Chapter 5
Molecular Information Fusion in Ondex

Jan Taubert and Jacob Köhler

Abstract Current biological knowledge is buried in hundreds of proprietary and public life-science databases available on the World Wide Web (WWW) and millions of scientific publications. Gaining access to this knowledge can prove difficult as each database may provide different tools to query or show the data and may differ in their structure and user interface or uses a different interpretation of biological knowledge than others. Systems approaches to biological research require that existing biological knowledge (data) is made available to support on the one hand the analysis of experimental results and on the other hand the construction and enrichment of models. Data integration methods are being developed to address these issues by providing a consolidated view of molecular information fused together from multiple databases. However, a key challenge for data integration is the identification of links between closely related entries in different life sciences databases when there is no direct information that provides a reliable cross reference. Here we describe and evaluate three data integration methods to address this challenge in the context of a graph-based data integration framework (the Ondex system). We give a quantitative evaluation of their performance in two different situations: the integration and analysis of different metabolic pathways resources and the mapping of equivalent elements between the Gene Ontology and a nomenclature describing enzyme function.

Keywords Data integration • Systems Biology • Life sciences • Biological databases • Molecular information • Ontologies • Pathways • Ondex

J. Taubert (✉)
Wellcome Trust Genome Campus, Hinxton, Cambridgeshire CB10 1SD, UK
e-mail: taubertjan@gmail.com

J. Köhler
Dow AgroSciences LLC, 9330 Zionsville Road, Indianapolis, IN 46268, USA

M. Chen and R. Hofestädt (eds.), *Approaches in Integrative Bioinformatics: Towards the Virtual Cell*, DOI 10.1007/978-3-642-41281-3_5,

5.1 Introduction

Over the last decade, biological research has changed completely. The reductionism approach of studying only a few biological entities at a time in the past is being replaced by the study of the biological system as a whole today. Systems Biology [1] seeks to understand how complex biological systems work by looking at all parts of biological systems and how they interact with each other and form the complete whole. Systems Biology can be seen as a cycle (see Fig. 5.1) consisting of the following steps:

- Having a testable hypothesis about a biological system
- Conducting experimental validation of hypothesis
- Capturing and analysis of experimental results (usually 'omics' data)
- Gain new insights (data) about a biological system from analysis results
- Refine model about a biological system to derive new hypothesis

This process requires that existing biological knowledge (data) is made available to support on the one hand the analysis of experimental results and on the other hand the construction and enrichment of models for Systems Biology.

Effective integration of biological knowledge from databases scattered around the internet and other information resources (e.g. experimental data) is recognised as a prerequisite for many aspects of Systems Biology research and has been shown to be advantageous in a wide range of use cases such as the analysis and interpretation of 'omics' data [2], biomarker discovery [3] and the analysis of metabolic pathways for drug discovery [4]. However, systems for data integration have to overcome several challenges. For example, biological data sources may contain similar or overlapping coverage, and the user of such systems is faced with the challenge of generating a consensus data set or selecting the 'best' data source. Furthermore,

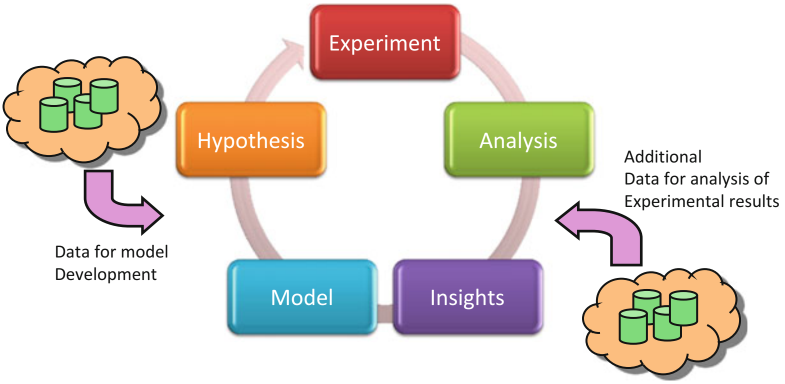
Experiment
Hypothesis
Analysis
Model
Insights
Data for model Development
Additional Data for analysis of Experimental results

Fig. 5.1 Systems Biology cycle of experiment, analysis, insights, model and hypothesis together with requirements for large data for analysis of experimental results and model development

there are many technical challenges to data integration, like different access methods to databases, different data formats, different naming conventions and erroneous or missing data.

To address these challenges and enable effective integration of data in support of Systems Biology research, the Ondex system [2, 5–7] which is presented in this chapter was created. The Ondex system provides an integrated view across biological data sources with the aim to enable the user to gain a better understanding of biology from integrated knowledge. Ondex has been supported by BBSRC (http://www.bbsrc.ac.uk/) as part of the systems approaches to biological research initiative (SABR) and is now mainly being developed at Rothamsted Research, Manchester University and Newcastle University. The first Ondex prototype was developed at University of Bielefeld.

This book chapter is a summary and extension to previous work published in [6, 8]. It adds a new dimension to previous work by presenting integration results across time and using *Homo sapiens* as selected organism for metabolic pathway resources. We will start out by surveying different life-science data integration systems. This overview is followed by establishing a selection of challenges data integration systems are faced with and dissecting how well current systems are dealing with them. We then give a brief motivation and introduction for the Ondex system. This is followed by presenting data integration and transformation methods motivated by the stated challenges. The performance of the data integration methods is then quantitatively evaluated in two different situations: the integration and analysis of different metabolic pathways resources and the mapping of equivalent elements between the Gene Ontology and a nomenclature describing enzyme function. A brief discussion is given at the end of this book chapter.

5.1.1 Survey of Current Data Integration Systems

Several data integration systems for use in biology and related domains are in use today. Some of them use a generic approach to answer a wide range of biological questions. Others are more limited in their scope and application domain. These systems are based on principles such as link integration and hypertext navigation, data warehouses, view integration and mediator systems, workflows and mashups [9].

Software tools that solve aspects of the data integration problem are being developed for some time. The early approaches, which produced popular software such as SRS [10], use indexing methods to link documents or database entries from different databases and provide a range of text and sequence-based search and retrieval methods for users to assemble related data sets. The methods used by SRS (and related tools) address what has been described as the technical integration challenge.

More recently, data integration approaches are developed that ‘drill down’ into the data and seek to link objects at a more detailed level of description. Many of these approaches exploit the intuitively attractive representation of data as graphs or networks with nodes representing things and edges representing how they are related. For example, a metabolic pathway could be represented by a set of nodes identifying the metabolites linked by edges representing enzymatic reactions. Data integration systems that exploit graph-based methods include PathSys [11] or BN++ [12] and the Ondex system [13]. Both BN++ and Ondex are available as open source software.

The Visual Knowledge and BioCAD [14] software tools provide good examples for how semantic networks can be used for representing biological knowledge. The definition of the integration data structure of Ondex has been inspired by this use of semantic networks in the biology domain.

Biozon [15] is a data warehouse which includes additional derived information, such as sequence similarity or function prediction, between data entries. STRING [16] shows that multiple information sources can be combined to provide evidence for the relationship between proteins. Similar to Biozon and STRING, Ondex facilitates the information fusion of other derived information between data entities. Such information has been successfully used to improve genome annotation of *Arabidopsis thaliana* in a use case of Ondex [17].

BNDB with BN++ is the most similar system to Ondex in terms of system design and methodology. The NeAT [18] toolkit highlights how graph analysis applied to biological networks can help to reveal new insights. Furthermore it is a good example of providing such functionality via a web page.

Concluding from the presented systems and common practice in Systems Biology [5, 19], the representation of biological data as graphs or networks is a preferred choice. The complexity of the graphs or networks varies from tool to tool, for example, NeAT works with simple node and edge lists, whereas BNDB/BN++ and Ondex use a semantic-enriched graph model. Some tools like Biozon or STRING focus on aspects of providing a ready integrated knowledge base to the users. On the other hand, tools like Ondex, BNDB/BN++ or PathSys provide the user with means to assemble integrated data sets on his/her own. Visual Knowledge/BioCAD or NeAT emphasise on the biological pathways and networks analysis.

Graphical user interaction is realised in a variety of ways. Knowledge base-focused projects like Biozon or STRING tend to use a web-based interface backed by a relational database. Other data integration toolkits like BNDB/BN++ or Ondex offer a database driven backend with a dedicated front-end application and possible web service-based access. NeAT or Visual Knowledge/BioCAD loads and integrates data in an ad hoc way as part of their analysis workflows.

