TransID – the Flexible Identifier Mapping Service

Hendrik Mehlhorn[1] and Falk Schreiber[1,2]

[1]Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Gatersleben, Germany, {mehlhorn,schreibe@ipk-gatersleben.de}

[2]Institute of Computer Science, Martin Luther University Halle-Wittenberg, Halle, Germany

Summary

During the last few decades biological and biochemical research methodology has changed significantly and provides more and more diverse and complex data. The integration of this data is now a crucial issue in life science research. One basic building brick in representing a biologically meaningful relationship between two data records is a pair of identifiers. A computationally supported way of preparing custom lists of these bricks can facilitate different use cases in data integration.
We present TransID, a web-based and easy to use tool for the preparation of mapping tables. TransID unifies the usage of different data sources via the BridgeDB framework. The user is able to supply a list of identifiers via the browser and receives a custom mapping table via e-mail, containing relevant mappings as well as links to corresponding data sources. TransID is available by browser using the URL `http://apps.ipk-gatersleben.de/TransID/`.

1 Introduction

A large number of diverse technologies such as high throughput phenotyping facilities, modern wet lab techniques, and bioinformatics tools produce a continuously increasing amount of data. This information is represented worldwide in a network of hundreds of biological databases. Such databases are usually structured by a set of entries, the focus of which depends on the respective database, covering, inter alia, biological entities (e.g., genes, transcripts, proteins), annotations (e.g., protein functions, gene families, structures), references (e.g., papers, patents, sequences), biological networks (e.g., metabolic pathways, signal-transduction pathways, protein-protein interaction networks), and experiment data (e.g., abundance of proteins, transcript levels, phenotypic data). Identifiers are used to address the individual database entries.

Recent life science projects often necessitate the integration of knowledge from various life science domains, for example, proteomics and metabolomics. The biological entities of these life science domains are proteins and metabolites. Information about these biological entities is available from entries in different biological databases, here, for example, UniProt with UniProt accessions and Brenda with EC numbers. The easiest way of representing relations between different database entries are identifier mapping tables. For example, if we would like to integrate a protein-protein interaction network comprising UniProt accessions with a metabolic network comprising EC numbers, we need an identifier mapping table to represent the information regarding which protein codes for which enzyme, i.e. which UniProt accession corresponds

to which EC number. The compilation of identifier mapping tables for the integration of entries of different biological databases is the aim of the presented tool TransID.

Despite the necessity of unique identifiers for data integration approaches there are several problems regarding identifiers. In general, identifiers are only unique in the context of a data source such as a database or ontology. Identifiers are unique within each database, but are generally not unique without the database context, which makes specification of the database origin necessary. The specification of a certain database origin often depends on the cross-referencing database. For instance, the UniProt entry *Q8W413* is cross-referenced as *UniProtAccession Q8W413* by the *ArrayExpress* database, as *UniProtKB/Swiss-Prot Q8W413* by the *European Nucleotide Archive (ENA)*, and as *Q8W413 (Uniprot)* by the *KEGG* database. In addition, there are identifier format inconsistencies such as missing postfix specifications pointing to certain splice variants or sequence versions. Initiatives such as the MIRIAM registry [1] and identifiers.org [2] aim towards the creation of globally unique and persistent URIs to cope with these problems, but this approach has not yet been established in most biological databases.

However, identifiers are an important base for the integration of biological knowledge. Identifier mappings are widely used for the integration of related database entries. An identifier mapping consists of two identifiers, which may have several types of mapping relations such as protein-to-annotation, gene-to-experiment data, gene-to-transcript, and so on. A set of identifier mappings is usually represented as an identifier mapping table. The integration of biological knowledge by identifier mappings eases important tasks of life science researchers such as the unification and annotation of data sets, the linking to continuative knowledge for exploration issues, and the compilation of mapping tables.

