BMB
reports

IdBean: a Java GUI application for conversion of biological identifiers

Sanghyuk Lee[1,2], *Bumjin Kim*[1,2], *Hyeonjin Kim*[3], *Hookeun Lee*[4] *& Ungsik Yu*[4,*]

[1]Division of Molecular Life Sciences, Ewha Womans University, [2]Ewha Research Center for Systems Biology, Ewha Womans University, Seoul 120-750, [3]Department of Radiology, Seoul National University Hospital, Seoul 110-744, [4]Lee Gil Ya Cancer and Diabetes Institute, Gachon University of Medicine and Science, Incheon 406-840, Korea

We have developed a biologist-friendly, stand-alone Java GUI application, IdBean, for ID conversion. Our tool integrated most of the widely used ID conversion services that provide programmatic access. It is the first GUI ID conversion application that supports the direct merging as well as comparison of conversion results from multiple ID conversion services without manual effort. This tool will greatly help biologists who handle multiple ID types for the analyses of gene or gene product lists. By referring to multiple conversion services, the number of failed IDs can be reduced. By accessing ID conversion service online, it will potentially provide the most up-to-date conversion results. The application was developed in modular form; however, it can be re-packaged into plug-in form. For the development of a bioinformatics analysis tool, the module can be used as a built-in ID conversion component. It is available at http://neon.gachon.ac.kr/IdBean/. [BMB reports 2011; 44(2): 107-112]

INTRODUCTION

For the integration and analysis of data generated from "-omics" style experiments, multiple tools and services are typically employed. As these tools and services support their own sets of IDs for genes or gene products, ID conversion problems often occur across databases. While there are widely used IDs employed by Ensembl (1), Entrez gene (2), NCBI RefSeq (3), UniProt (4), and HGNC (5), there is no universal identifier. A number of tools such as the David gene ID conversion tool (6), CRONOS (7), IDconverter (8), Onto-translate (9), PICR (10), Synergizer (11), and BioMart (12) reportedly perform the conversion of IDs among different ID types. However, as each ID conversion service has its own focused domain of applications, such as supported organisms and ID types, the number of IDs capable of being converted by a single service is limited. By combining and merging the results of multiple ID conversion services, it is possible to cover a wider range of species and identifier types. Further, the conversion results can be cross-checked.

As bioinformatics analyses usually involve various kinds of ID types, there is a need for building ID conversion capability into a bioinformatics analysis tool rather than requiring users to perform ID conversions manually or with separate tools. Usually, researchers rely on external ID conversion services, as only a few tools perform built-in ID conversion. For example, there are quite a lot of Gene Ontology analysis tools (13), but only a few, if not at all, provide built-in ID conversion. Users of the tool must locate the ID conversion service and then copy/paste converted IDs resulting from the external service into the analysis tool. Further, the number of IDs that can be converted by a single service is limited, resulting in some failed IDs. Locating and then trying additional conversion services that convert those failed IDs is a large burden for a typical biologist and substantially reduces the usability of the tool.

Publically available ID conversion services are usually provided in the form of a web page or as a stand-alone application, which cannot easily be incorporated into other applications. These on line web services can potentially provide the broadest range of data. Further, with proper maintenance, these services are more up-to-date and centrally managed, thus requiring fewer resources from the end-user. Except for the few research groups that can actually afford the resources required to provide and maintain their own ID conversion services, developers and users must rely on external ID conversion services. Only a few services provide API access for batch conversion or incorporation into other applications.

BridgeDB (14) is a recent effort for unified ID conversion. It currently provides a Java API for ID conversion utilizing BioMart, CRONOS, PICR, and Synergizer. The BridgeDB API is targeted to software developers and provides a standardized interface layer through which bioinformatics tools can access various ID conversion and mapping services. It also relieves tool developers from the requirement of developing ID conversion access to each ID conversion service. Lastly, it pro-

*Corresponding author. Tel: 82-32-899-6052; Fax: 82-32-899-6039; E-mail: ungsik@gachon.ac.kr

DOI 10.5483/BMBRep.2011.44.2.107

Received 3 November 2010, Accepted 15 December 2010

Keywords: Bioinformatics, Data integration, ID conversion, Java GUI application, Omics data analysis

vides its own mapping service via the web in local database form. BridgeDB is utilized for applications such as Cytoscape (15) and PathVisio/WikiPathways (16) for ID conversion. Design and implementation of the user interface (UI) requires a significant amount of effort. As the UI is fully integrated into the Cytoscape and PathVisio applications, it is difficult to reuse the UI implementation to other applications.

