Swainston *et al. J Cheminform (2016) 8:11*
DOI 10.1186/s13321-016-0123-9

Journal of Cheminformatics
a SpringerOpen Journal

SOFTWARE

Open Access

libChEBI: an API for accessing the ChEBI database

Neil Swainston[1,2*], Janna Hastings[2], Adriano Dekker[2], Venkatesh Muthukrishnan[2], John May[2,3], Christoph Steinbeck[2] and Pedro Mendes[1,4,5]

Abstract

Background: ChEBI is a database and ontology of chemical entities of biological interest. It is widely used as a source of identifiers to facilitate unambiguous reference to chemical entities within biological models, databases, ontologies and literature. ChEBI contains a wealth of chemical data, covering over 46,500 distinct chemical entities, and related data such as chemical formula, charge, molecular mass, structure, synonyms and links to external databases. Furthermore, ChEBI is an ontology, and thus provides meaningful links between chemical entities. Unlike many other resources, ChEBI is fully human-curated, providing a reliable, non-redundant collection of chemical entities and related data. While ChEBI is supported by a web service for programmatic access and a number of download files, it does not have an API library to facilitate the use of ChEBI and its data in cheminformatics software.

Results: To provide this missing functionality, libChEBI, a comprehensive API library for accessing ChEBI data, is introduced. libChEBI is available in Java, Python and MATLAB versions from http://github.com/libChEBI, and provides full programmatic access to all data held within the ChEBI database through a simple and documented API. libChEBI is reliant upon the (automated) download and regular update of flat files that are held locally. As such, libChEBI can be embedded in both on- and off-line software applications.

Conclusions: libChEBI allows better support of ChEBI and its data in the development of new cheminformatics software. Covering three key programming languages, it allows for the entirety of the ChEBI database to be accessed easily and quickly through a simple API. All code is open access and freely available.

Keywords: Cheminformatics, Database, API, Library, Java, Python, MATLAB, ChEBI

Background

ChEBI is a database and ontology of chemical entities of biological interest [1–3]. With a focus on small molecules, it contains names, chemical structures, synonyms, database cross-references, links to relevant literature, and classifications based on structural features and biological activity. ChEBI has been used as a resource for identifiers for the systematic annotation of chemicals in life science contexts, for example in metabolic models [4–6] and protein [7] and interaction databases [8]. It has also been used as a dictionary of names for chemical text mining [9] and as a source of semantic types for the growing chemical Semantic Web [10, 11].

ChEBI is made available via several access routes. Firstly, it is supported by a website with complex searching and browsing functionality (http://www.ebi.ac.uk/chebi/). Secondly, the data are available for download in full in several different download formats including relational database table data, flat files, the cheminformatics SDfile (structure-data file) format, and ontology formats OBO and OWL. Finally, there is a SOAP-based web service with several access methods that allow search and retrieval of any of the ChEBI content. However, for applications which make a heavy use of ChEBI content, the iterative search-and-retrieve strategy offered by the ChEBI web service may yield insufficient performance, while in order to implement applications which harness ChEBI's content from many of the different download formats, it is necessary to become familiar with the

*Correspondence: neil.swainston@manchester.ac.uk
[2] European Bioinformatics Institute, Hinxton, Cambridge CB10 1SD, UK
Full list of author information is available at the end of the article

ChEBI data model. ChEBI is extensively human-curated and, as such, duplicate entries are merged, ensuring that the database contains no redundant entries. Deprecated entries *are* retained but linked to a parent entry, which maintains integrity of the resource and avoids dropped ids and broken links. Due to this added layer of (necessary) complexity, it is inefficient for individual programming efforts to address this issue of id mapping and deprecated entries in repeated independent efforts. libChEBI hides this from the user, ensuring seamless access to all data within the repository.

To facilitate the integration of ChEBI into new and existing software tools, libChEBI, a shared, freely available application programming interface (API) library has been developed. This simple API hides complexity and intricacies of the ChEBI data model, providing a simple interface for accessing ChEBI data. libChEBI has been developed in a generic fashion and will be applicable to any software developers who use (bio)chemical data.

