UNIVERSITÀ
DEGLI STUDI
DI PADOVA

HEAD OFFICE: UNIVERSITÀ DEGLI STUDI DI PADOVA

Department of Biology

Ph.D. COURSE IN: Biosciences
CURRICULUM: Genetics, Genomics and Bioinformatics
SERIES 35°

Exploring biological signals: from pathway visualization to spatial transcriptomics

Coordinator: Prof. Ildikò Szabò
Supervisor: Prof. Gabriele Sales
Co-Supervisor: Prof. Chiara Romualdi

Ph.D. student: Davide Corso

Index

Ph.D. thesis synopsis

The present Ph.D. thesis will describe different projects concerning two main topics: the visualization of biological networks and the identification of spatially variable genes (SVGs), which are genes that have a spatial pattern of expression (they can be identified from Spatially Resolved Transcriptomics (SRT) data, as it preserves spatial information of the tissue's cells). Although those projects are involved in two different topics, they share a common theme, the visualization of biological data.

The visual analysis of data representation is a direct way to comprehend highly complex data. Indeed, an accurate network drawing is essential to convey and access graph information, which in turn may highlight key elements such as genes, interactions, or even communities. Similarly, visualization is necessary to investigate clear spatial patterns of expression of the SVGs identified, which can be related to the spatial changes of the tissue under study.

The first two projects regard the visualization of biological networks. Specifically, the main part of my Ph.D. was devoted to the itGraph project (the first to be described) and concerns a novel web tool to explore pathways of interest with three different

network perspectives. To enhance the visualization of the provided biological networks, it integrates features that are still lacking in other similar software, which also improves the tool's utility and the user experience. The full period of my Ph.D. was required to develop and optimize all the scripts, the server (backend), the database, and the client (front end), including the graphical interface.

The second project concerning network visualization is called MyoData, which was completed and published in the "Computational and Structural Biotechnology Journal". It is a comprehensive and integrated resource for single myofiber and nucleus miRNA:lncRNA:mRNA coregulatory networks, also evaluating their impact in relation to known pathways such as those present in the KEGG collection. It integrates a minimal version of the network visualization tool, which was important to understand the technical aspects of visualization library and layouts, and consequently optimize the development of itGraph structure.

The other two described projects, concern the identification of SVGs. At first, I developed SpatialDE, an R package wrapping of the SpatialDE Python method, whose purpose is to identify spatially variable genes. This wrapper can create a python environment inside the R domain and thus performs the original SpatialDE functions. It was created to respond to a challenge proposed at the EuroBioc2020 Conference and was published on Bioconductor in October 2021 with high-quality, well-documented, and interoperable software.

The last project is called VoyageR (the repository's name) and is still in progress. The aim is to conduct a benchmark for R and Python methods able to identify SVGs. Since this is one of the most popular analyses eventually performed on SRT data (that can preserve spatial information of the cells' tissue), SVGs can represent potential markers of biological processes and thus can be used for downstream analyses. Similar methods for the same purpose continue to be published; for this reason, a comprehensive benchmark, which is still lacking, could be helpful for users to choose the best suitable procedures for their use. Moreover, my project was designed to be extensible to simplify the addition of further methods.

To conclude, the present Ph.D. thesis describes different improvements of the respective topics. Producing an effective and scalable visualization is becoming a challenging task. For example, the growing size of available data is increasing the complexity of the described networks. This is especially true for large graphs, which present technical aspects that make it difficult to layout them nicely by algorithms, thus often results in incomprehensible 'hairball' from which it is difficult to extract information. Despite these issues, my projects were designed with specific optimization to provide new tools both for biologists and bioinformatic users: alternative visualization perspectives of pathway networks with the integration of extra features and providing one of the most common analysis methods for STR data in the R environment.

itGraph

Summary

Understanding complicated biological networks are essential to solve contemporary problems in Systems Biology. An accurate visualization is a common way to access network information, as it allows highlighting of key elements and simplifies the extraction of information. However, providing a suitable drawing is not a trivial task. As human perception is different among individuals, and network drawing may differ depending on what kind of information needs to be displayed, there may not exist a single best drawing. Nevertheless, I believe that optimizing specific aspects with a more precise focus on the target of the tool has greater impact and success, providing a more useful visualization. Indeed, the project aimed to create a web tool for pathway visualization, combining visual appeal, interactivity, and other features which are still lacking in other software.

One of the main integrations regards the nodes' positions of each network that are pre-computed to let the user wait only for the rendering time of the network objects.

To increase the biological accuracy, the tool integrates the subcellular location with three visualization perspectives: "simple network", "network with compartments", and "power graph". The first one is a traditional drawing of a graph. Nodes are simply encoded as points in the space but colored according to the respective location. The results obtained are usually aesthetically pleasing and useful mostly for small-medium graphs.

The second type regards the explicit representation of subcellular organelles as compound nodes. Biological networks describe biological entities which most act their functions in a specific location inside the cell. This type of representation provides an insightful way to understand interactions that define molecular processes that span different compartments. The used approach is a novel solution that allows for the creation of a non-minimal and non-simplified representation of hierarchical cell compartments.

The third type of visualization regards the Power Graph analysis proposed by Royer et al., 2008, a method to describe networks in a compact and less redundant representation, reducing the visual complexity of the network. This drawing approach allows for handling huge networks, otherwise not representable. Indeed, in classical representation, they result in hairballs, from which few insights can be gathered. A power graph can be useful to give a new insightful drawing of the original graph, as

it encodes high-density structure motifs that are widely represented in biological networks.

There are other integrated features to improve the user experience, like the conversion of the identifiers, which allows for mapping network nodes to different biological identifiers.

The tool provides more than 170 thousand pathways distributed among 14 species, for a total of more than 500 thousand of network visualizations. It is designed to be intuitive and user-friendly, with no bioinformatic expertise required.

1. Introduction

In this section, I will describe the main topics of my project, concerning a novel web tool for visualizing pathway networks. Biological networks are widely represented as graph models. Edges may represent distinct biological interactions, while nodes may describe various biological entities, like proteins or genes. This chapter starts with a description of different biological networks. I will discuss graph structures, properties, layouts, and complex designs able to depict graphical annotations, such as compound nodes used to illustrate subcellular compartments. The introduction will also describe additional technical and biological features that may enhance the utility of the visualization, and lastly, it will provide an overview of the state-of-the-art of similar tools with the explicit representation of subcellular locations.

Protein-Protein interaction network

Protein-protein interaction network (PPI) contains information about different proteins and their interactions in a specific biological process [1]. All the PPIs together that can take place within a cell build the "interactome" [2]. Proteins have a key role inside the cell, as their interactions are fundamental to perform their functions to

control molecular and cellular mechanisms. Thus, understanding the underlying biological information stored in PPI networks is essential for the knowledge of complex biological systems. The analysis of this type of network is important as it provides a perspective regarding the importance of the system components and makes quantitative predictions for system-level understanding [3].

Metabolic network

Cell growth and maintenance are performed with biochemical reactions catalyzed by enzymes that transform chemical substances (reactants) into other substances (products). Often there exists a path of reactions in which one product can be the reactant of the next reaction. It is also possible that metabolites are involved in different metabolic pathways, where they act as a co-substrate for a specific reaction and may have main roles for another reaction [4]. Metabolic networks contain information about substrates, product metabolites, and biochemical reactions involved [5].

The analysis of cellular metabolite levels, referred to as metabolomics, is a very complex task due to the typical high connectivity and complexity of such a network. However, it is important to gather underlying biological knowledge of cellular metabolism at large scale [6]. Moreover, an understanding of the generic properties of complex networks is useful to gather information about the structure of this network [7].

Signaling Network

Signaling Networks describe the process through which cells respond to a specific internal or external stimulus to coordinate the regulation of its activity and respond to changes in their immediate environment. These signals are bound by specific proteins, called receptors, that initiate the response process. Successively, this signal is converted by involving sequences and chemical reactions by other proteins [4] and carried through different cell compartments to get the desired phenotype [5]. There are three main classes of signaling molecules: hormones, signaling molecules of the endocrine system; neurotransmitters, signaling molecules of the nervous system; cytokines: signaling molecules of the immune system. Also, there are different classes of signaling: *intracrine* describes signals that are produced and remain inside the cell; *autocrine* describes a signal that is produced inside the cell, then secreted to the external environment and this signal can affect the same cell; *juxtacrine* describes a signal by the cell which in turn can affect adjacent cells through cell contact; *paracrine* describe a signal produced by a cell which in turn can affect nearby cells without requiring cell contacts; *endocrine* describe signal produced by a cell which in turn can reach other cells in different parts of the body through the circulatory system [4].

Gene Regulatory Networks

Gene regulatory network collects different interactions of different molecular species that control gene-product abundance [8]. This type of network is also denoted as GRN, it is inferred by gene expression data and provides information about regulatory interaction with potential targets. Over the years, different methods for inferring GRN have been proposed to gain new information from these networks, as they are not considered as final results. Indeed GRN can be referred to as a "blueprint" or "map" of molecular interactions, and such a network can be used to gather novel biological interactions to be further validated in wet-lab experiments [9].

Pathways and biological networks as a graph model

During the last decades, technological and scientific progress allows an increasing massive production of biological high throughput data in different fields, for example genomic, transcriptomic, and proteomic. This data is characterizing the field of biology in the current era and is leading to increasing both the size of data repositories and the complexity of the biological topics under study. However, all these technological improvements were essential to enriching knowledge of the biological aspects of interest.

One of the computational challenges of this scenario regards the management and the analysis of all these data. For many areas of computational biology, typically there is

one or more analysis that is sequentially performed starting from a huge dataset, filtering and simplifying them, and finally extracting the significant underlying biological information.

Systems Biology is one of these computational fields and it focuses on the comprehensive analysis of the relations among different biological factors, which leads to complex and different biological networks, e.g. protein-protein interaction (PPI), gene regulatory, or signaling network [5].

Biomolecular networks, often called pathways [10], are a standard model to describe and represent the reactions and actions of a series of molecules in a cell. A state change in a cell could be indeed produced by activating a specific pathway, which can lead to transcribed genes, new molecules, or new signals that are recognized to activate new downstream reactions. Thus, the study of pathways has a key role both in understanding cell processes and to interprets -omics data, as they provide the biological context for a given observation [11].

Pathways and biological networks are represented as a **graph** structure. Graphs are abstract mathematical objects that allow describing any type of relationship between entities or objects.

There are different types of graphs, but in a typical formal definition, a graph $G = (V, E)$ is composed of a series of vertices V (or nodes) and a series of edges E (or links). A subgraph $G' = (V', E')$ is a graph where V' is a subset of V and E' is a subset of E.

If all the edges of the graph have no direction the graph is called **undirected**. In this case, each edge can be crossed in both directions.

If the edge has an associated direction, the graph is called **directed** and they are useful to represent signal transduction pathways or gene regulation networks. Moreover, if the graph does not contain any cycle, it is called **directed acyclic graph** (DAG).

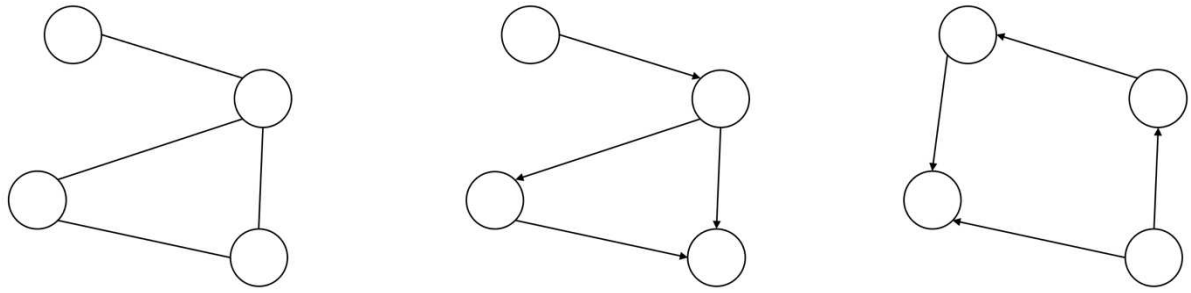


Fig. 1.1: A) undirected graph; B) directed graph; C) DAG: Directed Acyclic Graph.

A single graph that contains both directed edges and undirected edges, is called a **mixed graph** and it can be denoted with the following triple: $G = (V, E, \vec{E})$. Mixed graphs are useful to represent cell signaling pathways [12] in which some directed edges represent activation or inhibition action whereas other undirected edges describe physical binding protein interaction [13].

A **path** is a non-empty graph (an empty graph occur when $V = \emptyset$ and $E = \emptyset$) $P = (V, E)$ with a series of distinct vertices $V = \{x_0, x_1, \ldots, x_k\}$ and edges with the following pattern $E = \{(x_0, x_1), (x_1, x_2), \ldots, (x_{k-1}, x_k)\}$. The length of a path P is the number of edges [14].

A **weighted graph** is a graph with a number (weight) associated to every edge, referred to as edge-weighted, or associated to every vertex, referred to as vertex-weighted [15]. Weighted graphs are useful to describe the kinetics of biochemical reactions [16], and typically, the weight on the edges describes the relevance of the connection [17].

Another interesting type of graph is the **hypergraph** which is different from the ordinary structure previously described. It allows edges to connect more than two nodes and they can be useful to model metabolic networks, for example in reactions that involve four species ($X + Y \longrightarrow Z + T$) or to represent protein complexes composed of more than two nodes [14].

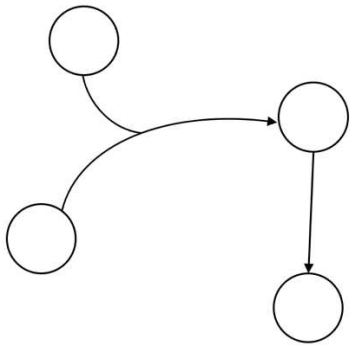


Fig. 1.2: Example of Hypergraph.

An undirected graph in which any two vertices are connected by exactly one path is called a **tree**. A **rooted tree** $T = (V, E, r)$ is a graph in which for every node $u \in V$, except for the node root r, there is a unique path from r to u. For every node $u \in V$, except for the node root r, exist a unique node v, called parent of u, if exists the following edge $(v, u) \in E$; in this case, u is called children of v. A node u is it called leaf

if it does not have a child, otherwise is it called internal node. All the nodes that are on the path from the node root r to u are the ancestors of u, except for u itself [18].

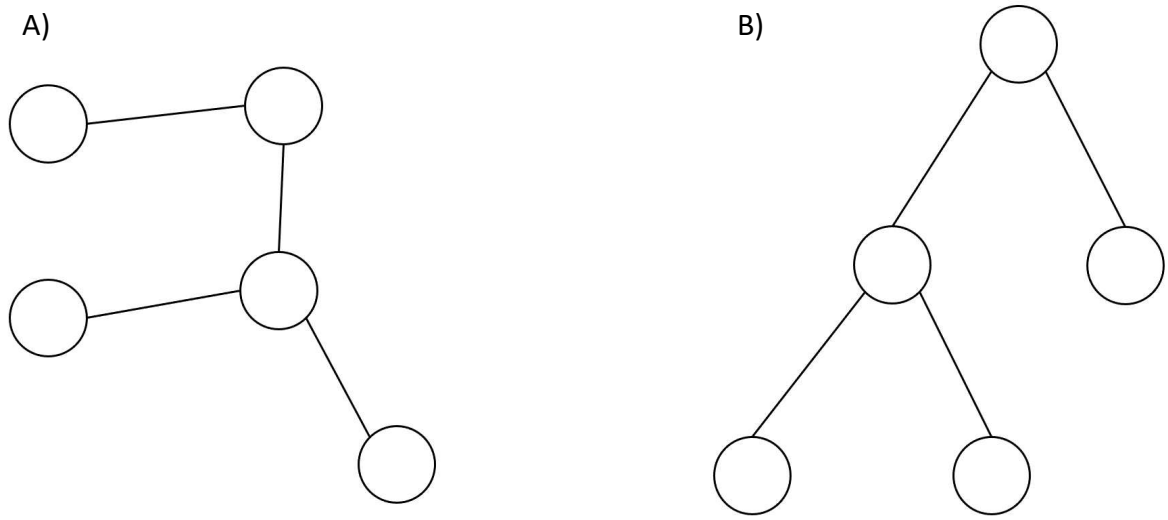


Fig. 1.3: A) Tree; B) Rooted Tree.

A **compound graph** $CG = (V, E, F)$ is defined as a set of vertices V, a set of adjacency edges E and a set of *inclusion edges* F. It is necessary that the inclusion graph $T = (V, F)$, is a rooted tree that shares the same vertices, and no adjacency edge (with each edge $e \in E$) connect a node to one of its descendants or ancestors [19]. The following figure show an example compound graph, built as follow:

$$V = \{a, b, c, d, e, f, g\}$$

$$E = \{(a, b), (a, f), (b, d), (d, e), (c, g)\}$$

$$F = \{fg, fb, fc, gd, ge\}$$

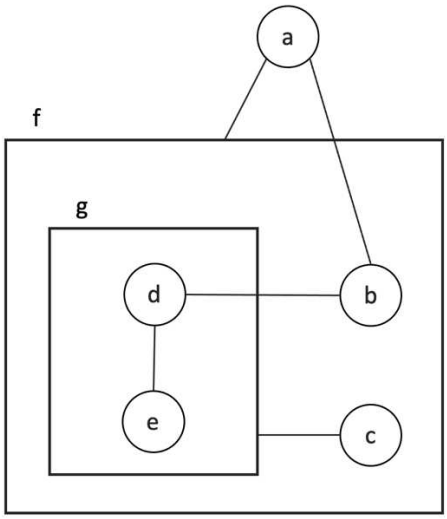


Fig. 1.4: Example of compound graph.

Nodes may also have multiple edges that connect each other, describing different relationships. In this case, it is useful to represent these interconnections with a **multigraph**, in which different edges exist between a couple of nodes describing different functions or interactions [14]. Multigraphs can be very useful to represent pathway interaction as they often have different links between the same nodes describing different functional relationships.

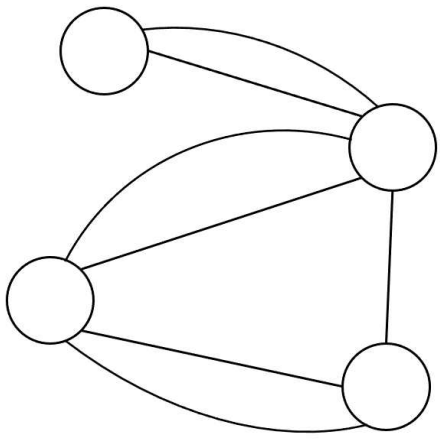


Fig. 1.5: Example of multigraph.

This mathematical object has a lot of properties and topological features that are widely used for various analyses, which are useful to respond to questions like: "which is the most important node", "which node is the bridge between two different communities", etc. It is also important for graph analysis and interpretation, a suitable graphical representation able to describe in clear way network structures, symmetries, and other features that are central for successively analyses or inferring new knowledge [17].

Network visualization and layout algorithms

Network drawing and visualization are important research tasks, essential for the interpretation and understanding of biomedical data. As previously described, the growing size of available data is also increasing the complexity of the described networks. Thus, producing an effective and scalable visualization is becoming a challenging task, especially for large networks [5]. Moreover, as human perception is different among individuals, and network drawing may differ depending on what kind of information needs to be displayed, there may not exist a single best drawing. The utility of a specific drawing may also depend on the type of user that will use it; for example, biologists would expect that two interacting proteins would be drawn next to each other [14].

For these reasons, network visualization is an important bottleneck that requires efficient algorithms and tools [17]. Despite these difficulties, over the years, several layout algorithms and software have been published to produce and provide suitable network drawings. Furthermore, some conventions, e.g. representing proteins as circles, and technical aesthetic criteria have been introduced to provide appealing results and improve the overall readability of the networks. However, the algorithms cannot optimize all the criteria as some of these properties contradict each other. Here, are briefly described some of the most common aesthetic criteria [14,20,21]:

- *minimization of edge crossing*: a high number of overlapping edges can affect the interpretation of nodes interactions;
- *minimization of edge bends*: minimizing the number of bends along the edge is important as it allows the human eye to easily follow an edge;
- *minimization of required area*: saving space is important also with a homogenous density of placed nodes and edges;
- *minimization of overlapping elements*: high number of overlapping nodes may influence the understanding of node information;
- *maximizing symmetry*: is important to reflect the symmetry of the graph when it contains symmetrical information;
- *clustering*: placing together similar nodes can help to understand some graph's structure;
- *uniforming edge length*: important to produce a regular graph.

Producing a proper graph drawing, even optimizing the described criteria, is still challenging as a "suitable visualization" may depend on the application of the network and individual user preferences [21]. In [22–24] co-authors performed different studies analyzing the impact of aesthetic properties on the readability of graph drawing. Their results describe that some of the most important criteria to improve readability are the minimization of edge crossing, minimization of edge bends, and maximization of symmetry. Moreover, when asked participants to draw graphs and lay them out "nicely" it was found that users also emphasized the clustering properties.

The readability of networks is a crucial requirement to correctly analyze graph information. Moreover, it becomes more difficult as the graph size increases, both for computational and biological aspects. For these reasons, there is a clear demand for appropriate layout algorithms able to produce suitable drawings that convey biological information in a comprehensible diagram.

At the core of any network visualization, there is a layout algorithm that decides how to position nodes and edges in order to produce suitable drawings that allow users to simplify the analysis and the understanding of graph information. Their goal is to find an organization and coordinates of nodes that highlight underlying structures. Here are briefly described (without too technical details) the most known layout algorithms [5,14,25]:

- *Random Layout Algorithm*: it places nodes in random coordinates on the screen. The results typically present a high number of edge crossings, so it may be useful only for small graphs.
- *Circular Layout Algorithm*: nodes are placed in succession in a circular arrangement. All the nodes have the same distance from the center and it minimizes the number of overlapping elements. It highlights nodes with the highest degree or nodes hub.
- *Hierarchical Layout Algorithm (HLA)*: developed by Sugiyama [26], it places nodes in different hierarchical groups, reducing the number of edge crossings. Even if it is scalable, efficient, and provides pleasant results, it is not suitable for large graphs (with thousands of nodes and edges respectively) as the minimization of edge crossing is an NP-complete problem.
- *Fruchterman & Reingold and Eades & Perter Algorithms*: Fruchterman & Reingold [27] and Eades & Perter [28], also known as Spring Embedded Algorithm, belong to the class of Force-Directed Algorithm (FDA). This class of algorithms is widely used in network layout, and it is one of the most implemented among different visualization libraries. It considers each node as an electrically charged element and each edge as a spring. Specifically, it computes attractive force between every pair of adjacent vertices and repulsive forces between non-adjacent vertices. Thus, in this model connected nodes attract each other, like a spring based on Hooke's law, and repulse non-connected nodes like electrically

charged particles based on Coulomb's law. This algorithm performs iteratively until an equilibrium state is reached and its computational complexity is $O(n^2)$ per iteration, where n is the number of node [29], thus their computation is very slow when applied to large graphs.

- *Kamada-Kawai Algorithm*: developed by T. Kamada and S. Kawai, is based on the concept of theoretic distance between nodes [30]. In this algorithm, the forces between nodes are computed based on the lengths of shortest paths between each couple of nodes.
- *Tree Layout Algorithm*: it places nodes in a tree arrangement without cycles and with a hierarchical organization of the nodes. It is efficient and scalable also for large graphs, but the main drawback is the placements of a huge number of nodes in the limited area of the screen.
- *Simulated Annealing Algorithm (SA)*: is an algorithm that represents the space of the visualization problem as a set of states, each one with associated energy. The goal is to find the state with the minimum energy (e.g., below a threshold). SA computation is composed of three steps: 1) definition of a starting point, typically chosen randomly; 2) SA selects points nearby the current solution and determines whether the new point has a better or worse associated energy than the current one. If better it becomes the next point, otherwise, the algorithm can still make it the next point to escape from a local minimum; 3) an evaluation based on predetermined criteria (e.g. the number of iterations exceeds the

maximum number of iterations) is performed to terminate the procedure. The algorithm is time-consuming for large graphs, as the search space increase depending on the number of nodes.

- *Clustering Layout Algorithm*: it reduces the visual complexity of the graph and allows to group together similar nodes based on the definition of specific metrics. These metrics can be content-based if they rely on node content, while if they rely on the structure of the graph, it is called structured-based. This layout improves the readability of the network and allows for the identification of important graph features like nodes hub, nodes degree, or connectivity.
- *Grid Layout Algorithm*: it places nodes on a 2-dimensional squared grid. This layout algorithm models the node graph as particles interacting together, specifically nodes closely related attract each other, while remotely related nodes repulse each other. Results avoid node overlapping but one limitation lies in the high number of edge crossing, which can affect the identification of complex blocks and nodes hub.
- *3D Layout Algorithm*: this layout positions nodes in a 3-dimensional space. Using three dimensions adds more available space making it easier to optimize aesthetic criteria and visualize large networks. This kind of layout has to include new features that are required for the dynamic change of the view, like transparency or depth.

Graphic rendering libraries and implementations

These algorithms can be used to find node coordinates that produce a pleasant arrangement of the network. Each layout has a specific optimization focus of aesthetic criteria, but not all the algorithms can be used for large networks. This is one of the major bottlenecks in the visualization of biological networks. As I will discuss more in detail in the next paragraphs, huge networks affect visualization in a wide range of aspects.

High throughput technologies allow the massive production of biological data, increasing the size and complexity of biological pathways and networks. Consequently, it is challenging to provide a readable network visualization that allows the analysis of the graph.

Regarding technical aspects, a suitable visualization is obtained through a multi-step process that tries to handle and optimize aspects such as the choice of the layout algorithm, the visualization library, or the data structure storing network information. Moreover, these features may also depend on the goal of the visualization, for example, its final application (whether as a web tool or a local application) and the types of graphs covered.

Furthermore, a suitable visualization must satisfy different requirements [5,25]:

- fast and clear rendering, particularly for huge networks;
- easy network queries through zoom and focus;

- integration of standard network annotation e.g. molecular function or node localization;
- provide different layouts and interactive functionalities;
- compatibility with import and export standard data format for biological networks.

Among the described parameters, one of the most important aspects is to provide the final rendering of the layout network in a fast and clear way. In particular, different technical characteristics affect this requirement and a lack of optimization of the following parameters can weigh the responsivity of the result:

- an efficient data structure that stores the network information;
- suitable rendering visualization library;
- choice of the layout algorithm.

The optimization of the data structure is a critical aspect of network visualization, especially for a web tool. For example, in the biological field, various annotations can be added to increase the biological accuracy of the entities in the network, and often these annotations are descriptive text, affecting the size of the object that has to be transferred in the HTML request. Indeed, in typical client-server-database communication, in addition to necessary default processing processes, there are response times that strongly depend on the size of the transferred data. Lack of optimization in this data structure could affect the waiting time of the user for the results, which in turn may affect the perception of the reliability of the tool. Moreover,

this kind of optimization is even more important for large networks (with thousands of nodes and thousands of edges, or more), and it is one of the main aspects for a web tool to take into consideration.