Many bioinformatics GUI applications are developed in Java environment since Java provides portability across many different platforms. The Java Swing library is used for the GUI part in almost all cases. Most GUI applications have common features such as menus, toolbars, status bars, progress visualization, and so on. As each application has its own unique "plumbing", integrating and reusing an application for another purpose is almost impossible in most cases. For these and other client application features, a rich client platform (RCP) provides a framework, in which the features can be quickly and simply put together. A RCP frees developers from being concerned with tasks that have little to do with an application's business logic. As a result, developers can focus on their own problems and worry less about infrastructure. Developed code is also easier to reuse in other projects. Since programs developed as plug-ins for a RCP are modular, they can be reused and combined seamlessly with other independently developed plug-ins. Application development using a RCP will facilitate the merging and integration of individual modules into a large bioinformatics application framework that encompasses many areas. Recently, several bioinformatics applications such as BioClipse (17), Edinburgh Pathway Editor (18), Instant JChem (19), Quantitative Biology Tool (20), and ChipInspector (21) were developed by applying a RCP to the above-mentioned problems. Eclipse RCP (22) and NetBeans Platform (23) are the most popular RCP frameworks.

OSGi (24) technology is a dynamic module system based on Java. The OSGi Service Platform provides functionality to Java for software integration and facilitates modular development. OSGi technology also provides standardized primitives that allow applications to be constructed from small, reusable, and collaborative components. These components can be composed into an application and then deployed. Eclipse RCP is an OSGi framework implementation. Cytoscape is an open source Java bioinformatics software platform for visualizing molecular interaction networks and integrating these interactions with gene expression profiles and other state data. Although one of the key reasons for the wide acceptance of Cytoscape is its plug-in architecture, which allows the platform to be extended for developer's own need, it was announced that a future version will be based on OSGi for a standardized plug-in architecture. The plan is to separate Cytoscape functionality into clearly defined layers or modules. Further, NetBeans will support OSGi in its next release.

Adaptation of RCP and OSGi technology for the development of Java bioinformatics applications will become even more popular in the near future.

RESULTS AND DISCUSSION

We developed a biologist friendly, simple, easy, and powerful Java GUI ID conversion application, IdBean, that integrates most of the publically available ID conversion services. Currently, to merge and compare the results from multiple ID conversion services, manual efforts are required, as there are delicate differences among different services. By accounting for such differences, the application enables direct fetching, merging, and comparison of ID conversions. It can also convert multiple IDs from one source database type into multiple target database types. Finally, it can fetch and compare ID conversion results from different conversion service providers. Fig. 1 shows a screenshot of an ID conversion example.

There are two modes of ID conversion-MultiService and MultiTarget mode.

MultiService mode conversion

Multiservice mode is the main mode of the IdBean application. In MultiService mode, a researcher can obtain, compare, and merge the conversion results from multiple conversion service providers. Following specification of an organism, source, and target ID type combination, the ID conversion service providers available for the selected combination are shown. One can select multiple ID conversion services for merging and comparing the conversion results.

MultiTarget mode conversion

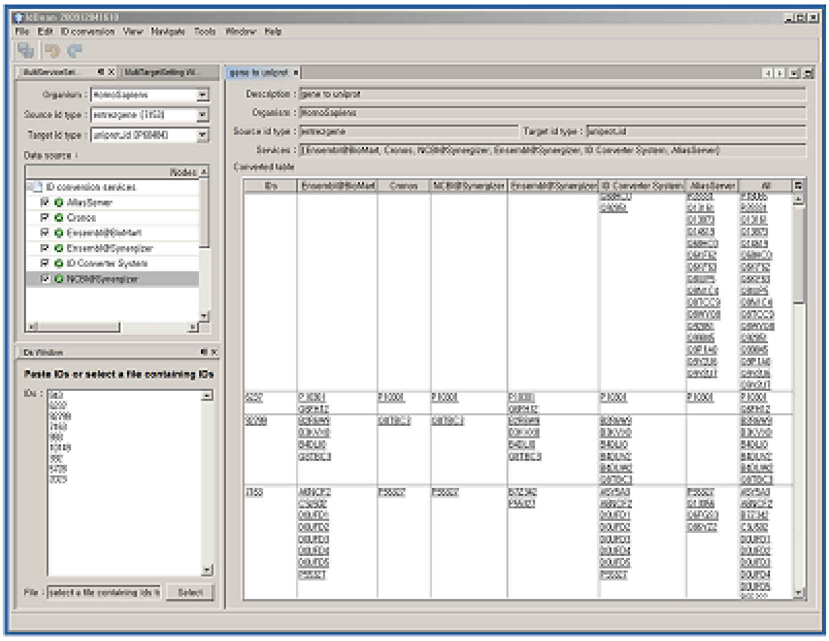
In MultiTarget mode, one can select a service provider and convert IDs into many types of available target IDs. The intended use of MultiTarget mode is to explore each individual conversion service provider. MultiTarget mode can be used to explore possible target ID types, identifying those supported for a given combination of an organism and source ID type. It also can be used to verify the correct ID format by trying out a well-known ID and then converting it into available ID types. Sometimes, different ID formats are used for a single ID type depending on the type of organism.

