ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ροές προγραμμάτων για την ανάλυση γονιδίων

Επιβλέπουσα
Καθηγήτρια
Άρτεμις Χατζηγεωργίου

Δημήτρης Παπαδημητρίου

Για το Πρόγραμμα Μεταπτυχιακών Σπουδών Επιστήμη και Τεχνολογία Υπολογιστών, Τηλεπικοινωνιών και Δικτύων στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

23 Μαΐου 2015

Αφιερώνω την εργασία στους γονείς μου...

Acknowledgements

First of all, I would like to thank the supervisor of this thesis, Dr Artemis Hatzigeorgiou for giving me the opportunity to work in such an interesting field and for her guidance through the whole process. Additionally, I want to thank Maria Paraskevopoulou, member of DIANA Lab, for her invaluable assistance and the sharing of her expertise during the course of this work. Also I would like to thank all the members of DIANA Lab for their valuable suggestions and interesting conversations regarding the thesis. Finally I want to thank my family for their continuous support.

Abstract

microRNAs (miRNAs) are a class of small, approximately 22 nucleotide long, non-coding, single stranded RNA molecules. They are found in diverse species and have a major role in numerous biological processes, mainly by acting as post-transcriptional gene regulators. Due to their great biological significance, software tools have been developed, facilitating biology researchers to perform complex computational tasks and query miRNA-related databases. Specifically, DIANA Lab exposes the functionality of microT-CDS and microT-v4 prediction algorithms, TarBase database and miRPath software as web services through a REST API. Additionally these services are integrated into Taverna Workflow Management System via a plugin, enabling their orchestration into workflows.

The aim of this study, is the improvement of the DIANA Lab Taverna plugin and the redesign of workflows provided by DIANA Lab to provide new features and handle error cases. A further objective is the integration of DIANA web services with Galaxy, a data integration and workflow management system used extensively by biology researchers.

The results of this thesis is the improved and upgraded to the latest Taverna version DIANA Lab Taverna plugin, which is deployed and available for use. In addition, the workflows are redesigned incorporating flow branching logic. Finally, regarding Galaxy, four tools were created, one for each web service, providing part of their functionality to Galaxy users.

Περίληψη

Τα microRNAs είναι μικρά, μήκους περίπου 22 νουκλεοτιδίων, μη κωδικά, μονόκλωνα μόρια RNA. Έχουν βρεθεί σε ποικίλα είδη και διαδραματίζουν καθοριστικό ρόλο σε πληθώρα βιολογικών διαδικασιών κυρίως μέσω της μέτα-μεταγραφικής ρύθμισης της έκφρασης γονιδίων. Εξαιτίας της μείζονος βιολογικής τους σημασίας έχουν αναπτυχθεί εργαλεία λογισμικού για να διευκολύνουν τους βιολόγους ερευνητές να εκτελούν πολύπλοκες υπολογιστικές διεργασίες και να επικοινωνούν με σχετικές βάσεις δεδομένων. Συγκεκριμένα η ερευνητική ομάδα του DIANA Lab εκθέτει ως υπηρεσίες ιστού, μέσω μιας REST διεπαφής, τη λειτουργικότητα των αλγόριθμων πρόβλεψης microT-CDS και microT-v4, της βάσης δεδομένων TarBase και του λογισμικού miRPath. Παράλληλα οι υπηρεσίες αυτές είναι ενσωματωμένες στο σύστημα διαχείρισης ροών εργασίας Taverna μέσω ενός plugin, καθιστώντας έτσι δυνατή την ενορχήστρωσή τους σε ροές εργασίας.

Σκοπός της εργασίας είναι η βελτίωση του DIANA Lab Taverna plugin και ο επανασχεδιασμός ροών εργασίας που παρέχει το DIANA Lab , ώστε να παρέχουν καινούριες δυνατότητες και να διαχειρίζονται περιπτώσεις λάθους. Επίσης έχει ως στόχο την ενσωμάτωση της λειτουργικότητας των υπηρεσιών ιστού του DIANA Lab στο Galaxy, ένα σύστημα ενσωμάτωσης δεδομένων και διαχείρισης ροών εργασίας που χρησιμοποιείται εκτενώς από ερευνητές.

Τα αποτελέσματα της εργασίας είναι το βελτιωμένο και αναβαθμισμένο στην τελευταία έκδοση του Taverna, DIANA Lab Taverna plugin, το οποίο είναι διαθέσιμο προς χρήση. Επίσης οι ροές εργασίας ανασχεδιάστηκαν ενσωματώνοντας λογική διακλάδωσης της ροής. Τέλος, σχετικά με το Galaxy, αναπτύχθηκαν τέσσερα εργαλεία, ένα για κάθε υπηρεσία ιστού, παρέχοντας μέρος της λειτουργικότητάς τους στους χρήστες του Galaxy.

Contents

List of Figures

1 Introduction

1.1 microRNAs

1.1.1 Definition and discovery

microRNAs (miRNAs) are a class of small, about 22 nucleotide (nt) long, non-coding, single stranded RNA molecules. They were first discovered in 1993 from Ambros lab studying the developmental events of Caenorhabditis elegans (C. elegans) [1]. Specifically, they found that lin-4 gene which is known to repress the level of lin-14 protein does not encode protein. Instead it produces small non-coding RNAs, one of them 22 nt long which contains sequences partially complementary to a repeated sequence element in the 3' untranslated region (UTR) of the lin-14 mRNA. The second finding was again in C. elegans in 2000 when it was found that let-7 gene encodes a 21 nt long RNA, complementary to elements in the 3' UTR of lin-41 gene leading to its downregulation [2]. The same year let-7 miRNA was found in many species including humans [3]. Since then numerous miRNAs have been discovered in increasing number and diverse species, including plants, flies, nematodes, mice and humans affecting gene expression in many regulatory pathways [4, 5].

1.1.2 Biogenesis

The steps of miRNA biogenesis are depicted in Figure 1. At first microRNAs are transcribed by RNA polymerase II (Pol II) [6] and in more rare cases by RNA polymerase III (Pol III) [7]. From this process long primary transcripts (pri-miRNAs) are created ranging in size from several hundred nt to several kilobases. These transcripts contain at least one, but usually more, stem-loop structures [8] called precursor miRNAS (pre-miRNAs). Pre-miRNAs are recognized and cleaved by the microprocessor complex into hairpin structures of about 70 nt long. Microprocessor complex consists of nuclear RNase III Drosha and nuclear protein DGCR8 in humans or Pasha in D. melanogaster and C. elegans [9, 10, 11, 12]. Then the pre-miRNAs pass from the nucleus to the cytoplasm with the help of exportin-5 protein [15]. In cytoplasm they encounter Dicer enzyme which is also a RNase III endonuclease. Dicer cleaves the loop of the pre-miRNA creating 22 nt long miRNA duplexes [13]. These duplexes are loaded onto an Ago protein where the one strand, the messenger strand is peeled away and degraded whereas the other, the guide strand, remains in order to generate the RNA-induced silencing complex (RISC), from which the gene expression regulation is performed. The choice of which strand enters the RISC largely lies in the relative stability of the duplex ends. Specifically in almost all cases the strand whose 5' end is less tightly paired is preferred [14].

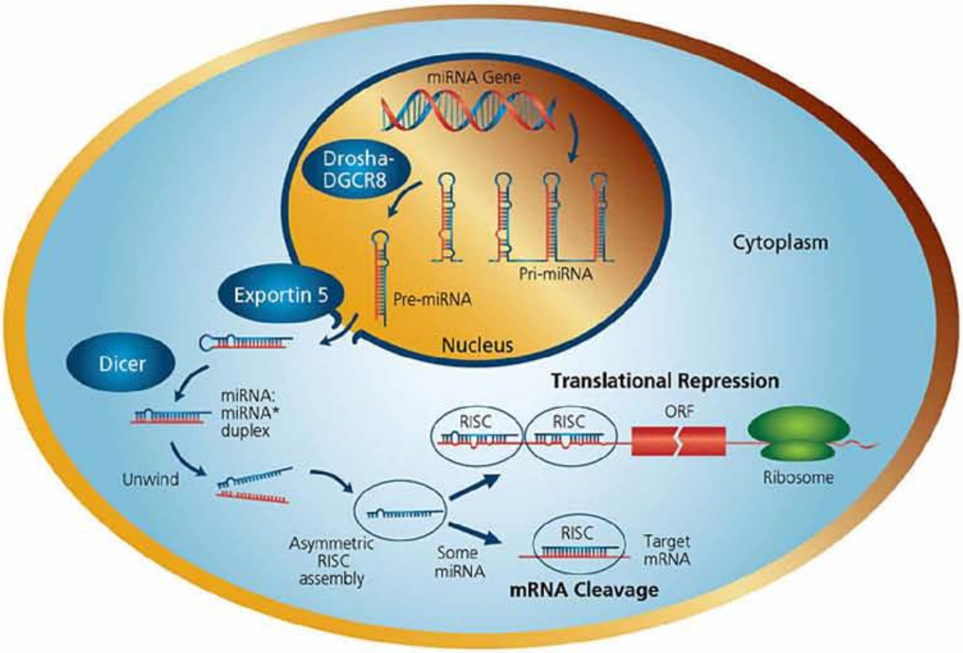


Figure 1: microRNA biogenesis and functionality [16]

1.1.3 Functionality

The basic functionality of miRNAs is that when in RISC, they pair with mRNAs and in this way they regulate gene expression. If the pairing is fully complementary then the target mRNA is cleaved. This behavior is most common in plants [17]. Otherwise, when the pairing is not fully complementary, case which is the most common in metazoan miRNAs, the mRNA can be destabilized, its translation can be repressed or a combination of the two can take place [1, 18]. The miRNA region where the paring takes place is called the seed and it is the 5' region of a miRNA, located on nucleotides 2–7 [19]. The miRNAs target specific areas in protein coding genes called miRNA Recognition Elements (MREs) [15]. They were first found in the 3' UTR of the mRNAs [21] but later were also found in the 5' UTR and in the coding sequence (CDS) [22].

miRNAs via the post-transcriptional gene regulation induce translational repression, mRNA degradation and gene silencing. As a consequence they have a crucial role in many biological procedures, such as development, stem cell differentiation and immune mechanisms. Also possible divergence in the miRNA concentration can lead to numerous pathological states, like cancer, cardiovascular diseases, autoimmune diseases and many more [23, 24, 25, 26, 27]. Due to these facts miRNAs are under investigation for their behavior and their therapeutic potential.

1.1.4 Target identification

A crucial field of miRNA related research is the identification of target mRNAS for specific miRNAs. This can be done either with computational or experimental techniques.

The computational techniques are typically the first step as they can provide new and unknown miRNA-mRNA interactions. In silico miRNA target recognition is mainly based on base pairing (Watson-Crick) of the seed miRNA region with areas in the 3' UTR of mRNAs. However, some researchers use the extended seed sequence of miRNAS which is nts 1-9 from the 5'end of a miRNA. Also some algorithms try to find seed pairings in regions at the 5' UTR and the CDS of mRNAs. Along with seed matching, many algorithms use additional features to improve their efficiency such as sequence conservation, free energy and site accessibility [28].

Experimental techniques are mainly used for the verification of the in silico predictions. Most of the studies use either direct or indirect experimental methods in order to determine the miRNA-mRNA interactions. Direct techniques aim to verify specific miRNA-mRNA interactions whereas indirect ones try to detect interactions in a wider scale.

Examples are Ago tagging and immunoprecipitation approaches which mainly identify targets regulated at translation level, transfection and microarray analysis techniques which provide targets at degradation level and proteome analyses which provide a more thorough approach as they can detect all kinds of repression [29].

Despite all of these techniques, the full spectrum of miRNA targets is still unknown, even for extensively researched species like human.

1.2 Web Services

Quoting W3C a Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards [30].

In simpler words Web Services expose an application in a standardized web format described by WSDL. Software agents can communicate with this service via SOAP messages. Some of the benefits that Web Services provide is that they are platform-independent, language independent and based on well known standards. In addition web services can be easily combined in workflows using workflow management systems.

With these advantages given, many bioinformatic service providers, such as EBI and NCBI and research labs eg DIANA Lab offer their applications functionality as web services easing in this way the life scientists and especially biologists to retrieve and process data.

1.2.1 RESTful Services

Representation State Transfer (REST) architecture, proposed by Fielding [32] has given new insights on what can be considered a web service and how it can be developed. Specifically REST services overcome the strict rules regarding the use of WSDL and SOAP and establish a more simple and lightweight web service paradigm.

REST compliant web services are based on a client server model where the server exposes a set of resources identified by URIs. The resource is a quite abstract notion as according to Fielding is any information that can be named. Resources are accessible and manipulated by clients using a fixed set of operations, essentially create, retrieve, update and delete. As almost all REST compliant services use HTTP, these operations usually correspond to HTTP methods PUT, GET, POST and DELETE. REST interactions are stateless meaning that each client request must contain all the required information for a successful interaction and cannot depend on previously stored context. Finally, there aren't any constraints regarding the format of the output but it is advised to be in a machine readable format, for example XML or JSON [31].

1.3 Workflow Management Systems

Biology researchers usually use large amounts of data coming from different sources and computational tools provided from various service providers. As a result a typical biological analysis is performed in a staged fashion where the output of a software tool, providing for example genomic data acquired from a database is the input to another. In addition these multi-step complex analyses are difficult to be reproduced. Lastly the use of diverse software tools most of the times demands a strong bioinformatic infrastructure that is not always available in biologists.

These challenges are handled effectively by the concept of workflows and workflow management systems (WfMS) which provide a well defined and user friendly way to specify the necessary steps of a complex computational task.

It is accurate to say that a workflow is a general term describing the necessary steps for the execution of a process. It can be visualized as a directed graph where nodes represent the required steps and edges either represent data flow or dependencies between them.

A workflow management system (WfMS) is a software that provides the appropriate infrastructure for a user to create, execute, and monitor workflows. Typically these systems provide a visual front-end that facilitates the workflow creation using the graph-based paradigm and an execution engine that handles the enactment of the tasks assigned and the data transfer between the nodes.

A great advantage of WfMS is that they provide to biologists a high-level application-oriented programming framework for composing their multi step in silico tasks. Also WfMS handle most of the details regarding the invocation and the result processing of complex software services and distributed data sources, letting biologists focus on the logic of the process. Finally, the reproducibility of a computational experiment is facilitated as the researchers can share the workflows they used with anyone that has the same WfMS installed.

Due to their benefits in scientific analysis a large number of scientific workflow systems have emerged recently, such as Taverna [33], Kepler [34] and Galaxy [35, 36, 37]. These tools share the basic common features but they also differ in some aspects, for instance in the ability to control the flow execution.

1.3.1 Taverna

Taverna is an open source workflow management system used to create and execute scientific workflows.

The Taverna suite of tools is written in Java. It contains Taverna Engine which handles the workflow execution and Taverna Workbench, a graphical user interface facilitating the orchestration of diverse services in a workflow. It also includes Taverna Server, which enables the remote execution of workflows and the Taverna command line tool used for faster enactment of workflows from a terminal without the GUI overhead.

Taverna installation incorporates services from many great bioinformatic service providers, such as BioMart [38], BioMOBY [39] and SoapLab [40]. Furthermore it is integrated with BioCatalogue, a web registry that provides registering, searching, documenting and monitoring health science web services [41]. This means that a user can include the services of Biocatalogue that are compatible with Taverna in a workflow. It is also integrated with myExperiment, a social networking site where Taverna users can browse and share workflows, allowing in this way the direct reuse of workflows [42]. In addition it supports built in invocation utilities for WSDL services, R scripts and Beanshell scripts. Finally, due to the fact that the services used can be from diverse service providers, Taverna has a lot of built-in local services, shim services, used to convert data to the appropriate formats.

Taverna has been created by myGrid team. Its latest stable version is 2.5 and at the time is in incubating process as an Apache project. Taverna is extensible as new services can be integrated in a Taverna installation by installing new service invocation plugins.

1.3.2 Galaxy

Galaxy is an open-source, web-based platform aiming to ease data intensive biomedical research. It provides data integration and workflow management capabilities. Specifically it enables users to collect data from various database resources such as UCSC Table Browser [43], BioMart and EuPathDB [44] and process them using a wide range of tools. Tools functionality range from simple data manipulation, eg merging columns to bioinformatic tools, such as format converters and analysis packages. New tools can be created, and shared via Galaxy making it a fully extensible platform.

Galaxy supports a data flow model operation where the output of one node is input for others. User actions are recorded and stored in a history log. This log can be saved and presented as a workflow in a workflow editor where the procedure can be easily reproduced, modified, documented and shared.

Galaxy is developed by the Galaxy team at Penn State and Johns Hopkins University. There is also a Galaxy community contributing to the project including from simple users to Galaxy developers and tool developers. The myExperiment site mentioned for Taverna also hosts Galaxy workflows uploaded from users.

1.4 DIANA Lab

Numerous studies highlight the biological importance of non-coding RNAs (ncRNAs). The more significant types of NcRNAs are the miRNAs and the more recently discovered lncRNAs, both in the center of biological research as they are involved in diverse biological processes.

The biological significance of miRNAs was analyzed previously in this thesis. Recent findings have also revealed the importance of lncRNAs. Some of the biological processes lcRNAs are involved are chromatin remodeling, regulation of cell cycles, structural scaffolding of nuclear protein substructures and binding to Polycomb repressive complexes. Also it has been found that they interact with miRNAs regulating in this way gene expression [45, 46].

DIANA Lab provide algorithms, databases and software in order to facilitate the interpretation and the archiving of biological data in a systematic framework. Specifically DIANA Lab provides web servers for target prediction algorithms, for instance microT-CDS and databases of experimentally validated miRNA targets on coding and non-coding RNAs, such as TarBase v7.0 and LncBase. Furthermore miRPath software is available, a tool that detects the putative altered molecular pathways caused by the expression of a single or multiple miRNAs. DIANA Lab also exposes part of this database and server functionality as REST services. Lastly it supports the orchestration of these services in sophisticated workflows enabling users without advanced bioinformatic infrastructure to perform multi-step miRNA-related analyses.

1.4.1 DIANA Lab Servers and Databases

DIANA Lab server and database functionality are of major importance for biologists to perform analysis tasks. A user friendly web interface is provided for each of them, facilitating in this way their utilization and enabling more complex tasks. Some miRNA related database and servers of DIANA Lab are the following:

1.4.1.1 microT web server

This server provides results based on DIANA microT-CDS algorithm which is the 5th version of the microT target prediction algorithm [48, 49], exhibiting the highest sensitivity and specificity compared to other prediction algorithms. The main feature that microT-CDS uses to predict miRNA targets is seed match between the miRNA extended seed sequence, 1-9 nts from the 5'end of the miRNA, targeting not only the 3'UTR of mRNAs but their CDS as well. Targeting of MREs in CDSs and 3'UTRs is evaluated independently. The algorithm is trained on a positive and a negative set of PAR-CLIP data defined MREs. Also microT-CDS uses additional features for the improvement of its accuracy such as conservation, free energy, site accessibility and target-site abundance.

The server provides miRNA targets predictions for Homo sapiens, Mus musculus, Drosophila melanogaster and C.elegans species and can be accessed from the URL: `http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=microT_CDS`

DIANA Lab also provides DIANA-microT web server v4, which gives results of microT version 4 target prediction algorithm [50]. The major difference of the two algorithms is that version 4 does not consider CDS MREs as putative miRNA targets, but instead it focuses on the 3'UTR of mRNAS. The DIANA-microT web server v4 URL is `http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=microtv4`

1.4.1.2 DIANA-TarBase

TarBase is the largest available database of experimentally verified miRNA targets, derived from manual curation of published articles. The first version of TarBase was released in 2006 and it was the first available database of experimentally validated miRNA targets. Its sixth version comprises more than 65.000 interactions spanning 21 species [51]. Latest version, TarBase 7.0 [52] contains more than half a million interactions from 24 species. DIANA Lab provides web user interface for the latest and the previous version of TarBase at the following URLs respectively `http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=tarbase` and `http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=tarbasev6`.

1.4.1.3 DIANA-miRPath

DIANA-miRPath is a miRNA pathway analysis web-server. It hosts DIANA miRPath software which uses as input miRNA in silico predicted targets from microT-CDS and/or miRNA experimentally validated targets from TarBase v6. It then combines the results with merging and meta-analysis algorithms to detect significantly targeted pathways by the selected miRNA(s) [53].

The server provides miRNA targeted pathways for Homo sapiens, Mus musculus, Drosophila melanogaster and C.elegans species and can be accessed from the following url: `http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=mirpath`

1.4.2 DIANA Lab REST compliant services

The servers and databases of DIANA Lab are of crucial importance for biologists but they have some limitations. For instance they do not provide results in a standardized machine readable format and are not easy to be accessed programmatically. Furthermore the execution of multi-step complex tasks using more than one DIANA Lab utilities and/or other research lab software is very difficult.

These limitations have led to the decision of exposing part of the server and database functionality of DIANA Lab as web services. Specifically target prediction algorithms microT v4 and microT-CDS v5, database TarBase v6.0 and miRPath v2.1 software are provided as RESTful web services. These services can be accessed directly from the website or using a programming language and they provide results in XML format. In addition the services are registered in the BioCatalogue repository.

1.4.2.1 DIANA-microT-CDS (v5) Web Service

DIANA-microT-CDS (v5) Web Service exposes the basic functionality of DIANA-microT web server v5.0 as a REST service.

The rest query is formatted using one or more genes and/or one or more miRNAs. If only genes are provided the service responds with the miRNAs that are predicted to target these genes. If only miRNAs are provides the service responds with their gene targets. If both miRNAs and genes are provided, the service response will identify which genes are targeted by which miRNAs. The user can also specify a threshold for the presented predictions. An example query is

```
http://62.217.127.8/DianaTools/microT_CDSApi?mirnas=hsa-miR-93-5p
```

which at the time being provides 443 results of target genes, along with other useful information such as the binding type, the location of the match and the conservation level.

Considering the REST architecture concepts mentioned in the introduction, in this example the resource identifier is the URL, the resource are the results of microT-CDS algorithm for a specific miRNA and the resource representation is the XML formatted result. Only retrieve REST operation is supported using HTTP GET method. Finally it is obvious that the request is stateless as it includes all necessary information.

1.4.2.2 DIANA-microT-ANN (v4) Web Service

DIANA-microT-ANN (v4) Web Service exposes the basic functionality of microT v4 algorithm as a REST service.

The rest query and the service response formats are the same with the ones discussed in DIANA-microT-CDS (v5) Web Service.

1.4.2.3 DIANA-TarBase v2.0 Web Service

DIANA-TarBase v2.0 Web Service provides a REST API for TarBase v6.0 database.

The user formulates the REST query using one or more genes and/or one or more miRNAs. An example of a TarBase query is:

`http://62.217.127.8/DianaTools/tarbaseApi?mirnas=hsa-let-7a-5p`

The service will provide an XML formatted response containing the gene names and ids that miRNA hsa-let-7a-5p has been experimentally validated to have interactions with, along with other useful information, such as the species where the interactions take place and the experimental validation methods that were used.

1.4.2.4 DIANA-miRPath v2.1 Web Service

DIANA-miRPath v2.1 Web Service exposes DIANA-miRPath server functionality as a REST service.

The required attributes for a successful query to the REST service are one or more microRNAs, a miRNA-gene interaction identification method for every miRNA provided and the species under investigation which can be human or mouse. More parameters can be passed to the HTTP URL query, but they are not necessary since they have default values. These are method for multiple miRNA result combination, soft intersection, microT prediction threshold, false discovery rate significance level, use of conservative statistics and gene filtering.

```
http://62.217.127.8/DianaTools/mirpathApi?mirnas=hsa-mir-125b-5p&
methods=Tarbase&species=human
```

In the above query the service will provide an XML formatted response containing, among others the id and the description of targeted pathways.

1.4.3 DIANA-web server Taverna Plugin

A great benefit of Web Services is that they can be easily combined in workflows using a workflow management system, enabling the easy execution of complex multi-step tasks. DIANA Lab to facilitate this functionality has created the DIANA Taverna service invocation plugin. This is a Java based project which allows users to invoke the DIANA rest services through the Taverna WMS tools.

In DIANA Taverna plugin four Taverna activities have been created, one for each of the DIANA REST services. An activity in Taverna terminology is the way for Taverna to connect and invoke third party services, using each time the appropriate libraries and/or protocols. From now on the terms activity and service in Taverna context are interchangeable.

The prerequisites to use the DIANA Lab plugin is to download and install Taverna 2.3 or 2.5 and use the following plugin update sites:

For Taverna 2.5: `http://192.185.115.65/~diana/DIANA_plugin_updated/test-plugins/`

For Taverna 2.3: `http://192.185.115.65/~diana/DIANA_plugin_v2_3/test-plugins/`

1.4.3.1 DIANA-microT-CDS (v5) activity

DIANA-microT-CDS (v5) activity is used by taverna to invoke the DIANA-microT-CDS (v5) rest service. The activities are visualized in the Taverna workbench. The DIANA-microT-CDS (v5) activity is depicted in Figure 2. The user must specify at least one of the input ports and at least one output port for the service to be executed properly.

