The Network Library: a framework to rapidly integrate network biology resources

Georg Summer\textsuperscript{1,2,\ast}, Thomas Kelder\textsuperscript{3}, Marijana Radonjic\textsuperscript{2}, Marc van Bilsen\textsuperscript{1}, Suzan Wopereis\textsuperscript{2} and Stephane Heymans\textsuperscript{1}

\textsuperscript{1}CARIM, Maastricht University, Maastricht, The Netherlands, \textsuperscript{2}TNO, Zeist, The Netherlands and \textsuperscript{3}EdgeLeap B.V, Utrecht, The Netherlands

\ast To whom correspondence should be addressed.

Abstract

Motivation: Much of the biological knowledge accumulated over the last decades is stored in different databases governed by various organizations and institutes. Integrating and connecting these vast knowledge repositories is an extremely useful method to support life sciences research and help formulate novel hypotheses.

Results: We developed the Network Library (NL), a framework and toolset to rapidly integrate different knowledge sources to build a network biology resource that matches a specific research question. As a use-case we explore the interactions of genes related to heart failure with miRNAs and diseases through the integration of 6 databases.

Availability and Implementation: The NL is open-source, developed in Java and available on Github (https://github.com/gsummer).

Contact: georg.summer@gmail.com

1 Introduction

Network biology aims to provide researchers with a methodology to use the ever-increasing amount of experimental data and public knowledge to unravel the complexity of the biological processes they study. During the last decades the scientific community has aggregated large amounts of biological and experimental knowledge stored in databases or similarly structured data sources. A researcher faces the challenge to connect information from these different sources in the formulation and support of research hypotheses. Many data sources are dedicated to a specific research area or methodology leading to a fragmentation of the data itself, but also the methods to access it. Many initiatives and projects like IMEx (Orchard et al., 2012) aim to unify and integrate different data resources, but often the choices and decisions made do not align perfectly with the requirements of an individual researcher. We present the Network Library (NL), set of software tools that help researchers to merge the resources they need to answer their particular research question. The NL is developed in Java and is available under an Apache 2.0 license on GitHub (https://github.com/gsummer). Neo4j (www.neo4j.com) serves as a persistent database for the NL to build networks for researchers. Neo4j is a NoSQL database departing from the relational model of conventional SQL databases to offer an optimized system to store and query graphs. The NL offers tools to create and annotate the nodes of a network, a set of parsers to merge interaction databases into a network, and examples of algorithms that change the network based on already stored edges and nodes (e.g. used to consolidate different data sources). Additionally the NL has functionality to query the network to extract information and sub-networks in different formats for use in other software applications. The NL creates a Neo4j database that can be accessed from different software tools that are able to connect to Neo4j [e.g. cyNeo4j (Summer et al., 2015) for Cytoscape (Saito et al., 2012), (Shannon et al., 2003), Gephi (Bastian et al., 2009) with Neo4j support plugin (https://marketplace.gephi.org/plugin/neo4j-graph-database-support/)]. The NL aims to provide researchers with the maximum level of control over how they want to build a particular network. This article describes the functionality and workflow of the NL and its components and discusses a use-case focused on how to combine protein-protein interactions, microRNA interactions and gene-disease associations to investigate a group of genes implicated in cardiac hypertrophy. During pathological cardiac hypertrophy the heart muscle thickens in order to compensate for the chronically increased workload imposed on the heart muscle. Although this response is considered adaptive initially, with time it becomes maladaptive and predisposes to the development of heart failure. On the NL website (http://bionetlib.wordpress.com) instructions are provided on how to build the NL components as well as further documentation.

2 Methods

The NL components are implemented in and require Java 7 and use Maven 3 as a build system. The source code is available at https://
github.com/gsummer. Table 1 lists the components of the NL with a short description and their GitHub repository name. More information on the available parsers, how to extend the NL and the use-cases can be found at http://bionetlib.wordpress.com.

2.1 Neo4j
Neo4j is a Java-based NoSQL database optimized to store a large-scale graph and query the graph efficiently and fast. Neo4j servers provide access to the graph via a HTTP REST interface or can be directly integrated into a Java application via a library. A major component of Neo4j is Cypher, a query language similar to SQL but designed to operate on a graph consisting of nodes connected by edges. Cypher allows the user to select nodes and edges based on properties (e.g. a name or a value larger than a threshold) but also based on topology (e.g. to extract all nodes and edges along the shortest path from A to B). Cypher can be used to aggregate data and calculate statistics from the graph and present them in a tabular format. Additionally the query language can be used to retrieve subgraphs based on topological properties as well as node and edge annotations. Neo4j is regularly updated with new features and freely available as a community edition at www.neo4j.com.