There are several tools supporting identifier mappings such as AliasServer, BioDBnet, PICR, and Synergizer (see Table 1). These tools mainly focus on special species, databases, or are based on pre-computed mapping tables. The TransID identifier mapping service presented here is designed to encounter these drawbacks. TransID is based on an easily extensible set of data sources to allow rapid reaction to further demands. In addition, TransID aims at the compilation of identifier mappings which are synchronous with the embedded data sources. This is particularly relevant for constantly updated information such as functional annotations of biological entities. TransID is designed for researchers across the whole life science community. Thus our tool utilizes data sources which comprise knowledge about a huge variety of species, and is available via the web.

2 Architecture and Methods

TransID is based on an easily extensible set of data sources and aims at the compilation of up-to-date identifier mappings across all species. The key to achieve these demands is a composition of IDMappers. All IDMappers are managed by the BridgeDB framework [7]. The BridgeDB framework is an external library which unifies the handling and usage of IDMappers.

An IDMapper is a generic component which utilizes an arbitrary data source such as a database, a web site, or an additional tool in order to gain identifier mappings. Each IDMapper acts as an adapter which is able to cope with the conceptual and technical characteristics of the embedded data source. All IDMappers share the IDMapper interface in order to unify their usage.

Table 1: Published tools capable of mapping biological database identifiers.

ID mapping tool	Reference	Scope
AliasServer	[3]	Protein ID aliases
AnnBuilder	[4]	Annotation (genes)
BioDBnet	[5]	Database integration
BioMart	[6]	Data management
BridgeDB	[7]	ID mapping
CRONOS	[8]	ID mapping
DICT	[9]	ID mapping (genes)
Ensembl BioMart	[10]	Data exploration
g:Profiler	[11]	ID mapping (genes)
IDClight	[12]	ID mapping
IDconverter	[12]	ID mapping
MatchMiner	[13]	ID mapping
Onto-Translate	[14]	ID mapping
PICR	[15]	ID mapping (proteins)
PIR ID Mapping	[16]	ID mapping (proteins)
Resourcerer	[17]	Annotation (genes)
SOURCE	[18]	Data exploration
Synergizer	[19]	ID mapping
UniProt	[20]	ID mapping

The BridgeDB framework acts as an abstraction layer which executes all available IDMappers independently from the embedded data sources. TransID registers a set of IDMappers to the BridgeDB framework, with which it is possible to execute these IDMappers in an uniform and easy way. If a new data source is to be utilized by TransID, a single IDMapper can be implemented and instantly used together with all other IDMappers. This is the main advantage of the approach using IDMappers, which makes the tool highly flexible and easily extendible for further demands.

In order to understand the functionality of TransID in more detail, we describe the main concepts subsequently.

Identifier types: An identifier type denotes a set of unique identifiers from a single data source, i.e., a database or ontology. There is a large quantity of different identifier types. One data source can use one or more identifier types to address the database entries. For instance, the Uniprot database exclusively comprises the identifier type Uniprot accession and the KEGG database comprises various identifier types such as KO ID, KO Reaction, and KO Compound. Many identifier types exclusively address databases entries such as UniProt accessions, EMBL primary accessions, and CAZy enzyme families. Such identifier types are commonly used and are usually persistent. Some identifier types arise from ontologies or naming systems such as EC Numbers, Gene Ontology IDs, and Interpro accessions. These identifier types are part of a hierarchical structure and easily allow the inference of properties of the annotated biological entities. Names for the denomination of genes and proteins are widely used to address these

biological entities and can also be interpreted as identifiers. The naming of these terms is often functional or phenomenological, and partially ambiguous, if the species context is unknown.

We assign a set of properties to each identifier type. An important property is a regular expression, which we have defined in order to identify and validate putative identifiers. We enrich identifier types with properties such as a regular expression, identifier type aliases, a URL pattern leading to the web site, and the type of referenced entity such as experiment, gene, or pathway.

Identifiers: We represent identifiers by identifier objects, which we associate with an identifier type as well as an organism code. We group multiple strings representing the same identifier into one identifier object.