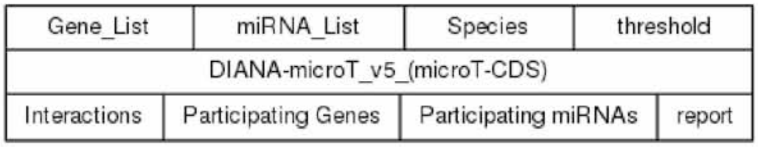
The input ports of the activity are the following:

- **Gene_List**: one or more genes using gene name or Ensembl 69 gene ids, eg PTEN, ENSG00000171862
- **miRNA_List**: one or more miRNAs using miRNA names OR MIMAT ids eg hsa-miR-24-3p, MIMAT0000080
- **threshold**: prediction score cut off value, ranging from 0.3 to 1. Default is 0.7

- **Species**: Species supported by the web server in org code, alias or full name format, eg hsa, human or homo sapiens

The output ports are:

- **Interactions**: miRNA-gene predicted interactions
- **Participating Genes**: Ensembl v69 gene ids of the predicted targets
- **Participating miRNAs**: miRNAs taking part in the predicted interactions in mature miRNA nomenclature (miRBase v18)
- **report**: General information regarding the results

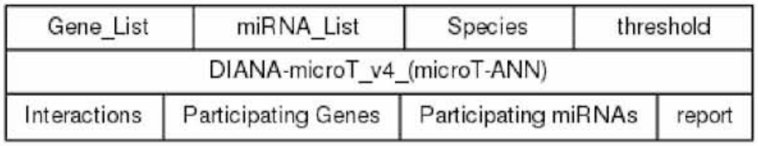


Gene_List	miRNA_List	Species	threshold
DIANA-microT_v5_(microT-CDS)			
Interactions	Participating Genes	Participating miRNAs	report

Figure 2: DIANA-microT-CDS (v5) activity

1.4.3.2 DIANA-microT-ANN (v4) activity

DIANA-microT-ANN activity invokes DIANA-microT-ANN rest service and has the same input and output ports as DIANA-microT-CDS activity, following the same conventions. Its visualization is depicted in Figure 3:

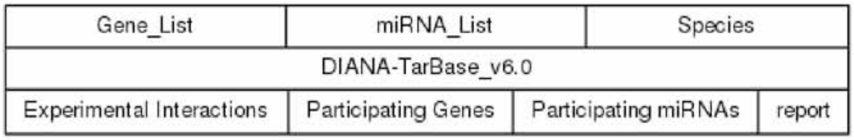


Gene_List	miRNA_List	Species	threshold
DIANA-microT_v4_(microT-ANN)			
Interactions	Participating Genes	Participating miRNAs	report

Figure 3: DIANA-microT-ANN (v4) activity

1.4.3.3 DIANA-TarBase v2.0 activity

DIANA-TarBase v2.0 activity is used by taverna to invoke the DIANA-TarBase v6.0 rest service. Its input and output ports descriptions are equivalent to the ones for DIANA-microT-CDS activity. The difference is that no prediction score cut off value is supported, as no pridictions take place.



Gene_List | miRNA_List | Species

DIANA-TarBase_v6.0

Experimental Interactions | Participating Genes | Participating miRNAs | report

Figure 4: DIANA-TarBase v2.0 activity

1.4.3.4 DIANA-miRPath v2.1 activity

This activity invokes the DIANA-miRPath REST Service and is depicted in Figure 5.

The input ports of the activity are the following:

- **Gene filtering**: A URL to a file containing a predefined list of genes expressed in investigated tissues. miRNA targets based on this list are filtered and only the expressed subset of genes is used for the pathway enrichment analysis.
- **List miRNA** - Validation: One or more miRNAs in miRNA name format, each followed by TarBase or microT-CDS method. If no method is specified or is not supported microT-CDS is used.
- **Merging Genes**: union, meaning all genes targeted by at least one input miRNA or intersection, meaning genes targeted by all selected miRNAs.
- **Merging Pathways**: union, meaning identification of all the significantly targeted pathways by the selected miRNAs indivindually or intersection meaning pathways with statistically significant results for all the selected miRNAs

- **Species**: Species under investigation, human or mouse.
- **Statistics Conservative**: true/false meaning use or not a conservative adjustment of Fisher's exact test
- **Statistics FDR**: true/false meaning False Discovery Rate enabled or disabled
- **threshold_microT**: prediction score cut off value, ranging from 0.3 to 1, applicable when microT-CDS is used.

The output ports are:

- **miRNAs-Pathways**: Pathway KeggIds significantly targeted by the input miRNA(s) along with related information like Pathway description, number and names of associated genes, pValue and participating miRNAs.
- **Pathways**: Only the pathway KeggIds significantly targeted by the input miRNA(s).
- **report**: General information about the results

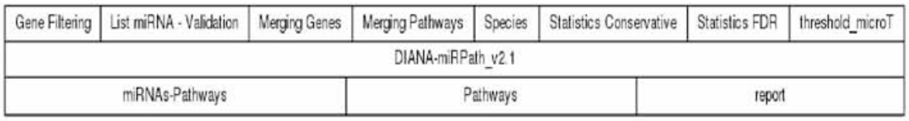


Figure 5: DIANA-miRPath v2.1 activity

2 Materials and Tools

2.1 Data

Data used for querying the REST services are combinations of miRNA and/or gene names or supported identifiers and come from miRBase v18 [54] and Ensembl v69 [55]. miRBase is a microRNA database and its v18 version contains 18226 entries of hairpin precursor miRNAs and 21643 mature miRNA products spanning in 168 species. The Ensembl project provides genome resources for chordates. It focuses on human genome data and currently supports 70 species.

2.2 Taverna Service Invocation Plugin

In order to invoke a service in Taverna, that is not of an already supported type, a service invocation plugin project has to be created. In the following sections exists a brief description of the steps needed to create such a project having as reference the DIANA Taverna plugin.

2.2.1 Project Creation

The first step is to create a template project from a maven archetype. This project contains the basic structure and functionality of a taverna service invocation plugin.

Apache Maven is a software project management and comprehension tool that can be used for building and managing Java-based projects. It provides between others a standard way to build projects and a clear definition of what the project comprises. In addition it provides a way to create project templates and also has deployment capabilities. These functionalities are organized and defined in pom.xml file [56]. Taverna service invocation plugin projects are created, build and deployed using maven.

The steps to create a new Taverna template service invocation plugin project from the command line are the following. First the command below must be issued.

```
mvn archetype:generate
    -DarchetypeCatalog=http://www.mygrid.org.uk/maven/repository/
```

The -D option specifies the remote catalog which hosts the wanted template project list. After the execution of the command, a command line wizard initiates which asks the user preferences about the version, the groupId, the artifactId and the basic package name of the Java project. The groupId and artifactID properties are maven specific where groupId specifies the id of the projects group, eg the company or organization of the team creating the project and the artifactId is the id of the project. More information regarding groupId and artifactId conventions can be found here: `https://maven.`

`apache.org/guides/mini/guide-naming-conventions.html`. After setting these properties the template project is created and can be imported in eclipse ide as a maven project.

2.2.2 Service Configuration

The basic class in the plugin is the one that implements the AsynchronousActivity interface. This class contains the basic implementation for connecting and invoking a service as well as processing the service results. In DIANA plugin four classes implement the AsynchronousActivity interface. These are DianaMicroT, DianaMicroTCDS, DianaMiRPath and DianaTarbase, one for every service.

When an activity like DianaTarbase is added to a workflow the default constructor creates an instance of it. After that the configure() method is called which performs the configuration of the activity using a configuration bean. For example DianaTarbase is implementing AsynchronousActivity<ConfigurationBean>, meaning that the configure method has a DianaConfigurationBean object as a parameter. A configuration bean typically contains basic configuration details of an activity instance that are needed for the activity to be able to call the service. For example the network address of the remote server hosting the services. The last configuration bean passed to configure() can be accessed via the getConfiguration() method.

Finally in configure() method the input and output ports of the activity are created. Both input and output ports can be more than one. The input ports is the place where the user passes the appropriate parameters that the activity needs to call the service. The output ports is the place where the activity provides the results of the service invocation to the user. All ports need to be specified at configuration but they do not need to be connected at execution time. However, at execution time all connected ports must have values assigned for the workflow to execute. In the below code segment is depicted the configuration of input and output ports of DianaTarbase activity.

```
protected void configurePorts() {
      private static final String GENE_INPUT = "Gene_List";
      private static final String MIRNA_INPUT = "miRNA_List";
      private static final String SPECIES = "Species";
```

```
     private static final String GENES_OUTPUT = "Participating
         Genes";
    private static final String MIRNAS_OUTPUT = "Participating
        miRNAs";
    private static final String INTERACTIONS_OUTPUT =
        "Experimental Interactions";
    private static final String REPORT_OUTPUT = "report";
    ...
    removeInputs();
    removeOutputs();

    addInput(GENE_INPUT, 1, true, null, byte[].class);
    addInput(MIRNA_INPUT, 1, true, null, byte[].class);
    addInput(SPECIES, 0, true, null, String.class);

    addOutput(INTERACTIONS_OUTPUT, 0);
    addOutput(REPORT_OUTPUT, 0);
    addOutput(GENES_OUTPUT,0);
    addOutput(MIRNAS_OUTPUT,0);
  }
```

At first the input and output port names are declared as constants to avoid misspelling. Methods removeInputs() and removeOutputs() remove existing input and output ports to avoid duplicates when the activity is reconfigured. The addInput() and addOutput() methods create and add a new input port and a new output port respectively using the provided arguments. The more important arguments are the first two of both methods and the last of the addInput() method. The first is the name of the port and the second is the depth of the port. A port with depth 0 will expect as input an individual value whereas a port with depth 1 will expect a list of values, a port with depth 2 expects a list of lists of values and so on. Finally the last parameter of addInput() method indicates the type of value that this input port expects the incoming data to be. For example, the species port is expecting an individual value which should be retrieved as a String.

2.2.3 Service Invocation

The invocation of the service is performed in executeAsynch() method. This method has two arguments, a map named inputs and a callback object. The inputs map has as keys the names of the input ports and as values references to the corresponding data that the user provided. The callback object performs the actual invocation by requesting a Runnable to run.

```
public void executeAsynch(final Map<String, T2Reference> inputs,
     final AsynchronousActivityCallback callback) {
   callback.requestRun(new Runnable() {

     public void run() {
        InvocationContext context = callback.getContext();
        ReferenceService referenceService =
            context.getReferenceService();

        if(inputs.get(GENE_INPUT)!=NULL) {
              T2Reference geneRef = inputs.get(GENE_INPUT);
              geneInput =(List<byte[]>)
                  referenceService.renderIdentifier(geneRef,byte[].class,
                  context);
              ...
           }

           if(inputs.get(MIRNA_INPUT)!=NULL) {
           T2Reference miRNARef = inputs.get(MIRNA_INPUT);
           miRNAInput = (List<byte[]>)
               referenceService.renderIdentifier(miRNARef,byte[].class,
               context);
           ...
        }

        if(inputs.get(SPECIES)!= NULL) {
            T2Reference speciesRef = inputs.get(SPECIES);
                speciesInput = (String)
                    referenceService.renderIdentifier(speciesRef,
                    String.class, context);
                ...
            }
            ...
           callback.receiveResult(outputs, new int[0]);
     }

   });
}
```

Inside the run() method the ReferenceService is retrieved from the callback object and from that the actual data of the user inputs are resolved. If the input port is connected, then the inputs map will have a non null reference to user data. These data will be retrieved and used to call the service and process the service results. For example through the gene and miRNA inputs the user provides the query parameters to the rest service.

After the query URL is build, the rest service is invoked using java net package. Subsequently, XStream library is used to parse and convert the XML results to Java objects. In this way the results are more efficiently processed.

```
   try {
      query_stringResults = ServiceHelperMicroTCDS.httpGet(url);
      microTresult_info = (MicroTInfo)
          xstream.fromXML(query_stringResults.get(0));
  } catch (IOException e) {
      report = "Could not invoke Diana-Tarbase service: " +
          e.getMessage();
      results = "Could not invoke Diana-Tarbase service: " +
          e.getMessage();
      errorInvokingService = true;
      e.printStackTrace();
    }
```

Finally the output ports are populated and passed again to the callback object, again as a map with keys the names of the output ports and values the references to the corresponding output data.

```
    Map<String, T2Reference> outputs = new HashMap<String,
        T2Reference>();

   T2Reference simpleRef = referenceService.register(interactions,
       0, true, context);
   outputs.put(INTERACTIONS_OUTPUT, simpleRef);

    T2Reference moreRef = referenceService.register(report, 0,
        true, context);
    outputs.put(REPORT_OUTPUT, moreRef);

    T2Reference geneRef = referenceService.register(genes_output,
        0, true, context);
    outputs.put(GENES_OUTPUT, geneRef);

   T2Reference miRNARef = referenceService.register(miRNAs_output,
       0, true, context);
   outputs.put(MIRNAS_OUTPUT, miRNARef);
   callback.receiveResult(outputs, new int[0]);
```

2.3 Taverna flows

2.3.1 Taverna Workbench basic functionality

Taverna uses SCUFL programming language to implement the workflows internally. In SCUFL a workflow is a graph of processors and links. A processor can have zero or more inputs and zero or more outputs. A processor with no input is considered a data source and a processor with no output is considered a sink. Taverna also supports nested workflows which are workflows within another workflow.

In Figure 6 is depicted the Design perspective of Taverna Workbench Bioinformatics 2.5.0 version. It is divided in two main sections. The left section is the service panel where the user can browse new services and validate or view details of existing workflows. The right section is the workflow editor where the user can create new workflows or modify existing ones. Workflow editor allows the user to modify the alignment of the services in the workflow, the visibility of the service ports and whether the nested workflows are expanded or not. A user can drag and drop a new service from the service panel in order to add it to a workflow. Then it can connect it with workflow input ports, workflow output ports, or with another service port. When a user decides to run a workflow a run workflow dialog window appears where the user can insert the input values for the workflow ports. The dialog window also contains annotation fields, such as port description, example values, workflow description and information regarding the workflow author.

When the workflow is executed, the results perspective is activated. It contains elaborate information regarding the results of the workflow and interaction capabilities. Specifically a user can see the workflow final results, intermediate results even at runtime, progress reports and previous executions. In addition the workflow can be canceled or paused and resumed at a later time.

SCUFL is mainly a data-flow language but it integrates some control constructs. The basic control capabilities of Taverna flows are explained in the following sections.

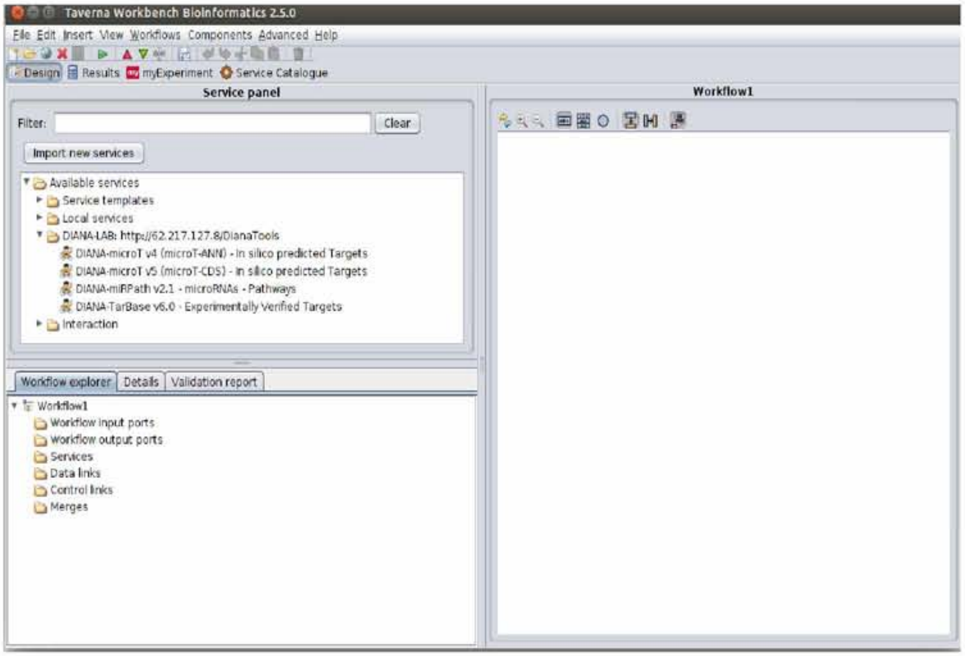

Taverna Workbench Bioinformatics 2.5.0
File Edit Insert View Workflows Components Advanced Help
Design Results myExperiment Service Catalogue
Service panel
Filter:
Clear
Import new services
Available services
Service templates
Local services
DIANA-LAB: http://62.217.127.8/DianaTools
DIANA-microT v4 (microT-ANN) - In silico predicted Targets
DIANA-microT v5 (microT-CDS) - In silico predicted Targets
DIANA-miRPath v2.1 - microRNAs - Pathways
DIANA-TarBase v6.0 - Experimentally Verified Targets
Interaction
Workflow explorer
Details
Validation report
Workflow1
Workflow input ports
Workflow output ports
Services
Data links
Control links
Merges
Workflow1

Figure 6: Taverna Worcbench version 2.5. Design Perpspective

2.3.2 Control links

Control links set dependencies between services that do not directly share data. It is a very useful construct when services that depend on an external factor compelling them to be executed in an orderly fashion have no direct data dependency to ensure that. This can be achieved in Taverna by right clicking a service and activating the Run After option selecting which service should it wait before enactment.

2.3.3 Looping

Taverna provides looping capabilities both implicitly and explicitly. The implicit iteration is based on an inherent Taverna property. Particularly, when an input port of a service expects a single item and receives a lis it is invoked for every element in the list and creates a list in the output, providing the results of the iteration. The same applies for nested workflows. Explicit looping makes iteration possible when it is not known beforehand the number of the required iterations. A user can right-click a service or a workflow, select Configure running and then Looping. A looping service dialog is activated where the user can determine the conditions that should be met for the iteration to stop. Also a delay time between the loops is specified.

2.3.4 Conditional invocation

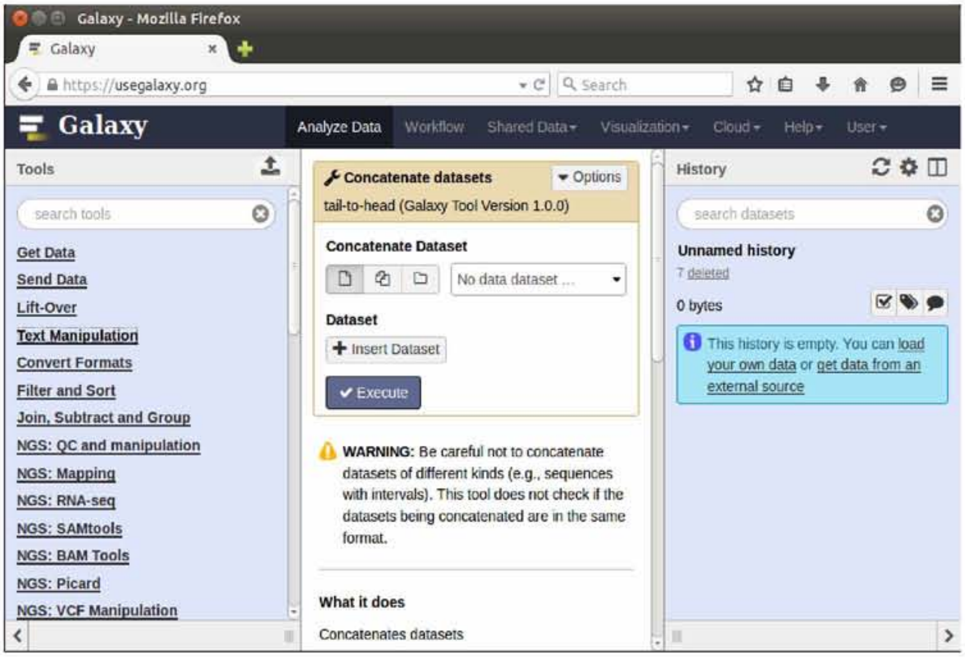
The conditional invocation relies on the same inherent property of Taverna. As mentioned above when a service or a nested workflow input port expects a single item but receives a list then it is invoked for every element in the list. Likewise, when it receives an empty list it will not be executed at all. So to achieve conditional invocation in Taverna, for every exclusive choice decision two nested workflows should be created. These workflows must have, between others, one input port of zero depth which depending on a previous step check will take as value either an empty or a non empty list. Only the workflow supplied with the non empty list will be executed. Equivalent results can be achieved with single nodes, such as Taverna services or Beanshell scripts provided that they have an input port of zero depth.

2.4 Galaxy

2.4.1 Basic functionality

The analysis workspace of the Main Galaxy Instance can be found in `https://usegalaxy.org/`, Figure 7. It is divided in four areas, the Navigation bar, the Tool panel, the Detail panel and the History panel. Navigation bar consists of links to the different Galaxy's sectors, such as data analysis, workflows and shared libraries. Tool panel lists the available analysis tools and datasources. The detail panel presents the interface for the selected tool or datasource. Finally the History panel contains Histories, a basic component of Galaxy. Histories present the results of user actions, called Datasets, as well as the tools or datasources used. Users can share Histories and they can be reused and edited. Users can also share, download and reuse Datasets. Another important component of Galaxy are Workflows. They can be extracted from Histories, describing the steps in an analysis process and can be reexecuted. Finally Pages is a Galaxy component which can contain all of the previous components along with documentation and scientific context, used mainly for learning and publication purposes. Main

Galaxy instance has a list of published Histories, Workflows, DataSets and Pages.



Concatenate datasets tail-to-head (Galaxy Tool Version 1.0.0)

Concatenate Dataset: No data dataset ...

WARNING: Be careful not to concatenate datasets of different kinds (e.g., sequences with intervals). This tool does not check if the datasets being concatenated are in the same format.

What it does

Concatenates datasets

This history is empty. You can load your own data or get data from an external source

Figure 7: Main Galaxy instance. Analysis workspace

Galaxy, does not use an explicit workflow language to implement workflows, but the required information of nodes and links is stored in a relational database. Like Taverna, a node in the workflow can have one or more input ports and one or more output ports. Galaxy however does not directly support control flow capabilities.

Galaxy tools can be easily shared by using the Galaxy ToolShed [47]. This can be accessed in `https://toolshed.g2.bx.psu.edu/`. It is a service that hosts repositories containing Galaxy Tools and other useful utilities. A user can publish his tool there simply by creating an account, then creating a repository and finally uploading the tool files to the repository. When uploaded, other users can install the tools to their local Galaxy installations via the Admin interface.

2.4.2 DIANA Tools integration

To enable invocation of DIANA REST services from Galaxy, four tools were created, one for each service. First a script, usually a Perl script, invoking the service has to be written. The actual invocation of the REST service is performed by Taverna command line tool executing a predefined flow. So in order for the DIANA Galaxy tools to be functional a Taverna installation with the DIANA Taverna plugin is required. The microT-CDS.pl script is the following:

```
#!/usr/bin/perl
use strict;
use warnings;

my $genes = $ARGV[0];
my $output = $ARGV[1];

system "sh {PATH_TO_TAVERNA}/executeworkflow.sh -outputdir
    {OUTPUT_DIRECTORY} -inputvalue genes $genes
    {PATH_TO_FLOW}/microTCDS.t2flow";

my $filename = {OUTPUT_DIRECTORY}/interactions";

open OUT, ">$ARGV[1]" or die "Cannot open $ARGV[1]:$!\n";

open(my $fh, ’<:encoding(UTF-8)’, $filename) or die "Could not open
    file ’$filename’ $!";

while (my $row = <$fh>) {
  chomp $row;
  print OUT "$row\n";
}

system "rm -rf {OUTPUT_DIRECTORY}";

close( OUT );
```

The script is executed using two arguments. The one is the genes that user provides using gene names or Ensembl 69 gene ids. The output argument is used so the script can pass the results to Galaxy. Then the Taverna command line tool is used with arguments the directory where the results will be stored, the user input data and the workflow. After the enactment of the flow the file containing the results is printed to the output file and then the output directory is erased.

Galaxy tools also need an XML tool definition file. The microT-CDS.xml file has the follwing contents.

```
<tool id="microT-CDS_id" name="microT-CDS">
  <description>microT-CDS algorithm</description>
  <command interpreter="perl">microTCDS.pl $genes $output</command>
  <inputs>
    <param name="genes" type="text" label="Genes"
      help="Gene Names or Ensembl 69 gene ids seperated with ," />
  </inputs>
  <outputs>
    <data format="tabular" name="output"
      label="miRNA-genes predicted interactions" />
  </outputs>
  <help>
    This tool provides miRNA-genes predicted interactions using
        microT-CDS prediction algorithm
  </help>
  <citations>
    <citation type="doi">10.1093/nar/gkt393</citation>
    <citation type="doi">10.1093/bioinformatics/bts043</citation>
  </citations>
</tool>
```

The command element specifies the command that will be executed and in which programming language the tool is written. The command is any Linux shell command, in this case Perl, which runs with Galaxy user's permissions. Then the inputs are specified. In this case there is only one input named genes. The text value for the type attribute denotes that the user provides the input writing a string in a textbox in the Galaxy tool interface. In miRPath tool the value of the type attribute is data meaning that the input uses a file from a previous execution, if available. The label and help attributes are displayed in the tool page making it more readable and informative. Then the outputs are defined. In this example one output exists in tabular format. Tabular means one or more columns of text data separated by tabs. This format is viewable from the browser. Lastly citation information is provided.

For the new tool to be included in the tool panel a new directory has to be created under galaxy/tools directory, eg dianaTools. This directory contains the perl scripts and the XML tool definition files of DIANA Lab tools. Finally, the two tool registry files tool_conf.xml.main and tool_conf.xml.sample have to be updated with the new tool descriptions. The registry files may vary in different versions. The updated configuration is shown in the below code listing.

```
<section id="diana" name="DianaTools">
    <tool file="dianaTools/microTCDS.xml" />
    <tool file="dianaTools/microTV4.xml" />
    <tool file="dianaTools/tarbase.xml" />
    <tool file="dianaTools/mirpath.xml" />
</section>
```

The new tools will be functional, after a restart of the local Galaxy installation.

3 Results

3.1 DIANA Lab Taverna Plugin

In Figure 8 are depicted the basic improvements implemented in the DIANA Lab Taverna plugin. Different executions are shown, along with the corresponding report output presenting examples of the new capabilities.

For a more detailed presentation of the implementation improvements in the DIANA Taverna plugin a simple flow containing microT-CDS Taverna service is used. The results for microT-ANN and TarBase services are equivalent. Some differentiations exists with miRPath service which are mentioned. Reports revealing the new capabilities can be found in Appendix A.1.1 section.