2.2 Workflow
The NL tools are designed to build up a graph for analysis in stages. Each stage adds more data or knowledge to the graph from a given source or computes it from the graph itself. In practice the creation and annotation of a set of nodes is the first step. After the definition of an initial set of nodes, the user then builds up the graph in an iterative fashion. Edges are imported based on criteria and settings fitting a particular data source (e.g. thresholds for scores). If during the subsequent analysis and investigation of the graph additional annotations and data sources are needed, they can be added without a rebuild of the graph.

2.3 Node set creation
The NL Primer is the principal tool to add nodes and their annotations to a graph. The default input is a delimiter-separated file (the delimiter can be user defined) with the first value of each row being an identifier that is searched for in the graph and all following values are annotations to be saved. The identifier is then searched in the search index (maintained by Neo4j) and, if not present, a new node is created and the identifier is indexed in the graph. The rest of the values in each row can be stored as either a property, a label or both. A label is a Neo4j specific method to assign a type to a node for the creation of a set of indices to optimize querying a graph. Additionally to adding a value as a property or a label, the value can be associated to the node via an index to allow for efficient searching. If in a subsequent step a value is used as an identifier, the Primer will return this node. This allows the user to build up an index to search nodes with the identifiers needed for the import of edges. The index allows for non-unique associations, e.g. one identifier mapping to more than one node, but leaves it to the user to actually use this feature. Next to a delimiter-separated file as an input, the Primer can also be extended to support more complicated file formats. One such parser is the miRBase (Kozomara and Griffiths-Jones, 2014) parser, it extracts mature miRNAs from the complete.dat file, stores the mature miRNAs as nodes and associates the stem-loop identifiers to each mature miRNA. An accompanying parser allows to process the Alias file of miRBase. Additional parsers are available to extract the diseases used in DisGeNET (Pitero et al., 2015) or the headings of large GMT files. The Primer contains an extendable interface and utility Java classes to quickly build and integrate new parsers in its framework. Detailed information on the available parsers can be found on http://bionetlib.wordpress.com. The actual storage and graph operations are separated from the parsing to allow users to focus on new parsers.

2.4 Edge import
The NL Edger follows a similar principle as the Primer. A parser packaged into the Edger is tasked with extracting edges from a given input. Typically the input is a text-based source, but can also be a web-service (e.g. Transcription Factor Encyclopedia parser) or similar. Combined with parser specific parameters edges will be extracted. Edges are defined as identifiers for start- and end-nodes, a type, a direction and a set of properties for the edge. Neo4j enforces directionality in the basic data-model, but it can be ignored during graph operations. An alternative is to add two edges for each undirected edge (as done by the STRING DB Links parser). After an edge is extracted, both the start- and end-node are searched in the graph index. The default behaviour is to save an edge in the graph only if both identifiers are not found. In the other case of multiple identifiers, the edge is added multiple times between all start- and end-nodes. Both behaviours of the Edger are tunable to either allow for new nodes to be created or to enforce that edges are created only once. Data sources might require the user to annotate nodes with a specific identifier. The node lookup behaviour of the Edger combined with parser parameterization provides the user with a granular control over what edges are merged into the graph. Similar as with the Primer, the Edger provides interfaces, utility classes and examples to build new parsers. A more detailed overview as well as a list of parsers is available on the accompanying website http://bionetlib.wordpress.com. Currently the Edger provides parsers for 18 different databases and data-formats (delimited text files, GMT files).

2.5 Network manipulation
The NL Curator provides functionality to modify an existing graph systematically and on a larger scale. The Curator is designed to conduct graph spanning operations that can alter the graph significantly. An example is the consolidation of multiple edge types, which is discussed in the results section. Another method available is the ability to rewire a specific set of edges between nodes. This allows for a basic randomization of a graph for testing and comparison purposes. The Curator is envisioned as a place to implement

| Table 1. Components of the Network Library |
|-------------------------------------------|
| Name           | Description                               | Repository      |
| Core           | Shared functionality between the different components | netlib-core     |
| Primer         | Creation, indexing and annotation of nodes | netlib-primer   |
| Edger          | Provides parsers to import edges from various data sources | netlib-edger    |
| Curator        | Used to manipulate a graph, e.g. by consolidating multiple edges into one | netlib-curato   |
| Scribe         | Methods to extract information from the graph | netlib-scribe   |
very specific tasks into a common framework. The interface of the Curator is kept at a low-level to give researchers the necessary flexibility.