Conversion result tab

The conversion result table summarizes the results of conversion. Each column in the table shows the conversion result obtained using each conversion service with a corresponding organism, source ID type, target ID type, and ID.

Usability features

The ID conversion result table can be customized by column visibility control, row sorting, column resizing, and column order rearrangement. We also provide a direct webpage hyperlink for the IDs in the table. The hyperlink will help users acquire further information about each ID. Copy/paste/save functions are provided in the table with user-specifiable column and row selections. Other usability functions such as window position rearrangement, resizing, and full-screen mode are au-



Organism : HomoSapiens
Source id type : entrezgene (1153)
Target id type : uniprot_id (P10484)
Data source :

	Nodes
ID conversion services	
☑ AliasServer	
☑ Cronos	
☑ Ensembl@BioMart	
☑ Ensembl@Synergizer	
☑ ID Converter System	
☑ NCBI@Synergizer	

Paste IDs or select a file containing IDs
IDs :
File : select a file containing ids fi Select

Description : gene to uniprot
Organism : HomoSapiens
Source id type : entrezgene Target id type : uniprot_id
Services : [Ensembl@BioMart, Cronos, NCBI@Synergizer, Ensembl@Synergizer, ID Converter System, AliasServer]
Converted table

IDs	Ensembl@BioMart	Cronos	NCBI@Synergizer	Ensembl@Synergizer	ID Converter System	AliasServer	All

Fig. 1. Screenshot of an ID conversion example.

tomatically available in the NetBeans framework.

Comparison table

We also implemented a "comparison table", which allows for easy comparison of the results obtained using the two ID conversion service providers.

1 : 1 table

A 1 : 1 conversion table for a user-selected column is also provided, which is useful when there are many converted IDs for a single source ID. The converted IDs can then be copied directly into other applications. Chained conversion can be achieved by copying the resultant IDs, followed by reconversion using other ID conversion service providers.

One of the key aspects of IdBean is its modular nature. Currently, the application is a stand-alone Java application. However, it can be converted into a NetBeans platform plug-in with simple re-packaging. The converted plug-in can be used as an ID conversion service provider plug-in for other NetBeans platform-based bioinformatics applications without further modification. Applications developed in modular form for a RCP can work together and even enhance each other even if developed independently. Developing an application in modular form for a RCP should reduce integration and interoperability problems. Compared to API-based code reuse, IdBean provides another level of code reusability. In developing bioinformatics applications, the extensibility and maintainability of the application is of main concern.

MATERIALS AND METHODS

We integrated seven programmatically accessible ID conversion services, AliasServer (25), BioMart, CRONOS, IDconverter, ID converter system (26), PICR, and Synergizer, into one application. Table 1 shows a brief summary of the ID conversion services. We did not try to implement our own ID conversion services, as we do not have enough resources to maintain in-house conversion services. Instead, we decided to integrate external ID conversion services. We also tried to integrate as many convenience features as possible into the developed application in order to increase its usability for biologists. After being downloaded and installed, the program can be operated as a stand-alone application. For each ID conversion request, the results were fetched on the fly from each contributing service provider and summarized in a table. Fig. 2 shows the overall architecture of the developed application.