5.1.2 Challenges for Data Integration

Biological knowledge such as protein interactions (Fig. 5.2a), metabolic pathways (Fig. 5.2b) or biological ontologies (Fig. 5.2c) can be interpreted or understood as a network or graph. Biological databases are, however, usually implemented using table centric data structures, which do not readily allow the utilisation of

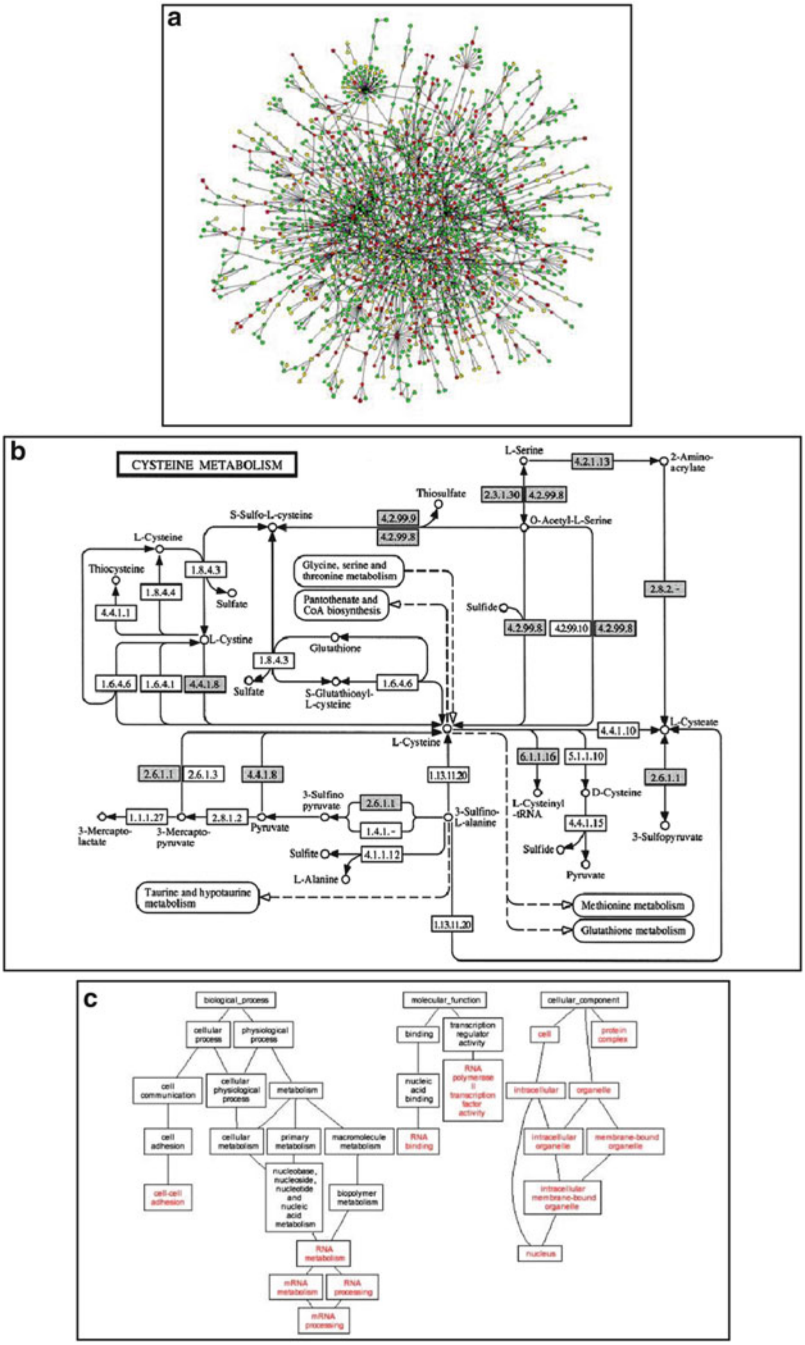

Fig. 5.2 Examples of biological knowledge as graphs: (**a**) protein interactions (Reproduced with permission from Jeong et al. [20] © Macmillan Magazines Ltd.), (**b**) metabolic pathways (Reprinted from Ogata et al. [21] with permission from Elsevier), (**c**) biological ontologies (Reprinted from Zhu et al. [22] under CC BY 2.0 licence © BioMed Central Ltd)

Table 5.1 Summarising outlined challenges for data integration systems

Challenge	Summary
First challenge	Representing biological data intuitively as a graph or network
Second challenge	Overcoming the syntactic and semantic heterogeneities between data sources
Third challenge	Provide a semantical consistent view on integrated information
Fourth challenge	Keep track of provenance during integration process
Fifth challenge	Domain-independent approach to data integration
Sixth challenge	Create a robust, usable and maintainable framework for data integration

graph analysis methods. Ondex uses a graph-based data structure which has been developed with an emphasis on providing integration of knowledge necessary for Systems Biology. Such a graph-based data structure should allow for the integration of heterogeneous data into a semantically consistent graph model and therefore support graph-based analysis algorithms and visualisation.

Biological data integration has to face the two problems of syntactic and semantic heterogeneity [23]. Syntactic heterogeneity is given by data being presented in different formats or as free text, containing spelling mistakes, wrong formatting or even missing data. Semantic heterogeneity is present in the different interpretations of data formats, symbols and names:

- Ambiguity of synonyms (exact/related), for example, Na(+)/K(+)-ATPase vs. just ATPase.
- Domain dependence of synonyms, for example, gene names in different organisms.
- Silent errors, like a typo in ENZYME Nomenclature is still valid entry (1.1.1.1 vs. 1.1.1.11).
- Unification references to other data sources can be ambiguous, for example, references to multiple splicing variants of a gene assigned to a protein.
- What is a gene, what is a protein and what is a transcript? Biological meaning is subject to interpretation and might vary.

To overcome syntactic and semantic heterogeneity in the data sources, knowledge modelling has to be adaptable for the respective domain of knowledge so that heterogeneous data sources can be transformed into a semantical consistent view. During this process it may be necessary to identify equivalent or redundant information in the data. Novel integration methods will have to be introduced to address this need. To establish trust in the integrated data, it is necessary to keep track of provenance during the whole data integration process.

Although this work has been mainly motivated by data from the life sciences, data integration is challenging in other data intensive sciences too. The integration methods should address this by being mostly domain independent. An example of a different application domain would be social networks. The methods presented in this chapter have been implemented as the core of the Ondex framework [2, 5]. One key aspect of the work on Ondex is to create a robust, usable and maintainable framework for data integration (Table 5.1).

Table 5.2 Challenges addressed by previous and current work

	First: data intuitively as graph or network	Second: addressing syntactic and semantic conflicts	Third: semantical consistent view	Fourth: track provenance	Fifth: domain independent	Sixth: robust, usable, maintainable framework
Visual Knowledge and BioCAD	Yes	No	Yes	No	No	Yes
Biozon	No	No	Yes	Yes	No	Yes
BNDB/BN++	Yes	Partially	Yes	No	Yes	Yes
STRING	Yes	No	Yes	Yes	No	Yes
NeAT	Yes	No	No	No	Yes	No

5.1.3 Comparison with Related Work

None of the previous presented data integration systems do address all the above-mentioned challenges as shown in Table 5.2.

The most important aspect not completely addressed by previous or related work is the second challenge of addressing syntactic and semantic heterogeneities between data sources in a systematic way. Knowledge base systems like STRING or Biozon use their own predefined database schema and load data from other data sources into this schema. During this process the mapping of source data to data objects in the system is hardwired and difficult to change. Overlapping or conflicting data between data sources usually does not get resolved. More complex systems like BNDB/BN++ provide adapters or parsers for different data sources and let the user of the system decide which selection of data source to integrate. Systems like NeAT or Visual Knowledge/BioCAD rely on the data to be in the correct format involving a larger amount of manual curation and work to be done upfront.

5.2 Motivation

Software designed for data integration in the life sciences has to address two classes of problem. It must provide a general solution to the technical (syntactic) heterogeneity, which arises from the different data formats, access methods and protocols used by different databases. More significantly, it must address the semantic heterogeneities arising from a number of sources in life-science databases. The most challenging source of semantic heterogeneity comes from the diversity and inconsistency among naming conventions for genes, gene functions, biological processes and structures among different species (or even within species). In recent years, significant progress in documenting the semantic equivalence of terms used in the naming of biological concepts and parts has been made in the development of a range of biological ontology databases which are coordinated

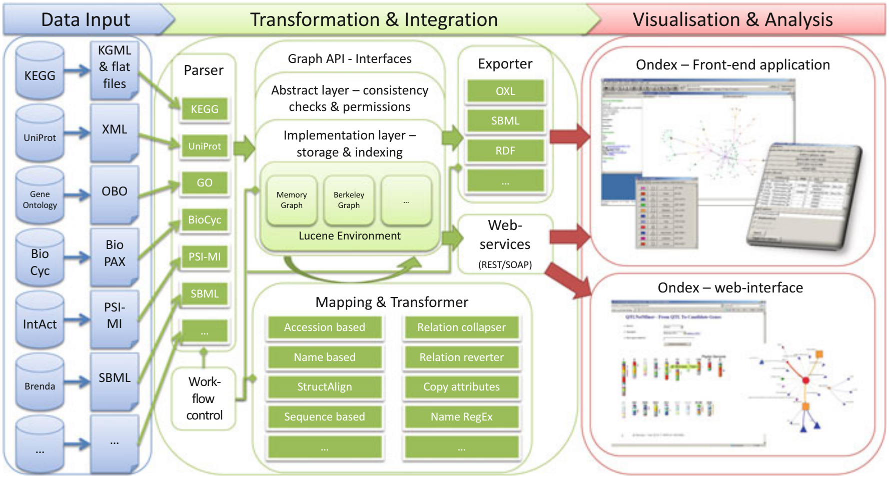

Fig. 5.3 Data integration in Ondex consists of three steps: (1) import and conversion of data sources into the data structure of Ondex (Data Input, *left*), (2) linking of equivalent or related entities of the different data sources and transformation into a semantical consistent graph (Transformation & Integration, *middle*), (3) knowledge extraction using the front-end application or web interface (Visualisation & Analysis, *right*)

under the umbrella of organisations such as the Open Biomedical Ontologies Foundry (http://www.obofoundry.org). However, the majority of biological terms still remain uncharacterised and therefore require automated methods to define equivalence relationships between them.

The integration of data in Ondex generally follows three conceptual stages as illustrated in Fig. 5.3: (1) normalising into the Ondex data structure in order to overcome predominantly technical heterogeneities between data exchange formats, (2) identifying equivalent and related entities among the imported data to overcome semantic heterogeneities at the entry level and (3) the data analysis, information filtering and knowledge extraction.