Identifier mappings: A pair of two identifiers with some kind of mapping relation is called an identifier mapping. The kind of mapping relation may be, inter alia, gene-to-protein, protein-to-publication, or protein-to-annotation. The inference of the kind of mapping relation from just two identifiers is generally not possible. For example, the mapping relation between an UniProt identifier and an EC number is most likely protein-to-annotation, but the mapping relation between two EMBL identifiers may, for instance, be gene-to-chromosome or gene-to-protein. Unfortunately, the information about the mapping relation between related identifiers is not present in most data sources. In addition, the question of the quality of an identifier mapping arises. This is especially the case for identifier mappings with a biological entity identifier such as a UniProt accession, a KEGG orthology ID, or an EMBL identifier and an annotation identifier such as an EC number, an Interpro identifier, or a GO identifier. There the annotation identifier may, for example, arise from a biological experiment or from electronic annotation. The uncertainty of both mapping relations and identifier mapping qualities weakens identifier mapping tools such as TransID. Thus, the result of these tools should be treated as candidate generation, and needs to be checked in each individual case.

IDMapper: An IDMapper is a generic adapter component for a special data source such as a database, a web site, or an additional tool. An IDMapper receives an identifier as input and delivers a set of identifiers as output. The combination of the input identifier with each output identifier yields an identifier mapping.

The interface of an IDMapper is defined by the BridgeDB framework and comprises, inter alia, a set of capabilities and a function to generate identifier mappings. The set of capabilities specifies the set of allowed identifier types of the input identifier, the set of possible identifier types of the output identifiers, and other properties. TransID is in principle able to map any identifier which type is supported by at least one IDMapper as an appropriate input identifier type. The identifier mapping function receives an input identifier as well as a set of desired target identifier types and fetches a set of output identifiers from the embedded data source. See the algorithm pseudocode (1) for a formal description of the identifier mapping operation executed by TransID. Depending on the kind of IDMappers used, the number of input identifiers, and the number of identifier mapping iterations, the identifier mapping operation may take several hours. This issue necessitates the delivery of the result of the identifier mapping operation via e-mail.

Currently we include 7 well-established data sources such as the European Nucleotide Archive (ENA) at the EBI, KEGG GENES from KEGG, the Protein Identifier Cross-Reference Service (PICR) at the EBI, and the UniProt Knowledgebase of the UniProt consortium.

3 Usage and Output

The user is able to execute the TransID identifier mapping service in three or four simple steps. In essence, the user needs to supply a set of identifiers (i), may adjust some identifier mapping parameters (ii) / (iii), and needs to submit an e-mail address (iv). The steps are described in more detail as follows.

(i) Identifier submission: In step (i), the user submits the identifiers of interest as input identifiers. The user is able to submit the set of identifiers using a text field, an Excel, or a text file.

(ii) Mode specification: In step (ii), the user selects the mode to proceed. With the default mode, the user jumps directly to step (iv). With the expert mode, the user continues with step (iii) in order to adjust expert settings. In addition, the user may specify optional settings. First, the user may specify the species of the input identifiers, which may increase the quantity of the resulting identifier mappings. Second, the user may narrow down the set of identifier types which are included in the result.

(iii) Expert adjustments (optional): In step (iii), the user may adjust expert settings. The user is able to initiate more exhaustive identifier mappings and to specify the result file format. TransID supports Excel files as well as tab delimited files as result file formats.

(iv) E-mail submission: In step (iv), the user submits an e-mail address. The result file will be sent to the user via e-mail because, depending on the user task, the identifier mapping process may last for several hours (depending on the quantity of the submitted identifiers in step (i) and the specified number of identifier mapping runs in step (iii)).

Figures 1 to 3 show an example session of TransID and the resulting mapping table. The result of the identifier mapping process is an identifier mapping table. An identifier mapping table consists of the submitted input identifiers (i), a heading (ii), and the obtained output identifiers (iii) in the body of the table. The submitted input identifiers (i) are listed in the first column. The heading (ii) is placed in the first row and comprises the identifier types of the output identifiers, one per column. The obtained identifier mappings (iii) in the body of the table are arranged by row according to the input identifiers and by column according to their identifier type. In case of multiple output identifiers of the same identifier type for one input identifier, the output identifiers are concatenated with the delimiter '; '.

