp,20)-(Met,8)
2007	784	41.7%	(Exp,20)-(Met,11)
2008	1048	54.3%	(Met,1)-(Exp,13)-(Met,12)-(Exp,7)
2009	1253	48%	(Obj,1)-(Exp,14)-(Met,14)
2010	1480	67.4%	(Exp,12)-(Met,11)-(Exp,9)-(Met,3)
2011	1402	47.1%	(Exp,14)-(Met,10)-(Exp,11)
2012	1550	53.7%	(Exp,18)-(Met,14)-(Exp,5)
2013	1452	50.7%	(Met,2)-(Exp,19)-(Met,13)
2014	1447	66.5%	(Met,4)-(Exp,17)-(Met,12)
2015	1326	58.1%	(Met,1)-(Exp,17)-(Met,12)-(Exp,2)
2016	423	61.2%	(Met,1)-(Exp,5)-(Met,5)-(Exp,16)-(Met,8)
2017	7	28.6%	(Exp,5)-(Met,3)-(Exp,4)-(Mod,1)-(Exp,8)-(Obs,2)-(Exp,10)-(Met,4)-(Obs,1)-(Met,3)-(Exp,11)-(Obs,1) (Bac,7)-(Obj,1)-(Exp,16)-(Goa,2)-(Res,1)-(Met,1)

absence of writing templates and validating systems, the writing style would vary as well.

From the writing guidelines and functional discourse definitions, we expect to see most of the functions. Although the paper writes solely about the methods, the details and method explanation include some observations and factual results. We can observe short intervals of `Goal` that express the aim of the task done. There is also background information for more details and clarification.

Although most of the patterns alternate between `Method` and `Experiment`, there is a considerable appearance of `Result`. The presence of `Result` leads to the question of whether this result is a general one derived from using the mentioned method in general, or is a result obtained from the study. The same applied to `Observation` where some tables and figures could be added to explain the methods used, not the result obtained.

The number of patterns could give an overview about the style of writing in functional discourse, but what if we would like to precisely link a paper to a specific pattern? e.g. our paper, using SAPIENTA, has a functional sequence in SPS format as (Goa,1)-(Met,4)-(Obs,1)-(Met,4)-(Bac,4)-(Met,2)-(Res,1)-(Met,2)-(Res,1)-(Met,15)-(Mod,6)-(Met,4)-(Goa,1)-(Met,12). Table 3.10 shows the cost of converting the paper sequence to the general patterns and the journal patterns based on the LCS similar-

Table 3.10. Cost of mapping the paper pattern to one of the six patterns and journal patters. Mean distances to all articles of the journals are added between brackets.

	P1	P2	P3	P4	P5	P26	BMC bio	PLoS ONE	BMC genetics	BMC Med
Paper	65	45	105	70	63	**43**	**34**(67.91)	68(71.96)	55(69.40)	**30**(56.06)

ity measure. The most similar pattern is P6 ((Bac,7)-(Met,22)-(Exp,11)-(Met,29)-(Res,2)). The longest common sub-sequence was 43 , the length of the Methods section of the paper is 58 the length of P6 is 71, the cost is (58+71-2(43)) = 43.

For journals, since the patterns did not cover a large percent of the articles, we calculated the mean distance to all articles belong to that journal in addition to the distance to the reported pattern. The most similar patterns and articles belong to the BMC Medical Genomics and BMC Bioinformatics journals respectively.
Since the ultimate aim is to recover the context that may include input, output data, operations and software, some discourse functions could help in normalising the sentences that handle the required information. For example:

- `Background` sentences could be excluded since they talk either about general information or describe data in more detail.
- The `Result` and `Observation` sentences could contain output data.
- `Method`, `Experiment` and `Model` sentences need a further layer of interpretation in order to extract such information.

3.5 Limitations

1. Knowing that 12% of the articles are not represented by the reported patterns, how could the patterns be more general and cover the whole corpus? Could normalising the length and successive functions enhance the accuracy of the patterns?
2. The Methods section is often written with subsections. There is no reflection of the subsection in the patterns. Does the first section always follow a pattern that differs from other subsections? We could expect how the section could finish by steady state, but how it could start? Could we find the first and last subsections patterns and compare them to the middle subsections.
3. Calculating the distance between patterns and sequences could be enhanced by considering the relations between functions. For example, substituting a function with another from the same category should cost less than substituting it with a function from another category.

4. How far can we trust the annotated functional discourse? The accuracy of the extracted patterns mainly depends on the accuracy of SAPIENTA. SAPIENTA original papers reported F-score of 51% and 81% for chemistry papers at the all sections level and abstract level, respectively. We got an F-score of 62% for microarray analysis's Methods section. To overcome this limitation, the reported functional discourse could be supported with another type of discourse like argumentative zone or another kind of processing to analyse if the analysed sentence is important in reporting the operations included the paper context.

3.6 Conclusion

We examined the writing style of the Methods section in term of functional discourse. The extracted patterns clearly showed that most of the Methods sections are dedicated to the purpose of explaining the methods. The methods and experiments with some little result are usually presented in the patterns. Methods section sentences are also a mixture of the current method and method from other studies and thus functional discourse alone is insufficient to isolate a particular method. Further interpretation is needed for functional discourse to specify the contexts that express the methods, data and software.

3.7 Additional Files

3.7.1 Additional file 1 — Journal's microarray publications

Additional file shows the number of a journal's microarray publications from year 1999 to 2017. The percentage of microarray publication is calculated for the top 20 journals that publish 100 microarray articles or more and plotted in Figure 3.3.

3.7.2 Additional file 2 — Articles types frequency

Additional file shows all articles types and their frequency.

3.7.3 Additional file 3 — MeSH terms and frequency

Additional file shows all MeSH terms, frequency and parentage to MeSH terms with and without the four main microarry MeSH terms, to see the general terms distribution over general terms.

3.7.4 Additional file 4 — Articles PMC IDs and other details

Additional file shows all retrieved PMC ID, with journal, type, subject and year of publication. It also contains the list of PMC IDs used in our corpus , and the list of the removed articles PMC IDs(865 have no Methods section as a result of Section Tagger tool, the specific 423 Research articles IDs as part of them, additional 30 articles are removed because they either do not match the ‘research article’ profile or failed to be annotated by SAPIENT). These numerical information used for Figures 3.2 .

3.7.5 Additional file 5 — Functional discourse distribution

Additional file shows the frequency of each function in each section. The percentage of a function to the functions in whole paper and for each section. The numerical data used to create and explain the Figure 3.5.

3.7.6 Additional file 6 — Functional discourse frequency for each sentence and transition matrix

The numerical data used to create Figures 3.7, 3.9 and 3.4.

Chapter 4

Identification and Normalisation of Operations and Data in the Computational Biology literature

Abstract

Background Understanding the work conducted in research is essential to validate and replicate the work presented in research articles. The first step in this endeavour must be to understand and identify the basic parts involved in the work conducted such as ***operations***, ***data***, ***software*** and ***databases***. It is of interest to extract the workflow of a paper that shows the relationships between these parts. We want to know how far we can extract the ***operations*** and ***data*** from plain text? Can ***operations*** and ***data*** be normalised to known concepts in knowledge resources such as the EDAM ontology? Can the relations between the ***operations*** and ***data*** be identified and generalised among the publications?

Objective This study evaluates the feasibility of using NLP and deep learning approaches for extracting and linking ***operations***, ***input data***, ***output data***, ***software*** and ***database*** reported in the methods and representing them with their normalised concepts in a viewable platform

Methods We have developed the ODNoR system that identifies ***operations*** and ***data*** terms from the primary literature, normalises the identified entities, and finds the relations between the ***data***, ***operations***, ***software*** and ***databases***.

Results We identified such entities with an F-measure between 92.46% and 78.10%. The F-scores for identifying relations ranged between 92.5% and 62%. The normalisation accuracy was between 96% and 84.35%. The annotation with relations is presented in BRAT format.

Conclusions The proposed end-to-end system achieved encouraging results and demonstrated the feasibility of using deep learning methods to extract the keys that represents the methods from the Methods section in microarray analysis literature.

4.1 Introduction

In scientific research, scientific methods are substantial part that are being proposed, modified, and used to solve scientific problems. Methods are typically described with prose in scientific publications. The description illustrates how the work is conducted, what components are involved and why they were chosen for use [7]. Given the number of publications, extracting and identifying these components is essential for providing information about which methods are available for a given task.

There are some works to extract the key components from the scientific articles. SienceIE [119] is a corpus for the SEMEval 2017 task 10. It contains 500 scientific articles of the Computer Science, Material Sciences and Physics domains. It includes three types of entities (called keyphrases): Tasks, Methods, and Materials and two relation types: hyponym-of and synonym-of. Augenstein et al. [119] and Luan et al. [120] introduced the state-of-the-art systems that extract entities and relations from this dataset.

SciERC [121] extended previous dataset, SienceIE, to include 500 annotated scientific abstracts, from AI domain, containing annotations for scientific entities, their relations, and coreference information between mentions. They defined six types for annotating scientific entities (Task, Method, Metric, Material, Other-ScientificTerm and Generic) and seven relation types (Compare, Part-of, Conjunction, Evaluate-for, Feature-of, Used-for, HyponymOf). They developed *SciIE*, BiLSTM model, to extract entities, relations and coreference resolution across sentences and achieved F-score of 64.2%, 39.3% and 48.2%, respectively.

SciREX [118] extended the previous dataset, SciERC, to include 438 fully annotated documents from ML articles available in Papers with Code website. It contains annotations for entity mentions (Dataset, Metric, Task, Method), coreferences, salient entities, and N-ary relation over the document (between clustered salient entities). They introduced an end-to-end neural model, using BERT-BiLSTM-CRF, to perform document level IE task. The model to outperform existing state-of-the-art models on subtasks and achieved F-score between 67%-27% for the end-to-end tasks.

SciRes [16] is a dataset of 3,088 manually annotated resource contexts, i.e. a sequence of words that appear around a particular citation. They used full articles from three sources: the ACL Anthology Reference Corpus (ARC), the NeurlPS Proceedings (NeurlPS), and the PubMed. They have focused on the resource citation hyperlinks mentioned in the scientific articles to extract the methods' components as (online-resource citation, resource role and resource function). They defined three general role types with 9 subtypes: Material (Data), Method (Tool, Code, Algorithm) and Supplement (Document, Website, Paper, License, Media). The function types are: Use, Produce, Introduce, Extend, Compare, and Other. They implemented a BERT-based classifier, *SciResCLF*, to classify the roles and functions of a resource based on the word sequences of resource contexts. Then *SciResREC*, another BERT-Based model, uses these classification labels as features for predicting the resource hyperlinks. SciResCLF classifier achieved an F-score between 90%-35% and SciResREC achieved an accuracy of 50% in predicting resources.

Our aim in this paper is to extract the main parts that express a method conducted and reported in the computational biology literature. Duck et al. [71] have previously developed bioNerDS, a named entity recogniser for bioinformatics software and databases in the literature [127]. They have also used the results of a survey of the literature using bioNerDS to give a global view of the networks of tools and databases used [11]. The next step is to identify the ***operations*** and ***data*** used in a Methods section for a single research article.

In our context, an ***operation*** is a specific/concrete computational process conducted in an experiment and presented in a manuscript as a part of the work done in a paper. The ***operation*** could have data inputs and produce output data. It could also have either input or output data and, in some cases, it could be mentioned without any data but an indication that a process has been done. We also have ***operations*** that did not do any kind of processing but assign the ***data*** a role or identification; for example:

Example 2 (PMC2582621)
"In total, 355 probes are included in this study."

data are items (or collections of items) that exist in digital form (i.e. representation), and can be potentially used as an input to, or output from, a database, software that processes those ***data*** to fulfil specific/concrete computational processes that are conducted in an experiment and presented in a manuscript. This description includes any ***data*** used in the experiment, even when no explicit ***operations*** are mentioned in the sentence context. It excludes the abstract data that are used for conceptual and explanatory purposes (e.g., discussion of strings, integers, etc..). It also excludes the real/physical data and any biological processes conducted on them, e.g. the temperature, unless those measurements become part of the experimental data.

In this paper we introduce and evaluate ODNoR, that accepts a plain text description of a method then produces normalised entities and identified relations between them. The entities includes ***data***, ***operation***, ***software*** and ***database***. With relations ***input***

data, output data, by software and ***from database***. The extracted entities and relations can be read as: the *data* ***is from*** *database* and ***is an input*** for an *operation* that is performed **by** *software* to produce ***an output*** *data*

The system comprises a new named-entity facility that recognises the ***operations*** and ***data***, combined with the use of bioNerDS to recognise ***software*** and ***databases***. The new recogniser is based on transformer technology that utilises the power of attention and word embedding and achieved the state-of-the-art performance on eleven common NLP tasks in 2018. For example, BERT [49] achieved F-score of 93.16%, 1.5% point absolute improvement, on SQuAD (Stanford Question Answering Dataset) for question answering task and 80.5% (7.7% point absolute improvement) on GLUE (General Language Understanding Evaluation) for all nine natural language understanding tasks.

Specifically, we used BioBERT [50], initialised with the BERT weights and trained on PubMed article abstracts and PubMed Central full texts. BioBERT outperforms the BERT on three biomedical text mining tasks: biomedical named entity recognition (0.62% F1 score improvement), biomedical relation extraction (2.80% F1 score improvement) and biomedical question answering (12.24% MRR improvement). They compared the performance of BERT and BioBERT on different biomedical detests, such as BC4CHEMD dataset for drug and chemical entities, CHEMPROT dataset for protein–chemica relations and BioASQ 6b-factoid dataset for question answering. We fine-tuned BioBERT on our labelled data to derive a model that is able to annotate a large quantity of data.

The system also recognises the relationships between the recognised entities using rule-based and machine learning systems.

Data are an essential part of the methods since they represent the dataset that is being used in an experiment. In computational biology, extensive use of data is required in dry or wet lab experiments. In the manuscripts, the data may be mentioned in a number of ways based on the purpose of that mention. For example:

- A DOI is referred to inside the text by reference or a link (doi 10.5061/dryad.478g5).
- Data may be reported by mentioning the file format where the data exist (.CLE in raw intensity (.CEL) file).
- Mention of biological term's names (Gene 0.4 protein), short name (Gp0.4).
- Mention of accession number as a reference to a database record (A2BC19).
- Mention of sequences (ACTATCTAGAGCGGCCGCTT).

- Reference to a biological concept that is used in the dataset (cluster, transcripts).
- Metadata that refer to real data (human and mouse genomes).
- Parameters for named and unnamed software (for example blastn, e-value and DUST filter in "*BLASTed (blastn, e-value = 106, DUST filter)*" or identity in "*identity >= 94%*").

Kafkas et al. [35] studied the percentage of the data mentions in the body of articles and supplementary files and found the supplementary files contain more mentions than the body.

Entity normalisation, also known as entity linking, simply aims to find a corresponding concept defined in a knowledge base (KB) or an ontology and link it to the annotated entities. Dictionary lookup approach is widely used to enhance the entity linking [68, 58]. Most of the biomedical entity normalisation studies in the last decades use the morphological information to normalise the entities [74]. The state-of-art rule-based system proposed by D'Souza and Ng [75] on two datasets, ShARe/CLEF eHealth Challenge corpus [76] and NCBI disease corpus [77]. It achieved F-score 90.75% and 84.65% on the two datasets. It defined ten kinds of rules at different priority levels, such as abbreviation expansion, to measure morphological similarities between disorder mentions and concepts in two Knowledge base: SNOMED-CT resource of the UMLS Metathesaurus [78] and MEDIC lexicon [79]. The current state-of-the-art is achieved by Ji et al. [86], which increased the accuracy of biomedical entity normalisation by 1.17%. They compare the performance of three transformers models (BERT [49], BioBERT [50] and ClinicalBERTB [87]) on three different datasets (ShARe/CLEF [76], NCBI [77] and TAC2017ADR [88]) and achieved the best F-score as 91.10% 89.06% 93.22%, respectively.

Normalising the mentions of ***operations*** and ***data*** is important to ensure that the operation has a common meaning. This in turn is a facility to populate such knowledge with instances from the literature. EDAM [34] is an ontology that describes bioinformatics and biomedical resources. The current version includes over 2,200 concepts that are described using names (terms), synonyms, definitions and other properties. The definitions of concepts include useful terms that can be used as control variables for finding the biological terms in the text. They can also be used for semantic annotation of workflows and web services, and as a standard for exchanging data. EDAM can also be used for verifying files and exchange formats, as some useful information about regular expressions can be used to validate the exchanged identifier values.

EDAM includes four top main sub-ontologies, `Topic`, `Data`, `Operation` and `Format`.

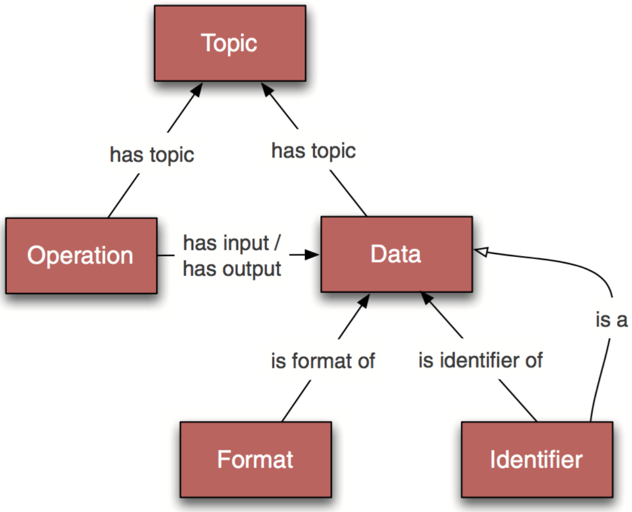


Figure 4.1. EDAM ontology concepts and relations. Data has a topic, is an output or input to an operation, identified by an identifier and has a format. Image is taken from Ison et al. [34].

A fifth sub-ontology, called `Identifier`, is rooted under `Data`. Table 4.1 shows the definitions and examples of the five main concepts. The sub-ontologies are linked by five types of relationships (see Figure 4.1). They are:
`Operation` has a `Topic`
`Data` has a `Topic`
`Operation` has input/output `Data`
`Format` is format of `Data`
`Identifier` is identifier of `Data`
An example of relations with instances is: Protein threading (is an `operation`) (has input `Data`) Protein structure (has output `Data`) Sequence-structure alignment.

Bio.tools [153] is used to normalise the software mentions and a further post-processing step. Bio.tools is a repository of tools' description that includes at least three core attributes (name, short description and homepage). The description may include up to 50 important scientific, technical and administrative attributes. One of the function attributes, which (if it exists), must contain at least one operation and may contain the data input and output. All of these, if they exist, are defined by EDAM's `Operation`, `Data(format)` ontologies.

As an example:

Bioconductor is a tool that is archived with three `Operations`

(`Analysis`, `Data handling` and `Statistical calculation`)

and accepts three types of `Data` input

(`Sequence`, `Data reference` and `Experimental measurement`)

and produces one of three types of `Data`

(`Mathematical model` (with `workflow` format), `Ontology data` or `Report`)

The `Topic` may be included as part of a tool's description. For the Bioconductor tool, four terms were associated: (`Bioinformatics`, `Computational biology`, `Data management` and `Statistics and probability`)

Relation extraction aims to extract and classify relational between identified entity mentions in plain text. Generally speaking, relation extraction modules can be classified into two categories, rule-based approaches relying on predefined patterns and machine learning methods based on well-designed features. Recent work involved larger corpora for extracting better patterns [63]. PATTY system [80] is based on mining algorithm that computes the n-gram combinations with large co-occurrence support. It processed two different corpora: the New York Times archive and the English edition of Wikipedia and achieved an accuracy between 75%-84.7%. RelEx [81] is a rule-based biomedical relationship extraction system. It extracted 150,000 relations between genes and proteins from set of one million MEDLINE abstracts with an F-score of 78%. Zhou et al. [89] achieved state-of-the-art in relation extraction from a plain text without the need to get high-level features from lexical resources such as WordNet or NLP systems like dependency parser and named entity recognizers (NER). They used Att-BiLSTM, on SemEval-2010 Task 8 dataset, to capture the most important semantic information in a sentence and achieved F-score of 84%. Soares et al. [90] introduced another state-of-the-art RE system using BERT. The learning method based on matching the blanks (MTB) that learns relation representations from entity resolution annotations without any further tuning for relation extraction. It achieved F-score of 89.5% on SemEval-2010 Task 8 dataset. Cohen et al. [91] reported the current state-of-the-art, on SemEval-2010 Task 8 dataset, which achieved 91.9% using BERT model. They used supervised span-prediction based system, similar to question answering (QA), for relation classification (RC).

4.2 Materials and Methods

Figure 4.2 represents a high-level overview of ODNoR. An outline of our approach is as follows:

1. Preparation of data
2. Annotation

Table 4.1. EDAM Concepts Definition. Table is taken from Ison et al. [34].

Sub-ontology	**Number of concepts**	**EDAM definition**	**General Definition**	**Examples**
`Operation`	558	Singular, bioinformatics-specific operations that are functions of tools, workflows or scripts, or can be performed manually	Can have special cases a)no input b)no output c)no input no output but change the current state	Protein threading - ID mapping - Service discovery - Molecular dynamics
`Data`	1,140	Types of data that are relevant in bioinformatics, commonly used as inputs, outputs or intermediate data of analyses, or provided by databases and portals	Information, represented in an information artefact (data record) that is ‘understandable’ by dedicated computational tools that can use the data as input or produce it as output	Annotated text-online course - DNA structure - Sequence alignment image - Evidence - Profile-profile alignment - Gene expression matrix
`Topic`	209	Application domains of bioinformatics tools and resources; topics of research, studies or analyses; approaches, techniques and paradigms within -or directly related to- Bioinformatics		Cell biology - Model organisms - Allergy, clinical immunology and immunotherapeutics
`Identifiers` (under Data)	528	Types of identifiers that identify biological or computational entities; including resource-specific data accessions. Several identifier concepts in EDAM include regular expressions and examples	The identifier is not unique, the same id may identify multiple things	NCBI genome accession - DOI (note: it has a regular expression property within the EDAM ontology (doi)?[0-9]2[0-9]4/.*) - CABRI accession
`Format`	347	Data formats commonly used in -and specific to- Bioinformatics. Many format concepts in EDAM include references to their definition and documentation		FASTA - T-Coffee format - debug - InChI

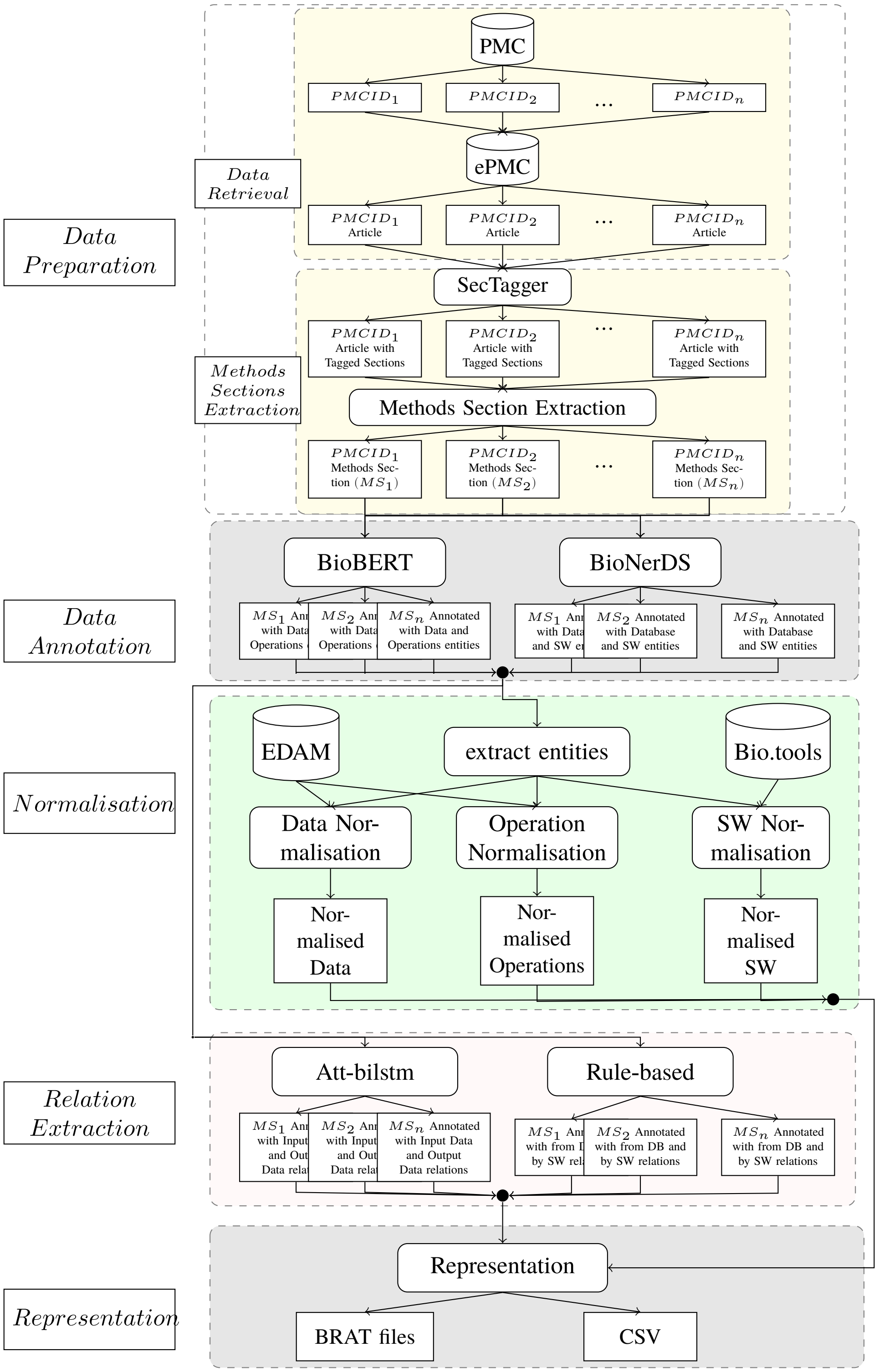

Data Preparation
Data Retrieval
PMC
$PMCID_1$
$PMCID_2$
...
$PMCID_n$
ePMC
$PMCID_1$ Article
$PMCID_2$ Article
$PMCID_n$ Article
SecTagger
$PMCID_1$ Article with Tagged Sections
$PMCID_2$ Article with Tagged Sections
$PMCID_n$ Article with Tagged Sections
Methods Sections Extraction
Methods Section Extraction
$PMCID_1$ Methods Section (MS_1)
$PMCID_2$ Methods Section (MS_2)
$PMCID_n$ Methods Section (MS_n)
Data Annotation
BioBERT
BioNerDS
MS_1 Ann with Data Operations
MS_2 Ann with Data Operations
MS_n Annotated with Data and Operations entities
MS_1 Ann with Data and SW en
MS_2 Annotated with Database and SW entities
MS_n Annotated with Database and SW entities
Normalisation
EDAM
extract entities
Bio.tools
Data Normalisation
Operation Normalisation
SW Normalisation
Normalised Data
Normalised Operations
Normalised SW
Relation Extraction
Att-bilstm
Rule-based
MS_1 Ann with Input and Out Data rela
MS_2 Ann with Input and Out Data rela
MS_n Annotated with Input Data and Output Data relations
MS_1 Ann with from D by SW rel
MS_2 Annotated with from DB and by SW relations
MS_n Annotated with from DB and by SW relations
Representation
Representation
BRAT files
CSV

Figure 4.2. System workflow. There are three main steps before presenting an experiment's basic parts; data preparation, annotation and relation extraction. Normalisation is an additional layer to present the entities in well-know resources concepts.

3. Normalisation
4. Relation extraction
5. Representation

ODNoR is designed and developed as a Named Entity Recogniser and Relation Extraction tool that recognises **O**peration and **D**ata mentions, **No**rmalises them and identifies their **R**elations in the literature.

Identifying the ***data*** and ***operation*** entities is the first step to construct the workflow data events of a research article.

After finding the components, it is important to find the relations between them, what are the ***input data*** and ***output data***, whether they are retrieved or deposited to a mentioned ***database***, and whether the operation is done by stated ***software***.

Once the entities and their relations are identified in the context of an event, the data event can be constructed. The series of the data events are then linked in order to form the workflow of the research paper's method.

A layer of normalising the ***data***, ***operations*** and ***software*** is added to link the data events to well-known resources.

4.2.1 Corpus creation

PubMed central [135] was searched for articles that were published between 1^{st} January 1998 and 31^{st} March 2021 using the MeSH term *microarray analysis*. In addition, to obtain full text articles, we restricted the PubMed search to include only those in the PubMed Central Open Access subset. The query used was:

```
``microarray analysis''[MeSH Terms] AND (``1998/01/01''[PubDate] :
``2021/03/31''[PubDate]) AND ``open access''[filter]
```

25 documents were randomly chosen for training and test purposes. Training and testing data were divided as 75% and 25%, respectively.

4.2.2 Data preparation

The ***data*** and ***operations*** were manually annotated as collaboration annotation, based on the rules in the guideline attached in Additional file 1. BRAT format annotated corpus, 25 Methods sections, is available in Additional file 2

Following is the definition of the entities and inclusion and exclusion criteria. Figure 4.3 shows the steps of data and operation annotations.

Operation description

In our context, the operations are a specific/concrete computational process conducted in an experiment and presented in a manuscript as a part of the work done by a paper. The operations could have data inputs and produce outputs data. It also could have either input or output data and in some cases it could be mentioned without any data but an indication that a process is done. We also has a kind of operation that did not do any kind of processing but assigns the data a role or identification in the context.

Operation exclusion criteria:

The following types of operations were not included in the operation annotation:

- Biological processes conducted on real/physical data.
- Non analytical context processes such as the process that explain how a software works in general.

Data description

Data are items (or collections of items) that exist in digital form (i.e. representation), and can be potentially used as an input to or output from DB, software that processes those data to fulfil a specific/concrete computational processes that is conducted in an experiment and presented in a manuscript. This description includes any used data in the experiment although no explicit operations are mentioned in the sentence context.

Data exclusion criteria:

The following types of data were not included in the data annotation:

- The real/physical data.
- Hypothetical mentions such as abstract data that are used for conceptual and explanation purposes (e.g., discussion of strings, integers, etc.).
- Data in files/tables/figures or in the supplementary files/tables/figures.

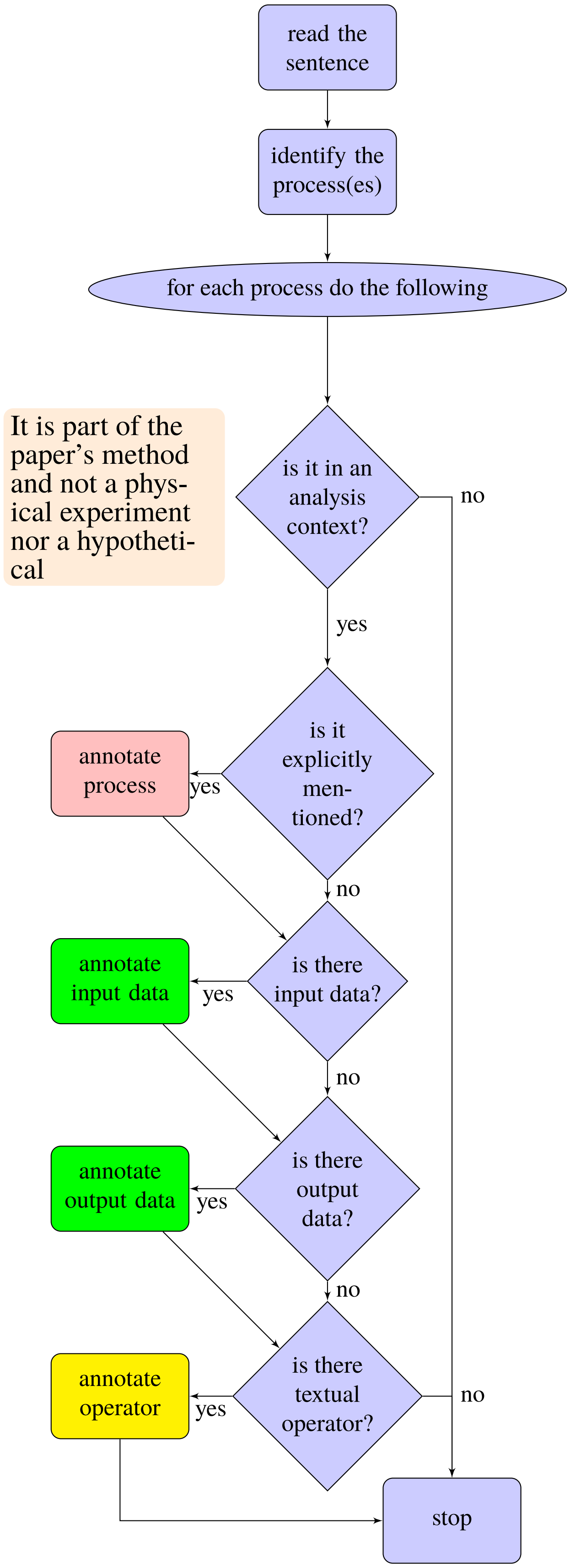

read the sentence
identify the process(es)
for each process do the following
It is part of the paper's method and not a physical experiment nor a hypothetical
is it in an analysis context?
no
yes
is it explicitly mentioned?
annotate process
yes
no
is there input data?
annotate input data
yes
no
is there output data?
annotate output data
yes
no
is there textual operator?
annotate operator
yes
no
stop

Figure 4.3. A flowchart shows the steps of annotation.

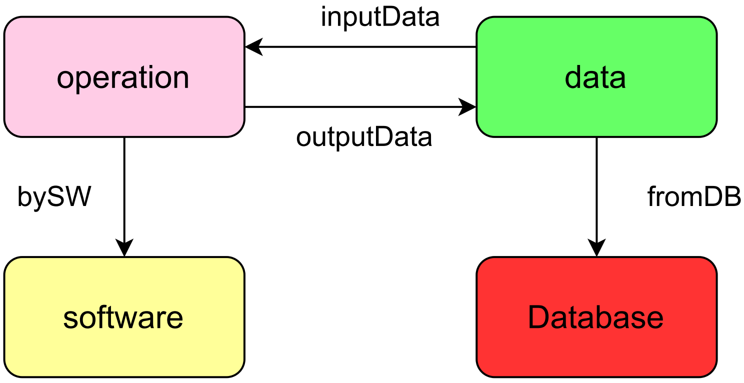


Figure 4.4. Four types of relations between extracted entities.

Relations description

As well as the two entities, we defined and manually annotated four types of relations (see Figure 4.4):
inputData, which defines a relation from ***data*** to an ***operation***.
outputData, which defines a relation from an ***operation*** to ***data***.
bySW, which defines a relation from an ***operation*** to ***software***.
fromDB, which defines a relation from ***data*** to a ***database***.

4.2.3 Automated annotation

We tested the usage of 4 systems and compared their capabilities in annotations; rule-based, BiLSTM-CRF, BERT and BioBERT. The four systems were tested and analysed separately. We are interested to see the performance over these systems and examine if the current state-of-art in biomedical named entity recognition BioBERT will produce better result than the BERT. We also would like to examine the performance of BiLSTM-CRF as one of previous NER state-of-the-art on CoNLL 2003 dataset [154].

Rule based

The texts were lexically analysed through the Sketch Engine tool [155] before making the rules that aimed to recognise the mention of an ***operation*** performed on the ***data***. The rules are written through the GATE JAPE environment [156]. Figure 4.5 shows an abstract representation of the main steps of the rule based system. The most important rules for annotating the ***operations*** are:

1. Test the verb of the sentence

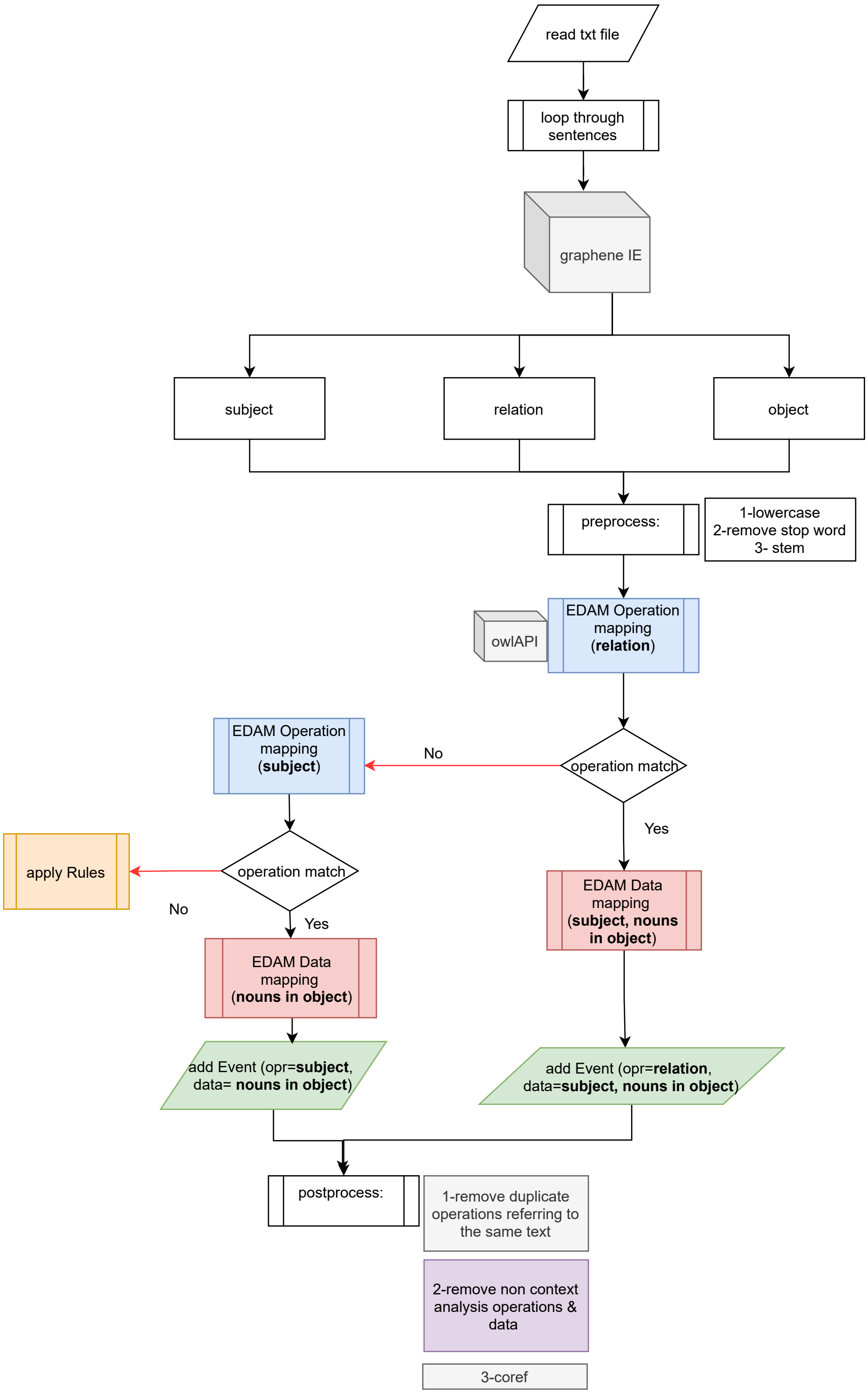

read txt file
loop through sentences
graphene IE
subject
relation
object
preprocess:
1-lowercase
2-remove stop word
3- stem
owlAPI
EDAM Operation mapping (**relation**)
operation match
No
Yes
EDAM Operation mapping (**subject**)
apply Rules
operation match
No
Yes
EDAM Data mapping (**subject, nouns in object**)
EDAM Data mapping (**nouns in object**)
add Event (opr=**subject**, data= **nouns in object**)
add Event (opr=**relation**, data=**subject, nouns in object**)
postprocess:
1-remove duplicate operations referring to the same text
2-remove non context analysis operations & data
3-coref

Figure 4.5. Abstract of the rule-based system.