Implementation

libChEBI provides a simple interface to the contents of the ChEBI database, built on top of the existing flat file download facility. Flat files are downloaded, unpacked and parsed as required, providing a simple API that is described below. As the flat files are updated on a monthly basis, libChEBI ensures that the most up-to-date version is automatically downloaded. This is the only online requirement of the library, and as such, once the flat files are downloaded, libChEBI can be used offline without any requirement for a connection to the ChEBI database (see Fig. 1). libChEBI provides access to the entire contents of the ChEBI database while removing the need for the user to be familiar with the ChEBI flat file format, or the internal secondary identifier mapping system. Regarding memory issues, the current size (January 2016 release) of all of the unzipped flat files that are parsed is 1.2 GB. However, only a subset of these files (up to 66 MB) is held in memory at any time. Files related to structures and references are not held in memory, as these clearly would cause an excessive memory burden. The library is accessible through Java, Python and MATLAB APIs, which are described in more detail elsewhere (see Fig. 2; Additional file 1: libChEBI API), with examples of use provided below.

Results and discussion

Java

The Java public interface consists of a number of classes, of which `uk.ac.manchester.ChebiEntity` is the primary entry point. The `ChebiEntity` constructor takes a `String`, representing the ChEBI id, as a parameter. `ChebiEntity` then provides a number of methods,

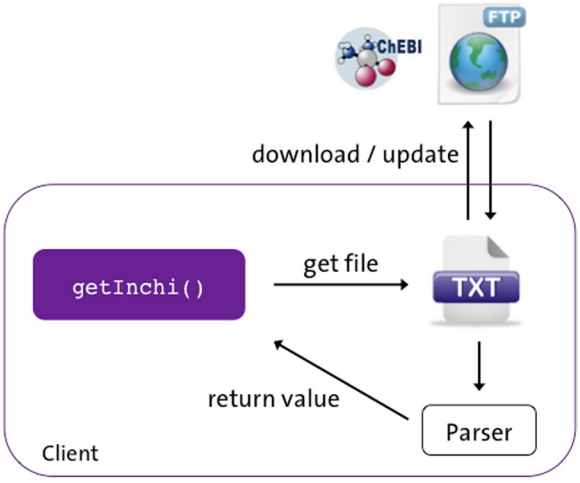

Fig. 1 Basic schema of the libChEBI architecture. Upon calling a libChEBI method (e.g. `getInchi()`), a check is performed to determine whether the current flat text file is available locally. If not, the file is retrieved from the ChEBI FTP site. The flat file is then parsed, and the requested value returned

providing access to the properties of the ChEBI entity. Example code, illustrating the retrieval of names synonyms for D-glucose, is shown in Box 1.

Python

Like Java, Python is supported by a similar interface with `libchebi.ChebiEntity` being the primary entry point. Example code is given in Box 2.

Matlab

MATLAB support is provided by exploiting the existing facility for bringing Java classes into the MATLAB Workspace. A simple wrapper method, `getChebiEntity(id)` is provided, which returns a Java `uk.ac.manchester.ChebiEntity` object. All methods of this object, such as `getName()`, are then callable from the MATLAB Workspace (see Box 3).

Software application areas utilising ChEBI

In recent years, ChEBI has become increasingly utilised by the systems biology community as a repository of persistent, unambiguous identifiers with which to semantically annotate models. Standardisation of the syntax of systems biology models was addressed with the introduction of the Systems Biology Markup Language (SBML) format over 10 years ago [12]. However, it was recognised that the semantics embedded within these models were non-standardised, with most models containing ambiguous metabolite names and identifiers. Such ambiguity made the interpretation and comparison of such models difficult [13], and their automated parameterisation with