The TransID user may choose to receive the identifier mapping table as a tab delimited file or as an excel file. Tab delimited files are very easily utilizable by many tools and custom

Data:
Set<IDMapper> M, // Used IDMappers
Set<Identifier> I, // Input identifiers
int n // Number of mapping iterations

Result:
Map<Identifier, Set<Identifier>> map // Identifier mappings

// fetch identifier mappings in n iterations
Set<Identifier> I_{new} = I;
Map<Identifier, Set<Identifier>> map = $\emptyset$;
for $1..n$ **do**
  Set<Identifier> $I_{cur} = I_{new}$;
  $I_{new} = \emptyset$;
  foreach *Identifier* $i \in I_{cur}$ **do**
    foreach *IDMapper* $m \in M$ **do**
      Set<Identifier> im = m.map(i);
      $map = map \cup (i, im)$;
      $I_{new} = I_{new} \cup im$;
    end
  end
end
// determine transitive identifier mappings in n iterations
Map<Identifier, Set<Identifier>> $map2$ = $\emptyset$;
foreach *Identifier* $i \in I$ **do**
  $map2 = map2 \cup (i, map.get(i))$;
end
for $2..n$ **do**
  foreach *Identifier* $i \in I$ **do**
    foreach *Identifier* $i2 \in map2.get(i)$ **do**
      $map2 = map2 \cup (i, map.get(i2))$;
    end
  end
end
return *map2*

Algorithm 1: Simplified pseudocode of the identifier mapping operation. The algorithm consists of two parts. In the first part of the algorithm, identifier mappings are computed in n identifier mapping iterations. In the first identifier mapping iteration, the input identifiers I are mapped with the given IDMappers M. In the second identifier mapping iteration, the result identifiers of the first identifier mapping iteration are mapped with the given IDMappers and so forth. In the second part of the algorithm, transitive identifier mappings are computed. First, the identifier mappings of the input identifiers are computed. Second, the identifier mappings are extended iteratively.