2.6 Network retrieval

Eventually users would want to extract sub-graphs or statistics from the network. The Scribe is a tool that can extract sub-graphs and store the result in a commonly used format (e.g., tabular separated list of edges or XGMML). Neo4j’s query language Cypher can be used to define a sub-set of nodes and edges to be included in the sub-graph. Additionally, the Scribe provides examples for other use-cases. Connecting a group of nodes through available edges or similar can be done easier in a small Java application than through Cypher. A second tool to access a Neo4j graph is the Cytoscape plugin cyNeo4j. The plugin lets users query a Neo4j server using Cypher and then analyze and visualize the results directly with Cytoscape. Both the Scribe and cyNeo4j only support a subset of the functionality of Cypher, expecting the result to be a collection of nodes and edges.

3 Results

The goal of the NL is to enable researchers to iteratively build a graph from multiple data sources to facilitate their biological network analysis pipelines. An example of a research case where the NL increases efficiency of a biological network analysis is an investigation of major risk factors for the development of heart failure. A starting point for this use-case is 20 genes implicated in cardiac hypertrophy identified by a literature review. The goal was to identify microRNAs (miRNAs) regulating these genes as potential therapeutic targets and to discover the mechanisms of how those miRNAs might affect cardiac hypertrophy. As a last step the genes affected by the miRNAs are investigated for associations with other diseases than heart failure to better understand the underlying biological context.

Mature miRNA are 20 to 25 nucleotide long RNA fragments that regulate protein expression through mRNA degradation or inhibition of protein translation. It is known that a specific miRNA can target multiple miRNAs and that one mRNA can also be targeted by multiple miRNAs. Target identification based on in silico prediction tools is a core strategy when investigating the biological effects of miRNAs. To identify miRNAs that regulate cardiac hypertrophy multiple miRNA-targeting prediction data-sources are integrated with protein–protein interactions. Figure 1 shows the workflow for investigating the cardiac hypertrophy genes, the full commands as a script, Cypher queries and additional information are available on the NL website.

The first step is to create nodes to represent genes and miRNAs. For genes we use Ensembl as a data source, where we excluded the miRNA genes. For miRNAs we used the mature miRNAs from miRBase. The first step creates the core set of nodes for the genes. Each Ensembl Gene ID in the input file is searched for in the graph and if not present a new node is created. In the same step the Ensembl Gene ID is added and indexed as an annotation to the node. For the current use-case we restricted the genes to be from Mus musculus and be located on chromosomes 1–19, X, Y or MT (excluding patches to the genome). After creating the nodes for the genes, other identifiers (such as RefSeq or Entrez Gene IDs) are used to annotate each node. Next to added information these annotations build up an identifier mapping system within the network. This is necessary since the different data sources for edges rely on different gene identifiers. To complete the annotation of the genes in the network, each gene is labeled with a biological type and furthermore the cardiac hypertrophy related genes are marked as such. The second part of the core node set are the mature miRNAs. The Mus musculus miRNAs are extracted directly from the full miRBase data file, resulting in one node for each mature miRNA annotation with the name, mature miRNA identifier (MIMAT) and the identifier for the precursor (MI). As for all identifier systems, also the miRNA one has gone through numerous changes in the last decade, requiring the import of an alias file. The aliases contain alternatives both for MIs and MIMATs. Since one precursor can yield multiple mature miRNAs, aliases for precursors can result in multiple return values. The default behaviour is to ignore multiple search results but can be modified using the allow_multi parameter. Similar to the genes, mature miRNAs are labeled as miRNAs with a Cypher query identifying them via the property mimat. The next step is to add the protein-protein interactions from STRING DB (Szklarczyk et al., 2014) to the graph. STRING Links is used as a data source and only edges with a combined score of 400 and above (or 0.4 on the website) were imported. The following step adds four different databases that provide miRNA targeting, 3 predicted: TargetScan (Shin et al., 2010), miRDB (Wong and Wang, 2014) and DIANA microT-CDS (Rezko et al., 2012) and 1 validated: miRTarbase (Chou et al., 2015). TargetScan predicts targeting for families of miRNAs (miRNA that share a seed sequence) rather than individual one, requiring an extra mapping during between miRNA family and mature miRNA. The other databases use cut-offs based on suggestions by the respective developers. A common practice in interpreting miRNA targeting predictions is to combine the information and only focus on predictions supported by multiple databases or tools (Dweep and Gretz, 2015). The Curator has a module to merge multiple edges between two nodes. The MIRC module counts the edges...
between a miRNA and one target, then divides the count by a user provided normalization factor (typically number of databases) and creates a new edge (miRc) with the result as a property. In the given case the property is named \textit{score}.