1.1. The verb refers to a process:

The verb is either syntactically or semantically similar to one of the EDAM Operation concepts. A list of words (in Appendix B) was created for the ***operations*** that fail to syntactically match one of the EDAM concepts, e.g.

the `Data retrieval` operation could be referred to by the verbs *download, obtain, get, accessed, etc...* There are 13 hard-coded mapping rules.

1.2. The verb of action:

The process could be preceded by a verb of action, e.g. verbs such as "*performed*", "*done*", "*employed*", "*conducted*", etc.. It comes with variant context, e.g. "*we performed an operation on data*" or "*we used a tool to perform the operation*" or "*a tool is used to perform the operation*".

1.3. Verb phrases starting with "used to":

Verb phrases starting with "*used to*" are frequent in the corpus. Examples of these phrases are: "operation/software used to do an operation", "operation/software used to process data" or "operation/software used to fulfil a task.

Example 3 (PMC4149277)
"A BRB-ArrayTool plug-in was used to perform regression analysis of time course expression data to identify the genes whose expression varies over time."

2. Test the subject of the sentence

2.1. The subject is the operation:

In this case, the verb is often an action verb such as "*performed*", "*done*", "*employed*", the subject is usually an expression of a process. The subject is first checked as to whether or not it is correctly mapped to an EDAM concept, since some subjects could be tools, software or methods performed.

2.2. The subject could be a composite of ***operation*** and ***data***. *e.g. the analysis of data is performed.*

2.3. The subject is a tool and used to perform the ***operation***

Example 4 (PMC3735399)
"Pathway enrichment analysis of interaction network GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR 0.05. GENECODIS is a function analysis tool of gene, and it integrates different information resources (GO, KEGG or SwissProt), searches and arranges gene set annotation by statistical significance"

Some of the ***data*** rules depend on first finding the ***operation***. If there was no ***operation*** performed, the inference of using ***data*** in the study or ***data*** resulting from previously mentioned ***operations*** were checked.

Example 5 (PMC2374988)
"The microarray breast cancer datasets considered in this work are described elsewhere [5,7,9,18,19]. For these cohorts we used the normalized data, which are available in the public domain (see [5,7,9,18,19])."

The most important rules in determining the ***data*** are:

1. Where the verb is an ***operation***, the subject is usually the ***data***. More ***data*** could occur in the rest of the sentence.
2. Where the subject is the ***operation***, we looked for ***data*** in the object and the rest of the sentence.

   Analysis of the preposition showed with what it frequently comes. e.g.

   - "*at*" is followed by ***data*** describing the condition of the ***operation***,
   - "*on*" usually comes with ***data***,
   - "*by*" either comes with either v-ing detailing the ***operations***, or using data/tools/ software or mentions of a tool name.
3. Since the ***data*** and method/software/tools could occur in the same place, post-processing is performed to make sure the tagged text was not a tool. We checked the annotation in both the bioNerDS and bio.tools.
4. The noun mentions were also checked in EDAM Data ontology and ePMC annotation resources to see if they matched any known data instance.

BiLSTM-CRF

We used a pre-trained model from NeuroNER [157]. We used NeuroNER because it is a well-documented and user-friendly named-entity recognition tool that accepts BRAT annotation. It is based on long short-term memory (LSTM) [158] and contained in three layers: "The character-enhanced token-embedding layer, Label prediction layer, and Label sequence optimization layer.". The parameters of the configuration file were fine-tuned to use a pre-trained model to annotate the data. We enabled using the CRF layer and used the PMC embedding file for pre-trained token embeddings. We left the default value for other parameters. It takes training, validation, and testing data in the BRAT format as inputs.

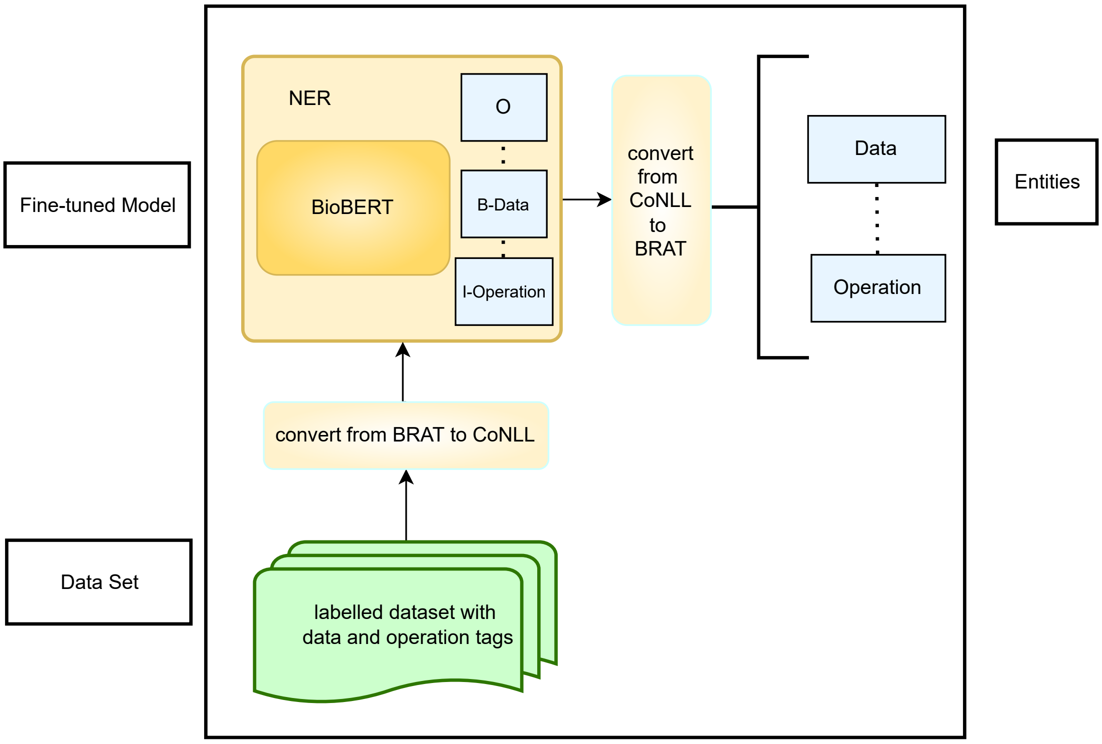

Fine-tuned Model
Data Set
NER
BioBERT
O
B-Data
I-Operation
convert from CoNLL to BRAT
Data
Operation
Entities
convert from BRAT to CoNLL
labelled dataset with data and operation tags

Figure 4.6. Supervised learning approach on our labelled data using the pre-trained model BioBERT. This model was fine-tuned on English version of the standard CoNLL-2003 NER dataset that uses BIO (Begin, Inside, Outside) tagging scheme. We did a modification to make it compatible with out BRAT annotation.

BERT

Bidirectional Encoder Representations from Transformers (BERT) [49] is a new open source model released by Google. It has pre-trained models that are trained on text-books and Wikipedia. It deals with inputs as WordPieces rather than tokens and uses multi-head attentions for faster and better performance. We used BERT Base, which has 12 encoder layers, 768 hidden units and 12 attention heads. The deep learning framework used is TensorFlow [46]. This model was fine-tuned on English version of the standard CoNLL-2003 Named Entity Recognition dataset [82] that uses BIO (Begin, Inside, Outside) tagging scheme. Since our annotation is in BRAT format, we first convert the BRAT to CoNLL format, perform the NER based on BRAT model, then finally convert the CoNLL result into BRAT format.

BioBERT

BioBERT [50] is a pre-trained model that is initialised with the BERT weights and then trained on PubMed abstracts and PMC full-text articles. It achieved better results in tagging biomedical articles than the normal BERT on three biomedical text mining tasks: biomedical named entity recognition, biomedical relation extraction and biomedical question answering. On biomedical named entity, BioBERT achieved F-score between 72.24% and 93.47% on BC4CHEMD dataset for drug and chemical. Figure 4.6 shows how we fine-tuned BioBERT on our labelled data to have a model suitable to our task. As mentioned above in BERT, the model was fine-tuned on CoNLL-2003 dataset [82] that uses BIO (Begin, Inside, Outside) tagging scheme. We did the same process for converting our BRAT annotation into CoNLL format, perform the NER based on BioBERT model, then finally convert the CoNLL result into BRAT format.

4.2.4 Integration with bioNerDS

BioNerDS [71] is a dictionary- and rule-based named entity recogniser for bioinformatics software and databases in the literature [127]. We used the improved version with post-processing filtering reported in [11] with an F-measure of 67% (precision 82%). We used bioNerDS to extract resources and we integrated it with our system and related the ***software*** with ***operations*** and ***databases*** with ***data***. We updated the dictionary by adding the new tools mentioned in bio.tools [153] to reflect up-to-date tools.

Since not all resources are explicitly classified as database or software, a classification process was done to find the type of the resource. The name of the annotated resource with a set of keywords used for annotation were tested to the regular expression (adapted from bioNerDS), then the dictionary was checked. The dictionary contains a hint if a SW or DB is found. In some cases, the resource matched more than one instance of a dictionary that can be software and database at the same time. In this case, the resource was tested against bio.tools; if there was a match, then it was SW. If not, then it was a DB. If there were no hints in the dictionary and regular expressions, the resource was tested to bio.tools and recorded as unknown if there was no match.

4.2.5 Normalisation

We used the EDAM ontology [34] (version 1.21) to normalise the annotated ***operations*** and ***data***. The EDAM Operation ontology is used for ***operations*** while ***data*** are normalised to the concepts in the EDAM Data and the EDAM Topic parts of the ontology. The OWL API [159] is used to deal with the ontologies.

We used cosine distance to test how the terms and concepts were syntactically similar. A threshold (0.51) was set to avoid far matches. Cosine similarity is calculated using only the dot product and magnitude of each vector, V1 . V2 / (|V1| * |V2|), and is therefore affected only by the terms the two vectors have in common. The vector representation is defined by Ukkonen [160] that based on finding common sub-strings of fixed length q (q-grams) between two strings. In our case, the q is 3. Cosine similarity takes into account the number of occurrences of each q-grams. Therefore, each string will be divided into q-grams, and the string profile will be the collections of q-grams along with the number of their occurrences. Only the value of the common q-grams are involved in calculating the Cosine similarity. We specifically used the implementation provided by the java-string-similarity library [161] that implemented different string similarity and distance measures such as Levenshtein edit distance, Longest Common Subsequence, cosine similarity, etc.. We compared the results obtained by a number of provided algorithms on samples of our dataset. We found that cosine similarity gives us the best result.

For operation normalisation, handcrafted rules were tested if there was no syntax match. There are 13 implemented rules with a list of words (in Appendix B) to relevant concepts. These rules are for concepts that are referred to with different terms e.g. `Data retrieval` concept can be expressed in the text as *downloading, obtaining or getting data*, calculation concepts could be expressed as *summing, subtracting, log transforming or averaging*.

We added `Assigning Role` as an additional operation concept in the EDAM Operation ontology. In some cases, the operation was not computational itself but it assigned a role to the data mentioned. For example, in the sentence: "*The data are included in the study*", the data here has no action performed on it but it has a role since it is used in the study. There was also another case as in the appearance of "*as*" where data was assigned a role as a named data

Example 6 (PMC4219025)
"We used Cytoscape [77] to visualize the networks with the strength of the gene-gene correlation as a co-factor."

Here *the strength of the gene-gene correlation* played a role of co-factor in the *Cytoscape* tool.

If the ***operation*** failed to be mapped syntactically or by the rules, the synonym set from WordNet was tested. We tested the similarity of the ***operation*** to the top 12 operation classes. The Leacock and Chodorow [162] algorithm was used to determine the semantic relevance of the two words. The algorithm tested the relatedness between the operation and the classes by examining the path lengths between them, as they represented in WordNet, and select the shortest the path as a measure of similarity:

$$LCH(s1, s2) = \frac{\text{-}\, Math. \log_e(LCS(s1, s2).length}{(2 * max\_depth(pos))}$$

We specifically used the implementation provided by WS4J (WordNet Similarity for Java) library [163] which counts up the number of edges between the senses in the 'is-a' hierarchy of WordNet. The value is then scaled by the maximum depth of the WordNet 'is-a' hierarchy. A relatedness value is obtained by taking the negative log of this scaled value.

If the mapping failed, the operation was mapped to the top class `Operation`. Algorithm 1 shows the steps of the ***operations*** normalisation.

For ***data*** mapping, we looked at both the `Data` and `Topic` ontologies and ePMC annotation. Although the Topic ontology has fewer classes than the Data ontology, the reason we normalised to the Topic ontology was the ***data*** used is referenced in a text in a general way, rather than specifying the item of data used and as it was listed in the EDAM ontology, e.g. *gene* mention was mapped to `Genes` as Topic it made more sense than mapping it to `gene ID`. If there was no `Data` concept detected, the ***data*** text was checked if it contained any indication of comparison (sign or word), then it was mapped to the `Experimental measurement data` concept. If there was no match, the ***data*** was mapped to the class `Data` in the Data ontology. We also add another layer of normalisation by checking if there is any EPMC annotation related to the annotated text.

4.2.6 Relation extraction

We defined and manually annotated four types of relations, as mentioned previously in Subsection 4.2.2: ***input data, output data, By software*** and ***from database***.

We used the system developed by Zhou et al. [89] which depends on Bidirectional Long short-term memory (BiLSTM) [158] and attention [48] to extract relations. It capture the most important semantic information in a sentence without the need to get high-level features from lexical resources such as WordNet or NLP systems like dependency parser and named entity recognizers (NER). The system was originally used for SemEval-2010 relation classification task [60] that contains 10,717 annotated examples covering nine relations: cause-effect, instrument-Agency, product-Producer, content-container, entity-Origin, entity-Destination, component-Whole,

Algorithm 1 Operation Normalisation.

```
procedure OperationNorm
    operationStem ← pre-process (operation)
    $dictionary$ ← WordNet dictionary
loop through annotated entities:
    top:
    $map \leftarrow EDAMmapping()$
    if map not null then:
        $normRule \leftarrow'' Syntax''$;
        goto loop.
    $map \leftarrow OperationRules$ (see Algorithm 2)
    if map not null then:
        $normRule \leftarrow'' Rules''$;
        goto loop.
    if operationlen $> 1$ then:
        goto top (Test the last word).
        if failed then:
            goto top (Test the first word).
    if failed then:
        do semantic similarity (synonym set)
        $normRule \leftarrow'' synonym''$;
    if failed then:
        map to top class operation
        $normRule \leftarrow'' Topclass''$;
    goto loop.
    close;

```

Algorithm 2 Rules of Syntax match for ***operations***.

```
procedure OperationRules
    operationStem ← pre-process (operation)
    $threshold$ ← 0.9
    $dictionary$ ← WordNet dictionary
top:
    $Verbs$ ← list of verbs lemma
loop:
    $cosineDistance$ ← The cosine distance between operation lemma and Verbs(i) lemma
    if cosineDistance $>$ threshold then return true
        goto loop.
        close;
    goto top.
```

member-Collection and Communication-Topic. There is an additional relation *other* that express the relations that are not defined. Att-BiLSTM outperformed the previous task results but its F-score, 84%, is overtaken by the relation extraction models that adapt transformers technology, with F-score of 91.9%. Although transformer-based approaches achieved higher scores in relation extraction, Att-BiLSTM achieved better result on our dataset. We modified the Att-BiLSTM model to extract input and output data relations. The format accepted should present each sentence with the two entities that have a relation. Each sentence should have only one relation. Since the

Table 4.2. Operations annotation results.

Model	**Trained Data**			**Test Data**		
	Prec	**Recall**	F_1**-Scores**	**Prec**	**Recall**	F_1**-Scores**
Rule Based	77.98	70.43	74.01	71.50	63.88	67.48
BiLSTM-CRF	94.58	89.15	91.79	85.93	76.82	81.12
BERT	99.15	97.29	98.21	93.38	85.98	89.52
BioBERT	97.69	98.94	98.31	93.38	91.56	92.46

Table 4.3. Data annotation results.

Model	**Trained Data**			**Test Data**		
	Prec	**Recall**	F_1**-Scores**	**Prec**	**Recall**	F_1**-Scores**
Rule Based	50.63	44.81	47.54	52.59	41.22	46.21
BiLSTM-CRF	90.01	84.54	87.19	73.46	54.58	62.63
BERT	98.41	92.89	95.57	78.17	70.70	74.25
BioBERT	93.86	98.94	96.33	72.87	84.15	78.10

original sentence could have more than ***data*** or ***operations***, the duplicate of a sentence was made to include only one relation between each mention of ***data*** and ***operation***. We assumed that every mention of ***data*** and operation in the same sentence had a relation. This scenario is true if there is only one operation mentioned in a sentence, but when more than one operation is present, it is not always true. The ***data*** mention could have a relation with two ***operations*** if it is an output data of one operation and an input of another. In some cases ***data*** could be an input for one and have no relation with the following ***operation***.

4.2.7 Representation

The current annotated and normalised terms are stored in BRAT [113] format to enable web-based representation. This includes the annotated entities and the relations. They are also available in XML and CSV format.

4.3 Results

4.3.1 Annotation

Table 4.2 shows the F-score results of the ***operations*** annotation in all four systems and table 4.3 shows the results of the ***data*** annotation. The best F-scores, 92.46% for ***operations*** and 78.10% for ***data*** annotations, were achieved by the model based on the BioBERT.

Table 4.4. Statistics describing the manually annotated corpus.

Characteristics	**Value**
Total number of documents	25
Number of sentences	740
Total operations mentions	621
Total data mentions	1,041
Average mentions of operation per document	25
Average mentions of data per document	42
Maximum mentions of operation in a single document	50
Maximum mentions of data in a single document	420

Table 4.5. BioNerDS evaluation results.

	Prec	**Recall**	F_1**-Scores**
Databases	40.43	55.88	46.91
Software	73.73	65.80	69.19

Table 4.6. Normalisation evaluation results.

	Prec	**Recall**	F_1**-Scores**
Operations	95.40	1	97.64
Data	84.35	1	91.51
Software	1	75.58	86.09

Table 4.4 shows the number of annotated entities in the manually annotated corpus.

4.3.2 Integration with bioNerDS

The annotated software and databases are manually evaluated and have the F-scores shown in Table 4.5.

4.3.3 Normalisation

Table 4.6 shows our normalisation evaluation. The accuracy of the ***operations*** mapping is 95.40%, which implies the EDAM Operation ontology is a good knowledge resource that represents the operational concepts.

More than 40% of ***data*** mentions (unique text) were mapped to the top class `Data`. This is because all mapping choices failed to find a match for the ***data*** mention and we linked it to the top class `Data`. This shows that a large part of ***data*** failed to find a class match.

The low percent of mapped ***data*** entities reflects the way the ***data*** are expressed by, and annotated, in the text. The data in the context of input to, or output from, an ***operation*** are mentioned in a more descriptive way than in a specific instance of data. Moreover, the same data are referred with different names in different sentences; for example, *the cell file, the data* and *the cells' information* could refer to the same instance of ***data*** in the text.

Although the ***data*** mapping is correct, it is not precise (specific). The classes in the Data ontology refer to a specific id, while it is just mentioned as a name in the text. For example, the ***data*** called *the prob* mapped to the correspondence concepts in EDAM Data ontology as `dbProb ID`, which is not precise. The matched Topic is `Probes` which presents a more relevant corresponding class. The same goes for *samples*; the `sample ID` is the Data ontology, while there is no mentioned ID. The `samples` Topic concept is more relevant.

There are some imprecise classes matched with the annotated entities. Using a synonym set for normalisation produces most of the false positives, e.g. *run* is mapped to `Calculation` where it is an operation. The syntax mapping also match the terms with imprecise classes, e.g. any *assign* is mapped to `NOE assignment`, *measure* is mapped to `Neurite measurement`, and for data referred by the word *value*, the class `E-value` is chosen.

The following is an example of the annotated ***operations*** and ***data*** with corresponding EDAM's Operation and Data classes for Methods section of article with PMCID PMC3735399. The Methods section of the paper is in Appendix C (Subsection C.1.1).

Annotated *operations* with normalisation

Sentence: *"GSE35957 was downloaded from Gene Expression Omnibus (GEO) database (http://www.ncbi.nlm.nih.gov/geo/), which is based on GPL570 [HG-U133_Plus_2] Affymetrix Human information Genome U133 Plus 2.0 Array Platform (Affymetrix, Santa Clara, CA, USA)."*

1. downloaded, `Data retrieval`

   $http://edamontology.org/operation\_2422$

Sentence: *"Microarray probe annotation information was downloaded from the Affymetrix Company, including all AffymetrixATH1(25K) gene chip probe information, and the probe annotation information files of the platform."*

2. downloaded, `Data retrieval`

   $http://edamontology.org/operation\_2422$

Sentence: *"The original data were preprocessed by Affymetrix [7,8] package in R language."*

3. preprocessed, `Process`

   $http://edamontology.org/operation\_0004$

Sentence: *"LIMMA [9] package in R language was used to identify the differentially expressed genes between the expression profile of five osteoporosis patients and four non-osteoporosis samples."*

4. identify, `Entity identification`

   $http://edamontology.org/operation\_3280$

Sentence: *"Multiple testing correction was performed by Bayesian method [10]."*

5. Multiple testing correction, `Correlation`

   $http://edamontology.org/operation\_3465$ (wrong mapping, it should be `calculation`)

Sentence: *"An FDR <0.01 and |logFC| >1 were chosen as thresholds for screening the differentially expressed genes."*

6. chosen as, `Assigning Role`

   $http://edamontology.org/operation\_11111$

7. screening, `Virtual ligand screening`

   $http://edamontology.org/operation\_0482$

Sentence: *"Therefore, we used HitPredict software (`http://hintdb.hgc.jp/htp/`) to search the differentially expressed genes that can interact with OPG gene."*

8. search, `Search`

   $http://edamontology.org/operation\_2421$

Sentence: *"This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relationships."*

9. find, `Recognition`

   $http://edamontology.org/operation\_2423$

10. visualize, `Visualisation`

    $http://edamontology.org/operation\_0337$

Sentence: *"In this study, we used MCODE to mine the modules from the protein-protein interaction network with degree >2."*

11. mine, `Analysis`

    $http://edamontology.org/operation\_2945$

Sentence: *"Further, we used Bingo [13] to annotate each module based on the hypergeometric distribution (FDR <0.05)."*

12. annotate, `Annotation`

    $http://edamontology.org/operation\_0226$

Sentence: *"GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR <0.05."*

13. biological pathway enrichment analysis, `Enrichment analysis`

    $http://edamontology.org/operation\_3501$

Annotated *data* with normalisation

Sentence: *"GSE35957 was downloaded from Gene Expression Omnibus (GEO) database (`http://www.ncbi.nlm.nih.gov/geo/`), which is based on GPL570 [HG-U133_Plus_2] Affymetrix Human information Genome U133 Plus 2.0 Array Platform (Affymetrix, Santa Clara, CA, USA)."*

1. GSE35957, `Data`

   $http://edamontology.org/data\_0006$

Sentence: *"Microarray probe annotation information was downloaded from the Affymetrix Company, including all AffymetrixATH1(25K) gene chip probe information, and the probe annotation information files of the platform."*

2. Microarray probe annotation information, `Gene annotation (homology information)`

   $http://edamontology.org/data\_3148$

Sentence: *"The original data were preprocessed by Affymetrix [7,8] package in R language."*

3. The original data, `Data`

   $http://edamontology.org/data\_0006$

Sentence: *"LIMMA [9] package in R language was used to identify the differentially expressed genes between the expression profile of five osteoporosis patients and four non-osteoporosis samples."*

4. differentially expressed genes, `Gene ID`

   $http://edamontology.org/data\_2295$

5. the expression profile, `Gene expression profile`

   $http://edamontology.org/data\_0928$

Sentence: *"An FDR <0.01 and |logFC| >1 were chosen as thresholds for screening the differentially expressed genes."*

6. FDR <0.01, `Experimental measurement data`

   $http://edamontology.org/data\_3108$

7. $|$ logFC $|$ >1, `Experimental measurement data`

   $http://edamontology.org/data\_3108$

8. thresholds, `Experimental measurement data`

   $http://edamontology.org/data\_3108$

9. the differentially expressed genes, `Gene ID`

   $http://edamontology.org/data\_2295$

Sentence: *"Therefore, we used HitPredict software (`http://hintdb.hgc.jp/htp/`) to search the differentially expressed genes that can interact with OPG gene."*

10. differentially expressed genes, `Gene ID`

    $http://edamontology.org/data\_2295$

Sentence: *"This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relationships."*

11. protein-protein interactions, `Protein interaction ID`

    $http://edamontology.org/data\_1074$

12. high confidence, `Evidence`

    $http://edamontology.org/data\_2042$

13. interactions, `Atom interaction data`

    $http://edamontology.org/data\_0906$

14. differentially expressed genes, `Gene ID`

    $http://edamontology.org/data\_2295$

15. interaction relationships, `Protein interaction record`

    $http : //edamontology.org/data_$0906

Sentence: *"In this study, we used MCODE to mine the modules from the protein-protein interaction network with degree >2."*

16. modules, `Data`

    $http : //edamontology.org/data_$0006

17. protein-protein interaction network, `Protein interaction ID`

    $http : //edamontology.org/data_$1074

18. degree > 2, `Experimental measurement data`

    $http : //edamontology.org/data_$3108

Sentence: *"Further, we used Bingo [13] to annotate each module based on the hypergeometric distribution (FDR <0.05)."*

19. module, `Data`

    $http : //edamontology.org/data_$0006

20. hypergeometric distribution, `Dirichlet distribution`

    $http : //edamontology.org/data_$1347

21. FDR < 0.05, `Experimental measurement data`

    $http : //edamontology.org/data_$3108

Sentence: *"GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR <0.05."*

22. all genes, `Gene ID`

    $http : //edamontology.org/data_$2295

23. interaction network, `Atom interaction data`

    $http : //edamontology.org/data_$0906

24. FDR < 0.05, `Experimental measurement data`

    $http : //edamontology.org/data_$3108

Table 4.7. Relations results.

	Trained Data			Test Data		
	Prec	**Recall**	**F_1-Scores**	**Prec**	**Recall**	**F_1-Scores**
Input data	94	91	93	78	88	83
Output data	84	88	86	71	55	62
From database	77.61	1	87.39	72.72	1	84.21
By software	83.15	1	90.80	86.05	1	92.50

4.3.4 Relation extraction

Table 4.7 shows the relations evaluation result. We used the Att-BiLSTM for extracting the relationship between ***data*** and ***operations***, whether an operation had input data, output data. It achieved F-scores of 83% and 62%, respectively.

We did a rule-based relation extraction to relate ***data*** with ***databases*** and ***operations*** with ***software***. It gave F-scores of 98.56% and 98%, respectively.

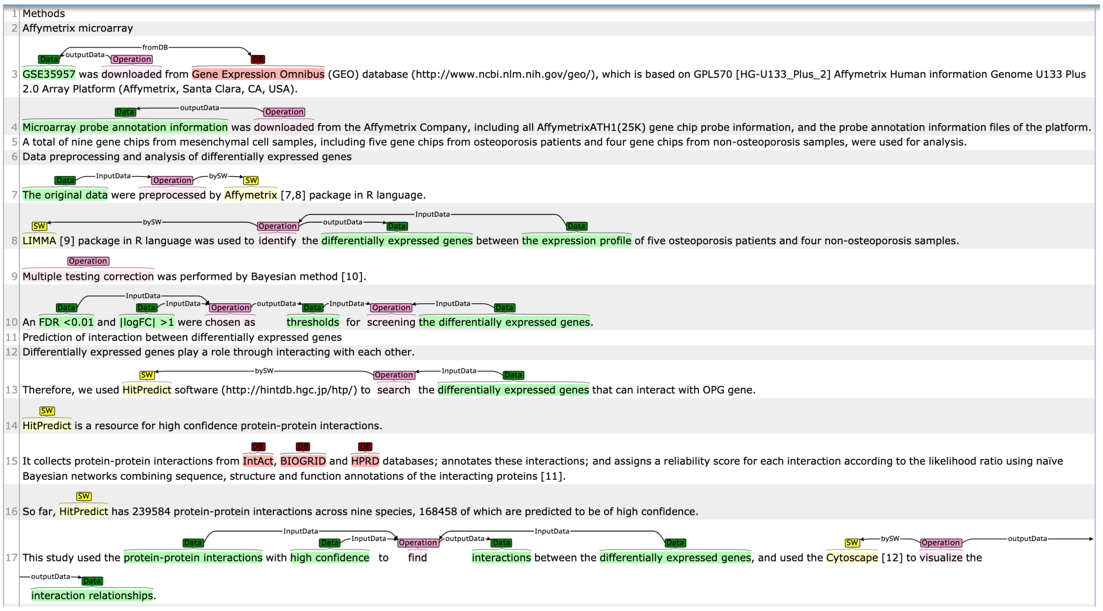
One of the chapter's contributions is presenting the possible ***operations*** of a tool. Although bio.tools presents this kind of information, our system presents a granular set of EDAM classes than referring to just the top classes and also discover any missed mentions. For example, the *R* tool has an association with 143 unique ***operations*** while it is expressed in bio.tools by three top classes of EDAM Operation ontology (`Analysis`, `Data handling` and `Visualisation`). It is also associated with the `Calculation` class that is not mentioned in bio.tools as an EDAM top `Operation` class.

4.3.5 Representation

The annotation of ***data***, ***operations***, ***software*** and ***databases*** with relations among them is represented in the BRAT format. Figure 4.7 shows the annotation with relations of the article with PMCID PMC3735399. Figures 4.8, 4.9 and 4.10 shows the normalisation representation in BRAT.

4.4 Discussion

Developing rule-based systems was challenging and time-consuming. The rule-based system gives moderate results (poor result for ***data***). The syntax structure of the texts is varied among the corpus. Implementing basic rules of analysing $<$ subject, verb, object $>$ could not be generalised well to include all varieties of the text. More flexible rules could have annotated meaningless data, while strict rules increased the accu-



1 Methods
2 Affymetrix microarray
3 GSE35957 was downloaded from Gene Expression Omnibus (GEO) database (http://www.ncbi.nlm.nih.gov/geo/), which is based on GPL570 [HG-U133_Plus_2] Affymetrix Human information Genome U133 Plus 2.0 Array Platform (Affymetrix, Santa Clara, CA, USA).
4 Microarray probe annotation information was downloaded from the Affymetrix Company, including all AffymetrixATH1(25K) gene chip probe information, and the probe annotation information files of the platform.
5 A total of nine gene chips from mesenchymal cell samples, including five gene chips from osteoporosis patients and four gene chips from non-osteoporosis samples, were used for analysis.
6 Data preprocessing and analysis of differentially expressed genes
7 The original data were preprocessed by Affymetrix [7,8] package in R language.
8 LIMMA [9] package in R language was used to identify the differentially expressed genes between the expression profile of five osteoporosis patients and four non-osteoporosis samples.
9 Multiple testing correction was performed by Bayesian method [10].
10 An FDR <0.01 and |logFC| >1 were chosen as thresholds for screening the differentially expressed genes.
11 Prediction of interaction between differentially expressed genes
12 Differentially expressed genes play a role through interacting with each other.
13 Therefore, we used HitPredict software (http://hintdb.hgc.jp/htp/) to search the differentially expressed genes that can interact with OPG gene.
14 HitPredict is a resource for high confidence protein-protein interactions.
15 It collects protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction according to the likelihood ratio using naïve Bayesian networks combining sequence, structure and function annotations of the interacting proteins [11].
16 So far, HitPredict has 239584 protein-protein interactions across nine species, 168458 of which are predicted to be of high confidence.
17 This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relationships.

Figure 4.7. An example of annotation with relations in BRAT format.

racy but reduced the recall. One of the limitations of the system is that it depends on the verb of the sentence. In the case where there are two verbs, but they are not recognised as being two main verbs, one operation will be missed. Another limitation is that the list of the verbs that implies an operation meaning should be collected prior to the annotation if it is not syntactically similar to the ***operations*** in the EDAM Operation ontology.

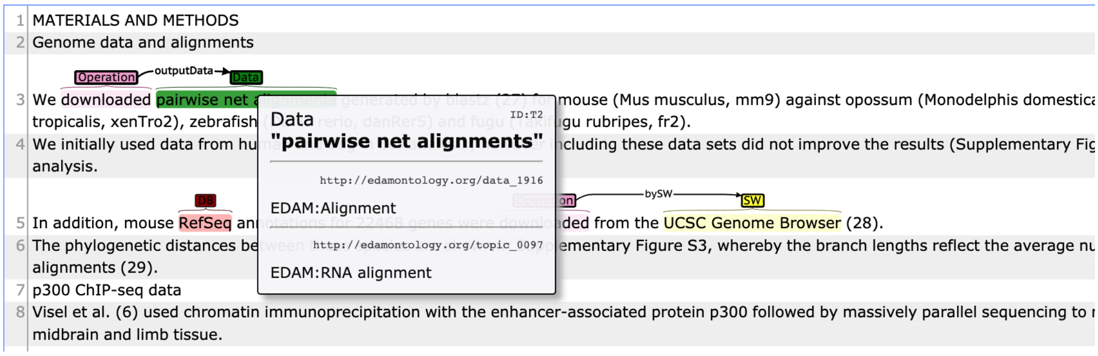
Moving to neural networks and transformers gave better results when the training was on the biomedical corpus. The BioBERT system gave the best result in annotating both the ***operations*** and ***data***.

4.4.1 Ambiguity

We encountered some challenges in annotating the ***operations*** and ***data***. Here are some examples of the ambiguous cases:

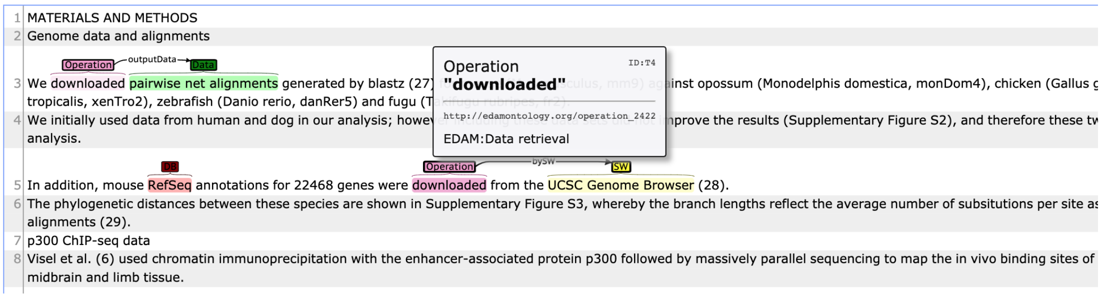
1. There are some entities with the same syntax but could be ***operations*** or ***data*** based on the context they are mentioned in.

   Example 7 (PMC4438953)
   "Complete amino acid sequences were downloaded from NCBI database in FASTA format and alignments were performed using ClustalW. Subsequently, alignments were adjusted using Bioedit 7.0 software with 60 % threshold for homology."


MATERIALS AND METHODS
Genome data and alignments
We downloaded pairwise net a… mouse (Mus musculus, mm9) against opossum (Monodelphis domestica
tropicalis, xenTro2), zebrafish … rerio, danRer5) and fugu (Takifugu rubripes, fr2).
We initially used data from hu… ncluding these data sets did not improve the results (Supplementary Fig
analysis.
In addition, mouse RefSeq an… otations for 22468 genes were downloaded from the UCSC Genome Browser (28).
The phylogenetic distances between … ementary Figure S3, whereby the branch lengths reflect the average nu
alignments (29).
p300 ChIP-seq data
Visel et al. (6) used chromatin immunoprecipitation with the enhancer-associated protein p300 followed by massively parallel sequencing to
midbrain and limb tissue.

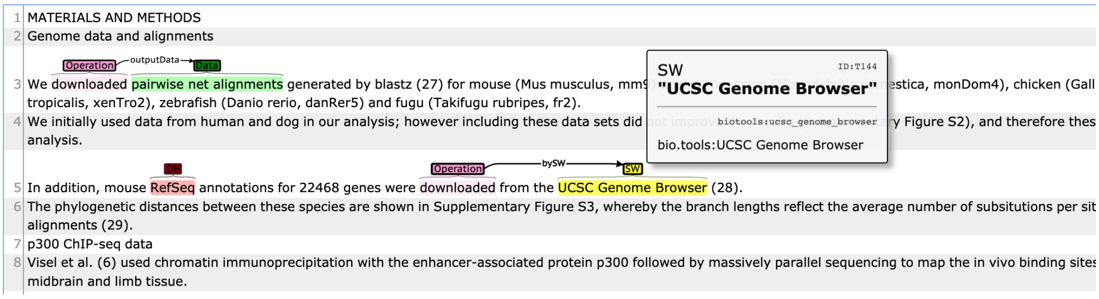
Data ID:T2
"pairwise net alignments"
http://edamontology.org/data_1916
EDAM:Alignment
http://edamontology.org/topic_0097
EDAM:RNA alignment

Figure 4.8. Data Normalisation in BRAT.


MATERIALS AND METHODS
Genome data and alignments
We downloaded pairwise net alignments generated by blastz (27) … against opossum (Monodelphis domestica, monDom4), chicken (Gallus g
tropicalis, xenTro2), zebrafish (Danio rerio, danRer5) and fugu (T…
We initially used data from human and dog in our analysis; howe… nprove the results (Supplementary Figure S2), and therefore these tw
analysis.
In addition, mouse RefSeq annotations for 22468 genes were downloaded from the UCSC Genome Browser (28).
The phylogenetic distances between these species are shown in Supplementary Figure S3, whereby the branch lengths reflect the average number of subsitutions per site as
alignments (29).
p300 ChIP-seq data
Visel et al. (6) used chromatin immunoprecipitation with the enhancer-associated protein p300 followed by massively parallel sequencing to map the in vivo binding sites of
midbrain and limb tissue.

Operation ID:T4
"downloaded"
http://edamontology.org/operation_2422
EDAM:Data retrieval

Figure 4.9. Operation Normalisation in BRAT.


MATERIALS AND METHODS
Genome data and alignments
We downloaded pairwise net alignments generated by blastz (27) for mouse (Mus musculus, mm9 … estica, monDom4), chicken (Gall
tropicalis, xenTro2), zebrafish (Danio rerio, danRer5) and fugu (Takifugu rubripes, fr2).
We initially used data from human and dog in our analysis; however including these data sets did … y Figure S2), and therefore thes
analysis.
In addition, mouse RefSeq annotations for 22468 genes were downloaded from the UCSC Genome Browser (28).
The phylogenetic distances between these species are shown in Supplementary Figure S3, whereby the branch lengths reflect the average number of subsitutions per sit
alignments (29).
p300 ChIP-seq data
Visel et al. (6) used chromatin immunoprecipitation with the enhancer-associated protein p300 followed by massively parallel sequencing to map the in vivo binding sites
midbrain and limb tissue.

SW ID:T144
"UCSC Genome Browser"
biotools:ucsc_genome_browser
bio.tools:UCSC Genome Browser

Figure 4.10. Software Normalisation in BRAT.

Alignment in the first sentence is an ***operation***, but *alignments* in the second sentence are ***data***.