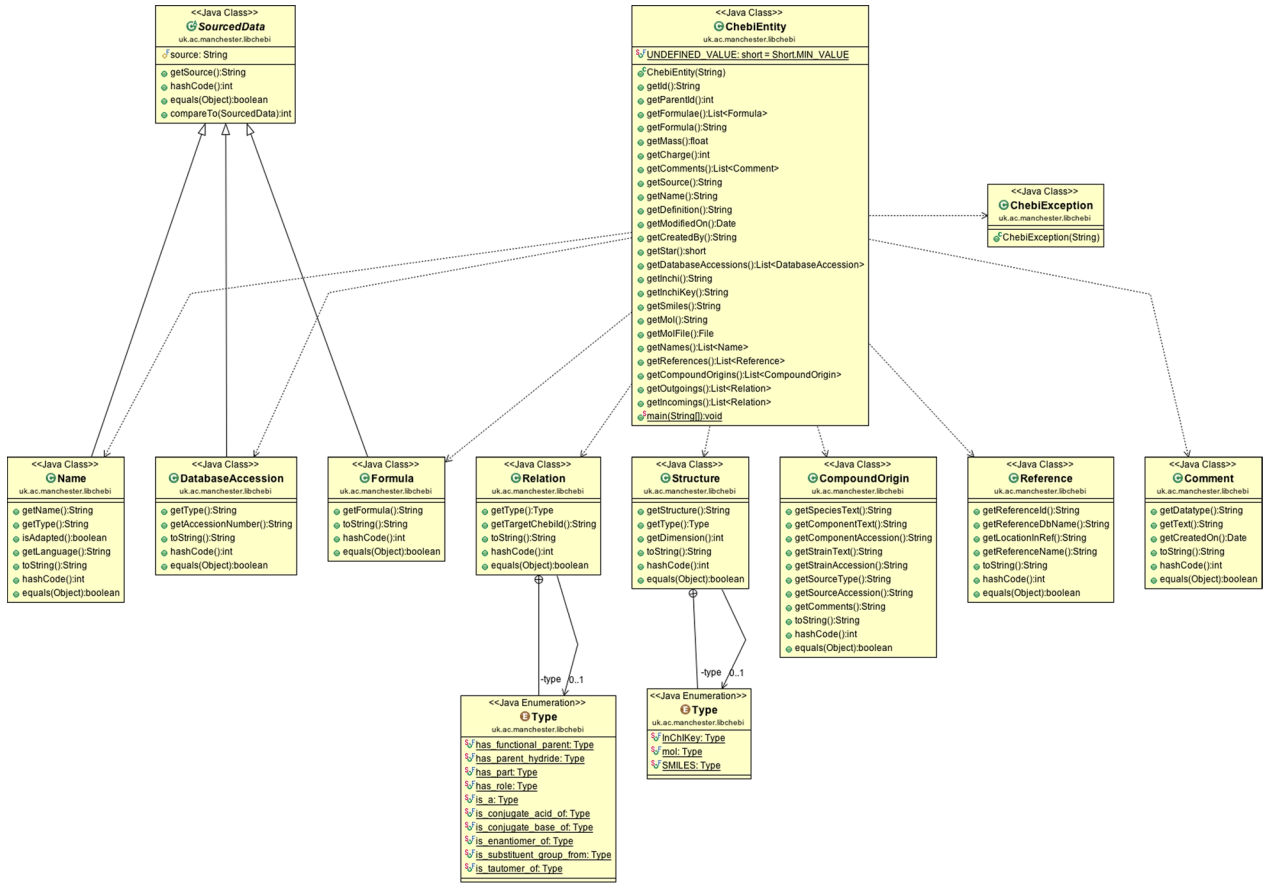

Fig. 2 UML class diagram of libChEBIj. All public and protected classes, fields and methods are shown. The Python version, libChEBIpy, follows the same class structure, and these classes are accessible in MATLAB through use of libChEBIm

Box 1 Example libChEBI Java code, illustrating the instantiation of a `ChebiEntity`, a call to the `getNames()` method, access of the returned `Names` objects, and an example of its resulting output

```
ChebiEntity chebiEntity = new ChebiEntity( "CHEBI:17634" );

for( Name name : chebiEntity.getNames() )
{
   System.out.println( name.getName() + "\t" + name.getSource() + "\t" + name.getLanguage() );
}
```

```
D-glucose       IUPAC   en
D-gluco-hexose IUPAC   en
dextrose        NIST Chemistry WebBook en
D(+)-glucose    ChemIDplus      en
grape sugar     ChemIDplus      en
D-(+)-glucose   NIST Chemistry WebBook en
Traubenzucker   ChemIDplus      de
```

experimental data impossible [14, 15]. This issue was partially solved with the introduction of the Minimum Information Requested In the Annotation of Models (MIRIAM) guidelines [16], which provided a facility for annotating model terms with standardised identifiers, such as those provided by ChEBI. Amongst the