3.1.1 Filtering results with species

When a user specifies only the gene input port, then the service has no information regarding the species. So if a gene has interactions, predicted

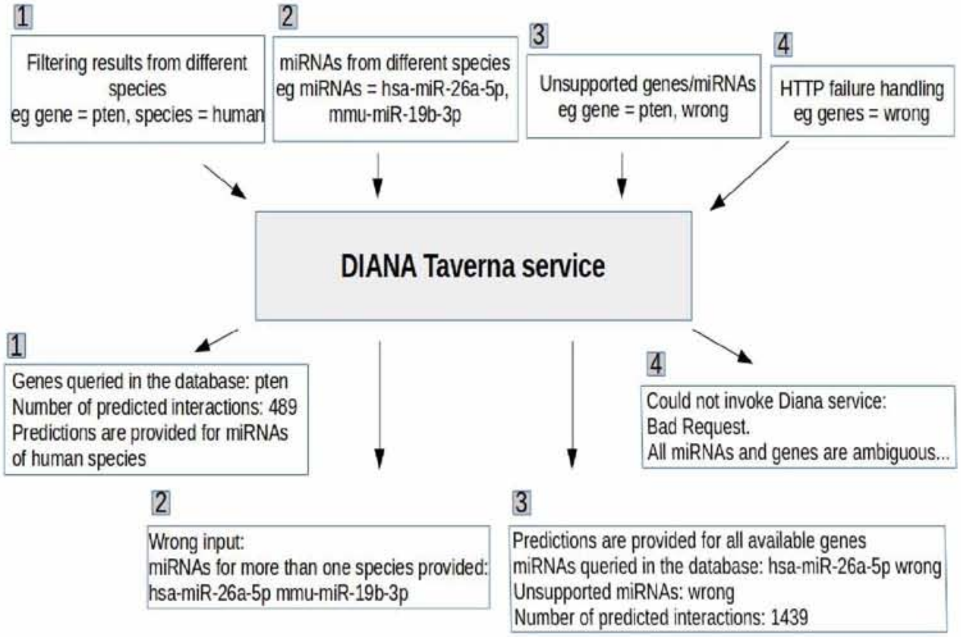

1
Filtering results from different species
eg gene = pten, species = human
2
miRNAs from different species
eg miRNAs = hsa-miR-26a-5p, mmu-miR-19b-3p
3
Unsupported genes/miRNAs
eg gene = pten, wrong
4
HTTP failure handling
eg genes = wrong
DIANA Taverna service
1
Genes queried in the database: pten
Number of predicted interactions: 489
Predictions are provided for miRNAs of human species
4
Could not invoke Diana service:
Bad Request.
All miRNAs and genes are ambiguous...
2
Wrong input:
miRNAs for more than one species provided:
hsa-miR-26a-5p mmu-miR-19b-3p
3
Predictions are provided for all available genes
miRNAs queried in the database: hsa-miR-26a-5p wrong
Unsupported miRNAs: wrong
Number of predicted interactions: 1439

Figure 8: Examples of Taverna plugin functionality improvements: The numbers denote different invocations and corresponding results. 1) Filtering results with species, 2) miRNAs from different species, 3) Unsupported genes/miRNAs, 4) HTTP failure handling

or experimentally validated, with mRNAs from more than one species the REST service API output provides interactions for all of them. microT-CDS, microT-ANN and TarBase Taverna services were modified to provide interactions only from one species. The algorithm with which the species value is determined is the following. First, if miRNAs are provided in miRNA name format then the species value is deducted from the three letter prefix of the miRNA name, e.g. hsa-miR-21-5p prefix is hsa which corresponds to human species. If no miRNA name is provided then the user can define the species using the new Species port. If Species port value is different from the species deducted from miRNA name, the latter prevails. Finally if none of the above can provide a valid species value, a default species list is used to filter the results. The first two values of this list are human and mouse. An example of this functionality is the first case of Figure 8. The microT-CDS algorithm provides 742 predicted interactions spanning three species, human,mouse and drosophila. By specifying the Species as human the interactions shown are limited to 489.

In miRPath, miRNA list in miRNA name format is a required input and the Species input port already exists, as it is used for the query to the REST API. The only modification implemented to improve the functionality is that if Species input port value does not match the species deduced from miRNA prefix, the latter prevails. Also if Species port input value is invalid or empty the species value is inferred from the miRNA prefix.

3.1.2 Unsupported genes/miRNAs

A user may provide as input for miRNAs or genes, values that are not supported. This can happen for various reasons, such as format mismatches or by mistake. The REST API results have two tags where the unsupported miRNAs and/or genes are listed. An example from the microT-CDS API is the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<microt_cds>
  <info>
    <miRNAs>hsa-miR-26b-5p wrongmirna</miRNAs>
    <genes>pten wronggene pten pten pten pten pten pten pten pten
        pten</genes>
    <result-size>1</result-size>
    <page-size>500</page-size>
    <current-page>1</current-page>
    <number-of-pages>1</number-of-pages>
    <time>0.036969900131226</time>
    <unsupported_miRNAs>wrongmirna</unsupported_miRNAs>
    <unsupported_genes>wronggene</unsupported_genes>
  </info>
  <results>
    <interaction rank="1" geneName="ENSG00000171862 (PTEN)"
      mirnaName="hsa-miR-26b-5p" score="0.999">
      <bindsite btype="7mer" location="18-46"
          score="0.0397612721018523" conservation="15" />
      ...
    </interaction>
  </results>
</microt_cds>
```

In all four Taverna services the implementation was modified to support the parsing of those elements, providing information about the unsupported genes and/or miRNAs in the report outputs. An example is the third case of Figure 8.

3.1.3 miRNAs from different species

A user is not supposed to query a Taverna service with miRNAs from different species. However the REST API provides results to such queries. Implementation was added in all the services checking if the supported miRNAs provided by the user belong to the same species. So first the query to the REST service is performed using all miRNAs provided by the user. Then a list is created containing only the supported miRNAs. This list is checked to verify whether it contains miRNAs from more than one species, using the miRNA name prefix. If a species inconsistency is detected then the loop breaks and the service terminates providing the user an informative error message. An example is the second case of Figure 8 where miRNAs from human and mouse are used. This functionality is not performed when the input miRNA list contains miRNAs in MIMAT id format.

3.1.4 HTTP failure handling

The java.net package is used to create HTTP GET requests to query the REST API through Java. The implementation is checking the HTTP response code and if it is 200 OK then the execution proceeds, otherwise it terminates giving the user an error message. Implementation was added so this message can be specific and not the same for all the cases that the REST API provides a code different than 200. Specifically the error response of the REST API is passed to a String variable and the specific reason and error code of failure are parsed. These values are then used to create the error message for the user. For example, in the fourth case of Figure 8, the microT-CDS service is invoked with genes equal to wrong and no other parameters. The REST API provides a 400 Bad Request HTTP code along with an informative message which is passed to the user output.

3.1.5 Duplicate genes

When the REST API of the services is queried with genes using gene name format, the genes tag in the result XML that shows the genes provided by the user contains duplicates. This for example can be seen in the XML output in Unsupported genes/miRNAs paragraph. Implementation was added so the report output will show only once the genes used for the query.

3.1.6 DIANA plugin version upgrade

After the plugin modifications the DIANA plugin was upgraded to 2.5 Taverna workbench version corresponding to Taverna software project 1.5 version. For this purpose a new Taverna plugin project was created using maven archetype from http://www.mygrid.org.uk/maven/repository/ remote catalog. Part of the command line maven project creation wizard is contained in the following listing.

```
$ mvn archetype:generate
    -DarchetypeCatalog=http://www.mygrid.org.uk/maven/repository/
...
[INFO] Generating project in Interactive mode
...
1: http://www.mygrid.org.uk/maven/repository/ ->
    net.sf.taverna.t2.archetypes:taverna-activity-archetype
    (Taverna service invocation plugin (activity))
Choose a number or apply filter (format: [groupId:]artifactId, case
    sensitive contains): : 1
Choose net.sf.taverna.t2.archetypes:taverna-activity-archetype
    version:
1: 1.0
2: 1.1
3: 1.2
4: 1.3
5: 1.5
Choose a number: 5: 5
...
Define value for property 'groupId': : gr.dianatools
Define value for property 'artifactId': : diana.services
Define value for property 'version': 1.0-SNAPSHOT: :
Define value for property 'package': gr.dianatools: :
Define value for property 'classPrefix': Example: : DIANA
Confirm properties configuration:
groupId: gr.dianatools
artifactId: diana.services
version: 1.0-SNAPSHOT
package: gr.dianatools
classPrefix: DIANA
 Y: : Y
```

```
[INFO]
    ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from
    Archetype: taverna-activity-archetype:1.5
[INFO]
    ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: gr.dianatools
[INFO] Parameter: artifactId, Value: diana.services
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: gr.dianatools
[INFO] Parameter: packageInPathFormat, Value: gr/dianatools
[INFO] Parameter: classPrefix, Value: DIANA
[INFO] Parameter: package, Value: gr.dianatools
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: groupId, Value: gr.dianatools
[INFO] Parameter: artifactId, Value: diana.services
...
[INFO] BUILD SUCCESS
```

The wizard asks first for which archetype from the remote catalog a project should be created. Then prompts the user to specify the project version. In this case the latest, 1.5 is used. Then the user determines the groupId, artifactId, project version and classPrefix values and the template project is created. After this process the implementation done in DIANA plugin 2.3 project was passed to the new project by making the necessary adjustments.

The command line wizard created a project called diana.services. This contains three subprojects, diana.services-activity, diana.services-activity-ui and diana.services-plugin. It also includes the parent pom.xml file which has general information, common configurations and dependencies for the subprojects. The diana.services-activity project contains the functionality of invoking the services, the diana.services-activity-ui contains mainly configuration details of the services as well as optional advanced capabilities. The diana.services-plugin project includes only a pom.xml file containing dependencies to the other two subprojects and the maven taverna plugin.

Taverna Workbench 2.5 provides a lot of new features, improvements and bug corrections comparing with 2.3 version, such as installation simplification, indication of looping configuration and many others. The Taverna Workbench Release Notes can be accessed in `http://www.taverna.org.uk/download/workbench/release-notes/#2.5.0`.

3.1.7 DIANA plugin deployment

The new Taverna plugins are deployed to a web server, which provides Taverna plugin update site URLs.

The deployment procedure has several steps. Specifically, if no other plugins are deployed in the specific server directory, the directory must be created beforehand and a pluginlist.xml containing just an empty plugins element must be put there. After the deployment of the project this file is updated.

In this case the project was deployed to the web server using FTP. For this purpose the following directives are configured in the parent pom.xml and the pom.xml of diana.services-plugin project.

First the web server is specified in the distributionManagement element of parent pom providing its ftp URL. Also in the same file an extension in the build element enables the use of FTP by specifying wagon-ftp library which is responsible for retrieving files from and deploying files to the server.

```
<project>
  ...
  <distributionManagement>
    <repository>
      <id>diana-plugin-site</id>
      <name>DIANA Plugin Site</name>
      <url>ftp://192.185.115.65/DIANA_plugin_updated/</url>
    </repository>
    <snapshotRepository>
      <id>diana-test-plugin-site</id>
      <name>DIANA Test Plugin Site</name>
      <url>ftp://192.185.115.65/DIANA_plugin_updated/test-plugins/</url>
    </snapshotRepository>
  </distributionManagement>
  ...
  <build>
  ...
    <extensions>
      <!-- Enabling the use of FTP -->
      <extension>
        <groupId>org.apache.maven.wagon</groupId>
         <artifactId>wagon-ftp</artifactId>
         <version>1.0-beta-6</version>
      </extension>
    </extensions>
  ...
  </build>
  ...
</project>
```

The ftp credentials are specified in maven settings.xml file, which is typically in $HOME/.m2 directory.

```
<settings>
...
  <servers>
    <server>
      <id>my-diana-plugin-site</id>
      <username>{ftpusername}</username>
      <password>{ftppassword}</password>
    </server>
  </servers>
</settings>
```

Again in parent pom, under the repositories element, the repositories where the project jars are deployed are specified. In the following snippet of parent pom two repository URLs are shown, one for the DIANA Repository where project releases are deployed and one for DIANA snapshot Repository where the project snapshots are deployed. These URLs, when the projects are deployed, will be the project update sites.

```
<repositories>
  ...
  <repository>
    <!-- The repository that your jars are deployed to -->
    <id>diana-repository</id>
    <name>DIANA Repository</name>
    <url>http://192.185.115.65/~diana/DIANA_plugin_updated/</url>
  </repository>
  <repository>
  <!-- The repository that your snapshot jars are deployed to -->
    <releases>
      <enabled>false</enabled>
    </releases>
    <snapshots />
    <id>diana-snapshot-repository</id>
    <name>DIANA snapshot Repository</name>
    <url>http://192.185.115.65/~diana/DIANA_plugin_updated/
    test-plugins/</url>
  </repository>
</repositories>
```

Finally in the parent pom.xml a build profile is created which customizes the deployment process by using the maven deploy plugin to include xstream library.

Maven taverna plugin specified at the pom.xml of diana.services-plugin project is using the above configurations to deploy the plugin to the remote site. Also it creates an XML file describing the plugin project, in this case it is called diana.services-plugin-1.0-SNAPSHOT.xml, and it is also uploaded to the server, Additionally it creates the pluginlist.xml file which updates the one already in the server's deployment directory.

With the above properly configured, the following commands must be executed in the project directory.

```
mvn install
mvn deploy -P deploy-xstream
```

The first compiles the source code, packages it into jar files, unless otherwise specified, and stores it into the local maven repository. Then mvn deploy -P deploy-xstream command is executed which copies the jars to the remote ftp server directory specified at the remote repository URL.

When this process completes successfully the remote repository URL is a fully functional update site.

3.2 DIANA Lab improved WorkFlows

In Figure 9 is depicted an example workflow provided by DIANA Lab. Figure 10 shows one of its nested subworkflows, used for the analysis in this section. In Appendix A.2.1 section are depicted all three workflows, with all the nested subworkflows modified to use the new Taverna plugin and to handle situations that caused them to fail.

The main improvement is the addition of control points checking if the output of specific steps match some criteria. If they match then the flow continues, otherwise it is terminated. This is achieved using the conditional invocation capability of Taverna. The control points are Beanshell nodes which contain JavaScript code using the Beanshell JavaScript language.

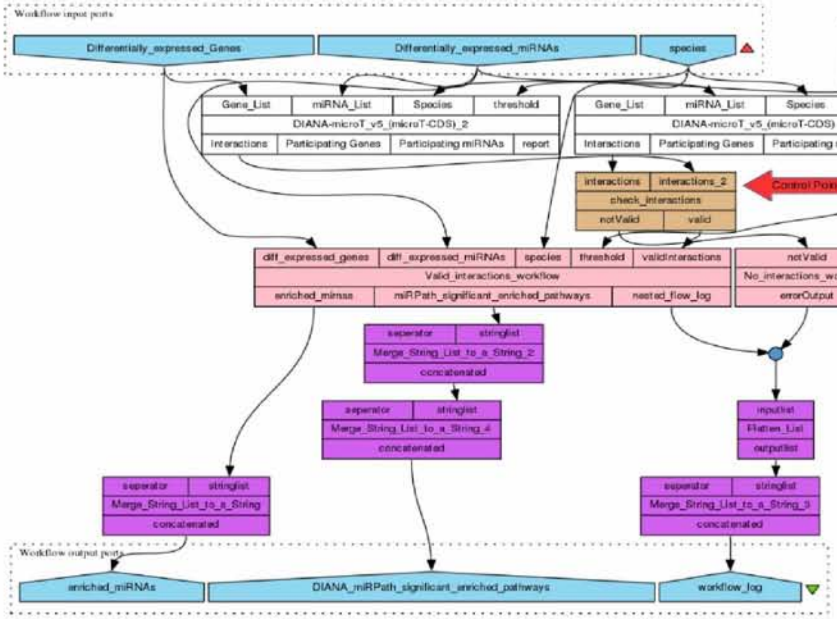


Figure 9: Example Workflow, performing enrichment analysis of predicted miRNA:gene interactions followed by targeted Pathway analysis. The red arrow denotes the control point deciding to which branch the flow should continue. The pink boxes are nested workflows, one of which is invoked, depending on whether both microT-CDS invocations have interactions.

3.2.1 DIANA Lab example workflow details

In this section workflow shown in Figure 8 will be examined in more detail in order to present its functionality and the improvements done. At first, for the workflow to execute the user provides a list of miRNAs, a list of genes and the species under investigation. Also a predetermined threshold equal to 0.7 is set for microT-CDS invocations. Using these values, two invocations of microT-CDS service take place, one with all the user input parameters and one with the miRNA list and the species values. This produces two interaction outputs. The one contains specific miRNA-gene interactions that input miRNAs have with the specified gene list while the other includes all interactions of the input miRNAs with genes of the specified species.

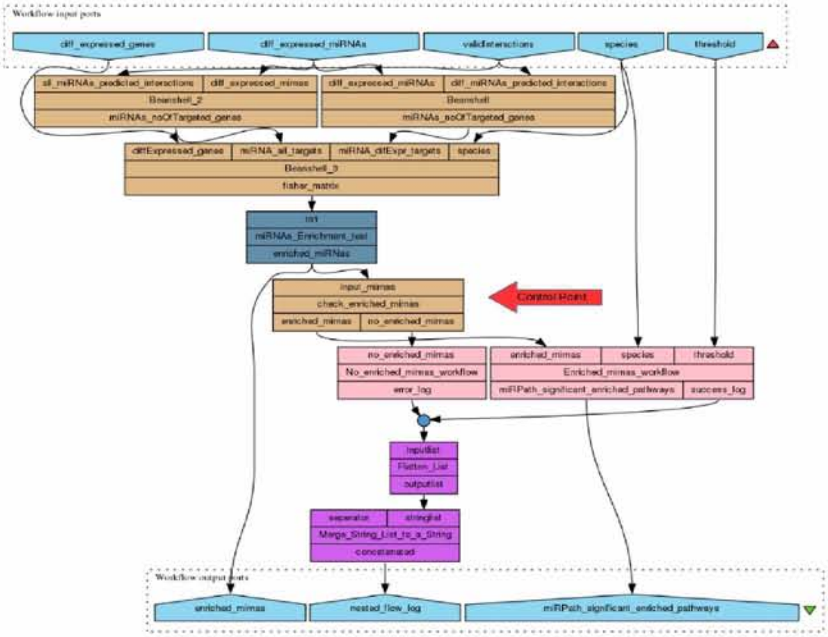


Figure 10: Valid_interactions_workflow nested flow. Invoked when both microT-CDS invocations have interactions. The red arrow denotes the control point deciding to which branch the flow should continue, depending on whether enriched miRNAs exist for the specific input.

In the case that no predicted interactions exist for at least one invocation, the flow should terminate with a descriptive error message. This functionality is incorporated using Beansell nodes as control points which split the flow into two nested subflows. Specifically Beanshell node named check_interactions has as input the interactions output of both microT-CDS invocations, two list outputs named valid and notValid and executes the following Javascript code to choose the branch that the flow should continue.

```
import java.util.ArrayList;

String temp1 = interactions.trim();
String temp2 = interactions_2.trim();

ArrayList valid = new ArrayList();
ArrayList notValid = new ArrayList();

if(temp1.toLowerCase().contains("ensemble gene id") &&
    temp2.toLowerCase().contains("ensemble gene id")) {
           valid.add(interactions + "splitParameter" +
               interactions_2);
} else {
          notValid.add(interactions + "splitParameter" +
              interactions_2);
}
```

If ”ensemble gene id” keyword exists in both input values, the script determines that both service invocations have results. As an effect, an element containing both interactions is added to the valid list output leaving the notValid list empty. If the keyword does not exist in at least one interaction, the valid list will be empty and the notValid will be populated. The valid list is fed as input to the zero-depth port called validInteractions of the Valid_interactions nested workflow which contains the rest workflow functionality. The notValid list is provided as input to the zero-depth port called notValid of the No_interactions nested workflow which includes only a Beanshell node creating the error output. Nested workflows that take as input in a zero depth port an empty list are not executed. Thus, conditional invocation is achieved enabling the termination of the flow when no adequate interaction results exist.

In the case that both microT-CDS invocations provide valid interaction output, the execution of Valid_interactions nested flow takes place, Figure 10. It has as input arguments, the original user provided input values, the predetermined threshold and the validInteractions input containing the previously predicted interactions. At first, two Beanshell nodes are invoked with arguments the user input miRNA list and the validInteractions input. The one node calculates the number of interactions that each provided miRNA has and the other the number of interactions that each miRNA has with the specific genes.

The results of these Banshells along with the original gene list and the species value are fed to another Beanshell node. This node produces a matrix, each row of which provides statistics for a participating miRNA. The first column is the name of the miRNA, the second column denotes the number of genes from the gene list that it does target and the third the number of genes from the gene list that it doesn't. Fourth column has as value the number of targets the miRNA has in all genes minus the number of targets in the specific gene list. Finally, the last column contains the difference between the total number of gene targets for the given species excluding from this the previous values.

Then, the matrix is provided as input to an R script, which calculates the p-value for each miRNA, a statistics measure expressing whether the miRNA-gene interactions have statistical significance or not. Particularly, if the calculated p-value of a miRNA is greater or equal to 0.05 the association is considered random and it is rejected. So the R script provides as output miRNAs considered to be statistically significant, called enriched miRNAs. Inside the R script, an if clause is inserted checking if the matrix input is empty. If it is then the R script terminates providing no enriched miRNAs value in its output port.

The output of the R script is leading to the second control point of this flow, the check_enriched_mirnas Beanshell. This Beanshell node checks if the R script output provides enriched miRNAs. Judging from that it populates the enriched_mirnas or the no_enriched_mirnas list output, activating in this way one of the nested workflows connected to them. If the enriched_mirnas output list is not empty then the nested workflow containing the miRPath service is executed leading to a successful workflow execution. Otherwise a nested workflow containing a Beanshell creating the error output is enacted.

In the case that enriched miRNAs do exist, they are provided as input to the miRPath service. Also the service is invoked using the predetermined threshold, the species, FDR enabled and union for merging genes input. This invocation provides pathways of the miRNA-gene interactions, along with supplementary info such as the p-Value of these interactions. To limit the pathways to the statistically significant ones, a Beanshell is used which filters the pathways with p-Value larger than 0.05.

Finally the statistically significant pathways are sent through data manipulation tools to the workflow output port DIANA_miRPath_significant_enric hed_pathways. The workflow has two more output ports, enriched_miRNAs and workflow_log. The first contains the enriched miRNAs calculated from the R script, while the last has information regarding the workflow execution.

When a nested workflow has list arguments as inputs it also creates list outputs. So, in order for the final workflow output to have single value outputs, data manipulation tools were used. For example for the workflow_log output port two local services were used, the Flatten_List and the Merge_String_List_to_a_String services. At first the error_output of No_inte ractions flow and the nested_flow_log output of Valid_interactions flow are merged. They are both lists from which the one is always empty and the other has a single value. So the result of the merge is a list of two sublists from which the one is empty. Then this list is fed to the Flatten List service which functionality is to flatten the provided input list by one level. In this case the list of lists becomes a simple list containing the elements of the sublists. Then this list is fed to the Merge_String_List_to_a_String service which has as output a String containing the values of the input list.

3.3 Galaxy integration

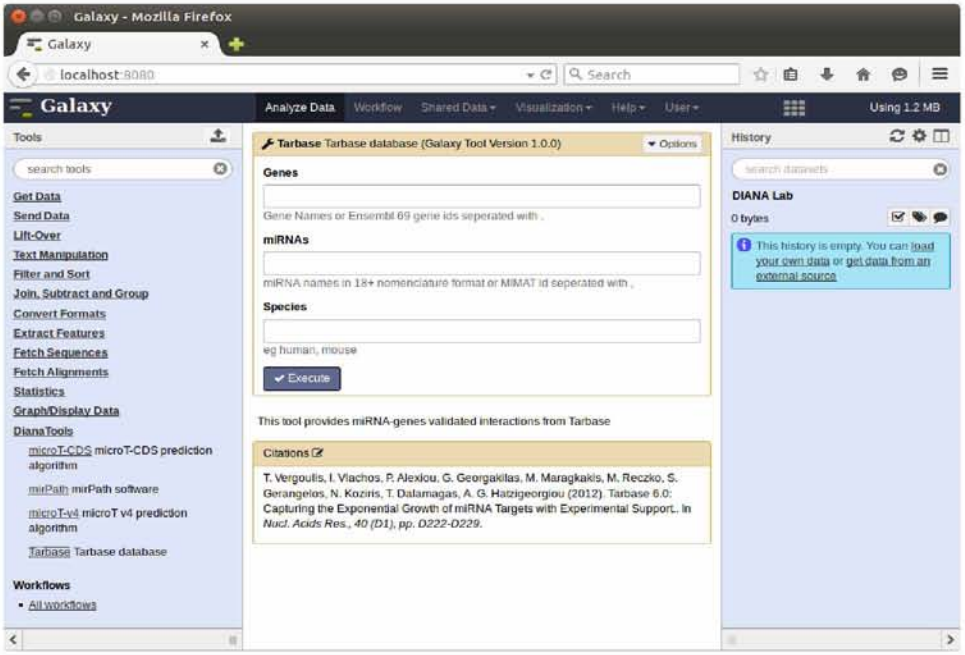
In Figure 11 is depicted the Galaxy interface of the DIANA Tarbase Galaxy tool. The input fields are shown, which are textboxes where the user can provide values, along with guidelines regarding the input. Also a small description for what the tool does is supplied as well as citation information. In miRPath interface there is a small differentiation as the input is not configured to be provided by the user but from the dataset of a previous execution.

These tools are operational but do not expose the full functionality of DIANA services, as they are not standalone tools but they typically represent Taverna flow invocations containing one DIANA service. With a single DIANA service more than one simple workflows can be created, since it has various inputs and outputs. For instance TarBase service, the one with the less input ports can be invoked in six different ways corresponding to six different flows. Solutions to this issue are discussed in later section. All DIANA Lab Galaxy tool interfaxes can be found in the A.3.1 section of the Appendix.

3.3.1 Using Galaxy Diana Lab Tools

A simple scenario will be presented which uses microT-CDS and miRPath tools. The scenario is to find targeted pathways of the 10 miRNAs with the biggest score on targeting PTEN gene, as it is calculated by the microT-CDS algorithm. This scenario does not aim to have biological significance but it is used for the presentation of the tools functionality. All steps are depicted in A.3.2 Appendix section.

First gene PTEN is provided to microT-CDS tool. Pressing execute the



Tarbase Tarbase database (Galaxy Tool Version 1.0.0)

Genes

Gene Names or Ensembl 69 gene ids seperated with ,

miRNAs

miRNA names in 18+ nomenclature format or MIMAT id seperated with ,

Species

eg human, mouse

This tool provides miRNA-genes validated interactions from Tarbase

T. Vergoulis, I. Vlachos, P. Alexiou, G. Georgakilas, M. Maragkakis, M. Reczko, S. Gerangelos, N. Koziris, T. Dalamagas, A. G. Hatzigeorgiou (2012). Tarbase 6.0: Capturing the Exponential Growth of miRNA Targets with Experimental Support.. In *Nucl. Acids Res., 40 (D1), pp. D222-D229.*

DIANA Lab

0 bytes

This history is empty. You can load your own data or get data from an external source

Figure 11: DIANA Lab Tarbase Galaxy Tool Interface

DIANA microT-CDS REST service is invoked through Taverna command line tool and a history item is added in the history panel denoting that the execution is under progress. When the execution finishes the history item is colored either green or red depending on the outcome of the process. In this case a successful execution has taken place and the user can click in the view data option on the history item to see the output. The output consists of three columns which are the name of the gene, the miRNA that targets it and the target prediction score. Then three data manipulation tools are used, one to sort the predicted-interactions output based on the score, column 3, one to select the first ten interactions and one to remove the gene and score columns and keep only the one with the miRNAs. These tools are invoked with input the output of the previous execution. Finally the output of the cut column tool which is a list of miRNAs is given as input to the miRPath tool which after the execution provides the output of the miRNAs-Pathways port of DIANA miRPath Taverna service, including pathway id and description.

3.3.2 Extract workflow in Galaxy

To extract a Wokflow from a History item the user must be logged into the Galaxy environment. Then the Extract Workflow option in the History panel must be selected. This action opens a window where the user can set the workflow name and decide which steps from the current history to incorporate in the workflow. After specifying these options the Create Workflow option is used. A new window is displayed, giving to the user the options to run or edit the workflow. To view the workflow and possibly modify some attributes the user can select the edit option, which opens the workflow canvas. This environment enables the user to modify the flow by adding or deleting links and nodes, changing the input values of the nodes and gives him other options as well. In this case the only modification made is to instruct the Galaxy Workflow to expect the gene name value to be provided at runtime instead of it having the value PTEN which was the value of the history item. Then the workflow can be saved and executed. The run option in the workflow canvas redirects to a run workflow dialog window. There, the steps of the workflow are presented. Also, the microT-CDS step provides a textbox where the user has to specify the input genes, eg TUSC2. Then the run workflow button executes the workflow, supplying the resulting Datasets in the previous History item or to a new one. This Workflow can be shared, published and downloaded making this test experiment fully reproducible for any gene(s) input. All steps described are depicted in the A.3.3 Appendix section.

4 Discussion

DIANA Lab provides many useful software tools to biology researchers to facilitate miRNA related tasks. Exposing them as web services has provided many benefits, such as ease of access and interoperability. However further improvements can be made.

The REST API can provide additional information concerning miRNAs and genes. For instance, if the web services provide information regarding ensembl transcripts, the interoperability with tools that use such information will be enabled. For example Get Pathways Galaxy tool developed by Bedoya-Reina et al. [57] gives KEGG pathways for input Ensembl transcripts. In addition, many Galaxy tools need as input genomic location coordinates. So for better integration with Galaxy, the REST API should provide such information enabling the Taverna services to have a BED output, for instance. BED is a data format, widely used in Galaxy, providing genomic coordinates by its three required fields: the name of the chromosome on which the genome feature exists and its starting and ending positions in the chromosome. It has also other nine optional fields, specifying more details. In addition, the REST API can be enhanced to support more DIANA Lab software, for example LncBase which is a database hosting elaborated information regarding miRNA-lncRNA predicted and experimentally validated interactions [58]. Also it should be updated to expose the newer version of TarBase v7.0 and latest miRPath. The new services can be combined in sophisticated workflows providing new insights, for example on miRNA-lncRNA interactions.

Some improvements in Diana Taverna plugin Java project implementation can take place. For example input validation for miRNAs and genes based on pattern matching. In this way miRNAs or genes that do not match some basic rules will not be used for the query and the user will be informed. For example wrongmirna value for miRNA which was used in a previous example will not be used for the query as it does not match the naming conventions of miRNAs either in miRNA name or MIMAT id format.

DIANA Lab Galaxy tools presented previously should be upgraded to support more inputs, outputs and options, in order to expose to Galaxy users the basic functionality of DIANA services. Since for every different combination of user input ports a different flow is needed, a solution can be to include along with the tool files a number of flows exposing the basic variations of the service invocations. Then the Perl script depending on the user input can determine which flow should be executed.

Sharing the DIANA Galaxy tools via the Tool Shed must be complemented with additional material. Specifically the flows that the command line tool will invoke must be provided along with instructions on where to store them. A more user friendly solution is to package the tools and the flows, along with bash scripts which will be responsible for installing the tools to the local installation and ensuring their functionality. This zip file will be disseminated to users along with simple instructions.

For local Galaxy installations bash scripts can be developed to provide an automated way for the user to generate a Galaxy tool for every flow that he creates. Specifically bash scripts having as input the flow and other details like where to store the temporary data can create the Perl script, the tool definition file and update the tool registry. In this way after the restart of

Galaxy the new tools will be fully functional.

DIANA Galaxy tools can be modified to execute specified Taverna workflows remotely, in a Taverna server. In this way the user saves computational resources and does not need to have workflows stored locally or the Taverna command line tool installed. Though the user must have the Taverna client installed to invoke the workflows in the Taverna server. In [59] an automated tool written in ruby is developed which creates a Galaxy tool from a Taverna flow and executes it into a remote Taverna server.

5 Conclusion

It is evident that research related to miRNAs is rising exponentially, proving the importance of this recently discovered non-coding RNA subset. As a result, related data volume is also increased, making storage and computations extremely heavy. These facts clearly state the absolute need for the constant improving of the software tools offering and manipulating miRNA related data. Also, exposing these functionalities as web services provides many advantages. Among others, it promotes interoperability and integration into diverse systems.

As a result of this thesis, the improvement of the Taverna plugin has provided a more stable environment for the biology researchers to combine the services into workflows and to receive expected and informative output from the flows executions. In addition, the redesign of the DIANA Lab workflows avoids unnecessary service invocations when no adequate results are available from a previous step, thus saving computational effort and providing more informative messages. Finally the integration of the functionality of DIANA web services into Galaxy makes them easily accessible through a web portal to a more wide range of biology researchers. In Taverna and Galaxy, the services can be used autonomously or combined with other from different organizations giving in this way new insights to the research.

Appendices

A Results

A.1 DIANA Taverna plugin

A.1.1 microT-CDS Taverna service

1) Providing results for human species, first in the default list.

input:

```
genes=pten
```

report output:

```
MicroT-CDS results:
Genes queried in the database: pten
DIANA-microT utilized threshold: 0.7 (default)
Number of predicted interactions: 489
Number of predicted interactions filtered due to species inconsistency: 253
Predictions are provided for miRNAs of homo sapiens species (Default list)
```

2) Providing results for human species, as specified.

input:

```
genes=pten species=homo sapiens threshold=0.7
```

report output:

```
MicroT-CDS results:
Genes queried in the database: pten
DIANA-microT selected threshold: 0.7
Number of predicted interactions: 489
Predictions are provided for miRNAs of homo sapiens species
```

3) No results for specified species

input:

```
gene=tusc2 threshold=0.995 species=mmu
```

report output:

```
MicroT-CDS results:
Genes queried in the database: tusc2
DIANA-microT selected threshold: 0.995
Number of predicted interactions: 0
No predictions for mus musculus species
```

4) Unsupported gene

input:

```
gene= tusc2,wrong
```

report output:

```
MicroT-CDS results:
Genes queried in the database: tusc2 wrong
Unsupported genes: wrong
DIANA-microT utilized threshold: 0.7 (default)
Number of predicted interactions: 74
Number of predicted interactions filtered due to species inconsistency: 19
Predictions are provided for miRNAs of homo sapiens species (Default list)
```

5) Unsupported miRNA

input:

```
miRNAs = hsa-let-7a-5p,wrong
```

report output:

```
MicroT-CDS results:
Predictions are provided for all available genes
miRNAs queried in the database: hsa-let-7a-5p wrong
Unsupported miRNAs: wrong
DIANA-microT utilized threshold: 0.7 (default)
Number of predicted interactions: 987
```

6) miRNAs from different species

input:

```
miRNAs = hsa-let-7a-5p,mmu-miR-486-5p
```

report output:

Wrong input: miRNAs for more than one species provided:
hsa-let-7a-5p mmu-miR-486-5p

7) HTTP: 400 Bad request
input:
miRNAs =wrong

report output:
Could not invoke DIANA-microT-CDS service: Bad Request. All miRNAs and genes are ambiguous...

8) Connectivity issues
input:
whatever but no connection to the internet.

report output:
Could not invoke DIANA-microT-CDS: Network is unreachable

9) Unsupported species
input:
miRNAS = mmu-miR-486-5p, genes=pten, species = cat

report output:
WARNING: Species provided: cat is invalid and will not be taken into consideration
MicroT-CDS results:
Genes queried in the database: pten
miRNAs queried in the database: mmu-miR-486-5p
DIANA-microT utilized threshold: 0.7 (default)
Number of predicted interactions: 1

10) Species different from miRNA prefix
input:
miRNAS = mmu-miR-486-5p, genes = pten,tusc2, species = hsa

report output:

MicroT-CDS results:
Genes queried in the database: tusc2 pten
miRNAs queried in the database: mmu-miR-486-5p
WARNING: Species homo sapiens doesn't match with the provided miRNAs and will not be taken into consideration
DIANA-microT utilized threshold: 0.7 (default)
Number of predicted interactions: 1

11)miRNAs from different species but one in MIMAT id format. No check made.
input:
mirnas=mmu-miR-19b-3p,MIMAT0000080 species=mmu

Some miRNAs are provided in MIMAT id format
MicroT-CDS results:
Predictions are provided for all available genes
miRNAs queried in the database: mmu-miR-19b-3p MIMAT0000080
DIANA-microT utilized threshold: 0.7 (default)
Number of predicted interactions: 1878

A.2 DIANA Lab Taverna pipelines

A.2.1 Improved Workflows

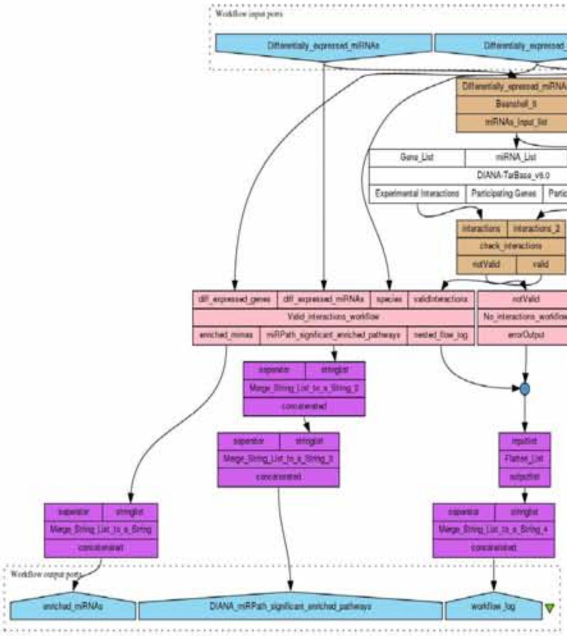


Figure 12: Example Workflow, performing enrichment analysis of validated miRNA:gene interactions followed by targeted Pathway analysis

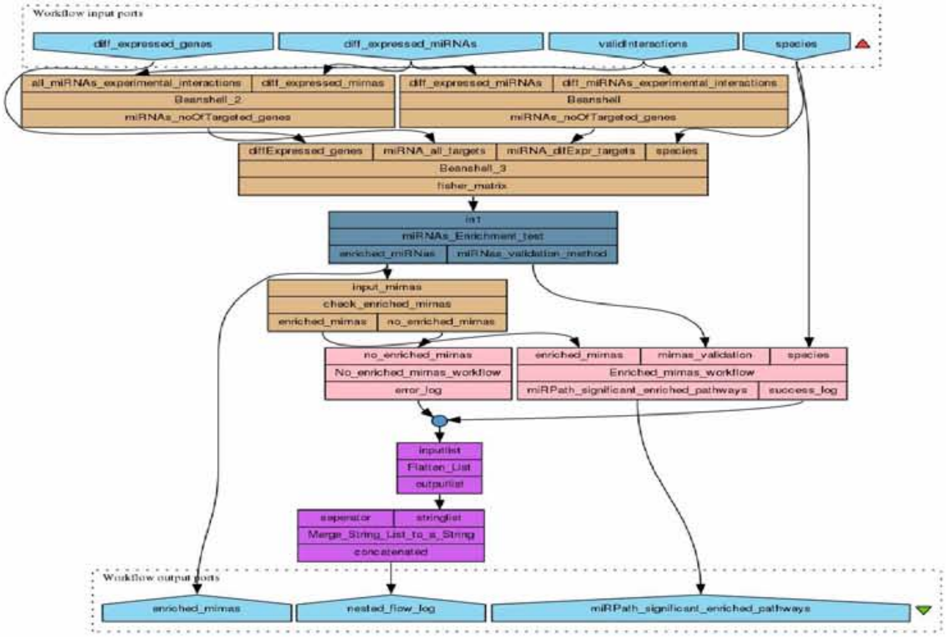


Figure 13: Valid interactions nested subworkflow of Example Workflow, performing enrichment analysis of validated miRNA:gene interactions followed by targeted Pathway analysis

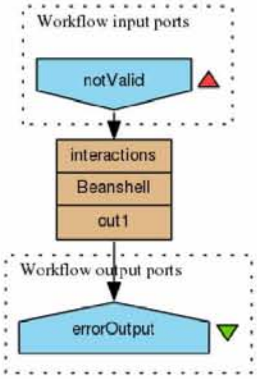


Figure 14: No interactions nested subworkflow, used in all three DIANA example workflows

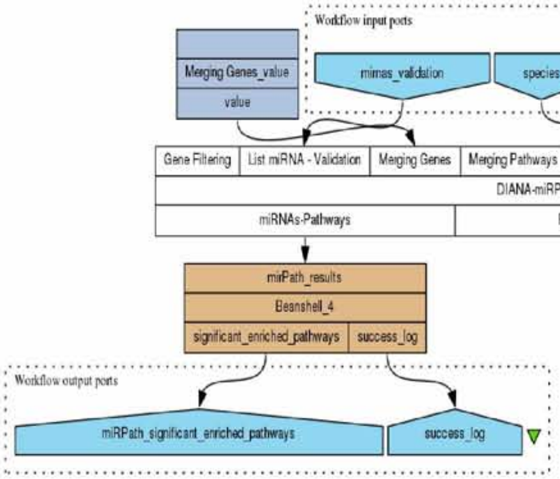


Figure 15: Enriched miRNAs nested subworkflow of Example Workflow, performing enrichment analysis of validated miRNA:gene interactions followed by targeted Pathway analysis

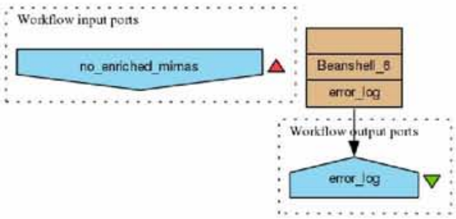


Figure 16: No Enriched mirnas nested subworkflow used in both DIANA example workflows performing enrichment analysis

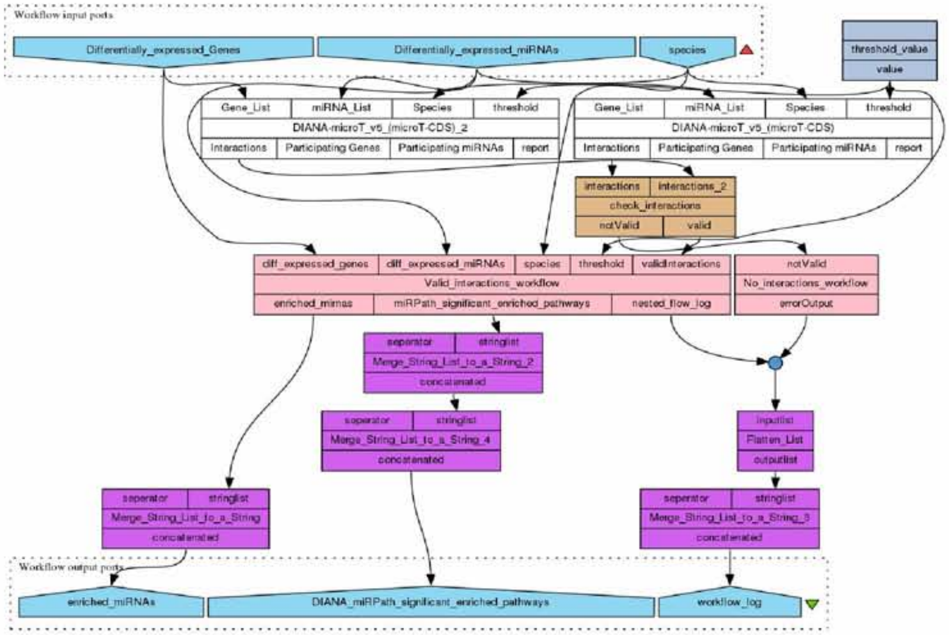


Figure 17: Example Workflow, performing enrichment analysis of predicted miRNA:gene interactions followed by targeted Pathway analysis

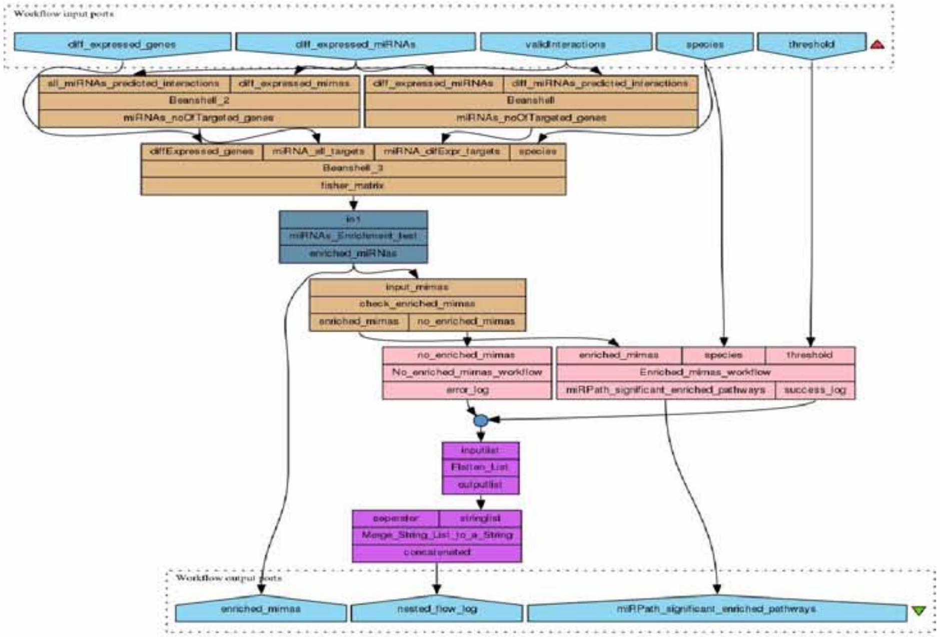


Figure 18: Valid interactions nested subworkflow of Example Workflow, performing enrichment analysis of predicted miRNA:gene interactions followed by targeted Pathway analysis

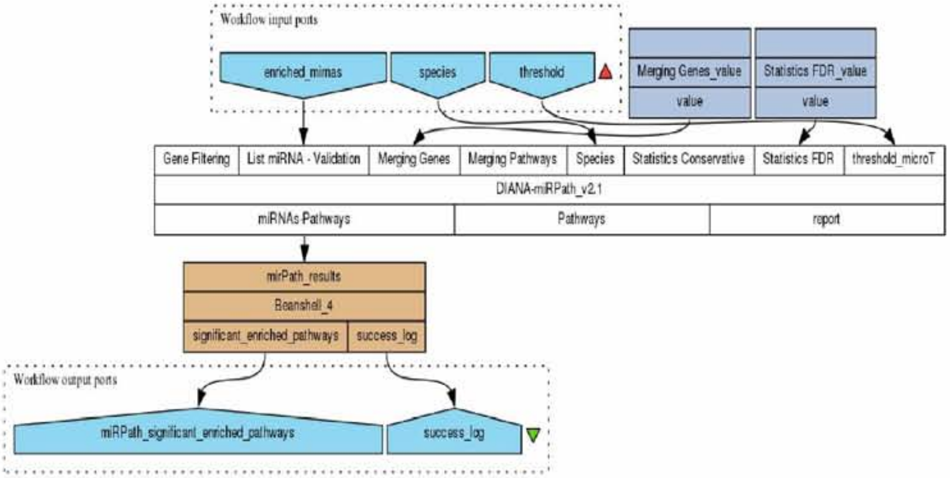


Figure 19: Enriched mirnas nested subworkflow of Example Workflow, performing enrichment analysis of predicted miRNA:gene interactions followed by targeted Pathway analysis

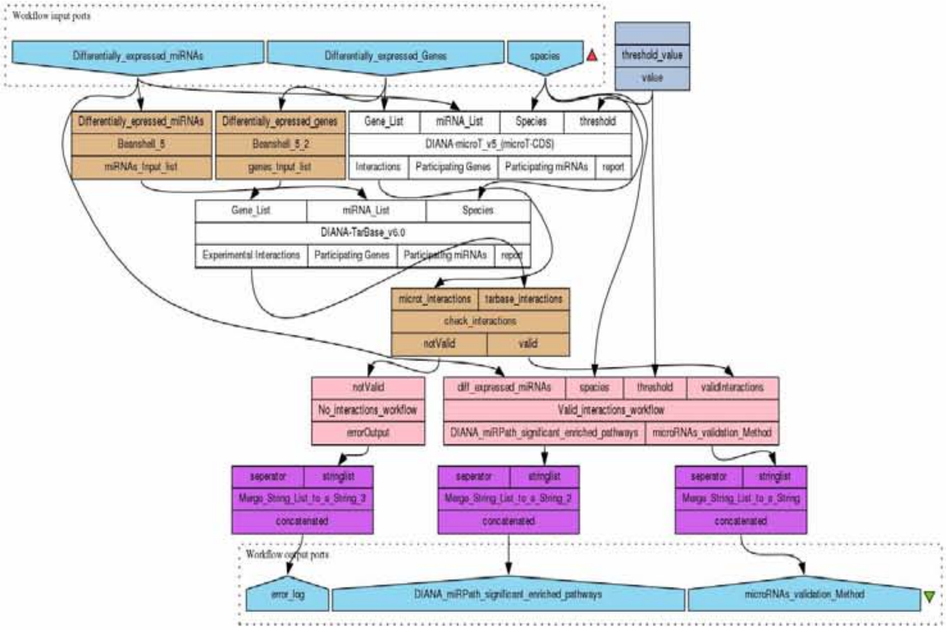


Figure 20: Example Workflow, "personalizing" the selection of miRNA-specific validated/predicted interactions, followed by miRNA Pathway analysis.

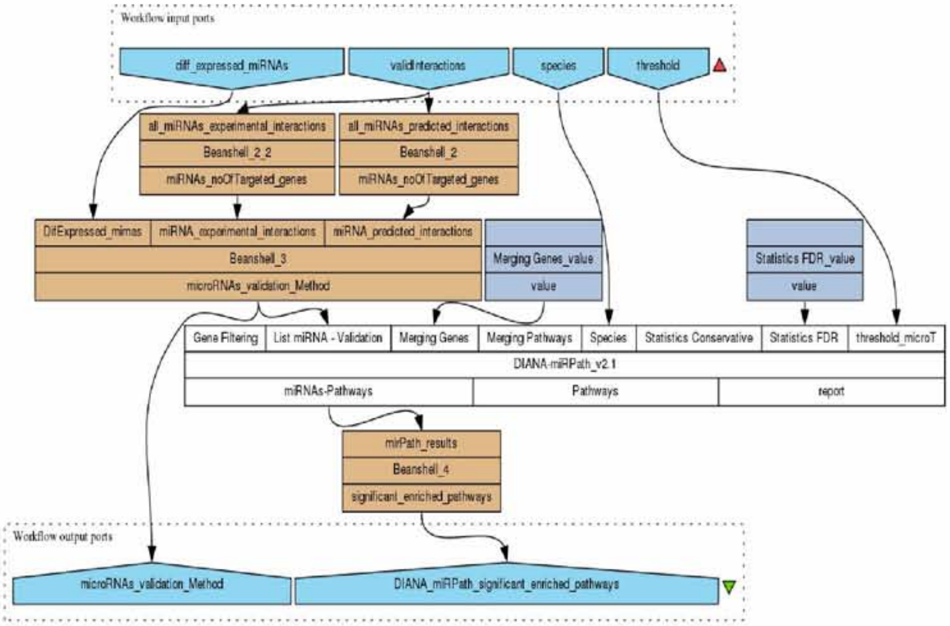
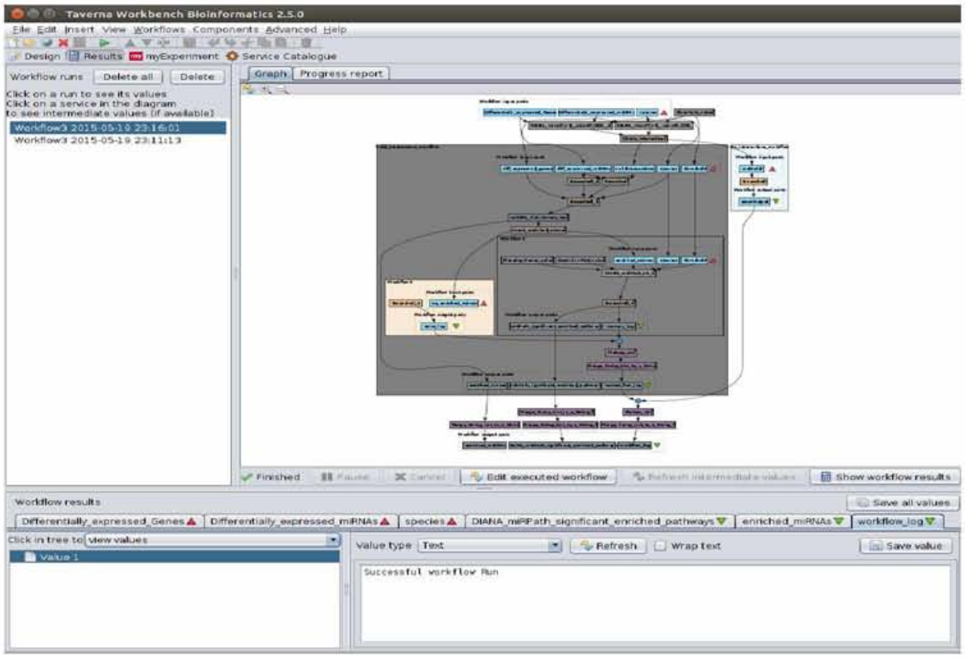


Figure 21: Valid interactions nested subworkflow of Example Workflow, "personalizing" the selection of miRNA-specific validated/predicted interactions, followed by miRNA Pathway analysis.

A.2.2 Workflow Results

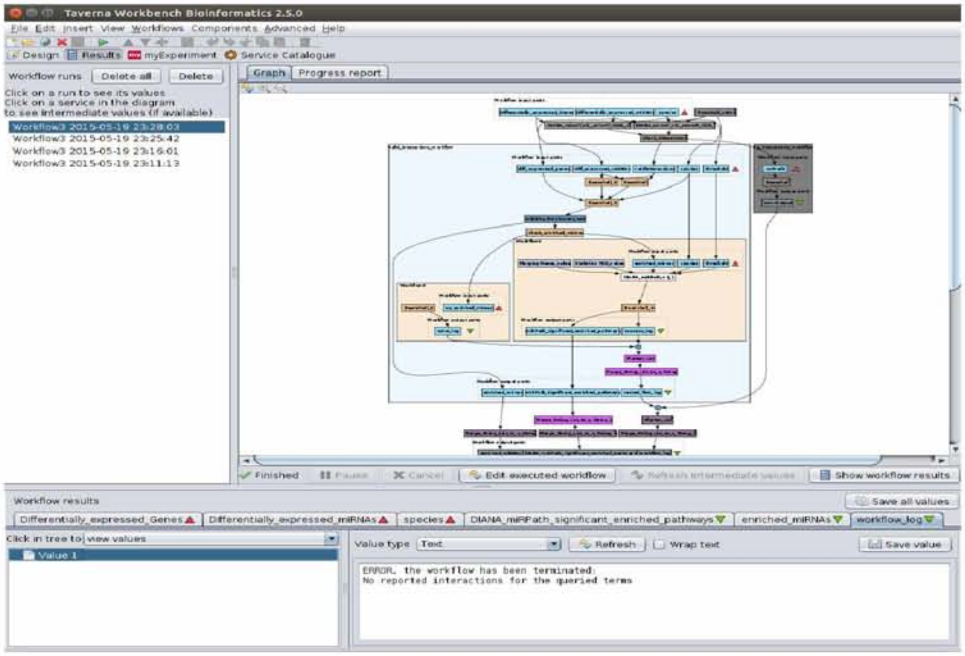

Taverna Workbench Bioinformatics 2.5.0
File Edit Insert View Workflows Components Advanced Help
Design Results myExperiment Service Catalogue
Workflow runs Delete all Delete
Graph Progress report
Finished Pause Cancel Edit executed workflow Refresh intermediate values Show workflow results
Workflow results Save all values
Differentially_expressed_Genes Differentially_expressed_miRNAs species DIANA_miRPath_significant_enriched_pathways enriched_miRNAs workflow_log
Click in tree to view values
Value 1
Value type Text Refresh Wrap text Save value

Click on a run to see its values
Click on a service in the diagram
to see intermediate values (if available)

Workflow3 2015-05-19 23:16:01
Workflow3 2015-05-19 23:11:13