Example 8 (PMC2919724)
"A global score for our reanalysed dataset and the original analysis was computed as the sum of log-scores for each individual TF."

Analysis often comes as an ***operation***. However, in this sentence, *the original analysis* is ***data***.

2. The ***data*** instances are not always mentioned as a singular or a compound name but as a long description.

Example 9 (PMC4268857)

"Analyses were performed on three data sets:

1. The publicly available TCGA, from which HE images, mRNA expression, genomic data including copy number variation and DNA methylation data, and corresponding clinical information for 489 HGSOC patient samples were obtained. This was downloaded from the TCGA data portal [4] and cBioPortal [25].

2. SEARCH data set with tissue samples and corresponding clinical information for 245 HGSOC samples (out of 516 ovarian cancer cases; Table 1). Patients were recruited after a diagnosis of ovarian cancer and if they were able to consent for participation in the study [3]. Key demographical and clinical data on the patients, including BRCA1 germ-line mutation status, were presented anonymously.

3. Nottingham Ovarian Cancer Study (NOT) data set with tissue samples and corresponding clinical data from 276 HGSOC samples (out of 507 ovarian cancer cases; Table 2). This is a retrospective study of ovarian cancer cases diagnosed between 1991 and 2011 [3]. For this study, the institutional research ethics boards (East Of England Cambridgeshire REC (for SEARCH) and Derbyshire REC (for NOT)) waived the need to obtain consent. Both local human research investigation committees approved each study. "

The first sentence is followed by three points that contain many sentences explaining the ***data*** used and where they are obtained and some metadata about them.

3. The context that explains the ***operations*** could start with the purpose of using an ***operation*** or how to do an ***operation*** followed by a series of other ***operations***.

Example 10 (PMC4438953)

"GO analysis was used to analyze the predominant functions of the differentially expressed genes according to the GO, which is the key functional classification of the NCBI (18,19). Fisher's exact test and $\chi 2$ test were used to classify the GO category, and the FDR (20) was calculated to correct the P-value; the smaller the FDR, the smaller the error in judging the P-value."

There is going to be a redundancy if we annotate both analysis references *the GO analysis* and *analyze* because they refer to the same ***operation***. We cannot apply the same rule for the second sentence. *The FDR was calculated* calculation is a separate ***operation*** done before the following ***operation*** correction in context *correct the P-value*

Example 11 (PMC2714961)

"We used the Oncomine database http://www.oncomine.org/main/index.jsp to conduct a meta-analysis of the number of studies comparing gene expression in normal prostate tissue with that of localized prostate tumor tissue [12]. The complete list of the studies used in the meta-analysis can be found in the supplementary materials (Table S1)."

In the first sentence, *Oncomine database* is not the medium used to do the *meta-analysis* ***operation*** as the syntax could suggest. It is the source of the data, as we can implicitly understand that from the following sentence since there is no explicit mention of data retrieval. This first sentence is an introductory sentence rather than a sentence that contains an ***operation*** or ***data***.

4.4.2 Limitations

1. The annotation and normalisation of ***data*** could be enhanced. One suggestion is that by excluding any data description, a limited number of words are expressing the data. Another is to develop more rules for data normalisation by studying the EDAM Data classes.
2. Enhance the relation extraction part by oversampling output data and adding "Other" relation to express no relation between a mention of ***data*** and an ***operation***.
3. Post-processing of the annotated terms could be done to improve the annotation accuracy, such as splitting the combined nouns, removing the annotation of the subtitles and removing the acronyms from ***data***.
4. The relation between the ***data*** and ***database*** did not show if the ***database*** was used as a sink or data source.

4.5 Conclusion

The ODNoR tool can get the key information mentioned in the Methods section, normalise them and find the relations between them. The ***data*** and ***operations*** are automatically annotated from the literature using BioBERT with a high accuracy. The relations between input, output data, operations, software and databases are expressed. The ***operations*** and ***data*** are normalised to EDAM Operation, Data and Topic ontologies. The software entities are normalised to bio.tools IDs. Now we have the components of the method, we can proceed with the process of a method extraction and form an abstract that combines these components in a flow of events.

4.6 Additional Files

4.6.1 Additional file 1 — Annotation Guideline

4.6.2 Additional file 2 — Annotated Corpus

Chapter 5

ODNoRFlow: Automated Reconstruction of Data Workflows from the Computational Biology Literature

Abstract

Background Understanding the work conducted in research papers is crucial to validate, evaluate and replicate it. With a growing number of publications, an additional challenge is to understand research methodologies described in prose. Representing them as smaller steps that contain the main parts of the research workflow in a computationally amenable form could facilitate faster understanding and enable replication.

Objectives This chapter aimed to explore how to automatically reconstruct data workflows that represent the methodology as presented in each research paper in the microarray analysis literature. We also explore how a discourse analysis can improve the result of the extracted representations.

Methods We introduce and evaluate ODNoRFlow, which processes the Methods section of a paper and produces linked data events that express the steps of the work reported in the paper as an abstract workflow. The system is applied to the Methods section in a corpus of 25 papers selected randomly from the microarray analysis literature.

We analysed the result of two types of discourse analysis, functional discourse and RST, by comparing the discourse function or relation to the extracted data events to build a pattern that help in improving the workflow precision result.

Results We extracted 25 abstract workflows from the used data set. We evaluated the quality of the extracted data events and obtained an F-measure was between 93.56% and 61.04% with a good inter-annotator agreement. We evaluated the possibility of using discourse analysis to improve the quality result of the extracted data events and found that the patterns extracted are not sufficient to filter out unrelated data events.

Conclusion We demonstrated the feasibility of automatically identifying and linking data events that express the steps of the work reported in the paper as an abstract workflow.

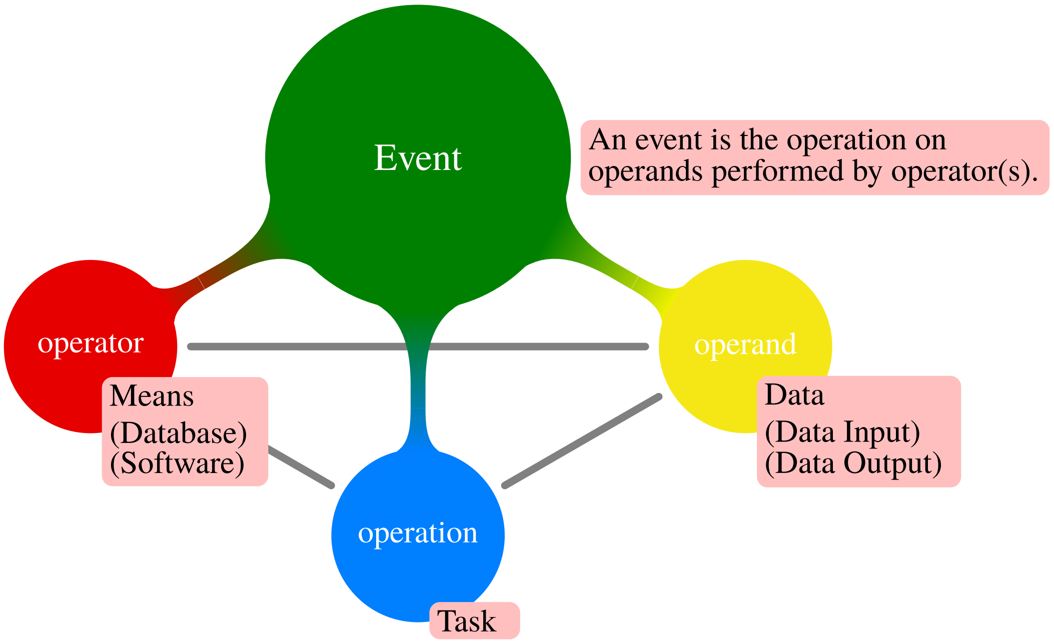
5.1 Introduction

The methods reported in the scientific literature are growing in quantity and complexity. Understanding and applying these methods requires huge efforts to explore them since they are not computationally formed. The key reason as for methodologies not being available in a computational form is that they are traditionally reported in the Methods section of a research paper as a free-text description of the steps of the work that has been reported, what components are involved and why they were chosen [7]. If methods were computationally extracted and represented from published methodologies then researchers could have the opportunity to identify trending methods in a period of time for a given task and which new methods old ones are replaced with. Similarly, they would be able understand how these methods changed over time and be able to link between the method developments and how the methods evolve.

The aim of this paper is to explore how to automatically extract a computational representation of methods reported in the Methods section from the computational biology literature, in particular using microarray analysis as a case study. We have previously developed ODNoR (Chapter 4), a named entity recogniser for identification of mentions of computational biology operations and data in the literature. In this chapter we aim to organise the ***operations*** and ***data*** identified in an abstract data/method workflow for a single research article. Specifically, we introduce and evaluate ODNoR-Flow, which accepts free-text of the Methods section and produces linked data events that represent the steps of the work reported in the paper. A data event is an operation on data performed by an operator(s). In our context, operators can be software (e.g. BLAST), tools (e.g. Microsoft Excel) or different methods (e.g. Student's t-test). A representation of a data event contains an operation, data, software and databases involved in the data event. We then organise the entity mentions and their relationships in data events in an abstract workflow. This abstract workflow demonstrates the work reported in a paper — Figure 5.1 shows an abstract level of a data event.

We tested and evaluate the proposed approach on 25 papers in the area of microarray analysis. For each abstract workflow, we compare all the extracted data events to the manually expected ones and score each data event based on the quality of the extraction, good, partial, missing or spurious. We studied the feasibility of using discourse analysis with the data events to improve the quality of the constructed workflow. Two types of discourse are used, the functional discourse and rhetorical structure theory.

Figure 5.1. Abstract level of an event



5.2 Materials and Methods

ODNoRFlow uses the output of the ODNoR system to reconstruct an abstract workflow of a paper. Once the entities and their relations are identified in the context of a single event by ODNoR, the data event is constructed. A series of the data events are linked in order to form the workflow of the paper. Figure 5.2 represents a high-level overview of ODNoRFlow, which comprises of the following steps:

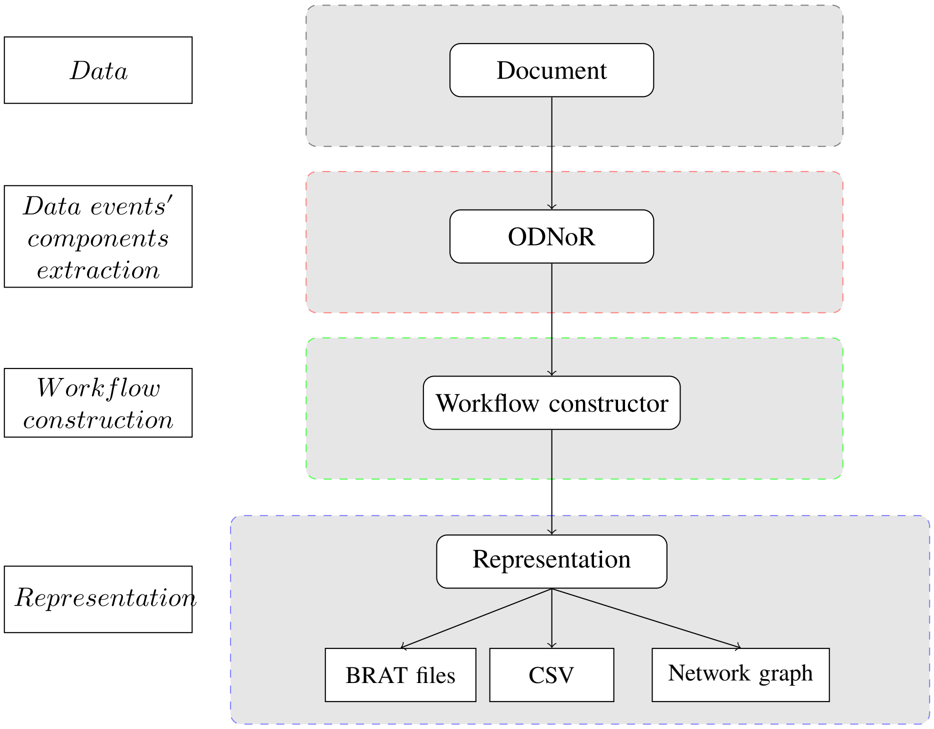
1. Data preparation
2. Data events' components Extraction
3. Workflow construction
4. Workflow representation

We then evaluated the extracted workflows and explored the role of discourse analysis in filtering out false positives.

5.2.1 Data preparation

PubMed central [135] was searched for articles that were published between 1^{st} January 1998 and 31^{st} March 2021 using the MeSH term *microarray analysis*. In addi-

Figure 5.2. System workflow.



tion, to obtain full text articles, we restricted the PubMed search to include only those in the PubMed Central Open Access subset. The query used was:

```
``microarray analysis''[MeSH Terms] AND (``1998/01/01''[PubDate] :
``2021/03/31''[PubDate]) AND ``open access''[filter]
```

25 documents were randomly chosen for training and test purposes. Training and testing data were divided as 75% and 25%, respectively. These are identical to the 25 documents that were used in Chapter 4.

5.2.2 Data events' components extraction

A data event consists of input and output data ("operands"), an operation, and software and databases ("operators") involved to accomplish a task. We used ODNoR (Chapter 4) to extract data event components, normalise them (e.g. to EDAM), and find the links between them (e.g. which operand links to which operation).

Table 5.1. Data event template. This template is filled by the entities extracted and mapped to the source knowledge using the ODNoR system.

			Extracted text	**Mapping**
Event No.				
Semantic Roles	operand	input data		(EDAM Data, EDAM Topic, ePMC annotation
		output data		(EDAM Data, EDAM Topic, ePMC annotation
	operation	operation		*(EDAM Operation)*
	operator	software		*(bio.tools)*
		database		

5.2.3 Workflow construction

In ODNoRFlow, we organise the extracted entities and relations into data events, and link them to construct a paper's workflow. We defined a template for a data event that was inspired by the representation of events in discourse analysis [97]. We build the data event structure based on the suggestions reported by Chambers and Jurafsky [102]. They show the steps of creating template structure of events in a specific-domain that has no previous template. Table 5.1 shows the data event template that we aim to fill by the extracted key components and their relations.

We have two types of data event: (explicit) data events and implicit data events. In an (explicit) data event, there is an explicit data process done and reported in a sentence. An implicit data event is an event that represents the ***data*** involved in a step of study but without being accompanied by an ***operation*** in the same sentence. The process done on the data could be mentioned in a previous sentence and the ***data*** in the current sentence are the result of applying the previous process (e.g. *In total, 355 probes are included in this study.* [PMC2582621]). Another case of an implicit data event is when the ***data*** refer to a kind of usage not in a form of a process (e.g. *For some datasets, the linkage relied on Ensembl external database identifiers.* [PMC2374988]).

5.2.4 Workflow representation

The workflow can be stored and represented in three different formats: the BRAT format, the CSV format and as a directed network graph. In the BRAT format, the events are viewed with relations and entities as text annotation. The internal data are stored in .ann files. The CSV format are commonly used in analytics. The workflow can be also viewed as a directed network graph where the arrows show the flow direction between ***data*** and ***operations***, and between ***operations*** themselves. We used Gephi [114] to represent and manipulate the workflow.

5.2.5 Workflow evaluation

There are two aspects used to validate the data events: *completeness*, which relates to entities and relations inside a data event, and *sequentiality of events in a workflow*, which relates to the order of the data events in the workflow.

For completeness, we want to consider how well the extracted data events represents i.e. captures a data event against the manual annotation. We used lenient, intermediate and strict evaluations to obtain the F-scores.

To evaluate our system we classified the result of the events (i.e. annotation) to multiple categories. We adapted evaluation metrics introduced by the Message Understanding Conference (MUC) [164], taking into account different errors categories in the process of the evaluation. We aimed to express how well the extracted information (i.e. annotations) represents i.e. captures a data event against the manual annotation. The evaluation process classified each data event as:

- *Good*, where the annotation represents all the key aspects of an event as reported in a paper that are necessary to understand the event.
- *Partial*, where an event annotation partially reflects the key aspects of the event. Here, the extracted representation might miss some components, but it still gives us a reasonable understanding of the event.
- *Missing*, where an event annotation is not captured by the system. This means that the systems fails to give the expected annotation and hence a useful representation of the event.
- *Spurious*, where an event annotation has no relation with the event mentioned in a given part of paper (i.e. the sentence). In other words, the event annotation is not usable.

We used the MUC-defined metrics to estimate the quality of the automatic extraction of data event annotations, as specified by the following formulae:

$$Precision = \frac{(Good + (w * Partial))}{Good + Partial + Spurious}$$

$$Recall = \frac{(Good + (w * Partial))}{Good + Partial + Missing}$$

$$F - score = \frac{2 * Recall * Precision}{Recall + Precision}$$

$$w = \begin{cases} 1 & \text{Lenient evaluation} \\ 0.5 & \text{Intermediate evaluation} \\ 0 & \text{Strict evaluation} \end{cases}$$

We also calculated the inter-annotator agreement (IAA) between two different raters on 20% (5 out of 25) of the corpus , and then analyses the specific-category agreements between the raters. In categorical measurements (as we have here), specific agreement represents the degree of agreement observed across all possible categories. If two raters are involved, the specific agreement for any category is the probability of one rater assigning an item to that category given that the other rater has also assigned that item to that category. It will test how much the two raters are agreed and this will give an indication as to how far we can trust the result produced by the system [165]. The specific category agreement, proposed by Dice [166] and extended by Uebersax [167], was used to consider multiple raters and categories including missing data. We used the agreement software package implemented by Girard [168]. It is calculated as following:

$$SAk = \frac{\sum_{i=1}^{n'} r_{ik}(r_{ik} - 1)}{\sum_{i=1}^{n'} r_{ik}(r_i - 1)}$$

n' — the number of items that were coded by two or more raters

r_{ik} — the number of raters that assigned item i to category k

r_i — the number of raters that assigned item i to any category

For sequentiality of events in a workflow, the order is tested at a single sentence level and at the sentence above the one examined. We tested both the manually annotated data events and automatically extracted one, to verify that the description of the methods in the body of the text is a reflection of the way the methods were actually applied. The scoring is done as follows:

- False positive events receive a zero score.
- True positive events are tested if they come in an appropriate order while, ignoring any false positives before them.

- If there is an event incorrectly reported and the following event should precede it, both of them receive a zero mark.
- We encapsulate the incorrectly ordered events at a minimum span so they do not affect the score of the following events.
- We then calculate the percentage of the correct order events over all the reported events to obtain the score of sequentiality of events in a workflow.

5.2.6 Data events and discourse analysis

In Chapter 3 we examined the discourse analysis as part of the pre-processing step. We looked at two types of discourse Functional discourse and RST.

Functional discourse. Liakata et al. [132] developed an automatic annotation tool to extract the core scientific concepts (CoreSC) in life science articles. The details of the eleven core concepts (`Hypothesis`, `Motivation`, `Goal`, `Object`, `Background`, `Method`, `Experiment`, `Model`, `Observation`, `Result` and `Conclusion`) are described in [133]. Short definitions for these functions are listed in Table 5.2.

Table 5.2. Functional discourse concepts definitions from [133]

Category	Function	Definition
Background	`Background`	Pre-existing facts and known information. It is not an old method.
Approach	`Method`	Pre-existing or new methods. It is a general description of the procedures conducted.
	`Model`	Theoretical model that contains equations, assumptions and the design of objects used.
	`Experiment`	Physical procedures that contain instruments, measurements, any special conditions and detailed steps of the processes.
Outcome	`Observation`	Simple, clear description of the experiment output. This includes the reference to the data presented in tables or figures.
	`Result`	Factual description of the experiment output. It is a further level of output description that could interpret, summarise, or compare a group of `Observation`.
	`Conclusion`	A general summary of what is found by the research and whether it supports or violates the research hypothesis.
Objective	`Hypothesis`	The research prediction that needs to be tested.
	`Goal`	The aim of the work conducted.
	`Motivation`	The importance of the work conducted.
	`Object`	The entity being studied, investigated, reported or discussed in the work conducted. It could be a physical or an abstract entity.

Rhetorical structure theory (RST). The rhetorical structure theory [108] is a hierarchical discourse relation analysis and covers the whole text in order to identify the relations. It is mainly divides the whole text into nuclear and satellite (NS) units and then

Table 5.3. RST relations adopted from [172].

contingency	evidence	concession	condition
restatement	evaluation	example	temporal
background	comment	reason	circumstance
enablement	consequence	explanation	means
summary	result	comparison	attribution
interpretation	antithesis	manner	purpose
definition	concession	condition	elaboration

assigns a relation between them. A *Nucleus* is the unit of text that holds a main idea and can be understood by itself, while a *satellite* is additional information and cannot be interpreted without a linked nucleus. Most of the existing tools [169, 170, 171] have been tested on the text of the Wall Street corpus. The state of the art tool was developed by [171] with an F score of 55.73%. The annotation tool Discourse Parsing from Linear Projection (DPLP) was developed by [172], achieved F-score 61.75% and extracted 28 (NS) high-level relations that are defined in [173]. Table 5.3 lists the RST relations.

We would like to explore how to use the discourse analysis either as a pre-processing step to filter sentences or parts of sentences that mainly talk about the methods reported in the Methods section, or as a post-processing step to filter out the spurious data events produced by the system. We presume that the discourse analysis could group the sentences into data events and non-data events sentences. For example, valid data events sentences should be only about methods and experiments, not about background or goals. If all the true positive data events have no `Background` function, and this function is only found in true negatives, then we could exclude the sentences with this function. We also hypothesise that `Elaboration` spans could be removed and the only the main part of the sentence is processed.

In case the system extracts invalid data events, we presume that there are some discourse patterns could be extracted and then used , as a post-processing step, to minimise this kind of invalid events.

5.3 Results

5.3.1 Data events' components extraction

Using ODNoR, we produced around 1,800 components with 1,200 relations from the corpus of 25 documents.

5.3.2 Workflow construction

Using ODNoRFlow, we obtained 694 events, with an average of 28 events in each document. The data events for each document are linked to form the workflow of the that document. As a result we constructed 25 workflows, one for each of the papers.

5.3.3 Workflow representation

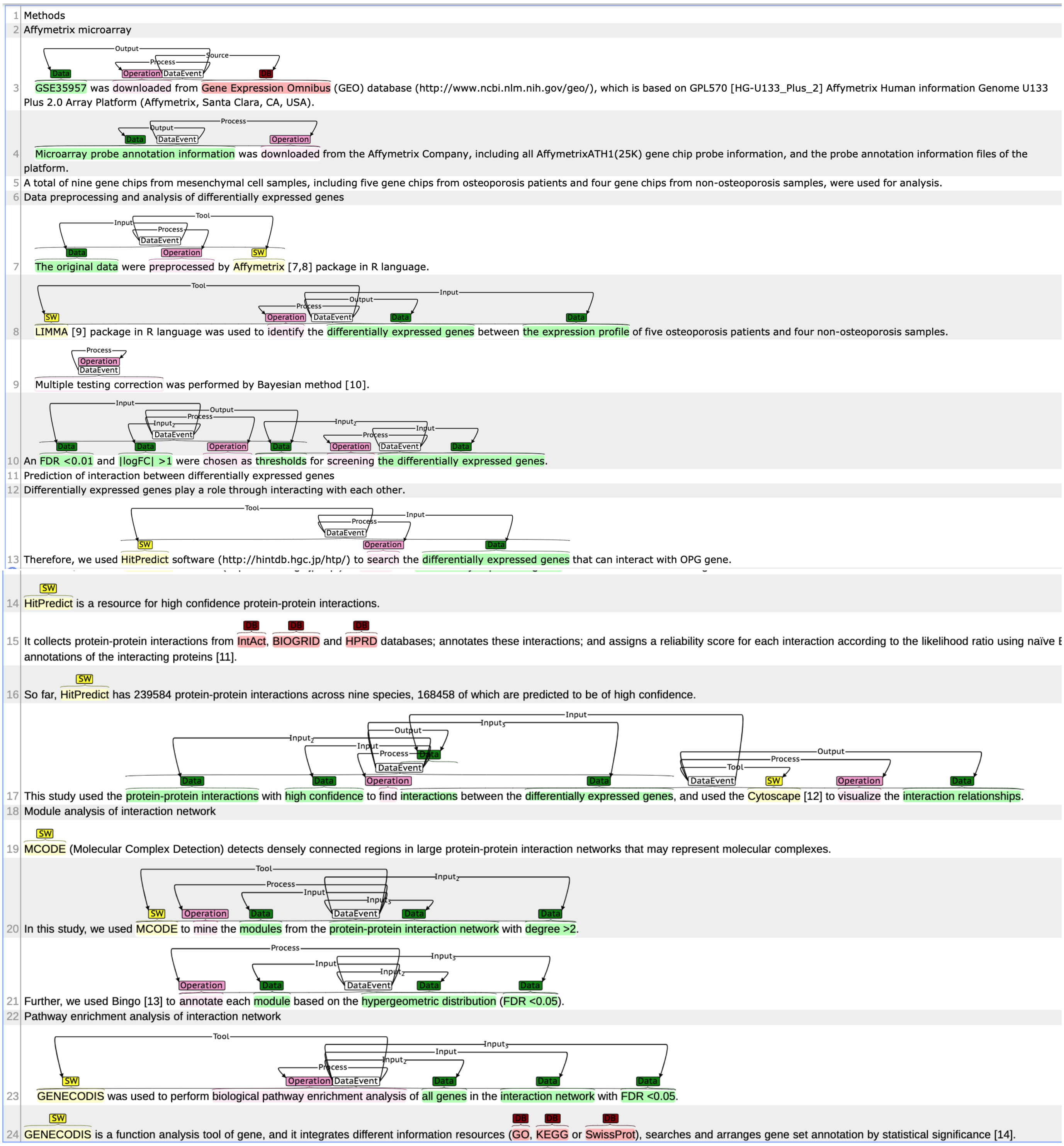
The data events are represented to include the text that comprises the components of the main events. For example, the data events extracted from the article with PMCID PMC3735399 are as follows:

1. *GSE35957 was downloaded from Gene Expression Omnibus*
2. *Microarray probe annotation information was downloaded*
3. *The original data were preprocessed by Affymetrix*
4. *LIMMA [9] package in R language was used to identify the differentially expressed genes between the expression profile*
5. *Multiple testing correction*
6. *FDR < 0.01 and $|$ logFC $| > 1$ were chosen as thresholds*
7. *thresholds for screening the differentially expressed genes*
8. *HitPredict software (http://hintdb.hgc.jp/htp/) to search the differentially expressed genes*
9. *protein-protein interactions with high confidence to find interactions between the differentially expressed genes*
10. *Cytoscape [12] to visualize the interaction relationships*
11. *MCODE to mine the modules from the protein-protein interaction network with degree > 2*
12. *annotate each module based on the hypergeometric distribution (FDR < 0.05*
13. *GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR < 0.05*

These data events are represented as components in different formats. For example, Figures (5.3 and 5.4) and Table 5.4 show the BRAT, network graph and CSV representations respectively of the above workflow (article with PMCID PMC3735399).

Table 5.4. An example of a workflow in the CSV format. The columns represent the sentence ID, event ID, operation, input data, output data, software or database. If there is more than one set of data, software or database involved in the event, they are separated by commas.

sent ID	event ID	operation	input data	output data	SWs	DBs
1	1	downloaded		GSE35957		Gene Expression Omnibus
2	2	downloaded		Microarray probe annotation information		
4	3	preprocessed	The original data		Affymetrix	
5	4	identify	the expression profile	differentially expressed genes	LIMMA	
6	5	Multiple testing correction				
7	6	chosen as	FDR $<$ 0.01, |logFC| $>$ 1	thresholds		
7	7	screening	the differentially expressed genes, thresholds			
9	8	search	differentially expressed genes		HitPredict	
13	9	find	high confidence, protein-protein interactions, differentially expressed genes	interactions		
13	10	visualize	interactions	interaction relationships	Cytoscape	
15	11	mine	modules, degree $>$ 2, protein-protein interaction network		MCODE	
16	12	annotate	module, hypergeometric distribution, FDR $<$ 0.05			
17	13	biological pathway enrichment analysis	interaction network, all genes, FDR $<$ 0.05		GENECODIS	



1 Methods

2 Affymetrix microarray

3 GSE35957 was downloaded from Gene Expression Omnibus (GEO) database (http://www.ncbi.nlm.nih.gov/geo/), which is based on GPL570 [HG-U133_Plus_2] Affymetrix Human information Genome U133 Plus 2.0 Array Platform (Affymetrix, Santa Clara, CA, USA).

4 Microarray probe annotation information was downloaded from the Affymetrix Company, including all AffymetrixATH1(25K) gene chip probe information, and the probe annotation information files of the platform.

5 A total of nine gene chips from mesenchymal cell samples, including five gene chips from osteoporosis patients and four gene chips from non-osteoporosis samples, were used for analysis.

6 Data preprocessing and analysis of differentially expressed genes

7 The original data were preprocessed by Affymetrix [7,8] package in R language.

8 LIMMA [9] package in R language was used to identify the differentially expressed genes between the expression profile of five osteoporosis patients and four non-osteoporosis samples.

9 Multiple testing correction was performed by Bayesian method [10].

10 An FDR <0.01 and |logFC| >1 were chosen as thresholds for screening the differentially expressed genes.

11 Prediction of interaction between differentially expressed genes

12 Differentially expressed genes play a role through interacting with each other.

13 Therefore, we used HitPredict software (http://hintdb.hgc.jp/htp/) to search the differentially expressed genes that can interact with OPG gene.

14 HitPredict is a resource for high confidence protein-protein interactions.

15 It collects protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction according to the likelihood ratio using naïve E annotations of the interacting proteins [11].

16 So far, HitPredict has 239584 protein-protein interactions across nine species, 168458 of which are predicted to be of high confidence.

17 This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relationships.

18 Module analysis of interaction network

19 MCODE (Molecular Complex Detection) detects densely connected regions in large protein-protein interaction networks that may represent molecular complexes.

20 In this study, we used MCODE to mine the modules from the protein-protein interaction network with degree >2.

21 Further, we used Bingo [13] to annotate each module based on the hypergeometric distribution (FDR <0.05).

22 Pathway enrichment analysis of interaction network

23 GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR <0.05.

24 GENECODIS is a function analysis tool of gene, and it integrates different information resources (GO, KEGG or SwissProt), searches and arranges gene set annotation by statistical significance [14].

Figure 5.3. An example of data event annotations in the BRAT format. The series of data events comprise the workflow of the article's method. The operations text is highlighted in green. Data are highlighted in pink, with the databases in red and software in yellow. A data event is annotated to contain all the text that combines the event's components. These components are connected by arrows representing the relations that we defined in the BRAT's Events. This includes one process with a zero or more than one input data, output data, software or databases. A sentence could include more than one data event (e.g. the sentence in line 17). If the databases and software are annotated outside an event context, they will not be included in any events (e.g. the sentences in lines 14, 15, 16 and 20.)

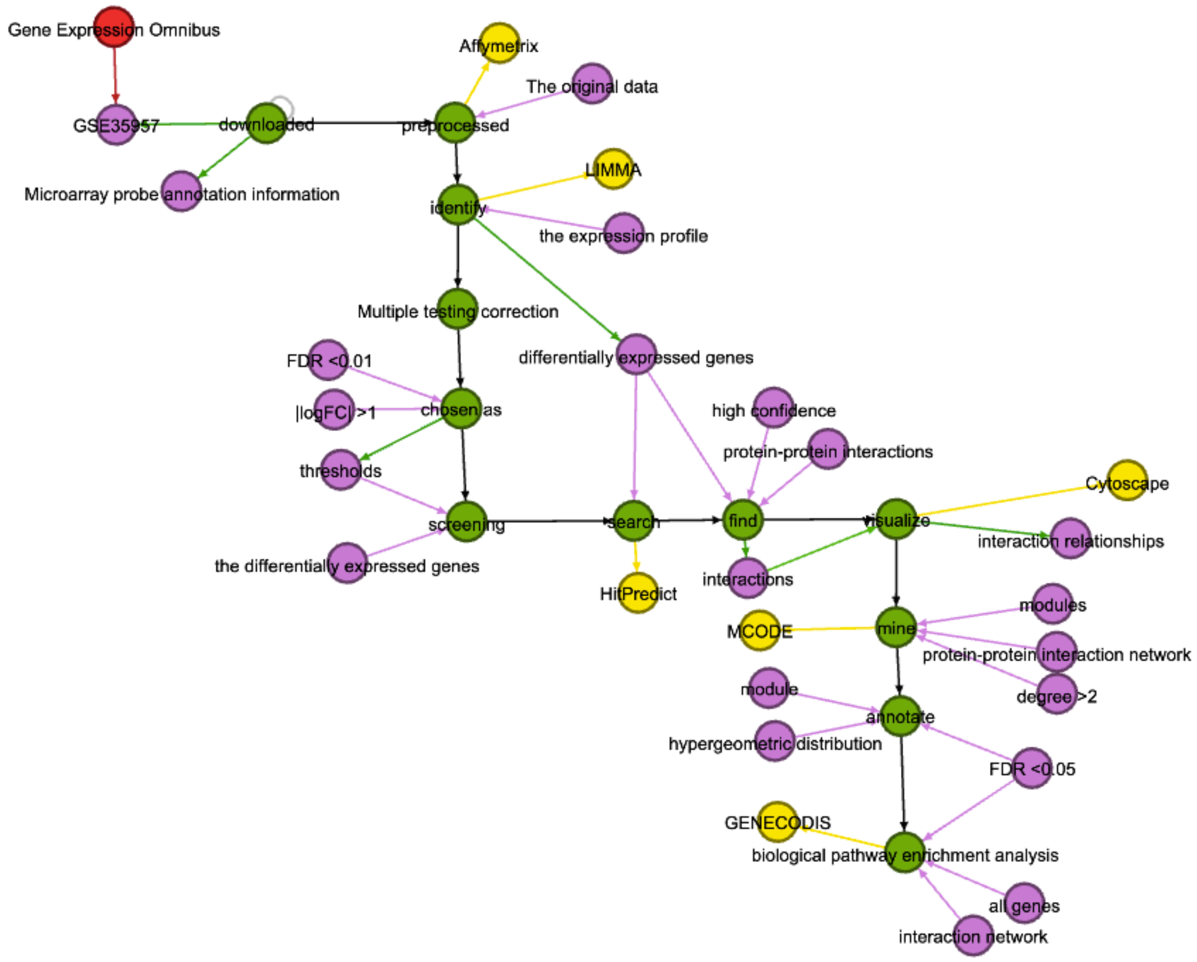


Figure 5.4. An example of a workflow in a network graph produced using Gephi software. The nodes are operations, data, software and databases as annotated from the text. We use the same colour schema as the used for the BRAT presentation above. The flow of the data events is shown with direct black arrows. We have another four types of links, green for output data from an operation, purple for input data to an operation, yellow between the software and operation and finally red between database and data. For example, there is a recursive arrow to the same operation *download* to express the first and second events. The condition *FDR* < *0.05* are used as input data for *annotate* and *biological pathway enrichment analysis* operations.

5.3.4 Workflow evaluation

For the workflow completeness, we evaluated each data event based on the evaluation categories. The result of the evaluation is shown in Table 5.5. It represents the distribution of the 25 documents' data events based on the evaluation categories. We obtained an F-score of 93.56%, 77.30% and 61.04% for the lenient, intermediate and strict evaluations, respectively. Table 5.6 shows the precision, recall and F-score for the retrieved data events. Appendix C includes the workflow evaluation for the article

PMC3735399. The 25 documents evaluations with analysis behind the false positives and false negatives data events are in additional file 1.

Table 5.5. The distribution of data events for each evaluation category for all 25 documents.

Category	**No. of data events**
Good	398
Partial	212
Missing	22
Spurious	62

Table 5.6. Data event F-scores for different type of evaluation.

Category	**Lenient**	**Intermediate**	**Strict**
Precision	90.77	75	59.23
Recall	96.52	79.75	62.97
F1	93.56	77.30	61.04

For inter-annotator specific agreement, the two raters agreed by 89.3% on the good data events and 77.4% on the partial events, with 80% on the missing and spurious cases. Table 5.7 shows the confusion matrix between two annotators, while Table 5.8 shows the specific agreement for each category.

Table 5.7. Confusion matrix of four categories and two annotators for five documents.

		Annotator 2				
		Good	Partial	Missing	Spurious	**Total**
Annotator 1	Good	88	9	0	1	98
	Partial	11	36	0	1	48
	Missing	0	0	2	0	2
	Spurious	0	0	2	0	2
	Total	99	45	4	2	150

Table 5.8. General observed agreement and category-specific agreement grouped by categorises

Type of agreement	**Estimate**
General observed agreement	0.84
Category specific Agreement	
Good	0.893
Partial	0.774
Missing	0.667
Spurious	0.000
Category specific agreement (three groups)	
Good	0.893
Partial	0.774
Poor (Missing+Spurious)	0.800
Category specific agreement (two groups)	
Good (Good+ Partial)	0.993
Poor (Missing+Spurious)	0.800

For sequentiality of events in a workflow, 98.86% of the data events that are manually annotated are correctly ordered, which means the flow of the data events as they reported in the text is the flow of how the are performed. For the events extracted by the system, 89.68% of the detected events are in the correct order. The false positives data events are incorrectly ordered and this minimises the correctly ordered events percentage. Here we provide two examples of the incorrectly reported order.

Example 12 (PMC4438953)
"Microarray data were obtained from three datasets, which consisted of 18, 57 and 38 appropriate samples, respectively. The miRNA microarray series contained data from 15 tumor samples and three healthy control samples, the mRNA microarray test series contained data from 26 tumor samples and 12 healthy control samples, and the mRNA microarray confirmation series contained data from 37 tumor samples and 20 healthy control samples. The three series were accessed at the National Centers for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO) database (http://www.ncbi.nlm.nih.gov/geo/), and the accession numbers were GSE28100, GSE9844 and GSE13601, respectively."

The system annotates the data events as follows:

Data event 1: *Microarray data were obtained from three datasets*

Data event 2: *The three series were accessed at the National Centers for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO) database , and the accession numbers were GSE28100, GSE9844 and GSE13601*

However, the logical order is that the event in the second sentence happens before the events in the first sentence: we would obtain the accession numbers first, then we would retrieve the data.

Example 13 (PMC2919724)
"To test enrichments of TFs targeting protein complexes, we constructed protein–protein interaction modules for each TF target."

The system annotates the events as follows:

Data event 1: *test enrichments of TFs targeting protein complexes*

Data event 2: *constructed protein to protein interaction modules for each TF target*

However, the logical order is first to construct, then to test. Since the system followed the order of the text, it rendered them in an incorrect order.

5.3.5 Data events and discourse analysis

We first present the results of considering the discourse analysis as a pre-processing step, and then as a post-processing step. We can say we cannot group the valid or invalid data events into two different groups of discourse and thus we cannot use the functional or RST discourse neither as pre-process nor post-process to improve the quality of the extracted data events.

Discourse as a pre-processing step

Functional discourse. In Chapter 3 we studied the functional discourse patterns in Methods section. It contains the distribution of functional discourse over a larger corpus. We have a general pattern that most of the Methods section functions follow: `Methods,` `Experiment` and `Models`, but it also could have other functions like `Result, Observation` and `Background`. We analysed the sentences that have data events and the sentences that have none, and found that they could be allocated to any kind of functions. Example of sentences that have valid data event and sentences that have none with the same discourse function are provided here.

First example, here are two sentences annotated as `Method` function. The first sentence has data event of expression correlation analysis done by Arabidopsis co-expression tool (ACT). The second sentence describes the method of the tool, but not part of the main work.