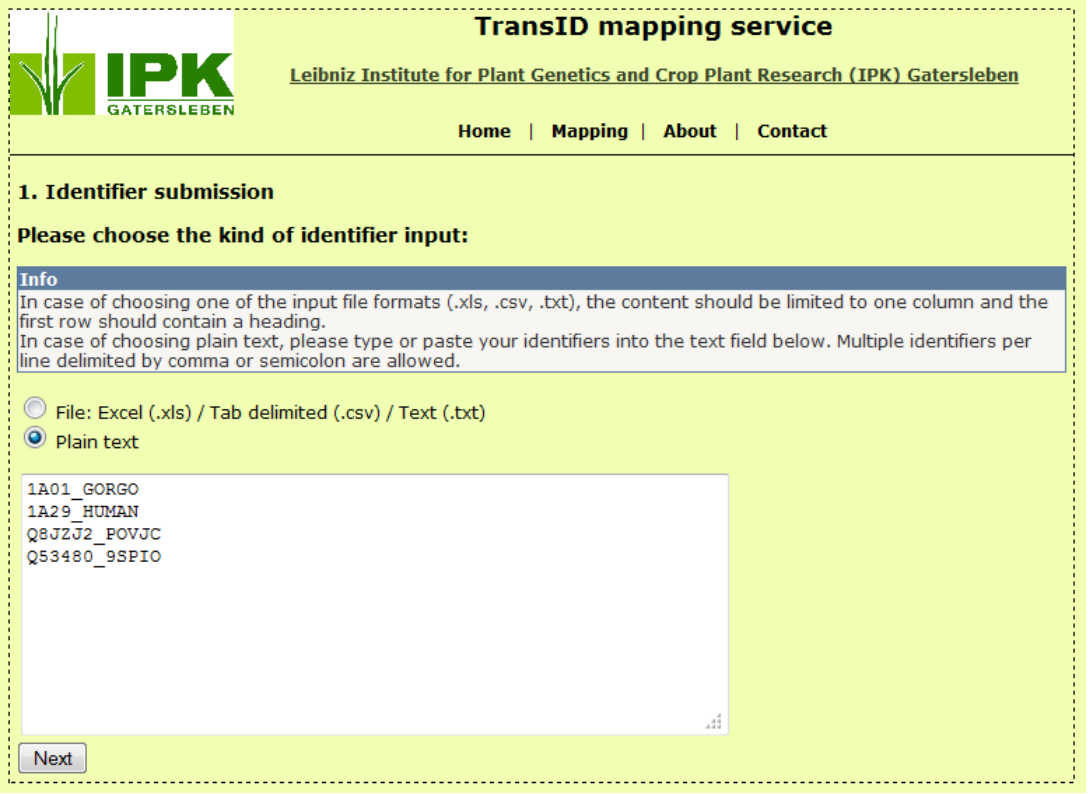

Figure 1: Startpage of TransID with initial search query.

applications. Excel files are very common in the life science community. In case of a tab delimited file, the columns are delimited by tabs and the rows are delimited by linebreaks. In case of an excel file, the data is organized in cells. Excel result files comprise two sheets. The first sheet comprises the identifier mapping table analogous to the tab delimited file. The second sheet comprises all the data of the first sheet enriched with hyperlinks to the referenced databases and output identifier web sites. The identifier types in the head are, if available, supported with hyperlinks to the website of the according data source. These hyperlinks give the user the opportunity to obtain information about the content and data origin of the data source. Each output identifier in the body of the table is supported, if available and if there is exactly one identifier in one cell, with a hyperlink to the web site of this identifier. These hyperlinks give the user the opportunity to instantly access the underlying database entry for further exploration issues.

4 Conclusion

We presented TransID, a tool for the preparation of identifier mapping tables. The tool aims at the compilation of up-to-date identifier mappings across all species for the broad life science community. It is available online. TransID is based on an easily extensible set of generic IDMappers which are managed by the framework BridgeDB for unified and easy usage. In order to receive the identifier mapping table in a result file, the user needs to supply a set of identifiers, may adjust some identifier mapping parameters, and needs to submit an e-mail address. Supported result file types are tab delimited files, which can be utilized by tools, as