```
Successful workflow Run
```

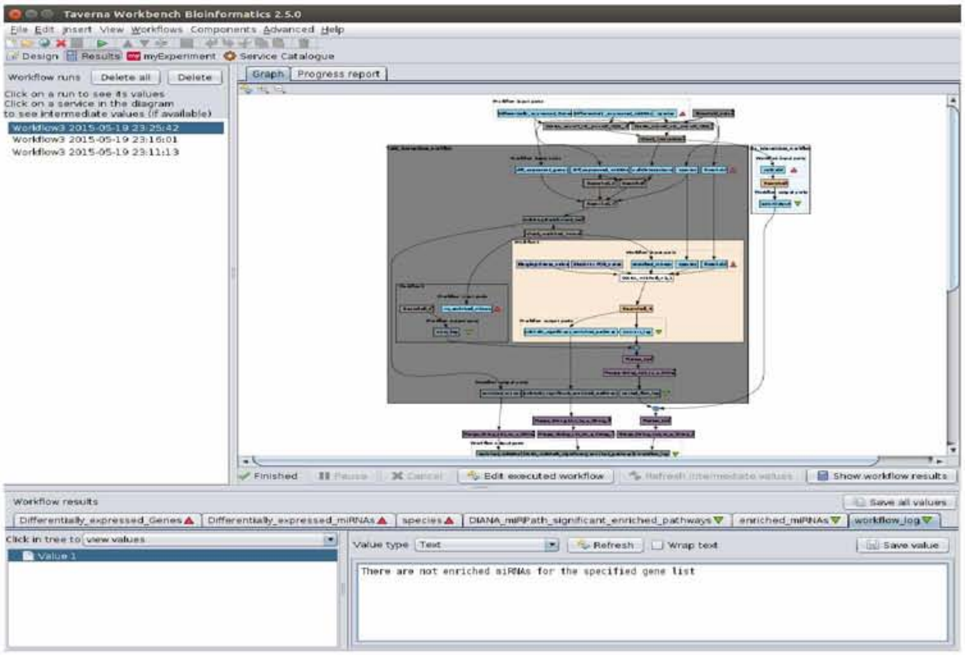
Figure 22: Successful flow execution. Taverna workbench, results perspective. The grey area is the part of the workflow that was executed. The workflow_log output is also depicted.



Taverna Workbench Bioinformatics 2.5.0
File Edit Insert View Workflows Components Advanced Help
Design Results myExperiment Service Catalogue
Workflow runs Delete all Delete
Click on a run to see its values
Click on a service in the diagram
to see intermediate values (if available)
Workflow3 2015-05-19 23:28:03
Workflow3 2015-05-19 23:25:42
Workflow3 2015-05-19 23:16:01
Workflow3 2015-05-19 23:11:13
Graph Progress report
Finished Pause Cancel Edit executed workflow Refresh intermediate values Show workflow results
Workflow results
Save all values
Differentially_expressed_Genes
Differentially_expressed_miRNAs
species
DIANA_miRPath_significant_enriched_pathways
enriched_miRNAs
workflow_log
Click in tree to view values
Value 1
Value type Text
Refresh
Wrap text
Save value

```
ERROR, the workflow has been terminated:
No reported interactions for the queried terms
```

Figure 23: Flow execution with no interactions. Taverna workbench, results perspective. The grey area is the part of the workflow that was executed. The workflow_log output is also depicted.

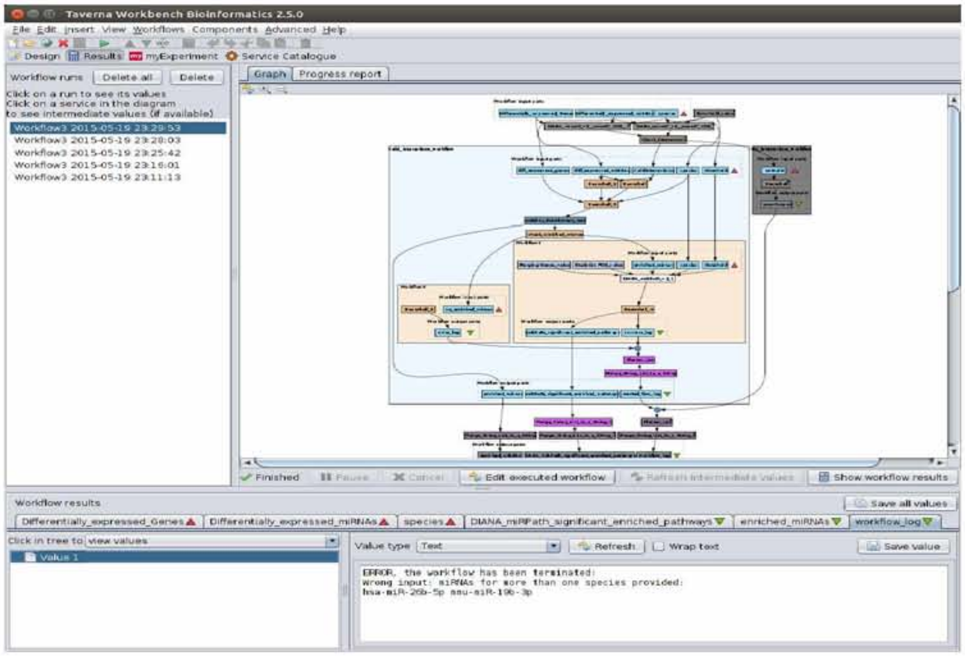

Taverna Workbench Bioinformatics 2.5.0
File Edit Insert View Workflows Components Advanced Help
Design Results myExperiment Service Catalogue
Workflow runs Delete all Delete
Graph Progress report
Finished Pause Cancel Edit executed workflow Refresh intermediate values Show workflow results
Workflow results Save all values
Differentially_expressed_Genes Differentially_expressed_miRNAs species DIANA_miRPath_significant_enriched_pathways enriched_miRNAs workflow_log
Click in tree to view values
Value 1
Value type Text Refresh □ Wrap text Save value

Click on a run to see its values
Click on a service in the diagram
to see intermediate values (if available)

Workflow3 2015-05-19 23:25:42
Workflow3 2015-05-19 23:16:01
Workflow3 2015-05-19 23:11:13

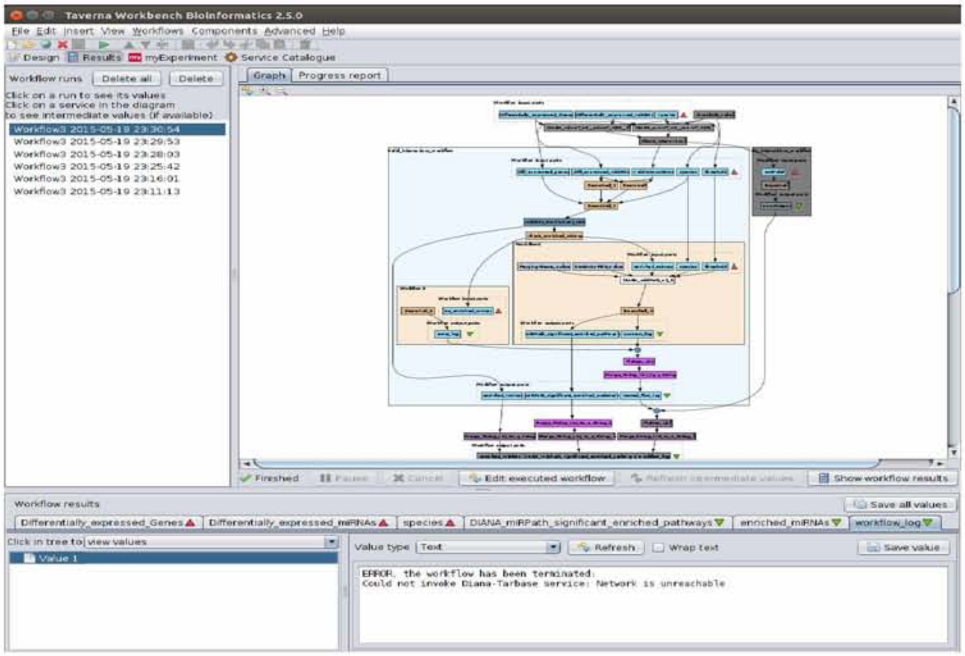
```
There are not enriched miRNAs for the specified gene list
```

Figure 24: Flow execution with no enriched miRNAs. Taverna workbench, results perspective. The grey area is the part of the workflow that was executed. The workflow_log output is also depicted.


Taverna Workbench Bioinformatics 2.5.0
File Edit Insert View Workflows Components Advanced Help
Design
Results
myExperiment
Service Catalogue
Workflow runs
Delete all
Delete
Click on a run to see its values
Click on a service in the diagram
to see intermediate values (if available)
Workflow3 2015-05-19 23:29:53
Workflow3 2015-05-19 23:28:03
Workflow3 2015-05-19 23:25:42
Workflow3 2015-05-19 23:16:01
Workflow3 2015-05-19 23:11:13
Graph
Progress report
Finished
Pause
Cancel
Edit executed workflow
Refresh intermediate values
Show workflow results
Workflow results
Save all values
Differentially_expressed_Genes
Differentially_expressed_miRNAs
species
DIANA_miRPath_significant_enriched_pathways
enriched_miRNAs
workflow_log
Click in tree to
view values
Value 1
Value type
Text
Refresh
Wrap text
Save value

```
ERROR, the workflow has been terminated:
Wrong input: miRNAs for more than one species provided:
hsa-miR-26b-5p mmu-miR-19b-3p
```

Figure 25: Flow execution with miRNAs from different species. Taverna workbench, results perspective. The grey area is the part of the workflow that was executed. The workflow_log output is also depicted.


Taverna Workbench Bioinformatics 2.5.0
File Edit Insert View Workflows Components Advanced Help
Design Results myExperiment Service Catalogue
Workflow runs
Delete all
Delete
Graph
Progress report
Finished
Pause
Cancel
Edit executed workflow
Refresh intermediate values
Show workflow results
Workflow results
Save all values
Differentially_expressed_Genes
Differentially_expressed_miRNAs
species
DIANA_miRPath_significant_enriched_pathways
enriched_miRNAs
workflow_log
Click in tree to
view values
Value 1
Value type
Text
Refresh
Wrap text
Save value

Click on a run to see its values
Click on a service in the diagram
to see intermediate values (if available)

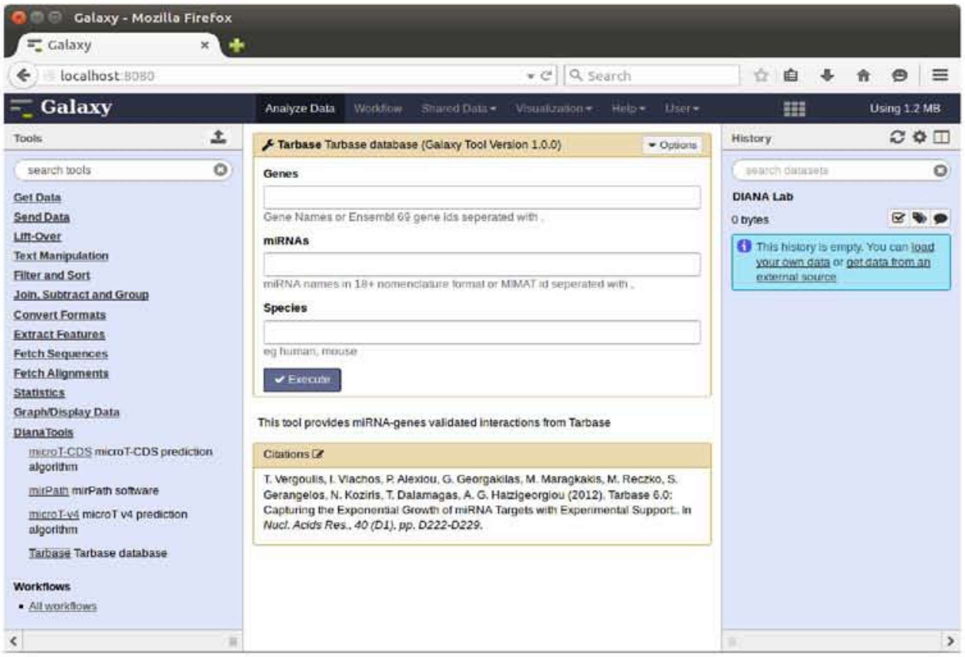
Workflow3 2015-05-19 23:30:54
Workflow3 2015-05-19 23:29:53
Workflow3 2015-05-19 23:28:03
Workflow3 2015-05-19 23:25:42
Workflow3 2015-05-19 23:16:01
Workflow3 2015-05-19 23:11:13