Example 14 (PMC3123201)
"An expression correlation analysis was performed for PSY using the freely available Arabidopsis co-expression tool (ACT)." ... ... "Importantly, the ACT tool uses NASC/GARNet data sets that were labeled, hybridized and analyzed using a standardized procedure thus providing a homogeneous and readily comparable data set."

Second example, here are two sentences with `Background` function. The first sentence has a valid data event, while the second does not.

Example 15 (PMC4550637)
"In this study, the gene expression microarray data set GSE4612 was downloaded from the Gene Expression Omnibus (GEO, http://www.ncbi.nlm.nih.gov/geo/). GSE4612 [8] is a gene expression profile data including six Mdr2 knockout (Mdr2-KO) mutant mice samples(3-month-old and 12-month-old) and six control mice samples(3-month-old and 12-month-old)."

Rhetorical structure theory. `Elaboration` comprised (80%) over all satellite relations. Eliminating part of sentences with this relation will eliminate a large part of the events. Table 5.9 shows the percent of each function from the corpus used in Chapter 3.

Table 5.9. RST annotation result.

Relation	**count**	**percent (100)**
`contingency`	1	0
`restatement`	2	0
`background`	3	0
`enablement`	4	0
`summary`	5	0
`interpretation`	6	0
`definition`	12	0
`evidence`	20	0
`evaluation`	31	0
`comment`	37	0
`consequence`	38	0
`result`	117	0.01
`antithesis`	581	0.05
`concession`	635	0.05
`example`	747	0.06
`reason`	1,768	0.15
`explanation`	1,886	0.16
`comparison`	2,131	0.18
`manner`	3,028	0.25
`condition`	12,435	1.03
`temporal`	12,584	1.04
`circumstance`	30,499	2.52
`means`	36,872	3.04
`attribution`	43,730	3.61
`purpose`	89,282	7.37
`elaboration`	974,946	80.48
Total	1,211,441	100

Discourse as a post-processing step

Following are examples of spurious data events with their discourse analysis by functional discourse and RST. Valid data events are also found for the same discourse function or relation. Therefore, we cannot use functional discourse as an eliminating criterion for spurious events.

Example 16 (PMC4709009)

"The identified genes, periodicity and cluster assignment for both NPSG and CCS studies are presented in S1 Table. All reads found with the probes for Prochlorococcus, Synechococcus, SAR11 and Ostreococcus are listed in S2 Table."

In the above example, the two sentences are `Elaboration` and `Result` sentences. However, we cannot exclude the Result sentences, since some contain true positives: the next example contains true positive data event that are annotated as `Result`.

Example 17 (PMC2998528)
" This generates information-less probesets while keeping the properties of the original probe intensity distribution."

Data event : <**operation**:*generates*, **output data**: *information-less probesets*>

An example of spurious data event with an `Elaboration` (RST) relation:

Example 18 (PMC4709009)
"Metatranscriptome sequence data, analyzed in this study, was obtained from NCBI Short Read Archive database"

The system annotates two data events:

Data event 1: <**operation**:*obtained*, **output data**: *Metatranscriptome sequence data*, **database**: *NCBI Short Read Archive database*>

Data event 2: <**input data** : *Metatranscriptome sequence data*, **operation** :*analyzed*, **database**: *NCBI Short Read Archive database*>

The second data event is a spurious data event, since the analysis was an `Elaboration` to the main part of the sentence.

5.4 Discussion

The accuracy of the extracted workflows is affected by the accuracy of the modules involved in extracting and connecting the components that comprise an event. It is possible that the main event's components are detected but the relations between them are connected incorrectly, for example, the output data depicted as input data. The relations to databases or software could be missed or linked to incorrect data and operations.

The workflow completeness is mainly affected by the accuracy of the ***data*** annotations (rather than ***operations***). 48% (30 out of 62) of the spurious events (false positives) are caused by either annotating incorrect ***data***, or the correctly annotated ***data*** failing to be connected to the ***operation*** of the related event. In either case, a false positive implicit data event is generated to represent ***data*** that are mentioned without an operation.

The reason behind missing the link between ***data*** and ***operation*** is caused mainly by the sentence detection system that failed to find the correct sentence boundaries. A further 25% (16 out of 62) of the spurious data events (false positives) are caused by annotating other people's work mentioned in the method description, descriptions of tools used or methods, or annotating a result context rather than a process one.

A total of 81% of the partial data events occurs mainly if one sentence contained more than one data event, and the span of one data event incorrectly includes more data, software or database that belongs to another data event. E.g.

Example 19 (PMC3123201)
"POCO tool was used to identify enriched elements and the POBO tool was used to verify the presence of identified elements in the PSY promoter."

The correct data events should be as follows:

Data event 1: *POCO tool was used to identify enriched elements*

*<**operation:**identify, **output data:**enriched elements, **software:**POCO>*

Data event 2: *POBO tool was used to verify the presence of identified elements in the PSY promoter*

*<**operation:**verify, **input data:**the PSY promoter, **output data:**the presence of identified elements, **software:**POBO >*

But the system annotates them as follows:

Data event 1: *POCO tool was used to identify enriched elements and the POBO tool was used to verify the presence of identified elements in the PSY promoter*

*<**operation:**identify, **input data:**identified elements,the PSY promoter, **output data:**enriched elements, **software:**POCO,POBO>*

Data event 2: *POBO tool was used to verify the presence of identified elements in the PSY promoter*

*<**operation:**verify, **input data:**identified elements,the PSY promoter,enriched elements, **software:**POCO,POBO>*

Data event 1 includes the same text as **Data event 2**. While the operations are correctly identified, the ***data*** and software are not.

In our inner-annotator agreement evaluation, the good and partial data events are more frequent than the missing and spurious data events. This creates imbalanced classes that will always give low value of the chance-adjusted agreement measure, like Cohen's kappa or Scott's pi, although the agreement is high [174, 165]. Cicchetti and Feinstein [175] suggested using a specific agreement category as a solution for this problem, and we applied that in our inner-annotator agreement. As a result, instead of having one value, the evaluation of the agreement will have different value for each category and we used that in our evaluation.

The functional discourse analysis did not show a significant improvement because the annotated data events can be reported in sentences belonging to any type of functional discourse. For example, the `Result` function can be matched with both correct and incorrect events. The Rhetorical Structure Theory did not show a significant improvement either. For example, there are true positive and false positive data events that have `Elaboration` relations. While the discourse analysis (especially the RST) is a useful methodology from a theoretical aspect. The current state of the art in discourse annotation tools cannot provide an acceptable accuracy.

We found that for some cases the argumentative zoning (AZ) [99] could help in differentiating between main work and other work by the two relations `Own` and `Other`. Applying AZ could help in filtering out the sentences that are solely talk about the other work that are not part of the work conducted. Here is an example.

Example 20
"HitPredict is a resource for high confidence protein-protein interactions. It collects protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction according to the likelihood ratio using naïve Bayesian networks combining sequence, structure and function annotations of the interacting proteins [11]."

The second sentence in the above example is a description of the methods applied by the resource mentioned in the first sentence. The second sentence contains two spurious data events with ***operations*** <collects/assign>. This sentence is, however, a description of other work and could be discovered by the AZ discourse and then filtered out.

5.4.1 Limitations

The method presented here has several limitations, which we discuss below.

1. Data events are linked only to previous and subsequent data events based on the order of their appearance in the text. There is no differentiation between data events that can go in parallel and the ones that must go in sequential order. Nested data events are also not shown in the abstract workflow, and are not marked with sub-order to show the main data event they belong to. Here is an example.

   Example 21 (PMC2374988)
   "Datasets were further normalized, if necessary, by transforming them onto a common log2 scale and shifting the median of each array to zero"

   The sub-processes are done for one purpose, *normalisation* but this will be extracted as three sequential data events.

2. The main components of a data event are combined only if they are mentioned in same the sentence. Any involved component mentioned later will not be added to the same data event. For example, if the mention of a software used was in the following sentence, the main data event will miss it.

   Example 22 (PMC4201588)
   "We perform feature selection based on the training data within each iteration of validation. We use existing MATLAB implementations of these algorithms [8], [42]."

   In this case, the first sentence contains the following data event:

   <***operation***: *feature selection,* ***input data:*** *training data*>, and the second sentence contains the software used, *MATLAB*. The main data event is failing to include the software mentioned in another sentence. Capturing related components outside the sentence boundary will improve the context of implicit data events and relate the ***data*** with the process applied to it.

   The same scenario applied to ***data***. We did not keep track of them out side the sentence. For that, we are not sure if the ***data*** mentioned in multiple sentences and share the same syntax, is the same data. This clearly shown in the directed network graph where the same syntax data are represented in the same node. For example, in Figure 5.4, the output data *deferentially gene expression* are used as input for another two operations *search* and *find*. This usage was separated by three processes. Although, the data have the same syntax in the three sentences they are mentioned in, we are not sure if they are the same actual data.

3. Data events are constructed and analysed at the sentence level. Constructing data events at upper levels, for example, at a paragraph or subsections level, might

introduce interesting patterns for independent tasks and facilitate the understating of the methods conducted.

4. We did not handle the coreferring expressions neither at data event nor at workflow level. Addressing the coreference for data will help in tracking all the processes happened to a specific data, in case it is referred by another expression, all over the workflow. It will also provide a better understanding of produced workflow.

5. Some of the false positives operations refer to work conducted by other people. Using argumentative zoning [129] could enhance the result but we did not test the feasibility of using this type of discourse.

6. We evaluated the data events as an indicator of the accuracy of a workflow. More metrics and evaluation methods should be considered to test how far the workflow reflects what is reported in the paper.

7. In some cases, ***data*** will lose the meaning if read alone without backing to the whole data event.

   Example 23 (PMC3735399)
   "This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes.."

   input data: *protein-protein interactions*

   input data: *high confidence*

   input data: *differentially expressed genes*

   output data: *interactions*

   The condition of *high confidence* loses its purpose if it is not read in the context.

5.5 Conclusions

In this paper, we introduced and evaluated ODNoRFlow, an automated method processing which accepts plain text from the literature and produces a workflow in a representation of linked data events that express the steps of the work reported in the text. Abstract workflows were represented in different readable formats that could enhance the understandability of the work conducted and enable indexing the literature for further analysis.

Workflows were extracted with an acceptable F-score between 61% and 93%. 88% of extracted workflows were reasonable (good and partial data events), while 12% (84 out of 694) were not acceptable (missing and spurious).

We also studied the feasibility of using the discourse analysis to improve the accuracy of abstract workflows. Although discourse analysis can be used as a supportive mechanism to understand the result of the extracted data events, the current discourse analysis results obtained cannot be used to cluster valid and invalid data events, and hence are not useful at present.

5.6 Additional Files

5.6.1 Additional file 1 — Data events evaluation

Additional file shows the data events evaluation based on the categories (good, partial, missing or spurious) and analysis of missing and spurious data events for 25 documents.

Chapter 6

Data Events in Microarray Analysis: a Survey of the Primary Literature

Abstract

Background Scientific methods are important. The information as to how they are used in research, however, is mostly trapped in the literature. Having methodologies described in a computational form would enable a series of questions to be asked about the methodological landscape, and would allow researchers to understand how methodologies in a particular area develop over time. This is of specific interest in computational domains, where methodologies applied to data change rapidly.

Objective This study aimed to examine computational methodologies reported in the microarray literature on a large scale. We focused on data events (i.e. operations applied to what data) and their components, and analysed frequent patterns as reported in the method sections in a corpus of primary literature.

Methods We have previously developed a text mining system that extracts data events from the literature. A data event contains an abstract representation of the work conducted in terms of ***input data***, ***output data***, ***operations***, and if there are any ***software*** or ***databases*** involved. The system is applied to the Methods section in a corpus of 16,604 papers from 1998 to 2021 from the microarray analysis literature. We analysed the extracted data events to determine the frequency of each component and the frequent relationships between the components.

Results We extracted 751,106 data event mentions and analysed 1,042,007 ***data***, 616,144 ***operations***, 63,508 ***software*** and 54,121 ***databases*** as mentioned in the literature. We found some significant data event patterns: for example, *data* are most frequently *retrieved* from the *GEO* database and *processed* by *R*. `Statistical calculation` has a significant relation with the `SPSS` software.

Conclusion Large-scale text mining and data event extraction can be used to extract and explore the patterns and trends in data usage and processing in the microarray analysis literature.

6.1 Introduction

Understanding scientific methods is an essential step to evaluate and reuse them. Researchers are faced by a large number of publications with methods rarely represented in a computationally-readable form (such as executable papers or executable workflows) but are rather described in prose in scientific publications. A computational representation of methods could reduce the time and effort required to read the detailed information in publications, and would enhance the ability to keep track of existing and recently published methods. Having methodologies described in a computational form would enable a series of questions to be asked about the methodological landscape, and would allow researchers to explore and understand how methodologies in a particular area develop over time.

This is of specific interest in computational domains, where methodologies applied to data change rapidly. We carried out this investigation on the computational analysis of microarray data. Microarray analysis has been used extensively in the last two decades. Within this period of time, methods for analysing the huge amount of the generated biological data have been continually developed [13, 14]. Capturing microarray data and developing the analytical instruments has produced a significant amount of software, a number of packages and databases [176].

We have previously developed a text mining system, ODNoR (Chapter 4) that extracts and normalises main data event entities reported in a research paper, and ODNoR-Flow (Chapter 5) that organises entity mentions and relations in linked data events to form an abstract workflow that demonstrates the work reported in a paper.

A data event is an operation on data. These data could be retrieved from a database so, that operations could be performed by a software. As a result, a representation of a data event contains an operation, data, software and databases involved in the data event. An abstract representation of a data event is:

<***input data:*** *data*, ***operation:*** *operation*, ***output data:*** *data*, ***software:*** *software*,

database: *database*>

Example 24 (PMC3735399)
"This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relationships."

The above sentence has two data events. The second data event will be as following:

Data Event:<***input data:*** *interactions*, ***operation:*** *visualize*, ***output data:*** *the interaction relationships*, ***software:*** *Cytoscape*, ***database:*** *NA*>

The data event with a layer of normalisation:

Data Event:<***input data:*** `Atom interaction data` $http://edamontology.org/data\_0906$

operation: `Visualisation` $http://edamontology.org/operation\_0337$,

output data: `Protein interaction record` $http://edamontology.org/data\_0906$,

software: *biotools:cytoscape*>

Here, we used ODNoRFlow to survey data events usage in a corpus of microarray analysis literature. This will help to give insights into the common practices of the microarray field and how they change. We used the system to extract data events from the literature, and then used these results to perform the following analyses:

1. Analyse the mentions of data event components.
2. Contrast how the relative usage of different components has changed over the last 20 years.
3. Analyse the frequent and significant relations between data event components across the whole corpus.
4. Analyse the trending components and if there are any increases or declines in the usage of the most frequent ***operations***, ***data***, ***software*** and ***databases***.

6.2 Methods

We used ODNoRFlow to extract the data events from the corpus used previously in Chapter 4. We then performed the analysis to find data events usage and relation patterns as specified below.

6.2.1 Dataset

PubMed central [135] was searched in March 2021 for articles that were published between 1^{st} January 1998 and 31^{st} March 2021 using the MeSH term `microarray analysis`. In addition, to obtain full text articles, we restricted the PubMed search to include only those in the PubMed Central Open Access subset. The query used was:

```
``microarray analysis''[MeSH Terms] AND (``1998/01/01''[PubDate] :
``2021/03/31''[PubDate]) AND ``open access''[filter]
```

6.2.2 Data event extraction

Data events appearing in the Methods section of each paper from the microarray analysis corpus were extracted using ODNoR and ODNoRFlow (Chapter 4 and 5). The extracted data events were used for all the analyses below.

We expected to have data events with components representing the work reported in a paper. An abstract example of the data event:

<input data: *data*, ***operation:*** *operation*, ***input data:*** *data*, ***software:*** *software*,

database: *database*>

The extracted ***data, operations***, and ***software*** were normalised to the knowledge resources. Mentions of ***data*** was mapped to EDAM's Data, Topic ontologies and Europe PMC annotation, ***operation*** mapped to EDAM's Operation ontology [34], and the ***software*** to bio.tools [153].

We had already implemented the rules of the normalisation in Chapter 4. They are mainly to match a mention to a syntax similar class that has a cosine distance within a given threshold. In case a mention failed to be mapped by one of the rules, it was mapped to the top class of the ontology searched.

6.2.3 Data event analyses

We first explored the data event components (i.e. ***data***, ***operation***, ***software*** and ***database***) as a stand-alone component: how are these were distributed over the Methods section, what the relations between them are, and the usage trends of these components. Then we look more closely into the relations between more than one component. The following shows how we performed these analyses.

Distribution

We measured the minimum, maximum, median and mean for the component mentions. For ***data*** and ***operations*** components we defined unique mentions at different levels. There is a unique *text*, where the mentions are written with the same syntax in the text; *lemma*, where different mentions share the same lemma; *class name*, where different mentions share the same name of an EDAM class; *class ID*, where different mentions share the same ID of an EDAM class regardless of the class name. This class ID refer to all mentions mapped to one of the class ID's synonyms. We will use Data EDAM classes for data mentions and Operation EDAM classes for operations mentions. We will use the preferred name when we refer to the class ID for an easy read. Example of class ID is `Statistical calculation` refers to a group of synonyms classes that includes `Significance testing, Hypothesis testing, Statistical analysis, Statistical testing, etc.`

We were interested to compare between the mentions usage at the mention level and document level. The mention level is the total number of mentions of a component in the corpus. We calculated the mean of a component mention per document as:

$$\text{Mean (C, Doc, men-level)} = \frac{\text{Number of all C mentions}}{\text{Number of all Doc}}$$

where C is the component, Doc is the documents that have a method section.

The document level metric is the number of documents that contain at least one mention of a component. We calculated the mean of a component mention per document as:

$$\text{Mean (C, Doc, doc-level)} = \frac{\text{Number of Doc that have at least one C mention}}{\text{Number of all Doc}}$$

where C is the component, Doc is the documents that have a method section.

Owing to the limited space, we are going to report the results for lemma mentions and class ID mentions.

Relative usage

We measured each data event component's relative usage for each year at the mention and document levels. It was the same mean formulas explained above except that it specifically considered the mentions and documents at year level. Relative usage of a component for a specific year (mention level):

$$\text{Relative usage (C, Doc, Y, men-level)} = \frac{\text{Number of all C mentions in Y}}{\text{Number of all Doc in Y}}$$

$$\text{Relative usage (C, Doc, Y, doc-level)} = \frac{\text{Number of Doc in Y with at least one C mention}}{\text{Number of all Doc in Y}}$$

where C is the component ,Y is the year, Doc is the documents that have a method section.

Due to the large amount of data given by the relative usage, we plotted only the relative usage of the top six entities for each component. We plotted the results of lemma mentions and class ID mentions.

Trends

We examined the usage trend for key concepts in each component. We checked if the forecasted usage matched the actual one. We used local regression methodology (loess), originally proposed by Cleveland [177] and further developed by Cleveland and Devlin [178], that uses neighbouring data to forecast future usage. The loess method is based on locally linear smoothing and can handle outlier relationships where nonlinear but smooth relations exist between data. We specifically used 'geom_smooth()', implemented in ggplot2 package [179] in R, with the method = 'loess' and formula 'y $\sim$ x'.

From the produced fitted line predicted by this method, we can tell how an individual component trends up or down and when over the years.

Association rules

We studied the relations between the components using association rules. We analysed the rules in two aspects; the document level and event level. The first aspect is at the document level, where the relations are extracted between all unique mentions in documents. In terms of association rules, the transactions will be documents and items are all components in that document. The second approach is at the event level, where the relations are extracted based on the unique mentions in each event. In terms of association rules; the transactions will be data events and the items are the components that have relations in the data events. We used class ID for data and operations and text for database and software as mentions for components.

Association rules are an unsupervised learning method that finds a relation or association between two items; when derived from a large dataset, an association rule shows how item Y is dependant on the appearance of item X: if an item X is used (or mentioned), then item Y will be used (or mentioned). Rules depend on *support*, *confidence* and *lift*. *Support* is a measure of how frequently X and Y occur together compared to all transactions. *Confidence* is a measure of how often items in Y appear in transactions that contain X only. *Lift* is the rate of the confidence to the expected confidence. It will measure how much our confidence has increased that Y will be used given that X was used. Greater lift values indicate stronger associations. Beside these measures, we used Fisher's exact test with a correction to find the significant rules [180]. This was to improve the quality of the returned rules and show only the rules that had strong relations.

$$\text{Support}(X \longrightarrow Y) = \frac{\text{Number of transactions that contain } (X \rightarrow Y)}{\text{Total number of transactions}}$$

$$\text{Confidence}(X \longrightarrow Y) = \frac{\text{Number of transactions that contain } (X \rightarrow Y)}{\text{Total number of transactions that contain } X}$$

$$\text{Lift}(X \longrightarrow Y) = \frac{\text{Confidence}}{\text{Support } (Y)} = \frac{\text{Support}}{\text{Support } (Y) \, * \, \text{Support } (X)}$$

$$\text{Lift}(X \longrightarrow Y) = \begin{cases} > 1 & \text{positive relationship} \\ 1 & \text{X and Y are independent} \\ < 1 & \text{negative relationship} \end{cases}$$

We used the Apriori algorithm [181] in R to calculate the required metrics and derive the rules. We used arulesVis R package [182] to visualise the significant rules. To visualise a large number of rules, we used group matrix visualisation [183] that clusters the rules to view them in one figure. It shows them as a matrix, where the left hand side of the rules are clustered and shown as columns, and the right hand side rules are shown in the rows. The circle in the intersection between a row and column shows there are rules. The size of a circle shows the support and the density of the colour shows the lift. The rules are organised where the rules with the highest lift are plotted first (left corner).

6.3 Results

6.3.1 Dataset

We have collected a total of 17,736 documents retrieved from PubMed using Entrez. Using the Section Tagger [139], a Methods section was identified in 16,604 documents, which were used in the experiments below. Table 6.1 shows some character-

Table 6.1. Corpus statistics.

Characteristics	Value
Number of articles	16,604
Number of sentences in the method section	1,147,882
Maximum number of sentences in a single document	469
Average number of sentences per document	69
Median number of sentences	63

Table 6.2. Number of extracted events from the corpus.

Characteristics	Value
Number of events in the Methods section	751,106
Average number of events per document	45
Median of events	42

istics of the corpus. The average and median numbers of sentences are 69 and 63, respectively.

6.3.2 Data event extraction

We have extracted 751,106 data events from the corpus, with an average of 45 data events per paper (see Table 6.2).

The normalisation of ***data*** shows a modest mapping result. 47.87%, 38,81% and 5% of unique data mentions (text) find a corresponding match in Data, Topic EDAM ontology and Europe PMC annotation, respectively.

The normalisation of ***operations*** shows a better result since 81.83% of unique mentions (text) match a class in Operation EDAM ontology.

6.3.3 Individual components analyses

Data

Distribution We have extracted a total of over a million data mentions from the corpus (the average of 63 per document and two per sentence) (see Table 6.3 for more detailed statistics).

Table 6.4 further shows frequent mean mention-level and document-level based on the lemmatised mentions and normalised class ID. The major observations are as follows:

- *gene, datum, sample* and *p value* are top lemmas respectively.

- `Data, Expression Data, Gene ID (GeneDB)` and `Experimental measurement` are top class IDs respectively.

- At the mention level, class ID `Data` has a very high mean (33.84) compared to means of corresponding lemmas (*datum* and *dataset*) and other class ID mentions. This is because most of the data mentions mapped to this class ID.

- 81% of the documents in the corpus contains at least one mention that is normalised to class ID `Expression data` (mean at the document level is 0.81). This class has synonyms names such as `Microarray data, Gene expression data, Gene product profile, Protein expression data, Protein profile, RNA-seq data`, etc.

- 40% of the documents in the corpus statistically test their data by `P-value`.

- `Sequence` and `dbProb ID` are more frequent in documents than `Concentration`, while the latter is more frequent at the level of the corpus.

- Class ID `E-value` with a synonym class `Expectation value` matched to, a statistical estimate score, E-value, and also could be matched, by cosine similarity, to lemma *value* or a mention ends with lemma *value*. Both the class ID and the lemma of *value* appeared to be frequent based on the mean of the mentions.

Relative usage Figure 6.1 shows the relative usage of the top unique mentions (lemmas and class ID) at mention and document levels. The major observations are as follows:

- `Data` and `Expression data` are the most popularly used ***data*** over the years of the corpus with a mean always greater than 0.7 (class ID at the document level).

- At the mention level, the relative usage of top class ID mentions, apart from `Data`, shows relatively steady usage over the years.

- For lemmas at mention level, *gene* has a higher relative usage over the years compared to other lemmas apart from *datum*. This relative usage declined over the years. It goes from a mean mention level of 2.65 to 1.13. At the document level, the mean of documents that contains *gene* declined as well, 35% of documents in 2017 compared to 59% in 2006 and 2007.

- The relative usage of lemmas *p*, *p value* and the corresponding class P-value are rising over the time. At the document level, 46% of the documents contain a mention of P-value (years 2019 and 2020). This indicates a tendency to analyse the data using statistical significance tests.

Trends Figure 6.2 shows the fitted lines for the top six data mentions (class ID) learnt by local regression methodology (loess). If we look at the trends over time, we can see the following observations:

- Although the high use of the Expression data was noticeable, it fluctuated over the years.
- Gene ID reached its peak usage in the first five years, and was trending down slowly after then.
- P value usage increased sharply until 2010 then continued increasing slowly.
- The usage of E-value class stayed relatively stable over the years.
- Sample ID showed a consistent growth all over the years in the corpus.

Association relations

We obtained 201 significant rules that show the relations between data mentions (class ID) over the document. Figure 6.3 plots the first ten significant relations ordered by support value. Data is always co-used with Expression data, Sequence, Experimental measurements, GeneID and many other kinds from the data classes.

We tested the significant rules after removing the Data class from the testing set and the number of significant rules reduced to nearly a quarter of the original (59 rules). The rules are shown in grouped matrix format in Figure 6.4. The right hand side of the rules contains the classes Experimental measurements, Expression data and GeneID, the left hand side shows the classes in groups of rules. Some examples :

- An example of a high lift rule is that, Gene ID (GeneDB) and Sequence set are significantly mentioned in the same document with Experimental measurement.
- Database search results, Score and Expression data
- Root-mean-square deviation and Expression data

Table 6.3. Statistics of data mentions in the corpus.

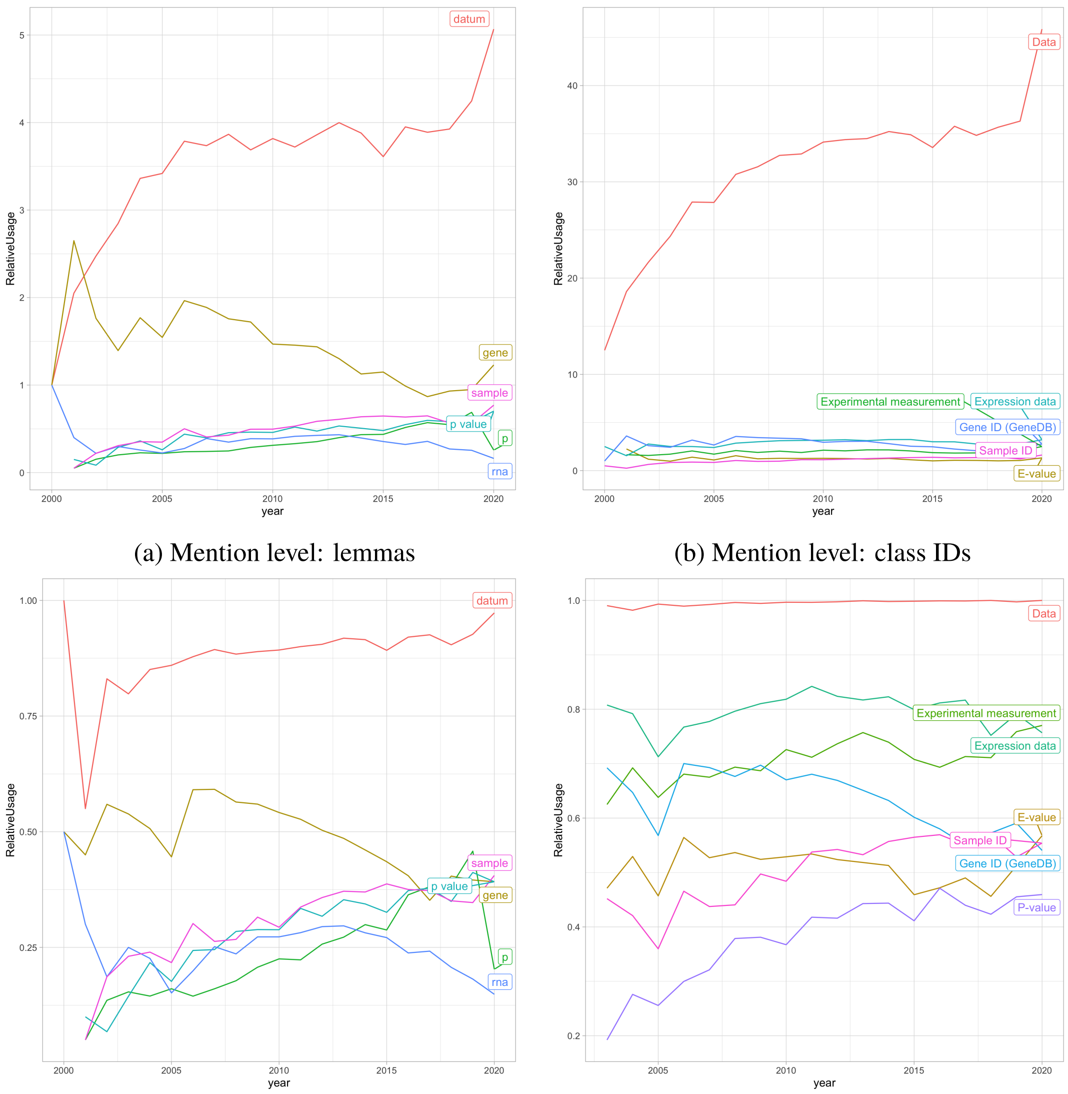
Characteristics	Data
Total mentions	1,042,007
Unique mentions (text)	405,466
Unique mentions (lemma)	274,375
Unique mentions (class name)	1,115
Unique mentions (class ID)	735
Mean of total mentions per document	62.76
Median of total mentions per document	56
Minimum mentions in a single document	1
Maximum total mentions in a single document	425
Number of sentences with at least one mention	488,484
Percent of sentences with at least one mention	42.56%
Mean of total mentions per sentence	2.13

Table 6.4. Frequent mention-level mean and document level mean for data lemmas and class ID. *Sample* data has a document level mean as 0.33 and 0.52 for lemma and class ID

Data lemmas				Data EDAM concepts			
Mention level		Document level		Mention level		Document level	
mention	mean	mention	mean	mention	mean	mention	mean
gene	1.37	datum	0.52	Data	33.84	Data	1
datum	0.92	gene	0.49	Expression data	3.03	Expression data	0.81
sample	0.55	sample	0.33	Gene ID (GeneDB)	2.81	Experimental measurement	0.71
p value	0.49	p value	0.32	Experimental measurement	2	Gene ID (GeneDB)	0.64
p	0.38	p	0.26	Sample ID	1.21	Sample ID	0.52
rna	0.37	rna	0.26	E-value	1.19	E-value	0.51
probe	0.37	gse	0.22	P-value	0.73	P-value	0.4
gse	0.29	value	0.19	Sequence	0.7	Concentration	0.34
value	0.27	result	0.19	dbProbe ID	0.69	Citation	0.32
result	0.24	probe	0.19	Concentration	0.56	Sequence	0.3
array	0.23	array	0.17	Matrix	0.53	Matrix	0.3
sequence	0.23	primer	0.17	Score	0.53	dbProbe ID	0.27
primer	0.22	microarray datum	0.16	Citation	0.5	Image	0.25
dna	0.21	sequence	0.14	Image	0.43	Database search results	0.25

Analysing the non-significant rules with high lift shows that it is also common to have:

- `Pathway` or `network` associated with `P-value`.
- `Q-value` and `P-value` are frequently mentioned together, where `Q-value` has synonym classes such as `FRD` and `Adjusted P-value`.
- `Database search results` and `Sample ID` are also frequently found together.



(a) Mention level: lemmas

(b) Mention level: class IDs

(c) Document level: lemmas

(d) Document level: class IDs

Figure 6.1. Relative usage over the years 2000 to 2020 at the mention level and document level of the top six mentions for data lemmas and class ID. The relative usage of document level ranged from 0 to 1 while it could be more than 1 in mention level. For example: the lemma *gene* mentioned at least once in yearly documents (mean at mention level range between 2.56 and 1.13 (a)). The lemma *gene* mentioned in 59% to 35% of the corpus documents over the 20 years (document level(c)).

Operations

Distribution We have extracted a total of over a 600K operation mentions from the corpus (the average of 37 per document and one per sentence) (see Table 6.5 for more detailed statistics).

Table 6.6 goes further and shows frequent mean mention-level and document-level based on the lemmatised mentions and normalised class ID. The major observations are as follows:

- *calculate, analyse, determine, obtain* and *identify* are top lemmas at the two

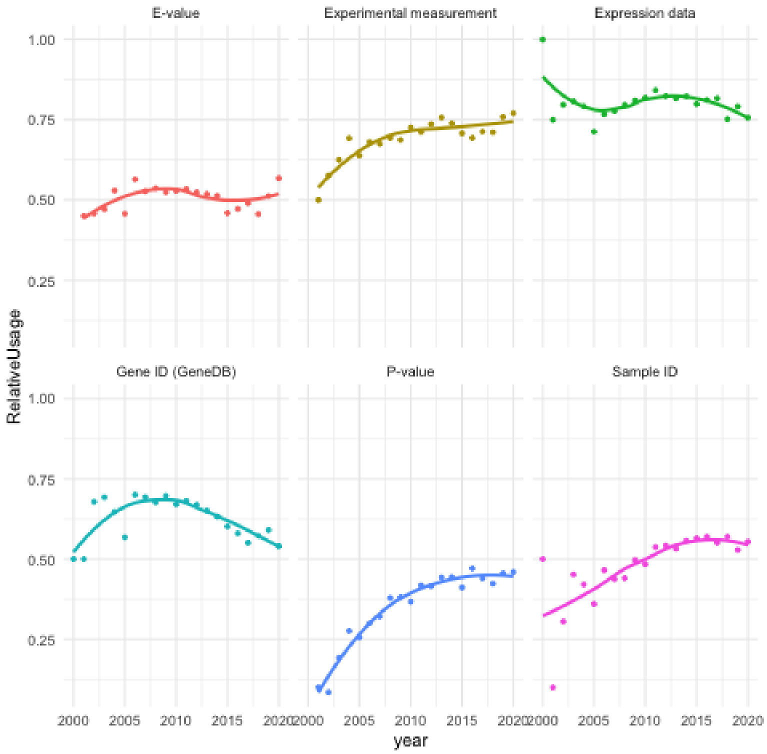


Figure 6.2. Trending of top six data mentions (class ID). The points are the actual relative usage of a mention in a year. The curve shows the forecasted usage based on previous actual usage.

levels of mentions respectively.

- `Calculation` is the frequent class ID. 94% of the documents in the corpus includes calculation processes. One document has a mean of 5.26 for this kind of process.

- `Visualisation` and `Statistical calculation` have the same mean at the document level (0.25). However. the `Visualisation` is more frequent at the mention level which means the results are more often visualised than analysed statistically in the same document.

- `Assigning Role` with lemmas *select, use* and *consider* are frequently used as operations mentions.

- `Analysis, Validation, Standardisation and normalisation` and `Filtering` are frequent operations that show additional processing on the data used in the study.

Relative usage Figure 6.5 shows the relative usage of the top unique mentions (lemmas and class ID) at mention and document levels. The major observations are as follows:

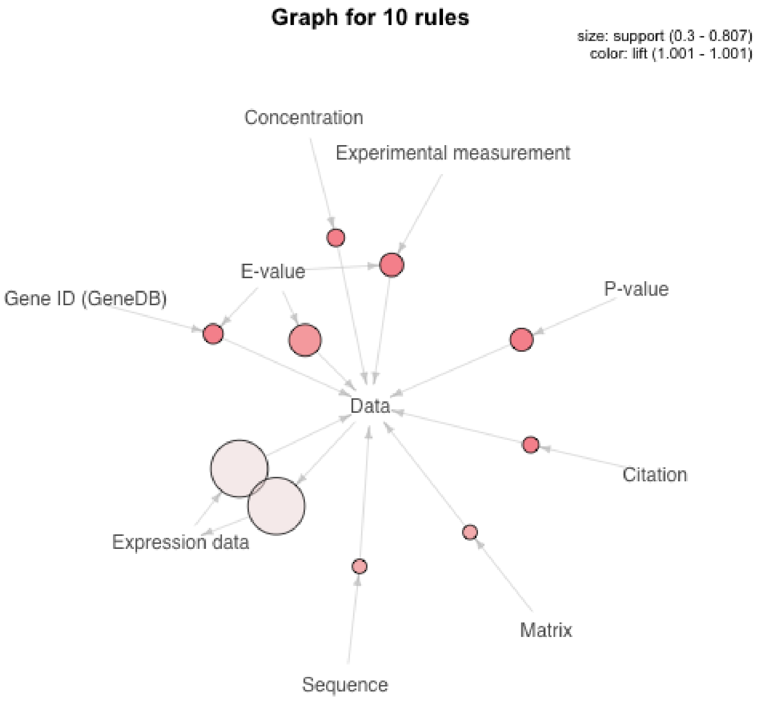


Figure 6.3. Correlations between data mentions (class ID). We plotted the first ten significant relations (ordered by support value); the size of the circle depends on the support value, the density of the colour shows the lift value. Example of two significant relations are: 1) with a high support `Data` and `Expression data` are mentioned in the same document, 2) with a lift >1 `E-value` and `Gene ID (GeneDB)` are mentioned in the same document with Data.

- EDAM classes `Calculation`, `Assigning Role`, `Data retrieval`, `Operation` and `Analysis` are highly used in data events all over the years. This usage is increased by years of the corpus.
- `Calculation` has the highest usage over the years and rose steadily by 10% every 5 years to 2010, but by a modest 4 % after then. The lemma *calculate* as well, except for the period of years between 2011 and 2019 at a document level where the highest preference was for the lemma *analyse* .
- `Comparison` is the least popular operation among the top operations. The usage over years stay steady.

Trends Figure 6.6 shows the trends over time. The main observations are:

- The trend of `Analysis` reached its peak in year 2015 and going down after that.
- `Assigning Role` and `Data retrieval` are steeply trending up over the time.