In order to make the Ondex system as extensible as possible, the second stage (middle bottom part in Fig. 5.3) has been separated both conceptually and practically. The motivations for doing this are to preserve original relationships and metadata from the original data source, make this integration step easily extensible with new methods, implement multiple methods for recognising equivalent data concepts to enhance the quality of integrated data and support reasoning methods that make use of the information generated in this step to improve the quality of integrated data.

The hypothesis here is that multiple methods for semantic data integration are necessary because of ambiguities and inconsistencies in the source data that will require different treatment depending on the source databases. In many cases, exact linking between concepts through unique names will not always be possible and therefore mappings will need to be made using inexact methods. Unless these inexact methods can be used reliably, the quality of the integrated data will be degraded.

To calibrate the presented data integration methods with well-structured data, the mapping of equivalent elements from the ontologies and nomenclatures extracted from the ENZYME [24] and GO [25] databases is used. To evaluate mapping methods in a more challenging integration task, the creation of an integrated data set from two important biological pathway resources, the Reactome [26] and HumanCyc [27] databases, is presented.

5.3 Methods

5.3.1 Data Import and Export

Following Fig. 5.3, the first step loads and indexes data from different sources. Ondex provides several options for loading data into the internal data warehouse, and a range of parsers have been written for commonly used data sources and exchange formats. In addition users can convert their data into an Ondex-specific XML or RDF dialect for which generic parsers are provided.

The role of all parsers is to load data from different data sources into the data structure used in the Ondex framework. In simple terms, this data structure can be seen as a graph, in which concepts are the nodes and relations are the edges. By analogy with the use of ontologies for knowledge representation in computer science, concepts are used to represent real-world objects [28]. Relations are used to represent the different ways in which concepts are connected to each other. Furthermore, concepts and relations may have additional properties and optional characteristics attached to them.

During the import process, names for concepts are lexicographically normalised by replacing non-alphanumeric characters with white spaces so that only numbers and letters are kept in the name. In addition, consistency checks are performed to identify, for example, empty or malformed concept names.

5.3.2 Data Integration Methods and Algorithms

The second step (following Fig. 5.3) links equivalent and related concepts and therefore creates relations between concepts from different data sources. Different combinations of mapping methods can be used to create links between equivalent or related concepts. Rather than immediately merging elements that are found to be equivalent, the mapping methods create a new equivalence relation between such concepts. After enough trust has been established in the results of the mapping methods by inspecting of these equivalence relations, then the information on similar elements can be fused, which is also known as molecular information fusion.

Each mapping method can be configured to create a score value reflecting the belief in a particular mapping and information about the parameters used. These scores are assigned as edge weights to the graph and form the foundation for the statistical analysis presented later. Additionally information on edges enables the user to track evidence for why two concepts were mapped by a particular mapping method.

Several constraints must be fulfilled before a mapping method creates a new link between two concepts. Under the assumption that the integrated data sources already contain all appropriate links between their own entries, new links are only created between different data sources. Biological databases often provide an NCBI taxonomy identifier for species information associated with their entries. If such identifiers are found in the graph, the mapping method ensures, in most cases, that relations are only created within the same species. In addition to species restriction, a mapping method takes the concept class of a concept into account. Only equal concept classes or specialisations of a concept class are considered to be included in a mapping pair.

5.3.2.1 Accession-Based Mapping

Most of the well-structured and managed public repositories of life-science data use accession coding systems to uniquely identify individual database entries. These codes are persistent over database versions. Cross references between databases of obviously related data (e.g. protein and DNA sequences) can generally be found using accession codes, and these can be easily exploited to link related concepts. Such concept accessions may not always present a one-to-one relationship between entries of different databases. For example, a GenBank accession found in the HumanCyc database is only unique for the coding region on the genome and not for the expressed proteins, which may exist in multiple splice variants. References presenting one-to-many relationships are call ambiguous. Concept accessions are indexed for better performance during information retrieval. Accession-based mapping by default uses only non-ambiguous concept accessions to create links between equivalent concepts, i.e. concepts that share the same references to other databases in a one-to-one relationship. This behaviour can be changed using a parameter.

Pseudocode

Let O denote the Ondex data structure consisting of a set of concepts $C(O)$ and a set of relations $R(O) \subseteq C(O) \times C(O)$. Every concept $c \in C(O)$ has a concept class $cc(c) \in CC(O)$, a data source identifier $ds(c) \in DS(O)$ and a list of concept accessions $ca(c) = \{(ca_1 \times \ldots \times ca_n) | ca_j \in CA(O)\}$. Each concept accession $ca \in CA(O)$ is a triple $ca = (ds, acc, ambiguous)$, where ds is the identifier of the data source from which the accession code acc is derived and $ambiguous$ is either true or false. The bijective function id assigns a consecutive number $n \in \mathsf{N}$ to concepts and relations in O separately starting with 1.

```
ignore ← true or false (default)
function AccessionBasedMapping(O, ignore) {
  for all $i \in [1..|C(O)|]$ do
    for all $j \in [i..|C(O)|]$ do
     if $\exists x \in ca(c_i) \wedge x \in ca(c_j) \wedge (\neg x.ambiguous \vee ignore)$ do
         if $ds(c_i) \neq ds(c_j) \wedge cc(c_i) = cc(c_j)$ do
          O.createRelation($c_i, c_j$)
}
```

Runtime Analysis

Assuming that the test if the two lists $ca(c_i)$ and $ca(c_j)$ have at least one concept accession in common takes linear time with respect to the length of the lists, for example, by using hashing strategies or ordered lists, and the average number of concept accessions on concepts is μ_{ca}, then the total runtime of accession-based mapping is $T(n) = \frac{1}{2}\left(n^2 + n\right) * \mu_{ca} \in O\left(n^2\right)$ where n is the number of concepts in the Ondex data structure.

5.3.2.2 Synonym Mapping

Entries in biological data sources often have one or more human-readable names, for example, gene names. Depending on the data source, some of these names will be exact synonyms such as the chemical name of a metabolite; others only related synonyms such as a general term for enzymatic function. Exact synonyms are especially flagged during the import process. Related synonyms are added to concepts as additional concept names. Concept names are preprocessed to strip all non-letter characters and stem special word cases before inserting them into the full-text index. Concept names are indexed for better performance and potentially fuzzy searches during information retrieval using the Apache Lucene (http://lucene.apache.org/) full-text indexing system. Fuzzy searches as supported by Lucene can be useful to overcome spelling mistakes, for example, PKM2 might be written as PK-M2 [29]. The default method for synonym mapping creates a link between two concepts if two or more concept names are matching (bidirectional best hits) to be able to cope with ambiguity of names. As a simple example of such ambiguity, the term ‘mouse’ shows that consideration of only one synonym is usually not enough for the disambiguation of the word, i.e. ‘mouse’ can mean computer mouse or the rodent *Mus musculus*. The threshold for the number of synonyms to be considered a match and an option to use only exact synonyms are parameters in the synonym mapping method.

Pseudocode

Let O denote the Ondex data structure consisting of a set of concepts $C(O)$ and a set of relations $R(O) \subseteq C(O) \times C(O)$. Every concept $c \in C(O)$ has a concept class $cc(c) \in CC(O)$, a data source identifier $ds(c) \in DS(O)$ and a list of concept names $cn(c) = \{(cn_1 \times \ldots \times cn_n) | cn_j \in CN(O)\}$. Each concept name $cn \in CN(O)$ is a tuple $cn = (name, exact)$, where *name* is the actual name of the concept and *exact* is either true or false. The bijective function *id* assigns a consecutive number $n \in \mathsf{N}$ to concepts and relations in O separately starting with 1.

```
num ← 1..$N$(default: 2)
exact ← true (default) or false
function SynonymMapping(O, num, exact) {
  for all $i \in$ [1..|$C$($O$)|] do
    for all$j \in$ [$i$..|$C$($O$)|] do
      if $|cn(c_i) \cap cn(c_j)| \geq num \wedge (\exists x \in cn(c_i) \cap cn(c_j) | x.exact \vee \neg exact)$ do
        if $ds(c_i) \neq ds(c_j) \wedge cc(c_i) = cc(c_j)$ do
          O.createRelation($c_i, c_j$)
}
```

Runtime Analysis

Assuming that the intersection of $cn(c_i)$ and $cn(c_j)$ can be found in linear time with respect to the size of the lists by using hashing strategies or ordered lists and the average number of concept names per concept is μ_{cn}, then the total runtime of synonym mapping is $T(n) = \frac{1}{2}\left(n^2 + n\right) * \mu_{cn} \in O\left(n^2\right)$ where n is the number of concepts in the Ondex data structure.

5.3.2.3 StructAlign Mapping

In some cases, two or more synonyms for a concept are not available in the data to be integrated. To disambiguate the meaning of a synonym shared by two concepts, the *StructAlign* mapping algorithm considers the graph neighbourhood of such concepts. A breadth-first search for a given depth ($\geq$1) starting at each of the two concepts under consideration yields the respective reachability list for each concept. *StructAlign* processes these reachability lists and searches for synonym matches of concepts at each depth of the graph neighbourhood. If at any depth one or more pairs of concepts which share synonyms are found, *StructAlign* creates a link between the two concepts under consideration.