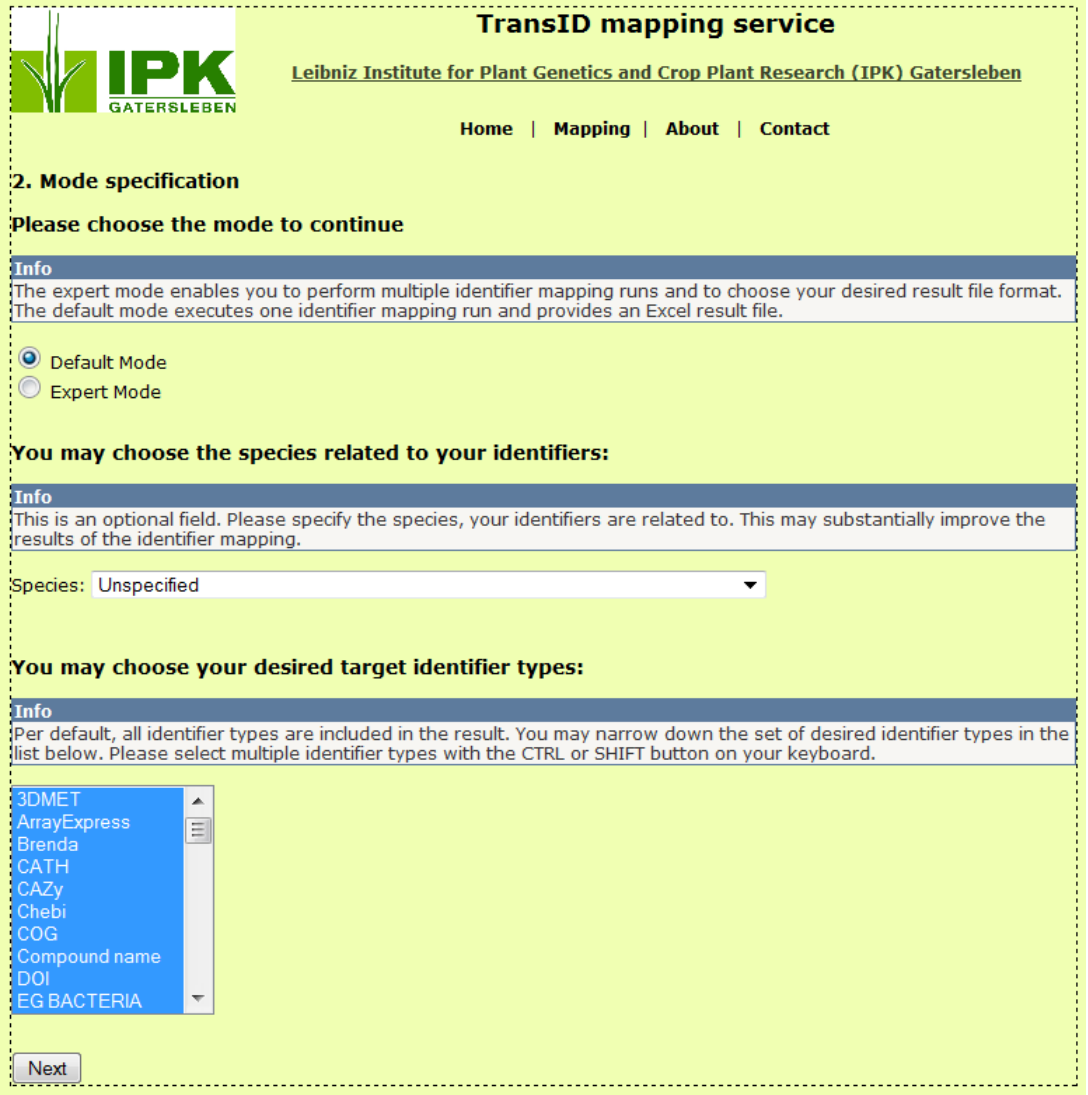

Figure 2: Mode specification page of TransID with initial search query.

Result information:

- Result file name (Excel) : MappingResult_55754.xls
- Number of mapping runs (1 or 2) : 1
- Submit time of the mapping : 2012-12-12-03:52:26
- Run time of the mapping process : 7 s, 27 ms
- Number of input identifiers : 4
- Proportion of mapped input IDs : 100 %
- Number of referenced databases : 26
- Number of identifier mappings : 107

Source IDs	DOI	EMBL	ENA	ENSEMBL	Gene ontology	H-Invitational	HOGENOM	HS
1A01_GORGO	10.1084/jem.174.	CAA42810; X60258	CAA42810; X6025		GO:0002474; GO:0006		P30375	
1A29_HUMAN		AAB47873; AAB53373; AAD02224; A		ENSG0000023		HIT000194904; H		
Q8JZJ2_POVJC		AB074581; BAB93045; BAB93045.1	AB074581; BAB93		GO:0003688; GO:0005			1tt
Q53480_9SPIO		AAB34223; AAB34223.1; S75873	AAB34223; S7587		GO:0005215			1b

Figure 3: Part of the resulting e-mail and the attached Excel file for the search query.

well as Excel files, which support hyperlinks to the referenced data.

References

[1] N. Le Novère, A. Finney, M. Hucka et al. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nature Biotechnology*, 23(12):1509–1515, 2005.

[2] N. Juty, N. Le Novère and C. Laibe. Identifiers.org and MIRIAM registry: community resources to provide persistent identification. *Nucleic Acids Research*, 40(D1):D580–D586, 2012.