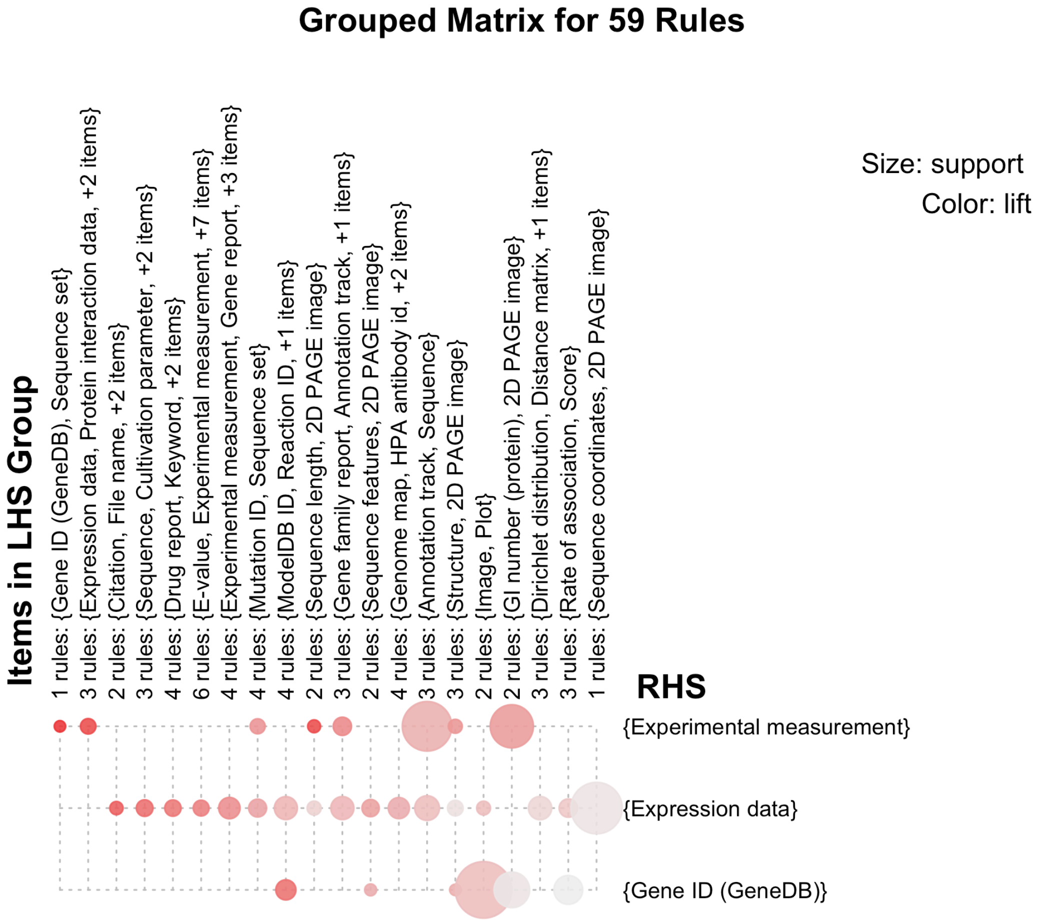


Figure 6.4. Data EDAM classes significant association rules without Data class. The number of rules are reduced to nearly the quarter. Data EDAM classes significant association rules. The left hand side of the rules are clustered and shown as columns, and the right hand side are shown in the rows. In the matrix, the size of a circle shows the support and the dense of the colour shows the lift. The rules are organised where the rules with the highest lift plotted first (left corner). For example, with a high lift, `Gene ID (GeneDB)` and `Sequence set` are significantly mentioned in the same document with `Experimental measurement`

- `Operation` sharply trends up until 2008 which climbing more slowly by then.

- `Calculation` gently trending up while `Comparison` bending down at the end of the period.

Association relations We obtained 83 significant rules that show the relations between ***operation*** mentions (class ID) over the document. Figure 6.7 plots an example of the first ten significant relations ordered by support value. Figure 6.8 shows a grouped matrix for 83 significant rules. The main remarks are that:

Table 6.5. Statistics of operation mentions in the corpus.

Characteristics	Operations
Total mentions	616,144
Unique mentions (text)	28,202
Unique mentions (lemma)	17,799
Unique mentions (class name)	748
Unique mentions (class ID)	399
Mean of total mentions per document	37.11
Median of mentions per document	35
Minimum mentions in a single document	1
Maximum total mentions in a single document	209
Number of sentences with at least one mention	455,364
Percent of sentences with at least one mention	39.67%
Mean of mentions per sentence	1.35

- `Data retrieval, Calculation` and `Analysis` are significantly used with a considerable number of ***operations***.
- `Annotation` and `Normalisation` are strongly associated, as well `Mapping` and `Comparison`.

Analysing the non-significant rules, we found that:

- `Assigning Role`, `Comparison`, `Data retrieval`, `Filtering`, `Entity identification`, `Operation`, `Standardisation and normalisation` and `Calculation` are mentioned in the same article with a high degree of confidence.
- `Analysis, Expression analysis, Standardisation and normalisation` and `Quantification` are highly used in the same context as well.
- The majority of the association rules are with a high order, which means the rule includes between five and seven ***operations***. Focusing on relations between two ***operations***, `Filtering` is highly mentioned with (`Mapping and Clustering`), (`Annotation` and `Feature selection`).

Software

Distribution There are 63,508 software mentions with 1,508 unique (text), of which only 873 can be mapped to bio.tools.

Table 6.7 shows the mean of mention level and document level of software mentions. `R` and `Bioconductor` are the top mentions for software.

Relative usage Figure 6.9 shows the relative usage of the top mentions at mention and document levels. The main observations are:

Table 6.6. Frequent mention-level mean and document level mean for operation lemmas and class ID. *calculate* is highly mentioned in a document with a mean of 1.44 at a mention level and 0.61 at a document level.

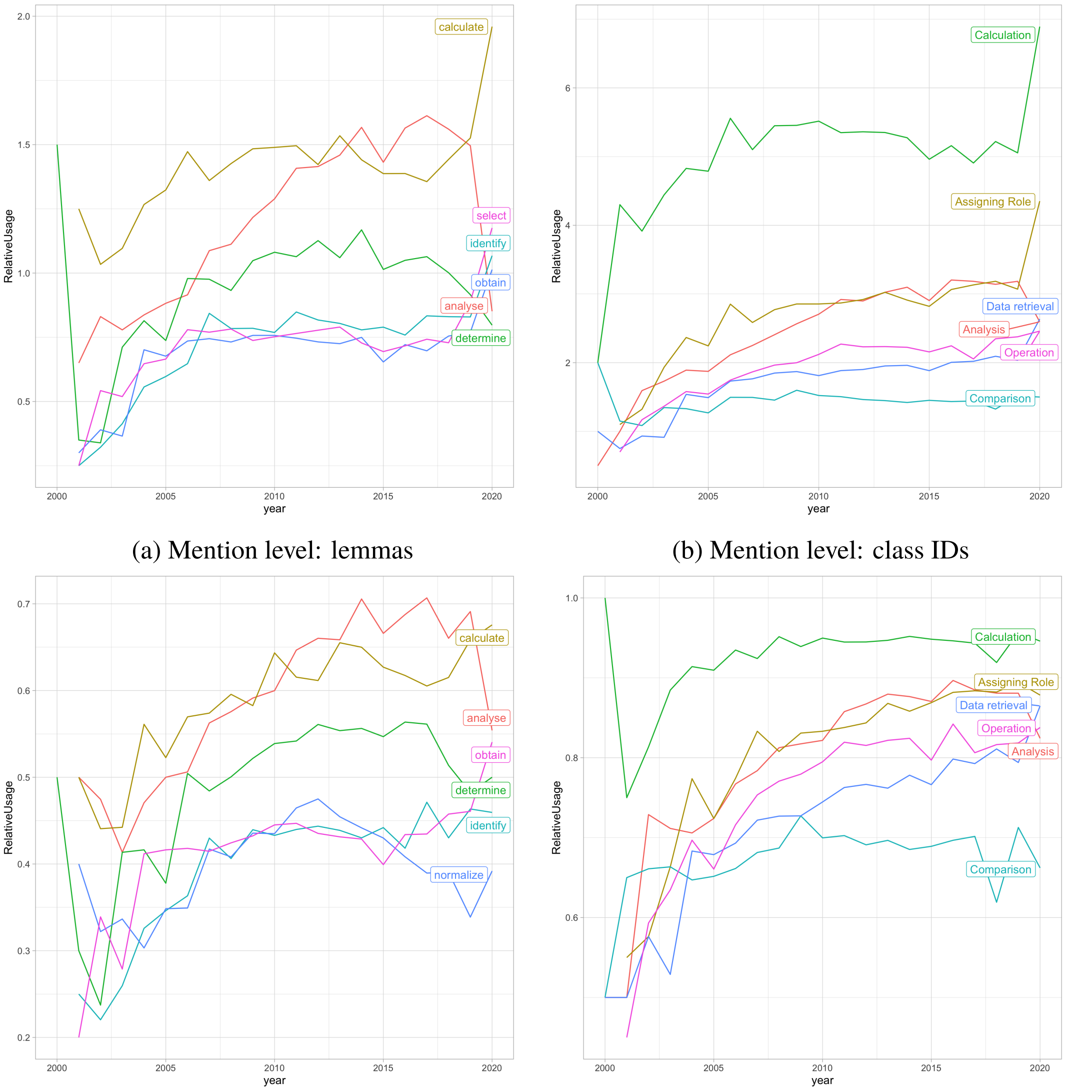
Operation lemmas				Operation EDAM concepts			
Mention level		**Document level**		**Mention level**		**Document level**	
mention	**mean**	**mention**	**mean**	**mention**	**mean**	**mention**	**mean**
calculate	1.44	analyse	0.63	Calculation	5.26	Calculation	0.94
analyse	1.35	calculate	0.61	Assigning Role	2.88	Analysis	0.85
determine	1.03	determine	0.53	Analysis	2.8	Assigning Role	0.84
identify	0.78	obtain	0.43	Operation	2.11	Operation	0.8
select	0.75	identify	0.43	Data retrieval	1.88	Data retrieval	0.76
obtain	0.73	normalize	0.42	Comparison	1.46	Comparison	0.69
compare	0.64	select	0.42	Validation	1.15	Standardisation and normalisation	0.59
assess	0.64	consider	0.4	Standardisation and normalisation	1.1	Validation	0.57
measure	0.64	compare	0.4	Generation	1.02	Correlation	0.56
normalize	0.63	use	0.4	Correlation	0.97	Generation	0.53
use	0.61	assess	0.38	Filtering	0.94	Filtering	0.47
consider	0.6	measure	0.36	Named-entity and concept recognition	0.78	Prediction and recognition	0.44
generate	0.59	generate	0.36	Feature selection	0.77	Quantification	0.44

- Bioconductor was the most second popular software, by the year 2015 the preference was for SPSS and DAVID.
- SPSS usage was very low at the beginning of the period, it passed the usage of all software, other than R, by the year 2015.
- Cluster showed a low but stable relative usage over years at the document level.

Trends Figure 6.10 shows the trends over time. The main observations are:

- R usage grew strikingly overall the period of the corpus.
- SPSS rose steadily by 1% every year to 2015, followed by a noticeable increase in the last five years.
- Bioconductor appeared in 2002, its usage reached its peak 2012 and it decreased by then evenly.
- DAIVID trend up steadily until 2015 and then trend down.

Association relations We minimised the support value to get rules. We identified ten significant association rules between ***software***. All of the ***software*** shown in these rules are associated with R. Removing R produced zero significant rules. Examples of significant relations:



(a) Mention level: lemmas

(b) Mention level: class IDs

(c) Document level: lemmas

(d) Document level: class IDs

Figure 6.5. Relative usage over the years 2000 to 2020 at the mention level and document level of the top six mentions for operation lemmas and class ID. The relative usage of document level range from 0 to 1 while it could be more than 1 in mention level. Calculation, Assigning Role, Data retrieval, Operation and Analysis usages are increased by years of the corpus (d).

- DAVID, Cytoscape and Network are strongly associated with R.
- Bioconductor and BLAT are significantly related with R,
- Ensembl and MySQL are significantly related with R.
- with a high lift, the affy and gcrma packages are significantly related to R.

Analysing the non-significant rules with a high lift degree, we found that:

- TreeView and Cluster software come together frequently.
- The affay package and Bioconductor are commonly mentioned together and with R.

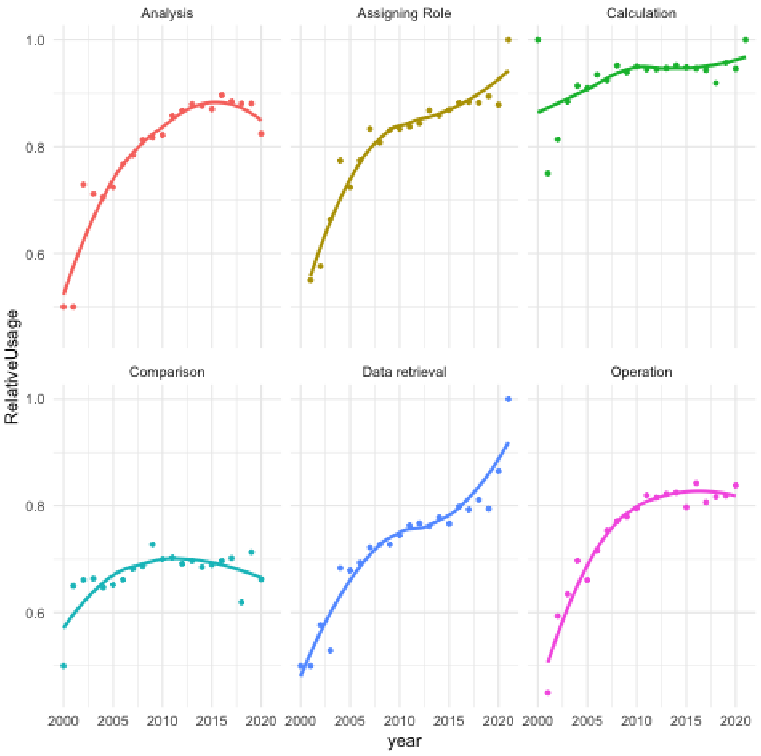


Figure 6.6. Trending of top six operation mentions (class ID). `Assigning Role, Calculation` and `Data retrieval` classes are trending up. `Comparison` and `Analysis` ended by trending down.

Table 6.7. Mention level mean and document level mean for software. R has a mean of 0.64 at mention level and 0.31 at the document level.

Software			
Mention level		**Document level**	
mention	**mean**	**mention**	**mean**
R	0.64	R	0.31
Bioconductor	0.18	Bioconductor	0.13
SPSS	0.14	SPSS	0.1
DAVID	0.1	DAVID	0.07
SAM	0.09	SAM	0.05
UCSC Genome Browser	0.07	Cluster	0.04
BLAST	0.06	GraphPad Prism	0.04
Ensembl	0.06	UCSC Genome Browser	0.04
Partek	0.05	BLAST	0.04
Cluster	0.05	Ensembl	0.04
GraphPad Prism	0.05	Network	0.04
Network	0.04	limma	0.03

- `Biocoductor` and `R` association relation has the highest support.

Databases

Distribution There are 54,121 ***database*** mentions. There are 1,577 unique mentions (text). Table 6.8 shows the mean of mention level and document level of some

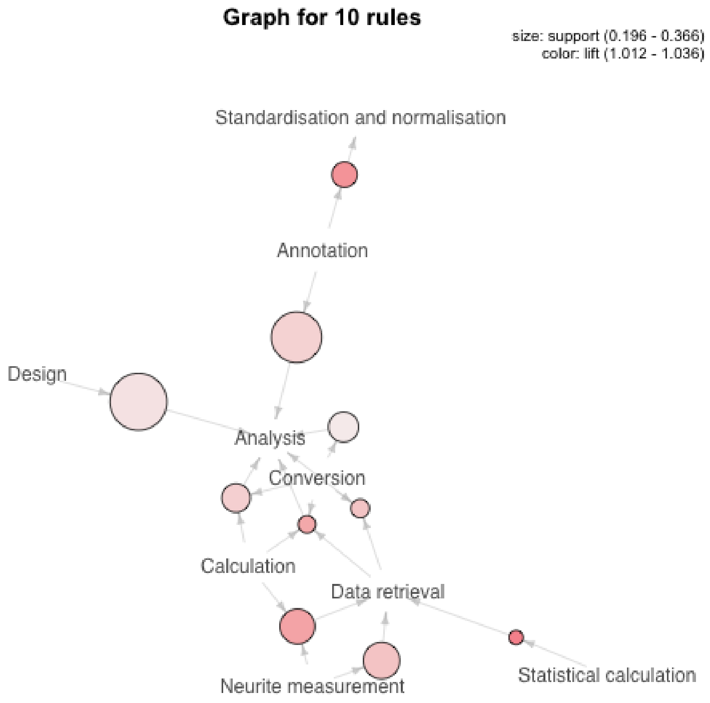


Figure 6.7. Correlations between operation mentions (class ID). We plotted the first ten significant relations (ordered by support value); the size of the circle depends on the support value, the density of the colour shows the lift value. Example of two significant relations are: 1) with a high lift `Standardisation and normalisation` class is significantly mentioned with `Annotation`, 2) with a high support `Annotation` is mentioned with Analysis.

annotated ***databases***. `GEO (Gene Expression Omnibus)` has the highest mean mentions at both levels (0.52 and 0.28 respectively). `GO (Gene Ontology)`, `ATCC (American Type Culture Collection)`, `KEGG (Kyoto Encyclopedia of Genes and Genomes)` and `GenBank` are the top used ***databases*** based on the mean of the mentions.

Relative usage Figure 6.12 shows the relative usage of the top mentions at mention and document levels. The main observations are:

- At the mention level, in the first eight years `GO` was the most used database. This was overtaken by `GEO` which reached its peak in 2013 then slowly declined but still had a relatively high usage compared to other databases. At the document level, `GEO` was the highest since 2004 and remained so until the end of the period.
- `ATCC` document level usage was higher than `GO` from the period of 2013 till 2018. `GO` returned to its higher usage by then until the end of the period.
- `GeneBank` usage declined since 2005. The mean of relative usage went from 0.25 to 0.05 in 2019.

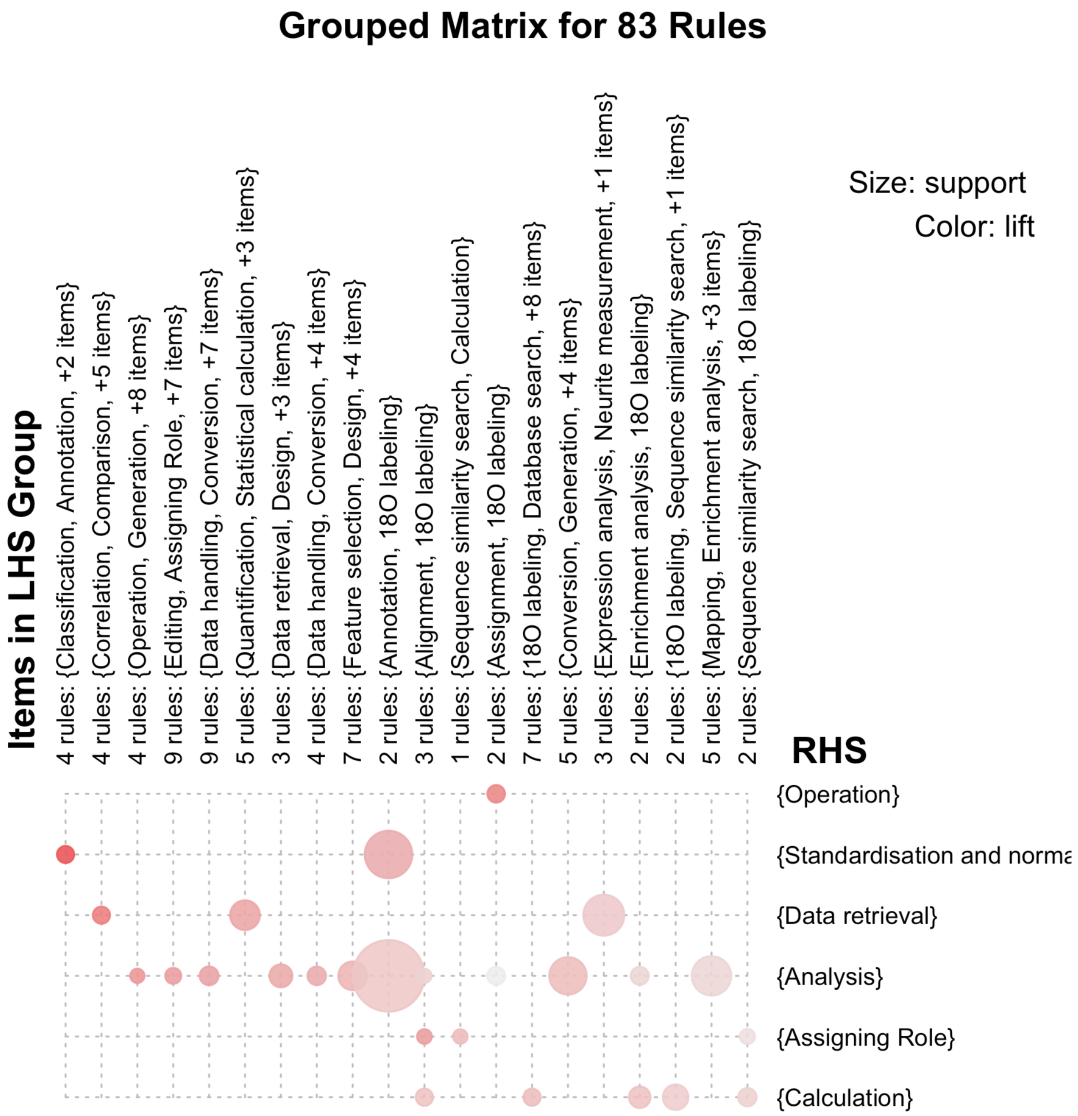
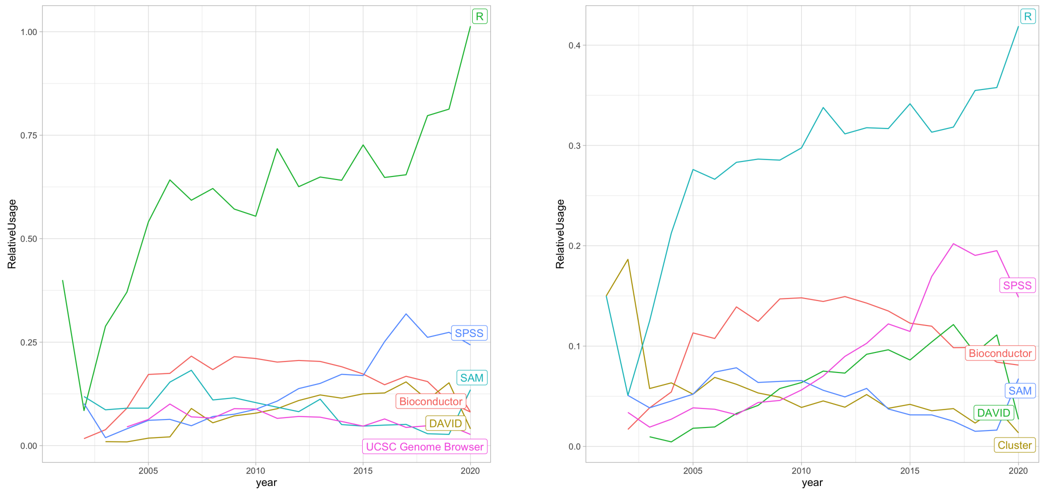


Figure 6.8. There are 83 significant association rules between operations mentions (class ID). An example of grouped relations `Standardisation and normalisation` on right hand side of the relation has six relations. Two with a high support (size of the circle) `Annotation and 18O labeling`. The other four relations with high lift (colour of the circle) with `Annotation, Assigning Role, Calculation` and `Classification`.



(a) Mention level

(b) Document level

Figure 6.9. Relative usage of top software. `R` is the most frequent used software over the corpus. `SPSS` overcome the usage of all software, other than `R`, by the year 2015.

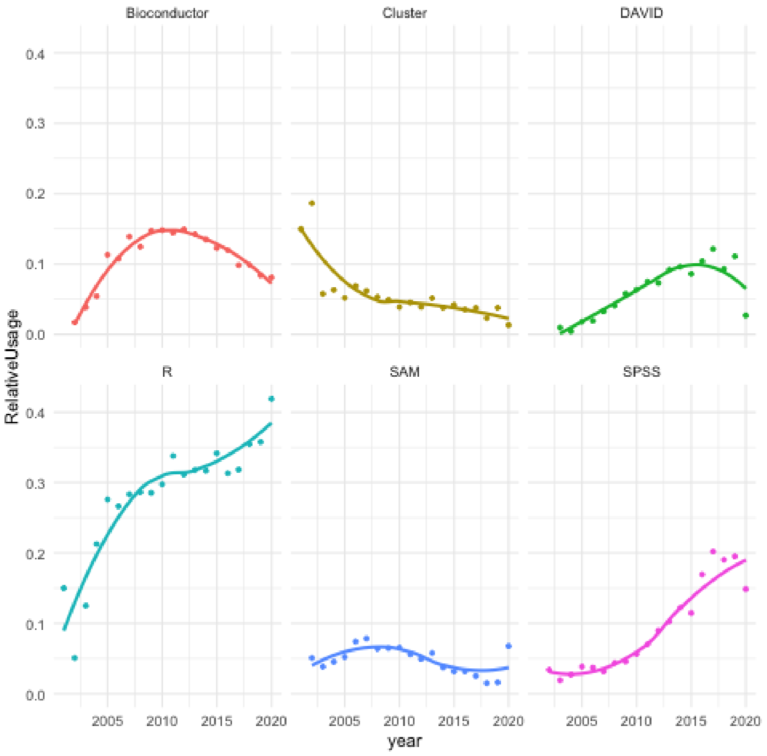


Figure 6.10. Trending of top six software. `R` and `SPSS` are trending up. `Bioconductor` and `DAIVD` end up by trending down.

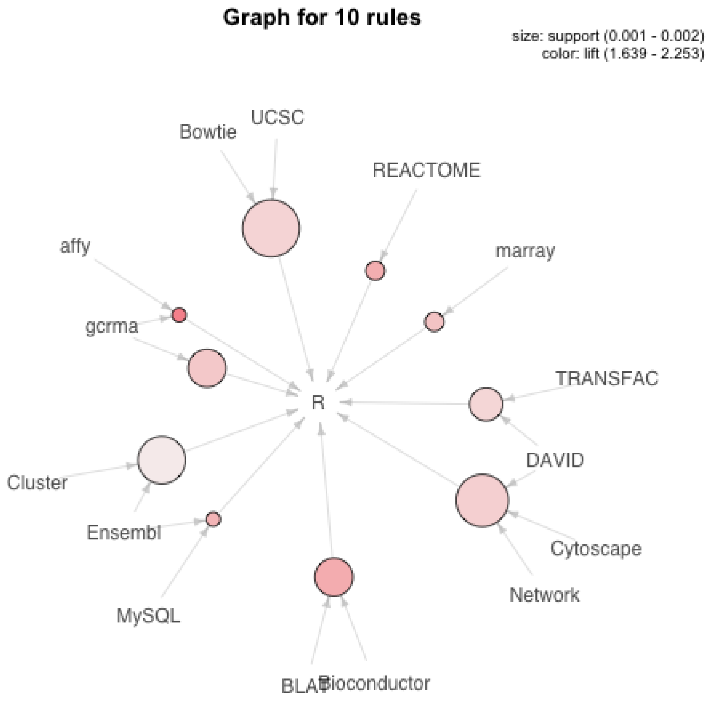


Figure 6.11. Correlations between software mentions (text). We plotted the ten significant relations; the size of the circle depends on the support value, the density of the colour shows the lift value. Example of two significant relations are: 1) `Bioconductor` and `BLAT` are significantly related to `R`, 2) with a high lift, the `affy` and `gcrma` packages are significantly related to `R`.

Trends Figure 6.13 shows the trends over time. The main observations are:

- The usage of `GEO` had a steep upward trajectory from nearly zero to nearly 33% in the first 11 years, it then declined slowly by around 10% at the end of the period.
- `GO` usage fluctuated over the years but ended up with upward usage in the last five years.
- `ATCC` and `KEGG` showed a similar slow increase in usage.
- `GeneBank` usage trended down slowly all over the years.

Association relations

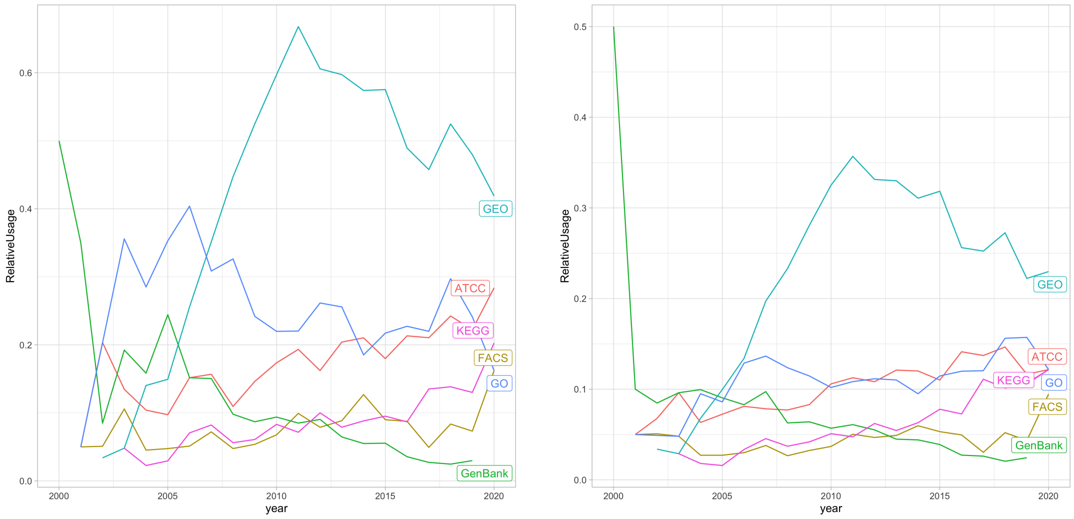
We minimised the support as we did in software association relations. There are five significant association rules between databases mentions (see Figure 6.14).

- `Gene Expression Omnibus (GEO)` is significantly associated with 9 ***databases***.
- `DEG` and `KEGG` are associated with `GEO` with a high support among the relations.

- GO and WGCNA are associated with GEO with a high support and high lift among the relations.

Analysing the non-significant rules with a high lift degree, we found that:

- The Biomolecular Interaction Network Database (BIND) and Human Protein Reference Database (HPRD) are highly mentioned within the same documents.
- dbEST and GenBank are highly mentioned within the same documents.
- The five databases, Gene Ontology (GO), Gene Ontology Annotation (GOA), AmiGO, GEO and Kyoto Encyclopedia of Genes and Genomes (KEGG), are frequently mentioned in the same document.



(a) Mention level

(b) Document level

Figure 6.12. Relative usage of the top six databases. GEO is the most popular database. Mention level of GO was more than GEO until 2007.

6.3.4 Linking components

We studied the relations between two pairs of components then between all components. We studied the relations at two levels; document level and data event level.

Operation and Data

Document level

Examples of significant relations between ***operations*** and ***data***:

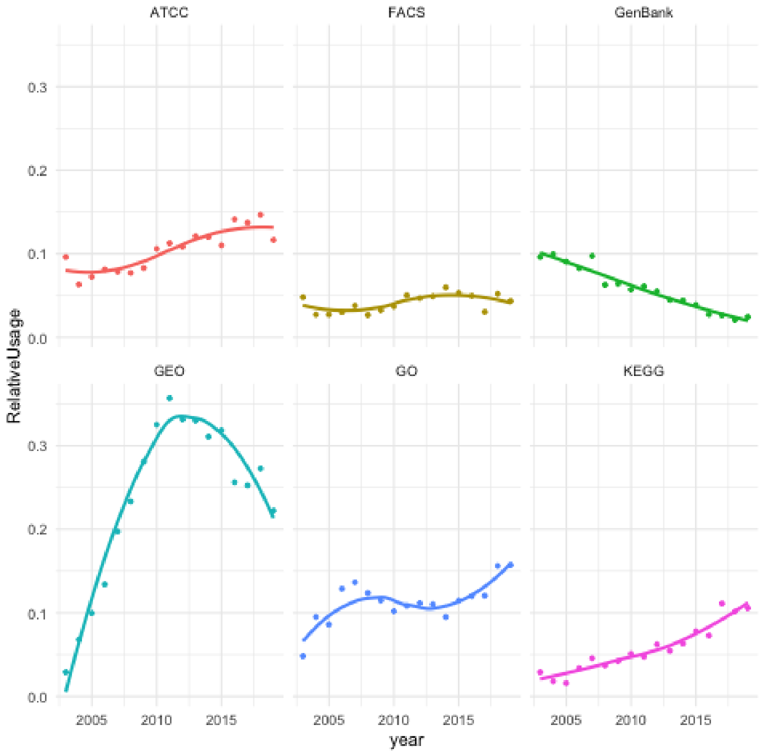


Figure 6.13. Trending of top six databases. Most showed a slow upward trending. GenBank is trending down.

Table 6.8. Mention level mean and document level mean for databases. GEO has a mean mention level of 0.52 and a mean of 0.28 at the document level.

Databases			
Mention level		**Document level**	
mention	**mean**	**mention**	**mean**
GEO	0.52	GEO	0.28
GO	0.25	GO	0.11
ATCC	0.18	ATCC	0.11
KEGG	0.09	KEGG	0.06
GenBank	0.08	GenBank	0.05
FACS	0.08	FACS	0.04
GSEA	0.06	RefSeq	0.03
RefSeq	0.05	Guide	0.03
Guide	0.03	GSEA	0.03
MEM	0.03	Blue	0.02
UniGene	0.02	MEM	0.02

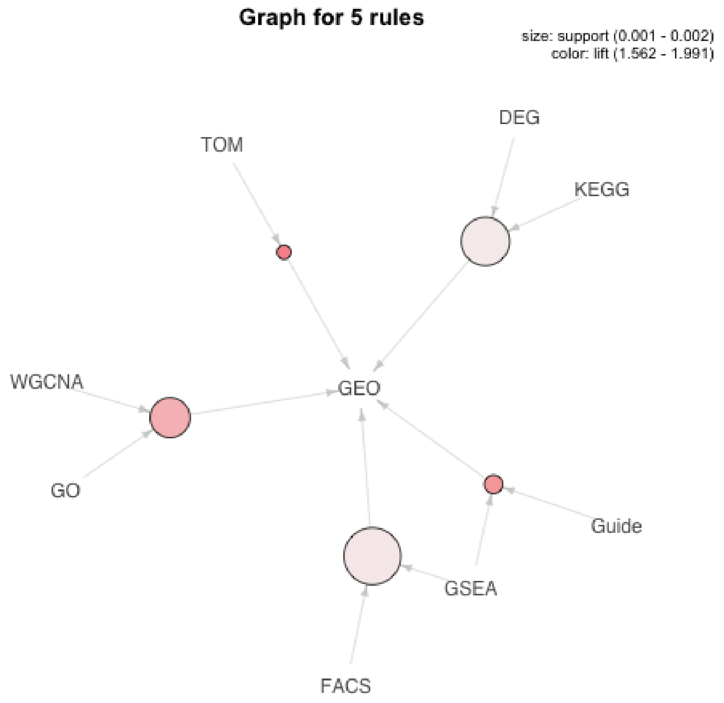


Figure 6.14. Correlations between database mentions (text). We plotted the first ten significant relations (ordered by support value); the size of the circle depends on the support value, the density of the colour shows the lift value. Example of a significant relation is `GO` and `WGCNA` are associated with `GEO` with a high support and high lift among the relations.

`Rel<`**`operation:`**` Comparison, `**`data:`**` Cultivation parameter, Score>`

-`Cultivation parameter` class includes classes such as `Temperate, Cultivation conditions`, etc.

`Rel<`**`operation:`**` Visualisation, Statistical calculation `**`data:`**` Gene ID (GeneDB)>`

`Rel<`**`operation:`**` Annotation, Conversion, Alignment, Assignment `**`data:`**` Expression data>`

Examples of frequent but non-significant relations:

`Rel<`**`operation:`**` Design, Comparison, `**`data:`**` Expression data, PCR primers>`

`Rel<`**`operation:`**` Editing, `**`data:`**` Expression data, Gene ID (GeneDB), P-value>`

`Rel<`**`operation:`**` Editing, Standardisation and normalisation, `**`data:`**` Expression data, P-value>`

`Rel<`**`operation:`**` Named-entity and concept recognition, `**`data:`**` Pathway` `or network>`

Event level

`Rel<`**`operation:`**` Editing, `**`data:`**` Q-value, P-value>`

`Rel<`**`operation:`**` Ontology visualisation, `**`data:`**` Gene ID (GeneDB)>`

`Rel<`**`operation:`**` Classification, `**`data:`**` GO concept ID, Gene ID (GeneDB)>`

`Rel<`**`operation:`**` Statistical calculation, `**`data:`**` Experimental measurement>`

Operation and Software

Document level There are no significant relations on the document level since the operations mentions (class ID) are more frequent at the document level than software. We removed the frequent operations `Calculation` and `Analysis`.

Examples of significant relations between operations and software:

`Rel<`**`operation:`**` Data retrieval, `**`SW:`**` R>`

`Rel<`**`operation:`**` Standardisation and normalisation, Data retrieval, Operation, `**`SW:`**` Bioconductor>`

Examples of frequent but non-significant:

`Rel<`**`operation:`**` Standardisation and normalisation, Comparison, Operation, `**`SW:`**` R>`

Event level For example, the operation is frequently mentioned in the same events with the software.