At this point the network contains mouse genes and miRNAs. Genes are connected to each other through protein–protein interactions derived from STRING DB and miRNAs are linked to genes based on multiple prediction tools. Additionally the genes relevant for cardiac hypertrophy were marked as such. The resulting integrated network (46,221 nodes and 3,463,049 edges) allows the researcher to ask questions such as: Which miRNAs target most of the cardiac hypertrophy genes? Cypher is the preferred tool for Neo4j to answer such questions. The top 10 miRNAs targeting the cardiac hypertrophy genes are listed in Table 2 (Cypher query: Listing 1).

\begin{verbatim}
Listing 1 Cypher-query for Table 2
MATCH(h:hypertrophy)-[s:interacts_with]->(n)<-[r:miRc]-(:miRNA) WHERE r.score>=0.5 and s.combined_score>=0.7 RETURN h.name as snameined_s gene, m.name as miRNA, count(n) as num_targets order by num_targets desc
\end{verbatim}

Biological processes are complex and it is likely that the selected genes are part of different processes underlying the development of cardiac hypertrophy. Genes interacting with any of the selected cardiac hypertrophy genes probably also play a role in the disease. The combination of protein-protein interactions and miRNA targets in one graph allows for the identification of miRNAs that target the neighbourhoods of a cardiac hypertrophy gene. Table 3 (Cypher queries Listing 2) shows the top miRNAs targeting neighbours connected by a protein-protein interaction with a score of \(>700\) (high confidence based on the STRING website).

\begin{verbatim}
Listing 2 Cypher queries for Table 3
MATCH (h:hypertrophy)-[s:interacts_with]->(n)<-[r:miRc]-(:miRNA) WHERE r.score>=0.5 and s.combined_score>=0.7 RETURN h.name as snameined_s gene, m.name as miRNA, count(n) as num_targets order by num_targets desc
\end{verbatim}

Figure 2 (Cypher query Listing 3) places an example miRNA (mmu-miR-17-5p) into the context of the hypertrophy genes and their neighbours. MiR-17-5p targets the two hypertrophy-associated genes Irf9 and Smad7 directly, but also at least two direct neighbours of every other cardiac hypertrophy gene, the only exception being Ctsl.

\begin{verbatim}
Listing 3 Cypher queries for Figure 2
MATCH (h:hypertrophy)-[s:interacts_with]->(n)<-[r:miRc]-(:miRNA {name:"mmu-miR-17-5p"}) WHERE r.score>=0.5 and s.combined_score>=0.7 RETURN h.name as snameined_s gene, m.name as miRNA, count(n) as num_targets order by num_targets desc
\end{verbatim}

The genes implicated in cardiac hypertrophy and their regulators do not operate in isolation while modulating the development of heart failure. These genes and their regulators are part of a large underlying biological system and consequently might influence other disease processes as well. To investigate such disease to gene associations, the current