Box 2 Example libChEBI Python code, showing the instantiation of a `ChebiEntity`, a call to `get_name()`, `get_outgoings()` and the calling of a number of methods of the returned `Relation` objects

```
chebi_entity = ChebiEntity('CHEBI:15903')

print chebi_entity.get_name()

for outgoing in chebi_entity.get_outgoings():
        target_chebi_entity = ChebiEntity(outgoing.get_target_chebi_id())
        print outgoing.get_type() + '\t' + target_chebi_entity.get_name()

beta-D-glucose
is_enantiomer_of        beta-L-glucose
has_role        epitope
is_a    D-glucopyranose
```

Box 3 Example libChEBI MATLAB code, illustrating the instantiation of a `ChebiEntity`, and calls to the `getName()` and `getCharge()` methods

```
>> chebiEntity = getChebiEntity('CHEBI:30616');
>> chebiEntity.getName()

ans =

ATP(4-)

>> chebiEntity.getCharge()

ans =

    -4
```

first large-scale projects to apply these guidelines was that of the Yeast Consensus Model [17, 18], an international collaborative effort to develop a consensus metabolic reconstruction of *Saccharomyces cerevisiae*. This was followed by a similar effort for human metabolism [19, 20], resulting in comprehensive representations of cellular metabolism in which most cellular components are unambiguously identified, a majority of which with ChEBI identifiers.

The use of semantic annotations within models goes beyond just acting as a means of unambiguously identifying components. By providing identifiers linking to publicly available databases, the *content* of these databases can be accessed and used in model refinement, checking and expansion. For example, annotating a model with ChEBI identifiers allows chemical formulae, charge and structural information to be accessed automatically [21]. Such data can then be exploited in model building and checking pipelines such as the SuBliMinaL Toolbox [22], which include automated methods for metabolite charge state determination, reaction balancing and model merging. Application of these methods has led to the automated generation of genome-scale metabolic models of cellular metabolism from over 2000 species [23]. Keeping these models up to date requires automated access to the latest version of ChEBI, which until now required the development and maintenance of custom scripts by each development group, however, such automation is now seamlessly handled through libChEBI.

Although conceived primarily in reference to the requirements within systems biology, libChEBI has been designed in a generic fashion allowing applicability to a range of software applications that utilise chemical data. For example, as the number of annotated metabolites grows, ChEBI is increasingly being used as a reference for metabolite identification and analysis pipelines in metabolomics experiments [24, 25]. Such pipelines currently rely on custom scripts harnessing the SOAP web service, but will now be facilitated. Similarly, within the drug discovery pipeline ChEBI has been used as one of several systems within which chemicals can be classified or grouped in order for patterns to be evaluated in large-scale high-throughput data [26]. As secrecy is important in the drug discovery context, use of the downloadable

files from ChEBI is preferred in this context rather than web service queries. However, use of the download files suffers from the complexity of the underlying data model as described above, thus, provision of a targeted library will ease adoption. The reliable human curation and extensive collection of chemical synonyms that are present in the database have resulted in ChEBI becoming a source in text mining applications [27]. ChEBI is also used programmatically within the Bioclipse software platform [28] in diverse contexts including cheminformatics and chemical toxicology. libChEBI has been designed to support both this diverse range of applications and the development of future applications that exploit the contents of the ChEBI database.

Conclusions

libChEBI is introduced to provide simple programmatic access to the contents of the ChEBI database, and has been designed specifically for developers who wish to incorporate ChEBI data into their software. Future developments may include the support of additional programming languages and implementation of a search facility. However, as a community resource, the direction in which libChEBI develops will be determined by requests from the user community, and as such feedback on this resource is welcomed and encouraged.

Availability and requirements

Project name: libChEBI
Project home page: https://github.com/libChEBI
Operating system(s): Platform independent
Programming language: Java, Python, MATLAB
Other requirements: Java 1.7 or higher, Python 2.7 or higher, MATLAB 2013a or higher
License: MIT.

Additional file

Additional file 1. libChEBI API describes the API for each of the supported languages.

Authors' contributions

NS conceived the idea, design and coded the software, and led the writing of the manuscript. JH helped write the manuscript. AD and VM provided support with the ChEBI data model and download files. PM and CS contributed to the development of the idea and to seeking funding for the project. All authors read and approved the manuscript.