`Rel<`**`operation:`**` Clustering, `**`SW:`**` Cluster, TreeView>`

`Rel<`**`operation:`**` Standardisation and normalisation, `**`SW:`**` R, limma >`

`Rel<`**`operation:`**` Expression analysis, Statistical calculation, `**`SW:`**` R, Bioconductor >`

`Rel<`**`operation:`**` Design, `**`SW:`**` Primer3>`

`Rel<`**`operation:`**` Gene functional annotation, `**`SW:`**` DAVID >`

Operation and Databases

Document level There are no significant relations at the document level since the operations mentions (class ID) are more frequent at the document level than databases. We removed the frequent operations `Calculation` and `Analysis` to obtain some significant relations with other classes. Example of a significant relation:

`Rel<`**`operation:`**` Data retrieval, `**`DB:`**` GEO>`

Example of non-significant but a high lift relation:

`Rel<`**`operation:`**` Deposition, Operation, `**`DB:`**` GEO>`

Event level Examples of non-significant but a high lift relation:

`Rel<`**`operation:`**` Pathway analysis `**`DB:`**` KEGG(Kyoto Encyclopedia of Genes and Genomes))>`

`Rel<`**`operation:`**` Beacon Designer, `**`DB:`**` Design>`

Data and Software

Document level Example of frequent relations:

`Rel<`**`data:`**` Experimental measurement, P-value, Gene ID (GeneDB),`

`Expression data, `**`SW:`**` R, Bioconductor >`

Event level Example of frequent relations:

`Rel<`**`data:`**` PCR primers, `**`SW:`**` Primer Premier>`

`Rel<`**`data:`**` PCR primers, `**`SW:`**` Primer3>`

`Rel<`**`data:`**` P-value, Q-value, `**`SW:`**` R>`

`Rel<`**`data:`**` Pathway or network, Protein interaction data, `**`SW:`**` Cytoscape>`

`Rel<`**`data:`**` Gene expression profile, Expression data, Gene ID (GeneDB), `**`SW:`**` Cluster, TreeView>`

Data and Databases

Document level Example of frequent relations:

`Rel<`**`data`**`: Expression data, P-value, Gene ID (GeneDB),` **`DB`**`: GEO >`

`Rel<`**`data`**`: Expression data, Gene ID (GeneDB),` **`DB`**`: GO >`

Event level Example of frequent relations:

`Rel<`**`data`**`: PCR primers,` **`DB`**`: Beacon Designer>`

`Rel<`**`data`**`: Gene symbol,` **`DB`**`: HGNC(HUGO Gene Nomenclature Committee)>`

`Rel<`**`data`**`: Sequence,` **`DB`**`: dbEST, GenBank>`

`Rel<`**`data`**`: Mathematical model, Pathway or network,` **`DB`**`: KEGG>`

`Rel<`**`data`**`: Raw microarray data,` **`DB`**`: GEO >`

`Rel<`**`data`**`: Expression data, Accession, GEO accession number, Sample ID, Protocol, File name,` **`DB`**`: GEO >`

This relation is a group of five relations. At the event level, database GEO and `Expression data` have an extra relation with different data such as `Accession, GEO accession number, Sample ID, Protocol` and `File name`

Software and Databases

Document level We lowered the support and remove the frequent software (`R` and `Bioconductor`) to have different frequent relations. Examples of non-significant but lift>1 are:

`Rel<`**`SW`**`: Entrez Gene,` **`DB`**`: GO, NCBI Entrez>`

`Rel<`**`SW`**`: Cytoscape, SPSS,` **`DB`**`: GO, KEGG>`

`Rel<`**`SW`**`: Cluster, TreeView,` **`DB`**`: GEO>`

Event level

Examples of non-significant but lift>1 are:

`Rel<`**`SW`**`: ArrayExpress,` **`DB`**`: GEO>`

`Rel<`**`SW`**`: MINT, IntAct,` **`DB`**`: BIND (The Biomolecular Interaction Network Database), HPRD (Human Protein Reference Database)>`

Operation, Data, Software and Databases

Document level

Extracting significant rules for all components generates thousands of rules associated with the most frequent components. We removed the frequent components such as `R, Data` and `Calculation` and obtained 1,643 rules. Figure 6.16 shows the grouped matrix for these rules. The right hand side of the significant rules is `GEO,` `Expression data` or `Analysis`. Figure 6.15 plots ten examples of these rules. Here are some examples:

`Rel<`**`data`**`: Expression data, `**`operation`**`: Read pre-processing, `**`SW`**`: Bioconductor `**`DB`**`: GEO>`

`Rel<`**`data`**`: Expression data, `**`operation`**`: Data retrieval, `**`SW`**`: MATLAB `**`DB`**`: GEO>`

`Rel<`**`data`**`: Expression data, `**`operation`**`: Data retrieval, `**`SW`**`: UCSC Genome Browser `**`DB`**`: GEO>`

`Rel<`**`operation`**`: Data retrieval, `**`data`**`: Gene ID (GeneDB), Expression data, `**`SW`**`: Entrez Gene>`

`Rel<`**`operation`**`: Analysis, `**`data`**`: Gene ID (GeneDB), Expression data, `**`SW`**`: NetAffx>`

`Rel<`**`operation`**`: Data retrieval, Analysis, `**`data`**`: Gene ID (GeneDB), `**`DB`**`: DEG>`

Event level

To analyse links between all data event components, we removed the `Data` and `R` since they are frequent in events. There were no significant links identified, but the frequent ones were for example:

`Rel<`**`data`**`: Microarray data, `**`operation`**`: Deposition, `**`DB`**`: GEO>`

`Rel<`**`data`**`: Gene ID (GeneDB), `**`operation`**`: Analysis `**`SW`**`: DAVID, `**`DB`**`: GO>.`

`Rel<`**`data`**`: Network, `**`operation`**`: Visualisation, Construction, `**`SW`**`: Cytoscape>`

For the lift measurement, the events with higher value included:

`Rel<`**`data`**`: Array, Pathway over-representation report, `**`SW`**`: DAVID, `**`DB`**`: GO, DEG, KEGG>`

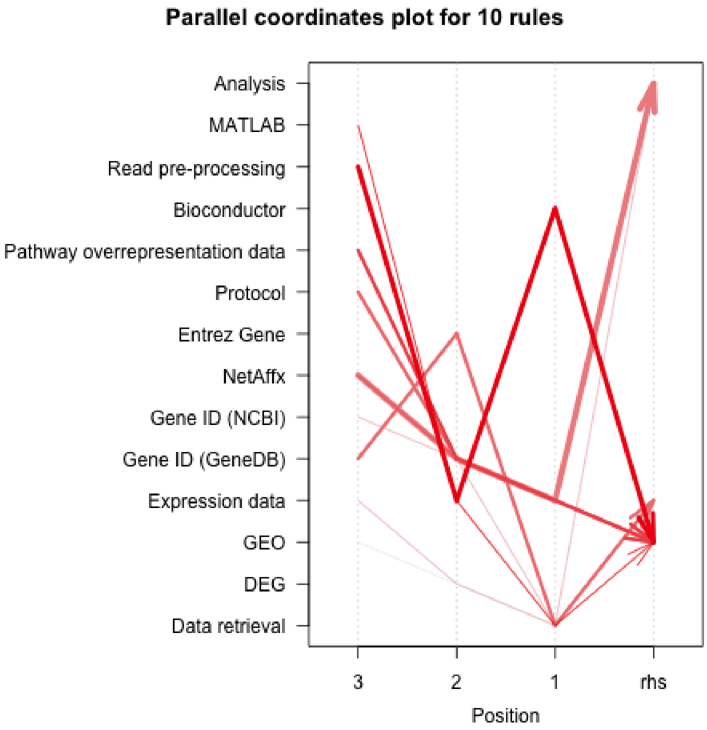


Figure 6.15. Example of 10 significant association rules between all components at the document level shown in parallel coordinates. We removed frequent components such as `Data, R` and `Calculation`. An example: Bioconductor (position 3), Expression data (position 2), Read pre-processing (position 1) => GEO(rhs) this relation can be represented as `Rel<`**`data:`**` Expression data, `**`operation:`**` Read pre-processing, `**`SW:`**` Bioconductor `**`DB:`**` GEO>`

`Rel<`**`data:`**` File name,Gene expression profile identifier, `**`SW:`**` affy, Bioconductor, `**`DB:`**` GO, DEG, KEGG>`

6.4 Discussion

The corpus shows that 40% of the sentences of Methods section describes the work conducted by mentions of ***operations*** and ***data***. It also shows that each sentence has an average of one ***operation*** and two ***data*** instances, which is expected (input/output data instances linked by an operation).

The normalisation of ***operations*** shows a better result than ***data*** normalisation in terms of mentions left to be mapped to the top classes `Operation` and `Data` (18% versus 52% at unique text mentions). Although with a modest mapping result, normalising the data to EDAM concepts gives an idea of type of classes used in microarray analysis such as `Expression data, Gene ID, Sequence, dbProb ID`, etc. and we can also follow whether the usage of these classes are trending up or decline through the year of the corpus.

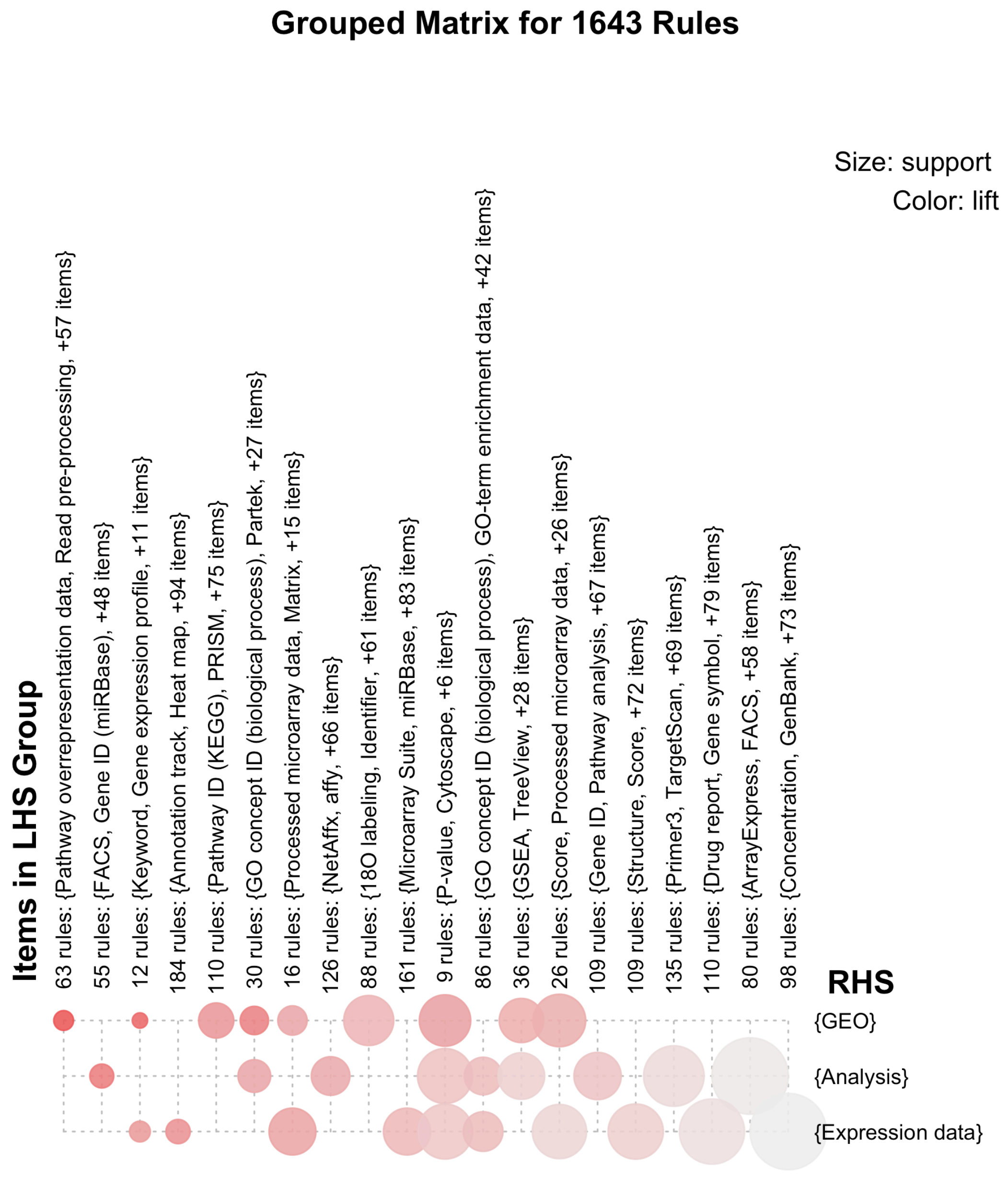


Figure 6.16. Significant association rules between all components at the document level. We removed frequent components such as `Data, R` and `Calculation` and we used class ID mentions for data and operations.

This survey produced more normalisation classes than the one produced in Chapter 4. This normalisation result needs to be analysed to write more normalisation rules that would find better class match to a mention. For example, data mentions such as *RNA, DNA, cDNA* found an exact match in Topic EDAM but matched to class `Data` since the possible similar classes are not within an acceptable similar syntax distance. *RNA*, could be matched to `RNA-seq data, RNA profile` or `RNA sequence` Data classes.

From the mean of the mention level, the same ***operation*** or ***data*** could be repeated more than once in the same document. The mean at the mention level of the software and databases mentions did not indicate the same thing. Moreover, most of the databases and software mentions have a low document level usage compared to other components. It is noticeable that the usage of operations and data in describing the work conducted is more than mentioning the software or database used.

Studying the rules at two different levels shows different level of relations. At the document level we can have a general idea of frequently mentioned components at the document level. This can be filtered by significant measurement as well. At the event level we were zoning the study group between the components that have relation in data events. We did not obtain significant relations, but we could spot the relations that were hidden by frequent components at the document level. For example, `Expression data` with database `GEO` are frequent at the document level. At the event level, they are combined with more data that are not appeared at the level of document.

Studying the significant rules is important, however, non-significant but frequent rules show interesting results as well. For example, `P-value` is frequently mentioned with `FDR` as class name, a synonym to class ID `Q-value`. These two data are usually handled by software `R`.

Although we visualised the relations in group matrices, we cannot explore all the significant or frequent relations. However, if we are interested in a specific component, we can investigate its relations through the produced rules.

Duck et al. [11] stated that the usage of software is significantly different among different biomedical literature fields; the mentions of software and databases are high in bioinformatics. We found that in this paper domain, microarray analysis, the mentions of the software and databases are less than the mentions reported by Duck et al. [11] in their survey. They also showed that 78% of the surveyed resources were used only once over 15 years. In our analysis, 44% of resources were mentioned once over 20 years.

6.5 Conclusion

In this paper, we have demonstrated the feasibility of using the large-scale automatically extraction of methodology (i.e. operations, data, databases and software) to identify the usage patterns from a large corpus of microarray articles. Our survey highlights the changes over 20 years where new “types” of data events emerged and old ones declined. For example, `Data retrieval, Assigning role, Sample ID, P value, R, SPSS` and `KEGG` show no sign of levelling off.

We explore the relations between components operations, data, databases and software and found some interesting patterns such as genes or expression data could be deposited to GEO database and retrieved using UCSC Genome Browser or Entrez Gene. The networks could be visualised by Cytoscape.

Overall, the result obtained gives an insight into the data events components characteristics and provides a general overview of core computational biology common processes and data.

Chapter 7

Discussion and Conclusion

This chapter discusses the various topics covered within this thesis, in particular highlighting the key findings that answer the research questions, and explores the challenges and limitations of this work, as well as potential future work.

Identifying, following and understanding methodologies reported in the plethora of publications demands huge effort as these methods are usually represented in prose in scientific publications. Representing methodologies in a computational model would offer a great opportunity to search, analyse and better understand the methodological landscape in a given domain. Figure 7.1 provides an overview of the primary steps developed and discussed in this thesis:

1. Corpus generation and filtering
2. Document pre-processing
3. ODNoR operation and data name recognition, normalisation and relation extraction
4. ODNoRFlow workflow extraction
5. Workflow storage and representation
6. Pattern generation to identify commonly used ***operations***, ***data***, ***software*** and ***databases*** in the extracted workflows.

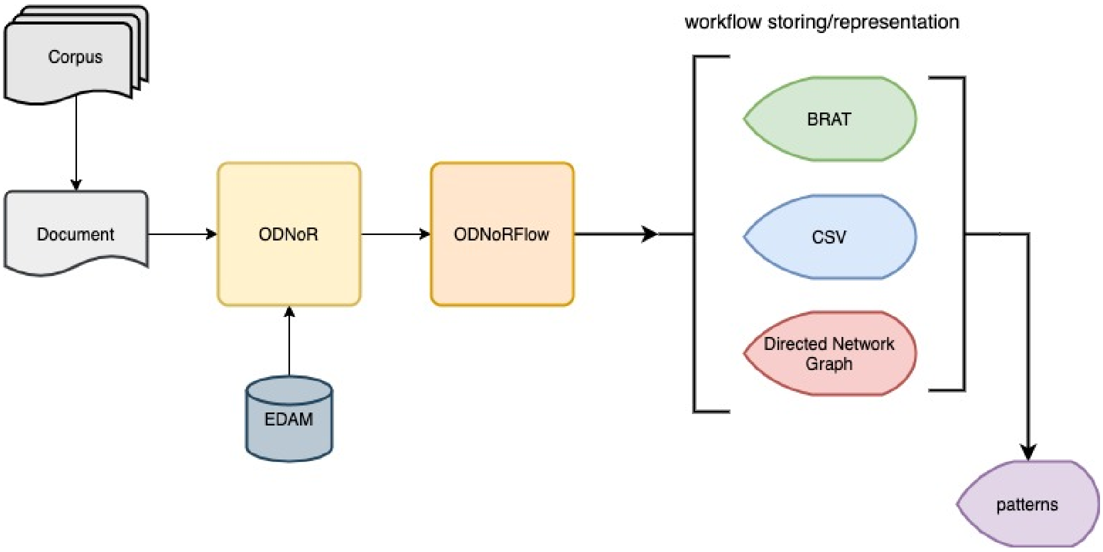


Figure 7.1. Graphical representation of the thesis' primary steps.

7.1 How our findings answer the research questions

7.1.1 Revisiting the research questions and objectives

The main hypothesis of this thesis is that — by using text mining techniques — we will be able to automatically identify, extract and represent computational biology methods that are reported in individual scientific articles. To check that hypothesis, we aimed to design, develop and evaluate a methodology to extract and represent data event workflows from the computational biology literature, using microarray analysis as a case study.

Overall, we have found that — in the microarray analysis literature — we can extract an abstract representation of method workflows. We now review each objective and list the relevant findings that help answer our research questions and reach our aim.

- **Objective 1: To conduct a survey of how methods sections are written.**

  This objective answered the first research question. We surveyed the microarray analysis literature in Chapter 3 to explore how methods sections are written in terms of functional discourse. Discourse functions were extracted by the **SAPIENTA** system. We found that most of the Methods section discourse functions are `Methods` and `Experiments`, with few mentions of `Models` and `Results`. We identified six common patterns (listed below in the SPS format with shortened names for the functions: Bac, Met, Mod, Exp, Mot, Goa, Hyp, Obj, Obs, Res, Con as short for `Background`, `Method`, `Model`, `Experiment`, `Motivation`, `Goal`, `Hypothesis`, `Object`, `Observation`, `Result` and `Conclusion`).

  Pattern P1 (Met,1)-(Exp,19)-(Met,13)-(Exp,4)-(Met,2).

  Pattern P2 (Met,13)-(Bac,1)-(Met,6)-(Res,3)-(Met,14)-(Res,1)-(Exp,14)-(Obs,1).

  Pattern P3 (Obj,1)-(Met,1)-(Res,2)-(Exp,13)-(Res,1)-(Met,2)-(Res,1)-(Met,4)-(Exp,23)-(Res,3)-(Exp,13)-(Bac,1).

  Pattern P4 (Exp,2)-(Res,5)-(Met,7)-(Exp,1)-(Res,1)-(Exp,6)-(Met,9)-(Res,4)-(Exp,17)-(Res,2)-(Mod,6)-(Exp,5)-(Met,7).

  Pattern P5 (Res,3)-(Con,2)-(Bac,4)-(Mot,1)-(Res,3)-(Con,1)-(Res,5)-(Met,2).

  Pattern P6 (Bac,7)-(Met,22)-(Exp,11)-(Met,29)-(Res,2).

  We explored and reported the usage of these patterns over the years that were covered by the corpus. We also reported the patterns usage in the top four microarray analysis journals. Moreover, we demonstrated the relation usage between the discourse functions over the paper's section. We found 12 statistically significant relations that show there was a high chance when a number of sentences of one function was increased in a section, that the other function was increased as well. We found that there is a strong positive relationship between (`Model` and `Method`), (`Goal` and `Object`), and finally between (`Result` and `Observation`). There is also a fairly strong positive relationship between (`Method` and `Experiment`), (`Background` and `Motivation`), (`Conclusion` and `Hypothesis`), (`Hypothesis` and `Result`), (`Observation` and `Object`), (`Observation` and `Goal`), (`Object` and `Result`) and finally between (`Goal` and `Result`).

We explored and reported the transition probabilities between the functions. We could say that in the Methods section, the functions that belong to the approach category (i.e. `Experiment`, `Method` and `Model`) are highly probable to come in a successive pattern. It is also noticeable from the state diagram that the transition for any function has a high chance of being to `Method`.

As reported in Chapter 5, the rhetorical structure theory (RST) shows that the majority of the Method sections sentences have a discourse relation of `Elaboration`, which makes no significant difference to reporting or using them in finding the writing structure patterns.

- **Objective 2: To explore the use of text mining techniques, such as named entity recognition (NER), discourse analysis (DA) and domain relation extraction (RE) to identify data events.**

  This objective targeted the second research question, which we addressed by exploring text mining techniques in Chapter 2, applying named entity recognition and relation extraction techniques in Chapter 4 and the discourse analysis in Chapter 3.

  At the early stages of the project, we explored rule-based NER and studied the syntax of the text to design rules that could extract the ***data*** and ***operations***. However, it gave a modest result compared to the machine and deep learning techniques. We used the transformers pre-trained on a large number of data: specifically, we used BioBERT, which is a BERT model trained on PubMed articles, obtaining an F-score between 78% and 92%.

  We used bioNerDS [71] to extract the other two main components of the work reported: ***software*** and ***databases***.

  We linked these main components with four types of relations: ***input data, output data, by software*** and ***from database***. Relation extraction was performed by using machine learning and rule-based systems, and we obtained an F-score between 62% and 92%.

We also explored the benefits of discourse analysis. In Chapter 5, we tested the functional and RST discourse outcomes to improve the result extracted by the ODNoR system. Using one type of discourse did not enhance the result. Combining multiple types of discourse could improve the accuracy of the extracted events to some extent. However, the level of the accuracy of the current state-of-the-art systems is not sufficient to be applied reliably to the computational biology field.

Overall, the results we obtained in this thesis indicate that text mining can be used to identify data events, but that the accuracy may still be relatively low for recognising and linking specific components.

- **Objective 3: To model the data events mentioned in methods by defining templates that contain information about processes conducted and reported in the text.**

  This objective targeted the third question and was addressed in Chapter 5. We defined a data event as the operation on data performed by operator(s). In our context, the data is processed by operators that can be software (e.g. BLAST), tools (e.g. Microsoft Excel) or different specific methods (e.g. Student's t-test).

  The representation of a data event was inspired by the representation of the events in discourse analysis [97]. A template for a data event was based on the suggestions reported in [102]. We represented the semantic roles of an event as the components involved in accomplishing a task. They are **operations**, **input data**, **output data**, **software** and **databases**. An abstract representation of the data event is:

  <***input data:****data*, ***operation:****operation*, ***input data:****data*, ***software:****software*, ***database:****database*>

  We defined two types of data events. The first type records explicit events done by operators; we refer to these simply as *data events*. Another type is called *implicit* data events, which keep track of a reference to the data used without an accompanying task in the same sentence.

- **Objective 4: To represent extracted data events in abstract workflow diagrams, and evaluate the workflow construction in a case study on microarray analysis research.**

This objective targeted the third research question and was addressed in Chapter 5. Data events from a paper's Methods section are combined in a sequence order to construct the workflow of that paper. The events are ordered as they are mentioned in the Methods section. The workflow can be viewed as a directed connected graph, a sequence of records in a CSV file or a sequence of events in the BRAT format. Figure 7.2 represents an abstract workflow for a main article's Method section shown in Appendix C.

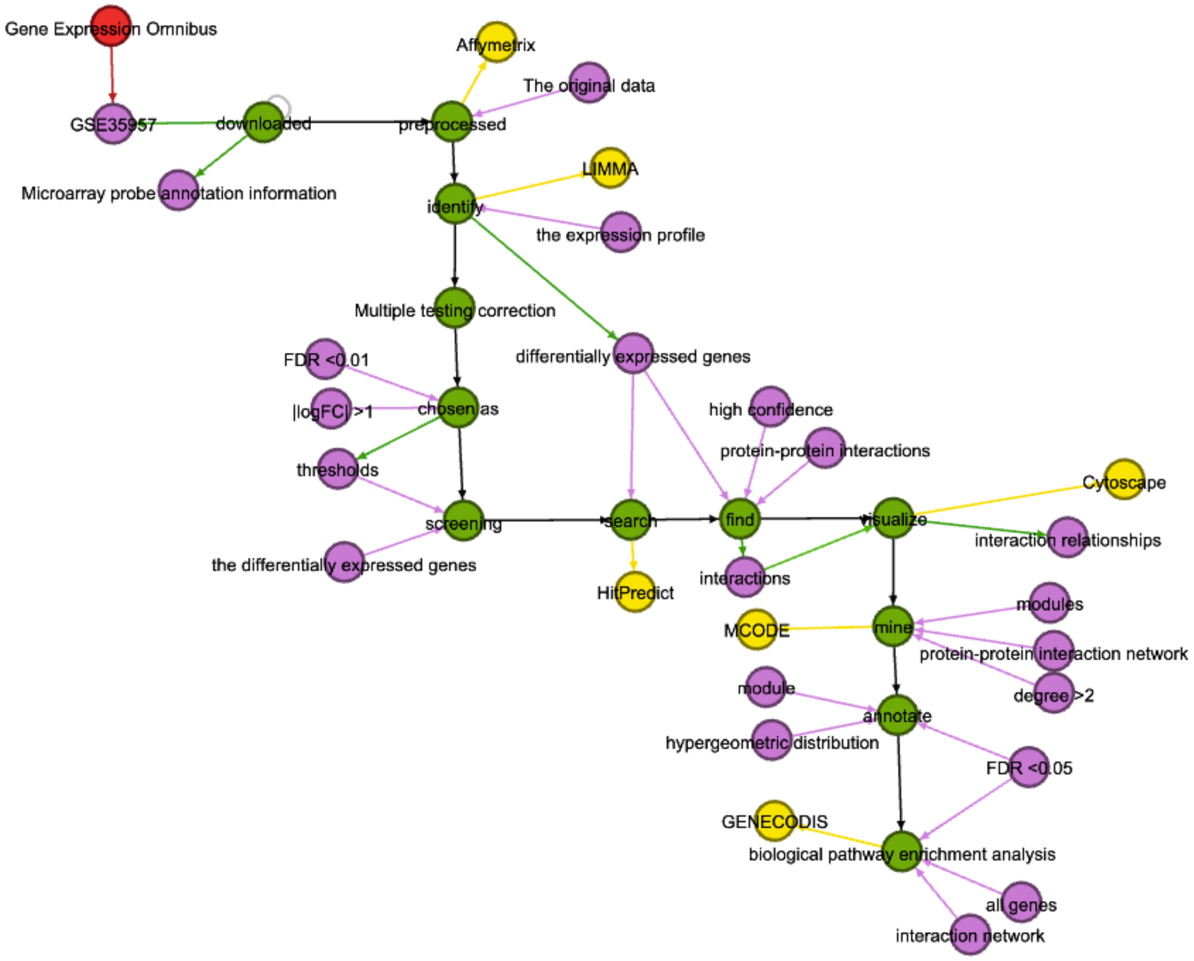


Figure 7.2. An example of a workflow abstract in a network graph.

Evaluation of generated workflows was a challenge. We separately evaluated each step that was required to be performed before constructing the abstract workflow in Chapter 4. ***Operations***, ***data***, ***software*** and ***databases*** annotations are evaluated. Relations between the components (***input data, output data, by software*** and ***from database*** relations) were also evaluated. We manually evaluated the workflows by examining *completeness* (which relates to entities and relations inside an event , and *sequentiality of events in a workflow* (which relates to the order of the data events in the workflow) in Chapter 5. For completeness, we considered how well the extracted data events represent i.e. capture a data event against the manual annotation. We obtained an F-score between 61% and 93.56% with a substantial inter-annotator agreement.

- **Objective 5: To demonstrate how the computational data event workflows can be used to identify common patterns of method over time.**

  This objective targeted the fourth research question and was explored in Chapter 6. We were able to extract and analyse the abstract workflows for the microarray analysis papers over the last 20 years. We analysed the distribution and relative usage of ***data, operations, software*** and ***databases*** over time. We explored the trending of some key concepts for each component.

  We presented data events components association relations as reported in documents and as they appeared in the extracted data events. We first presented the patterns of each component separately, and then we found the patterns of pairs (e.g. ***data*** and ***operations***) and all four components together.

7.2 Challenges

In undertaking this work, the following challenges were met:

1. The ambiguity of free-text data is one of the main concerns in extracting an abstract workflow. The literature refers to the data and operations in a narrative style rather than in a form of a standardised schema or ontology. This is a well-known problem in particular in the field of life sciences. For example, Thessen and Patterson [37] stated that there are no "comprehensive" standards, no intention to share data, and there is no common data infrastructure for the life sciences.

2. The writing structures are different between sentences; extracting the ***operation*** and ***data*** from plain text using a rule-based approach was challenging and time-consuming. An ***operation*** could be a subject, verb or object. It could be combined with ***data*** in the same noun phrase. It also could come in the context of the purpose of introducing a task or using the software. ***Data*** could be any noun that is required or produced by a process. Collecting all the nouns that could be

data and trying to normalise them to well-known resources (e.g. Data EDAM - Topic EDAM - Europe PMC annotation) did not produce an effective result in the ***data*** annotation. Although we overcame this challenge and obtained a reasonable result in annotation using transformer technology, some of the annotated data failed to find corresponding concepts that represented them in knowledge resources like EDAM.

3. The processes of manual annotation and the quality evaluation of each single step are subject to human frailty and subjectivity. These processes need a considerable amount of time and multiple annotators and evaluators to obtain a good judgement of the results obtained.

4. We did our best to benefit from the previously implemented systems and to employ them to perform the required tasks. However, this process was challenging, starting from searching the available and working systems, testing them, and integrating them with our system.

7.3 Limitations

The work presented in this thesis has the following limitations:

1. Abstract workflows are extracted only from a subset of the open-access PMC - that is, the field of microarray studies. Drawing a broader conclusion from our results needs testing in other fields and on other papers.

2. The accuracy of entity annotation, relation extraction and the integrated systems affected the overall system and might cast some doubt on some of the conclusions drawn.

3. ***Data*** detection and normalisation require further work to improve the accuracy of the extraction and mapping of ***data***. Metadata can be added as well to enhance the knowledge of the extracted ***data***.

4. We did not handle the coreferring expressions neither at data event nor at workflow level. Addressing the coreference for data will help in tracking all the processes happened to a specific data, in case it is referred by another expression, all over the workflow. It will also provide a better understanding of produced workflow.

5. There is no difference between operations as being a task or a method. Adding a layer of annotation or metadata will enable extracting the methods' names and analysing them.

6. The writing style patterns and abstract workflows are extracted to represent sequential sentences in the Methods section without considering to divide or organise them based on the reported subsections in the Methods section. No further temporal information processing has been performed.

7.4 Future work

1. **Improving data extraction, modelling and classification.**

   Data can be an item that has a value and, potentially, can be an input to or an output from a database, a software process or an algorithm calculation. It is important to improve the level of the data annotation, classification and metadata. For example, increasing the level of granularity of extracted data would increase the accuracy of the annotated data and thus enhance the normalisation precision.

   Data instances can be named, unnamed data or metadata. *Named data* can be reported in the text as a group or individual instances. *Group named data* refers to a dataset in form of a file format, supplementary file or DOI. *Individual named data* refers to individual instances such as accession number (reference to a database), term name, term short name or sequence pattern. *Unnamed data* is used for data that are mentioned as a general concept (e.g. protein sequences, transcripts). *Metadata* is either for data about data or for software parameters. Adding a layer of metadata could help in constructing a standardisation environment that organises data usage and exchange. An example of metadata is classifying the instances where data are mentioned in the text into a taxonomy of instances types. The possible instances of data in microarray analysis literature are presented in Figure 7.3.

   Another kind of metadata is to classify the annotated instances based on the general data type defined in the computational community (e.g. string, integer or records). The OntoDT ontology [184] can be used to define this kind of data type. It is a general datatype ontology that adopts the ISO/IEC 11404 standard for representing the datatypes. The definition of a datatype in computer science comprises possible types of data, their potential values and how they are stored and operated [185]. The OntoDT datatype class has three main features: value, operations, and datatype property (Figure 7.4). The latter defines whether the datatype is numeric, ordered, bounded, etc. For example, positive integer data type is numeric, ordered and the minimum value is bounded by zero. The datatype taxonomy is divided into three main datatypes: primitive, generated and user-defined (Figure 7.5). A primitive datatype includes twelve subclasses

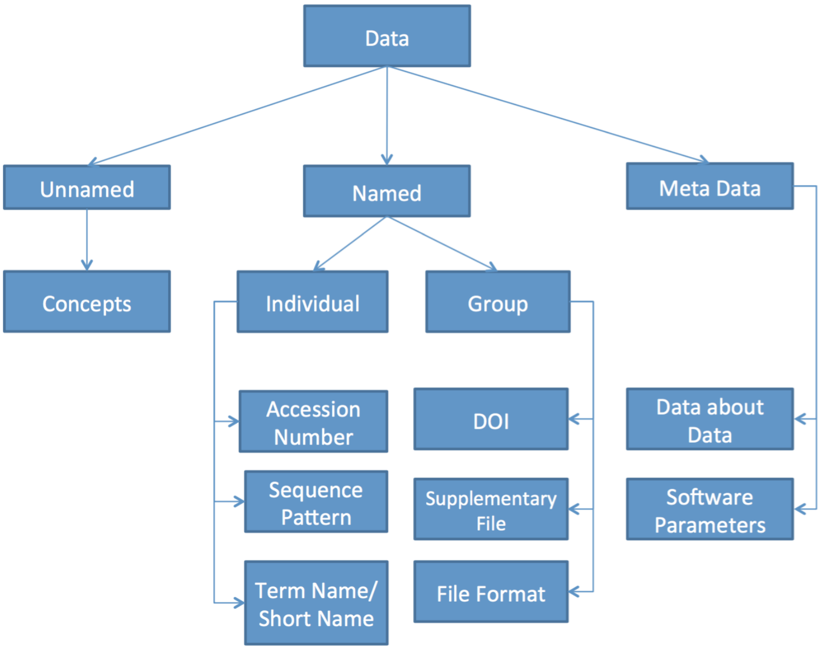


Figure 7.3. Data instance mentions in the microarray analysis literature

(real, scaled, integer, complex, rational, date-and-time, enumerated, ordinal, character, discrete, boolean and void datatypes). A generated datatype is a directive information entity that defines the conceptual operations that result in a datatype. There are nine subclasses: table, class, tuple, array, sequence, bag, set, pointer, choice and procedure datatypes. A user-defined datatype depends on what a user wants to define, for example, a tree or a directed cyclic graph.

2. **Moving from an abstract workflow to an executable workflow.**

   This will need an assessment of the available executable workflow environment like Galaxy [27], Kepler [28] and Taverna [29] and build an exchange medium to convert from the annotated events to the format the workflow environments can accept.

   Building an environment that easily explains/replicates someone's work would significantly accelerate the process of applying other people's methods with a different range of data and comparing the results obtained by the old one.

   Prior to that, there is a need to enhance the current state of the extracted workflows. This includes improving the relation extraction accuracy, the sequences of data events and showing clearly the dependency between them and grouping the related events that are doing one task or referring to commonly used methods within the literature.

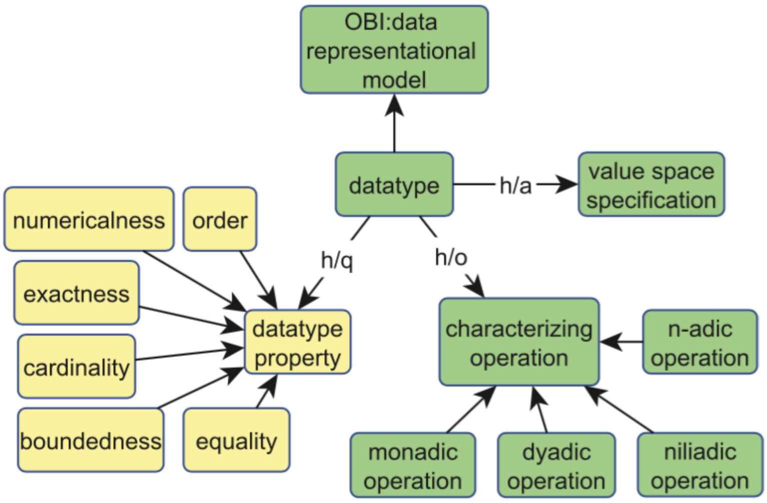


Figure 7.4. The datatype schema definition is a subclass of data representation model defined in Ontology for Biomedical Investigations (OBI) and has operation (h_o) defined by operations, has quality (h_q) of some properties and has attributes (h_a) of values. Image is taken from Panov et al. [184].

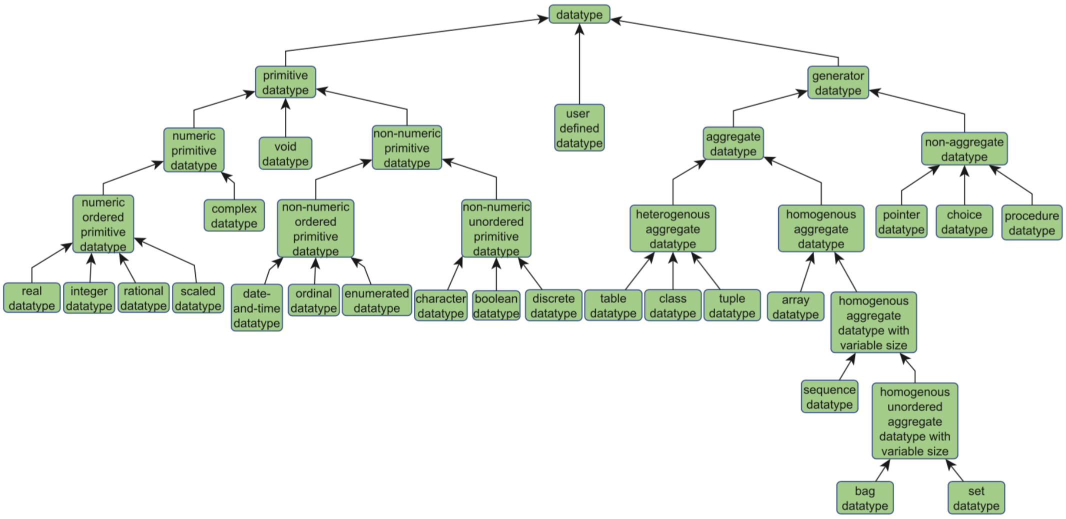


Figure 7.5. OntoDT taxonomy, which includes three main datatypes: primitive, generated and user-defined datatypes. Each type has a number of subtypes. The total number of defined datatypes is 25. Image is taken from Panov et al. [184].

The classification of input data and output data achieved a modest accuracy in our system. It is good to know, with a high level of accuracy, what is the exact input of an operation, and what output data the software can produce. If we can infer this kind of information, we could generalise this relation over the literature and use it to build small, executable workflows.

It is also important to find the co-reference between data and link them to build a relation over the document, track the changes done over the data and whether it is an output from one process and used as input for another.

The flow of the dependent events could be enhanced by applying temporal extraction techniques and changing the subject discourse to differentiate between dependent and independent events.

Grouping related data events into a sub-workflow could be done at different levels. One of them is producing an abstract workflow for each sub-section or each paragraph and then grouping these sub-workflows into appropriate parallel or sequential sequences. Another level is pre-defining sub-workflows for the common data events that could be generalised over the literature, for example, the one related to the statistical analysis. The part of the text that explains that group of data events will be mapped to one of the pre-defined sub-workflows.

3. **Applying our system in a different domain.**

   It is of interest to know if common data events are the same or different in other life science domains such as specific medical or biological studies. It is also interesting to see what kind of data events would be extracted from different data science domains.

4. **Studying the context of the data events**

   Relations between data events and their components can be analysed in depth. For example, which events must precede/follow an event. Can we perform statistical analysis before specifying or correcting the P value? How is this reported in the literature? Can we use some existing knowledge for that? Input data and output data for an operation and software can form pre-defined data events and help in “guessing” the similar data events components in a context where one component is missed or not reported.

   We found that some resources can have a role of being software or database. For example, Microsoft Excel could be a tool if used to calculate computational processes, or a database if used as a storing/retrieving environment for data. Studying the context of operations and data in data events contexts will help in classifying the role of the resources used.

5. **Improving the discourse analysis**

   We found that the currently available discourse analysis systems have a low F-score over the microarray analysis domain. It would be important to improve the existing tools, train them on other fields and relations that are related to data events.

   The available discourse analysis systems could be also combined to provide multi-discourse features for a data event. Some of these features will help in organising abstract workflows into parallel sub-workflows like the one find `change in topic` but is still a `Method` used as a part of the authors' `Own` experiment. The relation `Change in topic` is from RST, function `Method` from functional discourse and function `Own` from argumentative zoning.

6. **Classifying data events types**

   We introduced two data events type: a general data event and an implicit data event. More fine-grained classification of data event types could help in classifying which events will change data and which events will not. For example, retrieving, clustering or classifying data will not change the data, as it will only filter them based on conditions and/or could add metadata to the new subset to differentiate it from the source group. Representing the data will not change the data, but it will the change the shape of data. For example, computational data can be viewed as line, curve or even a graph. What should we call this type of data event based on the action done on data?