Pseudocode

Let O denote the Ondex data structure consisting of a set of concepts $C(O)$ and a set of relations $R(O) \subseteq C(O) \times C(O)$. Every concept $c \in C(O)$ has two additional attributes assigned: (a) a concept class $cc(c) \in CC(O)$ characterising the type of real-world entity represented by the concept (e.g. a gene) and (b) a data source identifier $ds(c) \in DS(O)$ stating the data source (e.g. HumanCyc) the concept was extracted from. Every relation $r \in R(O)$ is a tuple $r = (f, t)$ with f the 'from'-concept and t the 'to'-concept of the relation. To improve performance the algorithm is making use of indexing structures for concept names and the unique identifier returned by the bijective function *id* which assigns a consecutive number $n \in \mathsf{N}$ to concepts and relations in O separately starting with 1.

```
index ← searchable index of concept names for concepts
cutoff ← maximal depth of graph neighbourhood search
function StructAlign(O, index, cutoff) {
  matches ← new map of concepts to sets of concepts
  // search for concept name hits
  for all $c \in C(O)$ do
    for all $n \in cn(c) | n.exact$ do
     hits ← index.search(n.name)
     for all $c' \in hits$ with $ds(c) \neq ds(c') \wedge cc(c) = cc(c')$ do
        matches[c].add(c')

 connectivity←new map of concepts to sets of concept
  // calculate direct neighbourhood
  for all $r \in R(O)$ with $r = (f, t)$ do
    if $ds(f) = ds(t) \wedge f \neq t$ do
      connectivity[f].add(t)
      connectivity[t].add(f)
  reachability ← clone(connectivity)
  // modified breadth first search with depth cutoff
  for all $i \in [1..cutoff]$ do
    for all $(x, (y_1 \ldots y_n)) \in reachability$ do
      for all $j \in [1..n]$ do
        reachability[x].addAll(connectivity[$y_i$])
  // look at neighbourhood of bidirectional matches
  for all $(a, (b_1 \ldots b_n)), (b_i, (c_1 \ldots c_m)) \in matches | a \in$
    $(c_1 \ldots c_m)$ do
    na ← reachability[a]
    nb ← reachability[$b_i$]
    for all $x \in na$ do
      if $\exists y \in matches[x] | y \in nb$do
        O.createRelation(a,$b_i$)
}
```

Runtime Analysis

Assuming the search for a concept name in the list of concept names takes logarithmic time with respect to the length of the list (e.g. using a self-balancing binary search tree [30]) and operations to manipulate maps and sets take constant time using hashing strategies, the runtime analysis is: Let c be the number of concepts, μ_{cn} the average number of concept names associated with a concept, r be the number of relations, μ_r the average number of relations per concept in the Ondex data structure and Δ a time constant for operations on maps and sets. The worst-case runtime of the StructAlign algorithm is then:

1. Search for concept name matches

$$T_1(c,r) = c * \mu_{cn} * \log(c * \mu_{cn}) * c * \Delta$$

2. Calculation of direct neighbourhood

$$T_2(c,r) = r * 2 * \Delta$$

3. Modified breadth-first search with depth cut-off

$$T_3(c,r) = \mathit{cutoff} * c * \mu_r * \Delta$$

4. Finding bidirectional matches in neighbourhood, log(c) search time for $\exists\, y$

$$T_4(c,r) = c^2 * c * \Delta$$

$$T\,(c,r) = T_1 + T_2 + T_3 + T_4$$

$$\begin{aligned} T\,(c,r) &= c * \mu_{cn} * \log(c * \mu_{cn}) * c * \Delta + r * 2 * \Delta \\ &\quad + \mathit{cutoff} * c * \mu_r * \Delta + c^2 * c * \Delta \end{aligned}$$

Within a fully connected graph, the number of relations is $r = c * (c-1)/2$ and $\mu_r = c - 1$.

$$T(c) = \begin{pmatrix} c * \mu_{cn} * \log(c * \mu_{cn}) * c + c * (c-1) \\ + \mathit{cutoff} * c * (c-1) + c^2 * c \end{pmatrix} * \Delta$$

$$T(c) = \left(c^2 * \mu_{cn} * \log(c * \mu_{cn}) + (1 + \mathit{cutoff}) * c * (c-1) + c^3\right) * \Delta$$

$$T(c) \in O\left(c^3\right)$$

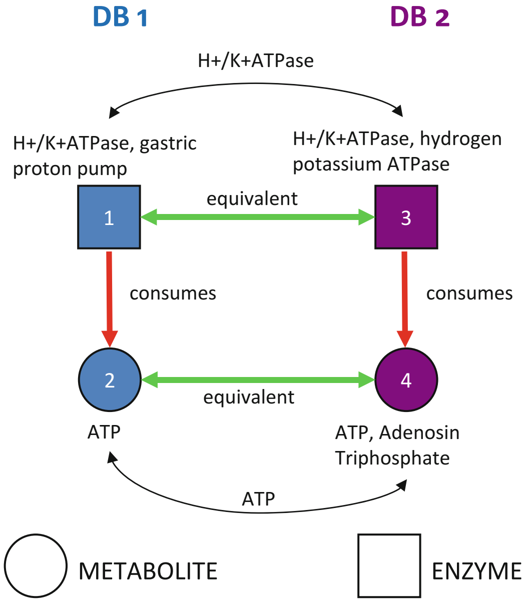
DB 1
DB 2
H+/K+ATPase
H+/K+ATPase, gastric proton pump
H+/K+ATPase, hydrogen potassium ATPase
1
3
equivalent
consumes
consumes
2
4
equivalent
ATP
ATP, Adenosin Triphosphate
ATP
METABOLITE
ENZYME

Fig. 5.4 Worked example for StructAlign. Different *shades* are used to distinguish data sources. *Node shape* represents different classes of concepts, *square* for enzymes and *circle* for metabolites. *Round arrows* show matching synonyms, whereas *vertical arrows* represent existing knowledge from data sources and *horizontal arrows* are created by StructAlign (color figure online)

Here the average number of concept names per concept is $\mu_{cn} \ll c$. Hence the algorithm has a worst-case runtime of $O(c^3)$. Although the expected runtime on sparse graphs is $O(c^2)$ as the number of neighbours reachable for a certain depth in a sparse graph is much smaller than the number of total concepts in the graph.

Worked Example for StructAlign

Figure 5.4 shows a simple example graph of metabolites (circles) and enzymes (rectangles) originating from two data sources DB1 (left) and DB2 (right). All concepts except for concept 2 have two synonyms (exact one listed first). The 'consumes' relation (vertical arrows) is present in both data sources DB1 and DB2.

StructAlign starts to consider the first pair of concepts, here concepts 1 and 3, which share at least one exact synonym (H+/K+ATPase) and are of the same concept class (enzyme). The reachability list of concept 1 includes concept 2 and the reachability list of concept 3 includes concept 4. The undirected breadth-first search of StructAlign will find concepts 2 and 4 both being present at depth 1. As concepts 2 and 4 share at least one exact synonym (ATP) and are of the same concept class (metabolite), StructAlign collected enough evidence to create a new relation (horizontal arrows) between concepts 1 and 3. In the next step, StructAlign proceeds to the next pair of concepts 2 and 4 between DB1 and DB2, which share at least one exact synonym and will map them as being equivalent (horizontal arrows) because of the name match present between concepts 1 and 3.

5.3.2.4 Other Data Integration Methods

In addition to the mapping methods presented afore and evaluated in this study, the following selection of mapping methods shows how other information can be incorporated to deduce new relationships between concepts. This functionality is similar to that seen in Biozon [15]. A more complete list of data integration methods can be found on the Ondex web page (http://www.ondex.org).

Transitive Mapping

Transitive relationships between concepts are inferred from existing relations. For example, if concept A is identified to be equivalent to concept B and concept B is known to be equivalent to concept C, then a new equivalent relationship between concept A and concept C is created by this mapping method.

Sequence Similarity Mapping

The computation of the similarity of gene or protein sequences is achieved by exporting the sequence data into a FASTA [31] file and performing the matching using BLAST [32] or TimeLogic Decypher (http://www.timelogic.com). The results are used to create relations between concepts representing the genes or proteins. The BLAST bit score and e-Value is assigned as attributes on these relations.

External2go Mapping

The GO consortium provides reference lists of GO terms that map terms to other classification systems, for example, EC [24] enzymes or PFAM domains. The *external2go* mapping parses these lists and creates relations between entries of the GO database and entries of the other classification system.

These few examples together with the methods listed on the web page illustrate the wide range of information which is utilised by mapping methods in Ondex including simple name matches, sequence similarity search, orthology prediction, graph-pattern matching and even complex text mining-based information retrieval. Furthermore it is not difficult to add new mapping methods to Ondex.

5.3.3 Data Transformation Methods

After similar or equivalent concepts have been identified by mapping methods, the relation collapse functionality is used to merge or fuse such clusters of similar concepts connected by equivalence relations into one single concept. During

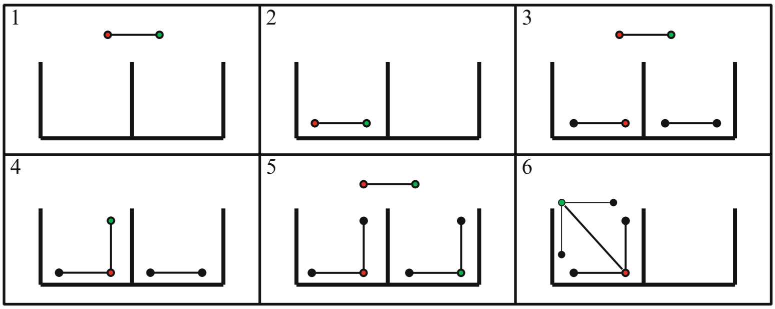

Fig. 5.5 Clustering of concepts, *1–2*; start new cluster, *3–4*; expand existing cluster, *5–6*; merge two existing clusters

this collapsing process, the molecular information of each original concept gets transferred onto the newly created fused concept, henceforth called molecular information fusion.