Regarding the choice of layout algorithm, from a technical point of view, it is strictly linked to the choice of the rendering libraries, and one could affect the other. Often network visualization libraries have internal implementation of layout algorithms, but not all the layout types are implemented in graphic libraries. Thus, both choices are strictly linked as one may filter out some options of the other. These choices are even more critical whether it is necessary to compute the network layout in a real-time and interactive way, as the user may also have to wait for the computation time before being able to use the result. As described in the previous paragraph, the utility of the algorithms may also depend on the size of the network, as not all the algorithms are suitable for an interactive tool, due to a high computation time.

The possible choices of the layout algorithm are even more restricted whether the graph has complex structures like compound nodes. These types of graphs have been used to represent complex relationship structures [18,31,32], sometimes even with a high level of nesting of compound nodes. Due to their atypical structure, they require layout algorithms specifically developed. Over the years, some works have been published that have tried to focus on the layout of hierarchical graphs [33–35] with poor performance for undirected graph instances. Dogrusoz *et al.* [19], proposed the *CoSE* algorithm, which computes the layout for undirected graphs based on the

Fruchterman & Reingold method. Their implementation inserts compound nodes as part of the physical system described by the force-directed. Furthermore, their implementation is able to keep graph components together with the integration of a measure called "gravitational force", manage multiple nesting levels of compound nodes and compute the layout without node fixed size. Balci *et al.* [36], proposed a new layout algorithm for compound graphs, called *fCoSE* (a Fast Compound Spring Embedder), a faster version of the CoSE algorithm which also supports a set of constraints. This algorithm integrates the speed of spectral drawing [37] and the quality of the force-directed layout.

Cola is an alternative algorithm to fCoSE, able to compute both the layout of compound graphs and manage constraints, and it is the result of a series of works [38–40]. Their approach extends the force-directed layout algorithm allowing the insertion of separation constraints such as the positioning of nodes on different horizontal layers, placing nodes on fixed positions, positioning of nodes within defined boundaries (e.g. compound node), or automatic node separation to avoid node overlaps. Cola can manage different types of constraints but has a high computational cost, while fCoSE, with a complexity $O(n + m)$ with $n = |V|$ and $m = |E|$, has a faster computation time so that it can be used in real-time on small or medium graph size.

Both CoLa and fCoSE are implemented in Javascript as extensions of the Cytoscape.js library [41], which is one of the most used libraries in the field of network visualization tools. In addition, CoLa can also be used with the D3.js Javascript library. To my

knowledge, these implementations are the only ones available and usable that can compute layouts for compound graphs.

Biological annotation to improve network visualization

The visual analysis of data representation is a direct way to comprehend highly complex data. As previously described, networks and graphs are represented as node-link diagrams, where nodes are objects or entities, and edges are relations between nodes. This kind of diagram is intuitive and works very well for small-medium instances. However, with the increase in the size of a network, the complexity of the visualization similarly increases.

Representing networks with thousands of nodes and thousands of edges often results in incomprehensible 'hairball' from which it is difficult to extract information [42]. Although huge networks present technical aspects that cannot be fully optimized, such as the minimization of edge-crossing, there are some methods, such as clustering, able to simplify the graph representation, helping the user to perform a visual analysis of the network. Moreover, despite the great potential of layout algorithms to highlight hidden patterns of graphs, often they do not focus on systematic analysis, which makes it difficult to extract information relative to biological questions [5]. Thus, it is important to improve the network representation with different strategies, like graphic design conventions, to highlight key aspects of the graph. Indeed, a display is said to

be effective if it represents the correct and real information in an easily comprehensible way [14].

Typically, entities of the biological network represent proteins, genes, or metabolites, while edges can describe different interactions. Genes or proteins can be drawn as circles, while metabolites as triangles. Different arrow types for edges can represent different interaction types like activation, inhibition, or binding as described by the Systems Biology Graphical Notation (SBGN) [43–46] and shown in Fig. 1.6.

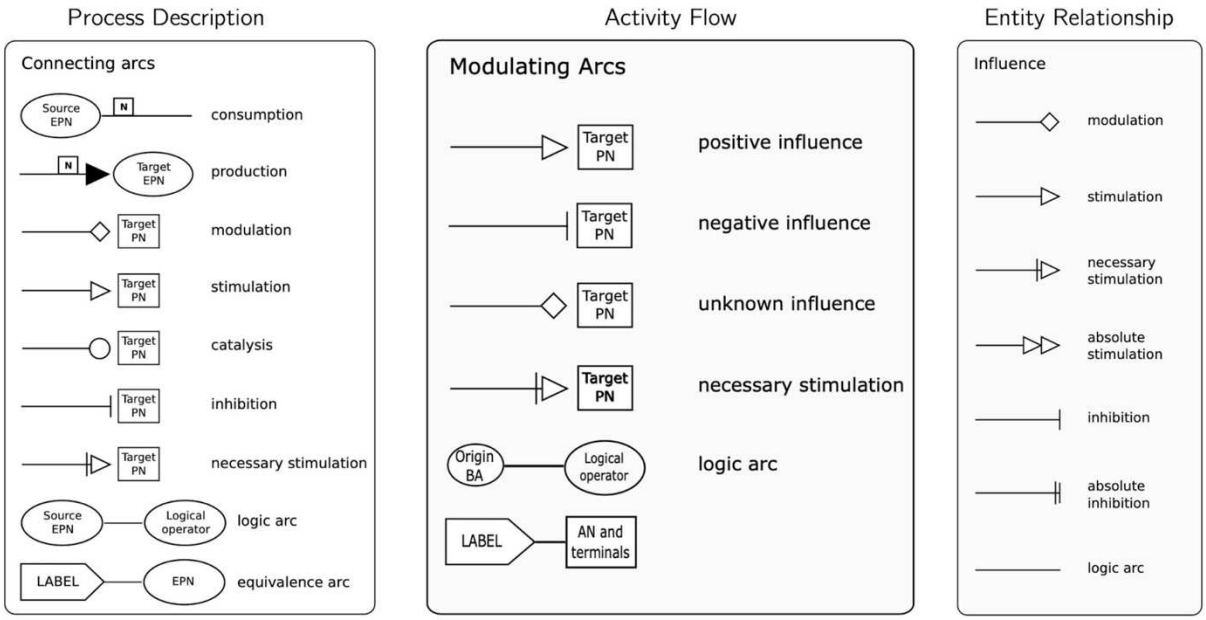


Fig. 1.6: Quick reference of SBGN edges symbols. The "Process Description" section describes all the processes taking place in a biological system. The "Activity flow" describes the flow of activity in a biological system. The "Entity Relationship" describes all the relations involving the entities of a biological system. Image adapted from Rougny et al. 2019; Mi et al. 2015; Sorokin et al. 2015.

Another visual aspect concerns the colors, which can be fundamental for network annotation. This attribute can represent categorical information such as protein localization, biochemical reaction role (product or substrate), or gene expression.

Proteins perform their function in one or more cellular compartments, and in specific cases, some proteins can interact with each other only when they are co-localized. Consequently, cellular localization can be integrated as node information with a graphical attribute like color. The same color should correspond to the same location, and vice versa. Furthermore, for the graphic libraries able to display compound graphs, the cellular localization may be represented as a compound node that contains all the nodes belonging to that compartment. This type of visualization is used by CellWhere [47], and CellNetVis [48], which integrate multiple nested compartments that reflect a simple cellular hierarchy, or Pathway Commons [49]. Furthermore, if the network view includes compound nodes, it would be useful for the user to have the possibility to expand and collapse the compound node manually. This feature would allow highlighting or hiding specific parts of the graphs, and consequently helps the navigation and analysis of other parts of the network.

Nodes can also be colored according to the gene expression experimentally measured by the user imported into the network visualization.

Like nodes, edges can also be colored to describe the different ways in which a given interaction has been obtained. Different colors can help the user to intuitively understand whether a specific interaction has been obtained experimentally, through prediction, or text mining. One of the most popular tools that use this type of annotation is STRING [50].

The node and edge transparency can be modified to describe the significance of a node or an interaction. Similarly, also the node size can be modified to represent its importance, based on a continuous measure, such as the log fold-change or the p-value of a gene.

Similar to the graphic attributes, the biological description of a pathway network can be very useful for the user to contextualize the function of a node or interaction. For example, having the three categories annotations of Gene Ontology may help to study and analyze the network. In particular, having the possibility of knowing the annotation of a node (gene or protein) from external sources, contextual to the specific pathway, can be very useful to know the function of a protein or the activation mechanism described by an edge.

Since these annotations are very descriptive, it is not worth saving them within the data structure of the network as it would increase the size of the object, leading to slower client-server communication. With modern technologies, it is possible to overcome this problem by requesting them in real-time with the APIs from external databases.

In summary, good visualization needs to integrate graphical and textual biological annotation. An effective annotation, provided in an easy and comprehensible way, can simplify the study of the network. However, it is important to note that integrating a high number of annotations can result in the opposite effect providing an unintuitive network.

State of the art of web tool with subcellular location

One of the most important integrations of the presented project is the visualization of biological networks with the explicit representation of cellular compartments. As already mentioned, some proteins may not interact until they are located in the same location, thus the visualization of the network within the cellular organization may be essential to understand, for example, molecular processes that span compartments [5]. In previous years, this type of visualization was considered by few tools as it presents difficulties both for technical and biological aspects. The cell has a complex organization, both among cells belonging to the same organism and among organisms of different species. For example, there are a lot of differences between eukaryotic and prokaryotic cells. This complexity has made it difficult to provide a suitable graph visualization with the cellular organization.

The tool presented here provides a novel and automatic solution for this problem, for this reason, this paragraph will describe the state of the art of web tools that integrate network visualization with the explicit representation of cellular compartments.

Different tools provide alternative approaches, each one with a different focus. CellWhere [47] includes a series of principal cellular compartments. Starting from an input gene list, protein-protein interaction networks are created looking for interactions in Mentha [51]. Successively, locations are obtained from UniProt and/or

Cellular Component (GO) and then mapped up to fifty cellular compartments of CellWhere. However, the user may add a new location or enter a score to prioritize some in case of multiple mapping. The network representation is interactive and represented with Cytoscape.js. All the cellular compartments mapped in CellWhere are drawn as compound nodes. The only exception is the cell membrane which is represented in the background as a non-interactive couple of nodes connected by edges, representing the phospholipid bilayer. Clicking on nodes, the user is redirected to the UniProt page of the relative gene, while clicking on the edge, the Mentha interaction evidence is shown. Each drawn network can be downloaded in HTML or XGMML format.

CellNetVis [48] allows for a dynamic exploration of biological networks within a diagram that represents the cellular organization. Networks can be viewed by importing an XGMML file format and nodes must have the "Selected CC" or "Localization" attribute to be mapped to one of twenty-one available cellular compartments. The tool also integrates the possibility to search cellular compartments for human, mouse, and bovine genes with Ensembl, Entrez, InnateDB, or UniProt identifiers. Networks are drawn with the D3 JavaScript library and nodes positions are computed in real-time with a force-directed layout algorithm implemented in that library and modified to constraint nodes to be placed in the respective location. Furthermore, the algorithm can identify and resolve nodes' overlap, however, with huge networks this problem is still present.

After a network is loaded, the algorithm computation tries to find the nodes' positions according to the relative compartment in which they are placed. As Cytoscape.js also D3.js provides an interactive visualization: nodes and compartments (even if they are not considered nodes) can be moved by the user. It is also possible to search for nodes through their labels and when a node is clicked, the attribute panel is filled with useful information relative to the node and with the dropdown menu through which is possible to change its location. If the user changes the location, the algorithm restarts its computation, updating the nodes' positions. Although this feature keeps the positions updated according to the results of the layout, it may result in the opposite effect as it will change eventually the user modified position, leading to a loss of the mental map of the graph. Regarding the graphic draw of the network, the tool allows modifying the color and the label of the nodes. It is also possible to download the image of the graph both in SVG and PNG formats.

CellMap [52] is a tool in which the network is composed over an image of the cell structure. As described in [53], it presents basic interactive functionalities, such as zoom-in or zoom-out, and the user may visualize the location of a query node over that image. Users can search nodes of interest by gene name or UniProt identifier. Additionally, other identifiers can be found and added to the draw as nodes, creating an interaction network among all the inserted elements. Clicking on a node, its interaction edges are highlighted with an interaction score, which represents the confidence of experimentally measured interaction (0 as low confidence and 1 as high

confidence). This use case is useful if the user has few genes or identifiers and is interested in visualizing their locations and interactions.

Furthermore, the tool allows the visualization of all known interaction partners of a protein query. Through the search of the UniProt identifier or gene name of the protein of interest, all the interactor genes, and their respective location can be seen as well as the confidence score associated to the edges. Nodes are colored according to the color compartments.

Some disadvantages of the visualization provided by CellMap lie in the fact that the network is not interactive as the previously described tools. Nodes cannot be moved, and labels are only visible with a "mouseover" event. Thus, if the user is interested to view multiple nodes simultaneously, it may be difficult to do with these features. Furthermore, a network with a high number of nodes produces overlaps between nodes and edges, making it difficult to study the interaction through this visualization.

SPV [54] is a straightforward web tool that provides visualization of signaling pathways and protein interactions networks. The library provides two types of layouts, one for the protein-protein interaction network and one where the nodes are placed in a structure composed of four layers: extracellular, cellular membrane (receptors), nucleus (transcription factors), and a bottom layer describing phenotypes. The visualization is interactive, and the user can move nodes. Clicking on network elements (nodes and edges) a popup will be shown with all the information about the selected element. Nodes are represented as circles and colored differently based on the

type of the represented entity. Similarly, edges have different arrows based on the type of described process: activation, inhibition, binding, etc. Users may also filter edges based on a score, which can be chosen among different options and this is useful to highlight important connections. Lastly, SPV provides the basic functionalities such as the "reset layout", "save image" or "export" of the network.

Pathway Commons [49] integrates various web applications allowing different analyses of pathways and molecular interaction information. This section will describe only the network visualization app. By searching for gene name or pathway title, a list of pathways is displayed with each entry showing the origin database, title, and the number of involved entities. Clicking an entry will open the relative network visualization created through Cytoscape.js, which shows the network in SBGN format. Although networks have compound nodes representing cellular compartments, they are typically inserted side by side and not in a suitable hierarchical context, like the structure of the cell. The network visualization is interactive, and the layout is obtained with the algorithm of CoSE and fCoSE. Clicking on a node will open a tooltip showing detailed information about the selected element. The visualization also has a menu bar with different features: a link to the original database, a biological description of the pathway, and a download button to get the network in different formats, such as PNG, SIF, SBGN, and JSON. Moreover, it is also possible to expand/collapse compound nodes and apply the centering of the network or reset the layout. The user can also search for a node of interest through its label.

The Pathway Commons resource also includes other types of visualization. Indeed, it is also possible to search one or more genes and obtain the visualization of their interaction network or the results of an enrichment analysis that draws enriched pathway titles as nodes. Even in these cases, the visualization is created with the Cytoscape.js library. At the time of this writing, Pathway Commons is at version 12 and includes 5772 pathways and 22 databases for the human species.

The described tools provide different solutions to the challenge of viewing biological networks with the explicit representation of subcellular locations, each one with a specific focus but also with some limitations. One of the most intuitive obstacles for this type of visualization regards how many cellular compartments are necessary and how to draw them in a hierarchical context. Some of those tools chose a fixed number of cellular compartments. By default, CellWhere, CellNetVis, and SPV include 50, 21, and 4 locations respectively. This choice sets a basic limit to the possibility to map many nodes and to the final representation of the cellular structure. CellNetVis also has the same limitation. Its cellular organization is fixed by its diagram that aims to represent the cell structure, but the twenty-one fixed locations are not able to cover a wide range of different mapping. In addition, in this tool, the layout computation starts after the loading of the network, and in this phase, is difficult for the user to interact with the network as the layout is still computing the nodes positions.

SPV was created to be simple and straightforward with a focus on causal relations. It provides a visualization with three compartments: extracellular, cellular membrane,

nucleus, and a bottom layer used to place phenotypes of the signaling networks. This representation is minimal and very specific, and it may be not suitable for covering network visualization with cellular compartments. It was developed starting from Mentha [51], which stores protein interaction networks and continued with SIGNOR [55], which collects causal signaling information.

CellMap represents the cellular structure with a cartoon image of the cell and successively, the networks are drawn over that image. It is also possible to modify the images by adding new shapes and assigning them to new cellular compartments. However, networks are created only starting from genes of interest and, depending on the use case, edges can represent "all the interactions with a specific protein (one-against-all)" or "all the interactions among all the involved proteins (all-against-all)". These use cases are very specific and may not be suitable to create a pathway interaction network by inserting manually all the involved nodes.

In Pathway Commons, compound nodes, which represent the cellular compartment, are obtained by the SBGN format of the pathways. However, they are not drawn to respect the hierarchical cellular organization. This resource also presents the problem of overlapping nodes in large networks, which is one of the major bottlenecks of network visualization. For example, in the "Regulation of PTEN mRNA translation" Reactome pathway, it is possible to see many overlapping nodes. Although the tool provides the possibility to reset the layout, the fCoSE algorithm is not able to solve this problem entirely. Similarly, CellWhere, CellNetVis and SPV didn't solve this problem

entirely. The larger the network, the more it will include overlapping elements. In particular, with the provided example networks of these tools, the resulting layout provides an unclear visualization, with nodes and node' labels overlapping with each other.

itGraph integrates novel solutions to these problems, both for overlapping nodes and representation of cellular compartments, and also integrates new important features for biological networks that are still lacking in other software. The next chapter describes the aim of the project with the chosen focus.

2. Aim of the project

This project aims to create and provide an easily accessible tool to explore and visualize a dynamic and interactive pathway of interest.

Providing a suitable visualization of biological networks is not a trivial task. There are some evaluations that must be considered based on the target users, visualization, and usefulness of the tool. The best draw for any network may not exist, and it is also difficult to understand what a good visualization is. Although the optimization of aesthetic criteria has been introduced to provide appealing results, providing a suitable visualization is still a challenging task. The perception of a specific network design may change among different users, both for aesthetic tastes and for its utility [14]. Consequently, I believe that optimizing certain aspects with a more precise focus on the target of the tool has greater impact and success, providing a useful visualization. In this regard, the aim of the project was to create a web resource for pathway visualization, with a precise focus on technical aspects and increasing the usefulness of the final result. For the latter purpose, different features and annotations have been integrated that are still missing in other tools.

The main objectives of the project are listed as follows:

- increasing biological accuracy by integrating explicit representation of cellular compartments;
- reduction of the visual complexity of the network through the Power Graph Analysis;
- integration of biological annotation and features to enhance the visualization;
- optimization of the user experience by creating a user-friendly tool.

In particular, these aspects have been optimized based on specific choices and integration of features that should help the user with network analysis. Most of the covered issues are technical which can help the user experience.

Some other features have not been considered, for example, the possibility to download the network in different formats, such as BioPAX, SBGN, or GML. However, the tool provides the download of the network in JSON, which is a very versatile format and easy to parse. Moreover, even if a lot of pathways have been integrated, in the actual release it is not possible to import a custom network.

The following chapter "Materials and Methods" will describe in detail how the networks were obtained and how the layouts were pre-computed. It will also describe all the technical aspects regarding the building of the tool.

Next, with the "Results and Discussion", results and their usefulness for visualization and analysis will be discussed. Finally, the "Conclusion" and "Future perspectives" will explain the major outcomes obtained with the tool and some ideas for future releases and improvements.

3. Materials and Methods

Retrieve pathways from graphite

The first step to build networks was to collect pathways from the R package graphite [56], which allow the conversion of pathway topology to gene/protein networks in simple interaction format, SIF (Fig. 3.1).

					Source		Target			
	species	database	nativeId	title	src_type	src	dest_type	dest	direction	type
0	athaliana	kegg	ath:00020	Citrate cycle (TCA cycle)	TAIR	AT1G01090	TAIR	AT1G34430	directed	Process(indirect)
1	athaliana	kegg	ath:00020	Citrate cycle (TCA cycle)	TAIR	AT1G01090	TAIR	AT1G54220	directed	Process(indirect)
2	athaliana	kegg	ath:00020	Citrate cycle (TCA cycle)	TAIR	AT1G01090	TAIR	AT3G13930	directed	Process(indirect)
3	athaliana	kegg	ath:00020	Citrate cycle (TCA cycle)	TAIR	AT1G01090	TAIR	AT3G25860	directed	Process(indirect)
4	athaliana	kegg	ath:00020	Citrate cycle (TCA cycle)	TAIR	AT1G01090	TAIR	AT3G52200	directed	Process(indirect)

Fig. 3.1: Example of the SIF (Simple Interaction Format) file. Each row describes an edge of the network, with the columns "direction" and "type" providing information about the direction and the biological process involved, respectively.
The source node of an edge is identified by the columns "src_type" and "src", describing the type of the identifier and the ID value, respectively. Similarly, the target node is represented by the columns "dest_type" and "dest". The values of "species", "database", "nativeId", "title" reports information about the pathway, thus they are the same for each row.

Pathways are usually characterized by the presence of metabolites and compounds. This is important as, in some interactions, compounds act as mediators or bridges between two elements. However, measuring gene expression and metabolite

concentrations require different experimental techniques. For this reason, the two signals are infrequently captured together on the same samples. Graphite adopts a signal propagation strategy able to reconstruct the pathway network with only a subset of the original entities. This strategy allows the users to get pathway variants including only proteins, metabolites, or preserving both types of elements ("mixed" pathways).

Our strategy was to avoid huge networks for the previously mentioned problems, so I collected all protein pathways (and all the relative information) from graphite respecting the following threshold: the sum of the number of nodes and the number of edges of each pathway must be equal to or lower than 2000:

$$|V| + |E| \ 2000$$

I chose to ignore huge networks as they still are a bottleneck of network visualization and even with this constraint, I retrieved more than 98% of the total pathways provided by graphite.

A total of 173817 pathways were obtained, distributed among different databases, such as Kegg, Reactome, Pathbank, Pharmgkb, Smpdb, and covering the following 14 species: Arabidopsis thaliana, Bos taurus, Caenorhabditis elegans, Canis lupus familiaris, Drosophila melanogaster, Danio rerio, Escherichia coli, Gallus gallus, Homo sapiens, Mus musculus, Rattus norvegicus, Saccharomyces cerevisiae, Sus scrofa and Xenopus laevis.

Among all the collected pathways, occurring nodes presented different types of identifiers, such as TAIR (The Arabidopsis Information Resource) [57], UniProt [58], EntrezID [59], FlybaseCG [60], Alias, Ensembl [61], ORF [62], each one with a different number of occurrences and percentage (Fig 3.2).

Type	Occurrences	Percentage
ALIAS	1259	1.082331
ENSEMBL	97	0.083388
ENTREZID	47402	40.750325
FLYBASECG	1833	1.575785
ORF	1218	1.047084
TAIR	2335	2.007342
UNIPROT	62179	53.453745

Fig. 3.2: Occurrences and percentage of nodes identifiers among all obtained pathways. Total nodes: 116323.

Conversion of identifiers

One of the new features integrated into itGraph, which is still lacking in other software, is the possibility for the user to require the conversion of node identifiers. Both biology and bioinformatics are highly characterized by crossing references between databases. This is mainly evident among their identifiers, where each provides specific annotation for a specific type of biological molecules, such as genes, transcripts, and proteins [63]. Mapping these databases' IDs is essential to facilitate the exchange of each specific annotation. For example, ids of gene entities can be described by the Ensembl Gene

(Ensembl) and EntrezID (NCBI), while protein entities can be identified with Ensembl Protein (Ensembl) and UniProt (UniProt) ids. It is possible also to map a gene identifier to a protein type, obtaining multiple results, as reflection of a single gene that can lead to the production of different proteins. Thus, it is essential for a biological tool to provide a mapping between all of them.

itGraph is strictly linked to "graphite", providing the same type of conversions. All the possible conversions are pre-computed and obtained through the following Bioconductor R packages:

- Arabidopsis thaliana: ''`org.At.tair.db`''
- Bos taurus: ''`org.Bt.eg.db`''
- Caenorhabditis elegans: ''`org.Ce.eg.db`''
- Canis lupus familiaris: ''`org.Cf.eg.db`''
- Drosophila melanogaster: ''`org.Dm.eg.db`''
- Danio rerio: ''`org.Dr.eg.db`''
- Escherichia coli: ''`org.EcK12.eg.db`''
- Gallus gallus: ''`org.Gg.eg.db`''
- Homo sapiens: ''`org.Hs.eg.db`''
- Mus musculus: ''`org.Mm.eg.db`''
- Rattus norvegicus: ''`org.Rn.eg.db`''
- Saccharomyces cerevisiae: ''`org.Sc.sgd.db`''
- Sus scrofa: ''`org.Ss.eg.db`''

- Xenopus laevis: ‘‘`org.Xl.eg.db`’’

Not all of these databases provided the mapping of the same types, however, in general, the tool provides the conversion of all the collected pathway nodes to the following identifiers (where applicable): Alias, Gene Name, Ensembl Gene, Ensembl Prot, Ensembl Trans, EntrezID, Enzyme, Flybase, FlybaseCG, Orf, Pfam, Refseq, Sgd, Tair, UniProt, Wormbase, and Zfin.

Retrieving subcellular location for pathway nodes

As Fig. 3.1 shows, more than 94% of total nodes belong to UniProt or NCBI (EntrezID) databases. One of the aims of the tool is to increase the biological accuracy of the network by integrating the subcellular location for each node. For this purpose, from UniProt and NCBI, I collected all the Gene Ontology Cellular Component (hereafter CC) associated with their ids. The CC is one of three categories of the controlled vocabulary of Gene Ontology and provides the locations relative to cellular structures in which a gene product performs its function. Specifically, I parsed the “*dat*” file format of SwissProt and Trembl, and for each of their nodes, I saved the entries of its CC of the Gene Ontology and the provenance of the file, whether SwissProt or Trembl. Similarly, the same information was obtained from the NCBI for their EntrezID ids. Each collected CC entry is associated with an evidence code that indicates how the annotation to a particular term is supported. Evidence codes belong to up to six

general categories: experimental evidence, phylogenetic evidence, computational evidence, author statements, curatorial statements, and automatically generated annotations. Fig. 3.3 shows an example table of the format of the gathered information.

CC retrieved from SwissProt and Trembl

ID	Prov	GO:CC	Evi. Code
U1	swissprot	cell membrane	IBA
U1	swissprot	cell membrane	HDA
U1	swissprot	cytoplasm	HDA
U3			
...	...	...	...
U20	trembl	cytosol	ISS
...	...	...	...
U30	swissprot	cytoplasm	HDA
U31	trembl	nucleus	IDA
...	...	...	...
U40	swissprot	mitochondrion	ISO
U41	trembl	cytosol	IEA
...	...	...	...