   The type of data events that change the data can be classified as well. For example, there are general purpose data events and field-specific data events. The top classes in the Operation EDAM ontology could be a good starting point to define and classify the data events types.

7.5 Summary of thesis contributions

This thesis provides the following research contributions:

1. Extracts and analyses the common writing patterns in terms of functional discourse of the Methods sections (Chapter 3).

2. Provides the annotation guidelines for labelling the data and operations mentioned in scientific text, and manually annotates the Methods section of 25 documents that contain 740 sentences, 621 operations and 1,041 data with averages of 25, 42 per document and one or two per sentence, respectively (Chapter 4). This is a first corpus with this type of information annotated.

3. Develops and evaluates the ODNoR system (Chapter 4) that annotates operations and data reported in conducting the work reported in research papers. It also finds associations between data, operations, database and software mentions. It also links the extracted software to bio.tools, operations and data to a well-known resource (EDAM). There are two operation classes added to EDAM's Operation ontology. The first class is `Assign Role` that assigns data a role in the context, while the second class is `Assignment` that assigns a value to data.

4. Develops the ODNoRFlow system (Chapter 5) that reconstructs an abstract workflow of the processes reported with associated data. It uses the output of ODNoR system to construct data events that represent the input data of an operation and the output data, with any involved software or databases. The analysis in (Chapter 5) finds out that using one type of discourse analysis does not improve the result of data events accuracy. The current discourse annotation tools' accuracy in the field of microarray analysis affects the result obtained. The discourse theoretical concepts hold a noteworthy interpretation of what is going on. There is a need to improve the implemented software and adapt them to the computational biology field.

5. A survey of the full PubMed Central literature for data event components usage. It provides a corpus of abstract workflows of 16,604 methods section. It analyses the temporal changes in usage patterns among the data event components and the significant relations between them (Chapter 6).

7.6 Final conclusions

The work presented in this thesis investigated various approaches to extract abstract workflows of the work reported in the computational biology literature. In particular, we used data events to represent the main parts of an event performed on data. The main parts involved operations, data, software and databases. We normalised these components to well-known knowledge resources. Furthermore, we extracted their usage of them over 20 years. Overall, the thesis provides a new computational framework that contributes to the automated extraction, representation and analysis of methods used in the computational biology literature.

References

[1] Peder Olesen Larsen and Markus Von Ins. “The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index”. In: *Scientometrics* 84.3 (2010), pp. 575–603.

[2] Donald C Comeau, Rezarta Islamaj Doğan, Paolo Ciccarese, Kevin Bretonnel Cohen, Martin Krallinger, Florian Leitner, Zhiyong Lu, Yifan Peng, Fabio Rinaldi, Manabu Torii, et al. “BioC: a minimalist approach to interoperability for biomedical text processing”. In: *Database* 2013 (2013).

[3] Donald C Comeau, Chih-Hsuan Wei, Rezarta Islamaj Doğan, and Zhiyong Lu. “PMC text mining subset in BioC: about three million full-text articles and growing”. In: *Bioinformatics* 35.18 (2019), pp. 3533–3535.

[4] Yifan Peng, Catalina O Tudor, Manabu Torii, Cathy H Wu, and K Vijay-Shanker. “iSimp in BioC standard format: enhancing the interoperability of a sentence simplification system”. In: *Database* 2014 (2014).

[5] Jiao Li, Yueping Sun, R Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. “Annotating chemicals, diseases, and their interactions in biomedical literature”. In: *Proceedings of the fifth BioCreative challenge evaluation workshop*. The Fifth BioCreative Organizing Committee. 2015, pp. 173–182.

[6] Timothy M Errington, Alexandria Denis, Nicole Perfito, Elizabeth Iorns, and Brian A Nosek. “Reproducibility in Cancer Biology: Challenges for assessing replicability in preclinical cancer biology”. In: *Elife* 10 (2021).

[7] Richard H Kallet. “How to write the methods section of a research paper”. In: *Respiratory care* 49.10 (2004), pp. 1229–1232.

[8] PLOS. *Submission Guidelines*. [Online; accessed 14-July-2016]. 2016. URL: `http://journals.plos.org/ploscompbiol/s/submission-guidelines`.

[9] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific data* 3.1 (2016), pp. 1–9.

[10] Mark D Wilkinson, Michel Dumontier, Ijsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan Willem Boiten, Luiz Bonino Da Silva Santos, Philip E Bourne, et al. “Addendum: The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific data* 6 (2019), p. 6.

[11] Geraint Duck, Goran Nenadic, Michele Filannino, Andy Brass, David L Robertson, and Robert Stevens. “A Survey of Bioinformatics Database and Software Usage through Mining the Literature”. In: *PLOS ONE* 11.6 (2016).

[12] Daniel R Masys. “Linking microarray data to the literature”. In: *Nature Genetics* 28.1 (2001), pp. 9–10.

[13] Zena M Hira and Duncan F Gillies. “A review of feature selection and feature extraction methods applied on microarray data”. In: *Advances in bioinformatics* 2015 (2015).

[14] Hagit Shatkay and Ronen Feldman. “Mining the biomedical literature in the genomic era: an overview”. In: *Journal of computational biology* 10.6 (2003), pp. 821–855.

[15] James M Eales, John W Pinney, Robert D Stevens, and David L Robertson. “Methodology capture: discriminating between the ”best” and the rest of community practice”. In: *BMC bioinformatics* 9.1 (2008).

[16] He Zhao, Zhunchen Luo, Chong Feng, Anqing Zheng, and Xiaopeng Liu. “A context-based framework for modeling the role and function of on-line resource citations in scientific literature”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 5206–5215.

[17] Dictionary by Merriam-Webster. *Methodology*. [Online; accessed 15-August-2016]. URL: http://www.merriam-webster.com/dictionary/methodology.

[18] Dictionary by Merriam-Webster. *Methods*. [Online; accessed 15-August-2016]. 2015. URL: http://www.merriam-webster.com/dictionary/methods.

[19] John PA Ioannidis, David B Allison, Catherine A Ball, Issa Coulibaly, Xiangqin Cui, Aedín C Culhane, Mario Falchi, Cesare Furlanello, Laurence Game, Giuseppe Jurman, et al. "Repeatability of published microarray gene expression analyses". In: *Nature genetics* 41.2 (2009), pp. 149–155.

[20] Alvis Brazma, Pascal Hingamp, John Quackenbush, Gavin Sherlock, Paul Spellman, Chris Stoeckert, John Aach, Wilhelm Ansorge, Catherine A Ball, Helen C Causton, et al. "Minimum information about a microarray experiment (MIAME)—toward standards for microarray data". In: *Nature genetics* 29.4 (2001), pp. 365–371.

[21] Tim F Rayner, Philippe Rocca-Serra, Paul T Spellman, Helen C Causton, Anna Farne, Ele Holloway, Rafael A Irizarry, Junmin Liu, Donald S Maier, Michael Miller, et al. "A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB". In: *Bmc Bioinformatics* 7.1 (2006).

[22] Michael Huerta, Gregory Downing, Florence Haseltine, Belinda Seto, and Yuan Liu. "NIH working definition of bioinformatics and computational biology". In: *US National Institute of Health* (2000).

[23] Ung-Sik Yu, Sung-Hoon Lee, Young-Joo Kim, and Sang-Soo Kim. "Bioinformatics in the post-genome era". In: *BMB Reports* 37.1 (2004), pp. 75–82.

[24] Geraint Duck. "Extraction Of Database And Software Usage Patterns From The Bioinformatics Literature". PhD thesis. University Of Manchester, School of Computer Science, 2015.

[25] Dictionary by Merriam-Webster. *Workflow*. [Online; accessed 12-August-2016]. URL: http://www.webster-dictionary.org/definition/workflow.

[26] Wikipedia. *Scientific Workflow System*. [Online; accessed 12-August-2016]. URL: https://en.wikipedia.org/wiki/Scientific_workflow_system.

[27] Jeremy Goecks, Anton Nekrutenko, and James Taylor. “Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences”. In: *Genome biology* 11.8 (2010).

[28] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A Lee, Jing Tao, and Yang Zhao. “Scientific workflow management and the Kepler system”. In: *Concurrency and Computation: Practice and Experience* 18.10 (2006), pp. 1039–1065.

[29] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R Pocock, Anil Wipat, et al. “Taverna: a tool for the composition and enactment of bioinformatics workflows”. In: *Bioinformatics* 20.17 (2004), pp. 3045–3054.

[30] Dictionary by Merriam-Webster. *Data*. [Online; accessed 11-June-2016]. URL: `http://www.merriam-webster.com/dictionary/data`.

[31] Wikipedia. *Data*. [Online; accessed 15-August-2016]. URL: `https://en.wikipedia.org/wiki/Data`.

[32] Ivo D Dinov, Daniel Rubin, William Lorensen, Jonathan Dugan, Jeff Ma, Shawn Murphy, Beth Kirschner, William Bug, Michael Sherman, Aris Floratos, et al. “iTools: a framework for classification, categorization and integration of computational biology resources”. In: *PLoS One* 3.5 (2008).

[33] John M Hancock and Marketa J Zvelebil. *Concise Encyclopaedia of Bioinformatics and Computational Biology*. John Wiley & Sons, 2014.

[34] Jon Ison, Matúš Kalaš, Inge Jonassen, Dan Bolser, Mahmut Uludag, Hamish McWilliam, James Malone, Rodrigo Lopez, Steve Pettifer, and Peter Rice. “EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats”. In: *Bioinformatics* 29.10 (2013), pp. 1325–1332.

[35] Şenay Kafkas, Jee-Hyub Kim, Xingjun Pi, and Johanna R McEntyre. “Database citation in supplementary data linked to Europe PubMed Central full text biomedical articles”. In: *Journal of biomedical semantics* 6.1 (2015).

[36] Dryad. *Dryad Digital Repository Dryad*. [Online; accessed 29-June-2016]. 2016. URL: `http://datadryad.org/`.

[37] Anne E Thessen and David J Patterson. “Data issues in the life sciences”. In: *ZooKeys* 150 (2011).

[38] Natalya F Noy, Deborah L McGuinness, et al. *Ontology development 101: A guide to creating your first ontology*. 2001.

[39] James Malone, Andy Brown, Allyson L Lister, Jon Ison, Duncan Hull, Helen Parkinson, and Robert Stevens. “The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation”. In: *Journal of biomedical semantics* 5.1 (2014).

[40] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. “Gene Ontology: tool for the unification of biology”. In: *Nature genetics* 25.1 (2000), pp. 25–29.

[41] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[43] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[44] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. “A comprehensive guide to bayesian convolutional neural network with variational inference”. In: *arXiv preprint arXiv:1901.02731* (2019).

[45] Savelie Cornegruta, Robert Bakewell, Samuel Withey, and Giovanni Montana. “Modelling radiological language with bidirectional long short-term memory networks”. In: *arXiv preprint arXiv:1609.08409* (2016).

[46] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. “Tensor2Tensor for Neural Machine Translation”. In: *CoRR* abs/1803.07416 (2018). URL: `http://arxiv.org/abs/1803.07416`.

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Rai-

son, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[50] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. “BioBERT: pre-trained biomedical language representation model for biomedical text mining”. In: *arXiv preprint arXiv:1901.08746* (2019).

[51] Iz Beltagy, Kyle Lo, and Arman Cohan. “Scibert: A pretrained language model for scientific text”. In: *arXiv preprint arXiv:1903.10676* (2019).

[52] John McNaught. *Text Mining: Background [Powerpoint slides]*. [Online; accessed 24-June-2016]. 2016. URL: `http://personalpages.manchester.ac.uk/staff/john.mcnaught/intranet/COMP61332/node3.html`.

[53] Marti A Hearst. “Untangling text data mining”. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics. 1999, pp. 3–10.

[54] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. Vol. 999. MIT Press, 1999.

[55] Sophia Ananiadou and John McNaught. *Text mining for biology and biomedicine*. Citeseer, 2006.

[56] David Bousfield, Johanna McEntyre, Sameer Velankar, George Papadatos, Alex Bateman, Guy Cochrane, Jee-Hyub Kim, Florian Graef, Vid Vartak, Blaise Alako, et al. "Patterns of database citation in articles and patents indicate long-term scientific and industry value of biological data resources". In: *F1000Research* 5 (2016).

[57] Michael Krauthammer and Goran Nenadic. "Term identification in the biomedical literature". In: *Journal of biomedical informatics* 37.6 (2004), pp. 512–526.

[58] Wei Shen, Jianyong Wang, and Jiawei Han. "Entity linking with a knowledge base: Issues, techniques, and solutions". In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2014), pp. 443–460.

[59] Nguyen Bach and Sameer Badaskar. "A review of relation extraction". In: *Literature review for Language and Statistics II* 2 (2007), pp. 1–15.

[60] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals". In: *arXiv preprint arXiv:1911.10422* (2019).

[61] Dongxu Zhang and Dong Wang. "Relation classification via recurrent neural network". In: *arXiv preprint arXiv:1508.01006* (2015).

[62] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. "Position-aware attention and supervised data improve slot filling". In: *Conference on Empirical Methods in Natural Language Processing*. 2017.

[63] Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. "More data, more relations, more context and more openness: A review and outlook for relation extraction". In: *arXiv preprint arXiv:2004.03186* (2020).

[64] Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. "BioRED: A Comprehensive Biomedical Relation Extraction Dataset". In: *arXiv preprint arXiv:2204.04263* (2022).

[65] Kuicai Dong, Yilin Zhao, Aixin Sun, Jung-Jae Kim, and Xiaoli Li. “DocOIE: A Document-level Context-Aware Dataset for OpenIE”. In: *arXiv preprint arXiv:2105.04271* (2021).

[66] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. “DocRED: A large-scale document-level relation extraction dataset”. In: *arXiv preprint arXiv:1906.06127* (2019).

[67] Martin Gerner, Goran Nenadic, and Casey M Bergman. “LINNAEUS: a species name identification system for biomedical literature”. In: *BMC bioinformatics* 11.1 (2010).

[68] Jitendra Jonnagaddala, Toni Rose Jue, Nai-Wen Chang, and Hong-Jie Dai. “Improving the dictionary lookup approach for disease normalization using enhanced dictionary and query expansion”. In: *Database* 2016 (2016).

[69] Aaron Cohen. “Unsupervised gene/protein named entity normalization using automatically extracted dictionaries”. In: *Proceedings of the acl-ismb workshop on linking biological literature, ontologies and databases: Mining biological semantics*. 2005, pp. 17–24.

[70] Deyu Zhou, Dayou Zhong, and Yulan He. “Biomedical relation extraction: from binary to complex”. In: *Computational and mathematical methods in medicine* 2014 (2014).

[71] Geraint Duck, Goran Nenadic, Andy Brass, David L Robertson, and Robert Stevens. “bioNerDS: exploring bioinformatics’ database and software use through literature mining”. In: *BMC bioinformatics* 14.1 (2013).

[72] Dietrich Rebholz-Schuhmann, Miguel Arregui, Sylvain Gaudan, Harald Kirsch, and Antonio Jimeno. “Text processing through Web services: calling Whatizit”. In: *Bioinformatics* 24.2 (2008), pp. 296–298.

[73] Şenay Kafkas, Jee-Hyub Kim, and Johanna R McEntyre. “Database citation in full text biomedical articles”. In: *PloS one* 8.5 (2013).

[74] Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang. “CNN-based ranking for biomedical entity normalization”. In: *BMC bioinformatics* 18.11 (2017), pp. 79–86.

[75] Jennifer D'Souza and Vincent Ng. "Sieve-based entity linking for the biomedical domain". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2015, pp. 297–302.

[76] Lorraine Goeuriot, G Jones, L Kelly, J Leveling, A Hanbury, H Müller, S Salanterä, H Suominen, and G Zuccon. "ShARe/CLEF eHealth Evaluation Lab 2014". In: *Task 3: Information Retrieval to Address Patients' Questions when Reading Clinical Reports. Proc. of CLEF 2013*. 2013.

[77] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. "NCBI disease corpus: a resource for disease name recognition and concept normalization". In: *Journal of biomedical informatics* 47 (2014), pp. 1–10.

[78] Keith E Campbell, Diane E Oliver, and Edward H Shortliffe. "The Unified Medical Language System: toward a collaborative approach for solving terminologic problems". In: *Journal of the American Medical Informatics Association* 5.1 (1998), pp. 12–16.

[79] Allan Peter Davis, Thomas C Wiegers, Michael C Rosenstein, and Carolyn J Mattingly. "MEDIC: a practical disease vocabulary used at the Comparative Toxicogenomics Database". In: *Database* 2012 (2012).

[80] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. "PATTY: A taxonomy of relational patterns with semantic types". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 2012, pp. 1135–1145.

[81] Katrin Fundel, Robert Küffner, and Ralf Zimmer. "RelEx-Relation extraction using dependency parse trees". In: *Bioinformatics* 23.3 (2007), pp. 365–371.

[82] Erik F Sang and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition". In: *arXiv preprint cs/0306050* (2003).

[83] Wenzheng Zhang, Wenyue Hua, and Karl Stratos. "EntQA: Entity Linking as Question Answering". In: *arXiv preprint arXiv:2110.02369* (2021).

[84] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. "Gerbil–benchmarking named entity recognition and linking consistently". In: *Semantic Web* 9.5 (2018), pp. 605–625.

[85] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. "Robust disambiguation of named entities in text". In: *Proceedings of the 2011 conference on empirical methods in natural language processing*. 2011, pp. 782–792.

[86] Zongcheng Ji, Qiang Wei, and Hua Xu. "Bert-based ranking for biomedical entity normalization". In: *AMIA Summits on Translational Science Proceedings* 2020 (2020), p. 269.

[87] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. "Clinicalbert: Modeling clinical notes and predicting hospital readmission". In: *arXiv preprint arXiv:1904.05342* (2019).

[88] Kirk Roberts, Dina Demner-Fushman, and Joseph M Tonning. "Overview of the TAC 2017 Adverse Reaction Extraction from Drug Labels Track." In: *TAC*. 2017.

[89] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 207–212.

[90] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. "Matching the blanks: Distributional similarity for relation learning". In: *arXiv preprint arXiv:1906.03158* (2019).

[91] Amir DN Cohen, Shachar Rosenman, and Yoav Goldberg. "Relation Classification as Two-way Span-Prediction". In: *arXiv preprint arXiv:2010.04829* (2020).

[92] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Cristian Ursu, Marin Dimitrov, Mike Dowman, Niraj Aswani, Ian Roberts, Yaoyong Li, et al. *Developing Language Processing Components with GATE Version 5:(a User Guide)*. Citeseer, 2009.

[93] Burr Settles. "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text". In: *Bioinformatics* 21.14 (2005), pp. 3191–3192.

[94] Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. "Developing a robust part-of-speech tagger for biomedical text". In: *Panhellenic Conference on Informatics*. Springer. 2005, pp. 382–392.

[95] About Education. *Discourse: Definitions and Examples*. [Online; accessed 27-August-2016]. 2016. URL: `http://grammar.about.com/od/d/g/discourseterm.htm`.

[96] Meyer Howard Abrams and Geoffrey Harpham. *A glossary of literary terms*. Cengage Learning, 2011.

[97] Bonnie Webber, Markus Egg, and Valia Kordoni. "Discourse structure and language technology". In: *Natural Language Engineering* 18.04 (2012), pp. 437–490.

[98] Jill Burstein, Daniel Marcu, and Kevin Knight. "Finding the WRITE stuff: Automatic identification of discourse structure in student essays". In: *IEEE Intelligent Systems* 18.1 (2003), pp. 32–39.

[99] Yoko Mizuta, Anna Korhonen, Tony Mullen, and Nigel Collier. "Zone analysis in biology articles as a basis for information extraction". In: *International journal of medical informatics* 75.6 (2006), pp. 468–487.

[100] Maria Liakata and Larisa Soldatova. "The art corpus". In: *Technical report, Aberystwyth University* (2015).

[101] Nathanael Chambers and Daniel Jurafsky. "Unsupervised Learning of Narrative Event Chains." In: *ACL*. Vol. 94305. Citeseer. 2008, pp. 789–797.

[102] Nathanael Chambers and Dan Jurafsky. "Template-based information extraction without the templates". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics. 2011, pp. 976–986.

[103] Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. "ACE 2005 multilingual training corpus". In: *Linguistic Data Consortium, Philadelphia* (2006).

[104] Roser Saurı, Lotus Goldberg, Marc Verhagen, and James Pustejovsky. *Annotating Events in English. TimeML Annotation Guidelines*. 2009.

[105] Raheel Nawaz, Paul Thompson, and Sophia Ananiadou. "Meta-knowledge annotation at the event level: comparison between abstracts and full papers". In: *Meta* (2012).

[106] William C. Mann and Maite Taboada. *Intro to RST, Rhetorical Structure Theory*. [Online; accessed 29-August-2016]. URL: `http://www.sfu.ca/rst/01intro/intro.html`.

[107] University of Pennsylvania. *The Penn Discourse Treebank*. [Online; accessed 29-July-2016]. URL: `https://www.seas.upenn.edu/~pdtb/tools.shtml`.

[108] William C Mann and Sandra A Thompson. "Rhetorical structure theory: Toward a functional theory of text organization". In: *Text-Interdisciplinary Journal for the Study of Discourse* 8.3 (1988), pp. 243–281.

[109] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. "The Penn Discourse TreeBank 2.0." In: *LREC*. Citeseer. 2008.

[110] Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. "The biomedical discourse relation bank". In: *BMC bioinformatics* 12.1 (2011).

[111] CJ van Rijsbergen. *Information Retrieval, 2nd edButterworths*. 1979.

[112] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[113] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. "BRAT: a web-based tool for NLP-assisted text annotation". In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2012, pp. 102–107.

[114] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. *Gephi: An Open Source Software for Exploring and Manipulating Networks*. 2009.

[115] Geraint Duck, Goran Nenadic, Andy Brass, David L Robertson, and Robert Stevens. “Extracting patterns of database and software usage from the bioinformatics literature”. In: *Bioinformatics* 30.17 (2014), pp. 601–608.

[116] Aleksandar Kovačević, Zora Konjović, Branko Milosavljević, and Goran Nenadic. “Mining methodologies from NLP publications: A case study in automatic terminology recognition”. In: *Computer Speech & Language* 26.2 (2012), pp. 105–126.

[117] Thomas Kappeler, Simon Clematide, Kaarel Kaljurand, Gerold Schneider, and Fabio Rinaldi. “Towards automatic detection of experimental methods from biomedical literature”. In: *Third International Symposium on Semantic Mining in Biomedicine (SMBM)*. Citeseer. 2008.

[118] Sarthak Jain, Madeleine Van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. “SciREX: A challenge dataset for document-level information extraction”. In: *arXiv preprint arXiv:2005.00512* (2020).

[119] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. “Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications”. In: *arXiv preprint arXiv:1704.02853* (2017).

[120] Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. “Scientific information extraction with semi-supervised neural tagging”. In: *arXiv preprint arXiv:1708.06075* (2017).

[121] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. “Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction”. In: *arXiv preprint arXiv:1808.09602* (2018).

[122] Alan G Gross, Joseph E Harmon, Michael Reidy, and Michael S Reidy. *Communicating science: The scientific article from the 17th century to the present*. Oxford: Oxford University Press, 2002.

[123] Hilda Bastian, Paul Glasziou, and Iain Chalmers. “Seventy-five trials and eleven systematic reviews a day: how will we ever keep up?” In: *PLoS medicine* 7.9 (2010).

[124] Sally Hopewell, Susan Dutton, Ly-Mee Yu, An-Wen Chan, and Douglas G Altman. “The quality of reports of randomised trials in 2000 and 2006: comparative study of articles indexed in PubMed”. In: *Bmj* 340 (2010).

[125] Joshua Carp. “The secret lives of experiments: methods reporting in the fMRI literature”. In: *Neuroimage* 63.1 (2012), pp. 289–300.

[126] Paul Glasziou, Douglas G Altman, Patrick Bossuyt, Isabelle Boutron, Mike Clarke, Steven Julious, Susan Michie, David Moher, and Elizabeth Wager. “Reducing waste from incomplete or unusable reports of biomedical research”. In: *The Lancet* 383.9913 (2014), pp. 267–276.

[127] Geraint Duck, Aleksandar Kovacevic, David L Robertson, Robert Stevens, and Goran Nenadic. “Ambiguity and variability of database and software names in bioinformatics”. In: *Journal of biomedical semantics* 6.1 (2015).

[128] Michael Stubbs. *Discourse Analysis: The Sociolinguistic Analysis of Natural Language*. Chicago: University of Chicago Press, 1983.

[129] Simone Teufel, Jean Carletta, and Marc Moens. “An annotation scheme for discourse-level argumentation in research articles”. In: *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 1999, pp. 110–117.

[130] Yufan Guo, Anna Korhonen, and Thierry Poibeau. “A weakly-supervised approach to argumentative zoning of scientific documents”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2011, pp. 273–283.

[131] Douwe Kiela, Yufan Guo, Ulla Stenius, and Anna Korhonen. “Unsupervised discovery of information structure in biomedical documents”. In: *Bioinformatics* 31.7 (2015), pp. 1084–1092.

[132] Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. “Automatic recognition of conceptualization zones in scientific articles and two life science applications”. In: *Bioinformatics* 28.7 (2012), pp. 991–1000.

[133] Maria Liakata. “Guidelines for the Multi-Annotation of Core Scientific Concepts (CoreSC)”. In: *Technical report, Aberystwyth University* (2011).

[134] Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin Batchelor. “Corpora for the Conceptualisation and Zoning of Scientific Papers.” In: *LREC*. Citeseer. 2010.

[135] Richard J Roberts. *PubMed Central: The GenBank of the published literature*. 2001.

[136] *SOAP web service*. [Online; accessed 18-April-2017]. URL: `https://europepmc.org/SoapWebServices`.

[137] *Attribute : Type of Article*. [Online; accessed 18-April-2017]. URL: `https://dtd.nlm.nih.gov/publishing/tag-library/2.1/n-g270.html`.

[138] Yufan Guo, Anna Korhonen, Maria Liakata, Ilona Silins, Johan Hogberg, and Ulla Stenius. “A comparison and user-based evaluation of models of textual information structure in the context of cancer risk assessment”. In: *BMC bioinformatics* 12.1 (2011).

[139] Şenay Kafkas, Xingjun Pi, Nikos Marinos, Andrew Morrison, and Johanna R McEntyre. “Section level search functionality in Europe PMC”. In: *Journal of biomedical semantics* 6.1 (2015).

[140] Philippe Fournier-Viger and Vincent S Tseng. “Mining top-k sequential rules”. In: *International Conference on Advanced Data Mining and Applications*. Springer. 2011, pp. 180–194.

[141] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Zhihong Deng, and Hoang Thanh Lam. “The SPMF open-source data mining library version 2”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2016, pp. 36–40.

[142] Lawrence R Rabiner and Biing-Hwang Juang. “An introduction to hidden Markov models”. In: *IEEE ASSP Magazine* 3.1 (1986), pp. 4–16.

[143] Pierre Brémaud. “Discrete-Time Markov Models”. In: *Markov Chains*. New York: Springer, 1999, pp. 53–93.

[144] Giorgio Alfredo Spedicato, Tae Seung Kang, Sai Bhargav Yalamanchi, Deepak Yadav, and Ignacio Cordón. “The markovchain package: a package for easily handling Discrete Markov Chains in R”. In: *The Comprehensive R Archive*

Network (2016). URL: https://cran.r-project.org/web/packages/markovchain/vignettes/an_introduction_to_markovchain_package.pdf.

[145] Cees Elzinga, Sven Rahmann, and Hui Wang. “Algorithms for subsequence combinatorics”. In: *Theoretical Computer Science* 409.3 (2008), pp. 394–404.

[146] Alexis Gabadinho, Gilbert Ritschard, Matthias Studer, and Nicolas S Müller. “Mining sequence data in R with the TraMineR package: A user’s guide”. In: *Technical report, Department of Econometrics and Laboratory of Demography* (2009).

[147] Alexis Gabadinho, Gilbert Ritschard, Matthias Studer, and Nicolas S M$\ddot{u}$ller. “Extracting and rendering representative sequences”. In: *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*. Springer. 2009, pp. 94–106.

[148] John W Tukey. *Exploratory data analysis*. Vol. 2. Reading, Mass.: Addison-Wesley Publishing Company, 1977.

[149] Barbara Gastel and Robert A Day. *How to write and publish a scientific paper*. Santa Barbara, California: ABC-CLIO, 2016.

[150] Theuns Kotzé. “Guidelines on writing a first quantitative academic article”. In: *Pretoria: University of Pretoria* (2007).

[151] *BMC Bioinformatics | Research article*. [Online; accessed 12-March-2017]. URL: https://bmcbioinformatics.biomedcentral.com/submission-guidelines/preparing-your-manuscript/research-article.

[152] *PLOS|ONE Submission Guidelines*. [Online; accessed 12-March-2017]. URL: https://journals.plos.org/plosone/s/submission-guidelines#loc-guidelines-for-specific-study-types.

[153] Jon Ison, Kristoffer Rapacki, Hervé Ménager, Matúš Kalaš, Emil Rydza, Piotr Chmura, Christian Anthon, Niall Beard, Karel Berka, Dan Bolser, et al. “Tools and data services registry: a community effort to document bioinformatics resources”. In: *Nucleic acids research* 44.D1 (2016).

[154] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. “Neural architectures for named entity recognition”. In: *arXiv preprint arXiv:1603.01360* (2016).

[155] Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. “Itri-04-08 the sketch engine”. In: *Information Technology* 105 (2004).

[156] Hamish Cunningham. “GATE, a general architecture for text engineering”. In: *Computers and the Humanities* 36.2 (2002), pp. 223–254.

[157] Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. “NeuroNER: an easy-to-use program for named-entity recognition based on neural networks”. In: *Conference on Empirical Methods on Natural Language Processing (EMNLP)* (2017).

[158] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. “LSTM neural networks for language modeling”. In: *Thirteenth annual conference of the international speech communication association*. 2012.

[159] Matthew Horridge and Sean Bechhofer. “The owl api: A java api for owl ontologies”. In: *Semantic web* 2.1 (2011), pp. 11–21.

[160] Esko Ukkonen. “Approximate string-matching with q-grams and maximal matches”. In: *Theoretical computer science* 92.1 (1992), pp. 191–211.

[161] Thibault Debatty. *Java string similarity*. `https://github.com/tdebatty/java-string-similarity`. 2013.

[162] Claudia Leacock and Martin Chodorow. “Combining local context and WordNet similarity for word sense identification”. In: *WordNet: An electronic lexical database* 49.2 (1998), pp. 265–283.

[163] Hideki Shima. *WordNet Similarity for Java*. `https://github.com/dmeoli/WS4J`. 2013.

[164] Nancy Chinchor, Lynette Hirschman, and David D Lewis. “Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3)”. In: *Computational linguistics* 19.3 (1993), pp. 409–450.

[165] George Hripcsak and Daniel F Heitjan. “Measuring agreement in medical informatics reliability studies”. In: *Journal of biomedical informatics* 35.2 (2002), pp. 99–110.

[166] Lee R Dice. “Measures of the amount of ecologic association between species”. In: *Ecology* 26.3 (1945), pp. 297–302.

[167] John S Uebersax. “A design-independent method for measuring the reliability of psychiatric diagnosis”. In: *Journal of Psychiatric Research* 17.4 (1982), pp. 335–342.

[168] Jeffrey M Girard. *Specific agreement coefficient*. `https://github.com/jmgirard/mReliability/wiki/Specific-agreement-coefficient`. 2020.

[169] Hugo Hernault, Helmut Prendinger, David A du Verle, and Mitsuru Ishizuka. “HILDA: A discourse parser using support vector machine classification”. In: *Dialogue & Discourse* 1.3 (2010), pp. 1–33.

[170] Yangfeng Ji and Jacob Eisenstein. “Discriminative Improvements to Distributional Sentence Similarity.” In: *EMNLP*. 2013, pp. 891–896.

[171] Shafiq R Joty, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. “Combining Intra-and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis.” In: *ACL (1)*. 2013, pp. 486–496.

[172] Yangfeng Ji and Jacob Eisenstein. “Representation Learning for Text-level Discourse Parsing.” In: *ACL (1)*. 2014, pp. 13–24.

[173] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. “Building a discourse-tagged corpus in the framework of rhetorical structure theory”. In: *Current and new directions in discourse and dialogue*. Springer, 2003, pp. 85–112.

[174] Alvan R Feinstein and Domenic V Cicchetti. “High agreement but low kappa: I. The problems of two paradoxes”. In: *Journal of clinical epidemiology* 43.6 (1990), pp. 543–549.

[175] Domenic V Cicchetti and Alvan R Feinstein. “High agreement but low kappa: II. Resolving the paradoxes”. In: *Journal of clinical epidemiology* 43.6 (1990), pp. 551–558.

[176] Jai Prakash Mehta and Sweta Rani. “Software and tools for microarray data analysis”. In: *Gene Expression Profiling: Methods and Protocols* (2011), pp. 41–53.

[177] William S Cleveland. “Robust locally weighted regression and smoothing scatterplots”. In: *Journal of the American statistical association* 74.368 (1979), pp. 829–836.

[178] William S Cleveland and Susan J Devlin. “Locally weighted regression: an approach to regression analysis by local fitting”. In: *Journal of the American statistical association* 83.403 (1988), pp. 596–610.

[179] Hadley Wickham, Winston Chang, and Maintainer Hadley Wickham. “Package ‘ggplot2’”. In: *Create Elegant Data Visualisations Using the Grammar of Graphics. Version* 2.1 (2016), pp. 1–189.

[180] Michael Hahsler and Kurt Hornik. “New probabilistic interest measures for association rules”. In: *Intelligent Data Analysis* 11.5 (2007), pp. 437–455.

[181] Rakesh Agrawal and Ramakrishnan Srikant. “Fast algorithms for mining association rules”. In: *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. Citeseer. 1994, pp. 487–499.

[182] Michael Hahsler. “arulesViz: Interactive Visualization of Association Rules with R.” In: *R J.* 9.2 (2017).

[183] Michael Hahsler and Radoslaw Karpienko. “Visualizing association rules in hierarchical groups”. In: *Journal of Business Economics* 87.3 (2017), pp. 317–335.

[184] Pančе Panov, Larisa N Soldatova, and Sašo Džeroski. “Generic ontology of datatypes”. In: *Information Sciences* 329 (2016), pp. 900–920.

[185] Clifford A Shaffer. *A practical introduction to data structures and algorithm analysis*. Prentice Hall Upper Saddle River, NJ, 1997.

[186] Jérôme Moreaux, Jean-Luc Veyrune, John De Vos, and Bernard Klein. “APRIL is overexpressed in cancer: link with tumor progression”. In: *BMC cancer* 9.1 (2009).

Appendices

Appendix A

Data and Operation Annotation Guidelines

A.1 Data Events and Data Descriptions

A.1.1 Data events description

Event is the operation on data performed by operator(s). In our context, the data is processed by a means of operators (software, methods, tools). The manual's target is to annotate the data, the operations and the operands in case they are expressed as operational task rather than a formal method, software or a tool. BioNerDS [71] is used to annotate the database and software names mentioned in the text. Figure A.1 shows the event components at the abstract level. Table A.1 shows the data event template that is going to be filled by annotated and mapped information.

Following is a method section annotated example. The red annotation is for database, software, tools or methods. The blue colour is for operations and the yellow is for data. The method section is quoted from the article with PMCID 2662875 [186]. The

Event template			**Extracted text**	**Mapping**
Event No.				
Semantic Roles	data	input data		
		output data		
	trigger	operation		
	operator	software		
		database		
		methodology		

Table A.1. Data event template

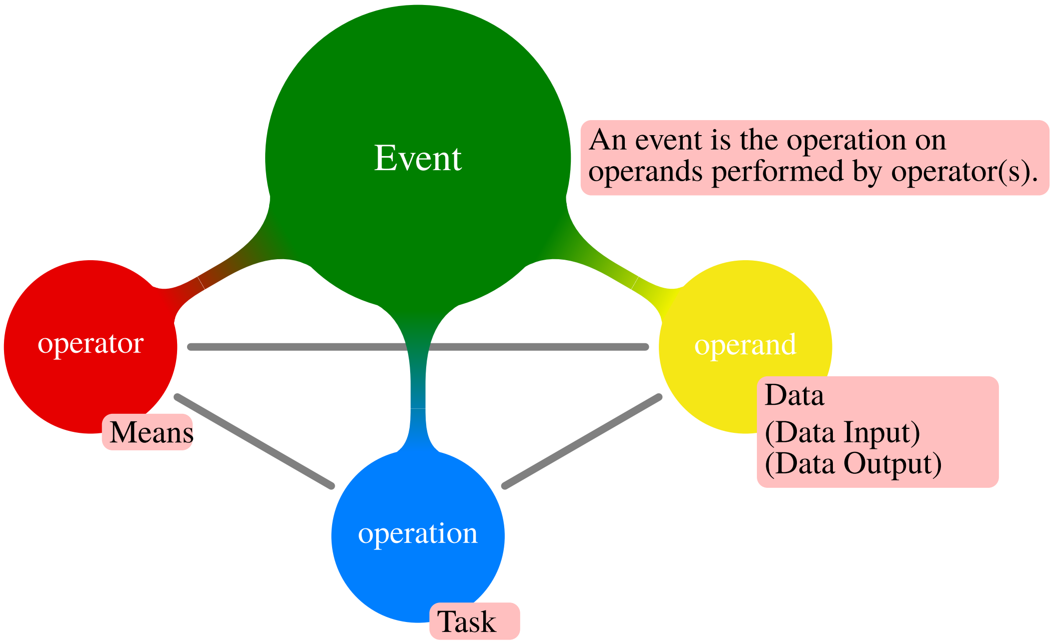


Figure A.1. Abstract level of an event

example is followed by filled templates of the information extracted.

Example 25 (PMC2662875)

Methods

Databases

We used Oncomine Cancer Microarray database http://www.oncomine.org[30] and Amazonia database http://amazonia.montp.inserm.fr/[31] to study gene expression of BAFF, APRIL, BCMA, TACI, BAFF-R and HS proteoglycans genes in 40 human tumor types and their normal tissue counterparts as indicated in Table 1 (Additional file 1). Only gene expression data obtained from a single study using the same methodology were compared. All data were log transformed, median centred per array, and the standard deviation was normalized to one per array [32].

Statistical analysis

Statistical comparisons were done with Mann-Whitney or Student t-tests.