Author details

[1] Manchester Centre for Synthetic Biology of Fine and Specialty Chemicals (SYNBIOCHEM), Manchester Institute of Biotechnology, University of Manchester, Manchester M1 7DN, UK. [2] European Bioinformatics Institute, Hinxton, Cambridge CB10 1SD, UK. [3] NextMove Software Ltd., Innovation Centre, Science Park, Milton Road, Cambridge CB4 0EY, UK. [4] School of Computer Science, University of Manchester, Manchester M13 9PL, UK. [5] Center for Quantitative Medicine, UConn Health, Farmington, CT 06030, USA.

Acknowledgements

All authors acknowledge the funding from the BBSRC under Grant BB/K019783/1, "Continued development of ChEBI towards better usability for the systems biology and metabolic modelling community". NS and PM also thank the BBSRC for funding under Grants BB/M017702/1, "Centre for synthetic biology of fine and speciality chemicals", and BB/M006891/1, "Enriching Metabolic PATHwaY models with evidence from the literature (EMPATHY)".

Competing interests

The authors declare that they have no competing interests.

Received: 19 October 2015 Accepted: 16 February 2016
Published online: 01 March 2016

References

1. Degtyarenko K, de Matos P, Ennis M, Hastings J, Zbinden M, McNaught A, Alcántara R, Darsow M, Guedj M, Ashburner M (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucl Acids Res 36:D344–D350
2. Hastings J, de Matos P, Dekker A, Ennis M, Harsha B, Kale N, Muthukrishnan V, Owen G, Turner S, Williams M, Steinbeck C (2013) The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. Nucl Acids Res 41:D456–D463
3. Hastings J, Owen G, Dekker A, Ennis M, Kale N, Muthukrishnan V, Turner S, Swainston N, Mendes P, Steinbeck C (2016) ChEBI in 2016: improved services and an expanding collection of metabolites. Nucleic Acids Res 44:D1214–D1219
4. Smallbone K, Messiha HL, Carroll KM, Winder CL, Malys N, Dunn WB, Murabito E, Swainston N, Dada JO, Khan F, Pir P, Simeonidis E, Spasić I, Wishart J, Weichart D, Hayes NW, Jameson D, Broomhead DS, Oliver SG, Gaskell SJ, McCarthy JE, Paton NW, Westerhoff HV, Kell DB, Mendes P (2013) A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. FEBS Lett 587:2832–2841
5. Messiha HL, Kent E, Malys N, Carroll KM, Swainston N, Mendes P, Smallbone K (2014) Enzyme characterisation and kinetic modelling of the pentose phosphate pathway in yeast. PeerJ PrePrints 2:e146v4
6. Le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, Snoep JL, Hucka M (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucl Acids Res 34:D689–D691
7. UniProt Consortium (2015) UniProt: a hub for protein information. Nucl Acids Res 43:D204–D212
8. Hermjakob H, Montecchi-Palazzi L, Lewington C, Mudali S, Kerrien S, Orchard S, Vingron M, Roechert B, Roepstorff P, Valencia A, Margalit H, Armstrong J, Bairoch A, Cesareni G, Sherman D, Apweiler R (2004) IntAct: an open source molecular interaction database. Nucl Acids Res 32:D452–D455
9. Jessop DM, Adams SE, Willighagen EL, Hawizy L, Murray-Rust P (2011) OSCAR4: a flexible architecture for chemical text-mining. J Cheminform 3:41
10. Samwald M, Jentzsch A, Bouton C, Kallesøe CS, Willighagen E, Hajagos J, Marshall MS, Prud'hommeaux E, Hassenzadeh O, Pichler E, Stephens S (2011) Linked open drug data for pharmaceutical research and development. J Cheminform 3:19
11. Chen B, Dong X, Jiao D, Wang H, Zhu Q, Ding Y, Wild DJ (2010) Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. BMC Bioinform 11:255
12. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novère N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED et al (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19:524–531
13. Krause F, Schulz M, Swainston N, Liebermeister W (2011) Sustainable model building the role of standards and biological semantics. Methods Enzymol 500:371–395