CC retrieved from NCBI

ID	Prov	GO:CC	Evi. Code
E1	ncbi	cell membrane	ISS
E2	ncbi	cell membrane	ISO
E2	ncbi	nucleus	HDA
E3			
...	...	...	...
E20	ncbi	membrane	ISO
...	...	...	...
E30	ncbi	chloroplast	ISA
E31	ncbi	mitochondrion	ISM
...	...	...	...
E40	ncbi	cytosol	IDA
...	...	...	...
...	...	...	...

Fig. 3.3: Example tables with example UniProt and NCBI ids with the information of the Cellular Component.

Successively, to retrieve the location for all pathway nodes obtained from graphite, including those that weren't obtained with UniProt or EntrezID identifier type, all of them were converted to the UniProt or EntrezID types (NCBI). The idea is to map all the conversion results to their relative information of CC gathered. In particular, the following steps were performed:

1) all the native pathway identifiers were first converted to UniProt. It was expected that in some cases, different original nodes were mapped to multiple

UniProt IDs each. For example, an EntrezID, which describes gene-specific information, can be mapped to multiple UniProt as they represent ids for protein type. Some identifiers were not converted successfully (Fig 3.4) and in this case, as described in step 5) they will be converted to the EntrezID and then mapped to the CC entries retrieved from the NCBI database.

Pathway nodes

species	type	node
athaliana	TAIR	T1
athaliana	TAIR	T2
…	…	…
ggallus	ENSEMBL	ENS1
…	…	…
hsapiens	UNIPROT	U1
hsapiens	UNIPROT	U2
hsapiens	ENTREZ	E1
…	…	…
mmusculus	ENTREZ	E2
…	…	…

Conversions

type	node	conv_type	conv_node
TAIR	T1	UNIPROT	U20
TAIR	T2		
…	…	…	…
ENSEMBL	ENS1		
…	…	…	…
UNIPROT	U1	UNIPROT	U1
UNIPROT	U2	UNIPROT	U2
ENTREZ	E1	UNIPROT	U30
ENTREZ	E1	UNIPROT	U31
…	…	…	…
ENTREZ	E2		
…	…	…	…

Fig. 3.4: Descriptive example of native pathway nodes converted to UniProt identifiers. Green highlights report the native node "Entrez E1" that is mapped to multiple UniProt. It is also visible that some nodes were not mapped to any conversion, for example, the "Tair T1" and the "Ensembl ENS1".

2) all the UniProt IDs gathered with the parsing of the SwissProt and Trembl files, also associated with the CC information and provenance, were mapped to the resulting conversion (Fig. 3.5);

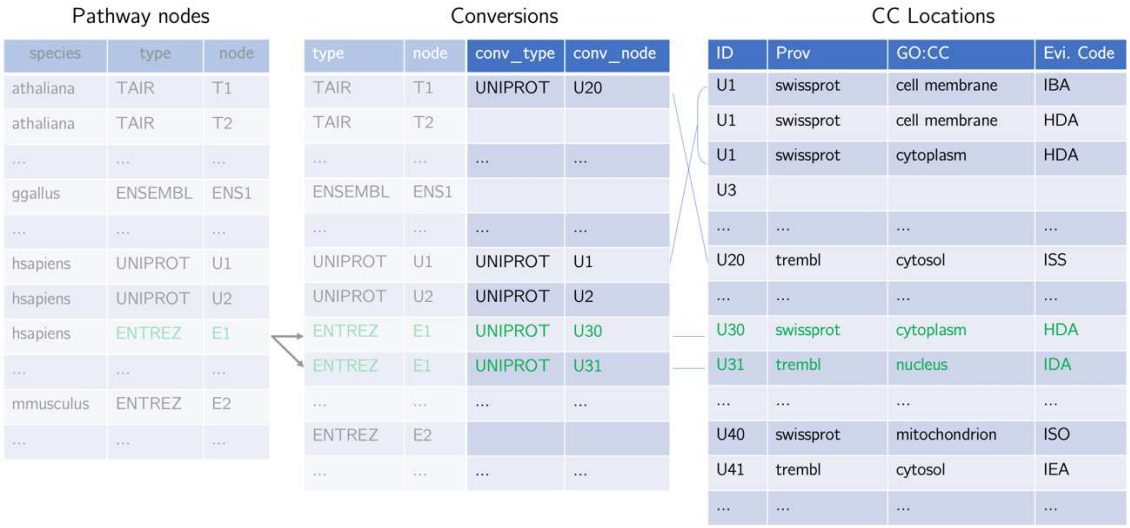


Fig. 3.5: Cellular Component retrieved from UniProt were mapped to the resulting conversion.

3) To maintain only the reliable mapping, CC entries were filtered based on the evidence code **category**. As Buza *et al.* performed in their work [64], I also created a rank on the categories of evidence codes (Fig. 3.6). Thus, if an UniProt ID was mapped to multiple CC entries with different evidence codes of a different category, I selected only the entries with the evidence code belonging to the better available category. It is important to note that, even after the filter, some conversions may be still mapped to multiple CC entries.

Method	Evi. Code	Inferred from	Score
Experimental	IDA	Direct assay	5
	HDA	High Throughput Direct assay	
	IMP	Mutant phenotype	
	IPI	Physical interaction	
	IGI	Genetic interaction	
	EXP	Experiment	
Author statement	TAS	Traceble author statement	4
	NAS	Non-Traceble author statement	
	IC	Curator	
Phylogenetically	IBA	Biological aspect of ancestor	3
Computational	ISS	Sequence or structural similarity	2
	ISA	Sequence alignment	
	ISM	Sequence model	
	ISO	Sequence orthology	
	RCA	Reviewed computational analysis	
	IGC	Genomic context	
Electronic	IEA	Electronic annotation	1

Fig 3.6: Table reporting the score associated to the different categories of evidence codes.

4) To prioritize the most reliable results, after the filter phase and only for those entries mapped with SwissProt provenance, all the associated CC locations were attached to the native pathway nodes. This choice was made to first obtain the best results, as SwissProt is the curated part of the UniProt database. In the case of multiple mapped locations, the native pathway node will inherit them all and the final decision will be performed during the building of the network: the most frequent compartment within the network is chosen (drawing criteria). The Trembl CC entries will only be used for those original nodes which have not been successfully mapped to CC locations by SwissProt or NCBI.

5) with the previous steps, for some of the native pathway nodes, at least a subcellular location from SwissProt was retrieved. All nodes that were not mapped to the UniProt conversion or not mapped to the CC entries did result with no location associated (Fig. 3.7). For those nodes without an associated location, I performed the same procedure (steps 1 - 4), converting them to the EntrezID and then mapping the CC entries retrieved from the NCBI database.

Pathway nodes

species	type	node
athaliana	TAIR	T1
athaliana	TAIR	T2
…	…	…
ggallus	ENSEMBL	ENS1
…	…	…
hsapiens	UNIPROT	U1
hsapiens	UNIPROT	U2
hsapiens	ENTREZ	E1
…	…	…
mmusculus	ENTREZ	E2
…	…	…

Conversions

type	node	conv_type	conv_node
TAIR	T1	UNIPROT	U20
TAIR	T2		
…	…	…	…
ENSEMBL	ENS1		
…	…	…	…
UNIPROT	U1	UNIPROT	U1
UNIPROT	U2	UNIPROT	U2
ENTREZ	E1	UNIPROT	U30
ENTREZ	E1	UNIPROT	U31
…	…	…	…
ENTREZ	E2		
…	…	…	…

Fig. 3.7: Conversion of native pathway nodes to UniProt. All the nodes that were not mapped to the UniProt conversion or not mapped to the CC entries are highlighted in red.

6) For all the nodes that didn't get at least a location with the SwissProt or NCBI CC, they have been mapped to the CCs obtained from Trembl (as described in step 4). As it provides automated annotations with less reliability than

SwissProt and NCBI, they have been used as the last chance to obtain at least a subcellular location.

After all these steps, all the CC locations mapped to the pathways nodes were converted to the respective labels of the UniProt SubCellular Location section of the database (https://www.uniprot.org/help/subcellular_location). As will be discussed later, this last conversion allows a coherent mapping between the retrieved locations of the pathway nodes and the labels obtained from the cell designs of the SwissBioPics. This procedure allows getting at least one location for the majority of pathway nodes. Specifically, I got locations for 78355 (67,3%) pathway nodes, while entities without this information were 37968 (32,7%) (Fig. 3.8).

species	type	node	GO:CC	prov
athaliana	TAIR	T1	cytosol	trembl
athaliana	TAIR	T2	chloroplast	ncbi
…	…	…	…	…
ggallus	ENSEMBL	ENS1		
…	…	…	…	…
hsapiens	UNIPROT	U1	cell membrane | cytoplasm	swissprot | swissprot
hsapiens	UNIPROT	U2	cytosol	ncbi
hsapiens	ENTREZ	E1	cytoplasm	swissprot
…	…	…	…	…
mmusculus	ENTREZ	E2	nucleus	
…	…	…	…	…

Fig. 3.8: Example table of pathway nodes with the associated single or multiple location. The provenance is shown in the "prov" column.

Defining the hierarchical cell structure

To reproduce a network visualization with cell compartments, one of the most challenging tasks was to reconstruct the hierarchical cell structure and organization with all its organelles. This task was even more complicated considering that the tool provides this type of network visualization for pathways of different species. To solve this problem, I built a hierarchical cell structure parsing and transforming the cell description available from SwissBioPics [65]. SwissBioPics is a freely available resource that provides images describing cell types from all kingdoms of life. Each cell design presents a list of the subcellular locations or organelles that compose its spatial organization. As itGraph covers pathways for bacteria (Escherichia coli), yeast (Saccharomyces cerevisiae), plant (Arabidopsis thaliana), and different animal cells, from that resource I collected the description design of different cell types and organisms: animal, animal epithelial, animal muscle, animal neuronal, animal photoreceptor, animal egg, animal spermatozoa, budding yeast, plant, rod-shaped bacteria two membranes gram neg, and virus-infected animal.

For each cell design provided by SwissBioPics and starting from the list of subcellular locations attached, I processed the list to provide a hierarchical order of that compartments. The resulting ordering was obtained by visual inspection of the diagram and cross-referencing the information provided by UniProt (https://www.uniprot.org/locations). Each entry in each resulting list has a specific

indentation to define its visual parent node. Fig. 3.9 shows the spatial organization of a morphology typical of Escherichia coli cells. The right side of the image shows the visual hierarchical description of the involved compartments. For example, the "Cytoplasm" has a greater indentation than the "Cell inner membrane" as it is placed visually inside of it.

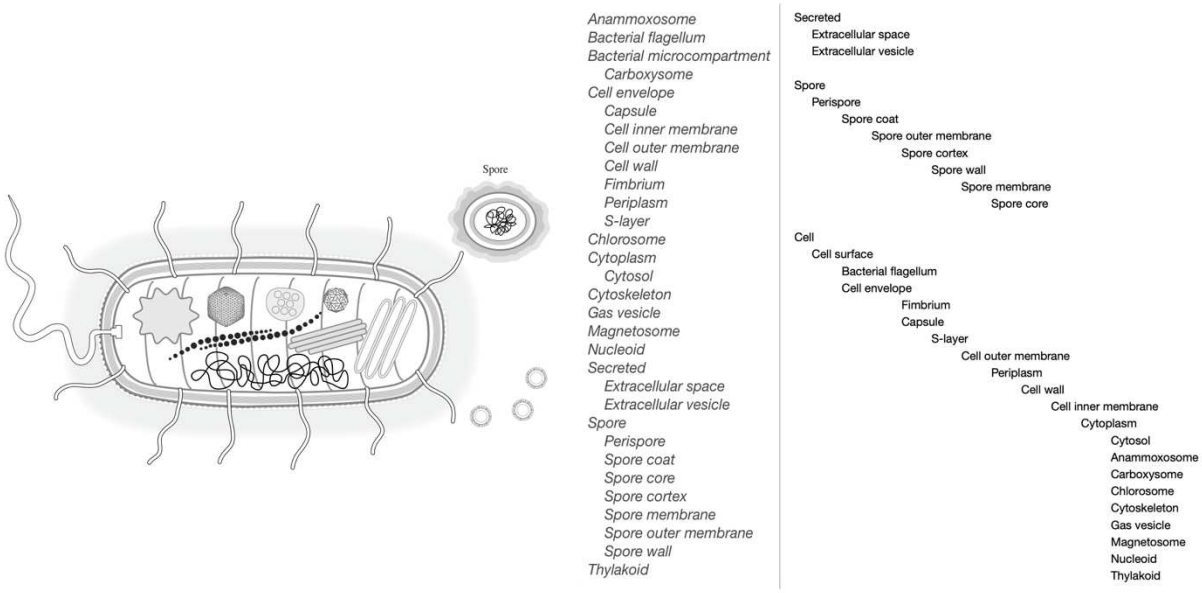


Fig. 3.9: SwissBioPics design of morphology typical of Escherichia coli with relative compartments. On the right side, there is the reconstruction of the visual.

From the resulting hierarchical list, I build a tree data structure able to explain the same **visual nesting relations** between parent-child nodes (Fig 3.10).

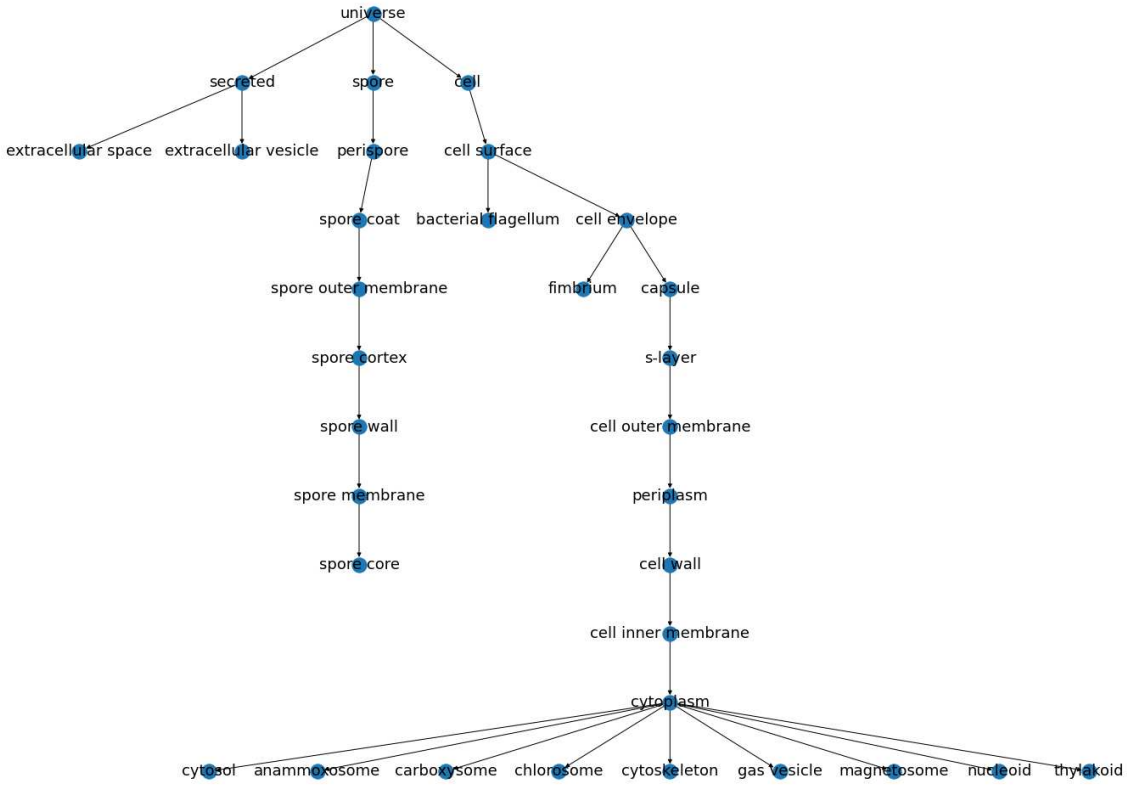


Fig. 3.10: Tree structure of the reconstructing hierarchical organization of the "rod-shaped bacteria two membranes gram neg." cell design.

The tree data structure is useful as it allows the use of graph algorithms to evaluate which compartment is the parent of a certain node. Indeed, for the explicit representation of subcellular location within the network, this object will be also useful as it allows the creation of compound node for each node location of the network and places it inside the right parent. As Fig. 3.10 shows, the root of the tree is the "*universe*" node, which was added manually and represents the graphic display where the network is drawn. The root is also considered a compound node in which all the nodes

with no location will be mapped or with a location that is not mapped to the chosen cell design.

Minimal tree reduction (MTR)

The collection of the cell design from SwissBioPics allows the creation of a tree structure able to describe the parent relationship of the node locations involved in a network. However, most of the locations involved in node networks do not cover all the subcellular locations of the tree structures, thus drawing a network with compound nodes of locations that don't have any occurrences, leads to increasing the complexity of the visualization. I solve this problem by creating an algorithm called Minimal Tree Reduction (hereafter MTR) able to reduce a tree structure to a set of minimal vertices. It is important to say that the following algorithm was created without any optimization as its execution is not performed in real-time during the visualization. Furthermore, it is applied only to the obtained tree structure of the cells (all the tree structures have a number of nodes lower than 100), which can be considered as small instances. Indeed, all the procedures described in this chapter, with some exceptions, are performed "*offline*".

This algorithm aims to reduce a tree to its minimal vertices, based on a set of vertices (locations) given in the input. Let's take as an example the tree structure of the Plant Cell, and let's say that the occurrence locations between all nodes in a network are the

following: *Cell membrane*, *Cytoplasm*, *Rough endoplasmic reticulum*, *Smooth endoplasmic reticulum*, *Nucleus*, *Nucleus lamina*, and *Nucleoplasm*. Thus, this set of locations will be the set of vertices to give in input.

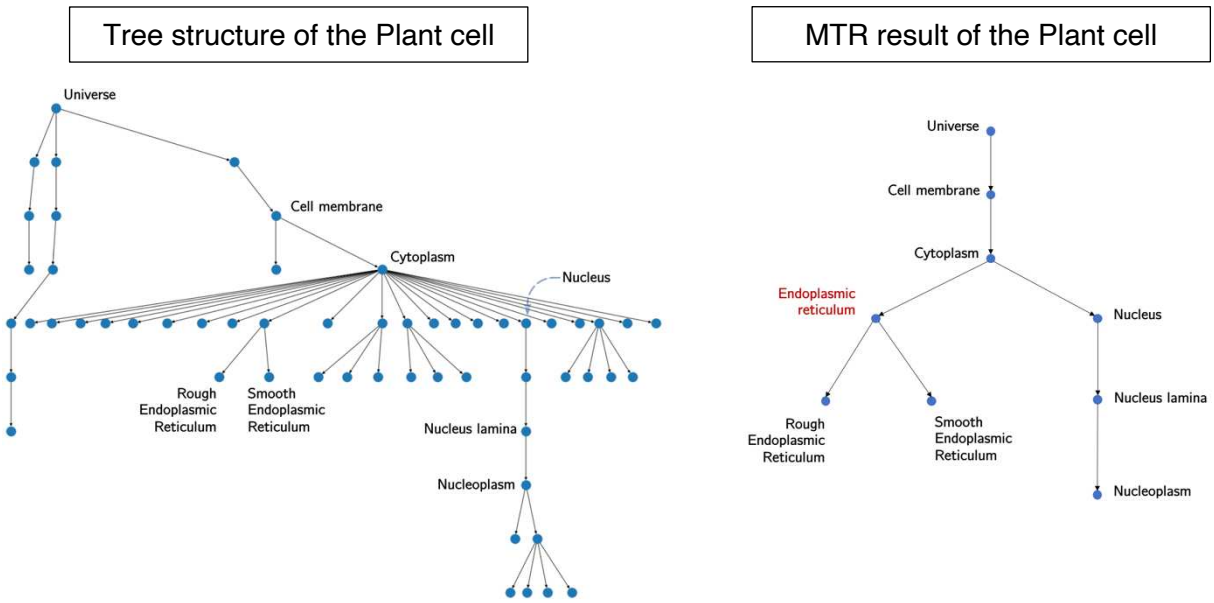


Fig. 3.11: On the left side is represented the tree structure of the Plant cell design. Labels are useful to highlight the example of the occurred locations in a network. On the right side, there is the new tree resulting from the MTR algorithm, based on the location provided by the user (shown with the labels). "Endoplasmic reticulum" is highlighted in red as it wasn't provided by the user but added by the algorithm because it is the first ancestor node among the "Rough endoplasmic reticulum" and the "Smooth endoplasmic reticulum".

The general idea of the algorithm is to remove unused vertices of the tree. However, the simple removal of nodes will break all the connections of the tree leading to a loss of the parent-node relationship of the nodes. As Fig. 3.11 shows, the resulting tree from the MTR algorithm inherits a reduced visual relation of the nesting organization of the organelles involved. Furthermore, highlighted in red there is the node labeled "Endoplasmic reticulum", which was not given in the input but was added by the algorithm as it is the first common ancestor between two nodes given in the input. This

algorithm design was made on purpose because it is important to provide a common ancestor between two input nodes to preserve as much as possible the visual relations among all nodes and their parent. For example, with the following nodes "*Endoplasmic Reticulum*" and "*Nucleus*" as the only occurrence locations of the network, this design allows to provide a network with the common ancestor "*Cytoplasm*" instead of having two parallel compound nodes linked to "*universe*".

The algorithm runs recursively with 4 phases, and in each iteration, the most convenient path is removed:

1) in the first step, a set of nodes that cannot be removed are collected, called "*saved nodes*". This set is composed of the root, all the hub nodes (nodes with several successors are considered as a hub), and the nodes' locations given in input;
2) in the second step, the shortest path among all pair nodes is computed. At the end of the iteration one of these paths will be contracted and the two nodes, source and target, will be linked to each other;
3) the collection of all shortest paths is sorted longest first, and each of these shortest paths is provided to a function that evaluates if it can be removed. Providing the longest first is useful because, for example, a removable shortest path composed of 6 edges prevents doing 6 further iterations;
4) each shortest path is evaluated for its contraction: if it has nodes (except for source and target) in common with the saved nodes collected in step 1, the path cannot be reduced. If there is no intersection with the saved nodes, the path is

contracted, and the source and the target are linked to each other. This step evaluation is repeated until a path is contracted, after which the first step is restarted.

It is important to note that in the first step, all the hub nodes are saved even if they are not included in the occurrence location of the network. Despite that, in different iterations, this recursive strategy can remove the hubs from the "saved" nodes. The solution to this problem is shown in Fig. 3.12. In that example, both the "Endoplasmic reticulum" and its successors are not the occurring locations of the network, nevertheless, for the first iteration, the "Endoplasmic reticulum" node is added to the saved nodes as it is a hub. During the iterations, all the paths that link the "Endoplasmic reticulum" to its successors are removable edges. Thus, contracting one of the two edges leads to not considering that node as a hub for the next iterations. This recursive strategy allows for the removal also of unnecessary hubs from the final result of the MTR algorithm.

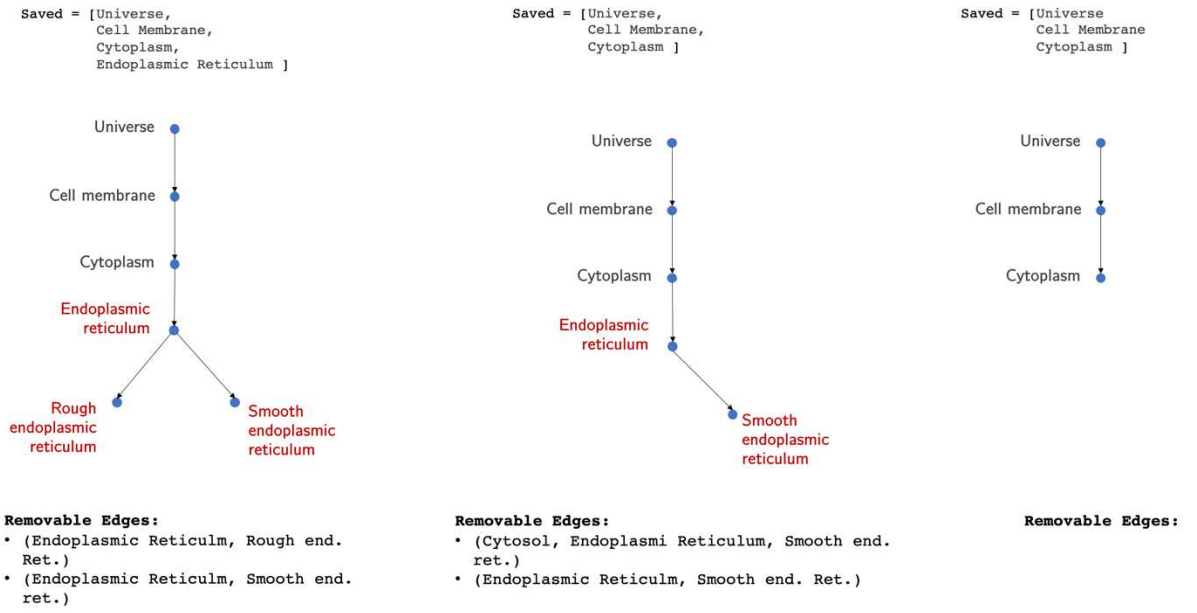


Fig. 3.12: An example illustration of three iterations of the MTR algorithm that remove hubs from the resulting tree. In this example nodes that are not occurrence location of the network are highlighted in red. In the first iteration (left image), even if "Endoplasmic reticulum" is not an occurrence location of the network, it belongs to the saved nodes as hubs. The two edges that connect endoplasmic reticulum to its successors are labeled as removable edges. Thus, one of the two edge is contracted. In the second iteration (central image), after the contraction of the edge, the "Endoplasmic reticulum" is no more considered as an hub, thus the edges that link the "Cytoplasm" to "Smooth endoplasmic reticulum" can be contracted producing the final result (right image).

All the steps described are performed until no more paths can be contracted. This strategy reduces a tree to a set of minimal vertices and consequently allows for the creation of an optimized visualization with a minimal number of cellular compartments.

Available types of networks

itGraph is a web tool developed to optimize technical aspects, even those considered secondary, which affect the responsiveness and use of the visualization. Furthermore,

two of the goals of itGraph were to increase the biological accuracy of the networks and to provide a visualization with three different perspectives:

- *simple network*: this is similar to the traditional network drawing, as nodes are simply encoded as points in the space but colored according to the compartment location. This type of visualization produces an appealing result and it is useful mostly for small-medium graphs;
- *network with compartments*: the main goal of this drawing is to provide a visualization with the explicit representation of the hierarchical organization of cell compartments. This perspective can be useful as it shows the nodes of the pathway within the respective location (drawn as compound nodes) and can help analyze molecular processes that span different subcellular locations;
- *power graph*: this perspective aims to reduce the visual complexity of the network. Specifically, this visualization is the result of the power graph analysis [66], which is a lossless conversion of biological networks into a compact and less redundant representation. This method can reduce network complexity by explicitly representing re-occurring graph motifs, such as Star, Clique, and Biclique, that are widely represented in a biological network.