Conversion service access

Each conversion service provider exposes programmatic access in a specific form, such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Representational State Transfer (REST), Javascript Object Notation (JSON), and a pure web page. We implemented ID conversion service-dependent clients for each service according to each service's protocol. We used wsimport JDK command for WSDL processing, Jackson library (27) for JSON parsing, and httpclient Java library (28) for scraping the web page. We used the SwingX library (29) for table visualization. SwingX contains many powerful extensions to the Swing GUI toolkit, including

Table 1. An overviewof publically available ID conversion services

ID conversion service	Supported organisms	Access protocol	Web site
AliasServer	Many	SOAP	http://cbi.labri.fr/outils/alias/
BioMart	Many	REST, WSDL	http://www.biomart.org
CRONOS	Human, mouse, rat, dog, cattle, fruit fly	WSDL	http://mips.helmholtz-muenchen.de/genre/proj/cronos/index.html
ID converter system	Human	REST	http://biodb.jp/
ID converter	Human, mouse, rat	HTTP	http://idconverter.bioinfo.cnio.es/
PICR	Many	WSDL, REST	http://www.ebi.ac.uk/Tools/picr/
Synergizer	Many	JSON-RPC	http://llama.med.harvard.edu/synergizer/translate/

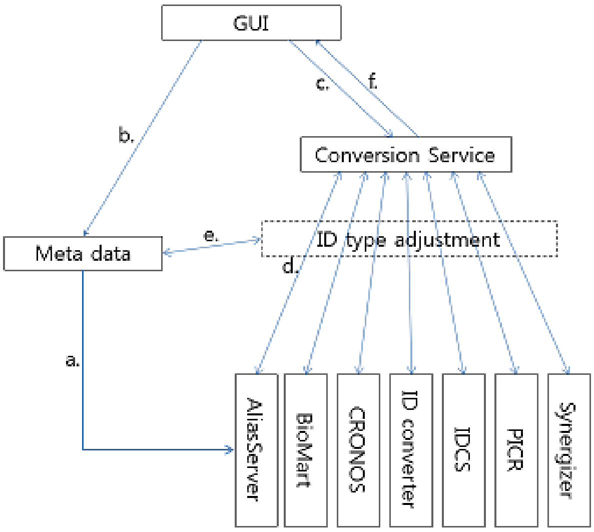


Fig. 2. Overall flow of the ID conversion process. (a) Collect metadata (supported organism, ID types) from each ID conversion service and adjust ID types. Metadata are embedded in the application. (b) Obtain adjusted metadata combinations. (c) Request ID conversion with the selected organism, source ID type, target ID type, and ID conversion services to use. To prevent UI freezing, request is processed in a new thread. (d) Query each ID conversion service provider and obtain the conversion results. A new thread is created and used for each individual ID conversion request. Further, progress status is monitored. (e) Use metadata to map an ID type between the adjusted ID type and the corresponding original ID type for each ID conversion service provider. (f) Return the merged conversion results.

new and enhanced components that provide functionalities that are commonly required by rich client applications. Other than these, we used standard Java.

For a simplified interface to a larger body of code, the GUI part, and to reduce dependencies of outside code on the inner workings of the ID conversion service access, we wrapped the collections of the ID conversion service clients into a single, well designed method.

Map < String, Set < String > > convert (Organism organism, IdType from, IdType to, List < String > ids)

The convert method also implemented threading in order to support concurrent ID conversion using multiple ID conversion services. Further, it implements progress monitoring for user feedback. For several of the ID conversion services, the number of IDs capable of being converted at a time is limited. The convert method accesses the client multiple times and gathers results.

When the application starts, it pings the hosts that provide an ID conversion service. If the response time is long or the server does not respond, the corresponding ID conversion service is disabled. For a service, which frequently takes a long time for conversion and results in a non-responsive progress bar, we imposed a timeout of 3 sec for each ID conversion request, and the corresponding conversion is marked as "Failed" in the conversion result table cell.

We developed our own implementation instead of applying BridgeDB API in an effort to support other ID conversion services that are not supported by the API. Also, their APIs did not fit our needs exactly.

Collecting metadata

Each service supports its own set of species, corresponding source, and target ID types. These metadata were pre-fetched from each service provider during the application development and embedded into the application in a form that enables species-wise selection of combinations of source and target ID types; no other data from service providers are stored in the application. New ID conversion services can be incorporated by including corresponding metadata into the code and adapting the ID conversion service access method.

ID type rearrangement

Initially, there were more than 250 different ID types, many of which are not familiar even to bioinformaticians. We tried to locate the origins of the ID types. In the case in which the origin of an ID type is identified, a brief description for that ID type is provided as a tooltip. A typical ID is also shown following the name of the ID type to hint at the proper ID format.