```
ERROR. the workflow has been terminated.
Could not invoke Diana-Tarbase service: Network is unreachable
```

Figure 26: Flow execution with no connectivity. Taverna workbench, results perspective. The grey area is the part of the workflow that was executed. The workflow_log output is also depicted.

A.3 Galaxy integration

A.3.1 DIANA Lab Galaxy tools



Tarbase Tarbase database (Galaxy Tool Version 1.0.0)

Genes

Gene Names or Ensembl 69 gene ids seperated with ,

miRNAs

miRNA names in 18+ nomenclature format or MIMAT id seperated with ,

Species

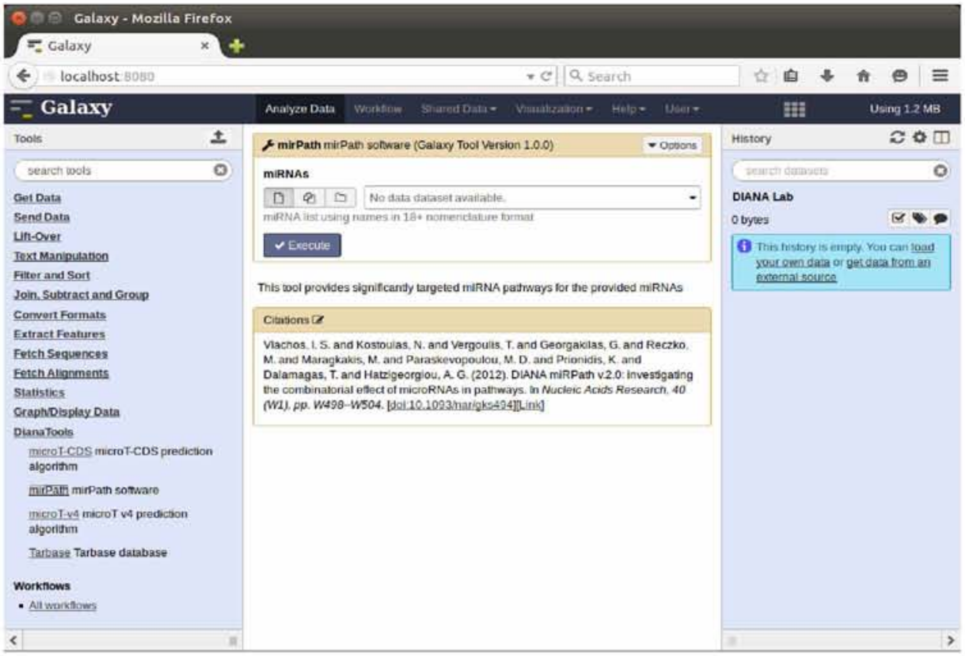
eg human, mouse

This tool provides miRNA-genes validated interactions from Tarbase

T. Vergoulis, I. Vlachos, P. Alexiou, G. Georgakilas, M. Maragkakis, M. Reczko, S. Gerangelos, N. Koziris, T. Dalamagas, A. G. Hatzigeorgiou (2012). Tarbase 6.0: Capturing the Exponential Growth of miRNA Targets with Experimental Support.. In *Nucl. Acids Res., 40 (D1). pp. D222-D229.*

This history is empty. You can load your own data or get data from an external source

Figure 27: DIANA Lab Tarbase Galaxy tool Interface



mirPath mirPath software (Galaxy Tool Version 1.0.0)

miRNAs

No data dataset available.

miRNA list using names in 18+ nomenclature format

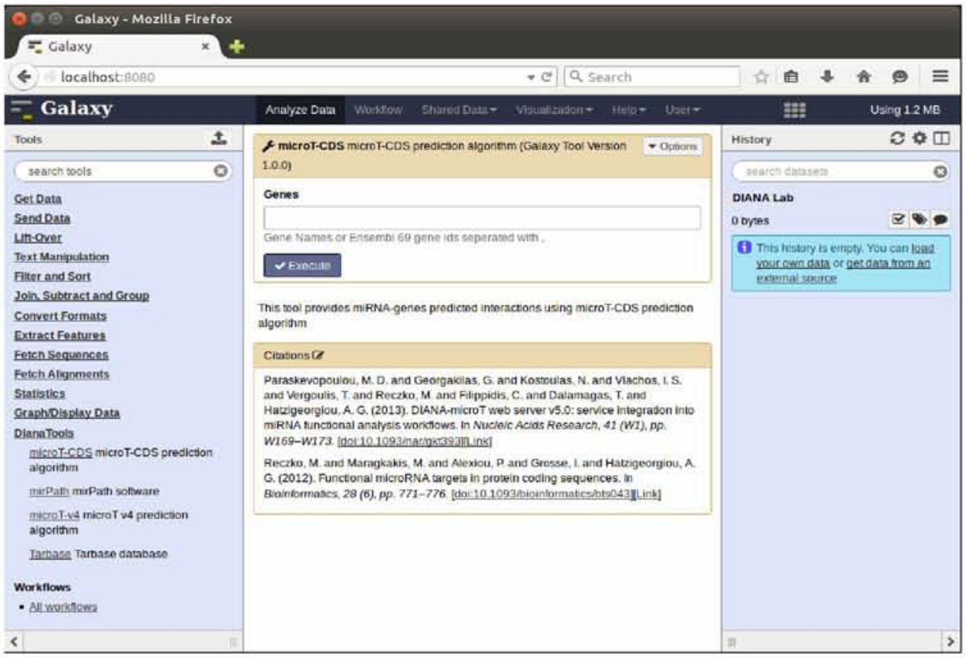
This tool provides significantly targeted miRNA pathways for the provided miRNAs

Citations

Vlachos, I. S. and Kostoulas, N. and Vergoulis, T. and Georgakilas, G. and Reczko, M. and Maragkakis, M. and Paraskevopoulou, M. D. and Prionidis, K. and Dalamagas, T. and Hatzigeorgiou, A. G. (2012). DIANA miRPath v.2.0: investigating the combinatorial effect of microRNAs in pathways. In *Nucleic Acids Research, 40 (W1), pp. W498–W504.* [doi:10.1093/nar/gks494][Link]

This history is empty. You can load your own data or get data from an external source

Figure 28: DIANA Lab miRPath Galaxy tool Interface



microT-CDS microT-CDS prediction algorithm (Galaxy Tool Version 1.0.0)

Genes

Gene Names or Ensembl 69 gene ids seperated with ,

This tool provides miRNA-genes predicted interactions using microT-CDS prediction algorithm

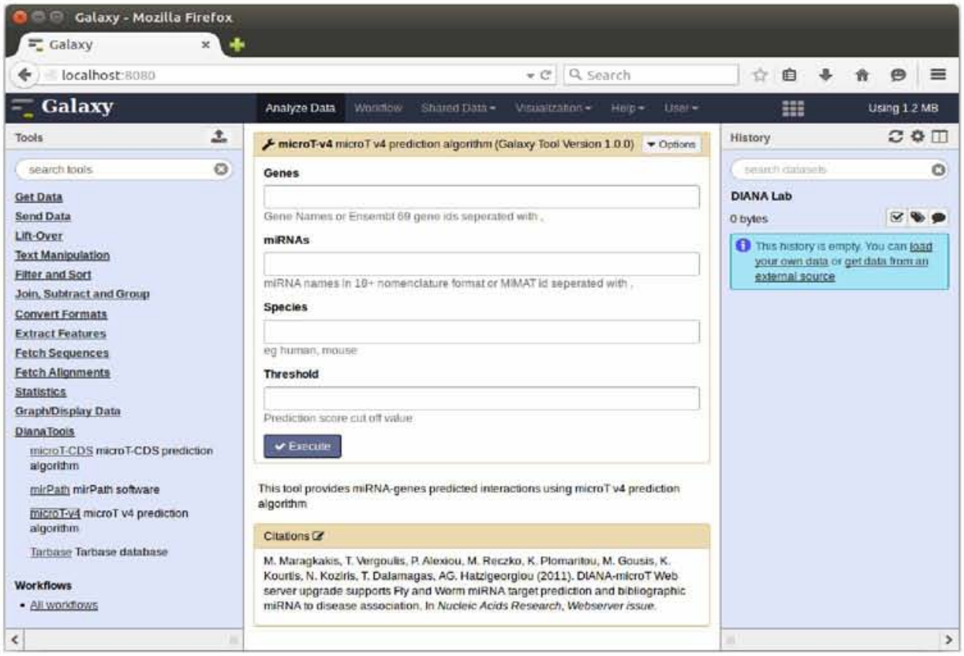
Citations

Paraskevopoulou, M. D. and Georgakilas, G. and Kostoulas, N. and Vlachos, I. S. and Vergoulis, T. and Reczko, M. and Filippidis, C. and Dalamagas, T. and Hatzigeorgiou, A. G. (2013). DIANA-microT web server v5.0: service integration into miRNA functional analysis workflows. In *Nucleic Acids Research, 41 (W1), pp. W169–W173.* [doi:10.1093/nar/gkt393][Link]

Reczko, M. and Maragkakis, M. and Alexiou, P. and Grosse, I. and Hatzigeorgiou, A. G. (2012). Functional microRNA targets in protein coding sequences. In *Bioinformatics, 28 (6), pp. 771–776.* [doi:10.1093/bioinformatics/bts043][Link]

This history is empty. You can load your own data or get data from an external source

Figure 29: DIANA Lab microT-CDS Galaxy tool Interface



microT-CDS microT-CDS prediction algorithm

mirPath mirPath software

microT-v4 microT v4 prediction algorithm

Tarbase Tarbase database

microT-v4 microT v4 prediction algorithm (Galaxy Tool Version 1.0.0)

Genes

Gene Names or Ensembl 69 gene ids seperated with ,

miRNAs

miRNA names in 18+ nomenclature format or MIMAT id seperated with ,

Species

eg human, mouse

Threshold

Prediction score cut off value

This tool provides miRNA-genes predicted interactions using microT v4 prediction algorithm

M. Maragkakis, T. Vergoulis, P. Alexiou, M. Reczko, K. Plomaritou, M. Gousis, K. Kourtis, N. Koziris, T. Dalamagas, AG. Hatzigeorgiou (2011). DIANA-microT Web server upgrade supports Fly and Worm miRNA target prediction and bibliographic miRNA to disease association. In *Nucleic Acids Research, Webserver issue.*

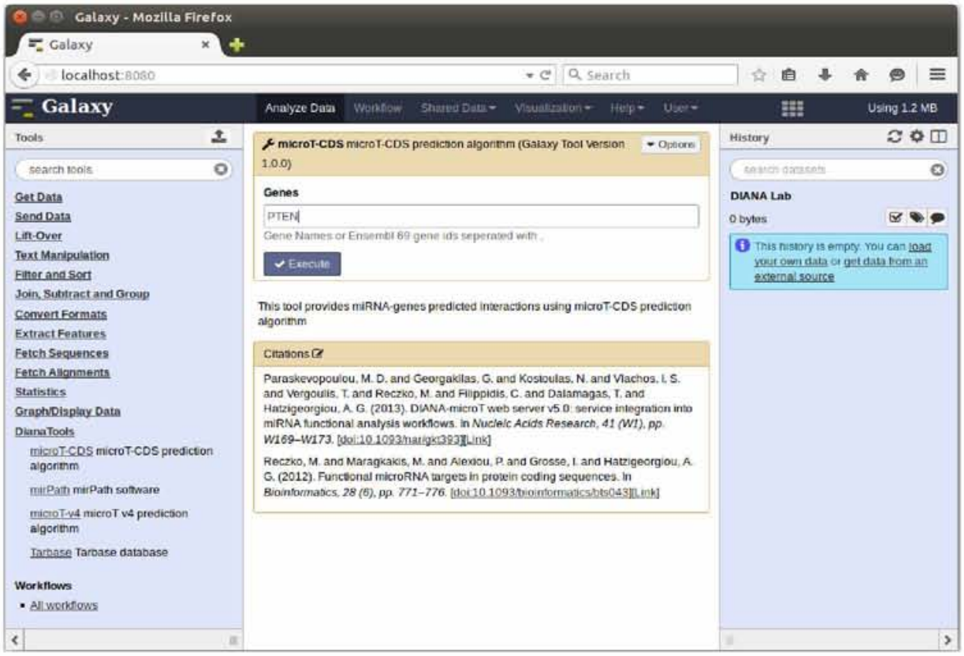
DIANA Lab

0 bytes

This history is empty. You can load your own data or get data from an external source

Figure 30: DIANA Lab microT-v4 Galaxy tool Interface

A.3.2 Scenario steps



microT-CDS microT-CDS prediction algorithm (Galaxy Tool Version 1.0.0)

Genes

PTEN

Gene Names or Ensembl 69 gene ids seperated with ,

This tool provides miRNA-genes predicted interactions using microT-CDS prediction algorithm

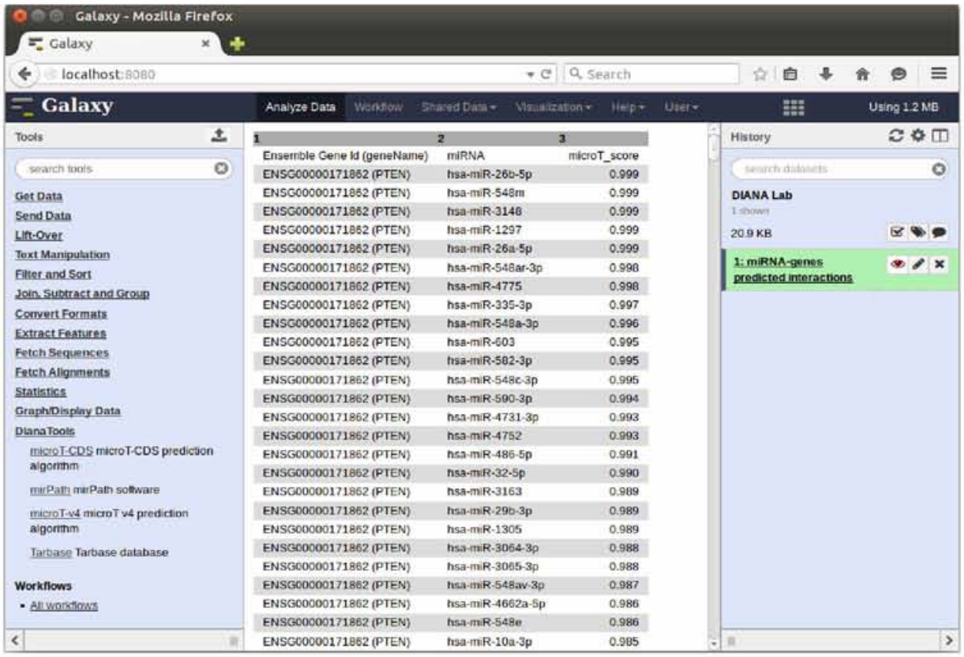
Citations

Paraskevopoulou, M. D. and Georgakilas, G. and Kostoulas, N. and Vlachos, I. S. and Vergoulis, T. and Reczko, M. and Filippidis, C. and Dalamagas, T. and Hatzigeorgiou, A. G. (2013). DIANA-microT web server v5.0: service integration into miRNA functional analysis workflows. In *Nucleic Acids Research, 41 (W1), pp. W169–W173.* [doi:10.1093/nar/gkt393][Link]

Reczko, M. and Maragkakis, M. and Alexiou, P. and Grosse, I. and Hatzigeorgiou, A. G. (2012). Functional microRNA targets in protein coding sequences. In *Bioinformatics, 28 (6), pp. 771–776.* [doi:10.1093/bioinformatics/bts043][Link]

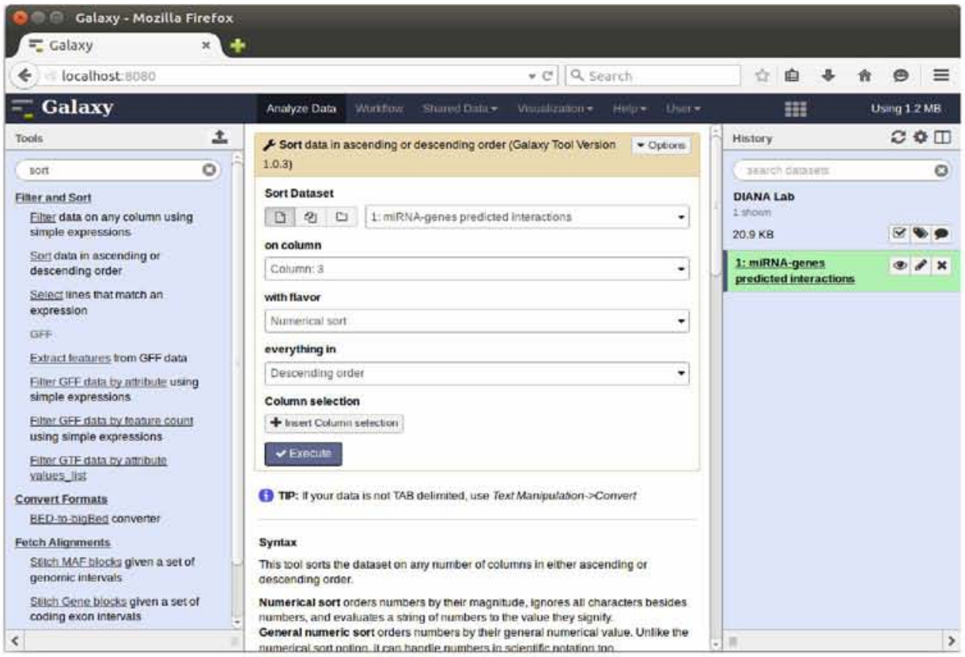
This history is empty. You can load your own data or get data from an external source

Figure 31: DIANA Lab microT-CDS invocation using PTEN gene name as input value



1	2	3
Ensemble Gene Id (geneName)	miRNA	microT_score
ENSG00000171862 (PTEN)	hsa-miR-26b-5p	0.999
ENSG00000171862 (PTEN)	hsa-miR-548m	0.999
ENSG00000171862 (PTEN)	hsa-miR-3148	0.999
ENSG00000171862 (PTEN)	hsa-miR-1297	0.999
ENSG00000171862 (PTEN)	hsa-miR-26a-5p	0.999
ENSG00000171862 (PTEN)	hsa-miR-548ar-3p	0.998
ENSG00000171862 (PTEN)	hsa-miR-4775	0.998
ENSG00000171862 (PTEN)	hsa-miR-335-3p	0.997
ENSG00000171862 (PTEN)	hsa-miR-548a-3p	0.996
ENSG00000171862 (PTEN)	hsa-miR-603	0.995
ENSG00000171862 (PTEN)	hsa-miR-582-3p	0.995
ENSG00000171862 (PTEN)	hsa-miR-548c-3p	0.995
ENSG00000171862 (PTEN)	hsa-miR-590-3p	0.994
ENSG00000171862 (PTEN)	hsa-miR-4731-3p	0.993
ENSG00000171862 (PTEN)	hsa-miR-4752	0.993
ENSG00000171862 (PTEN)	hsa-miR-486-5p	0.991
ENSG00000171862 (PTEN)	hsa-miR-32-5p	0.990
ENSG00000171862 (PTEN)	hsa-miR-3163	0.989
ENSG00000171862 (PTEN)	hsa-miR-29b-3p	0.989
ENSG00000171862 (PTEN)	hsa-miR-1305	0.989
ENSG00000171862 (PTEN)	hsa-miR-3064-3p	0.988
ENSG00000171862 (PTEN)	hsa-miR-3065-3p	0.988
ENSG00000171862 (PTEN)	hsa-miR-548av-3p	0.987
ENSG00000171862 (PTEN)	hsa-miR-4662a-5p	0.986
ENSG00000171862 (PTEN)	hsa-miR-548e	0.986
ENSG00000171862 (PTEN)	hsa-miR-10a-3p	0.985

Figure 32: DIANA Lab microT-CDS results for gene PTEN. miRNAs targeting this gene, the gene name and id, along with the prediction score are provided


Galaxy - Mozilla Firefox
Galaxy
localhost:8080
Search
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 1.2 MB
Tools
sort
Filter and Sort
Filter data on any column using simple expressions
Sort data in ascending or descending order
Select lines that match an expression
GFF
Extract features from GFF data
Filter GFF data by attribute using simple expressions
Filter GFF data by feature count using simple expressions
Filter GTF data by attribute values_list
Convert Formats
BED-to-bigBed converter
Fetch Alignments
Stitch MAF blocks given a set of genomic intervals
Stitch Gene blocks given a set of coding exon intervals
History
search datasets
DIANA Lab
1 shown
20.9 KB
1: miRNA-genes predicted interactions

Sort data in ascending or descending order (Galaxy Tool Version 1.0.3)

Options

Sort Dataset

1: miRNA-genes predicted interactions

on column

Column: 3

with flavor

Numerical sort

everything in

Descending order

Column selection

Insert Column selection

Execute

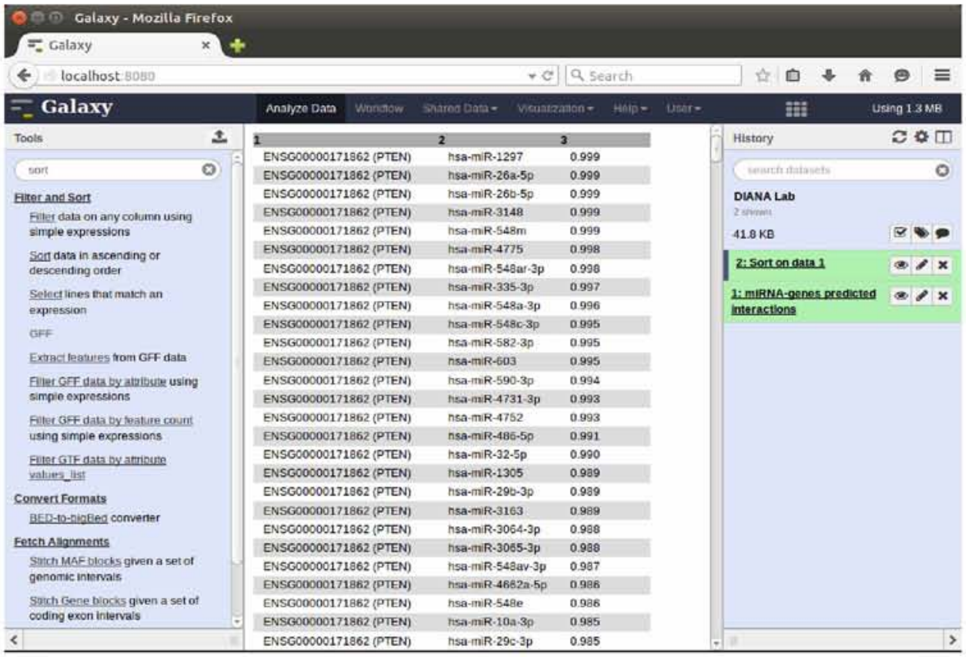
TIP: If your data is not TAB delimited, use *Text Manipulation->Convert*

Syntax

This tool sorts the dataset on any number of columns in either ascending or descending order.

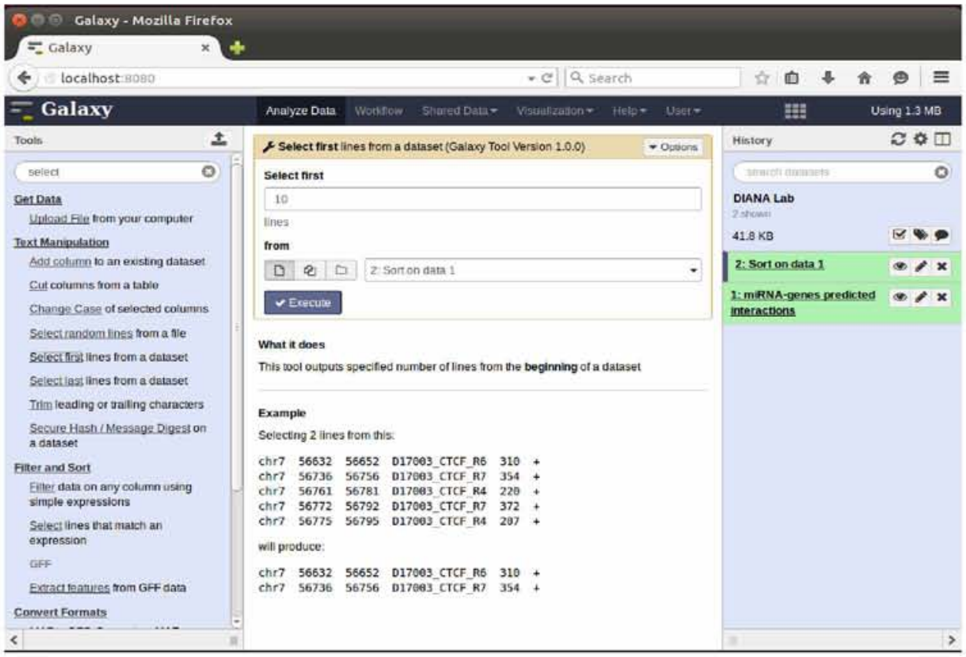
Numerical sort orders numbers by their magnitude, ignores all characters besides numbers, and evaluates a string of numbers to the value they signify.
General numeric sort orders numbers by their general numerical value. Unlike the numerical sort option, it can handle numbers in scientific notation too.

Figure 33: Galaxy Sort tool invoked using the results of microT-CDS tool. Sort is perfomed on column 3, the prediction score.


Galaxy - Mozilla Firefox
Galaxy
localhost:8080
Search
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 1.3 MB
Tools
sort
Filter and Sort
Filter data on any column using simple expressions
Sort data in ascending or descending order
Select lines that match an expression
GFF
Extract features from GFF data
Filter GFF data by attribute using simple expressions
Filter GFF data by feature count using simple expressions
Filter GTF data by attribute values_list
Convert Formats
BED-to-bigBed converter
Fetch Alignments
Stitch MAF blocks given a set of genomic intervals
Stitch Gene blocks given a set of coding exon intervals
History
search datasets
DIANA Lab
2 shown
41.8 KB
2: Sort on data 1
1: miRNA-genes predicted interactions

1	2	3
ENSG00000171862 (PTEN)	hsa-miR-1297	0.999
ENSG00000171862 (PTEN)	hsa-miR-26a-5p	0.999
ENSG00000171862 (PTEN)	hsa-miR-26b-5p	0.999
ENSG00000171862 (PTEN)	hsa-miR-3148	0.999
ENSG00000171862 (PTEN)	hsa-miR-548m	0.999
ENSG00000171862 (PTEN)	hsa-miR-4775	0.998
ENSG00000171862 (PTEN)	hsa-miR-548ar-3p	0.998
ENSG00000171862 (PTEN)	hsa-miR-335-3p	0.997
ENSG00000171862 (PTEN)	hsa-miR-548a-3p	0.996
ENSG00000171862 (PTEN)	hsa-miR-548c-3p	0.995
ENSG00000171862 (PTEN)	hsa-miR-582-3p	0.995
ENSG00000171862 (PTEN)	hsa-miR-603	0.995
ENSG00000171862 (PTEN)	hsa-miR-590-3p	0.994
ENSG00000171862 (PTEN)	hsa-miR-4731-3p	0.993
ENSG00000171862 (PTEN)	hsa-miR-4752	0.993
ENSG00000171862 (PTEN)	hsa-miR-486-5p	0.991
ENSG00000171862 (PTEN)	hsa-miR-32-5p	0.990
ENSG00000171862 (PTEN)	hsa-miR-1305	0.989
ENSG00000171862 (PTEN)	hsa-miR-29b-3p	0.989
ENSG00000171862 (PTEN)	hsa-miR-3163	0.989
ENSG00000171862 (PTEN)	hsa-miR-3064-3p	0.988
ENSG00000171862 (PTEN)	hsa-miR-3065-3p	0.988
ENSG00000171862 (PTEN)	hsa-miR-548av-3p	0.987
ENSG00000171862 (PTEN)	hsa-miR-4662a-5p	0.986
ENSG00000171862 (PTEN)	hsa-miR-548e	0.986
ENSG00000171862 (PTEN)	hsa-miR-10a-3p	0.985
ENSG00000171862 (PTEN)	hsa-miR-29c-3p	0.985

Figure 34: Galaxy Sort tool output, containing microT-CDS results sorted on prediction score


Galaxy - Mozilla Firefox
Galaxy
localhost:8080
Search
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 1.3 MB
Tools
select
Get Data
Upload File from your computer
Text Manipulation
Add column to an existing dataset
Cut columns from a table
Change Case of selected columns
Select random lines from a file
Select first lines from a dataset
Select last lines from a dataset
Trim leading or trailing characters
Secure Hash / Message Digest on a dataset
Filter and Sort
Filter data on any column using simple expressions
Select lines that match an expression
GFF
Extract features from GFF data
Convert Formats
Options
Execute
History
search datasets
DIANA Lab
2 shown
41.8 KB
2: Sort on data 1
1: miRNA-genes predicted interactions

Select first lines from a dataset (Galaxy Tool Version 1.0.0)

Select first

10

lines

from

2: Sort on data 1

What it does

This tool outputs specified number of lines from the **beginning** of a dataset

Example

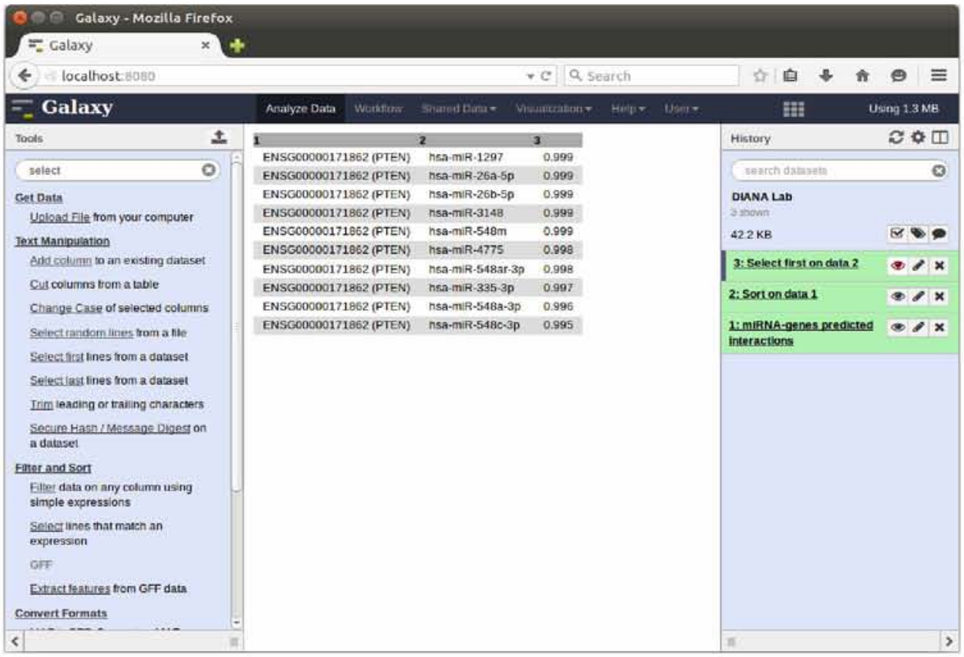
Selecting 2 lines from this:

```
chr7  56632  56652  D17003_CTCF_R6  310  +
chr7  56736  56756  D17003_CTCF_R7  354  +
chr7  56761  56781  D17003_CTCF_R4  220  +
chr7  56772  56792  D17003_CTCF_R7  372  +
chr7  56775  56795  D17003_CTCF_R4  207  +
```

will produce:

```
chr7  56632  56652  D17003_CTCF_R6  310  +
chr7  56736  56756  D17003_CTCF_R7  354  +
```

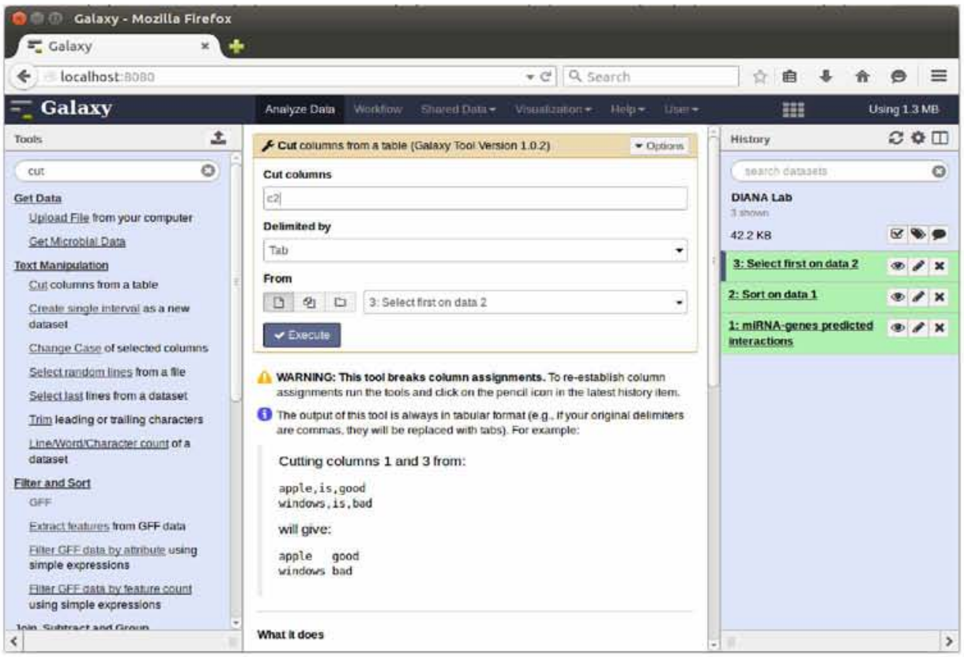
Figure 35: Galaxy Select first tool invoked using the results of the Sort tool. It is configured to select the first ten entries.



Galaxy - Mozilla Firefox
Galaxy
localhost:8080
Search
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 1.3 MB
Tools
select
Get Data
Upload File from your computer
Text Manipulation
Add column to an existing dataset
Cut columns from a table
Change Case of selected columns
Select random lines from a file
Select first lines from a dataset
Select last lines from a dataset
Trim leading or trailing characters
Secure Hash / Message Digest on a dataset
Filter and Sort
Filter data on any column using simple expressions
Select lines that match an expression
GFF
Extract features from GFF data
Convert Formats
History
search datasets
DIANA Lab
3 shown
42.2 KB
3: Select first on data 2
2: Sort on data 1
1: miRNA-genes predicted interactions

1	2	3
ENSG00000171862 (PTEN)	hsa-miR-1297	0.999
ENSG00000171862 (PTEN)	hsa-miR-26a-5p	0.999
ENSG00000171862 (PTEN)	hsa-miR-26b-5p	0.999
ENSG00000171862 (PTEN)	hsa-miR-3148	0.999
ENSG00000171862 (PTEN)	hsa-miR-548m	0.999
ENSG00000171862 (PTEN)	hsa-miR-4775	0.998
ENSG00000171862 (PTEN)	hsa-miR-548ar-3p	0.998
ENSG00000171862 (PTEN)	hsa-miR-335-3p	0.997
ENSG00000171862 (PTEN)	hsa-miR-548a-3p	0.996
ENSG00000171862 (PTEN)	hsa-miR-548c-3p	0.995

Figure 36: Select first tool results providing the first ten miRNA-gene interactions along with their prediction score and the gene name and id


Galaxy - Mozilla Firefox
Galaxy
localhost:8080
Search
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 1.3 MB
Tools
cut
Get Data
Upload File from your computer
Get Microbial Data
Text Manipulation
Cut columns from a table
Create single interval as a new dataset
Change Case of selected columns
Select random lines from a file
Select last lines from a dataset
Trim leading or trailing characters
Line/Word/Character count of a dataset
Filter and Sort
GFF
Extract features from GFF data
Filter GFF data by attribute using simple expressions
Filter GFF data by feature count using simple expressions
Join, Subtract and Group
History
search datasets
DIANA Lab
3 shown
42.2 KB
3: Select first on data 2
2: Sort on data 1
1: miRNA-genes predicted interactions
Options
Execute

Cut columns from a table (Galaxy Tool Version 1.0.2)

Cut columns

c2

Delimited by

Tab

From

3: Select first on data 2

WARNING: This tool breaks column assignments. To re-establish column assignments run the tools and click on the pencil icon in the latest history item.

The output of this tool is always in tabular format (e.g., if your original delimiters are commas, they will be replaced with tabs). For example:

Cutting columns 1 and 3 from:

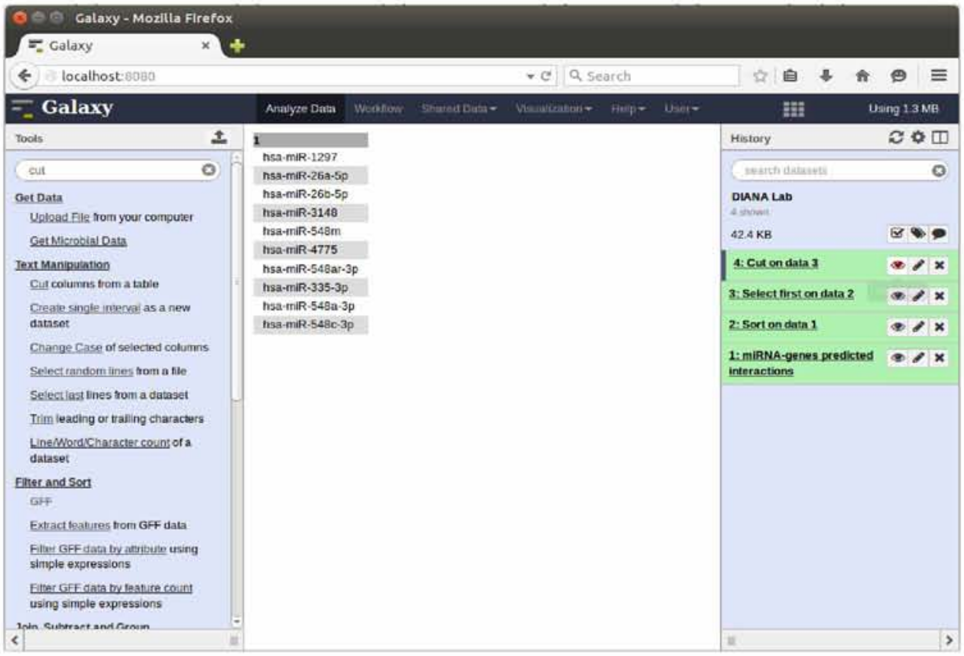
```
apple,is,good
windows,is,bad
```

will give:

```
apple   good
windows bad
```

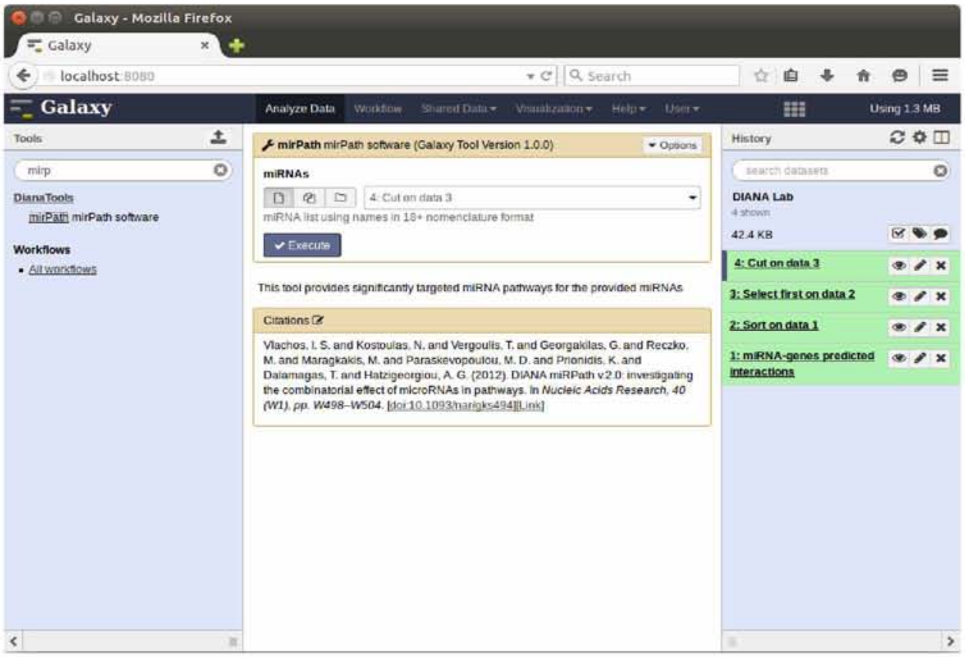
What it does

Figure 37: Galaxy Cut columns tool invoked using the results of the Select tool. It is configured isolate the second column which contains the miRNAs.


Galaxy - Mozilla Firefox
Galaxy
localhost:8080
Search
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 1.3 MB
Tools
cut
Get Data
Upload File from your computer
Get Microbial Data
Text Manipulation
Cut columns from a table
Create single interval as a new dataset
Change Case of selected columns
Select random lines from a file
Select last lines from a dataset
Trim leading or trailing characters
Line/Word/Character count of a dataset
Filter and Sort
GFF
Extract features from GFF data
Filter GFF data by attribute using simple expressions
Filter GFF data by feature count using simple expressions
Join, Subtract and Group
History
search datasets
DIANA Lab
4 shown
42.4 KB
4: Cut on data 3
3: Select first on data 2
2: Sort on data 1
1: miRNA-genes predicted interactions