The collapsing of concepts consists of three main operations:

- Finding cluster of similar concepts
- Creating single collapsed concept
- Removing original concepts

Clustering of concepts, which is illustrated in Fig. 5.5, starts with iteration over all equivalence relations. For each such relation, it is determined if at least one of the two concepts connected by this relation is already a member of a cluster. If this is not the case, the relation and the two concepts are considered as the first element of a new cluster (steps 1 and 2). If one of the two concepts is already an element of an existing cluster, then the relation is added to this cluster (steps 3 and 4). If the two concepts are elements of two different clusters, these clusters are merged (steps 5 and 6).

The algorithm works with four temporary sets: nodes_open, nodes_closed, edges_open and edges_closed. The 'open' sets contain all known elements yet to explore. The 'closed' sets contain all already processed elements. The routine iterates over all concepts in the Ondex graph. For each concept all its adjacent relations are explored. If an equivalence relation is found, it is added to the edges_open set. The concept is then moved to nodes_closed, and the algorithm proceeds to explore all adjacent concepts of the elements of edges_open and moves them to edges_closed. In this fashion the algorithm switches between 'node exploration' and 'edge exploration' until no further elements to be processed are found. To avoid visiting elements which have already been analysed again, they are stored in a binary search tree so that they can be quickly re-identified. Hence each initial concept of the iteration is checked against this data structure before processing it.

The actual collapse process, which is done for every identified cluster of concepts, consists of the following steps:

- A collapse core node is created in the Ondex graph. If many nodes are collapsed into a single node, all properties of the collapsed nodes are assigned to the single representative.
- The edges going to nodes outside the current concept cluster are passed over to the collapse core node.
- All concepts of the current concept cluster are removed from the Ondex graph.

Runtime Analysis

The 'contains' and 'add' operations on the set data types in this algorithm have a runtime of $O(\log(n))$ using tree-based set data types. Let c be the number of the concepts in the Ondex graph and let μ_{cs} be the average cluster size. Then the worst-case runtime of the concept clustering algorithm is

$$O = (c * \mu_{cs} * \log(\mu_{cs}))$$

Hence the overall complexity of the algorithm is linear logarithmic.

5.3.4 Evaluation Methods

The mapping algorithms presented here can be configured using different parameters. According to the selection of the parameters, these methods yield different mapping results. To evaluate their behaviour, two different test scenarios were used: the mapping of equivalent elements in ontologies and the integration and analysis of metabolic pathways.

The evaluation of a mapping method requires the identification of a reference data set, sometimes also referred to as a 'gold standard', describing the links that really exist between data and that can be compared with those which are computed. Unfortunately, it is rare that any objective definition of a 'gold standard' can be found when working on biological data sets, and so inevitably most such evaluations require the development of expertly curated data sets. Since these are time consuming to produce, they generally only cover relatively small data subsets, and therefore the evaluation of precision and recall is inevitably somewhat limited.

In the next section, the results of mapping together two ontologies, namely, the Enzyme Commission (EC) nomenclature [24] and Gene Ontology (GO) [25], are presented. In this case, the Gene Ontology project provides a manually curated mapping to the ENZYME Nomenclature called *ec2go*. Therefore, *ec2go* has been selected as the first gold standard. The cross references between the two ontologies contained in the integrated data were also considered as the second gold standard for this scenario.

The following section also presents the results from the evaluation of a mapping created between the two metabolic pathway databases Reactome and HumanCyc. Unfortunately, a manually curated reference set is not available for this scenario. Therefore, it was necessary to rely on the cross references between the two databases that can be calculated through accession-based mapping as the nearest equivalent of a gold standard for this scenario.

5.4 Results

The mapping algorithms were evaluated using the standard measures of precision (Pr), recall (Re) and F_1-score [33]:

$$\mathrm{Pr} = \frac{tp}{tp + fp} \quad \mathrm{Re} = \frac{tp}{tp + fn} \quad F_1 = \frac{2 * \mathrm{Pr} * \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}}$$

The accession-based mapping algorithm (Acc) was used with default parameters, i.e. only using non-ambiguous accessions. This choice has been made to obtain a 'gold-standard' through accession-based mapping, i.e. increasing the confidence in the relations created. When evaluating the synonym mapping (Syn) and StructAlign (Struct) algorithms, parameters were varied to examine the effect of the number of synonyms that must match for a mapping to occur. This is indicated by the number in brackets after the algorithm abbreviation (e.g. Struct(1)). A second variant of each algorithm in which related synonyms of concepts were used to find a mapping was also evaluated. The use of this algorithmic variant is indicated by an asterisk suffix on the algorithm abbreviation (e.g. Syn(1)*).

5.4.1 Mapping Methods: ENZYME Nomenclature vs. Gene Ontology

The goal of this evaluation was to maximise the projection of the Enzyme Commission (EC) nomenclature onto the Gene Ontology. This would assign every EC term one or more GO terms. This evaluation has been carried out twice, once in January 2008 and a second time in the January 2013. The comparison of both results highlights the improvements made to the mapping between the two ontologies during this period.

For the first evaluation in 2008, ec2go (revision 1.67, downloaded 2008/01/21) and gene_ontology_edit.obo (revision 5.661, downloaded 2008/01/21) obtained from ftp://ftp.geneontology.org were used. Additionally enzclass.txt (last update 2007/06/19) and enzyme.dat (release of 2008/01/15) were downloaded from ftp://ftp.expasy.org.

Table 5.3 Mapping results for ENZYME Nomenclature to Gene Ontology in 2008

	TP, FP	TP, FP	Pr, Re [%]	Pr, Re [%]	F_1-score	F_1-score
Method	ec2go	Acc	ec2go	Acc	ec2go	Acc
Ec2go	8063, 0	8049, 14	100.00, 100.00	99.83, 84.82	100.00	91.71
Acc	8049, 1441	9490, 0	84.82, 99.83	100.00, 100.00	91.71	100.00
Syn(1)	7460, 934	7462, 932	88.87, 92.52	88.90, 78.63	90.66	83.45
Syn(1)*	7605, 2581	7606, 2580	74.66, 94.32	74.67, 80.15	83.35	77.31
Syn(2)*	4734, 374	4738, 370	92.68, 58.71	92.76, 49.93	71.89	64.91
Syn(3)*	2815, 117	2816, 116	96.01, 34.91	96.04, 29.67	51.21	45.34
Struct(1)	1707, 63	1712, 58	96.44, 21.17	96.72, 18.04	34.72	30.41
Struct(1)*	1761, 279	1766, 274	86.32, 21.84	86.57, 18.61	34.86	30.63
Struct(2)	7460, 934	7462, 932	88.87, 92.52	88.90, 78.63	90.66	83.45
Struct(2)*	7605, 2581	7606, 2580	74.66, 94.32	74.67, 80.15	83.35	77.31
Struct(3)	7460, 934	7462, 932	88.87, 92.52	88.90, 78.63	90.66	83.45
Struct(3)*	7605, 2581	7606, 2580	74.66, 94.32	74.67, 80.15	83.35	77.31

Ec2go imported mapping list (1st gold standard), *Acc* accession-based mapping (2nd gold standard), *Syn* synonym mapping, *Struct* StructAlign, * allow related synonyms, *TP* true positives, *FP* false positives, *Pr* precision, *Re* recall, F_1-score. Synonym mapping was parameterised with a number that states how many of the names had to match to create a link between concepts. StructAlign was parameterised with the depth of the graph neighbourhood

For the second evaluation in 2013, ec2go (revision 1.487, downloaded 2012/12/22) and gene_ontology_edit.obo (daily built, downloaded 2012/12/22) have been retrieved, together with enzclass.txt (release of 2012/11/28) and enzyme.dat (release of 2012/11/28).

The data files were parsed into the Ondex data structure and the mapping algorithms applied using the Ondex pipeline. To determine the optimal parameters for this particular application case, different combination of the mapping algorithms with the variants and parameter options as described above have been systematically tested. Table 5.3 summarises the mapping results and compares the performances with the ‘gold standards’ data sets from ec2go and by accession mapping (Acc) generated during our analysis in 2008. Table 5.4 shows the same information for analysis results produced in 2013.

The first two rows of Tables 5.3 and 5.4 show the performance of the ‘gold standard’ methods tested against themselves. As can be seen by reviewing the F_1-scores in the subsequent rows of Tables 5.3 and 5.4, the most accurate synonym mapping requires the use of just one synonym. It does not help to search for further related synonyms (Syn(1,2,3)*). The explanation for this is that the EC nomenclature does not distinguish between exact and related synonyms. Therefore, concepts belonging to the EC nomenclature have only one preferred concept name (exact synonym) arbitrarily chosen to be the first synonym listed in the original data sources. A large number of entries in the EC nomenclature only have one synonym described, which explains the low recall of Syn(2)* and Syn(3)*.