The naming of the ID types is inconsistent across the ID conversion service providers. For example, entrez gene is named as entrezgene, entrez_gene, geneid, gene_id, and locuslink_id by different ID conversion services. Even in the case in which the actual combination of an organism, source, and

Table 2. ID type rearrangement for UniProt ID types

ID conversion service	Original	Rearranged
AliasServer	Swissprot_name Swissprot_ac Trembl_id	Swissprot_name Uniprot_id Trembl_id
CRONOS	Uniprot_id	Swissprot_id
Ensembl@BioMart	Uniprot_swissprot_accession Uniprot_swissprot Uniprot_sptrembl	Swissprot_id Swissprot_name Trembl_id
Ensembl@Synergizer	Uniprot_accession	Uniprot_id
ID converter	Swissprot_ac_name	Swissprot_name
IDCS	Uniprot_id	Uniprot_id
NCBI@Synergizer	Uniprot	Swissprot_id
PICR	Swissprot Uniprot_sptrembl	Swissprot_id Trembl_id

target ID type is the same, there can still occur delicate differences in the resulting organism, source, and target ID type combination among the different services. Direct merging and comparison of the conversion results without manual effort is made infeasible by these minor differences in ID type naming. We checked the names of the supported ID types for all of the services and made rearrangements if necessary in order to enable direct merging and comparison.

For example, Table 2 shows the original UniProt ID type names and rearranged names for ID conversion services. In the table, uniprot_id includes both swissprot_id and tremble_id. For Ensembl@Synergizer and IDCS, the conversion results in and of themselves do not allow differentiation between swissprot_id and tremble_id. They are designated as uniprot_id. We also provide 'pseudo' uniprot_id for some ID conversion services for easy merging and comparison of the conversion results. For BioMart and PICR, they do provide separate ID types, swissprot_id and tremble_id, but do not provide the combined uniport_id ID type. The conversion result for 'pseudo' uniprot_id ID type for these cases is the same as the merged conversion results for swissprot_id and tremble_id. For CRONOS and NCBI@Synergizer, the conversion results for pseudo uniprot_id are the same for the respective swissprot_ID type conversion results.

Use of NetBeans platform

Most available bioinformatics Java GUI applications were developed by using Swing. We plan to develop bioinformatics applications for various areas by adapting and integrating existing applications as needed. The extensibility and maintainability of the application is one of main concern, which is why we adapted the rich client framework. We chose the NetBeans platform over Eclipse RCP, as NetBeans platform is a pure Swing-based framework. This allows the developed application to be operable as long as the Java Runtime environment is available. On the other hand, Eclipse RCP needs a machine and/or an OS dependent library.

As the IdBean application was developed as a NetBeans Platform application, it should work together with other plug-in applications without extra integration efforts. The software architecture is much clearer and more reliable than if the application had been developed without the NetBeans framework. Further, IdBean has been divided into a set of modules, each communicating through well-defined APIs. Our reliance on standard technologies such as Java, Swing, and NetBeans Platform should ensure wider accessibility, stability, and extendibility.

Acknowledgements

Funding: This work was supported by a grant (2007-03983) from the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology of the Korean government.