\begin{table}[h]
\centering
\caption{Top 10 miRNAs targeting the cardiac hypertrophy genes}
\begin{tabular}{|l|l|l|}
\hline
miRNA & No. of targets genes & Target genes \\
\hline
mmu-miR-466l-3p & 3 & Adgr4, Atf3, Il10 \\
mmu-miR-363-3p & 2 & Smad7, Rgs3 \\
mmu-miR-27b-3p & 2 & Il10, Map2k7 \\
mmu-miR-20a-5p & 2 & Smad7, If9 \\
mmu-miR-25-3p & 2 & Smad7, Rgs3 \\
mmu-miR-32-5p & 2 & Smad7, Rgs3 \\
mmu-miR-92b-3p & 2 & Smad7, Rgs3 \\
mmu-miR-1192 & 2 & Hsp4, Smad7 \\
mmu-miR-17-5p & 2 & Smad7, If9 \\
mmu-miR-106a-5p & 2 & Smad7, If9 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{miRNAs targeting the neighbourhood of cardiac hypertrophy genes}
\begin{tabular}{|l|l|l|}
\hline
Hypertrophy gene & miRNA & No. of targeted neighbours (%) & No. of neighbours \\
\hline
Atf3 & mmu-miR-466l-3p & 29 (13.0%) & 223 \\
Smad7 & mmu-miR-17-5p & 28 (9.9%) & 281 \\
Smad7 & mmu-miR-669c-3p & 26 (9.3%) & 281 \\
Smad7 & mmu-miR-19b-3p & 26 (9.3%) & 281 \\
Smad7 & mmu-miR-137-3p & 26 (9.3%) & 281 \\
Smad7 & mmu-miR-124-3p & 25 (8.9%) & 281 \\
Il10 & mmu-miR-27b-3p & 25 (8.9%) & 281 \\
Smad7 & mmu-miR-466l-3p & 25 (11.0%) & 227 \\
Smad7 & mmu-miR-106a-5p & 24 (8.5%) & 281 \\
\hline
\end{tabular}
\end{table}
The graph needs to be extended. As a source for disease to gene associations DisGeNET is used. DisGeNET integrates multiple different knowledge sources and scores the disease to gene associations based on available evidence. DisGeNET uses the Unified Medical Language System (UMLS) to identify diseases. The UMLS is extensive and to avoid dragging its entirety into the network we extract only the diseases references in DisGeNET. The NL Primer provides a parser that extracts and creates the diseases from the data-file of DisGeNET (adding 14,619 nodes). Similar to miRNAs, also diseases were labeled as such for easier use in Cypher queries. DisGeNET relies on human gene symbols as a gene identifier, whereas the network at this point only included mouse gene symbols. In many cases only the capitalization differentiates the gene symbols from human to mouse. However, regularly similar genes cannot be mapped uniquely, resulting in multiple associations between a human and mouse genes. Ensembl provides an inter-species mapping from mouse to human that can be added as an annotation to the network. With the human gene symbols in place the DisGeNET associations (484,553 new edges) can be merged into the network.

**Listing 4** Cypher queries for Figure 3

MATCH (h:hypertrophy)-[s:interacts_with]->(n)<-[r:miRc]-(m:miRNA {name:"mmu-miR-17-5p"}) WHERE h.name="Smad7" and r.score >= 0.5 and s.combined_score > 0.7 and a.score > 0.1 RETURN n,a,d,s,n2

In Figure 3 (Cypher queries Listing 4) the focus is on the genes that are targeted by miR-17-5p and neighbours of Smad7, and the diseases they are associated with. Egfr, Stat3 and Tgfbr2 are the genes associated most with diseases, although a majority of these diseases are related to or are a type of cancer. Overall the 24 genes are associated with 186 diseases. The overlaps in disease associations can be a starting point to investigate the shared, core processes between types of diseases or point researchers towards fields that might hold additional information and data about genes they are interested in.

**4 Discussion**

Biological knowledge and information is stored in many different data sources and collecting and connecting this data is an immensely useful tool to generate and support research hypotheses. Researchers have to either rely on others to attempt such data aggregation projects and hope the decisions made fit their research questions, or they have to embark on this task themselves. We developed the NL to aid researchers in building a network from publically available knowledge that supports them in the interpretation and analysis of their experimental data. The NL is designed to maximize the control users have while building their networks, but at the same time provides a unified and systematic approach during this process. To achieve this, the NL is split into individual components, each focusing on a specific task. The Primer is used to create and annotate nodes in a network, whereas the Edger combines parsers for

![Fig. 3. Disease associations of genes that are targeted by mmu-miR-17-5p and neighbours of Smad7](https://academic.oup.com/bioinformatics/article-abstract/32/17/i473/2450766)
different network-oriented data sources. The Curator provides a flexible interface to develop algorithms that manipulate an existing network based on its properties (e.g. miRNA targeting edges are merged to have a weighted consolidated edge). The final component of the NL is the Scribe, a tool to query a network for statistics or subnetworks. We presented a use-case focusing on the interactions among genes, miRNAs and diseases in the context of heart failure. Combined with the NL website (http://bionetlib.wordpress.com) we provide examples, tutorials and documentation on how to use the NL to integrate different data sources to build biological networks that help answer specific research questions.