Before computing the layouts, network objects were built as follows:

1) for each network in the SIF format obtained from graphite, a compression function was applied to reduce the drawing redundancy of the edges, as shown

in Fig. 3.13. Specifically, for each pair of nodes, all the edges between them were compacted to a single edge with a general direction derived from all the specific ones of the original edges, and all the information inherited from them was compacted as well. Here is described an example of this compacted process, where two nodes u and v are linked with the following edges (with the respective direction and annotation):

Source	**Target**	**Direction**	**Biological Process**
u	v	$directed$	$BP1$
v	u	$undirected$	$BP2$
v	u	$directed$	$BP1$

the derived single edge is created as (u, v) with the *'undirected'* direction considered as the most general one, and all the original information about the biological process was saved as well, together with the specific direction in which each took place.

The resulting single edge is defined as follows:

Source	**Target**	**Direction**	**Biological Process**
u	v	$undirected$	• $Process: BP1; Direction: [forward, backward]$ • $Process: BP2;\ Direction: [undirected]$

As described by the example, all the information regarding the direction of each biological process was saved and adapted to be coherent with the new arrangement of the source and target order.

This strategy allows to store networks into compact structures and removes the drawing redundancy while saving all the original information in one single edge (Fig. 3.13).

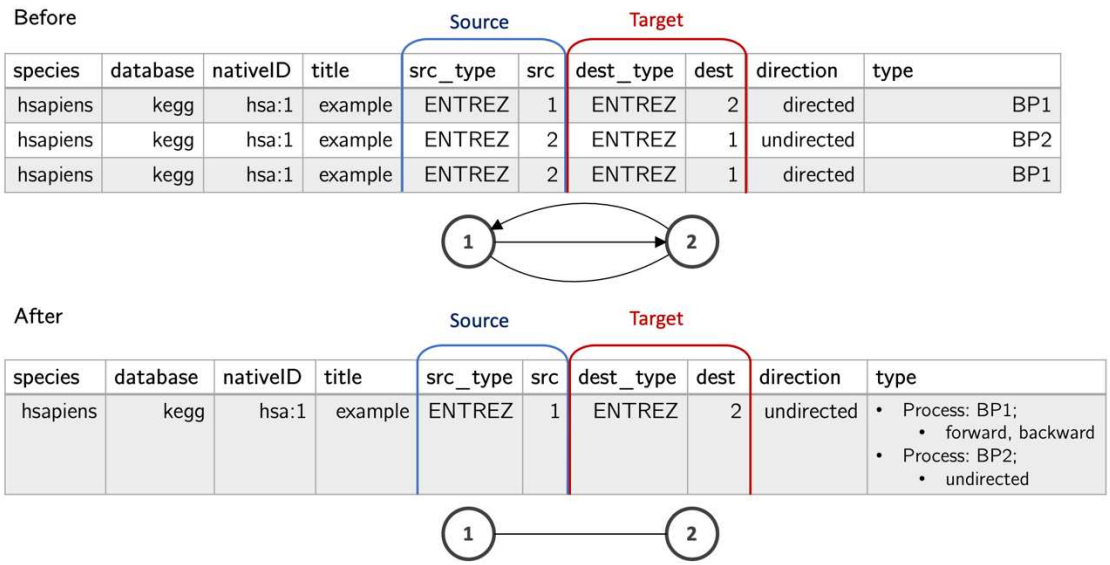


Fig 3.13: A graphical example of the removal drawing redundancy of the edges. From the original interactions described in the SIF file, a single edge is derived with the most general direction, saving all the original information about the biological processes and the specific direction in which each takes place.

2) as explained in the paragraph "Retrieving subcellular location for pathway nodes", most of the pathway nodes were annotated with a subcellular location. In this step, for each pathway, a single-entry location was chosen for each node. In the case of nodes mapped to multiple locations, a single compartment was

chosen using a graphic criterion. Specifically, among the possible locations for that node, the most frequent compartment among the other nodes was chosen;

3) in this step, the first type of network, "simple network", was built for each pathway. Each node object in Cytoscape.js was filled with different information: id, label (the value of the identifier of the node), node type (the type of the identifier, e.g. UniProt, Gene Name, EntrezID, …), location, provenance (source of location, e.g. SwissProt, NCBI, Trembl) and other technical information as width or height of the node. By default, all the nodes have the same size, thus the same width and height, except for compound nodes;
4) based on the occurring node's location of the current pathway, it is necessary to choose the right tree cell structure. All cell designs obtained by SwissBioPics cover all 14 pathway species. However, three of these 14 species have only a single tree hierarchical structure that can cover node locations. Specifically, *Arabidopsis thaliana*, *Escherichia coli*, and *Saccharomyces cerevisiae* have only one possible cell hierarchy: Plant, Rod-shaped two membrane gram-negative, and Budding yeast, respectively. All other species have multiple hierarchical cell designs to which node locations can be linked. To create a comprehensive design for these species, all animal cell designs were merged providing a single hierarchical tree structure. So, in this step, based on the species all the pathway nodes locations were mapped to a single specific hierarchical cell design;

5) in this step the MTR algorithm was applied finding essential compartments to describe a minimal hierarchical cell organization of all the organelles involved in the pathway;
6) merging the information of the simple network previously created, and the minimal tree of the cell compartments, a new object for the visualization of "Network with compartments" was built. In particular, for each compartment in the reduced tree, a compound node was added to the simple network. Furthermore, each original node of the simple network has been associated with the 'parent' information, which indicates the ID of the compound node to which it belongs. If the location of a node is present in the reduced tree, then the id of its compartment will be added to its parent parameter. Similarly, by evaluating the predecessor of each node in the reduced tree, each added compartment was associated with the parent parameter as well, meaning the id of the predecessor compartment. In this way, each compartment was added to the Cytoscape.js network object keeping the minimal hierarchical organization of the cell obtained with the MTR algorithm;
7) starting from the simple network, the graph object of the applied power graph was obtained. The utility of this type of visualization lies in its potential to reduce the visual complexity of the graph drawing. As already mentioned, huge networks (with thousands of nodes and edges) are one of the bottlenecks of visualization, as they often result in "hairballs" from which it is difficult to

extract information. A large number of nodes and edges with a lot of edges crossing increases the complexity of the gathering of information process. There are also a lot of technical problems associated with huge networks, but a possible solution to display such networks is to reduce their complexity. The Power Graph is a lossless representation of networks, which reduces their complexity by explicit representing recurring network motifs Star (a single node connected to many other nodes), Clique (a complete graph, where all the nodes are connected to each other), and Biclique (also called a complete bipartite graph, where all nodes in one group interact with all nodes in another group) [66]. These motifs are widely represented in biological networks: for example, the Star can represent a hub protein or the Clique can describe a protein complex Fig. 3.14.

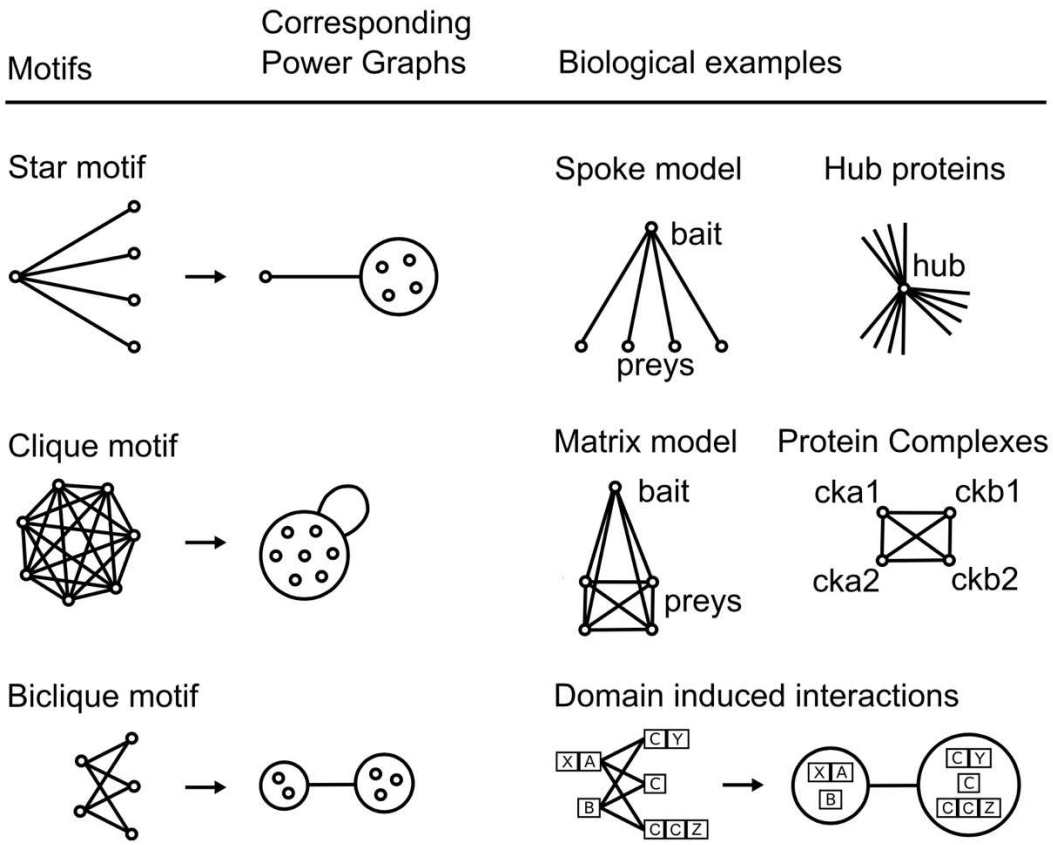


Fig. 3.14: Representation of the network motifs Star, Clique, and Biclique, with their respective conversions in the compressed structure of the power graph. On the right side, there are examples of some possible biological meanings of the three motifs. Image adapted from Royer et al., 2008.

The resulting encoding is composed of two basic elements: a power node which is a set of the original nodes of the compressed network and power edges which are connections between power nodes.

The power graph algorithm acts in two phases: first, there is the identification of potential power nodes with a hierarchical clustering based on neighborhood similarity. In the second phase, power edges are searched between nodes and the collected potential power nodes. As Fig. 3.15 shows, original edges are added if no power edge abstracts them. The result is a new encoding of the

original network with reduced complexity and obtained without any loss of information.

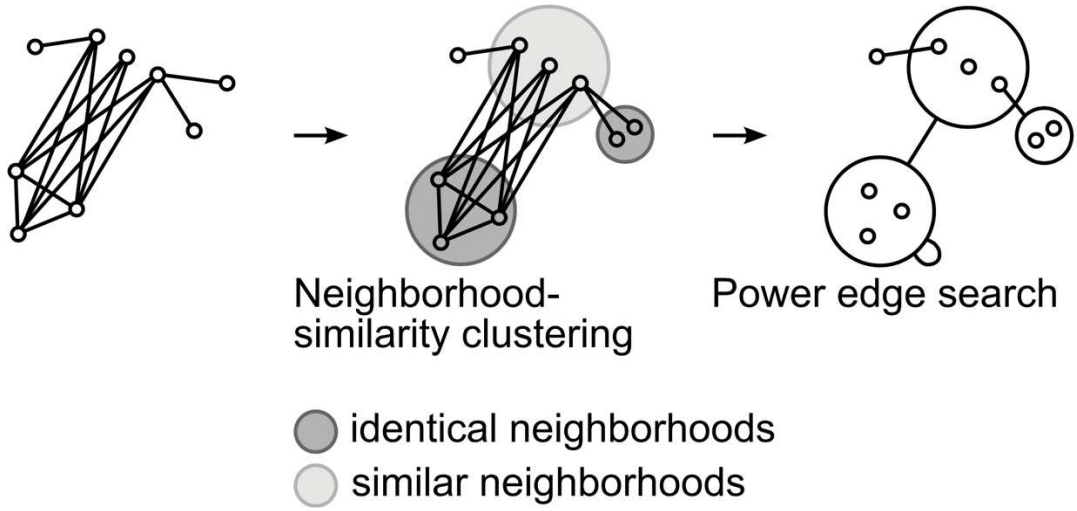


Fig. 3.15: An illustration of the two phases of the power graph algorithm. Image adapted from Royer et al., 2008.

For each pathway network, I applied the power graph algorithm. The output of the method provides a text file describing all the original nodes given in the input, a set of power nodes, a text description of the nested organization of the identified power nodes to each other, and the identified power edges. By parsing the output, the Cytoscape.js object of the power graph was created. In particular, after adding all the nodes of the pathway, each power node was successively added to the object as a compound node. Through the textual description of their nested organization, the parent relation of each original node and each power node were inferred and added to them.

Computing networks layouts

After creating the three Cytoscape.js objects, the node positions of each network were obtained by computing its layouts. One of the key aspects of itGraph optimization lies in the fact that all the networks provided by the tool have an already computed layout for each visualization type. Indeed, the tool is developed to minimize the user waiting time before being able to interact with the network. By pre-computing nodes' positions of each network visualization, the user has to wait only for the graphical rendering time of each node and edge.

Although there are many layout algorithms, in graphics libraries there are not many implementations of those algorithms able to compute layouts for traditional graphs and networks with compounds. Two implementations able to adapt to this type of request are the fCoSE and Cola algorithms, both implemented in the Cytoscape.js graphics library.

To pre-calculate the nodes' positions of the networks, I used the SyBLaRS repository (SYstems Biology LAyout & Rendering Service) (https://github.com/iVis-at-Bilkent/syblars/), which is a web service to lay out graphs in different formats, like SBGNML, SBML, GraphML, and JSON, and/or produce corresponding images. This service can also run locally, and it was developed to support many Cytoscape.js layout

algorithms implementations, including fCoSE and Cola. It can provide the image of the resulting layout and/or the computed position for each node in a JSON file format. By giving in input all the created network objects to a local instance of SyBLaRS, I was able to compute nodes' positions using sequentially fCoSE and Cola algorithms. In particular, fCoSE has a faster computation and also provides results with fewer edge crossing and node-edge overlap than Cola. However, the latter has better results in avoiding overlapping nodes. In the first computation, I used fCoSE to position nodes, obtaining what could be considered as a good baseline solution both for its computation time and the optimization of layout metrics. Successively, Cola has been used starting from the position found by fCoSE. With this strategy, Cola is able to solve and avoid overlapping nodes without drastically changing the layout, as those initial positions of the nodes were already close to the convergence threshold of the Cola algorithm and thus a good approximated solution.

All the computed network layouts, including the relative biological information attached, were saved in the database of itGraph.

Inheriting positions of the converted identifiers

itGraph also integrates the conversion of identifiers for all the pathway nodes. The user can map the entire network or a single node into a new type of identifier. In this latter aspect, the user can also use one of the conversion result ids as the display label

of the node. The results of each conversion are pre-computed as already explained in the section "Conversion of identifiers".

In a visualization of biological networks, this feature is strictly linked to a strategy to solve the position of the conversion results. As mentioned above, the mapping of a node can provide multiple results, but it is necessary to apply a strategy for the inheritance of the position and to avoid overlapping nodes. Specifically, for each conversion, there are two types of relationships: one-to-one or one-to-many. Typically, the first kind of relationship happens in conversion between identifiers that describe the same type of entity, for example, gene to gene. In this possibility, the inheriting of the node position has the easiest case as the new node inherits the exact coordinates of the original node. Fig. 3.16 shows the example of a conversion one-to-one from EntrezID to a Gene Name. The converted node inherits the same coordinates.

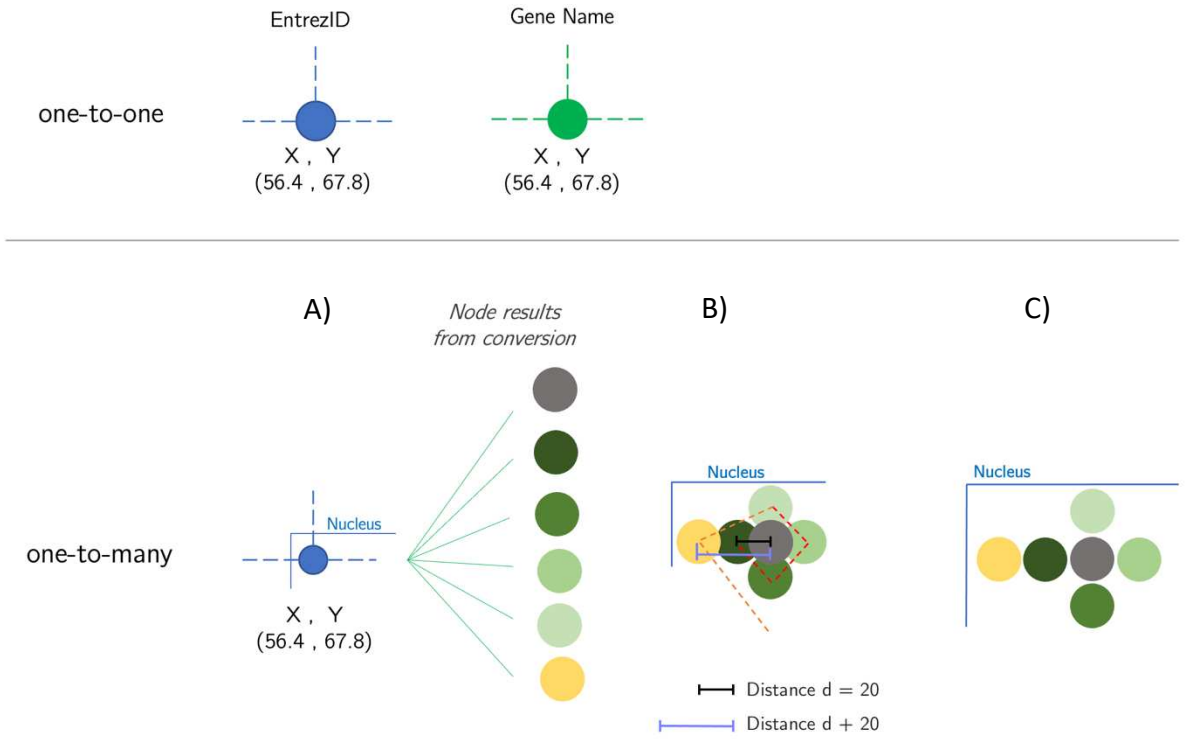


Fig. 3.16: Example procedure of the position inherited by the converted nodes. In the case of "one-to-one" conversion (top part of the image), the converted node inherited the same coordinate as the original one. For the "one-to-many" conversion type, all the resulting nodes are placed in the area of the coordinates. Specifically, the first node is placed at the same coordinates, while the remaining ones are placed on the left, bottom, right, and top of the original node, with an initial shift value. After the four directions, the shift value is increased, and the placement restarts from the left position. Specifically: A) multiple nodes result from the conversion. B) Each one is placed as described, and eventually, the compound node is resized automatically by the Cytoscape.js environment. C) A made-up function to remove overlapping elements is executed, and eventually, the compound node is resized automatically by the Cytoscape.js environment.

In the case of a one-to-many relationship, a strategy is needed to avoid multiple overlapping nodes, because if all the converted nodes would have inherited the same coordinates, the visual display on the network will result in multiple circles and labels overlapping each other in the same spot. This situation is even worse considering that it may happen for different node conversions in the same network, leading to an unsuitable visualization where the user has to solve several overlapping elements manually. For this reason, I adopted a strategy that can place multiple nodes in the

same area of the original coordinates, without any overlap. Specifically, starting from the inherited coordinates, the algorithm places the first entity in the same positions, and the remaining nodes are placed shifted in four directions: top, bottom, left, and right. If there are other nodes to be placed, the algorithm increases the shift value and restarts the positioning in the four directions. In the end, it runs a function to remove completely overlapping nodes.

The conversion may lead to producing many more nodes than those of the original network and it may happen with a conversion that leads to a different biological entity, such as EntrezID (gene) to RefSeq (Transcripts, Protein). In this case, the approach may not be enough to solve all the overlaps. To solve this problem, the user can reuse the button to instantly apply the same function to remove the remaining overlapping elements, instead of manually moving the nodes.

Fig. 3.16 illustrates the described strategy with an example conversion resulting in six nodes. As shown, all the nodes are placed sequentially starting from the original coordinate, and then positioned with a shift to the left, bottom, right, and top of the original node. The shift is increased every four iterations. This example also shows the borders of the compound "Nucleus" still containing the converted nodes, as this approach works within the Cytoscape.js environment which can automatically resize compound nodes to a suitable size to contain all its nodes. Indeed, Fig. 3.16 shows the different steps: A) the original node which is placed in the top-left corner of the compound node; B) after positioning all the converted nodes, they are partially

overlapping each other and the compound node is automatically resized; C) using a made-up function is possible to remove overlaps, and again automatically resize the compound node. Specifically of this latter function, to remove the overlaps it was used the library of WebCola (https://github.com/tgdwyer/WebCola) [67], which provides a function to compute new coordinates, giving in input all the boundaries of the shape of each node. Moreover, the WebCola function can resolve any overlaps produced in cascade by the new coordinates obtained for a node. Indeed, for each node, this function provides new coordinates describing the non-overlapping display of the entire network without drastically changing the obtained layout. Thus, using the Cytoscape.js libraries, all the nodes are moved to their computed positions, and by moving them all the compound nodes are automatically resized, leading to a clearer visualization.

To my knowledge, this is the first time that the removal of overlapping elements is integrated into a web tool. I believe that this feature can lead to a clearer visualization and helps the user study the network.

Technical development of scripts and tool

All the steps were mainly performed using made-up scripts in Python 3.9, except for the retrieved pathways from R package graphite which were obtained with an R script.

Some of the described steps also required some external tools. For example, the drawn networks were obtained using the JavaScript library Cytoscape.js [41]. Layouts were pre-computed using the SyBLaRS repository (SYstems Biology LAyout & Rendering Service, https://github.com/iVis-at-Bilkent/syblars/), and the following layout algorithms (implemented in Cytoscape.js), fCoSE and Cola, were used to compute networks layouts. In particular, to compute all the layouts of each network of each pathway, I created a Docker image with all the necessary environment and library to run the SyBLaRS server, NodeJS for the client request, and Python for some scripts to perform the call of the request and parse the output. Since there are three types of network objects for each pathway and considering that more than one hundred and seventy thousand pathways were collected from graphite, I made up a strategy to parallelize the layout computation of more than five hundred thousand networks. Starting from the docker image, a Singularity Image [68] has been created. Singularity is a container platform that allows to create and run containers that package up software in a way that is portable and reproducible, for example in large HPC clusters. Indeed, from the created singularity image, I run several containers on our HPC cluster, each one able to run a computation of 100 pathways (300 networks). This parallelization allows for minimizing the overall computation time of the layout, as the execution of each network was independent of the others.

All Power Graph networks were obtained using the application published in the paper [66].

The server and the website of the tool were developed using Python 3.9 and React.js respectively. React.js is an efficient JavaScript library for building user interfaces. An SQLite database was created to contain all the information about the pathways, networks, subcellular location, nodes' positions, and converted identifiers. Information regarding the session of the user, for example, the type of network and the pathway details (species and databases), were stored in the PostgreSQL database as anonymous information.

The graphic user interface is designed to be intuitive and user-friendly, with no bioinformatic expertise required.

4. Results and discussion

itGraph, an optimized tool for pathway visualization

Here I presented itGraph (at the time of writing is available at: https://sales.bio.unipd.it/itgraph/), a web tool to visualize more than 170 thousands biological pathways, distributed among 14 species, with three different network perspectives.

The entire tool has been designed to be optimized for various utilities and targets, both for the visualization and technical aspects. Also, the structure of the database is designed to support fast requests and the tool is configured to be fast and user-friendly in all its aspects, with no bioinformatic expertise required. All the optimizations are essential to provide visualization and biological features that are still lacking in other software.

The layout of each network is pre-computed to let the user wait only for the communication between the client-server and for the rendering time of the network objects. Networks are clearly displayed thanks to the combination of two network layout algorithms, fCoSE, and Cola, that can work with graphs with and without compound nodes. Using both sequentially, I obtained node positions that minimize

different aesthetic metric criteria, preventing overlapping elements, and providing a clear visualization. Furthermore, given that this layout approach allows for fast computing of the nodes' positions, it is also used for real-time applications such as the button of "Reset Layout", suitable for new fast computing positions.

Three biological visualization perspectives

One of the biological aspects that I have integrated to enhance the utility of the visualization is the subcellular location, both as single information of a node (through the color) and as the explicit representation of the cellular compartment as a compound node. Besides that, the tool provides three network types for different perspectives and to highlight specific features of the biological visualization: "simple network", "network with compartments", and "power graph". Each network is built starting from a SIF file obtained from the R graphite package and then parsed to compress and remove the redundant edges object. Specifically, if there were different edges between two specific nodes, each describing a different biological process, those links were compressed into a single edge with the most general abstracting direction and saving all the relative biological information from the compressed edges. This step was necessary to initially reduce the complexity of the network and compress its information into a more useful graph description. Starting from this object, all the network types described above were created. The first one is a traditional drawing of

a graph, where nodes are simply encoded as points in the space, but each one is colored according to the respective location. The results obtained are usually aesthetically pleasing. Fig. 4.1 shows an example result of the *"Hypertrophic cardiomyopathy"* Kegg pathway for the mouse (*Mus musculus*).

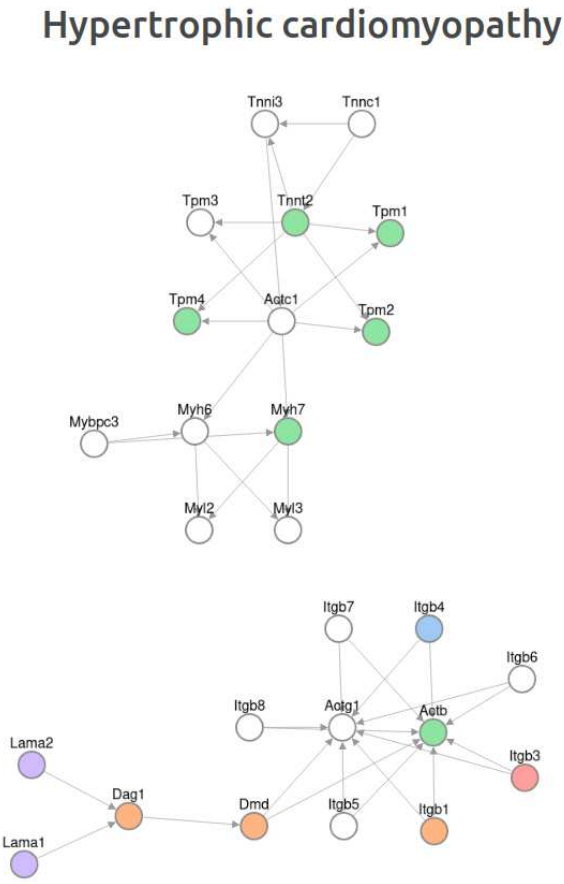


Fig. 4.1: An example result of the "simple network" type, displaying the Kegg pathway of Mus musculus "Hypertrophic cardiomyopathy".