14. Li P, Dada JO, Jameson D, Spasic I, Swainston N, Carroll K, Dunn W, Khan F, Malys N, Messiha HL, Simeonidis E, Weichart D, Winder C, Wishart J, Broomhead DS, Goble CA, Gaskell SJ, Kell DB, Westerhoff HV, Mendes P, Paton NW (2010) Systematic integration of experimental data and models in systems biology. BMC Bioinform 11:582
15. Swainston N, Jameson D, Li P, Spasic I, Mendes P, Paton NW (2010) Integrative Information Management for Systems Biology. In: Lambrix P (ed) Proceedings of the 7th international conference, DILS 2010, Gothenburg, Sweden, August 25–27, 2010. Lecture notes in computer science (DILS) 6254:164–178
16. Le Novère N, Finney A, Hucka M, Bhalla US, Campagne F, Collado-Vides J, Crampin EJ, Halstead M, Klipp E, Mendes P, Nielsen P, Sauro H, Shapiro B, Snoep JL, Spence HD, Wanner BL (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat Biotechnol 23:1509–1515
17. Herrgård MJ, Swainston N, Dobson P, Dunn WB, Arga KY, Arvas M, Blüthgen N, Borger S, Costenoble R, Heinemann M, Hucka M, Le Novère N, Li P, Liebermeister W, Mo ML, Oliveira AP, Petranovic D, Pettifer S, Simeonidis E, Smallbone K, Spasić I, Weichart D, Brent R, Broomhead DS, Westerhoff HV, Kirdar B, Penttilä M, Klipp E, Palsson BØ, Sauer U et al (2008) A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. Nat Biotechnol 26:1155–1160
18. Dobson PD, Smallbone K, Jameson D, Simeonidis E, Lanthaler K, Pir P, Lu C, Swainston N, Dunn WB, Fisher P, Hull D, Brown M, Oshota O, Stanford NJ, Kell DB, King RD, Oliver SG, Stevens RD, Mendes P (2010) Further developments towards a genome-scale metabolic model of yeast. BMC Syst Biol 4:145
19. Thiele I, Swainston N, Fleming RM, Hoppe A, Sahoo S, Aurich MK, Haraldsdottir H, Mo ML, Rolfsson O, Stobbe MD, Thorleifsson SG, Agren R, Bölling C, Bordel S, Chavali AK, Dobson P, Dunn WB, Endler L, Hala D, Hucka M, Hull D, Jameson D, Jamshidi N, Jonsson JJ, Juty N, Keating S, Nookaew I, Le Novère N, Malys N, Mazein A et al (2013) A community-driven global reconstruction of human metabolism. Nat Biotechnol 31:419–425
20. Swainston N, Mendes P, Kell DB (2013) An analysis of a 'community-driven' reconstruction of the human metabolic network. Metabolomics 9:757–764
21. Swainston N, Mendes P (2009) libAnnotationSBML: a library for exploiting SBML annotations. Bioinformatics 25:2292–2293
22. Swainston N, Smallbone K, Mendes P, Kell D, Paton N (2011) The SuBliMinaL Toolbox: automating steps in the reconstruction of metabolic networks. J Integr Bioinform 8:186
23. Büchel F, Rodriguez N, Swainston N, Wrzodek C, Czauderna T, Keller R, Mittag F, Schubert M, Glont M, Golebiewski M, van Iersel M, Keating S, Rall M, Wybrow M, Hermjakob H, Hucka M, Kell DB, Müller W, Mendes P, Zell A, Chaouiya C, Saez-Rodriguez J, Schreiber F, Laibe C, Dräger A, Le Novère N (2013) Path2Models: large-scale generation of computational models from biochemical pathway maps. BMC Syst Biol 7:116
24. Haug K, Salek RM, Conesa P, Hastings J, de Matos P, Rijnbeek M, Mahendraker T, Williams M, Neumann S, Rocca-Serra P, Maguire E, González-Beltrán A, Sansone SA, Griffin JL, Steinbeck C (2013) MetaboLights–an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucl Acids Res 41:D781–D786
25. Xia J, Wishart DS (2011) Web-based inference of biological patterns, functions and pathways from metabolomic data using MetaboAnalyst. Nat Protoc 6:743–760
26. Hastings J, Magka D, Batchelor C, Duan L, Stevens R, Ennis M, Steinbeck C (2012) Structure-based classification and ontology in chemistry. J Cheminform 4:8
27. Batista-Navarro R, Rak R, Ananiadou S (2015) Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics. J Cheminform 7:S6
28. Spjuth O, Alvarsson J, Berg A, Eklund M, Kuhn S, Mäsak C, Torrance G, Wagener J, Willighagen EL, Steinbeck C, Wikberg JE (2009) Bioclipse 2: a scriptable integration platform for the life sciences. BMC Bioinform 10:397