4.1 Network size and performance
The network build for the presented use-case contains 60,480 nodes and 3,947,602 edges. While it is possible to fit such a large network into Cytoscape using an extremely high-end PC, the use of Neo4j as a backend allowed us to build and perform the described analysis on a notebook with an i7-5500 U CPU @ 2.40 GHz, 8 GB RAM and an SSD drive. The complete build process of the network takes ~6 min and the final network requires roughly 800 MB (after log-file removal).

4.2 Extending the NL
As of version 1.0 of the NL 18 different types of input for node annotation (e.g. miRBase, DisGeNET) and edge data sources (e.g. STRING DB, TargetScan) are supported. Both Primer and Edger are designed to abstract the network and storage related task to allow input parsers to be as lean as possible. The implemented parsers can serve as examples and blueprints for other researchers to implement either new parsers or new spins on already supported data sources. We highly encourage others to contribute to the extension of the NL. Next to more parsers, the roadmap for the NL includes a switch to an OSGi module system, adaptation of the upcoming binary communication protocol for Neo4j, and improved extraction of subnetworks through the Scribe.

4.3 Identifier mapping limitations
The NL does not inherently provide an identifier mapping functionality. The search index used to identify nodes is populated by the Primer, placing the responsibility to have the necessary identifiers for the import of edges at the user. The goal of this separation is to keep the functionality of the NL as granular as possible to allow the user to account for such problems either by prefixing during the import steps or the use of a different identifier (if available).

5 Conclusion
The NL provides researchers with a toolset to combine different biological knowledge databases and build integrated network resources that fit their particular needs. The NL is highly extendable and customizable, allowing researchers to add (as of yet) unsupported databases with ease and ensure networks are built as desired. To illustrate the utilization of the NL, we provided a use-case that combined 6 different databases to help unravel the regulation of a set of cardiac hypertrophy genes and how those genes might affect other diseases.

Funding
The project was supported by the EU FP7 MEDIA (no. 261409) project. Development to support clinical data integration is part of the EU FP7 HOMAGE (no. 05507) project.

Conflict of Interest: none declared.

References
Bastian, M. et al. (2009) Gephi: an open source software for exploring and manipulating networks. ICWSM, 8, 361–362.
Chou, C.H. et al. (2015) miRTarBase 2016: updates to the experimentally validated mirna-target interactions database. Nucleic Acids Res., 44, D239–D247.
Dweep, H. and Gretz, N. (2015) mirwalk2.0: a comprehensive atlas of microRNA-target interactions. Nat. Methods, 12, 697–697.
Kim, S. et al. (2015) Pubchem substance and compound databases. Nucleic Acids Res., 44, D1202–D1213.
Kozomara, A. and Griffiths-Jones, S. (2014) mirbase: annotating high confidence microRNAs using deep sequencing data. Nucleic Acids Res., 42, D68–D73.
Orchard, S. et al. (2012) Protein interaction data curation: the international molecular exchange (imex) consortium. Nat. Methods, 9, 345–350.
Pireiro, J. et al. (2015) Dignet: a discovery platform for the dynamical exploration of human diseases and their genes. Database, 2015, bav028.
Recevo, M. et al. (2012) Functional microRNA targets in protein coding sequences. Bioinformatics, 28, 771–776.
Saito, R. et al. (2012) A travel guide to cytoscape plugins. Nat. Methods, 9, 1069–1076.
Shannon, P. et al. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res., 13, 2498–2504.
Shin, C. et al. (2010) Expanding the microRNA targeting code: functional sites with centered pairing. Mol. Cell, 38, 789–802.
Summer, G. et al. (2015) cyNeo4j: connecting Neo4j and Cytoscape. Bioinformatics, 31, 3868–3869.
Szklarczyk, D. et al. (2014) String v10: protein–protein interaction networks, integrated over the tree of life. Nucleic Acids Res., gkt1003. page
Wong, N. and Wang, X. (2014) mirdb: an online resource for microRNA target prediction and functional annotations. Nucleic Acids Res., 43, D146–D152.