The second type regards the explicit representation of subcellular organelles as compound nodes. Biological networks describe biological entities which most act their functions in a specific location inside the cell. This type of representation provides an insightful way to understand interactions that define molecular processes that span

different compartments. It provides an advantage over the output of other tools that do not simplify the representation of hierarchical cell compartments, thus leading to a high level of redundancy. This approach allows us to obtain more than 200 cellular compartments, including even the less common ones, distributed among the 14 species integrated by the tool. Each compartment is inserted so that it is a descendant of a given node both in terms of inclusion and display. Thanks to the MTR algorithm, the drawn graph is minimal, as it does not insert compound nodes of cellular compartments not referenced in the pathway, maintaining a minimal hierarchy of cellular organization obtained from the removed parent nodes. Fig. 4.2 shows the result of the same pathway network, with subcellular compartments drawn as compound nodes.

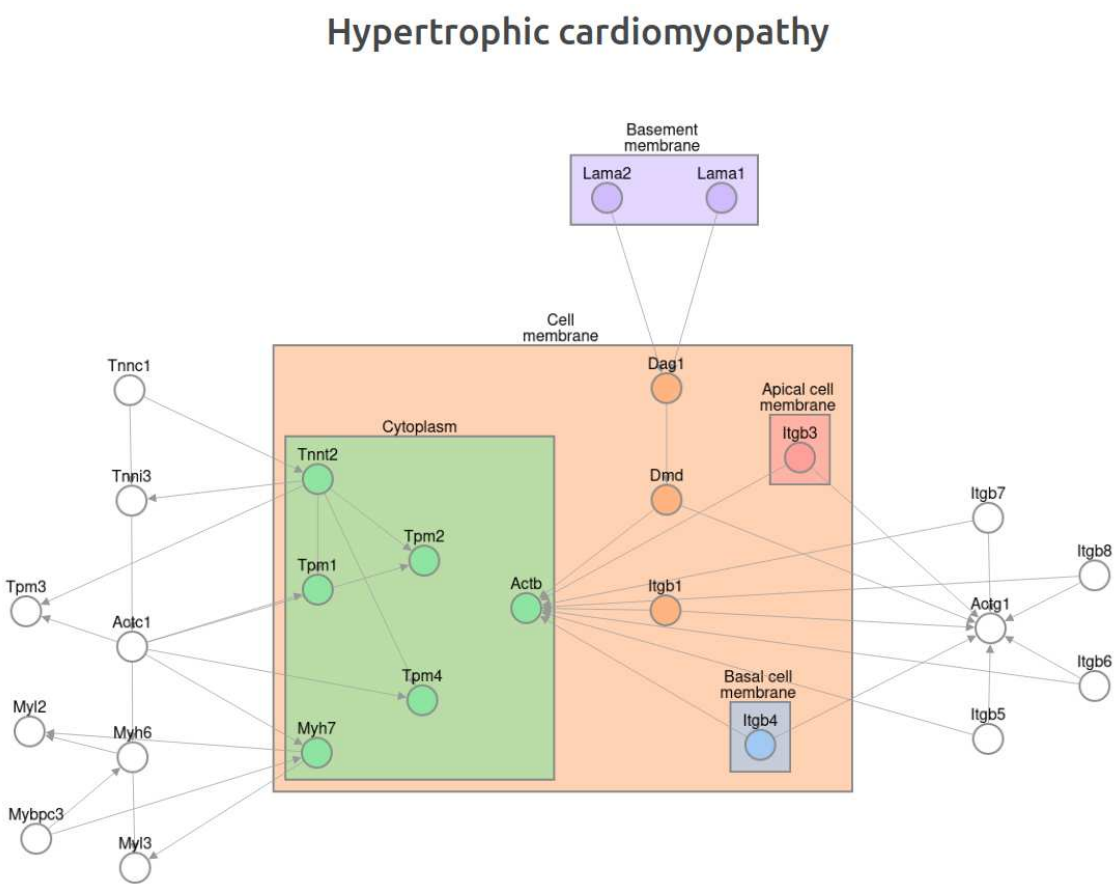


Fig. 4.2: "Hypertrophic cardiomyopathy" Kegg pathway of Mus musculus shown as "Network with compartments" type.

Our strategy is a novel solution both for the biological visualization of the entire hierarchy of the cell and for the technical storage of that organization, and it can be applied to represent cells of other kingdoms of life. This perspective covers the organization of cellular organelles for all the species of itGraph, each one designed to be as close as possible to the real organization of that cell. Lastly, it is important to say that the other approaches of the previous tool and this solution highlight the need for an informatic tool or object that can describe the hierarchical organization of the cell, also which can be useful for purposes other than viewing networks. For example, the

tree structures that I made, together with the MTR algorithm, are a starting point for this bioinformatic aspect that can be even further optimized and enhanced with other detailed biological compartments or information.

The third type of visualization is built through the power graph analysis. This method describes networks in a compact and less redundant representation, without any loss of information. Fig. 4.3 shows the resulting Power Graph of the *"Hypertrophic cardiomyopathy"* Kegg pathway of Mus musculus.

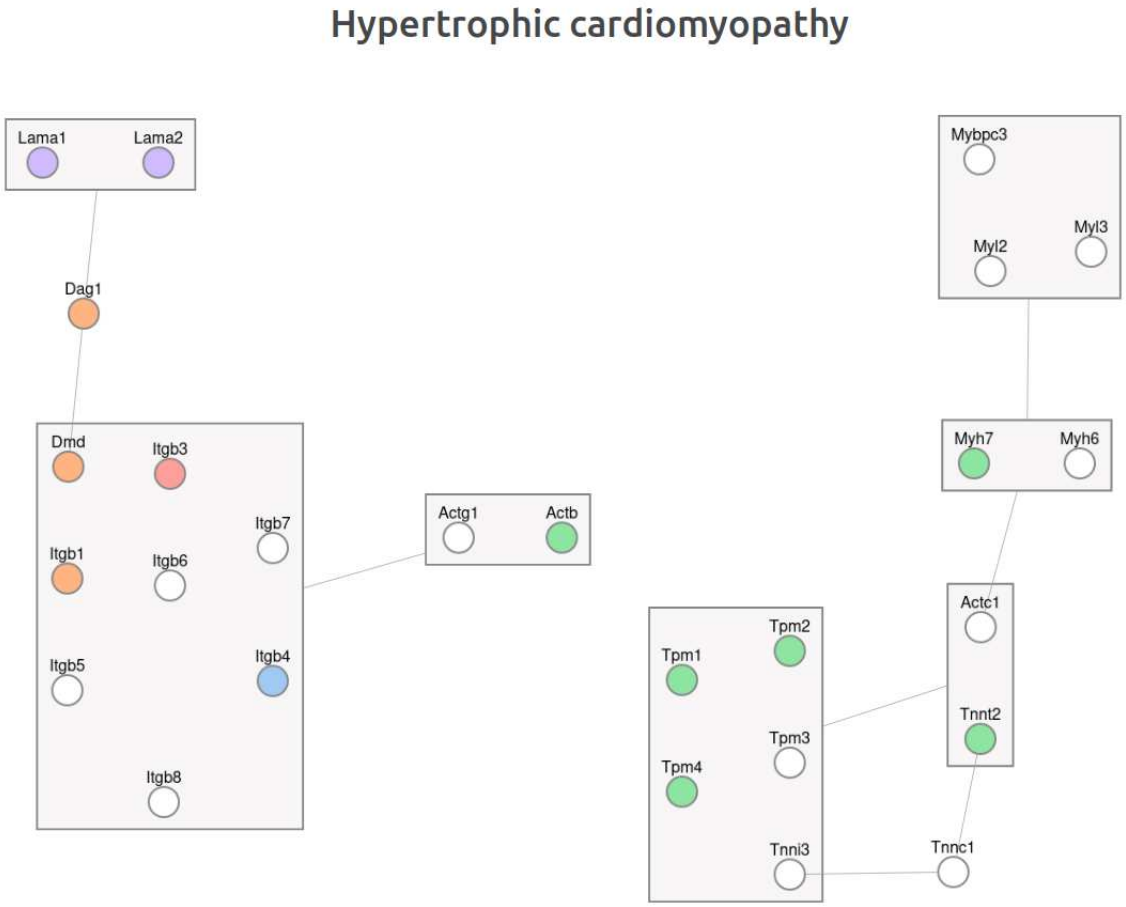


Fig. 4.3: Resulting Power Graph of the "Hypertrophic cardiomyopathy" Kegg pathway of Mus musculus.

This compression reduces the visual complexity of the network by explicitly representing motifs that are widely represented in biological networks. Such motifs are Star, Clique, and Biclique and they can represent hub proteins, protein complexes, and domain interactions respectively. This new representation is obtained with the addition of two basic structures: power nodes and power edges. By adding new elements, the power graphs analysis provides a new encoding of the graph that can describe the same information abstracting many edges with the power edge. Moreover, power nodes and power edges can help users to quickly understand the interactors of a given node of interest. This drawing approach allows handling huge networks with a lot of edges and edge crossing, otherwise not representable as they result in hairballs, from which few insights can be gathered. A power graph can be useful to give an insightful drawing of the original graph, as it encodes high-density structure motifs that are widely represented in biological networks. Furthermore, it is important to say that the method is based only on the topological information of the networks, consequently, curated network topology provides more reliable results from a biological point of view. Even if the results are not valid for biological analysis, the reduction of the visual complexity remains useful for fast viewing of the interactions of specific nodes of interest.

Additional features

Another major contribution offered by the tool is related to the conversion of the identifiers, which allows for mapping network nodes to different identifiers. To my knowledge, this feature has never been implemented in other network visualization tools and it can be used both to convert nodes of the entire network or manually single nodes. In this latter case, each result has a specific link to the page on the source database and the user has also the possibility to use one of the results as the label of the node. By doing this, the label will be colored red highlighting the fact that it was modified from its native type.

As already explained, strictly linked to the conversion of identifiers there is a management of the inherited node's position when mapping results are obtained. In particular, this handling comes up when the user requires a conversion of the entire network. Depending on the type of conversion, one-to-one or one-to-many, a specific approach is used. In the simplest case, one-to-one type, the new node inherits the same coordinates as the original nodes. In the one-to-many conversion, it happens that one node, describing the relative gene entity, is converted to its different protein identifiers'. So it was necessary to create an approach able to spread all the resulting nodes starting from the single coordinates of the original node. In these cases, all the resulting nodes inherit the original coordinate with the addition of a shift value that places them in the same area but partially overlapping. In the final step, those overlaps

are solved with a specific function that can be even activated by the user with the "Remove Overlapping Nodes" button. In particular, this function is useful as it removes the node's overlap without recomputing further iterations of the layout and thus preventing the loss of the mental map by the user.

The conversion of the identifiers of the entire network was designed as one of the first operations for the interested user, allowing him to start the analysis and the visualization of the network with another type of id, as each converted node inherits and elaborates the initial positions stored in the database and not the current ones on the network.

The user can search for a pathway of interest both on the homepage and during a network visualization in the specific panel "List". By changing the species through the dropdown menu, the search is performed for the chosen species. The results are divided by source databases that provide pathways matching the inserted searched text. By clicking on the title of a pathway, the visualization of the "simple network" of that pathway is loaded, showing the nodes with their 'native' labels.

Another integration of the tool is the possibility for the user to color nodes based on the imported data. For example, the user may be interested in viewing the nodes colored according to the log fold-change or p-values. This feature is present in the panel "Data", and it is possible to choose a color palette from a list and match node labels with or without the case-sensitive param. The import is designed to be simple and straightforward, as it requires a tabular file (TSV) with just two columns and with

no header. The first column must contain the label of the nodes while the second one the numeric values on which compute the color shade of the palette. Each matched node will be colored according to its numeric value. This feature may help the user to analyze the biological pathway networks contextualized with the gene expression measurements.

Furthermore, the tool provides the possibility to share the current session with colleagues. In particular, it allows the export of a snapshot of the positions and the colors of the nodes. Through this operation, the user can save the snapshot of the network and share the file with colleagues, which are to work locally on the same pathway visualization. Anyone who has the exported session file can import it at any time and work on the same snapshot on the network.

Regarding the user experience, each network node contains specific information regarding the identifier type, location, color, and size. Similarly, edges contain information concerning all the interactions, such as the biological processes involved and their direction between the involved nodes. The visualization is obtained with the Cytoscape.js library, thus it is interactive. Nodes can be clicked and moved, and placing the mouse over nodes all its interactors are highlighted. For the network types "simple network" and "power graph" there is also a legend for the colors of occurring cellular compartments and placing the mouse over that label, will highlight all the nodes carrying that specific location.

All the tools are developed to be intuitive and fast as much as possible. The user graphic interface is designed to be user-friendly, with no bioinformatic expertise required.

In the next paragraph, all the web interface is explained and described.

Web interface

I designed this tool to be simple, and intuitive with a user-friendly interface.

Fig. 4.4 shows the homepage, divided into three sections: the first one contains the name of the tool; the second part is where the user can start to operate: there is a search bar, that allows searching for a pathway title, a dropdown menu list to choose the species, and finally, a section with the results. In that section, there are the database boxes with the respective pathway list. Note that, without performing any search operation, thus with an empty search text, a preview of twenty pathway list is shown for each box. Each box is scrollable and placing the mouse over a pathway will show the number of nodes and edges of that pathway.

The third part regards the footer of the page, containing links to the Department of Biology and the University of Padova. At the center of this section, there is a brief citation text, linking to the publication and the laboratory of my group.

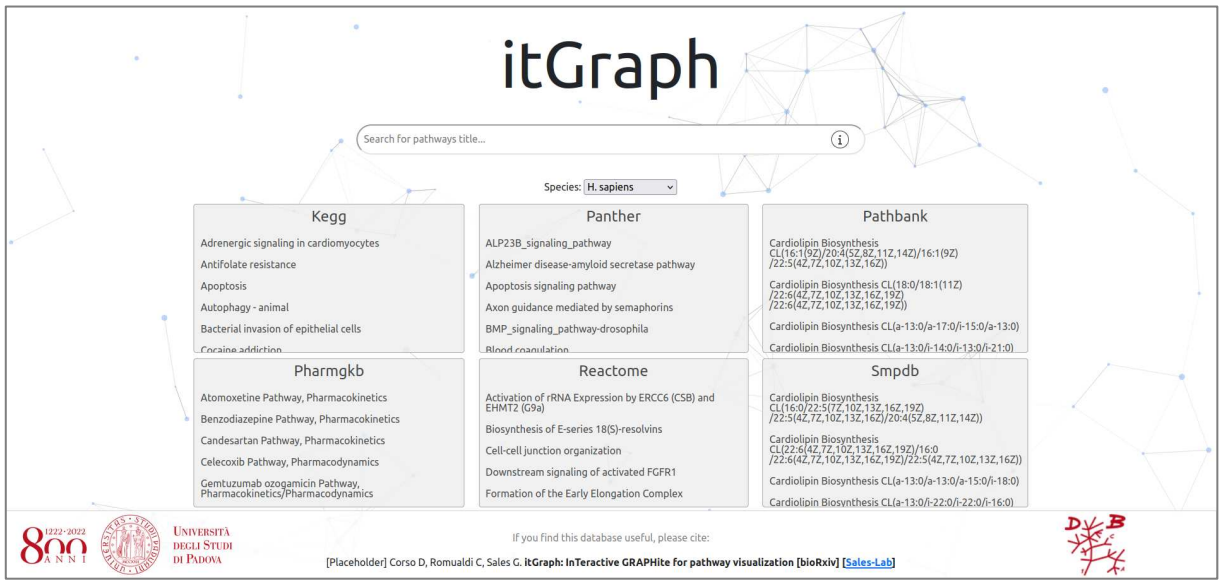


Fig. 4.4: Homepage of itGraph.

Fig. 4.5 shows the 14 species currently integrated into the tool, and through the dropdown menu, the user can choose the species of interest among the following: Arabidopsis thaliana, Bos taurus, Caenorhabditis elegans, Canis lupus familiaris, Drosophila melanogaster, Danio rerio, Escherichia coli, Gallus gallus, Homo sapiens, Mus musculus, Rattus norvegicus, Saccharomyces cerevisiae, Sus scrofa, and Xenopus laevis.

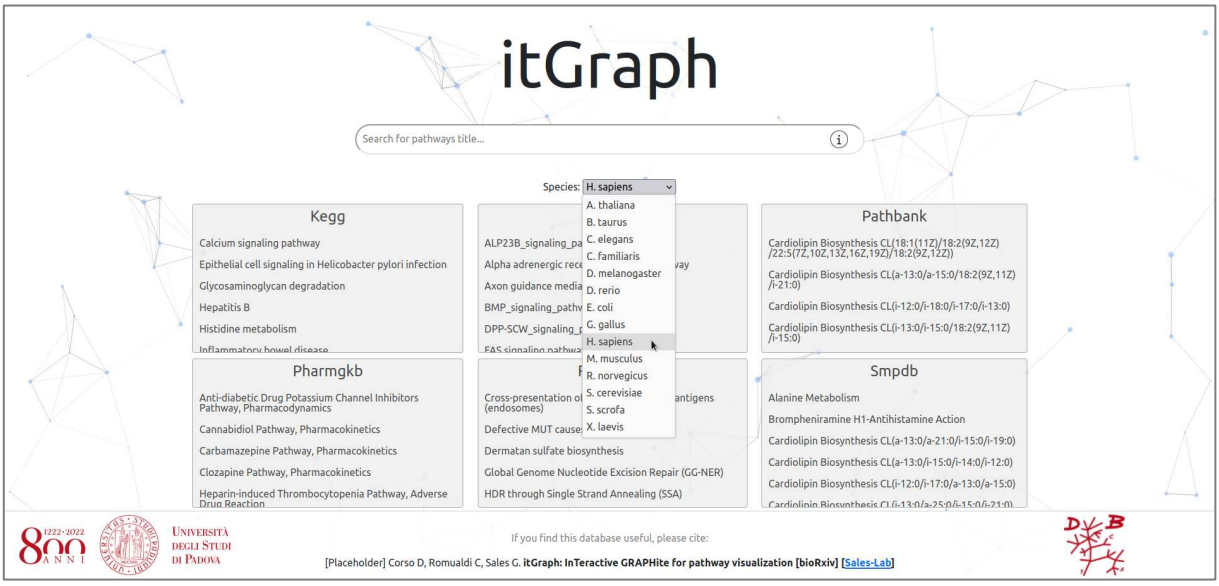


Fig. 4.5: Homepage of itGraph with the available species.

By clicking on the pathway title, the user is redirected to the relative page of the network visualization, which is divided into four parts: the title of the pathway, which also contains a button to go back to the home page, the network drawing, the toolbar, and the panel (Fig. 4.6).

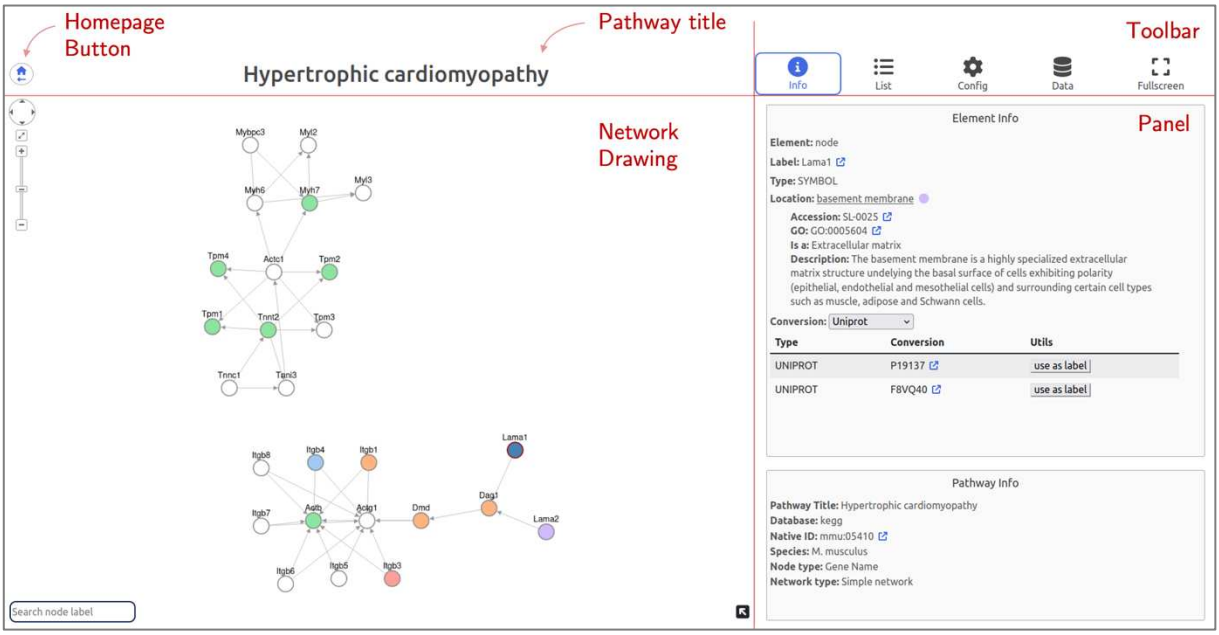


Fig. 4.6: Viewer page of the network, divided into four parts: the title of the pathway, which also contains a button to go back to the home page, the network drawing, the toolbar, and the panel.

The major part of the screen is dedicated to the interactive background of Cytoscape.js with the drawn network. Scrolling up and down with the mouse, activate the zoom-in and zoom-out respectively. These operations can also be performed with the pan-zoom located in the top-left part of this section, which contains additional features to center and move the draw in the four directions (top, bottom, left, and right). All the network elements can be clicked to display the relative information in the panel "Info", and nodes can be also moved. Moreover, by placing the mouse over a node, all its first neighbors are highlighted as shown in Fig. 4.7.

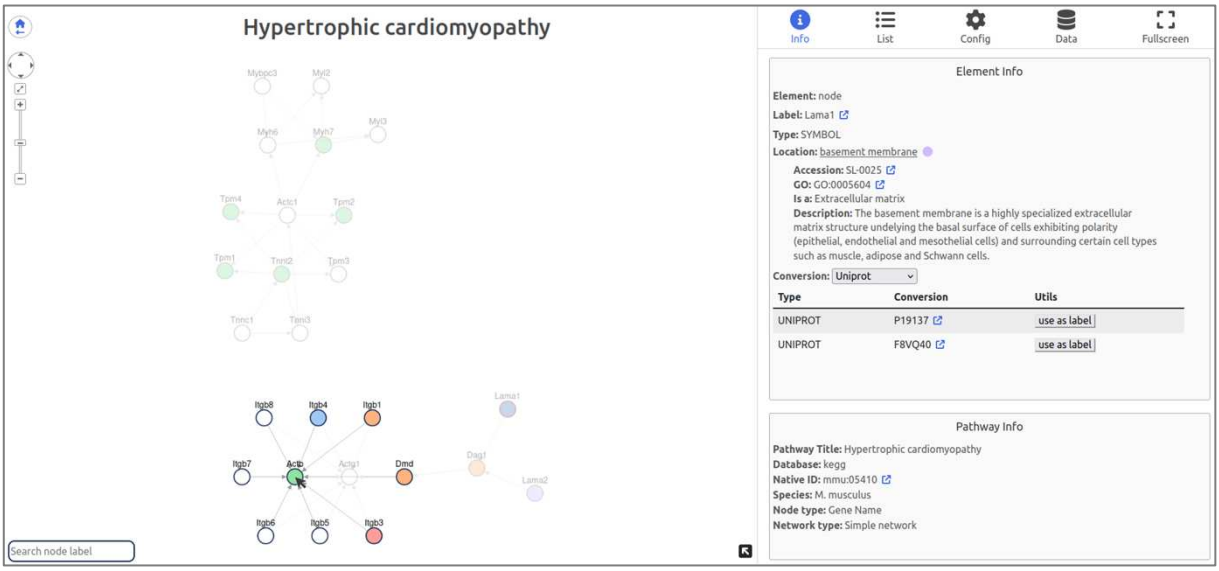


Fig. 4.7: Viewer page of the network. The mouse over the 'Actb' node highlights all its neighbors.

At the left-bottom of the section, there is an input text bar, which allows searching a node by its label. When the node's labels are matched, they are colored blue navy, otherwise, the input text will turn red marking that no results are obtained. At the right bottom of this section, there is a small icon that can be minimized/maximized to show the color legends of the subcellular locations. By placing the mouse over a listed location, all the nodes belonging to that location are highlighted (Fig. 4.8).

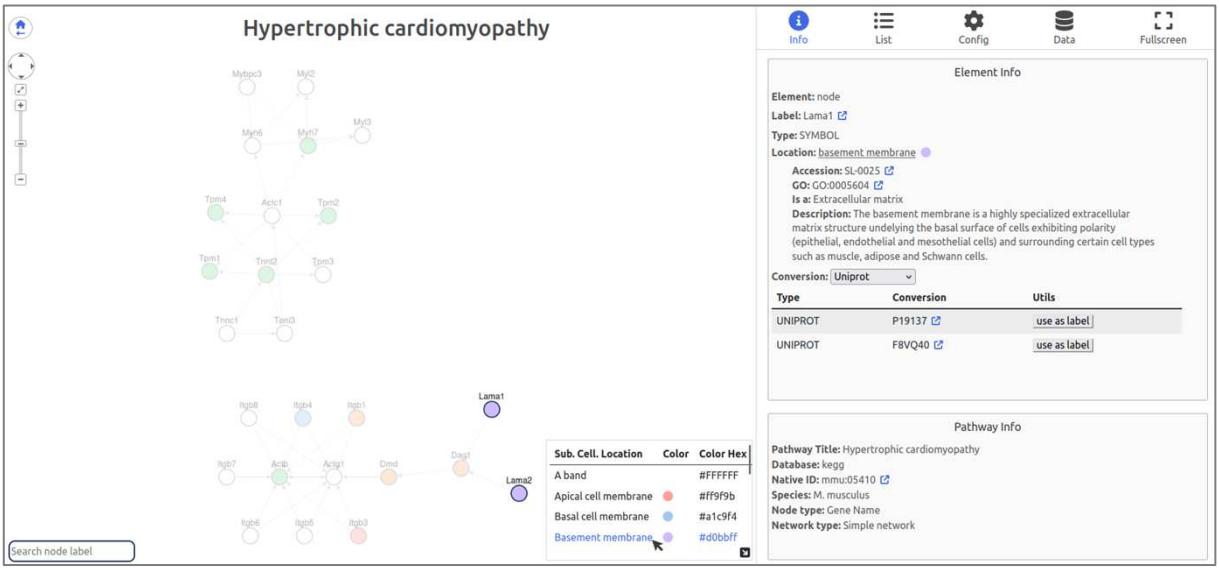


Fig. 4.8: The mouse over the label 'Basement membrane' highlights all the corresponding nodes.

The toolbar shows 5 options panel: Info, List, Config, Data, and Fullscreen.