REFERENCES

1. Hubbard, T. J., Aken, B. L., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P., Clarke, L., Coates, G., Fairley, S., Fitzgerald, S., Fernandez-Banet, J., Gordon, L., Graf, S., Haider, S., Hammond, M., Holland, R., Howe, K., Jenkinson, A., Johnson, N., Kahari, A., Keefe, D., Keenan, S., Kinsella, R., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Megy, K., Meidl, P., Overduin, B., Parker, A., Pritchard, B., Rios, D., Schuster, M., Slater, G., Smedley, D., Spooner, W., Spudich, G., Trevanion, S., Vilella, A., Vogel, J., White, S., Wilder, S., Zadissa, A., Birney, E., Cunningham, F., Curwen, V., Durbin, R., Fernandez-Suarez, X. M., Herrero, J., Kasprzyk, A., Proctor, G., Smith, J., Searle, S. and Flicek, P. (2009) Ensembl 2009. *Nucleic. Acids. Res.* **37**, D690-697.
2. Maglott, D., Ostell, J., Pruitt, K. D. and Tatusova, T. (2007) Entrez Gene: gene-centered information at NCBI. *Nucleic. Acids. Res.* **35**, D26-31.
3. Pruitt, K .D., Tatusova, T., Klimke, W. and Maglott, D. R. (2009) NCBI Reference Sequences: current status, policy and new initiatives. *Nucleic. Acids. Res.* **37**, D32-36.
4. The UniProt Consortium (2009) The Universal Protein Resource (UniProt) 2009. *Nucleic. Acids. Res.* **37**, D169-174.
5. Bruford, E. A., Lush, M. J., Wright, M. W., Sneddon, T. P., Povey, S. and Birney, E. (2008) The HGNC Database in 2008: a resource for the human genome. *Nucleic. Acids. Res.* **36**, D445-448.
6. Huang, da W., Sherman, B. T., Stephens, R., Baseler, M. W., Lane, H. C. and Lempicki, R. A. (2008) DAVID gene ID conversion tool. *Bioinformation* **2**, 428-430.
7. Waegele, B., Dunger-Kaltenbach, I., Fobo, G., Montrone, C., Mewes, H. W. and Ruepp, A. (2009) CRONOS: the cross-reference navigation server. *Bioinformatics* **25**, 141-143.
8. Alibés, A., Yankilevich, P., Cañada, A. and Díaz-Uriarte, R. (2007) IDconverter and IDClight: conversion and annotation of gene and protein Ids. *BMC Bioinformatics* **8**, 9.
9. Khatri, P., Desai, V., Tarca, A. L., Sellamuthu, S., Wildman, D. E., Romero, R. and Draghici, S. (2006) New

Onto-Tools: Promoter-Express, nsSNPCounter and Onto-Translate. *Nucleic. Acids. Res.* **34**, W626-W631.

10. Cote, R. G., Jones, P., Martens, L., Kerrien, S., Reisinger, F., Lin, Q., Leinonen, R., Apweiler, R. and Hermjakob, H. (2007) The Protein Identifier Cross-Referencing (PICR) service: reconciling protein identifiers across multiple source databases. *BMC Bioinformatics* **8**, 401.
11. Berriz, G. F. and Roth, F. P. (2008) The Synergizer service for translating gene, protein and other biological identifiers. *Bioinformatics* **24**, 2272-2273.
12. Smedley, D., Haider, S., Ballester, B., Holland, R., London, D., Thorisson, G. and Kasprzyk, A. (2009) BioMart-biological queries made easy. *BMC Genomics* **10**, 22.
13. Gene Ontology tools [http://www.geneontology.org/GO.tools.shtml].
14. van Iersel, M. P., Pico, A. R., Kelder, T., Gao, J., Ho, I., Hanspers, K., Conklin, B. R. and Evelo, C. T. (2010) The BridgeDB framework: standardized access to gene, protein and metabolite identifier mapping services. *BMC Bioinformatics* **11**, 5.
15. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B. and Ideker, T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research* **13**, 2498-2504.
16. van Iersel, M. P., Kelder, T., Pico, A. R., Hanspers, K., Coort, S., Conklin, B. R. and Evelo, C. (2008) Presenting and exploring biological pathways with PathVisio. *BMC Bioinformatics* **9**, 399.
17. Spjuth, O., Alvarsson, J., Berg, A., Eklund, M., Kuhn, S., Mäsak, C., Torrance, G., Wagener, J., Willighagen, E. L., Steinbeck, C. and Wikberg, J. E. (2009) Bioclipse 2: a scriptable integration platform for the life sciences. *BMC Bioinformatics* **10**, 397.
18. Edinburgh Pathway Editor [http://www.bioinformatics.ed.ac.uk/epe/].
19. Instant JChem [http://www.chemaxon.com/products/instant-jchem/].
20. Quantitative Biology Tool [http://www.semanticbits.com/what_we_do/software_solutions/qbt.php].
21. ChipInspector [http://www.genomatix.de/chipinspector.html].
22. Eclipse [http://www.eclipse.org].
23. NetBeans [http://www.netbeans.org].
24. OSGi [http://www.osgi.org].
25. Iragne, F., Barre, A., Goffard, N. and De Daruvar, A. (2004) AliasServer: a web server to handle multiple aliases used to refer to proteins. *Bioinformatics* **20**, 2331-2332.
26. Imanishi, T. and Nakaoka, H. (2009) Hyperlink Management System and ID Converter System: enabling maintenance-free hyperlinks among major biological databases. *Nucleic. Acids. Res.* **37**, W17-W22.
27. Jackson library [http://jackson.codehaus.org].
28. httpclient Java library [http://hc.apache.org].
29. SwingX library [http://swinglabs.org].