The use of the more complex StructAlign algorithm, which uses the local graph topology to identify related concepts, has low recall when only a single synonym is

Table 5.4 Mapping results for ENZYME Nomenclature to Gene Ontology in 2013

	TP, FP	TP, FP	Pr, Re [%]	Pr, Re [%]	F_1-score	F_1-score
Method	ec2go	Acc	ec2go	Acc	ec2go	Acc
Ec2go	8120, 0	8117, 3	100.00, 100.00	99.96, 77.57	100.00	87.35
Acc	8117, 2347	10464, 0	77.57, 99.96	100.00, 100.00	87.35	100.00
Syn(1)	6954, 498	7024, 428	93.32, 85.64	94.26, 67.13	89.31	78.41
Syn(1)*	7413, 2181	7538, 2056	77.27, 91.29	78.57, 72.04	83.70	75.16
Syn(2)*	4673, 537	4748, 462	89.69, 57.55	91.13, 45.37	70.11	60.58
Syn(3)*	2841, 189	2886, 144	93.76, 34.99	95.25, 27.58	50.96	42.77
Struct(1)	1449, 77	1466, 60	94.95, 17.84	96.07, 14.01	30.04	24.45
Struct(1)*	1541, 293	1562, 272	84.02, 18.98	85.17, 14.93	30.96	25.40
Struct(2)	7041, 605	7116, 530	92.09, 86.71	93.07, 68.00	89.32	78.59
Struct(2)*	7413, 2273	7538, 2148	76.53, 91.29	77.82, 72.04	83.26	74.82
Struct(3)	7041, 605	7116, 530	92.09, 86.71	93.07, 68.00	89.32	78.59
Struct(3)*	7413, 2273	7538, 2148	76.53, 91.29	77.82, 72.04	83.26	74.82

Ec2go imported mapping list (1st gold standard), *Acc* accession-based mapping (2nd gold standard), *Syn* synonym mapping, *Struct* StructAlign, * allow related synonyms, *TP* true positives, *FP* false positives, *Pr* precision, *Re* recall, F_1-score. Synonym mapping was parameterised with a number that states how many of the names had to match to create a link between concepts. StructAlign was parameterised with the depth of the graph neighbourhood

required to match and a depth cut-off of 1 is used (Struct(1) and Struct(1)*). This almost certainly results from differences in graph topology between EC nomenclature and Gene Ontology. The Gene Ontology has a more granular hierarchy, i.e. there is more than one hierarchy level between two GO terms mapped to EC terms, whereas the EC terms are only one hierarchy level apart. As the StructAlign depth cut-off search parameters are increased, more of the graph context is explored and accordingly the F_1-scores improved.

Across both tables, the highest F_1-scores come from Syn(1), Struct(2) and Struct(3), respectively. Including the related synonyms into the search (the * algorithm variants) did not improve precision. Neither did extending the graph neighbourhood search depth from Struct(2) to Struct(3) as all the neighbourhood matches had already been found within search depth 2.

During the integration of data from these data sets for this evaluation in 2008, some inconsistencies in the ec2go mapping list have been observed. The identification of such data quality issues is often a useful side effect of developing integrated data sets. The inconsistencies identified are listed in Table 5.5 and were revealed during the import of the ec2go data file after preloading the Gene Ontology and EC nomenclature into Ondex.

Presumably most of the problems are due to the previously disjoint development of both ontologies, i.e. GO references that were transferred or EC entries being deleted or vice versa. A few of the inconsistencies were possible typo errors. It remains a possibility that other 'silent' inconsistencies are still in ec2go that these integration methods would not find.

Table 5.5 Inconsistencies in ec2go in 2008

Accession	Mapping	Reason for failure
GO:0016654	1.6.4.-	Enzyme class does not exist, transferred entries
GO:0019110	1.18.99.-	Enzyme class does not exist, transferred entries
GO:0018514	1.3.1.61	Enzyme class does not exist, deleted entry
2.7.4.21	GO:0050517	GO term obsolete
GO:0047210	2.4.1.112	Enzyme class does not exist, deleted entry
1.1.1.146	GO:0033237	GO term obsolete
GO:0016777	2.7.5.-	Enzyme class does not exist, transferred entries
GO:0004712	2.7.112.1	Enzyme class does not exist, possible typo
2.7.1.151	GO:0050516	GO term obsolete

Every inconsistency was checked by hand against gene_ontology_edit.obo, enzclass.txt and enzyme.dat

A more recent analysis of data files used in 2013 revealed that the above presented inconsistencies have been corrected. The only inconsistencies identified in the newer data were:

- 1.3.5.6 to GO:0052889 (GO term is biological process, not molecular function)
- 2.5.1.46 to GO:0050983 (GO term is biological process, not molecular function)
- 2.1.1.35 to GO:0009021 (GO term obsolete)

5.4.2 Mapping Methods: Reactome vs. HumanCyc

The Reactome and HumanCyc pathway resources are both valuable for biologists interested in metabolic pathway analysis. Due to the different philosophies behind these two databases [34], however, they do have differences in their contents. Biomedical scientists wishing to work with biochemical pathway information would therefore benefit from a combined view of Reactome and HumanCyc and so this makes a realistic test. These two databases were chosen for this evaluation, because both pathway databases annotate metabolites and proteins in the pathways with standardised ChEBI [35] and UniProt [36] accessions, respectively. It is therefore possible to evaluate the precision, recall and F_1-score of the different mapping methods using accession-based mapping between these accession codes as a ‘gold standard’.

For this evaluation the BioPAX [37] representations of the Reactome database (release 43 from 2012/12/10) obtained from http://www.reactome.org/download and the HumanCyc database (release 16.5 from 2012/11/06) obtained from http://humancyc.org/download.shtml were used. The Reactome database contained 1,387 metabolites and 4,650 proteins. The HumanCyc database contained 1,983 metabolites and 2,690 proteins. The evaluation results from the mapping between metabolites from these two databases are summarised in Table 5.6.

Accession-based mapping between metabolites found 856 out of 1,387 possible mappings. A closer look reveals that ChEBI identifiers are not always assigned

Table 5.6 Mapping results for Reactome and HumanCyc databases – metabolites

Method	TP	FP	Pr [%]	Re [%]	F$_1$-score
Acc	856	0	100.00	100.00	100.00
Syn(1)	218	530	29.14	25.47	27.18
Syn(1)*	468	1598	22.65	54.67	32.03
Syn(2)*	144	420	25.53	16.82	20.28
Syn(3)*	40	184	17.86	4.67	7.41
Struct(2)	238	606	28.20	27.80	28.00
Struct(2)*	430	1506	22.21	50.23	30.80
Struct(3)	238	606	28.20	27.80	28.00
Struct(3)*	430	1506	22.21	50.23	30.80

Acc accession-based mapping (gold standard), *Syn* synonym mapping, *Struct* StructAlign, * allow related synonyms, *TP* true positives, *FP* false positives, *Pr* precision, *Re* recall, F1-score. Synonym mapping was parameterised with a number that states how many of the names had to match to create a link between concepts. StructAlign was parameterised with the depth of the graph neighbourhood

Table 5.7 Mapping results for Reactome and HumanCyc databases – proteins

Method	TP	FP	Pr [%]	Re [%]	F$_1$-score
Acc	2826	0	100.00	100.00	100.00
Syn(1)	10	28	26.32	0.35	0.70
Syn(1)*	514	226	69.46	18.19	28.83
Syn(2)*	14	0	100.00	0.50	0.99
Struct(2)	46	36	56.10	1.63	3.16
Struct(2)*	288	112	72.00	10.19	17.85
Struct(3)	46	36	56.10	1.63	3.16
Struct(3)*	288	112	72.00	10.19	17.85

Acc accession-based mapping (gold standard), *Syn* synonym mapping, *Struct* StructAlign, * allow related synonyms, *TP* true positives, *FP* false positives, *Pr* precision, *Re* recall, F1-score. Synonym mapping was parameterised with a number that states how many of the names had to match to create a link between concepts. StructAlign was parameterised with the depth of the graph neighbourhood

to metabolite entries, most notably in HumanCyc. Therefore, the accession-based mapping does miss possible links and cannot be used naively as a gold standard for this particular application case. In this evaluation, accession-based mapping underestimates possible mappings, which leads to low precision for synonym mapping and StructAlign. A random set of the false-positive mappings returned by Syn(2)* and Struct(3) has been manually reviewed, and this revealed that a large number of the mappings made sense and metabolites shared very similar chemical names. Subject to further investigation, this example shows that relying only on accession-based data for integration might miss out some important links between data sources.

The evaluation results from the mapping between proteins from Reactome and HumanCyc are summarised in Table 5.7.

The accession-based mapping between proteins uses the UniProt accessions available in both Reactome and HumanCyc. Entries from HumanCyc can be labelled with two or more UniProt accessions representing multiple proteins involved in the same enzymatic function, whereas Reactome entries usually only have one UniProt accession. This results in one-to-many hits between Reactome and HumanCyc explaining why a total of 2,826 instead of only 2,690 mappings were found. This is a good example of how the differences in the semantics between biological data sources make it difficult to define a gold standard for evaluating integration methods.

The key finding from this evaluation based on mapping protein names is that due to different protein naming conventions in each of the two databases, name-based mapping methods cannot perform well. Manual inspection of a subset of false-negative mappings and their protein names reveals that HumanCyc is using longer names describing enzymatic functions (e.g. cytidine deaminase, cytidine aminohydrolase), whereas Reactome uses short gene names (e.g. CDA, CDD).

5.4.2.1 Visualising Results

Data integration involving large data sets can create very large networks that are densely connected. To reduce the complexity of such networks for the user, information filtering, network analysis and visualisation (see Fig. 5.3, step 3) are provided in a front-end application for Ondex [2]. The combination of data integration and graph analysis and visualisation has been shown to be valuable for a range of data integration projects in different domains, including microarray data analysis [2], support of scientific database curation [38, 39] and assessing the quality of terms and definitions in ontologies such as the Gene Ontology [40].