```
1
hsa-miR-1297
hsa-miR-26a-5p
hsa-miR-26b-5p
hsa-miR-3148
hsa-miR-548m
hsa-miR-4775
hsa-miR-548ar-3p
hsa-miR-335-3p
hsa-miR-548a-3p
hsa-miR-548c-3p
```

Figure 38: Galaxy Cut columns tool results containing only the miRNA list.



mirPath mirPath software

mirPath mirPath software (Galaxy Tool Version 1.0.0)

miRNAs: 4: Cut on data 3

miRNA list using names in 18+ nomenclature format

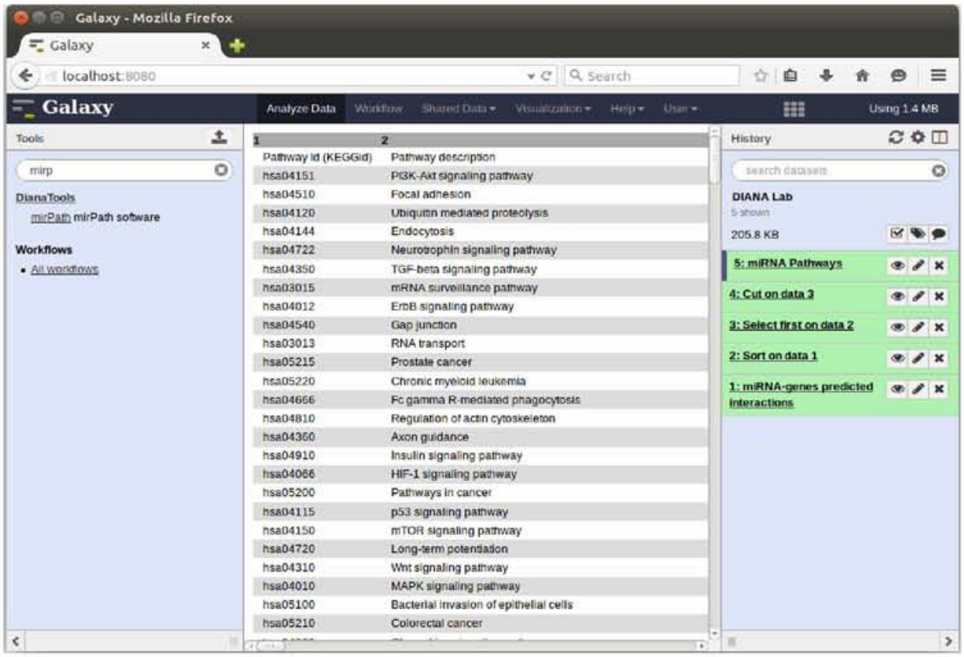
This tool provides significantly targeted miRNA pathways for the provided miRNAs

Vlachos, I. S. and Kostoulas, N. and Vergoulis, T. and Georgakilas, G. and Reczko, M. and Maragkakis, M. and Paraskevopoulou, M. D. and Prionidis, K. and Dalamagas, T. and Hatzigeorgiou, A. G. (2012). DIANA miRPath v2.0: investigating the combinatorial effect of microRNAs in pathways. In *Nucleic Acids Research, 40 (W1), pp. W498–W504*. [doi:10.1093/nar/gks494][Link]

DIANA Lab
4 shown
42.4 KB

4: Cut on data 3
3: Select first on data 2
2: Sort on data 1
1: miRNA-genes predicted interactions

Figure 39: Galaxy miRPath tool invocation, using the miRNA list output of the Cut columns tool



1	2
Pathway id (KEGGid)	Pathway description
hsa04151	PI3K-Akt signaling pathway
hsa04510	Focal adhesion
hsa04120	Ubiquitin mediated proteolysis
hsa04144	Endocytosis
hsa04722	Neurotrophin signaling pathway
hsa04350	TGF-beta signaling pathway
hsa03015	mRNA surveillance pathway
hsa04012	ErbB signaling pathway
hsa04540	Gap junction
hsa03013	RNA transport
hsa05215	Prostate cancer
hsa05220	Chronic myeloid leukemia
hsa04666	Fc gamma R-mediated phagocytosis
hsa04810	Regulation of actin cytoskeleton
hsa04360	Axon guidance
hsa04910	Insulin signaling pathway
hsa04066	HIF-1 signaling pathway
hsa05200	Pathways in cancer
hsa04115	p53 signaling pathway
hsa04150	mTOR signaling pathway
hsa04720	Long-term potentiation
hsa04310	Wnt signaling pathway
hsa04010	MAPK signaling pathway
hsa05100	Bacterial invasion of epithelial cells
hsa05210	Colorectal cancer

5: miRNA Pathways

4: Cut on data 3

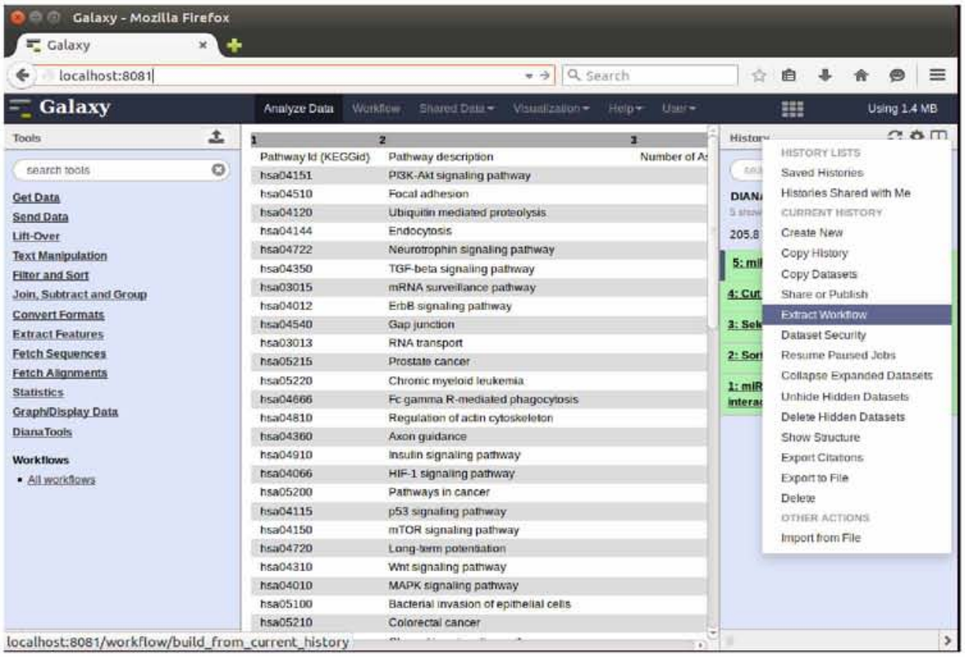
3: Select first on data 2

2: Sort on data 1

1: miRNA-genes predicted interactions

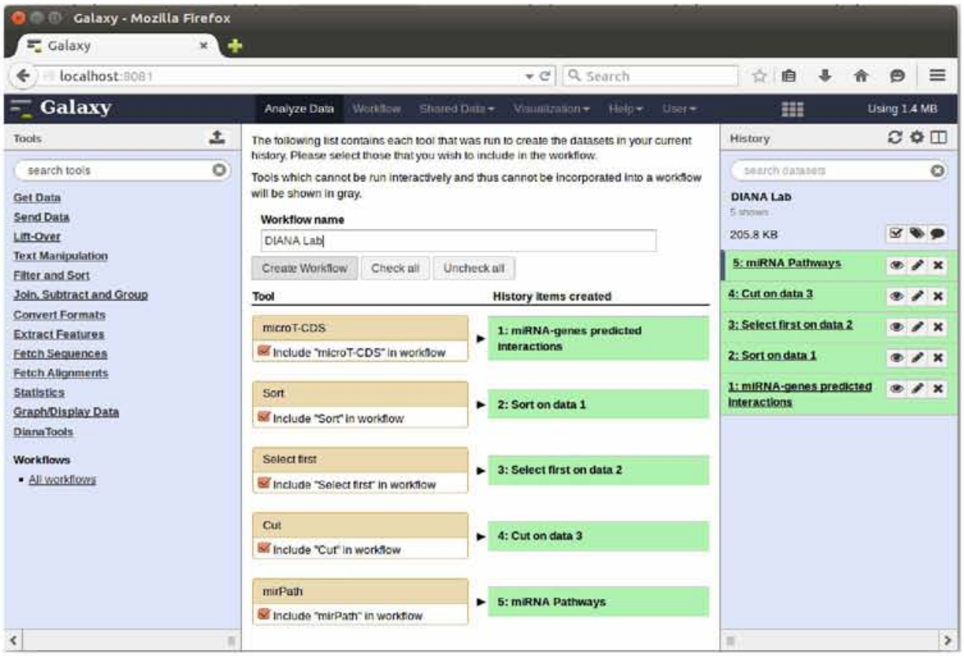
Figure 40: Galaxy miRPath tool results. The two first columns are depicted, containing Pathway KEGG id and Pathway description

A.3.3 Workflow Extraction



1	2	3
Pathway Id (KEGGid)	Pathway description	Number of A
hsa04151	PI3K-Akt signaling pathway	
hsa04510	Focal adhesion	
hsa04120	Ubiquitin mediated proteolysis	
hsa04144	Endocytosis	
hsa04722	Neurotrophin signaling pathway	
hsa04350	TGF-beta signaling pathway	
hsa03015	mRNA surveillance pathway	
hsa04012	ErbB signaling pathway	
hsa04540	Gap junction	
hsa03013	RNA transport	
hsa05215	Prostate cancer	
hsa05220	Chronic myeloid leukemia	
hsa04666	Fc gamma R-mediated phagocytosis	
hsa04810	Regulation of actin cytoskeleton	
hsa04360	Axon guidance	
hsa04910	Insulin signaling pathway	
hsa04066	HIF-1 signaling pathway	
hsa05200	Pathways in cancer	
hsa04115	p53 signaling pathway	
hsa04150	mTOR signaling pathway	
hsa04720	Long-term potentiation	
hsa04310	Wnt signaling pathway	
hsa04010	MAPK signaling pathway	
hsa05100	Bacterial invasion of epithelial cells	
hsa05210	Colorectal cancer	

Figure 41: Create a Workflow for a given History. The option Extract Workflow in the History options menu is used.



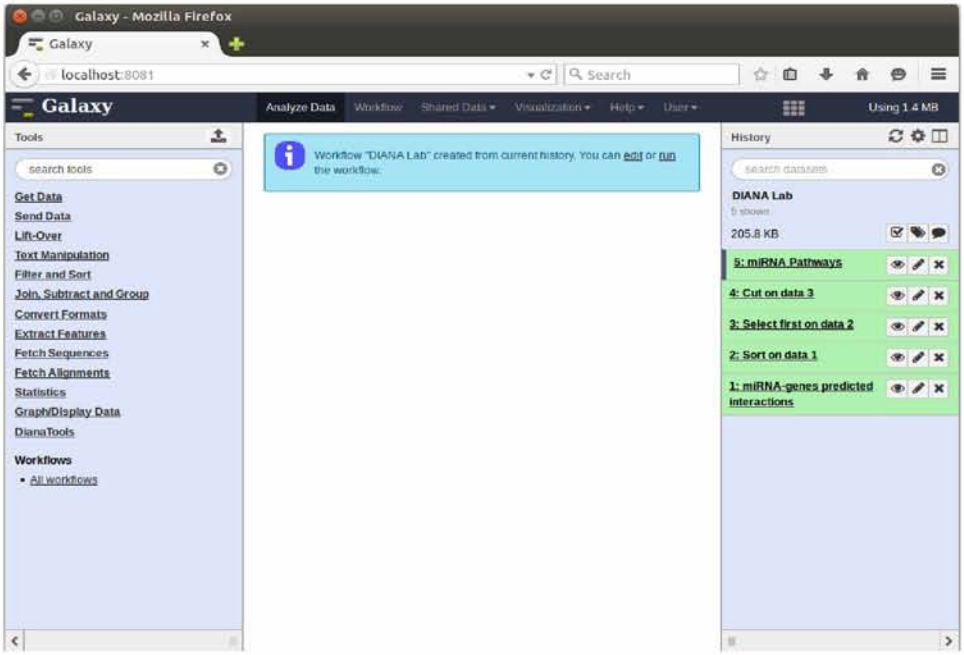
The following list contains each tool that was run to create the datasets in your current history. Please select those that you wish to include in the workflow.

Tools which cannot be run interactively and thus cannot be incorporated into a workflow will be shown in gray.

Workflow name: DIANA Lab

Tool	History items created
microT-CDS ☑ Include "microT-CDS" in workflow	1: miRNA-genes predicted interactions
Sort ☑ Include "Sort" in workflow	2: Sort on data 1
Select first ☑ Include "Select first" in workflow	3: Select first on data 2
Cut ☑ Include "Cut" in workflow	4: Cut on data 3
mirPath ☑ Include "mirPath" in workflow	5: miRNA Pathways

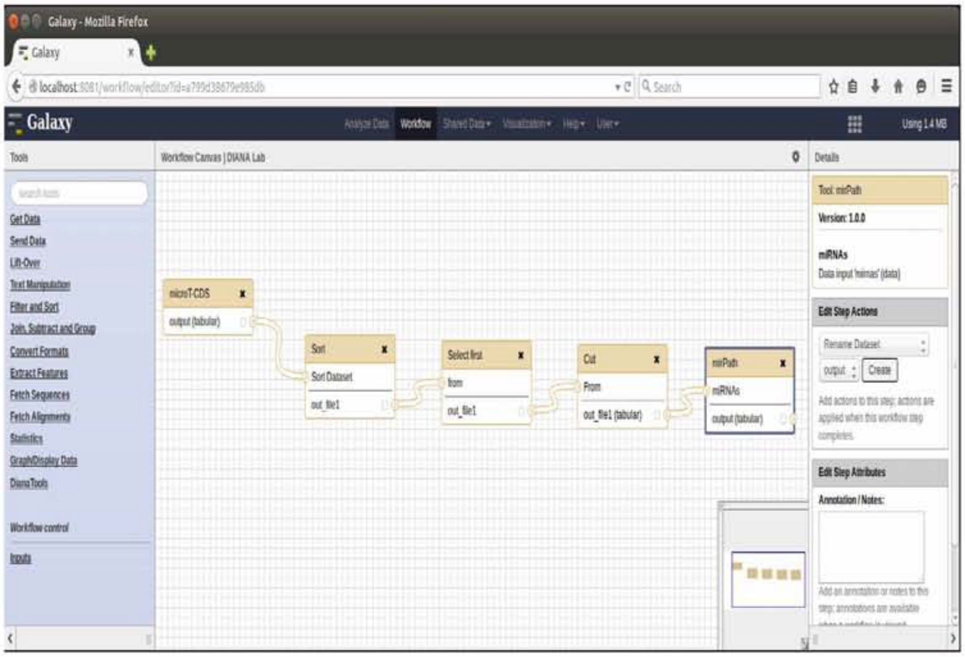
Figure 42: Dialog window where the user specifies the name of the worflow along with the steps from the history that it should contain



Workflow "DIANA Lab" created from current history. You can edit or run the workflow.

5: miRNA Pathways
4: Cut on data 3
3: Select first on data 2
2: Sort on data 1
1: miRNA-genes predicted interactions

Figure 43: Window informing that the workflow is created



localhost:8081/workflow/editor?id=a799d38679e985db

Add actions to this step; actions are applied when this workflow step completes.

Add an annotation or notes to this step; annotations are available

Figure 44: Galaxy workflow canvas. In this window new Workflows can be created, or already existing flows can be modified. In this case the Workflow containing the previously executed steps is depicted

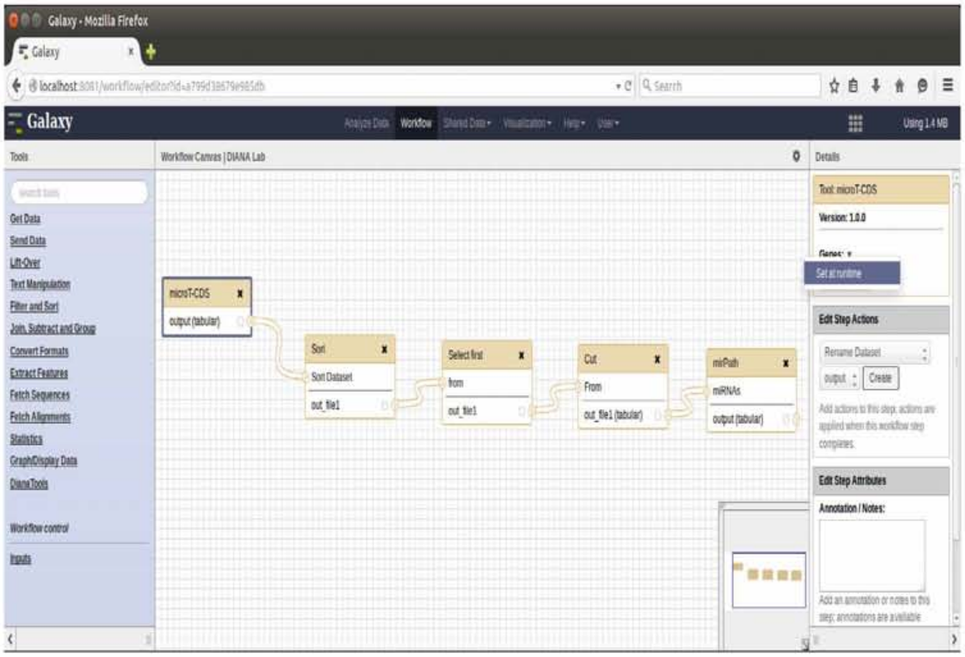
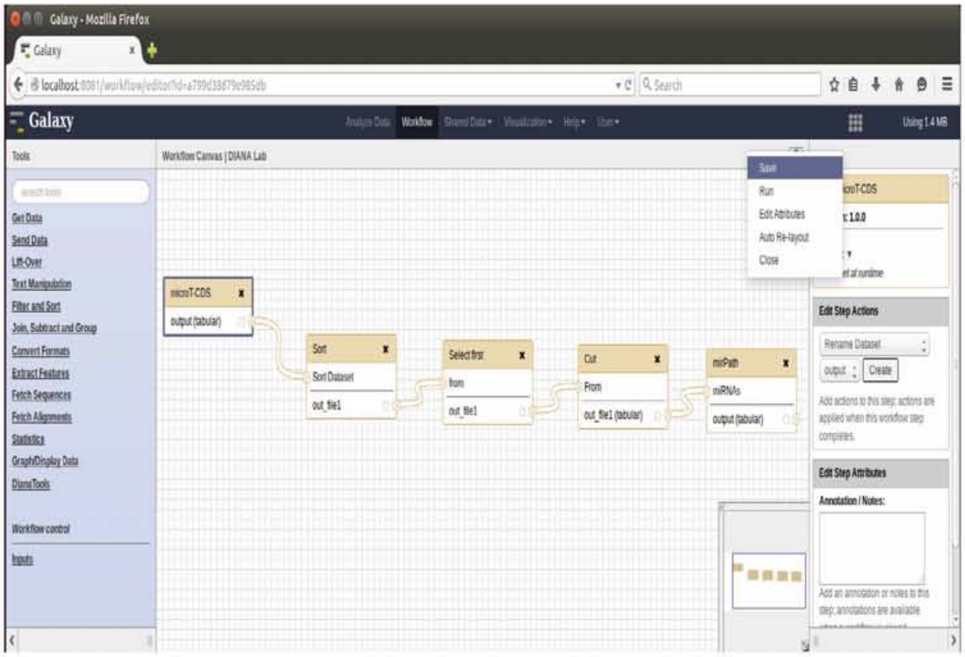


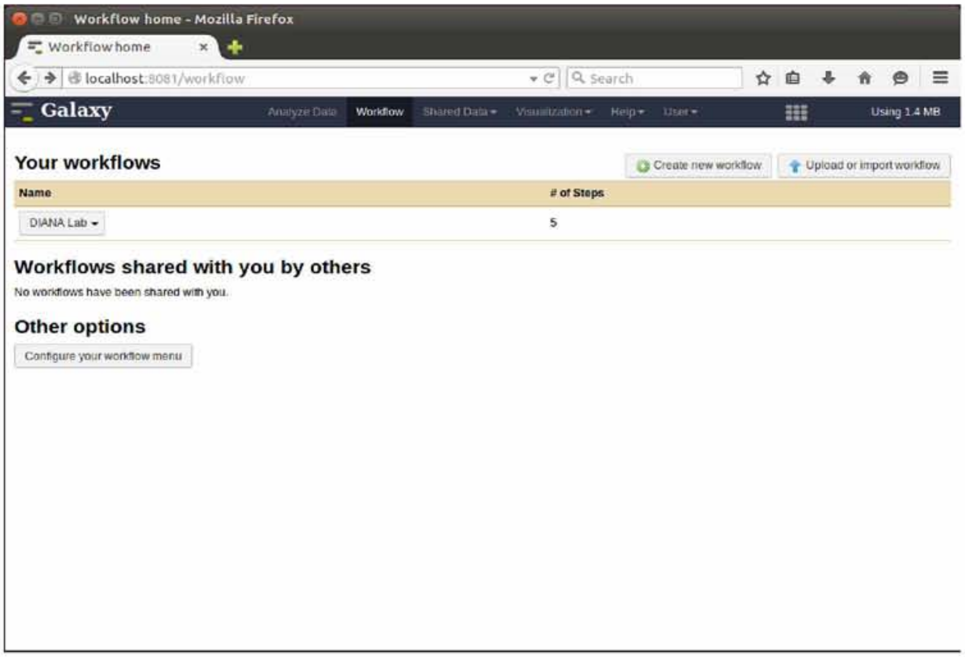
Figure 45: Configuring the workflow to ask for the input value at runtime



Add actions to this step; actions are applied when this workflow step completes.

Add an annotation or notes to this step; annotations are available

Figure 46: The modifications made in the Galaxy workflow canvas must be saved to have effect



Your workflows

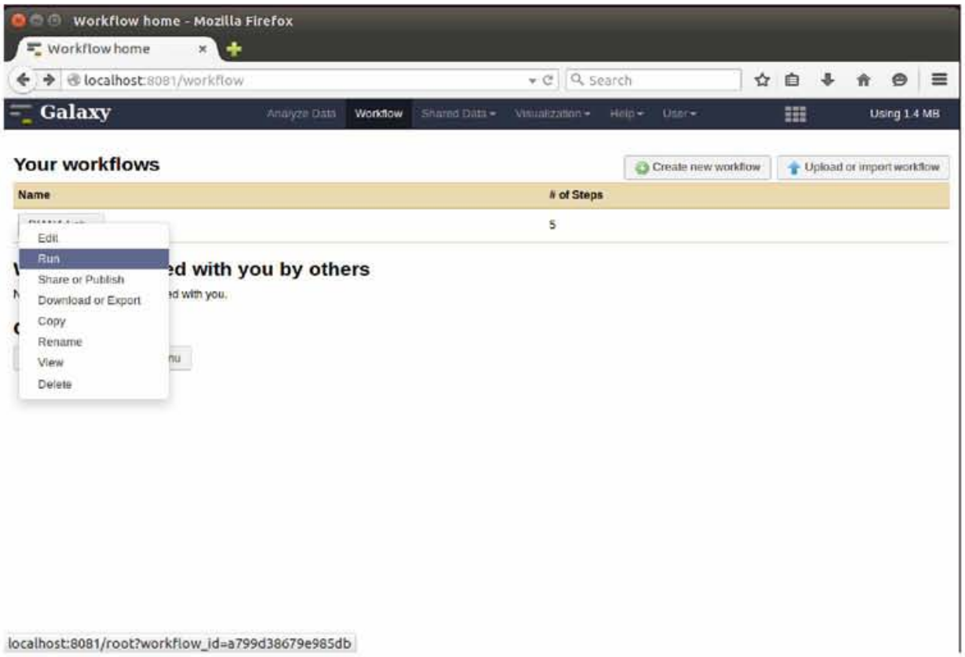
Name	# of Steps
DIANA Lab	5

Workflows shared with you by others

No workflows have been shared with you.

Other options

Figure 47: Galaxy Workflows page. Provides an overview of the workflows, ways to use them, and the ability to create or import a new workflow.



Your workflows

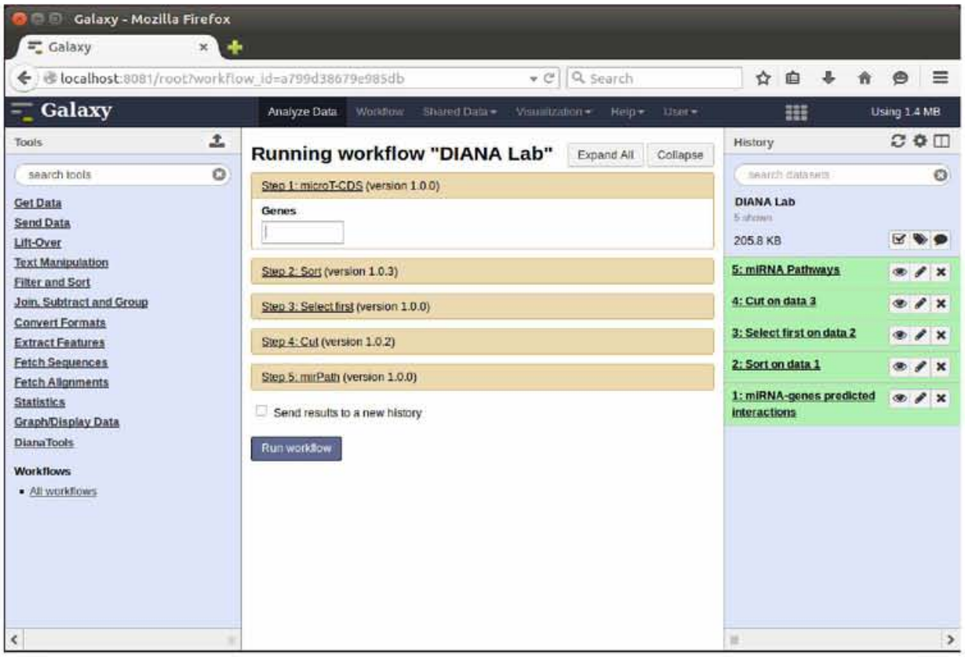
Name	# of Steps
	5

ed with you by others

ed with you.

localhost:8081/root?workflow_id=a799d38679e985db

Figure 48: Selecting to run the DIANA Lab Workflow from Galaxy Workflows page



Running workflow "DIANA Lab"

Step 1: microT-CDS (version 1.0.0)

Genes

Step 2: Sort (version 1.0.3)

Step 3: Select first (version 1.0.0)

Step 4: Cut (version 1.0.2)

Step 5: mirPath (version 1.0.0)

□ Send results to a new history

DIANA Lab

5 shown

205.8 KB

5: miRNA Pathways

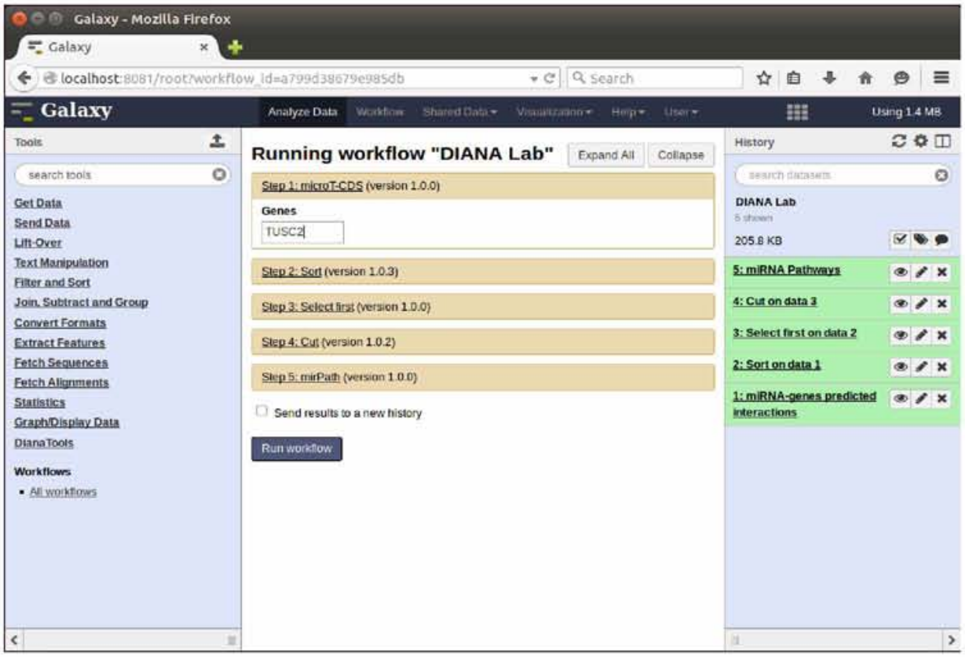
4: Cut on data 3

3: Select first on data 2

2: Sort on data 1

1: miRNA-genes predicted interactions

Figure 49: DIANA Lab Workflow execution page. It describes the steps consisting the Workflow and prompts for any user provided arguments



Running workflow "DIANA Lab"

Step 1: microT-CDS (version 1.0.0)

Genes

TUSC2

Step 2: Sort (version 1.0.3)

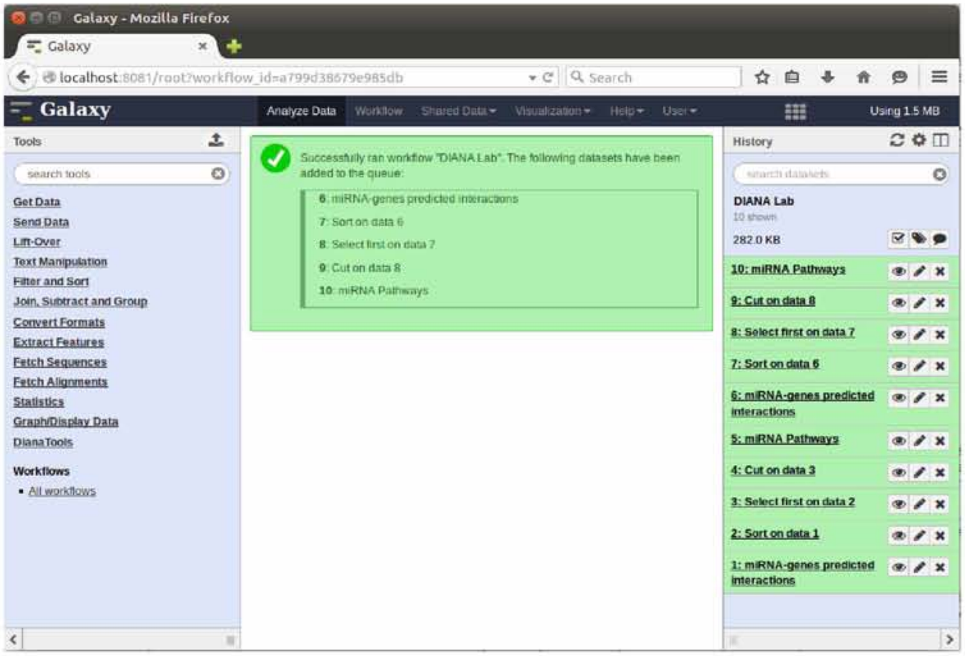
Step 3: Select first (version 1.0.0)

Step 4: Cut (version 1.0.2)

Step 5: mirPath (version 1.0.0)

□ Send results to a new history

Figure 50: Providing argument to DIANA Lab Workflow gene name TUSC2



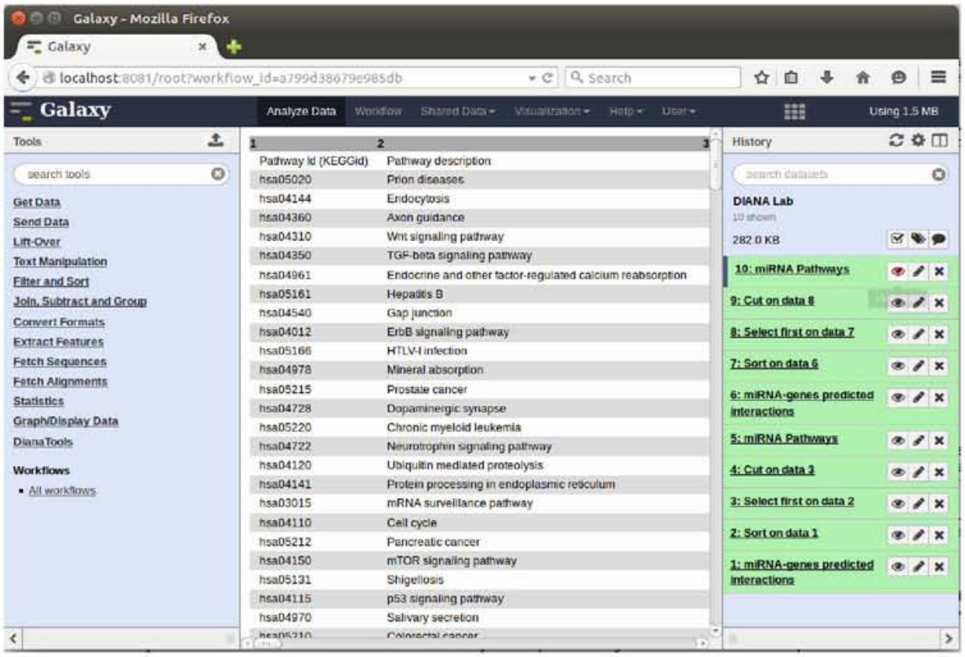
Successfully ran workflow "DIANA Lab". The following datasets have been added to the queue:

- 6: miRNA-genes predicted interactions
- 7: Sort on data 6
- 8: Select first on data 7
- 9: Cut on data 8
- 10: miRNA Pathways

DIANA Lab
10 shown
282.0 KB

10: miRNA Pathways
9: Cut on data 8
8: Select first on data 7
7: Sort on data 6
6: miRNA-genes predicted interactions
5: miRNA Pathways
4: Cut on data 3
3: Select first on data 2
2: Sort on data 1
1: miRNA-genes predicted interactions

Figure 51: Workflow execution results. It shows information about the outcome of the workflow execution, along with the resulting Datasets.



1	2
Pathway Id (KEGGid)	Pathway description
hsa05020	Prion diseases
hsa04144	Endocytosis
hsa04360	Axon guidance
hsa04310	Wnt signaling pathway
hsa04350	TGF-beta signaling pathway
hsa04961	Endocrine and other factor-regulated calcium reabsorption
hsa05161	Hepatitis B
hsa04540	Gap junction
hsa04012	ErbB signaling pathway
hsa05166	HTLV-I infection
hsa04978	Mineral absorption
hsa05215	Prostate cancer
hsa04728	Dopaminergic synapse
hsa05220	Chronic myeloid leukemia
hsa04722	Neurotrophin signaling pathway
hsa04120	Ubiquitin mediated proteolysis
hsa04141	Protein processing in endoplasmic reticulum
hsa03015	mRNA surveillance pathway
hsa04110	Cell cycle
hsa05212	Pancreatic cancer
hsa04150	mTOR signaling pathway
hsa05131	Shigellosis
hsa04115	p53 signaling pathway
hsa04970	Salivary secretion
hsa05210	Colorectal cancer

Figure 52: miRPath pathways provided by the Workflow execution. The two first columns are depicted, containing Pathway KEGG id and Pathway description

References

[1] Lee, R.C., R.L. Feinbaum, and V. Ambros, *The C. elegans Heterochronic Gene lin-4 Encodes Small RNAs with Antisense Complementarity to lin-14*, 1993. 75(5): p. 843-854.

[2] Reinhart, B.J., et al., *The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans.*, Nature, 2000. 403(6772): p. 901-6.

[3] Pasquinelli, A.E., et al., *Conservation of the sequence and temporal expression of let-7 heterochronic regulatory RNA.*, Nature, 2000. 408(6808): p. 86-9.

[4] Bartel, B. and D.P. Bartel, *MicroRNAs: at the root of plant development?*, Plant Physiol, 2003. 132(2): p. 709-17.

[5] Berezikov, E. and R.H. Plasterk, *Camels and zebrafish, viruses and cancer: a microRNA update.*, Hum Mol Genet, 2005. 14 Spec No. 2: p. R183-90.

[6] Cai, X., C.H. Hagedorn, and B.R. Cullen, *Human microRNAs are processed from capped, polyadenylated transcripts that can also function as mRNAs.*, RNA, 2004. 10(12): p. 1957-66.

[7] Borchert, G.M., W. Lanier, and B.L. Davidson, *RNA polymerase III transcribes human microRNAs.* Nature structural & molecular biology, 2006. 13(12): p. 1097- 101.

[8] Lee, Y., et al., *MicroRNA maturation: stepwise processing and subcellular localization.* EMBO J, 2002. 21(17): p. 4663-70.

[9] Lee, Y., et al., *The nuclear RNase III Drosha initiates microRNA processing.* Nature, 2003. 425(6956): p. 415-9.

[10] Denli, A.M., et al., *Processing of primary microRNAs by the Microprocessor complex.* Nature, 2004. 432(7014): p. 231-5.

[11] Gregory, R.I., et al., *Human RISC couples microRNA biogenesis and posttranscriptional gene silencing.* Cell, 2005. 123(4): p. 631-40.

[12] Han, J., et al., *The Drosha-DGCR8 complex in primary microRNA processing.* Genes & development, 2004. 18(24): p. 3016-27.

[13] Hutvagner, G., et al., *A cellular function for the RNA-interference enzyme Dicer in the maturation of the let-7 small temporal RNA.* Science, 2001. 293(5531): p. 834-8.

[14] Khvorova, A., A. Reynolds, and S.D. Jayasena, *Functional siRNAs and miRNAs exhibit strand bias.* Cell, 2003. 115(2): p. 209-16.

[15] Kim, V.N., *MicroRNA precursors in motion: exportin-5 mediates their nuclear export.* Trends in cell biology, 2004. 14(4): p. 156-9.

[16] miRNA Pathway, accessed online May 2015, http://www.sigmaaldrich .com/life-science/functional-genomics-and-rnai/mirna/learning-center /mirna-introduction.html

[17] Bartel, B. and D.P. Bartel, *MicroRNAs: at the root of plant development?* Plant Physiol, 2003. 132(2): p. 709-17.

[18] Lim, L.P., et al., *Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs.* Nature, 2005. 433, 769–773.

[19] Bartel, D.P., *MicroRNAs: target recognition and regulatory functions.* Cell, 2009. 136(2): p. 215-33.

[20] Kim, V.N., J. Han, and M.C. Siomi, *Biogenesis of small RNAs in animals.* Nature reviews. Molecular cell biology, 2009. 10(2): p. 126-39.

[21] Friedman, R.C., et al., *Most mammalian mRNAs are conserved targets of microRNAs.* Genome research, 2009. 19(1): p. 92-105.

[22] Hafner, M., et al., *Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP.* Cell, 2010. 141(1): p. 129-41.

[23] Dai, R. and S.A. Ahmed, *MicroRNA, a new paradigm for understanding immunoregulation, inflammation, and autoimmune diseases. Translational research : the journal of laboratory and clinical medicine*, 2011. 157(4): p. 163- 79.

[24] *Wang, H. and D.Q. Peng, New insights into the mechanism of low high-density lipoprotein cholesterol in obesity.* Lipids in health and disease, 2011. 10: p. 176.

[25] Erson, A.E. and E.M. Petty, *MicroRNAs in development and disease.* Clinical genetics, 2008. 74(4): p. 296-306.

[26] Guay, C., et al., *Diabetes mellitus, a microRNA-related disease?* Translational research : the journal of laboratory and clinical medicine, 2011. 157(4): p. 253- 64.

[27] Ono, K., Y. Kuwabara, and J. Han, *MicroRNAs and cardiovascular diseases.* The FEBS journal, 2011. 278(10): p. 1619-33.

[28] Peterson, S.M.; Thompson, J.A.; Ufkin, M.L.; Sathyanarayana, P.; Liaw, L.; Congdon, C.B. *Common features of microRNA target prediction tools.* Front. Genet. 2014, 5, 23.

[29] Orom UA and Lund AH. *Experimental identification of microRNA targets.* Gene, 2010. 451: 1–5.

[30] W3C. *Web Services Glossary* Accessed online `http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/`, May 2015

[31] W3C. *Web Services Architecture* Accessed online `http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/`, May 2015

[32] Fielding R: *Architectural Styles and the Design of Network-based Software Architectures.* PhD thesis. University of California, Irvine, Information and Computer Science; 2000.

[33] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, et al. *The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud.* Nucleic Acids Res, 41 (Web Server issue) (2013), pp. W557–W561

[34] B Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, *Scientific workflow management and the kepler system.* Research articles, Concurr. Comput. : Pract. Exper., vol. 18, no. 10, pp. 1039–1065, 2006.

[35] Goecks, J, Nekrutenko, A, Taylor, J and The Galaxy Team, *Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences.* Genome Biol. 2010 Aug 25;11(8):R86.

[36] Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J., *Galaxy: a web-based genome analysis tool for experimentalists.* Current Protocols in Molecular Biology. 2010 Jan; Chapter 19:Unit 19.10.1-21.

[37] Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A., *Galaxy: a platform for interactive large-scale genome analysis.* Genome Research. 2005 Oct; 15(10):1451-5.

[38] Durinck,S., Moreau,Y., Kasprzyk,A., Davis,S., De Moor,B., Brazma,A. and Huber,W. *Biomart and bioconductor: a powerful link between biological databases and microarray data analysis.* 2005 Bioinformatics, 21, 3439–3440.

[39] Wilkinson M., Schoof H., Ernst R. and Haase D., *BioMOBY successfully integrates distributed heterogeneous bioinformatics web services.* The PlaNet Exemplar Case. 2005 Plant Physiol., 138, 5–17.

[40] S. Pillai, V. Silventoinen, K. Kallio, M. Senger, S. Sobhany, J. G. Tate, S. S. Velankar, A. Golovin, K. Henrick, P. Rice, P. Stoehr, and R. Lopez, *Soap-based services provided by the european bioinformatics institute.* Nucleic Acids Research, vol. 33, no. Web-Server-Issue, pp. 25–28, 2005.

[41] Bhagat,J., Tanoh,F., Nzuobontane,E., Laurent,T., Orlowski,J., Roos,M., Wolstencroft,K., Aleksejevs,S., Stevens,R., Pettifer,S. et al. *BioCatalogue: a universal catalogue of Web Services for the life sciences.* NucleEnrichedic Acids Res., 38, W689–W694, 2010

[42] Goble, C.A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., Borkum, M., Bechhofer, S., Roos, M., Li, P., and De Roure, D. (2010): *myExperiment: a repository and social network for the sharing of bioinformatics workflows.* Nucl. Acids Res. 38 (suppl 2): W677-W682.

[43] Karolchik, D., Hinrichs, A.S., Furey, T.S., Roskin, K.M., Sugnet, C.W., Haussler, D., and Kent, W.J. *The UCSC Table Browser data retrieval tool.* 2004, Nucleic Acids Res. 32: D493–D496.

[44] Aurrecoechea C, Barreto A, Brestelli J, Brunk BP, Cade S, Doherty R, Fischer S, Gajria B, Gao X, Gingle A. *EuPathDB: The Eukaryotic Pathogen database.* Nucleic Acids Res. 2013;41:D684-D691.

[45] Jeggari, A., D.S. Marks, and E. Larsson, *miRcode: a map of putative microRNA target sites in the long non-coding transcriptome.* Bioinformatics, 2012. 28(15): p. 2062-3.

[46] Ponting, C.P., P.L. Oliver, and W. Reik, *Evolution and functions of long noncoding RNAs.* Cell, 2009. 136(4): p. 629-41.

[47] Daniel Blankenberg, Gregory Von Kuster, Emil Bouvier, Dannon Baker, Enis Afgan, Nicholas Stoler, the Galaxy Team, James Taylor and Anton Nekrutenko, *Dissemination of scientific software with Galaxy ToolShed.* Genome Biology 2014, 15:403

[48] Paraskevopoulou MD, Georgakilas G, Kostoulas N, Vlachos IS, Vergoulis T, Reczko M, Filippidis C, Dalamagas T, Hatzigeorgiou AG., *DIANA-microT web server v5.0: service integration into miRNA functional analysis workflows.*, Nucleic Acids Res. 2013 Jul;41(Web Server issue):W169-73.

[49] Reczko M, Maragkakis M, Alexiou P, Grosse I, Hatzigeorgiou AG., *Functional microRNA targets in protein coding sequences.*, Bioinformatics. , 2012 Jan 27.

[50] M. Maragkakis, T. Vergoulis, P. Alexiou, M. Reczko, K. Plomaritou, M. Gousis, K. Kourtis, N. Koziris, T. Dalamagas, AG. Hatzigeorgiou *DIANA-microT Web server upgrade supports Fly and Worm miRNA target prediction and bibliographic miRNA to disease association* Nucleic Acids Research 2011 (Webserver issue) [PubMed].

[51] T. Vergoulis, I. Vlachos, P. Alexiou, G. Georgakilas, M. Maragkakis, M. Reczko, S. Gerangelos, N. Koziris, T. Dalamagas, A. G. Hatzigeorgiou *Tarbase 6.0: Capturing the Exponential Growth of miRNA Targets with Experimental Support.* Nucl. Acids Res. (2012) 40 (D1): D222-D229.

[52] I. S. Vlachos, M. D. Paraskevopoulou, D. Karagkouni, G. Georgakilas, T. Vergoulis, I. Kanellos, I-L. Anastasopoulos, S. Maniou, K. Karathanou, D. Kalfakakou, A. Fevgas, T. Dalamagas and A. G. Hatzigeorgiou. *DIANA-TarBase v7.0: indexing more than half a million experimentally supported miRNA:mRNA interactions.* Nucl. Acids Res. (2014)

[53] I. S. Vlachos, N. Kostoulas, T. Vergoulis, G. Georgakilas, M. Reczko, M. Maragkakis, M. D. Paraskevopoulou, K. Prionidis, T. Dalamagas, A. G. Hatzigeorgiou *DIANA miRPath v.2.0: investigating the combinatorial effect of microRNAs in pathways* Nucleic Acids Research 2012 (Web server issue)

[54] Kozomara,A. and Griffiths-Jones,S. (2011) *miRBase: integrating microRNA annotation and deep-sequencing data.* Nucleic Acids Res., 39, D152–D157.

[55] Flicek,P., Amode,M.R., Barrell,D., Beal,K., Brent,S., Carvalho-Silva,D., Clapham,P., Coates,G., Fairley,S., Fitzgerald,S. et al. (2012) *Ensembl 2012.* Nucleic Acids Res., 40, D84–D90.

[56] Apache. Maven Project. Accessed Online http://maven.apache.org/, May 2015.

[57] Bedoya-Reina O, Ratan A, Burhans R, et al., *Galaxy tools to study genome diversity.*, Gigascience 2013;2:17.

[58] Paraskevopoulou MD, Georgakilas G, Kostoulas N, Reczko M, Maragkakis M, Dalamagas TM, Hatzigeorgiou AG., *DIANA-LncBase: experimentally verified and computationally predicted microRNA targets on long non-coding RNAs.*, Nucl. Acids Res. (2013) 41 (D1)

[59] Karasavvas,K., Wolstencroft,K., Mina,E., Cruickshank,D., Williams,A., De Roure,D., Goble,C. and Roos,M. , *Opening new gateways to workflows for life scientists.*, Stud. Health Technol. Inform., (2012) 175, 131–141