Event Template			**Text Extraction**	**Mapping**
Event No.			1	1
Semantic Roles	data	data in	-	
		data out	gene expression data	Gene expression data
	operation	trigger	obtain	Data Retrieval
	operator	Software	-	-
		DataBase	-	-
		Method	-	-

Table A.2. Data Event 1

Event Template			**Text Extraction**	**Mapping**
Event No.			2	2
Semantic Roles	data	data in	gene expression data	Gene expression data
		data out	-	-
	operation	trigger	compare	Comparison
	operator	Software	-	-
		DataBase	-	-
		Method	-	-

Table A.3. Data Event 2

Event Template			**Text Extraction**	**Mapping**
Event No.			3	3
Semantic Roles	data	data in	data	Data
		data out	-	-
	operation	trigger	log transformed	Calculation
	operator	Software	-	-
		DataBase	-	-
		Method	-	-

Table A.4. Data Event 3

Event Template			**Text Extraction**	**Mapping**
Event No.			4	4
Semantic Roles	data	data in	median, array	Experimental measurement data, Array
		data out	-	-
	operation	trigger	centred	Normalization
	operator	Software	-	-
		DataBase	-	-
		Method	-	-

Table A.5. Data Event 4

Event Template			**Text Extraction**	**Mapping**
Event No.			5	5
Semantic Roles	data	data in	the standard deviation, one, array	Experimental measurement data, Experimental measurement data, Array
		data out	-	-
	operation	trigger	normalized	Normalization
	operator	Software	-	-
		DataBase	-	-
		Method	-	-

Table A.6. Data Event 5

Event Template			**Text Extraction**	**Mapping**
Event No.			6	6
Semantic Roles	data	data in	-	-
		data out	-	-
	operation	trigger	Statistical comparison	Statistical analysis
	operator	Software	-	-
		DataBase	-	-
		Method	Mann-Whitney or Student t-tests	

Table A.7. Data Event 6

A.1.2 Operation description

In our context, the operations are a specific/concrete computational process conducted in an experiment and presented in a manuscript as a part of the work done by a paper. The operations could have data inputs and produce outputs data. It also could have either input or output data and in some cases it could be mentioned without any data but an indication that a process is done. We also has a kind of operation that did not do any kind of processing but assigns the data a role or identification in the context.

A.1.3 Data description

Data are items (or collections of items) that exist in digital form (i.e. representation), and can be potentially used as an input to or output from DB, software that processes those data to fulfil a specific/concrete computational processes that is conducted in an experiment and presented in a manuscript.

This description includes any used data in the experiment although no explicit operations are mentioned in the sentence context. It excludes the abstract data that are used for conceptual and explanation purposes (e.g., discussion of strings, integers, etc.). It also excludes the real/physical data and any biological processes conducted on them, e.g. the temperature, unless those measurements become part of the experimental data.

A.2 How to Annotate

When you read a sentence, ask: “Does it include a specific (explicit/implicit) computational process in an analytical context?”

Then if so, ask: “Is there any mention of the input/output data for the determined process?”

if yes, then apply the operations data annotation rules (Sections A.2.1 and A.2.2) . If there is no data mention, then annotate the process according to the Section A.2.1.

Figure A.2 shows the annotation steps in a flowchart diagram.

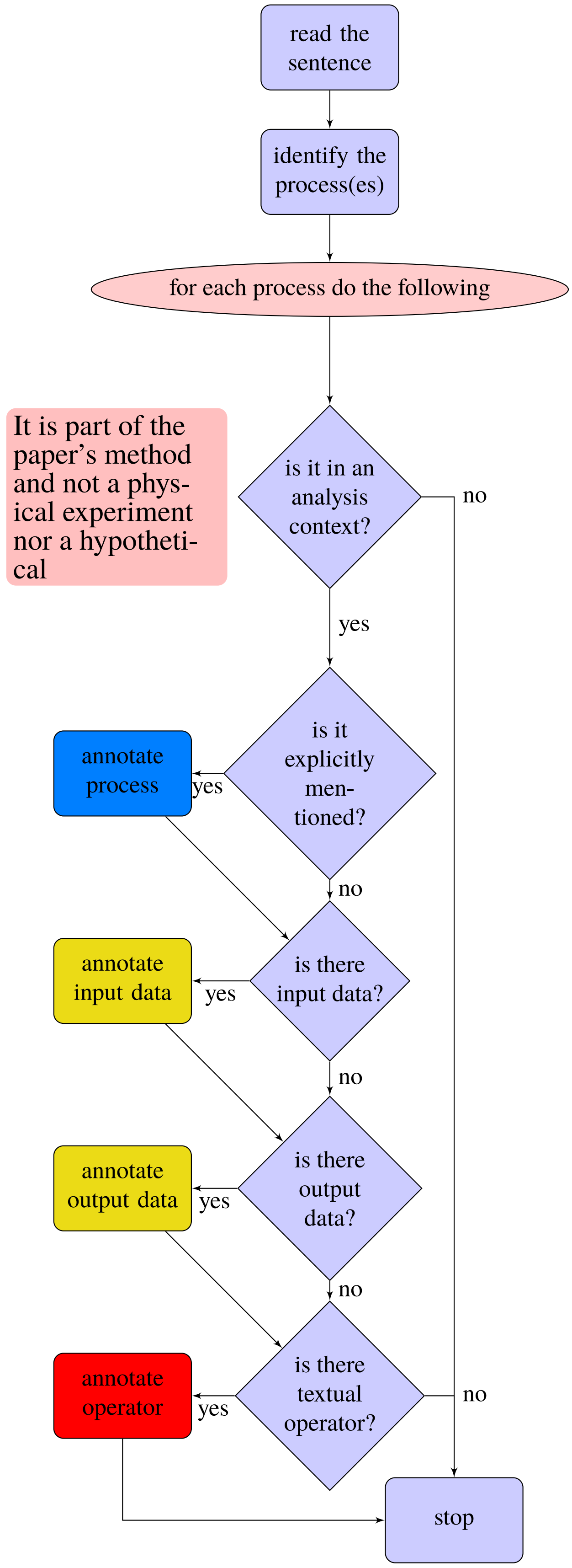

read the sentence
identify the process(es)
for each process do the following
It is part of the paper's method and not a physical experiment nor a hypothetical
is it in an analysis context?
no
yes
annotate process
yes
is it explicitly mentioned?
no
annotate input data
yes
is there input data?
no
annotate output data
yes
is there output data?
no
annotate operator
yes
is there textual operator?
no
stop

Figure A.2. A flowchart shows the steps of annotation

A.2.1 How to annotate operations

As explained previously, we are looking for operations conducted in the manuscripts. The operations that are part of other work are out of our scope. It is also important to notice that operations may or may not be mentioned with a data in the text. Following are some cases the annotator may encounter.

1. Operations and Operators.

   The operation may mentioned in the form of "a process is done using a sw/method/tool" or "a sw/method/tool is used/developed to perform a process". We are interested in the process itself (operation) not the sw/method/tool (means or the operators).

   Examples:(note here the **PMC ID** where the example is quoted from- the data is yellow highlighted - The operation is blue highlighted and the means are red highlighted)

   Example 26 (PMC4438953)
   "Fisher's exact test and X2 test were used to classify the GO category, and the FDR (20) was calculated to correct the P-value; the smaller the FDR, the smaller the error in judging the P-value."

   - For the first operation, the GO category is classified through the Fisher's exact test and X2 test. The tests are the methods applied, but the target operation is "the classification".

   Example 27 (PMC4268857)
   "Otsu's thresholding and smoothing was then performed to estimate a stromal fraction [40]."

   -note here "Otsu's thresholding and smoothing" is a facility/mean to do the target operation "estimating".

2. Hierarchical/nested operations.

   Example 28 (PMC4268857)
   "Gene-expression-based validation of the method was performed by generating a stromal gene list [22] and performing univariate Pearson's correlation testing between stromal quantification and expression."

   - note here the validation is done by generating a list and performing a test. Although here the test is a kind of method that we are not interested in (Section A.2.1 case 1), it is annotated here since it is the only mentioned operation done on the data (stromal quantification and expression)

3. A sentence may include more than one operation, either on the same data or different data.

   Example 29 (PMC2662875)
   “Only gene expression data obtained from a single study using the same methodology were compared. All data were log transformed, median centred per array, and the standard deviation was normalized to one per array [32].”

4. A sentence that has an operation that is part of the paper may include no data.

   Example 30 (PMC2714961)
   “We used an extension of Stouffer’s method [13] for the meta-analysis.”

5. Assign role operations

   This kind of operation did do an actual action on the data and did not change nor filter them. It is mainly assign a role to the data. The indication of using the data or including it in the study.

   `Assign roles = include consider compromise used emply`

   Example 31 (PMC2374988)
   “ The microarray breast cancer datasets considered in this work are described elsewhere [5,7,9,18,19].”

   Example 32 (PMC2582621)
   “In total, 355 probes are included in this study.”

   Example 33 (PMC2374988)
   “For these cohorts we used the normalized data, which are available in the public domain (see [5,7,9,18,19]).”

   - please note if the assign role is mentioned with another explicit operations that do main action and use the data in the sentence as input or out put then there is no need to annotate the assign role operations

   Example 34 (PMC4008137)
   “For analyzing the homologous gene clusters having m:n:p relationships between species, the gene clusters having a maximum of four related genes in each species were considered.”

   Here the operation is analysis (for analyzing). the “considered” should not be annotated.

Operation exclusion rules

Make sure not to annotate any operations that are not part of the conducted experiment. A kind of these operations may be mentioned in the description of what a method/sofware does in general in term of processing.

Example 35 (PMC3735399)
"HitPredict is a resource for high confidence protein-protein interactions. It(refers to a software) collects protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction according to the likelihood ratio using naïve Bayesian networks combining sequence, structure and function annotations of the interacting proteins [11]."

A.2.2 How to annotate data

1. Data gathered from a data source. The data source can be a database/dataset/data file/website/publication or extraction from another data set. In most cases data are considered output data from a retrieving process.

Example 36 (PMC2714961)
" The GWAS data for this analysis were retrieved from the Cancer Genetic Markers Susceptibility (CGEMS) database, http://cgems.cancer.gov/about/executive_summary.asp."

Example 37 (PMC4550637)
"In this study, the gene expression microarray data set GSE4612 was downloaded from the Gene Expression Omnibus (GEO, http://www.ncbi.nlm.nih.gov/geo/)."

Example 38 (PMC2582621)
" The Affymetrix Latin Square spike-in data U133A were retrieved from (12)."

Example 39 (PMC4008137)
" Orthologous gene clusters of human, mouse and rat were generated from the OMA database [36]."

Example 40 (PMC4438953)
" Microarray data were obtained from three datasets, which consisted of 18, 57 and 38 appropriate samples, respectively."

Example 41 (PMC2686605)
" Data were collected from in-house IFN-treated microarrays and more than 28 publications (18-45) identified through literature searches where high-throughput analysis (microarray or proteomic) was performed on cells/tissues treated only with IFNs."

2. Data that are input to or output from a computational process done by a software/method/calculation or an instrument. This include the required parameters/conditions to complete the processes.

   Examples:

   Example 42 (PMC2582621)
   " The probe self-folding energy was computed by RNAStructure [version 4.5, function OligoWalk (21)]."

   -"The probe self-folding energy" is output data

   Example 43 (PMC2582621)
   " Duplexing energy was computed by the current NN model with the parameters from Ref.(17)."

   -"Duplexing energy" is output data. "the parameters" is input data.

   Example 44 (PMC59472)
   " Data files were imported into Excel and the companies' internal controls were removed."

   -"Data files" and "the companies' internal controls" are input data

   Example 45 (PMC59472)
   " Intensity differences for pairs of control signals (S1-S2) were calculated, as well as average signals for each pair ((S1+S2)/2)."

   -"Intensity differences", "average signals" are output data.

   -"pairs of control signals" , "each pair" are input data.

   -Note here S1 and S2 are not annotated because they are strings used for explanation purposes.

3. Data that are input to or output from a previous mentioned operations(sometimes implicit operation). A hint of using this data in the methods's experiment is included in the context. Notice here the usage of the data itself is the key of

the annotation. We are not interested in further explanation of data details and what the dataset could contain. Point 2 in exclusion rules (Section A.2.2) explains the case of data details with examples. Examples:

Example 46 (PMC2374988)
"We also created an automated computational pipeline (Perl scripts on a Linux platform) to crosslink the annotation provided for each dataset with UniGene. For some datasets, the linkage relied on Ensembl [48] external database identifiers. Thus each probe was associated with a universal gene name."

-The second sentence gives more precise details about the data used in the operation mentioned in the first sentence.

Example 47 (PMC3074119)
"A random forest of 1000 trees was trained using these 12 features. The output consisted of the probabilities for the three classes and the class with the majority vote. The final prediction was made by summing over all probabilities for target gene assignments i and reporting the gene with the highest sum as the target gene."

4. Labelled/defined/identified data. Data that have a role more than being input or output of a process. Examples:

Example 48 (PMC2686605)
"Gene lists were analysed and genes that demonstrated 1.5-fold or more differential expressi were identified as IRGs."

IRGs here is a kind of identified data that are labelled to be used later in the manuscript.

Example 49 (PMC4268857) *"We subdivided tumours as negative (no staining in any tumour cell), weak positive (all tumour cells weakly stained compared to stromal cells), positive (all tumour and stromal cells equally stained) or heterogeneous (combination of positive and negative/weak staining) staining."*

-"tumours" is the input data.

-"negative, weak positive, positive and heterogeneous" output labelled data.

Example 50 (PMC4438953)
"Fisher's exact test and X2 test were used to select the significant pathways, and the threshold of significance was defined by the P-value (<0.05) and FDR (<0.05) (22-24)."

- “the threshold of significance” was defined in the context by the value of the annotated parameters (P-value and FDR). It is also here a kind of input data (parameter) to the tests.

Example 51 (PMC4438953)
“ A threshold was set at fold-change>4 , P<0.01 and FDR<0.01 , from which the TSCC-associated differentially expressed miRNAs were selected .”

Example 52 (PMC4438953)
“ The numerical data are presented as the mean+-standard deviation .”

The data in this context are not abstract data, since the presented here means calculated/identified.

Data exclusion rules

1. Real/physical data mentions

   Examples:

   Example 53 (PMC3735399)
   “A total of nine gene chips from mesenchymal cell samples, including five gene chips from osteoporosis patients and four gene chips from non-osteoporosis samples, were used for analysis.”

   Example 54 (PMC117803)
   “mRNA isolated from Wnt-3A exposed cells was reverse transcribed and labeled with Cy5 (red) and cDNA from CCM treated cells labeled with Cy3 (B).”

   Example 55 (PMC2612032)
   “Exon-skipping events were generated using experimental data kindly provided by Abdueva [3].”

   The experimental data here are physical data.

2. Hypothetical mentions

   The mention where the data are in a non event context or no computational process is being applied on them. Such as the data in the context of demonstration of methods, software or previous studies, or how they are dealing generally with data; the data that are not used in the method manuscripts. However, if a paper implements a software and the method section is all about how the software is dealing generally with data, then this is considered as abstract data. If the method used some data that go through the software and are processed to produce an output, then this is an analysis context that we are looking for. Here are some examples:

- Non data event context

  Example:

  Example 56 (PMC2582621)
  "Compiled data and computational scripts used in this study are available upon request."

- The purpose of using a database/software if there is no direct mention of the manipulated data.

  Example:

  Example 57 (PMC2714961)
  "We used the Oncomine database http://www.oncomine.org/main/index.jsp to conduct a meta-analysis of the number of studies comparing gene expression in normal prostate tissue with that of localized prostate tumor tissue [12]."

  Example 58 (PMC2662875)
  "We used Oncomine Cancer Microarray database http://www.oncomine.org[30] and Amazonia database http://amazonia.montp.inserm.fr/[31] to study gene expression of BAFF, APRIL, BCMA, TACI, BAFF-R and HS proteoglycans genes in 40 human tumor types and their normal tissue counterparts as indicated in Table 1 (Additional file 1)."

  Example 59 (PMC3735399)
  "HitPredict is a resource for high confidence protein-protein interactions."

- The Explanation of how methods/software/previous studies are dealing with data in general. Examples

  Example 60 (PMC3735399)
  "It(refers to a software) collects protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction according to the likelihood ratio using naïve Bayesian networks combining sequence, structure and function annotations of the interacting proteins [11]."

3. Data in files/tables/figures or in the supplementary files/tables/figures.

   Example:

   Example 61 (PMC2714961)
   “The complete list of the studies used in the meta-analysis can be found in the supplementary materials (Table S1).”

4. Platforms

   Example 62 (PMC4550637)
   “The platform of this microarray data is GPL339 [MOE430A] Affymetrix Mouse Expression 430A Array.”

A.3 What to Annotate

A.3.1 How long is the operation span?

It is recommended to annotate the one action that express the process.

A.3.2 How long is the data span?

The longest span that includes the used or resulted data.

A.3.3 How long is the event span?

The longest span that includes all the event's parameters. The operation, the input data, output data, any mentioned software or databases.

A.3.4 Do we annotate the data between parentheses?

`Yes, if it used as input or output to a process.`

Example 63 (PMC4268857) *"Scoring was performed according to intensity (using stromal cells as internal positive controls) and percentage of stained cells by two independent observers (FCM and MJL)."*

A.3.5 How granular is the annotation?

We are looking for a fine-grained annotation unless it divides the same data. The key point here is to differentiate between different inputs and outputs required for a process.

Here are some examples:

- How to annotate **between..and**
  - in the form "data z between data x and data y"

    Example 64 (PMC4438953)
    "The differentially expressed mRNAs between the TSCC and normal control samples were identified using the limma method (17)."

- in the form "kindOfOperation between data x and data y"

Example 65 (PMC4268857)
"Correlation between automated and manual scoring was performed using the Jonckheere-Terpstra test for trend."

Example 66 (PMC2919724)
"We then compared the number of protein-protein interactions among TF targets, and between TF targets and non-targets using a cumulative hypergeometric test."

- How to annotate **in**?
  - in the form " data z in data x". If the data x is not a data source. The are both data.

Example 67 (PMC3735399)
"GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR < 0.05."

-genes and the interaction network are both input data.

- How to annotate **of**?
  - in the form " data z of data x". In the case the data x is not a data source and it is input data needed to produce the output data z.

Example 68 (PMC4550637)
" Unwanted noise of the raw microarray data was filtered out in the preprocessing stage."

-Unwanted noise is data output and the raw microarray data is input data.

Example 69 (PMC4550637)
" The target mRNAs of the differentially expressed miRNAs were predicted based on TargetScan (http://www.targetscan.org/) version 5.2."

  - In case the data z is an already calculated value of the data x then "data z of data x" should be annotated as one data, since they express one thing.

Example 70 (PMC2998528)
" The behaviour of the three preprocessed procedures was analysed."

- How to annotate **for..of..** ?
    - in the form "data z is computed for data x of data y"
      In case of data x and data y are two required inputs to conduct the process.

      Example 71 (PMC4438953)
      P-values were calculated for the GO terms of all the differentially expressed genes.

- How to annotate **with**
    - in the form "data x with feature y".

      Example 72 (PMC2919724)
      "Predicted binding sites with a posterior probability > 0.5 were used in our analysis."

      Example 73 (PMC4262513)
      "Interactions with a score >0.8 were selected."

A.4 Entities

-The main annotations are **Data** and **Operation**.

A.5 Relations

Two relations are defined between the main entities.

- **inputData** : an input data to an operation.
- **outputData** : an output data from an operation at the same sentence or a previous sentence.

There are two more relations where SW and Databases are associated with data and operaions. The SW and Databases are annotated through bioNerDS.

- **bySW** : an operation performed by a software.
- **fromDB** : an output data from a database.

A.6 Annotation through BRAT

- There is an annotation schema created for operation and data and the relations. Stored in annotation.conf.
- The plain documents are on PlainDocs folder.
- The annotated files and relations are on BRAT_GS_ODDS_4Rel

Appendix B

Verbs list for hard-code normalisation

B.1 Stem of words used for normalisation and Rule based

We used the stem in Table to manually map operations to the corresponding EDAM classes.

Table B.1. Stem of words used for normalisation and Rule based.

EDAM Operation Class	stem of similar words
Data retrieval	referenc, obtain, download, collect, assembl, gather, consolidate, combine, extract
Assigning Role	as, set, consider, take, chosen, act, acts, identify, select, include, include study, choose, chose, used, use, use study, define, consider work
Calculation	add, insert, subtract, log transform, log2 transform, derive, scale, rescale, permut, obtain, compute, average, centre, center, summarize, summarise, determin, shift, shifting, test, sum, permute
Classification	categoriz, categoris, defin, divide, subdivid, dichotomized, dichotomiz dichotomis, grade, rank, segment, indicate, group
Filtering	match, remove, exclude, delete, reduce, focus, discard, leav, leaving
Amplification detection	amplification, amplify
Recognition	find , identify
Validation	evaluate, verify, verification, check
Modelling and simulation	train, execut
Editing	update, update extend, extend, adjust, adjustment, correct, fdr-adjuste, modifi, modify
Design	implement, develop
Mapping	map, link, associat, crosslink
Analysis	assess, asses, reassess, reasses, assessment, reassessment, estimate, mine

Appendix C

An Example of a Workflow Evaluation

C.1 Workflow evaluation

We are going to present an example of workflow evaluation of the methods' section of the article with PMCID PMC3735399. At the beginning we presented the prose of the methods section of the article. Then we did the completeness evaluation where lenient, intermediate and strict evaluations were presented. At the end we did the evaluation is for sequentiality of data events in a workflow where the events order is evaluated.

C.1.1 Methods section example

Example 74 (PMC3735399)
Methods

Affymetrix microarray

GSE35957 was downloaded from Gene Expression Omnibus (GEO) database (`http://www.ncbi.nlm.nih.gov/geo/`), which is based on GPL570 [HG-U133_Plus_2] Affymetrix Human information Genome U133 Plus 2.0 Array Platform (Affymetrix, Santa Clara, CA, USA). Microarray probe annotation information was downloaded from the Affymetrix Company, including all AffymetrixATH1(25K) gene chip probe information, and the probe annotation information files of the platform. A total of nine gene chips from mesenchymal cell samples, including five gene chips from osteoporosis patients and four gene chips from non-osteoporosis samples, were used for analysis.

Data preprocessing and analysis of differentially expressed genes

The original data were preprocessed by Affymetrix [7,8] package in R language. LIMMA [9] package in R language was used to identify the differentially expressed genes be-

tween the expression profile of five osteoporosis patients and four non-osteoporosis samples. Multiple testing correction was performed by Bayesian method [10]. An FDR <0.01 and |logFC| >1 were chosen as thresholds for screening the differentially expressed genes.

Prediction of interaction between differentially expressed genes

Differentially expressed genes play a role through interacting with each other. Therefore, we used HitPredict software `(http://hintdb.hgc.jp/htp/)` *to search the differentially expressed genes that can interact with OPG gene. HitPredict is a resource for high confidence protein-protein interactions. It collects protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction according to the likelihood ratio using naïve Bayesian networks combining sequence, structure and function annotations of the interacting proteins [11]. So far, HitPredict has 239584 protein-protein interactions across nine species, 168458 of which are predicted to be of high confidence. This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relationships.*

Module analysis of interaction network

MCODE (Molecular Complex Detection) detects densely connected regions in large protein-protein interaction networks that may represent molecular complexes. In this study, we used MCODE to mine the modules from the protein-protein interaction network with degree >2. Further, we used Bingo [13] to annotate each module based on the hypergeometric distribution (FDR <0.05).

Pathway enrichment analysis of interaction network

GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR <0.05. GENECODIS is a function analysis tool of gene, and it integrates different information resources (GO, KEGG or SwissProt), searches and arranges gene set annotation by statistical significance [14].

C.1.2 Completeness evaluation

Table C.1 shows the events workflow as extracted by manual annotation and system and identified them uniquely by row ID. Each system event is classified as good (true positive (TP)) and the row is coloured with the light green if the system event matches the manual annotation, as spurious (false positive (FP)) and the row is coloured with the blue if the system event is mistakenly extracted by the system or as missing (false negative (FN)) if the system failed to extract the corresponding manual annotated events. Partial, if the system partially extracts the corresponding manual annotated events and the row is coloured by purple.

Table C.1 shows three sub tables for the abstract workflow of the Methods section of the article with PMC3735399.

The first three rows are false positive because in row 1 (sent ID 3) the data were physical data not computational ones. In 2,3 (sent ID 11) the operations were a description of how the software works in general.

The rows IDs 4-11 and 15 are true positive.

12-14 and 15 are false negative since they include extra or missing some data or software.

In case of rows 12 and 13 (sent ID 13) the two events belong to the same sentence and for that some data are connected to the second operation while the should not. The sentence was:

Example 75 (PMC3735399/ Sentence 13)
"This study used the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relationships."

The system successfully extracted and related the output data to the two operations *(find, visualize)*. However, for the 2nd event, all the input data were incorrect. For the first one, there was an extra input data.

For row 14, the number *(2)* was missed from the condition **14 degree > 2** and the software *MCODE*.

For row 16, *GENECODIS* was a missed SW.

Table C.1. Completeness evaluation: Table (a) shows the data events evaluation based on manual annotation (Table (b)) and system annotation (Table (c)).

(a) data events evaluation

rowID	TP(good)	(partial)	FN(missing)	FP(spurious)
1				1
2				1
3				1
4	1			
5	1			
6	1			
7	1			
8	1			
9		1		
10		1		
11	1			
12		1		
13		1		
14		1		
15	1			
16		1		
total	7	6	0	3
out of	16	16	16	16

	Lenient	Intermediate	Strict
Precision	.81	.63	0.44
Recall	1	.77	0.54
F_1**-Scores**	.90	.69	0.48

(b) Manual annotation

rowID	sentID	eventID	eventType	operation	inputdata	outputData	SWs	DBs	txt
1									
2									
3									
4	1	1	DataEvent	downloaded		GSE35957		Gene Expression Omnibus	GSE35957 was downloaded from Gene Expression Omnibus
5	2	2	DataEvent	downloaded		Microarray probe annotation information			Microarray probe annotation information was downloaded
6	4	3	DataEvent	preprocessed	The original data		Affymetrix		The original data were preprocessed by Affymetrix
7	5	4	DataEvent	identify	the expression profile	differentially expressed genes	LIMMA		LIMMA [9] package in R language was used to identify differentially expressed genes between the expression profile
8	6	5	DataEvent	Multiple testing correction					Multiple testing correction
9	7	6	DataEvent	chosen as	FDR <0.01,|logFC| >1	thresholds			FDR <0.01 and |logFC| >1 were chosen as thresholds
10	7	7	DataEvent	screening	the differentially expressed genes,thresholds				thresholds for screening the differentially expressed genes
11	9	8	DataEvent	search	differentially expressed genes		HitPredict		HitPredict software (http://hintdb.hgc.jp/htp/) to search the differentially expressed genes
12	13	9	DataEvent	find	high confidence,protein-protein interactions,differentially expressed genes	interactions			protein-protein interactions with high confidence to find interactions between the differentially expressed genes
13	13	10	DataEvent	visualize	interactions	interaction relationships	Cytoscape		Cytoscape [12] to visualize the interaction relationships
14	15	11	DataEvent	mine	modules,degree >2,protein-protein interaction network		MCODE		MCODE to mine the modules from the protein-protein interaction network with degree >2
15	16	12	DataEvent	annotate	module,hypergeometric distribution,FDR <0.05				annotate each module based on the hypergeometric distribution (FDR <0.05
16	17	13	DataEvent	biological pathway enrichment analysis	interaction network,all genes,FDR <0.05		GENECODIS		GENECODIS was used to perform biological pathway enrichment analysis of all genes in the interaction network with FDR

(c) System annotation.

rowID	SyS_sentID	SyS_eventID	SyS_eventType	SyS_operation	SyS_inputdata	SyS_outputData	SyS_SWs	SyS_DBs	SyS_txt
1	3	3	DataEvent	for analysis	samples,nine gene chips				nine gene chips from mesenchymal cell samples, including five gene chips from osteoporosis patients and four gene chips from non-osteoporosis samples, were used for analysis
2	11	10	DataEvent	collects	interaction	protein-protein interactions,reliability score,interactions	IntAct	HPRD	collects protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction
3	11	11	DataEvent	annotates	protein-protein interactions,interaction	reliability score,interactions	IntAct	HPRD	protein-protein interactions from IntAct, BIOGRID and HPRD databases; annotates these interactions; and assigns a reliability score for each interaction
4	1	1	DataEvent	downloaded		Affymetrix microarray,GSE35957,Methods		Gene Expression Omnibus,GEO	MethodsAffymetrix microarrayGSE35957 was downloaded from Gene Expression Omnibus (GEO
5	2	2	DataEvent	downloaded		Microarray probe annotation information,the probe annotation information files,AffymetrixATH1(2 gene chip probe information			Microarray probe annotation information was downloaded from the Affymetrix Company, including all AffymetrixATH1(25K) gene chip probe information, and the probe annotation information files
6	4	4	DataEvent	preprocessed	original data		R		original data were preprocessed by Affymetrix [7,8] package in
7	5	5	DataEvent	identify	four non-osteoporosis samples,the expression profile of five osteoporosis patients	the differentially expressed genes	R		R language was used to identify the differentially expressed genes between the expression profile of five osteoporosis patients and four non-osteoporosis samples
8	6	6	DataEvent	Multiple testing correction					Multiple testing correction

(c) System annotation (Cont.).

rowID	SyS_sentID	SyS_eventID	SyS_eventType	SyS_operation	SyS_inputdata	SyS_outputData	SyS_SWs	SyS_DBs	SyS_txt
9	7	7	DataEvent	chosen as	>1,differentially expressed genes,An FDR <0.01	thresholds			An FDR <0.01 and $\lvert logFC \rvert$ >1 were chosen as thresholds for screening the differentially expressed genes
10	7	8	DataEvent	screening	An FDR <0.01,differentially expressed genes,thresholds,>1				An FDR <0.01 and $\lvert logFC \rvert$ >1 were chosen as thresholds for screening the differentially expressed genes
11	9	9	DataEvent	search	differentially expressed genes		HitPredict		dict software (http://hintdb.hgc.jp/htp/) to search the differential
12	13	12	DataEvent	find	interaction relationships,the protein-protein interactions with high confidence,the differentially expressed genes	interactions	Cytoscape		the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relation
13	13	13	DataEvent	visualize	interactions, the protein-protein interactions with high confidence,the differentially expressed genes	interaction relationships	Cytoscape		the protein-protein interactions with high confidence to find interactions between the differentially expressed genes, and used the Cytoscape [12] to visualize the interaction relation
14	15	14	DataEvent	mine	modules,the protein-protein interaction network,the,degree >				mine the modules from the protein-protein interaction network with degree >
15	16	15	DataEvent	annotate	module,FDR <0.05,hypergeometric distribution				annotate each module based on the hypergeometric distribution (FDR <0.05
16	17	16	DataEvent	biological pathway enrichment analysis	FDR <0.05,genes,interaction network		....		biological pathway enrichment analysis of all genes in the interaction network with FDR <0.05

C.2 Sequentiality of data events in a workflow evaluation

Table C.2 (a) shows the order of the events from the manual annotation. It got correct order for all events.

Table C.2 (b) shows the order of the events from the system annotation. It got 13 correct order out of 16 system events.

Table C.2. Sequentiality of data events in a workflow evaluation: Following are Table (a) that shows the data events evaluation based on manual annotation and Table (b) that shows the data events evaluation based on system annotation.

(a) Order evaluation: manual annotation

rowID	order	sentID	eventID	eventType	operation	inputdata	outputData	SWs	DBs	txt
4	1	1	1	DataEvent	downloaded		GSE35957		Gene Expression Omnibus	GSE35957 was downloaded from Gene Expression Om
5	1	2	2	DataEvent	downloaded		Microarray probe annotation information			Microarray probe annotation information was downloa
6	1	4	3	DataEvent	preprocessed	The original data		Affymetrix		The original data were preprocessed by Affymetrix
7	1	5	4	DataEvent	identify	the expression profile	differentially expressed genes	LIMMA		LIMMA [9] package in R language was used to identify differentially expressed genes between the expression pr
8	1	6	5	DataEvent	Multiple testing correction					Multiple testing correction
9	1	7	6	DataEvent	chosen as	FDR <0.01,|logFC| >1	thresholds			FDR <0.01 and |logFC| >1 were chosen as thresholds
10	1	7	7	DataEvent	screening	the differentially expressed genes,thresholds				thresholds for screening the differentially expressed gen
11	1	9	8	DataEvent	search	differentially expressed genes		HitPredict		HitPredict software (http://hintdb.hgc.jp/htp/) to search differentially expressed genes
12	1	13	9	DataEvent	find	high confidence,protein-protein interactions,differentially expressed genes	interactions			protein-protein interactions with high confidence to find interactions between the differentially expressed genes
13	1	13	10	DataEvent	visualize		interaction relationships	Cytoscape		Cytoscape [12] to visualize the interaction relationships
14	1	15	11	DataEvent	mine	modules,degree >2,protein-protein interaction network		MCODE		MCODE to mine the modules from the protein-protein interaction network with degree >2
15	1	16	12	DataEvent	annotate	module,hypergeometric distribution,FDR <0.05				annotate each module based on the hypergeometric distr (FDR <0.05
16	1	17	13	DataEvent	biological pathway enrichment analysis	interaction network,all genes,FDR <0.05		GENECODIS		GENECODIS was used to perform biological pathway analysis of all genes in the interaction network with FD
correct order total	13	out of	13							

(b) Order evaluation: system annotation.

rowID	order	SyS_sentID	SyS_eventID	SyS_eventType	SyS_operation	SyS_inputdata	SyS_outputData	SyS_SWs	SyS_DBs	SyS_txt
1	0	3	3	DataEvent	for analysis	samples,nine gene chips				nine gene chips from mesenchymal cell samples, i five gene chips from osteoporosis patients and four from non-osteoporosis samples, were used for anal
2	0	11	10	DataEvent	collects	interaction	protein-protein interactions,reliability score,interactions	IntAct	HPRD	collects protein-protein interactions from IntAct, B and HPRD databases; annotates these interactions; assigns a reliability score for each interaction
3	0	11	11	DataEvent	annotates	protein-protein interactions,interaction	reliability score,interactions	IntAct	HPRD	protein-protein interactions from IntAct, BIOGRID HPRD databases; annotates these interactions; and a reliability score for each interaction
4	1	1	1	DataEvent	downloaded		Affymetrix microarray,GSE35957,Methods		Gene Expression Omnibus,GEO	MethodsAffymetrix microarrayGSE35957 was downloaded from Gene Expression Omnibus (GE
5	1	2	2	DataEvent	downloaded		Microarray probe annotation information,the probe annotation information files,AffymetrixATH1(2 gene chip probe information			Microarray probe annotation information was downloaded from the Affymetrix Company, includ all AffymetrixATH1(25K) gene chip probe inform and the probe annotation information files
6	1	4	4	DataEvent	preprocessed	original data		R		original data were preprocessed by Affymetrix [7,
7	1	5	5	DataEvent	identify	four non-osteoporosis samples,the expression profile of five osteoporosis patients	the differentially expressed genes	R		R language was used to identify the differentially expressed genes between the expression profile of five osteoporosis patients and four non-osteopor
8	1	6	6	DataEvent	Multiple testing correction					Multiple testing correction

(b) Order evaluation: system annotation (Cont.).

rowID	order	SyS_sentID	SyS_eventID	SyS_eventType	SyS_operation	SyS_inputdata	SyS_outputData	SyS_SWs	SyS_DBs	SyS_txt
9	1	7	7	DataEvent	chosen as	>1,differentially expressed genes,An FDR <0.01	thresholds			An FDR <0.01 and |logFC| >1 were chosen as thre for screening the differentially expressed genes
10	1	7	8	DataEvent	screening	An FDR <0.01,differentially expressed genes,thresholds,>1				An FDR <0.01 and |logFC| >1 were chosen as thresholds for screening the differentially expressed genes
11	1	9	9	DataEvent	search	differentially expressed genes		HitPredict		dict software (http://hintdb.hgc.jp/htp/) to search th
12	1	13	12	DataEvent	find	interaction relationships,the protein-protein interactions with high confidence,the differentially expressed genes	interactions	Cytoscape		the protein-protein interactions with high confiden interactions between the differentially expressed g and used the Cytoscape [12] to visualize the intera
13	1	13	13	DataEvent	visualize	the protein-protein interactions with high confidence,interactions,the differentially expressed genes	interaction relationships	Cytoscape		the protein-protein interactions with high confiden interactions between the differentially expressed g and used the Cytoscape [12] to visualize the intera
14	1	15	14	DataEvent	mine	modules,the protein-protein interaction network,the,degree >				mine the modules from the protein-protein interaction network with degree >
15	1	16	15	DataEvent	annotate	module,FDR <0.05,hypergeometric distribution				annotate each module based on the hypergeometric (FDR <0.05
16	1	17	16	DataEvent	biological pathway enrichment analysis	FDR <0.05,genes,interaction network				biological pathway enrichment analysis of all genes in the interaction network with FDR <
correct order total	13									
out of	16									

Appendix D

Discourse Evaluation

D.1 Functional discourse evaluation at functions level and document level

SAPIENTA annotation is evaluated among the 25 documents. *A* denotes to SAPIENTA annotation and *B* denotes to our annotation. Table D.1 shows the evaluation at the functions level while Table D.2 details the evaluation among the 25 documents.

Table D.1. Functional discourse evaluation at functions level

Annotation	**Match**	**Only A**	**Only B**	**Overlap**	**Prec.B/A**	**Rec.B/A**	**F1.0-a.**
Background	5	4	38	0	0.1163	0.5556	0.1923
Conclusion	0	2	0	0	1.0000	0.0000	0.0000
Experiment	29	70	2	0	0.9355	0.2929	0.4462
Goal	2	5	1	0	0.6667	0.2857	0.4000
Hypothesis	1	0	1	0	0.5000	1.0000	0.6667
Method	346	88	144	12	0.7012	0.7892	0.7426
Model	27	11	24	1	0.5288	0.7051	0.6044
Object	2	25	15	0	0.1176	0.0741	0.0909
Observation	0	12	26	0	0.0000	0.0000	0.0000
Result	15	33	23	0	0.3947	0.3125	0.3488
Macro summary					0.4961	0.4015	0.3492
Micro summary	427	250	274	13	0.6071	0.6283	0.6175

Table D.2. Functional discourse evaluation at documents level

Document	Match	Only A	Only B	Overlap	Prec.B/A	Rec.B/A	F1.0-a.
PMC2714961.txt	4	5	5	0	0.4444	0.4444	0.4444
PMC2662875.txt	4	0	0	0	1.0000	1.0000	1.0000
PMC2686605.txt	6	4	4	0	0.6000	0.6000	0.6000
PMC2582621.txt	2	8	8	0	0.2000	0.2000	0.2000
PMC2374988.txt	23	37	37	0	0.3833	0.3833	0.3833
PMC4550637.txt	18	2	4	2	0.7917	0.8636	0.8261
PMC4438953.txt	17	22	22	1	0.4375	0.4375	0.4375
PMC4303952.txt	10	8	8	0	0.5556	0.5556	0.5556
PMC4268857.txt	32	16	16	0	0.6667	0.6667	0.6667
PMC4292761.txt	9	7	7	0	0.5625	0.5625	0.5625
PMC4289221.txt	18	6	6	0	0.7500	0.7500	0.7500
PMC4219025.txt	27	20	20	0	0.5745	0.5745	0.5745
PMC4262513.txt	19	7	7	0	0.7308	0.7308	0.7308
PMC4201588.txt	16	16	15	1	0.5156	0.5000	0.5077
PMC4149277.txt	15	17	17	0	0.4688	0.4688	0.4688
PMC4709009.txt	16	6	6	0	0.7273	0.7273	0.7273
PMC3806816.txt	36	4	4	3	0.8721	0.8721	0.8721
PMC4008137.txt	4	3	27	0	0.1290	0.5714	0.2105
PMC3735399.txt	13	4	4	1	0.7500	0.7500	0.7500
PMC3250460.txt	21	1	1	1	0.9348	0.9348	0.9348
PMC3123201.txt	11	10	9	1	0.5476	0.5227	0.5349
PMC3074119.txt	50	16	15	1	0.7652	0.7537	0.7594
PMC2998528.txt	25	17	18	1	0.5795	0.5930	0.5862
PMC2876170.txt	7	9	9	0	0.4375	0.4375	0.4375
PMC2919724.txt	24	5	5	1	0.8167	0.8167	0.8167
Macro summary					0.6096	0.6287	0.6135
Micro summary	427	250	274	13	0.6071	0.6283	0.6175