A particularly useful feature in the Ondex front-end is to visualise an overview of the types of data that have been imported into Ondex. This overview is called the Ondex meta-graph. It is generated as a network based on the data structure used in Ondex, which contains a type system for concepts and relations. Concepts are characterised using a class hierarchy and relations have an associated type. This information about concept classes and relation types is visualised as a graph with which the user can interact to specify semantic constraints – such as changing the visibility of concepts and relations in the visualisation and analysis of the integration data structure.

As an illustration, the integration of Reactome and HumanCyc for this evaluation results in more than 61,000 concepts and 113,000 relations. The mapping methods were run with optimal parameters identified in the previous section. After filtering down to a specific pathway using methods available in the Ondex front-end, it was possible to extract information from the integrated data as presented in Fig. 5.6.

Figure 5.6a displays parts of the *MAP kinase cascade* pathway from HumanCyc (nodes and edges in black) mapped to the corresponding entries from Reactome (indicated by bidirectional edges to blue nodes). It is now possible to visualise the differences between the two integrated pathways. Reactome contains more protein entities about a specific enzymatic function (e.g. proteins similar to *phospho-MEK*).

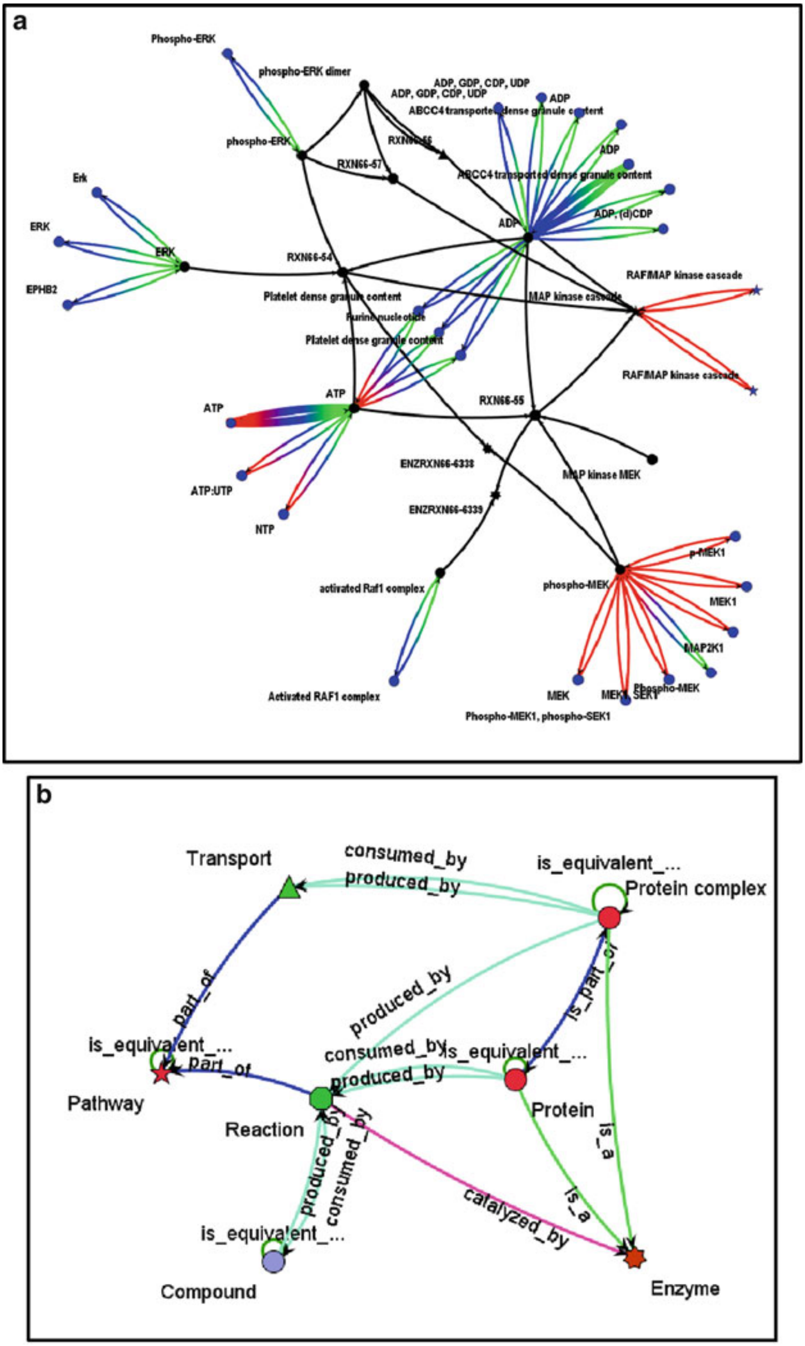

Fig. 5.6 (**a**) *MAP kinase cascade* pathway from HumanCyc with entities from Reactome. Equivalence relations are coloured by method (*red* = accession, *blue* = synonym, *green* = StructAlign) and thickness by StructAlign score. (**b**) Meta-graph providing an overview of the integrated data; *node colour* and *shape* distinguish classes; *edge colour* distinguishes different relation types (color figure online)

HumanCyc provides a larger pathway composed of more proteins than the pathway in Reactome, as the pathway concept maps to two different Reactome entries (stars, RAF/MAP kinase cascade).

The meta-graph is shown in Fig. 5.6b. This visualisation shows that the integrated data set consists of pathways (Pathway), reactions (Reaction) which are part of these pathways, metabolites (Compound) consumed or produced by the reactions, enzymes (Enzyme) catalysing the reactions and several combinations of proteins (Protein) and protein complexes (Protein complex) constituting the enzymes. The meta-graph provides the user with a useful high-level overview of the conceptual schema for this integrated data.

The last step to complete the molecular information fusion of the data presented in Fig. 5.6a would be to select the best equivalence relations and use the relation collapse data transformation to merge similar concept nodes together. To reduce the number of false-positive mappings, one would choose only such equivalence relations which are found by a combination of data integration methods (different edge colours) and at the same time carry a high confidence score (edge thickness) assigned by the data integration methods.

5.5 Discussion

Alternative methods for creating cross references (mappings) between information in different but related data sources have been presented. This is an essential component in the integration of data having different technical and semantic structures. Two realistic evaluation cases were used to quantify the performance of a range of different methods for mapping between the concept names and synonyms used in these databases. A quantitative evaluation of these methods shows that a graph-based algorithm (StructAlign) and mapping through synonyms can perform as well as using accession codes. In the particular application case of linking chemical compound names between pathway databases, the StructAlign and synonym-based algorithms outperformed the most direct mapping through accession codes by identifying more elements that were indirectly linked. Manual inspection of the false-positive mappings showed that both StructAlign and synonym mapping methods can be used where accession codes are not available to provide links between equivalent data source concepts. The combination of all three mapping methods yields the most complete projection between different data sources. This is an important result, because it is not always possible to find suitable accession code systems that provide the direct cross references between databases once you move outside the closely related data sources that deal with biological sequences and their functional annotations.

A similar approach to StructAlign called 'SubTree Match' has been described in [41] for aligning ontologies. This work extends this idea into a more general approach for data integration for biological networks and, furthermore, presents a formal evaluation in terms of precision and recall.

A particular challenge in this evaluation has been to identify suitable 'gold standard' data sets against which to assess the success of the algorithms developed. The results presented here are therefore not definitive, but represent the best quantitative comparison that could be achieved in the circumstances. Therefore, these results represent a pragmatic evaluation of the relative performance of the different approaches to concept name matching for data integration of life-science data sources.

Acknowledgements We would like to thank all current and previous contributors to the Ondex system (see www.ondex.org). The main part of this work has been carried out at Rothamsted Research. Rothamsted Research receives grant in aid from the Biotechnology and Biological Sciences Research Council (BBSRC). This work was supported by BBSRC SABR award BB/F006039/1 and TSB project TP 5082–33372. JT also would like to thank EMBL-EBI for allowing time to write this chapter.

WWW Link List (In Order of First Occurrence)

Name of resource	Brief description	WWW link
Ondex system	Data integration, visualisation and analysis framework for life-science data	http://www.ondex.org
BBSRC	Biotechnology and Biological Sciences Research Council in the United Kingdom	http://www.bbsrc.ac.uk
SRS	Sequence Retrieval System for biological data	http://www.instem.com/solutions/srs.html
PathSys	Graph-based system for creating a combined database of biological pathways, gene regulatory networks and protein interaction maps	http://biologicalnetworks.net/PathSys/
BN++ and BiNA	Biological data warehouse combined with biological network analyser	http://www.bina.unipax.info/
BioCAD	Integrated software for biosystem reverse engineering	http://biosoft.kaist.ac.kr/
Biozon	Unified biological knowledge resource with emphasis on protein and DNA characterisation and classification	http://www.biozon.org
STRING	Database of known and predicted protein interactions	http://string-db.org/
NeAT	Network Analysis Tools	http://rsat.bigre.ulb.ac.be/rsat/index_neat.html
OBO	Open Biomedical Ontologies Foundry	http://www.obofoundry.org
ENZYME (EC)	Nomenclature Committee of the International Union of Biochemistry and Molecular Biology	http://www.chem.qmul.ac.uk/iubmb/enzyme/

(continued)

(continued)