[3] F. Iragne, A. Barré, N. Goffard and A. de Daruvar. AliasServer: a web server to handle multiple aliases used to refer to proteins. *Bioinformatics*, 20(14):2331–2332, 2004.

[4] J. Zhang, V. Carey and R. Gentleman. An extensible application for assembling annotation for genomic data. *Bioinformatics*, 19(1):155–156, 2003.

[5] U. Mudunuri, A. Che, M. Yi and R. M. Stephens. bioDBnet: the biological database network. *Bioinformatics*, 25(4):555–556, 2009.

[6] S. Haider, B. Ballester, D. Smedley, J. Zhang, P. Rice and A. Kasprzyk. BioMart Central Portal—unified access to biological data. *Nucleic Acids Research*, 37(S2):W23–W27, 2009.

[7] M. van Iersel, A. Pico, T. Kelder, J. Gao, I. Ho, K. Hanspers, B. Conklin and C. Evelo. The BridgeDb framework: standardized access to gene, protein and metabolite identifier mapping services. *BMC Bioinformatics*, 11(5), 2010.

[8] B. Waegele, I. Dunger-Kaltenbach, G. Fobo, C. Montrone, Mewes and A. Ruepp. CRONOS: the cross-reference navigation server. *Bioinformatics*, 25(1):141–143, 2009.

[9] D. W. Huang, B. T. Sherman, R. Stephens, M. W. Baseler, H. C. Lane and R. A. Lempicki. DAVID gene ID conversion tool. *Bioinformation*, 2(10):428–430, 2008.

[10] A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox and E. Birney. EnsMart: a generic system for fast and flexible access to biological data. *Genome Research*, 14(1):160–169, 2004.

[11] J. Reimand, M. Kull, H. Peterson, J. Hansen and J. Vilo. g:Profiler–a web-based toolset for functional profiling of gene lists from large-scale experiments. *Nucleic Acids Research*, 35(S2):W193–W200, 2007.

[12] A. Alibés, P. Yankilevich, A. Cañada and R. Díaz-Uriarte. IDconverter and IDClight: conversion and annotation of gene and protein IDs. *BMC Bioinformatics*, 8(9), 2007.

[13] K. J. Bussey, D. Kane, M. Sunshine, S. Narasimhan, S. Nishizuka, W. C. Reinhold, B. Zeeberg, W. Ajay and J. N. Weinstein. MatchMiner: a tool for batch navigation among gene and gene product identifiers. *Genome Biology*, 4(4), 2003.

[14] S. Drăghici, S. Sellamuthu and P. Khatri. Babel's tower revisited: a universal resource for cross-referencing across annotation databases. *Bioinformatics*, 22(23):2934–2939, 2006.

[15] R. G. Côté, P. Jones, L. Martens, S. Kerrien, F. Reisinger, Q. Lin, R. Leinonen, R. Apweiler and H. Hermjakob. The Protein Identifier Cross-Referencing (PICR) service: reconciling protein identifiers across multiple source databases. *BMC Bioinformatics*, 8(401), 2007.

[16] C. H. Wu, L.-S. L. Yeh, H. Huang et al. The Protein Information Resource. *Nucleic Acids Research*, 31(D1):345–347, 2003.

[17] J. Tsai, R. Sultana, Y. Lee, G. Pertea, S. Karamycheva, V. Antonescu, J. Cho, B. Parvizi, F. Cheung and J. Quackenbush. RESOURCERER: a database for annotating and linking microarray resources within and across species. *Genome Biology*, 2(11), 2001.

[18] M. Diehn, G. Sherlock, G. Binkley et al. SOURCE: a unified genomic resource of functional annotations, ontologies, and gene expression data. *Nucleic Acids Research*, 31(D1):219–223, 2003.

[19] G. F. Berriz and F. P. Roth. The Synergizer service for translating gene, protein and other biological identifiers. *Bioinformatics*, 24(19):2272–2273, 2008.

[20] R. Apweiler, A. Bairoch, C. H. Wu et al. Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 32(D1):115–119, 2004.