The Info panel contains two boxes: the first one is "Element Info", which will render the information of a clicked element (node or edge), and the second one shows the pathway information ("Pathway Info"). When the user clicks on a node (Fig. 4.8), the upper box shows various information such as the label, the type of the identifier, the location, and the conversion feature. In particular, by clicking on the location text, a brief description of that subcellular location will appear with two external links to the UniProt and Ebi QuickGO sites of that location. The conversion feature allows the user to display all the mapping of the clicked node, to a specific new type of identifier. Moreover, the user can use one of the conversion results as a node label, and by doing that the new label will be colored red as a marker of a change. Using the original identifiers type of the network as the label will reset its color to black.

Similarly, when the user clicks on the edge (Fig. 4.9), that box renders various information, such as the labels and the locations of the source and the target, the direction of the edge, and the occurring biological process between these two nodes.

The "Pathway Info" box shows the following details: the title, the source database, the native ID with an external link to its visualization on the source database site, the species of the pathway, the type of the native identifiers, and the type of the current network.

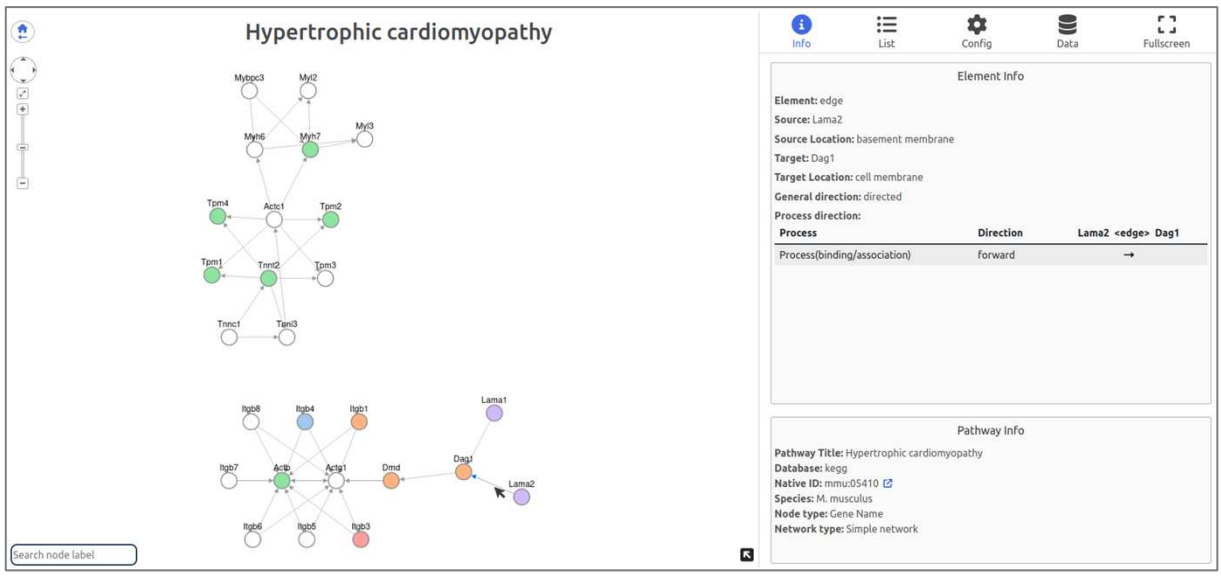


Fig. 4.9: Panel "Info" showing information about the edge between "Lama2" and "Dag1" and the pathway.

The list panel is similar to the central section of the home page (Fig. 4.10). It contains an input text bar to perform the same searching operation, a dropdown menu to choose the desired species, and the resulting database boxes with the respective pathway list. In this panel, users can open and close each box. They can even change the order of the boxes by simply dragging and dropping each box. Inserting a text in

the search bar will perform the search operation on the itGraph database and the matching resulting pathways (of that species) are displayed in the respective boxes. By changing the species through the dropdown menu, the search with the same text is performed.

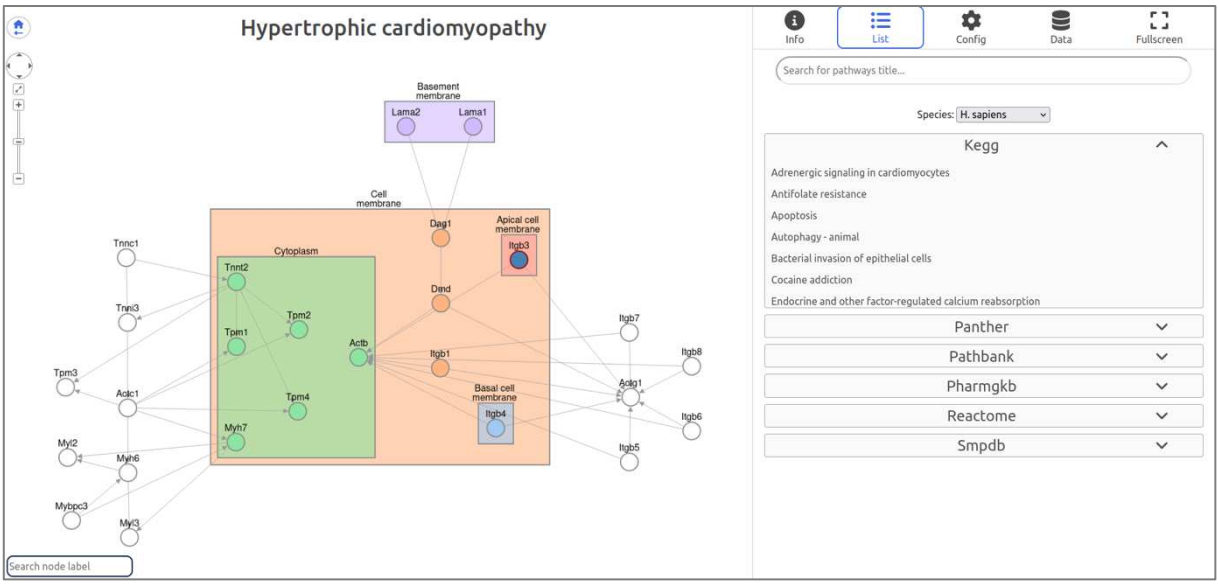


Fig. 4.10: Viewer page of the network displaying "Network with compartments" type. The "List" panel is shown.

The config panel (Fig. 4.11) contains different features that allow the user to execute different operations on the network. The first one regards the conversion of identifiers for all the nodes of the network. Through the dropdown menù, the user can choose the new type of identifier and by clicking on the "Convert" button, it performs the requested mapping. It is important to note that this feature is designed eventually as one of the first procedures to set a suitable working network, as the algorithm to handle the inheriting nodes' position work on their initial pre-computed coordinates.

The central section of this panel allows the user to change the type of the displayed network, along with four buttons to handle the visualization of the network:

- "Enhances Layout", allows performing further iterations of the layout algorithms optimizing the positions of the nodes;
- "Reset Layout", will recompute, starting over, a new layout of the network. As described, this function is obtained by combining sequentially the fCoSE and the Cola algorithms;
- "Remove Overlap": allows removing node overlaps. For the network with the explicit representation of subcellular compartments and power graph, this function can solve overlaps even between nodes and compound nodes that do not represent the right parent of those nodes.
- Reset Zoom: this feature allows to reset the zoom and center the graph to the available network screen.

In the last section, there is a link to the Power Graph publication, with an image that summarizes the encoding performed by the method. This illustration can help the user to recognize which motifs are encoded in the power nodes.

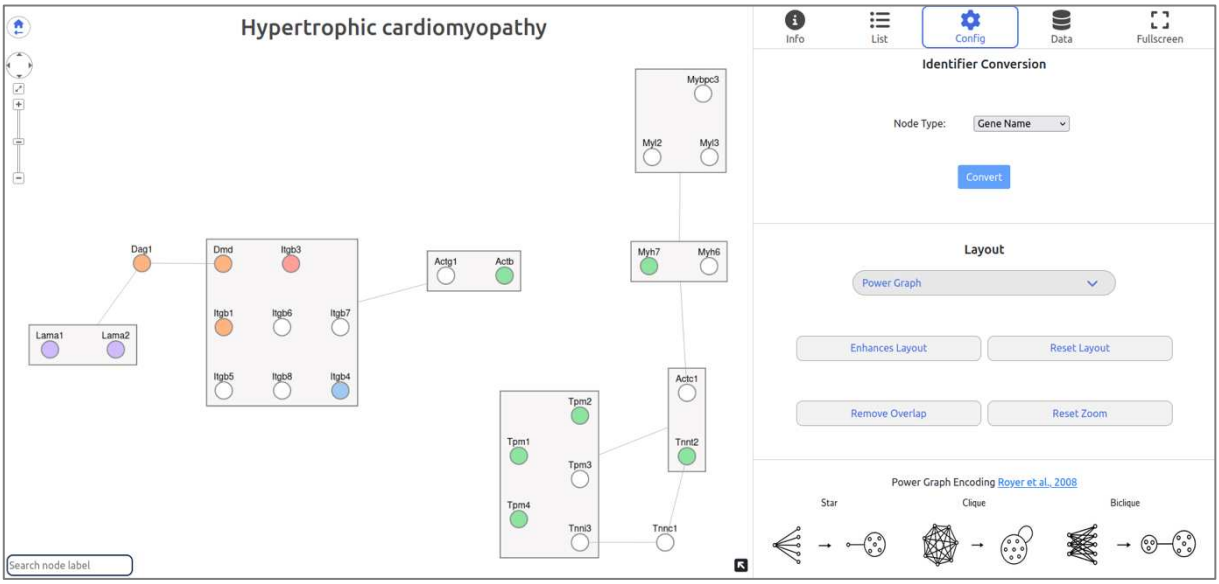


Fig. 4.11: Viewer page of the network displaying "Power Graph" type. The "Config" panel is shown.

The data panel is divided into two major sections (Fig. 4.12): the first section regards the customization of nodes' color based on the uploaded file. For example, the user may be interested in viewing the nodes colored according to the computed log fold-changes or p-values, by choosing a color palette and matching nodes label with or without the case-sensitive param. This feature requires a tabular file (TSV) with just two columns and with no header. The first column must contain the label of the nodes while the second one the numeric values on which computing the color shade of the palette. Each matched node will be colored according to its numeric value. These features may help the user to analyze the biological pathway networks contextualized with his gene expression measurements. The second section of this panel gives the possibility to export a session file, which can be useful to save a precise snapshot of

the colors and positions of the network. This file is also sharable, and at any time by importing it any user can load that saved snapshot of the network.

Furthermore, at the bottom of this panel, there are buttons to save the image of the network or to download it as a text JSON file.

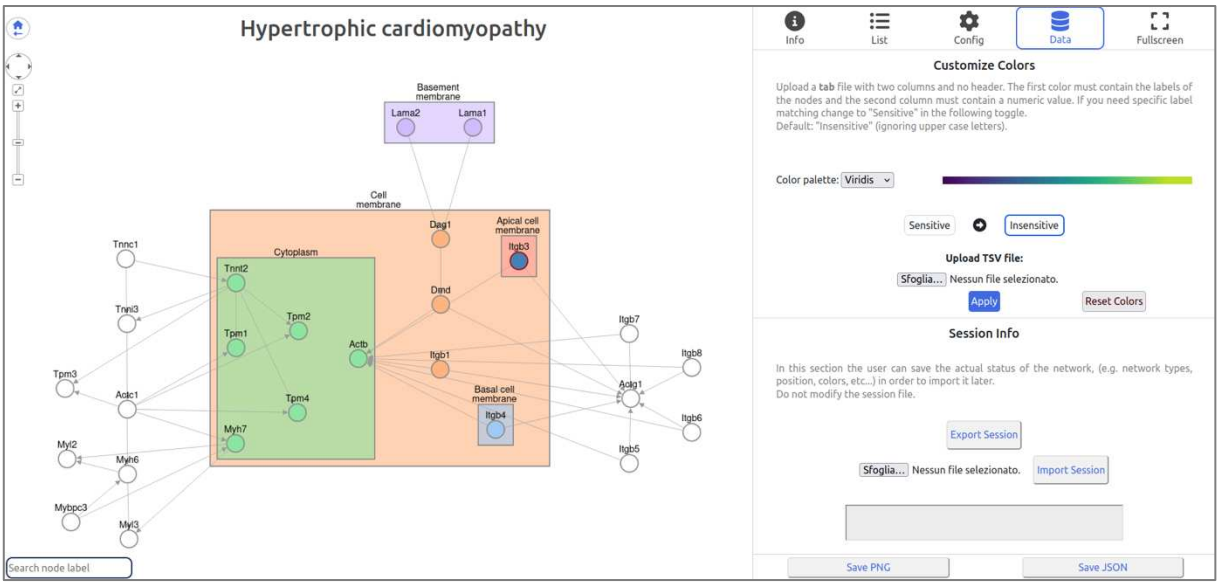


Fig. 4.12: Viewer page of the network displaying "Network with compartments" type. The "Data" panel is shown.

The last option in the toolbar allows the user to visualize the network in fullscreen, hiding the title and the panel (Fig. 4.13).

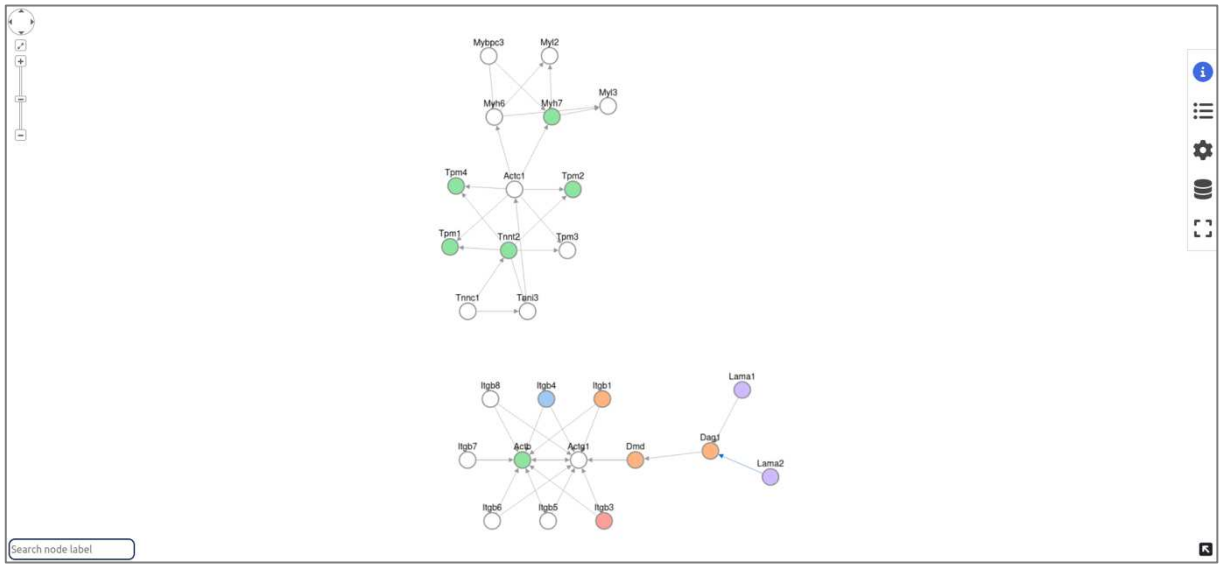


Fig. 4.13: Network drawing section in fullscreen.

The interface of the tool was developed with React.js, which is an efficient JavaScript library for building user interfaces. The graphic user interface is designed to be intuitive and user-friendly, with no bioinformatic expertise required. In general, the entire tool, including the databases, was designed and developed to provide fast responses to user requests.

5. Conclusions

Here I presented itGraph, a novel web tool for pathway visualization. The aim was to provide a user-friendly tool to explore pathways of interest.

It was developed focusing on the following aspects and integrations, as I believed that specific focus and optimization allow for providing a more useful tool with resolute goals and characteristics for the users:

- increasing biological accuracy by integrating explicit representation of cellular compartments;
- reduction of the visual complexity of the network;
- integration of biological annotation and features to enhance the visualization;
- optimize the user experience by creating a user-friendly tool.

itGraph limits the maximum size of visualized pathways. In particular, the sum of the number of nodes and edges must be equal to or lower than 2000, as huge graphs are still a bottleneck of the networks' visualization both for their layout and for the speed and responsiveness of their render. Even applying this constraint, the number of

collected and provided pathways is more than 170 thousand, distributed among 14 species, resulting in more than 510 thousand networks with a pre-computed layout.

The range of itGraph pathway list is far greater than those provided by Pathways Common, which (at the time of writing, at version 12) covers only the human species, providing 5,772 pathways.

itGraph can be considered a comprehensive resource to visualize pathways networks with three different biological perspectives, each one with a precise focus. The first type is the simplest, which represents traditional draws of the network integrating subcellular location as node color. The aesthetic appeal of the resulting layout, in addition to the interactive features of the tool, makes this type of visualization useful mostly for small-medium graphs. However, bigger graphs with a curated topology are clearly represented in this type of visualization. In the second representation, the tool provides a non-trivial network representation including subcellular organelles as compound nodes. The third type regards the power graph analysis to describe the network into a compact and less redundant representation, reducing the visual complexity. The power graph, beyond giving an insightful encoding of biological motifs, can be helpful as it reduces the visual complexity of the network, and even if it does not provide a biological reliable compression, it still can help the user to quickly analyze specific interactions of nodes of interest.

The conversion of identifiers is a useful feature for biological analysis, as it allows mapping nodes to different identifiers, and the results can be applied instantly to the

network. Each conversion result comes with an external link to its page on the source database. This is useful as it provides intuitive cross-references between the annotations provided by different databases. To handle multiple conversions applied to the network, I solved the problem of how to place multiple nodes in the same area of the original node, without any overlaps. Derived from this strategy, I have integrated into itGraph a function to remove overlapping nodes, which can work even for compound graphs, such as the "network with compartments" and "power graph". In these cases, this function can solve overlaps even between nodes and compound nodes that do not represent the right parent of those nodes.

The users can also manage to import their own data to color network nodes according to their values, contextualizing the pathway visualization to their experimental measures. At any time, the users can save the state of its visualization and by exporting the session, the nodes' colors and positions are saved into a shareable text file. By importing it, the file will resume the network at that state.

To conclude, the present project describes a tool that aims to provide a suitable visualization of biological pathways with the subcellular location, in addition to biological features that can help users to analyze a pathway of interest.

6. Future perspectives

itGraph is built to work only on stored pathways obtained from graphite. Although can be considered a closed system, it can be enhanced with other visualization features and analysis operations, without de-structuring the entire organization of the tool.

Concerning the first aspect, it could be useful to expand and collapse compound nodes, therefore improving the visualization of compound graphs and at the same time reducing the necessary space to draw the network.

Moreover, to improve the rendering speed of the elements in the web tool, network object rendering could be performed using GPU (Graphics Processing Unit). This characteristic is likely to increase the responsiveness of the network as well as reduce the waiting time for the user before the network becomes available.

Another interesting characteristic that can be integrated is the annotation of the known protein complexes recognized and encoded by the power graph. Having such a description will surely increase the biological accuracy of this network representation and would help the user to decode those motifs in their biological context and functions.

Similarly, the tool structure could accommodate other types of computations. Cytoscape.js is one of the most important libraries for web network visualization and is constantly updated and provides many functions to perform advanced analysis. For example, it allows us to compute various network traversals. It also provides functions for shortest paths, or network centralities operations like betweenness, that measure which nodes are essential for the communication between different parts of the networks. Furthermore, in Cytoscape.js, several plugins are often created by the community, and they are quite interesting and useful for visualization, thus the adopted visualization can be anytime enhanced with other plugins and expansions.

Another potential study to include is the enrichment analysis. In particular, it would be valid for the user to identify enriched entries over the entire list of pathways of a certain species. Successively, the tool could link to the network visualization of that pathway, integrating the user nodes.

Currently, the tool does not provide the possibility to import user networks, as the tool is designed to run on pre-computed data. Nevertheless, the three different types of networks are useful for various analyses, and also the power graph application could be applied to a run-time as it typically computes the encoding in less than a second.

Finally, future releases could provide pathways also with the integration of metabolites, useful to understand which molecules act as a bridge between two elements.

References

1. Pavlopoulos GA, Secrier M, Moschopoulos CN, Soldatos TG, Kossida S, Aerts J, et al. Using graph theory to analyze biological networks. BioData Mining. 2011. doi:10.1186/1756-0381-4-10
2. Cusick ME, Klitgord N, Vidal M, Hill DE. Interactome: Gateway into systems biology. Hum Mol Genet. 2005;14. doi:10.1093/hmg/ddi335
3. Jorďn F, Nguyen TP, Liu W chung. Studying protein-protein interaction networks: A systems view on diseases. Brief Funct Genomics. 2012;11. doi:10.1093/bfgp/els035
4. Fionda V. Networks in biology. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics. 2018. doi:10.1016/B978-0-12-809633-8.20420-2
5. Hauschild AC, Pastrello C, Rossos AEM, Jurisica I. Visualization of biomedical networks. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics. 2018. doi:10.1016/B978-0-12-809633-8.20430-5
6. Çakir T, Patil KR, Önsan ZI, Ülgen KÖ, Kirdar B, Nielsen J. Integration of metabolome data with metabolic networks reveals reporter reactions. Mol Syst Biol. 2006;2. doi:10.1038/msb4100085
7. Jeong H, Tombort B, Albert R, Oltvait ZN, Barabasi AL. The large-scale organization of metabolic networks. The Structure and Dynamics of Networks.

2011. doi:10.1515/9781400841356.211

8. Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. Nature Reviews Molecular Cell Biology. 2008. doi:10.1038/nrm2503

9. Emmert-Streib F, Dehmer M, Haibe-Kains B. Gene regulatory networks and their applications: Understanding biological and medical problems in terms of networks. Frontiers in Cell and Developmental Biology. 2014. doi:10.3389/fcell.2014.00038

10. Cary MP, Bader GD, Sander C. Pathway information for systems biology. FEBS Letters. 2005. doi:10.1016/j.febslet.2005.02.005

11. Villaveces JM, Koti P, Habermann BH. Tools for visualization and analysis of molecular networks, pathways, and -omics data. Adv Appl Bioinforma Chem. 2015;8. doi:10.2147/AABC.S63534

12. Ma'ayan A. Insights into the organization of biochemical regulatory networks using graph theory analyses. Journal of Biological Chemistry. 2009. doi:10.1074/jbc.R800056200

13. Ma'ayan A, Jenkins SL, Neves S, Hasseldine A, Grace E, Dubin-Thaler B, et al. Formation of regulatory patterns during signal propagation in a mammalian cellular network. Science (80-). 2005;309. doi:10.1126/science.1108876

14. Agapito G. Visualization of biological pathways. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics. 2018. doi:10.1016/B978-0-12-809633-8.20497-4

15. Beretta S, Denti L, Previtali M. Graph theory and definitions. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics. 2018. doi:10.1016/B978-0-12-809633-8.20421-4

16. Bhalla US, Iyengar R. Emergent properties of networks of biological signaling pathways. Science (80-). 1999;283. doi:10.1126/science.283.5400.381

17. Koutrouli M, Karatzas E, Paez-Espino D, Pavlopoulos GA. A Guide to Conquer the Biological Network Era Using Graph Theory. Frontiers in Bioengineering and Biotechnology. 2020. doi:10.3389/fbioe.2020.00034

18. Fukuda KI, Takagi T. Knowledge representation of signal transduction pathways. Bioinformatics. 2001;17. doi:10.1093/bioinformatics/17.9.829

19. Dogrusoz U, Giral E, Cetintas A, Civril A, Demir E. A layout algorithm for undirected compound graphs. Inf Sci (Ny). 2009;179. doi:10.1016/j.ins.2008.11.017

20. Haleem H, Wang Y, Puri A, Wadhwa S, Qu H. Evaluating the Readability of Force Directed Graph Layouts: A Deep Learning Approach. IEEE Comput Graph Appl. 2019;39. doi:10.1109/MCG.2018.2881501

21. Siebenhaller M, Nielsen SS, McGee F, Balaur I, Auffray C, Mazein A. Human-like layout algorithms for signalling hypergraphs: Outlining requirements. Brief Bioinform. 2018;21. doi:10.1093/bib/bby099

22. Purchase H. Which aesthetic has the greatest effect on human understanding? Lecture Notes in Computer Science (including subseries Lecture Notes in

Artificial Intelligence and Lecture Notes in Bioinformatics). 1997. doi:10.1007/3-540-63938-1_67

23. Purchase HC, James MI, Cohen RF. An Experimental Study of the Basis for Graph Drawing Algorithms. ACM J Exp Algorithmics. 1997;2. doi:10.1145/264216.264222

24. Purchase HC, Pilcher C, Plimmer B. Graph drawing aestheticscreated by users, not algorithms. IEEE Trans Vis Comput Graph. 2012;18. doi:10.1109/TVCG.2010.269

25. Agapito G, Guzzi PH, Cannataro M. Visualization of protein interaction networks: Problems and solutions. BMC Bioinformatics. 2013;14. doi:10.1186/1471-2105-14-S1-S1

26. Sugiyama K, Tagawa S, Toda M. Methods for Visual Understanding of Hierarchical System Structures. IEEE Trans Syst Man Cybern. 1981;11. doi:10.1109/TSMC.1981.4308636

27. Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. Softw Pract Exp. 1991;21. doi:10.1002/spe.4380211102

28. Eades P. A Heuristic for Graph Drawing. 1984.

29. Rahman MK, Haque Sujon M, Azad A. BatchLayout: A Batch-Parallel Force-Directed Graph Layout Algorithm in Shared Memory. IEEE Pacific Visualization Symposium. 2020. doi:10.1109/PacificVis48177.2020.3756

30. Kamada T, Kawai S. An algorithm for drawing general undirected graphs. Inf

Process Lett. 1989;31. doi:10.1016/0020-0190(89)90102-6

31. Sugiyama K, Misue K. A generic compound graph visualizer/manipulator: D-ABDUCTOR. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 1996. doi:10.1007/bfb0021834

32. Raitner M. HGV: A library for hierarchies, graphs, and views. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2002. doi:10.1007/3-540-36151-0_22

33. Sugiyama K, Misue K. Visualization of Structural Information: Automatic Drawing of Compound Digraphs. IEEE Trans Syst Man Cybern. 1991;21. doi:10.1109/21.108304

34. Sander G. Layout of compound directed graphs. 1996. doi:10.22028/D291-25806

35. Eades P, Feng QW, Lin X. Straight-line drawing algorithms for hierarchical graphs and clustered graphs. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 1997. doi:10.1007/3-540-62495-3_42

36. Balci H, Dogrusoz U. fCoSE: a fast compound graph layout algorithm with constraint support. IEEE Trans Vis Comput Graph. 2021. doi:10.1109/TVCG.2021.3095303

37. Koren Y. On spectral graph drawing. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2003;2697. doi:10.1007/3-540-

45071-8_50

38. Dwyer T, Koren Y, Marriott K. IPSEP-COLA: An incremental procedure for separation constraint layout of graphs. IEEE Transactions on Visualization and Computer Graphics. 2006. doi:10.1109/TVCG.2006.156