Name of resource	Brief description	WWW link
GO	The Gene Ontology	http://www.geneontology.org/
Reactome	Curated knowledgebase of biological pathways in humans	http://www.reactome.org
HumanCyc	Encyclopedia of Homo sapiens Genes and Metabolism	http://humancyc.org/
NCBI Taxonomy	Provides a taxonomy browser, taxonomy resources and other information	http://www.ncbi.nlm.nih.gov/taxonomy
GenBank	GenBank is the NIH genetic sequence database	http://www.ncbi.nlm.nih.gov/genbank/
Apache Lucene	Open source full-text indexing system	http://lucene.apache.org
BLAST	The Basic Local Alignment Search Tool	http://blast.ncbi.nlm.nih.gov
Decypher	Hardware accelerated sequence aligner	http://www.timelogic.com
PFAM	Large collection of protein families	http://pfam.sanger.ac.uk
Ec2go	Mapping file from EC to GO	http://www.geneontology.org/external2go/ec2go
ChEBI	Chemical Entities of Biological Interest	http://www.ebi.ac.uk/chebi
UniProt	Universal Protein Resource is a catalog of information on proteins	http://www.uniprot.org

References

1. Biotechnology and Biological Sciences Research Council (2007) Systems biology. http://www.bbsrc.ac.uk/publications/topic/systems-biology.aspx
2. Köhler J, Baumbach J, Taubert J, Specht M, Skusa A, Ruegg A, Rawlings C, Verrier P, Philippi S (2006) Graph-based analysis and visualization of experimental results with ONDEX. Bioinformatics 22(11):1383–1390
3. Gaylord M, Calley J, Qiang H, Su EW, Liao B (2006) A flexible integration and visualisation system for biomarker discovery. Appl Bioinformatics 5(4):219–223
4. Fischer HP (2005) Towards quantitative biology: integration of biological information to elucidate disease pathways and to guide drug discovery. Biotechnol Annu Rev 11:1–68
5. Köhler J, Rawlings C, Verrier P, Mitchell R, Skusa A, Ruegg A, Philippi S (2005) Linking experimental results, biological networks and sequence analysis methods using Ontologies and Generalised Data Structures. In Silico Biol 5(1):33–44
6. Taubert J, Hindle M, Lysenko A, Weile J, Köhler J, Rawlings CJ (2009) Linking life sciences data using graph-based mapping. Paper presented at the proceedings of the 6th international workshop on data integration in the life sciences, Manchester, UK
7. Taubert J, Sieren KP, Hindle M, Hoekman B, Winnenburg R, Philippi S, Rawlings C, Köhler J (2007) The OXL format for the exchange of integrated datasets. J Integr Bioinform 4(3):63
8. Taubert J (2011) ONDEX - a data integration framework for the life sciences. Bielefeld University, Bielefeld
9. Goble C, Stevens R (2008) State of the nation in data integration for bioinformatics. J Biomed Inform 41(5):687–693. doi:S1532-0464(08)00017-8 [pii] 10.1016/j.jbi.2008.01.008
10. Etzold T, Ulyanov A, Argos P (1996) SRS: information retrieval system for molecular biology data banks. Methods Enzymol 266:114–128

11. Baitaluk M, Qian X, Godbole S, Raval A, Ray A, Gupta A (2006) PathSys: integrating molecular interaction graphs for systems biology. BMC Bioinformatics 7:55
12. Küntzer J, Blum T, Gerasch A, Backes C, Hildebrandt A, Kaufmann M, Kohlbacher O, Lenhof H-P (2006) BN++ – a Biological Information System. J Integr Bioinform 3(2):34. doi:10.2390/biecoll-jib-2006-34
13. Smith B, Ceusters W, Klagges B, Kohler J, Kumar A, Lomax J, Mungall C, Neuhaus F, Rector AL, Rosse C (2005) Relations in biomedical ontologies. Genome Biol 6(5):R46
14. Lee D, Kim S, Kim Y (2007) BioCAD: an information fusion platform for bio-network inference and analysis. BMC Bioinformatics 8(Suppl 9):S2. doi:1471-2105-8-S9-S2 [pii] 10.1186/1471-2105-8-S9-S2
15. Birkland A, Yona G (2006) BIOZON: a system for unification, management and analysis of heterogeneous biological data. BMC Bioinformatics 7:70. doi:1471-2105-7-70 [pii] 10.1186/1471-2105-7-70
16. Jensen LJ, Kuhn M, Stark M, Chaffron S, Creevey C, Muller J, Doerks T, Julien P, Roth A, Simonovic M, Bork P, von Mering C (2009) STRING 8 – a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Res 37(Database issue):D412–D416. doi:gkn760 [pii] 10.1093/nar/gkn760
17. Pesch R, Lysenko A, Hindle M, Hassani-Pak K, Thiele R, Rawlings C, Köhler J, Taubert J (2008) Graph-based sequence annotation using a data integration approach. J Integr Bioinform 5(2):94. doi:10.2390/biecoll-jib-2008-94
18. Brohee S, Faust K, Lima-Mendez G, Sand O, Janky R, Vanderstocken G, Deville Y, van Helden J (2008) NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. Nucleic Acids Res 36(Web Server issue):W444–W451. doi:gkn336 [pii] 10.1093/nar/gkn336
19. Dwyer T, Rolletschek H, Schreiber F (2004) Representing experimental biological data in metabolic networks. Paper presented at the proceedings of the second conference on Asia-Pacific bioinformatics, vol 29, Dunedin, New Zealand
20. Jeong H, Mason SP, Barabasi AL, Oltvai ZN (2001) Lethality and centrality in protein networks. Nature 411(6833):41–42. doi:10.1038/35075138
21. Ogata H, Goto S, Fujibuchi W, Kanehisa M (1998) Computation with the KEGG pathway database. Biosystems 47(1–2):119–128
22. Zhu H, Cabrera RM, Wlodarczyk BJ, Bozinov D, Wang D, Schwartz RJ, Finnell RH (2007) Differentially expressed genes in embryonic cardiac tissues of mice lacking Folr1 gene activity. BMC Dev Biol 7:128. doi:10.1186/1471-213X-7-128
23. Gardner SP (2005) Ontologies and semantic data integration. Drug Discov Today 10(14):1001–1007. doi:S1359-6446(05)03504-X [pii] 10.1016/S1359-6446(05)03504-X
24. Bairoch A (2000) The ENZYME database in 2000. Nucleic Acids Res 28(1):304–305
25. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat Genet 25(1):25–29. doi:10.1038/75556
26. Jupe S, Akkerman JW, Soranzo N, Ouwehand WH (2012) Reactome – a curated knowledgebase of biological pathways: megakaryocytes and platelets. J Thromb Haemost. doi:10.1111/j.1538-7836.2012.04930.x
27. Caspi R, Altman T, Dreher K, Fulcher CA, Subhraveti P, Keseler IM, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Ong Q, Paley S, Pujar A, Shearer AG, Travers M, Weerasinghe D, Zhang P, Karp PD (2012) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Res 40(Database issue):D742–D753. doi:10.1093/nar/gkr1014
28. Smith B (2004) Beyond concepts: ontology as reality representation. In: Varzi A, Vieu L (eds) Proceedings of FOIS. IOS Press, Amsterdam

29. Schuemie MJ, Mons B, Weeber M, Kors JA (2007) Evaluation of techniques for increasing recall in a dictionary approach to gene and protein name identification. J Biomed Inform 40(3):316–324. doi:S1532-0464(06)00097-9 [pii] 10.1016/j.jbi.2006.09.002
30. Knuth D (1997) Section 6.2.3: Balanced trees. In: The art of computer programming, vol 3, Sorting and searching, 2nd edn. Addison-Wesley, Reading, 1998. ISBN 0-201-89685-0
31. Pearson WR (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. Methods Enzymol 183:63–98
32. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25(17):3389–3402. doi: 10.1093/nar/25.17.3389
33. Goutte C, Gaussier E (2005) A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In: Losada DE, Fernandez-Luna JM (eds) European Colloquium on IR Research (ECIR'05), 2005, Springer Berlin Heidelberg, pp 345–359. http://dx.doi.org/10.1007/978-3-540-31865-1_25
34. Stobbe MD, Houten SM, Jansen GA, van Kampen AH, Moerland PD (2011) Critical assessment of human metabolic pathway databases: a stepping stone for future integration. BMC Syst Biol 5:165. doi:10.1186/1752-0509-5-165
35. Degtyarenko K, de Matos P, Ennis M, Hastings J, Zbinden M, McNaught A, Alcantara R, Darsow M, Guedj M, Ashburner M (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res 36(Database issue):D344–D350. doi:10.1093/nar/gkm791
36. Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LS (2004) UniProt: the universal protein knowledgebase. Nucleic Acids Res 32(Database issue):D115–D119. doi:10.1093/nar/gkh13132/suppl_1/D115 [pii]
37. Bader G, Cary M (2005) BioPAX – biological pathways exchange language. BioPAX workgroup. http://www.biopax.org/release/biopax-level2-documentation.pdf
38. Baldwin TK, Winnenburg R, Urban M, Rawlings C, Köhler J, Hammond-Kosack KE (2006) PHI-base provides insights into generic and novel themes of pathogenicity. Mol Plant Microbe Interact 19(12):1451–1462
39. Winnenburg R, Baldwin TK, Urban M, Rawlings C, Köhler J, Hammond-Kosack KE (2006) PHI-base: a new database for pathogen host interactions. Nucleic Acids Res 34(Database issue):D459–D464
40. Köhler J, Munn K, Rüegg A, Skusa A, Smith B (2006) Quality control for terms and definitions in ontologies and taxonomies. BMC Bioinformatics 7:212
41. Zhang L, Gu J-G (2005) Ontology based semantic mapping architecture. In: Fourth international conference on machine learning and cybernetics. IEEE