39. Dwyer T, Marriott K, Wybrow M. Topology preserving constrained graph layout. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2009. doi:10.1007/978-3-642-00219-9_22

40. Dwyer T. Scalable, Versatile and simple constrained graph layout. Comput Graph Forum. 2009;28. doi:10.1111/j.1467-8659.2009.01449.x

41. Franz M, Lopes CT, Huck G, Dong Y, Sumer O, Bader GD. Cytoscape.js: A graph theory library for visualisation and analysis. Bioinformatics. 2016;32: 309–311. doi:10.1093/bioinformatics/btv557

42. Baryshnikova A. Systematic Functional Annotation and Visualization of Biological Networks. Cell Syst. 2016;2. doi:10.1016/j.cels.2016.04.014

43. Novère N Le, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, et al. The Systems Biology Graphical Notation. Nature Biotechnology. 2009. doi:10.1038/nbt.1558

44. Rougny A, Touré V, Moodie S, Balaur I, Czauderna T, Borlinghaus H, et al. Systems Biology Graphical Notation: Process Description language Level 1 Version 2.0. Journal of integrative bioinformatics. 2019. doi:10.1515/jib-2019-

0022

45. Mi H, Schreiber F, Moodie S, Czauderna T, Demir E, Haw R, et al. Systems Biology Graphical Notation: Activity Flow language Level 1 Version 1.2. J Integr Bioinform. 2015;12. doi:10.2390/biecoll-jib-2015-265

46. Sorokin A, Le Novère N, Luna A, Czauderna T, Demir E, Haw R, et al. Systems Biology Graphical Notation: Entity Relationship language Level 1 Version 2. J Integr Bioinform. 2015;12. doi:10.2390/biecoll-jib-2015-264

47. Zhu L, Malatras A, Thorley M, Aghoghogbe I, Mer A, Duguez S, et al. CellWhere: Graphical display of interaction networks organized on subcellular localizations. Nucleic Acids Res. 2015;43. doi:10.1093/nar/gkv354

48. Heberle H, Carazzolle MF, Telles GP, Meirelles GV, Minghim R. CellNetVis: A web tool for visualization of biological networks using force-directed layout constrained by cellular components. BMC Bioinformatics. 2017;18. doi:10.1186/s12859-017-1787-5

49. Rodchenkov I, Babur O, Luna A, Aksoy BA, Wong J V., Fong D, et al. Pathway Commons 2019 Update: Integration, analysis and exploration of pathway data. Nucleic Acids Res. 2020;48. doi:10.1093/nar/gkz946

50. Szklarczyk D, Gable AL, Nastou KC, Lyon D, Kirsch R, Pyysalo S, et al. The STRING database in 2021: Customizable protein-protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Res. 2021;49. doi:10.1093/nar/gkaa1074

51. Calderone A, Castagnoli L, Cesareni G. Mentha: A resource for browsing integrated protein-interaction networks. Nature Methods. 2013. doi:10.1038/nmeth.2561

52. Dallago C, Goldberg T, Andrade-Navarro MA, Alanis-Lobato G, Rost B. CellMap visualizes protein-protein interactions and subcellular localization. F1000Research. 2018;6. doi:10.12688/f1000research.12707.2

53. Dallago C, Goldberg T, Andrade-Navarro MA, Alanis-Lobato G, Rost B. Visualizing Human Protein-Protein Interactions and Subcellular Localizations on Cell Images Through CellMap. Curr Protoc Bioinforma. 2020;69. doi:10.1002/cpbi.97

54. Calderone A, Cesareni G. SPV: A JavaScript signaling pathway visualizer. Bioinformatics. 2018;34. doi:10.1093/bioinformatics/bty188

55. Perfetto L, Briganti L, Calderone A, Perpetuini AC, Iannuccelli M, Langone F, et al. SIGNOR: A database of causal relationships between biological entities. Nucleic Acids Res. 2016;44. doi:10.1093/nar/gkv1048

56. Sales G, Calura E, Cavalieri D, Romualdi C. Graphite - a Bioconductor package to convert pathway topology to gene network. BMC Bioinformatics. 2012;13. doi:10.1186/1471-2105-13-20

57. Huala E, Dickerman AW, Garcia-Hernandez M, Weems D, Reiser L, LaFond F, et al. The Arabidopsis Information Resource (TAIR): A comprehensive database and web-based information retrieval, analysis, and visualization

system for a model plant. Nucleic Acids Res. 2001;29. doi:10.1093/nar/29.1.102

58. Bateman A, Martin MJ, Orchard S, Magrane M, Agivetova R, Ahmad S, et al. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res. 2021;49. doi:10.1093/nar/gkaa1100

59. Maglott D, Ostell J, Pruitt KD, Tatusova T. Entrez gene: Gene-centered information at NCBI. Nucleic Acids Res. 2011;39. doi:10.1093/nar/gkq1237

60. Gramates LS, Agapite J, Attrill H, Calvi BR, Crosby MA, dos Santos G, et al. FlyBase: a guided tour of highlighted features. Genetics. 2022;220. doi:10.1093/genetics/iyac035

61. Cunningham F, Allen JE, Allen J, Alvarez-Jarreta J, Amode MR, Armean IM, et al. Ensembl 2022. Nucleic Acids Res. 2022;50. doi:10.1093/nar/gkab1049

62. Cherry JM, Hong EL, Amundsen C, Balakrishnan R, Binkley G, Chan ET, et al. Saccharomyces Genome Database: The genomics resource of budding yeast. Nucleic Acids Res. 2012;40. doi:10.1093/nar/gkr1029

63. Chavan SS, Shaughnessy JD, Edmondson RD. Overview of biological database mapping services for interoperation between different “omics” datasets. Human Genomics. 2011. doi:10.1186/1479-7364-5-6-703

64. Buza TJ, Mccarthy FM, Wang N, Bridges SM, Burgess SC. Gene Ontology annotation quality analysis in model eukaryotes. Nucleic Acids Res. 2008;36. doi:10.1093/nar/gkm1167

65. Le Mercier P, Bolleman J, de Castro E, Gasteiger E, Bansal P, Auchincloss AH,

et al. SwissBioPics—an interactive library of cell images for the visualization of subcellular location data. Database. 2022;2022: baac026. doi:10.1093/database/baac026

66. Royer L, Reimann M, Andreopoulos B, Schroeder M. Unraveling protein networks with power graph analysis. PLoS Comput Biol. 2008;4. doi:10.1371/journal.pcbi.1000108

67. Dwyer T, Marriott K, Stuckey PJ. Fast node overlap removal. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2006. doi:10.1007/11618058_15

68. Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. PLoS One. 2017;12. doi:10.1371/journal.pone.0177459

MyoData

Summary

Skeletal muscle is the most abundant tissue in mammals and is responsible not only for their movement but also for metabolic functions. The smallest complete contractile system of skeletal muscle influencing its contraction velocity and metabolism are myofibers: large, multinucleated cells that are enwrapped by connective tissue to form fasciculi. Myofiber types are plastic and respond to specific stimuli by changing their traits and thus altering the physiology of the entire muscle to which they belong. In recent years, high throughput studies on single myofibers revealed that myofiber functional differences may also be mediated by the interplay between microRNAs (miRNAs), long non-coding RNA (lncRNAs), and mRNAs defining co-regulatory mechanisms. Although databases focusing on gene regulation by transcription factors and miRNAs have already been published, the integration of miRNAs and lncRNA activities in the regulation of gene expression at single-cell resolution has been

underappreciated. Furthermore, non-coding and coding RNAs interact with each other to regulate the actual gene expression patterns: miRNAs regulate coding RNAs through post-transcriptional mechanisms, and lncRNAs, in turn, regulate the expression of coding RNAs but also miRNA function by sponging them. For this reason, to better understand the molecular mechanisms involved in the functional specification of the different myofiber types, we integrated gene expression data of coding and non-coding RNAs to produce comprehensive lncRNAs-miRNAs-mRNAs interaction networks.

The present project describes MyoData, a database that collects gene expression data of coding and non-coding genes in single myofibers and uses them to produce interaction networks based on expression correlations. Indeed, it integrates miRNA:lncRNA:mRNA coregulatory networks for single myofiber and nucleus, also evaluating their impact on known pathways such as those present in the KEGG collection.

The interactive Network Viewer (NV) provided by this tool was created as a branch of itGraph, developing a minimal version of the entire structure that allows the network visualization through the Cytoscape.js library, as a stand-alone component. NV is a JavaScript (JS) package that can be installed in any JS graphic interface. It provides five basic functions, which are also integrated into itGraph: Enhance Layout, Reset Layout, Reset Zoom, Save PNG, and download JSON. Furthermore, the user can choose the maximum time of the layout computation.

Beyond the integrated Network Viewer, the database provides interactive charts for gene expression data. These interactive plots allow users to visualize precise values on the expression bar or show/hide/highlight expression levels of genes in two-way comparisons.

Despite the minimal structure, the development of the Network Viewer component was very important to understand and optimize technical aspects of itGraph. MyoData was designed as a user-friendly resource, requiring no bioinformatics expertise from the end user.

The result of this work was described in the paper "MyoData: An expression knowledgebase at single cell/nucleus level for the discovery of coding-noncoding RNA functional interactions in skeletal muscle. Comput Struct Biotechnol J. 2021 Jul 26;19:4142-4155. doi: 10.1016/j.csbj.2021.07.020. PMID: 34527188; PMCID: PMC8342900.".

MyoData: An expression knowledgebase at single cell/nucleus level for the discovery of coding-noncoding RNA functional interactions in skeletal muscle

Davide Corso, Francesco Chemello, Enrico Alessio, Ilenia Urso, Giulia Ferrarese, Martina Bazzega, Chiara Romualdi, Gerolamo Lanfranchi , Gabriele Sales, Stefano Cagnin.

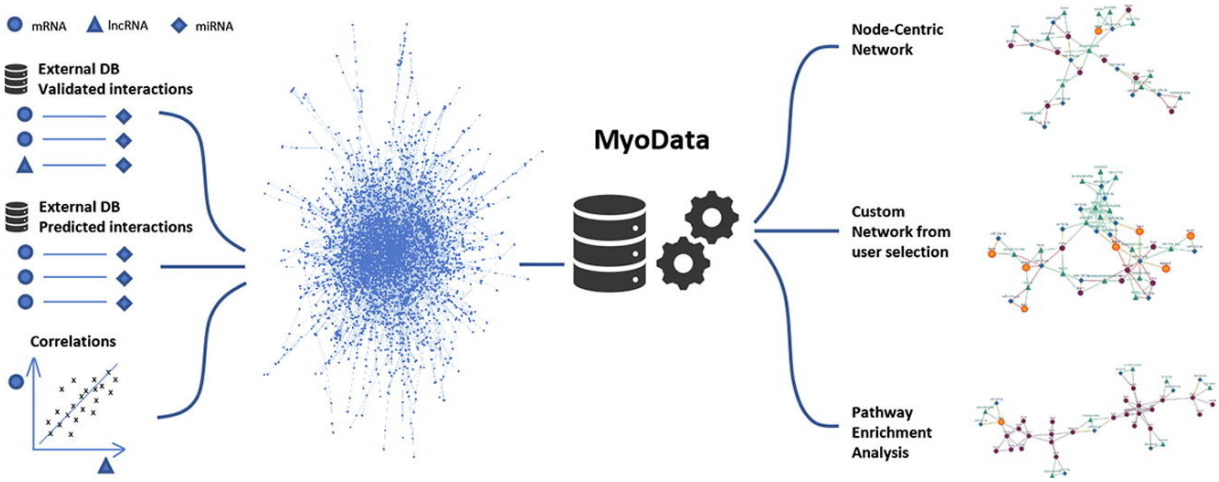
Highlights

- Regulation of gene expression through non-coding RNAs at single myofiber and nucleus resolution.
- Reinterpretation of KEGG pathways with microRNA and long non-coding RNA activities.
- miR-149, -214, and let-7e alter mitochondrial shape.
- The long non-coding RNA Pvt1 is a sponge for miR-27a.
- miR-208b regulates Sox6; miR-214 regulates both Sox6 and Slc16a3.

Abstract

Non-coding RNAs represent the largest part of transcribed mammalian genomes and prevalently exert regulatory functions. Long non-coding RNAs (lncRNAs) and microRNAs (miRNAs) can modulate the activity of each other. Skeletal muscle is the most abundant tissue in mammals. It is composed of different cell types with myofibers that represent the smallest complete contractile system. Considering that lncRNAs and miRNAs are more cell type-specific than coding RNAs, to understand their function it is imperative to evaluate their expression and action within single myofibers. In this database, we collected gene expression data for coding and non-coding genes in single myofibers and used them to produce interaction networks based on expression correlations. Since biological pathways are more informative than networks based on gene expression correlation, to understand how altered genes participate in the studied phenotype, we integrated KEGG pathways with miRNAs and lncRNAs. The database also integrates single nucleus gene expression data on skeletal muscle in different patho-physiological conditions. We demonstrated that these networks can serve as a framework from which to dissect new miRNA and lncRNA functions to experimentally validate. Some interactions included in the database have been previously experimentally validated using high throughput methods. These can be the basis for further functional studies. Using database

information, we demonstrate the involvement of miR-149, -214 and let-7e in mitochondria shaping; the ability of the lncRNA Pvt1 to mitigate the action of miR-27a via sponging; and the regulatory activity of miR-214 on Sox6 and Slc16a3. The MyoData is available at https://myodata.bio.unipd.it.



Graphical abstract

1. Introduction

Skeletal muscle is one of the most abundant organs in mammals as it accounts for 40–45% of the total body mass of healthy individuals. It is involved in body movement, metabolism, and protection of internal organs. Skeletal muscle is composed of different types of cells (neurons, blood cells, endothelial cells, etc.) [1] mixed with contractile myofibers, which are the tissue's parenchymal cells and exert the previously mentioned functions. Myofibers are large, multinucleated cells that are enwrapped by connective tissue to form fasciculi [2]. Skeletal muscles from different parts of the body have distinct physiological characteristics, such as in their metabolism, contractility, elasticity, and resistance to fatigue. Distinct physiological tracts of muscles reflect specific biochemical traits of myofibers that compose each muscle. Myofiber types are plastic and respond to specific stimuli by changing their traits and thus altering the physiology of the entire muscle to which they belong. Myofibers are canonically distinguished according to the expression of the different isoforms of the myosin heavy chain (MyHC). In humans, the identified myofibers include type 1 myofibers, which are mitochondria-rich and rely on oxidative metabolism; type 2a fibers, with oxidative fast-twitch characteristics; and the glycolytic

type 2x fibers [3]. In addition to the aforementioned fibers, mice have type 2b myofibers that are glycolytic fast-twitch myofibers [4]. Due to the plasticity of skeletal muscle, myofibers with mixed MyHC isoforms are also present (type 2a2x or 2x2b myofibers).

Aside from classifying myofibers by MyHC isoform content, a novel myofiber classification based on single-myofiber transcriptomic profiles was recently proposed that identifies specific transcriptional biomarkers for each myofiber type [5]. This method classifies myofibers as transcriptional slow (tS) and transcriptional intermediate (tI) with oxidative metabolism, and transcriptional fast (tF) with glycolytic metabolism. Transcriptional classification of myofibers appears to be more suitable to identify fibers in dynamic transition between different phenotypes.

Several non-coding RNAs, such as microRNAs (miRNAs), are involved in the specification of numerous muscle functions comprising development [6], pathology [7], and myofiber metabolism [5]. Not only do miRNAs participate in the regulation of muscle functions, but also long non-coding RNAs (lncRNAs) [8–11]. For example, we demonstrated that lncRNAs differentially expressed in slow and fast contracting myofibers regulate myofiber metabolism [12].

Complex cellular composition, fiber diversity, and dynamic changes of fiber phenotype imply that expression patterns at the single-cell level should be used to really understand the molecular bases of skeletal muscle regulation. This level of investigation is particularly important when dealing with non-coding RNAs because

this class of regulative molecules shows a stronger cell type-specific expression than coding RNAs [13–17]. Furthermore, it should be noted that in any differentiated cell, non-coding and coding RNAs form an intricate cross-talking network of interactions to regulate the actual gene expression patterns. As a result of these interactions, miRNAs regulate coding RNAs through post-transcriptional mechanisms [18], and lncRNAs, in turn, regulate the expression of coding RNAs [19] but also miRNA function by sponging them [20].

In this work, to better understand the molecular mechanisms involved in the functional specification of the different myofiber types, we integrated gene expression data of coding and non-coding RNAs to produce comprehensive lncRNAs-miRNAs-mRNAs interaction networks. Recently, different techniques have been developed to analyze gene expression at single-cell or single-nucleus level [21] permitting us to distinguish, at an unprecedented scale of analysis, not only how many differentially committed cells populate complex tissues but also how individual cells are affected and respond to different physio-pathological conditions [22,23]. One limitation of this type of analysis is that they allow the detection of only polyadenylated RNAs, excluding from the analysis non-polyadenylated mature miRNAs. To overcome this problem, we integrated available single nucleus RNA-seq (snRNA-seq) analyses on skeletal muscle tissue with our previously determined networks describing single myofibers gene interactions. Gene networks based on expression correlations are known to produce inferred interactions that result as false positives after experimental

validation. This approach is also less manageable and intuitive than the building of networks based on manually curated pathways. On the other hand, manually curated pathways do not consider the regulative action of miRNAs and lncRNAs. We introduced gene expression regulation based on non-coding RNAs in KEGG pathways to allow for a better description of specific changes in different myofiber types or in different studies based on snRNA-seq. We experimentally confirmed some interactions identified in our database showing the involvement of specific miRNAs in the regulation of the mitochondrial network. Moreover, we confirmed the activity of some lncRNAs as miRNA sponges and the role of some miRNAs in the regulation of genes that are known markers of myofiber specificity.

2. Material and Methods

2.1. Gene expression data and processing

Single myofiber gene expression data were collected from Gene Expression Omnibus (GEO) and Sequence Read Archive (SRA) databases using the following IDs: GSE98328, SRX2768351, SRX2768352, SRX2768353 [5], and GSE112716 [12]. For snRNA-seq, we used processed data retrieved from [24–26] as an example of muscle pathology, fiber typing, and ageing respectively. Microarray gene expression data were processed as follow. Agilent microarray mouse platform was re-annotated (Gencode annotation release vM22, evidence-based annotation of the mouse genome GRCm38, version M22 Ensembl 97) both for coding and non-coding RNAs. Microarray data were normalized using quantile normalization separately for protein-coding and long non-coding genes. The dataset includes 10 biological replicates for each myofiber type considered (1, 2A, 2A/2X, 2X, 2X/2B, and 2B). Myofibers were sub-grouped in transcriptional slow (type 1), transcriptional intermediate (type 2A, 2A/2X, and 2X), and transcriptional fast (type 2X/2B and 2B). RNA sequencing data for miRNA identification were mapped to the known mouse miRNA precursors from the miRBase

database (Ver. 19) using the mapper module of miRDeep with default settings. Quantize module was used to normalize read counts of mature miRNAs.

2.2. Gene expression correlation

We computed the level of expression correlation among different RNA categories using the Spearman index as follows: mRNA – lncRNA; mRNA – miRNA; lncRNA – miRNA.

Correlations were obtained using the *'cor'* function provided by the 'stats' library of the R language. All correlations were filtered based on specific thresholds: for the mRNA – lncRNA comparisons we required a correlation greater or equal to 0.45; for the miRNA – mRNA and miRNA – lncRNA comparisons, we selected correlations below −0.35. Furthermore, a permutational test was implemented to assess statistical significance: we computed an empirical p-value using 1,000 random permutations of the experimental measures.

2.3. Interactions between miRNAs and mRNAs and miRNAs and lncRNAs

We collected validated and predicted interactions from multiple sources. Specifically:

- miRNA – mRNA validated interactions were downloaded from TarBase v7.0 [27,28] and the Encyclopedia of RNA Interactomes [29], [30] (ENCORI: HITS-CLIP validation, data downloaded November 27, 2020)
- miRNA – mRNA predicted interactions were extracted from miRDB (v6.0) [31,32], miRmap (version of 10-Jan-2013) [33,34], RNA22 [35] (full sets of prediction of Mus musculus based on Ensembl 96, miRBase 22 and RNA22v2), PITA [36] (both files with zero flank and with a flank of 3 and 15 bases upstream and downstream)
- miRNA – lncRNA validated interactions were downloaded from DIANA-tool (LncBased v.2) [37] and ENCORI [29,30] (HITS-CLIP validation, data downloaded November 27, 2020).

All interactions were further filtered based on correlation results, using the same thresholds described in the previous section.

2.4. Functional circuits

We used the collected interactions to identify minimal functional circuits, defined as groups of three interacting nodes: one mRNA, one lncRNA, and one miRNA. We found a total of 9,625,735 circuits, divided as follows: 9,502 including validated interactions and 9,616,233 containing predicted interactions.

2.5. Node-Centric network

Each web page describing an mRNA, miRNA, or lncRNA displays a small network representing a selection of the functional circuits involving searched entry. As the complete network would be too large to be practically displayed, we designed a heuristic approach to identify the most relevant interactions to be included.

We collect all edges belonging to functional circuits and for each, we compute two weights as follows:

- A weight 'w' defined as the p-value of the correlation between the two endpoints of that edge.
- A weight 'wpg' (named after the fact that it will be later used to compute the PageRank importance of each node) defined as follows:

For edges obtained from circuits including predictions:

$$wpg = 1 - w$$

For edges obtained from circuits including validated interactions:

$$wpg = 1 - \frac{w}{sf}$$

where sf is the ratio between the number of edges coming from circuits obtained from predicted and from validated interactions.

The scale factor sf was devised to balance the relative importance of circuits including predicted and validated interactions. Indeed, the former are much more numerous

than the latter; if unchecked, this imbalance would risk obscuring almost completely the contribution of validated results in the final network.

Overall, this master network derived from function circuits includes 17,886 nodes and 1,243,206 edges.

The most relevant network centered at each node is then computed using the following procedure:

1) Starting from a node of interest n, we find the subgraph induced by its neighbors within a distance of two steps in the master network.
2) We compute the PageRank of each node using the '*wpg*' weights, and we select the top 30 nodes according to this metric. We balance types of nodes in such a list: in other terms, we try to collect 10 mRNAs, 10 miRNAs, and 10 lncRNAs to provide an even representation of the different RNA species.
3) We collect circuits involving the nodes identified in the previous step giving priority to validated interactions. This step is repeated until there are no more isolated nodes
4) Step #3 does not guarantee, by itself, that the resulting network will consist of a single connected component. Since that is our final objective, we apply the following transformation until multiple components remain:
    a) We pick the smallest and the largest components.
    b) We identify the two nodes with the highest PageRank inside those.

c) We link the nodes together by adding the edges along the shortest path connecting them to the network obtained in step #1

d) We add one extra node for each edge along the shortest path, in order to capture, if existing, the functional circuits having such edges as one of their sides. This step is guided by a global optimization procedure aimed at reducing the total number of nodes that have to be introduced.

2.6. Custom network from user selection

The Custom Network section gives the user the option to provide a list of up to 30 nodes (mRNAs, miRNAs, or lncRNAs). Our system will then generate a network representing the most relevant circuits including the nodes in the user selection. The procedure we use to build this network is similar to the one developed for the single nodes, but we employed some specific optimizations to obtain a solution in real time:

1) First of all, we keep in memory the master network, the PageRank score of all the nodes and the corresponding minimum spanning tree (MST).
2) Instead of starting from the collection of circuits, we directly compute the induced subgraph defined by the user selection.
3) If multiple connected components remain, we link them by extending the network to include the shortest path identified on the MST among the highest-scoring PageRanked nodes.

Because of graphical constraints, we limit the total number of nodes in the resulting networks to 150.

2.7. Single-nucleus network

We integrated snRNA-seq data into our network, starting from nucleus-type specific clusters obtained from [24–26]. We collected all the genes belonging to each cluster identified in single myofibers: myonuclei (type 1, 2A, 2X, 2B, Nr4a3+, Enah+, Ampd3 +), nuclei from satellite cells, neuromuscular junction, myotendinous junction, and myocytes. We built type-specific networks filtering functional circuits given their overlap with each group of genes and using the same procedure described in paragraph 2.5 (Node-Centric Network) to reorganize the network.

Clusters from [24] contain two nucleus categories: wild type (WT) or delta exon 51 (DEx51) of the gene encoding for dystrophin. In this case, we extended the filtering procedure to keep the circuits identified in the two subgroups separated.

2.8. Pathway construction

The topologies of all KEGG pathways were retrieved from the graphite package [38]. Each network was then extended to include predicted and validated interactions involving miRNAs or lncRNAs.

Starting from a set of nodes provided by the user (the query), we perform a series of hypergeometric tests to find the list of pathways significantly overlapping such query. To this end, we use the 'hypergeom.sf' implementation provided by the 'scipy' library [39] and we corrected results using the 'fdrcorrection' (Benjamini-Hochberg method) provided by the 'statsmodel' library [40].

Results of pathway enrichment analyses are displayed in a table. Each entry is linked to a detailed view of the corresponding pathway showing the following information: the nodes in common with the user query and the most relevant circuits involving protein-coding genes, lncRNAs, and miRNAs, that overlap the pathway.

2.9. Software implementation

All the information about expressions, correlations, and networks (nodes and topologies) is stored in a RocksDB database and accessed through the python-rocksdb library [41].

Network algorithms to compute shortest paths, minimum spanning trees, and PageRanks are implemented by the network library [42]. The web interface was built in JavaScript on top of React [43] and fontawesome icons [44]. We relied on React-Apexcharts [45] for the display of expression plots and on React-Table [46] for the generation of dynamic tables.

Finally, we extended Cytoscape-JS [47] and the layout engine cytoscape-cola [48] to render networks.

2.10. Primer design

Primers to amplify the genomic region containing miRNA genes and primers for qRT-PCR analyses were designed using the Primer3Plus algorithm (http://www.bioinformatics.nl/cgi-bin/primer3plus/primer3plus.cgi, Accessed on 18th of July 2021) and analyzed for dimers and secondary structure formation with OligoAnalyzer tool (Integrated DNA Technologies). Moreover, primers were tested using the in-silico PCR tool implemented in the UCSC Genome Browser. Primer sequences were reported in the Supplementary Table S1.

2.11. miRNA cloning

DNA regions coding for selected miRNAs were cloned in the pCMV-MiR vector (OriGene) including 200–300 bases upstream and downstream the pre-miRNA sequence.

2.11.1. PCR for inserts preparation

Genomic DNA extracted from C2C12 cells was used as template for the amplification of selected miRNA genes. PCR reaction mix was prepared as following: H_2O, 34.25 µl;

PCR Buffer 10X, 5 µl; $MgCl_2$[50 mM], 4 µl; dNTPs [10 mM], 3 µl; Primer Forward [10 µM], 1 µl; Primer Reverse [10 µM], 1 µl; Taq DNA Polymerase [2 U/µl], 0.75 µl; DNA [50 ng/µl], 1 µl. PCR amplification was done in an Eppendorf thermocycler using the following program: 5 min 95° C; (30 sec 95° C; 30 sec 58-61° C; 70 sec 72° C for 45 cycles); 10 min 72° C. Amplification was verified in 1.5% agarose gel and PCR products were purified with the GenEluteTM PCR Clean-Up Kit (Sigma-Aldrich) following the manufacturer protocol.

2.11.2. Plasmid and insert digestion, ligation and bacteria transformation

pCMV-MiR vector (Origene) and PCR products were digested with the same restriction enzymes in order to perform directional cloning (AscI and XhoI; New England BioLabs). Depending on the position of restriction enzymes in forward or reverse amplification primers, we were able to clone the amplicon to allow the expression of miRNA or miRNA antisense sequences (Supplementary Table S1). Restriction reactions were performed at 37° C for 90 min in the following reaction mix: H_2O to 50 µl; Cut Smart Buffer 10x, 1 µl; AscI [10 U/µl], 1 µl; XhoI [10 U/µl], 1 µl; Plasmid/PCR Insert, 1 µg. Digestion products were purified using the GenEluteTM PCR Clean-Up Kit (Sigma-Aldrich) following the manufacturer protocol.

50 ng of plasmid were used to ligate 1:4 M quantities of PCR product as follow: H_2O to 20 µl; T4 Ligation Buffer 10X, 2 µl; PCR amplicon 4 M with respect to the 50 ng of the vector; digested pCMV-MiR vector 50 ng; T4 DNA ligase [10 U/µl], 1 µl. The solution was incubated at 16° C overnight and then precipitated using sodium acetate and ethanol. Pellet was resuspended in 5 µl of H_2O RNAse free.

1 µl of ligation product and 40 µl of electro-competent bacteria (Escherichia coli bacteria DH10B) were mixed and the solution was subjected to an electrical field of 1.8 kV in a Gene Pulser II electroporator (BioRad). Then, 360 µl of SOC medium were added and after that, the bacterial solution was incubated at 37 °C for 1 h. Bacteria were plated on solid LB medium (LB + Agar) with kanamycin [50 µg/ml] and grown at 37° C overnight. Colony PCRs were performed in order to test the presence of the insert in the plasmid using 3 µl of a liquid bacterial culture as template. The PCR products were visualized in 2% agarose gel. For each plasmid, 5 µl of one of the positive colonies were regrown in 5 ml of LB + kanamycin medium at 37° C overnight to prepare the purified plasmid. The plasmid was extracted and purified using a PureLink HiPure Plasmid Miniprep kit (Invitrogen). To test the accuracy of the pre-miRNA sequences, all plasmids have been sequenced (Sanger Sequencing, Eurofins) and compared with the mouse reference genomic sequences derived by the UCSC Genome Browser.

2.12. C2C12 culture and cell transfection

C2C12 myoblasts were cultured on Tissue Culture dishes (Thermo Fisher Scientific) in proliferation medium (Dulbecco's modified Eagle's medium (DMEM), 10% fetal bovine serum, 1 U/ml Penicillin, 100 μg/ml Streptomycin) until reaching 80% of confluence. After cell detaching with Trypsin-ethylenediaminetetraacetic acid (Thermo Fisher Scientific) 40,000 or 60,000 cells were plated on each well of Multiwell Culture plates (Thermo Fisher Scientific) using medium without antibiotic. A sterile 13 mm round coverslip was positioned on the bottom of the wells before cell seeding. Cells were co-transfected with mitoRFP and pCMV-MiR (with cloned a specific miRNA or miRNA antisense) using the Lipofectamine 2000 (Thermo Fisher Scientific) as the transfecting agent. Transfection solution was prepared by combining and incubating two solutions at room temperature for 30 min, which contained: (solution 1) 3 μl of Lipofectamine 2000, 122 μl of Opti-MEM (Thermo Fisher Scientific); (solution 2) 2 μl of mito-RFP plasmid [100 ng/ul], 2 μl of pCMV-MiR with cloned miRNA or antisense [100 ng/ul], 121 μl of Opti-MEM (Thermo Fisher Scientific). Cells with the transfection solution were grown for 24 h at 37 °C in 5% CO_2 in a humidified incubator. After 24 h the medium was changed with a new medium containing G418 antibiotic [0.5 mg/ml] for 4 days. G418 antibiotic was used to positively select cells transfected with pCMV-MiR.

Pvt1 silencing was performed using antisense LNA GapmeRs (Exiqon) (Pvt1 1 ACCGTAGTAGAGTTAA; Pvt1 3 AGTCAACGCTTCACAT). Cells transfected with Lipofectamine 2000 and Antisense LNA GapmeR Negative Controls (Exiqon) were used as negative controls.

2.13. Mitochondrial network analysis

Survived cells to G418 selection were used to evaluate mitochondrial network. In fact, mitoRFP plasmid encodes for a fluorescent tag localized in the mitochondria, which is characterized by an excitation wavelength of 555 nm and an emission wavelength of 584 nm. After G418 selection, the culture medium was removed, and a first wash was carried out with 500 µl of phosphate-buffered saline (PBS). Cells were then fixed by adding 500 µl of 4% paraformaldehyde in PBS and incubated at room temperature for 15 min. Then, three washes with PBS were performed, slides on the bottom of the wells were recovered, rinsed in distilled H_2O, and mounted on glass slide. Slides have been observed through a confocal microscope, oil immersion objectives (63x of magnification), and exciting samples with a wavelength of 555 nm. Z-stack images of samples have been acquired and used for subsequent analyses to determine the degree of mitochondrial fragmentation.

The images were analyzed with the ImageJ software, using the MitoLoc plug-in [49]. To describe mitochondrial morphology, we used the fragmentation index (F.I.)

calculated as follows: $V_S = (V_{fragment} \,/\, V_{total}) \cdot 100$ and $F.I. = (\sum_1^X Vs \leq 20\%) \,/\, (\sum_1^X Vs)$.

2.14. Electron microscopy

Transfected C2C12 cells were fixed with 2.5% glutaraldehyde in 0.1 M sodium cacodylate buffer pH 7.4 for 1 h at 4° C, post-fixed with 1% osmium tetroxide and 1% in 0.1 M sodium cacodylate buffer for 2 h at 4° C. Samples were washed three times with water and then dehydrated in a graded ethanol series and embedded in an epoxy resin (Sigma-Aldrich). Ultrathin sections (60–70 nm) were obtained with an Ultro-tome V (LKB) ultramicrotome, counterstained with uranyl acetate and lead citrate, and viewed with a Tecnai G2 (FEI) transmission electron microscope operating at 100 kV. Images were captured with a Veleta (Olympus Soft Imaging System) digital camera.

2.15. Overexpression of miR-27a in mouse skeletal muscle

miR-27a was overexpressed in mouse muscles as described in [5].

2.16. RNA extraction and qRT-PCR analysis

Trizol (Thermo Fisher Scientific) was used to extract total RNA from C2C12 cells or skeletal muscles according to the manufacturer protocol. Briefly, 500 µl of Trizol (Thermo Fisher Scientific) per well of Multiwell Culture plates or 1 ml per 30 mg of

muscle were used. 1 vol of chloroform to 5 volumes of Trizol were added and vigorously mixed; then the solution was kept on ice for 15 min and then centrifuged at 4 °C at 12,000 rpm for 20 min. The upper aqueous phase was transferred in a new Eppendorf tube and RNA was precipitated using 1:1 vol of isopropanol. RNA was resuspended in H_2O RNAse free and tested for protein and phenol contaminations at the spectrophotometer. RNA integrity was tested with the 2100 Agilent Bioanalyzer.

RNA with RIN > 7 was used for the retrotranscription according to the following protocol. 1–3 μg of total RNA were mixed with 1 μl of oligod(T) [50 μM], 0.5 ul of random primers [20 μM], 1 μl of dNTPs and H_2O to bring the volume to 13 μl. The solution was heated to 65° C for 5 min and then killed on ice for 2 min. 4 μl of first-strand buffer 10X, 2 μl of DTT [0.1 M] and 1 μl of Superscript II (Thermo Fisher Scientific) were added to the previous solution and all incubated at 42° C for 2 h. Superscript II was inactivated incubating the mix at 70° C for 15 min.

EvaGreen molecule (Solis BioDyne) was used to perform qRT-PCR in the CFX thermocycler (BioRad) using the following PCR cycle: 15 min 95° C, (15 sec 95° C, 20 sec 60° C, 45 sec 72° C with the fluorescence reading, and 40 cycles), 3 min 72° C. Reaction mix was 6.6 μl of H_2O, 2 μl of Master Mix 5X, 0.2 μl of primer forward [10 μM], 0.2 μl of primer reverse [10 μM], 1 μl of cDNA [10 ng/μl].

miRNA analysis was performed using the TaqMan miRNA assays (Thermo Fisher Scientific). 10 ng of total RNA were used to retrotranscribe specific miRNAs and the U6 reference gene using the miRNA reverse transcription kit (Thermo Fisher

Scientific). Real-time PCR was performed on CFX thermocycler (BioRad) using the TaqMan Universal PCR Master Mix II, no UNG (Thermo Fisher Scientific) according to the manufacturer's protocol.

2.17. Luciferase assay

Myoblasts were transfected with pCMV-MiR vector containing the sequence for miR-27a or −214 and 100 pg/ml of pmirGLO Dual-Luciferase miRNA Target Expression Vector (Promega) containing the target sequence or a control sequence (primers for cloning are listed in Supplementary Table S1. Cloning was performed using SacI and XbaI restriction enzymes). Assays were performed using the Dual-Luciferase Reporter Assay (Promega), measuring firefly and renilla luciferase activities with Turner Designs TD-20/20 Luminometer (DLReady). miRNA transfections were independently replicated at least three times.

3. Theory and calculation

MyoData includes experimental data on gene expression on single myofibers and nucleus to calculate networks centered on each mRNA, miRNA, or lncRNA whose expression was measured. These networks are computed considering the fact that i) miRNAs induce the degradation of their targets and ii) lncRNAs may function as miRNA sponges. Therefore, interactions recorded in different databases among miRNAs and lncRNAs, and miRNAs and mRNAs were further filtered using the correlation among their expression profiles. Specifically, we require that miRNAs and lncRNAs, and miRNAs and mRNAs show negatively correlated expression patterns. On the contrary, the expression correlation between mRNAs and lncRNAs should be positive.

We have developed a heuristic approach that strikes a balance between the overall number of nodes included in each network, and their relevance, defined on the basis of the strength of their interactions and the topological distance to the user query. Moreover, the procedure results in a balanced selection over the three categories of nodes (mRNAs, miRNAs, and lncRNAs).

4. Results and discussion

4.1. MyoData resource

MyoData collects expression profiles of mRNAs, miRNAs, and lncRNAs in different myofibers and gives the user information to hypothesize their function in relationship with physio- and patho-logical differences.

The database has three main search functions:

1) The user can focus on a specific mRNA, miRNA, or lncRNA.
2) Given a list of genes, the software can extract the network containing their interactions. As described in "Materials and Methods", we limited to 30 the number of acceptable genes in order to compute the network in real-time and to display it graphically with an acceptable level of resolution.
3) As an alternative, a gene list can be used to perform a pathway enrichment analysis. Here, we employ KEGG pathways which we extended to include miRNAs and lncRNAs. These genes are usually absent in pathways but may nonetheless influence gene expression.

In all cases, MyoData accepts as an input gene symbols or ENSEMBL Gene identifiers (for mRNAs or lncRNAs) and miRNAs name (Fig. 1).

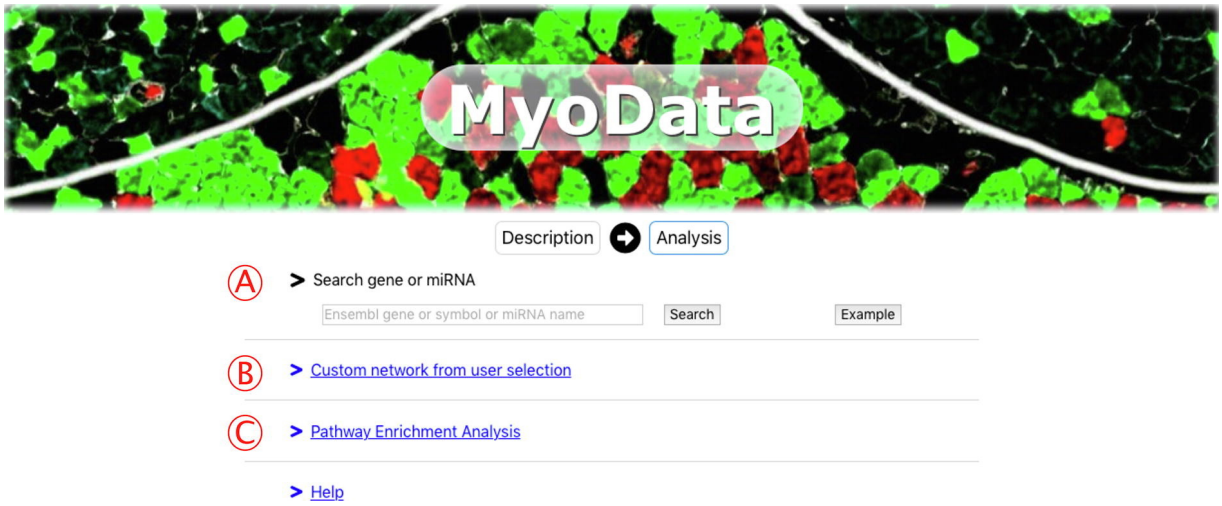


Fig. 1. Search functions in MyoData. (A) Users can search for a single mRNA, miRNA or lncRNA. Alternatively, (B) a list of up to 30 genes can be queried to generate a network or (C) a pathway enriched with miRNA and lncRNA functions.

4.1.1. Search for an entry: Retrieve expression on single myofibers, regulatory network centered on it

This page shows details about a single node: mRNA, miRNA, or lncRNA. It is subdivided into three sections:

1) A bar plot representing the expression values over all the available single myofibers which were experimentally assayed.
2) A network view, collecting the most relevant interactions.
3) Correlation tables.

In the first section, an interactive bar plot is shown, where each expression measure is colored according to the type of myofiber it belongs to. The website also offers the possibility to download all expression tables in three different formats: svg (vector graphics), png (raster graphics), and csv (textual) (Fig. 2A).

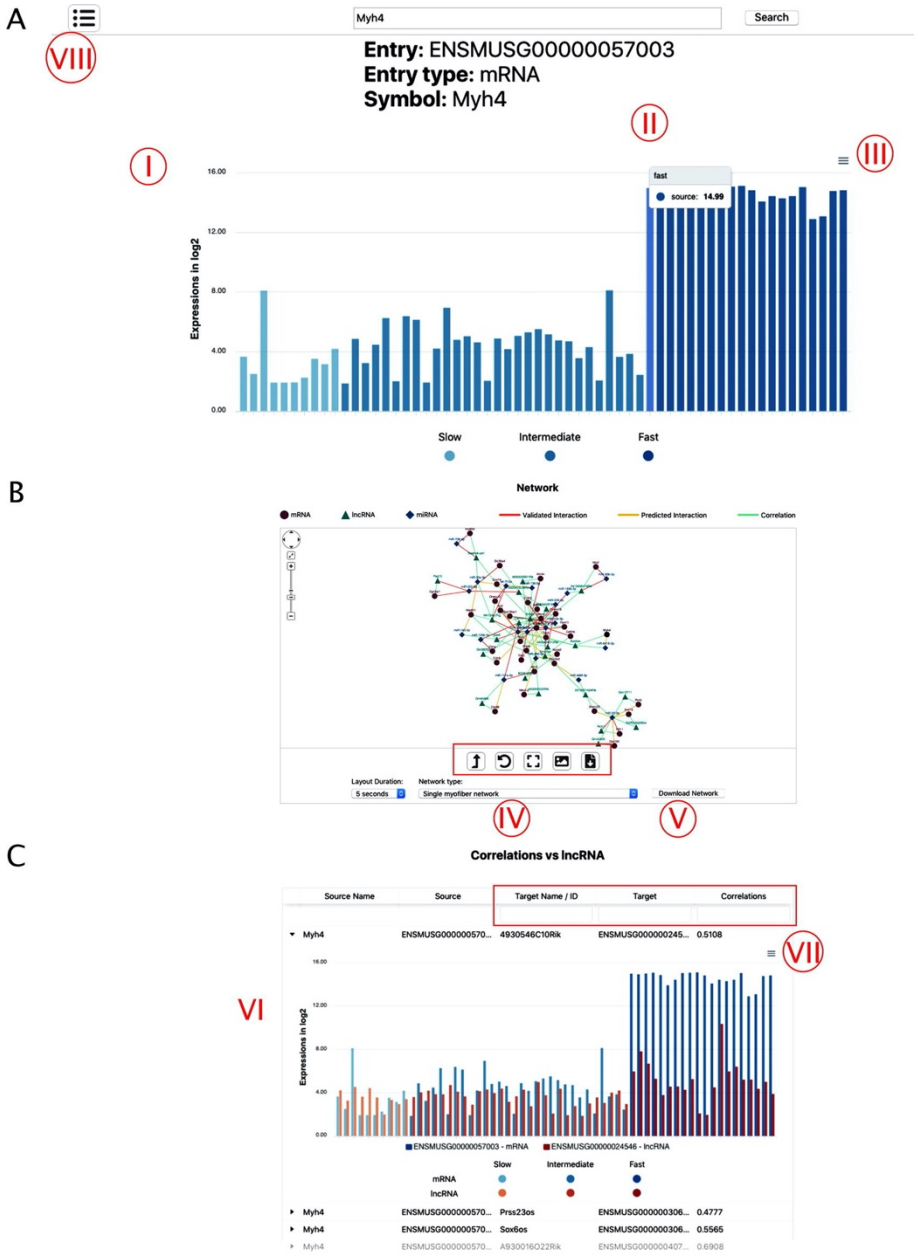


Fig. 2. Search for a single entry. (A) Bar plot showing expression values for each biological replicate using Myh4 as an example of entry. Different gradations of blue indicate different myofiber types (I). By moving the mouse over each bar, the precise expression value appears (II). Expression tables can be downloaded (III). (B) Network visualization using Myh4 as an example of entry. Query is colored in black. Red rectangle indicates buttons to manage the network. The network can be filtered according to single nucleus RNA-seq data (IV) and can be downloaded as a table (V). (C) Correlation description using Myh4 as an example of entry. The red rectangle indicates boxes used for filtering. By clicking on the arrow next each source name, a histogram appears describing the expression of correlated genes for each sample (VI). The image can be downloaded (VII). It is possible to move to different pages by clicking the indicated button (VIII). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The second section of the page displays a network collecting the interactions relevant to single myofiber types or single-nucleus cluster, which can be selected through a drop-down menu (Fig. 2B).

The layout is calculated in real-time, and the user has the option to limit the total number of seconds dedicated to this task. Moreover, different buttons give the user the possibility to further improve the layout, to reset the viewport (by centering and rescaling the network to fit the available space), to save a PNG image, or to export a tsv, or JSON file describing the network that can be later loaded into the stand-alone Cytoscape software for further analyses [50].

Network visualizations are completely interactive. By clicking on any node, its details are shown in a separate panel. Information presented includes node descriptions, Gene Ontology annotations, and external references. Similarly, edges are annotated with their correlation index, the name of the database from which they were derived the type of interactions they represent.

The third section of the page consists of a series of tables collecting the correlations computed between the selected entry and the other nodes in the database belonging to different RNA species. The user has the option to further filter the tables by searching for specific need IDs (miRBase IDs coming from miRBase v22 GRCm38). Partial matches are automatically handled: for instance, the substring “let-7a” would automatically match the full form “mmu-let-7a”. Each table row can be dynamically

expanded, by clicking on a button, to display graphically the expression profiles of the two correlated nodes (Fig. 2C).

4.1.2. Regulatory network from multiple entries searching

This page gives the user the ability to use a list of gene identifiers as a query. As described in the "Materials and Methods" section, we filter out those entries that are not present in our master network. The user has the option to display the list of such rejected IDs (Fig. 3A).

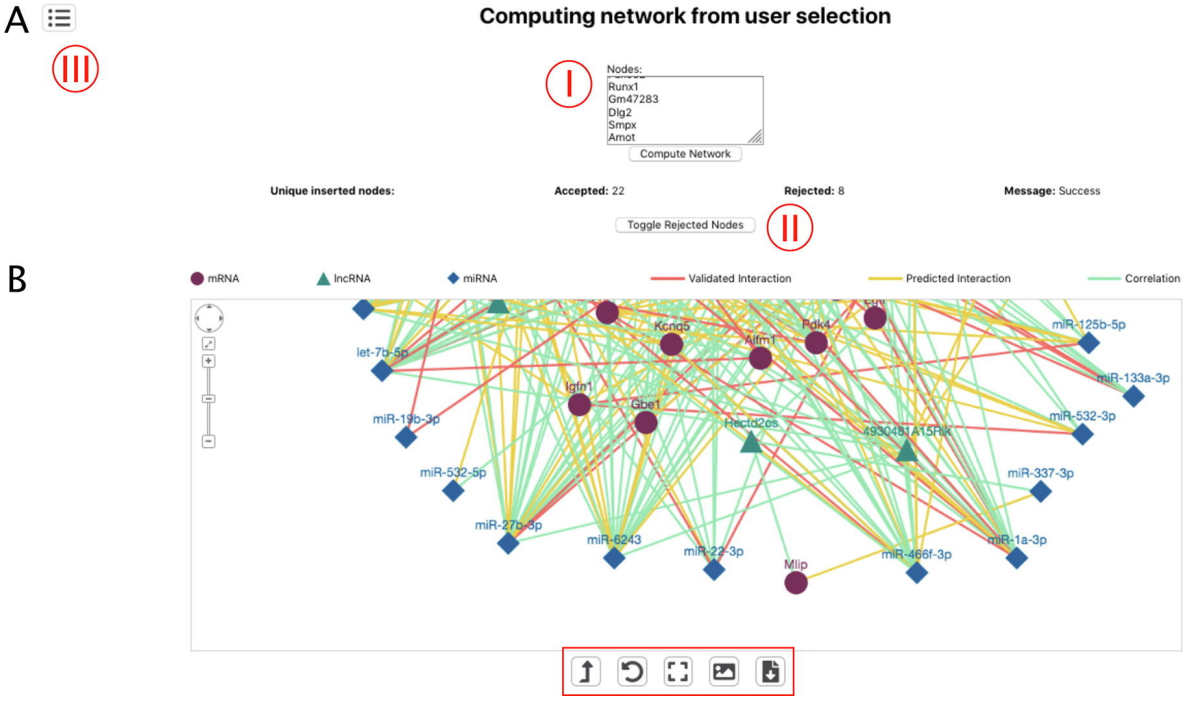


Fig. 3. Interaction network from a list of genes. (A) Page structure for a multiple query. A list of up to 30 genes can be pasted in the box (I). After clicking on the "Compute Network" button the network will be calculated. Rejected genes can be visualized by clicking on the "Toggle rejected nodes" button (II). (B) Resulted network. The red rectangle indicates buttons to manage the network. It is possible to move to different pages by clicking the indicated button (III). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

After filtering, a network is computed and displayed. The graphical format is similar to that described in the previous section, with the only difference that the layout computation can be performed for longer periods of time (up to 30 s; Fig. 3B).

4.1.3. Pathway enrichment analysis

MyoData is implemented to perform a pathway enrichment analysis. This will result in the display of a table including the titles of significantly enriched pathways, their dimension, the size of the intersections with user-provided nodes, and finally the adjusted p-values for the statistical tests.

By clicking on each pathway title, MyoData will switch to the visualization of the pathway topology, extended with the most important functional circuits (miRNA-lncRNA-mRNA interactions; Fig. 4A and B).

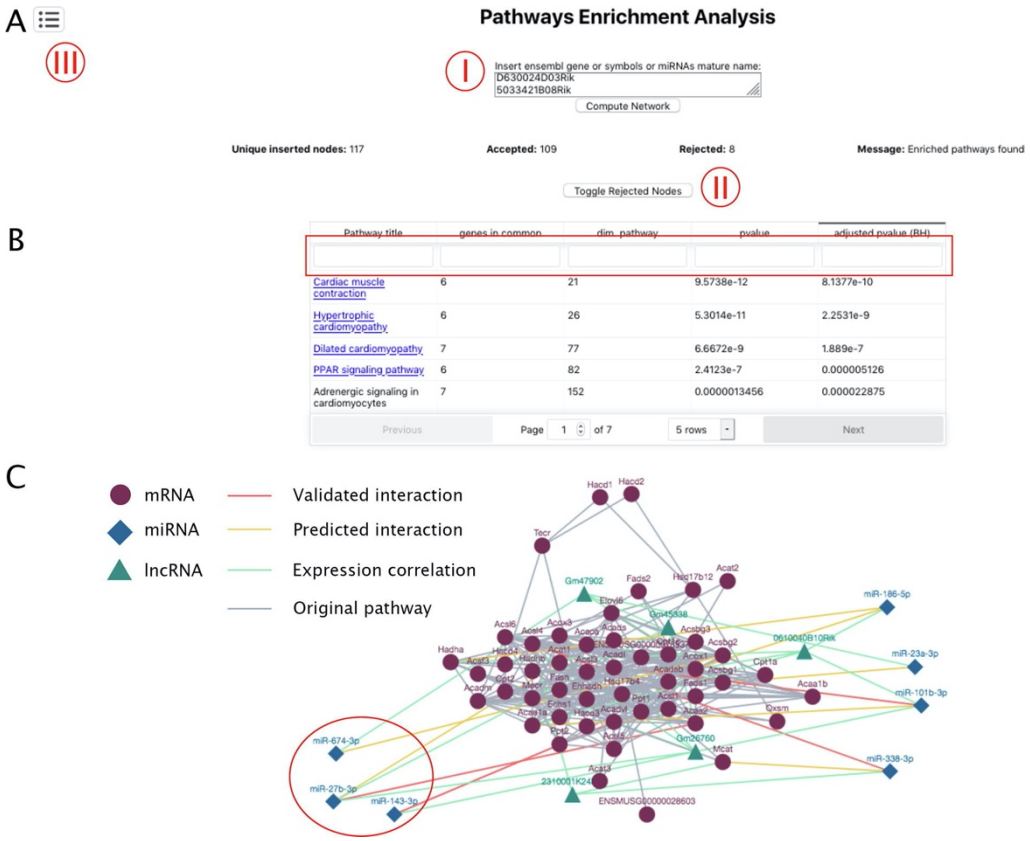


Fig. 4. Pathways enrichment analysis. (A) Query page. In the box (I) the user can paste gene symbols and by clicking on the "Compute Network" button pathways the enrichment will be calculated. Rejected genes can be visualized by clicking on the "Toggle Rejected Nodes" button (II). (B) Results appear in a table that can be filtered according to the name of the pathway (Pathway title), number of genes identified in the pathway (genes in common), the dimension of the pathway (dim. Pathway), and statistics (pvalue and adjusted pvalue) (red rectangle). It is possible to move to different pages by clicking the indicated button (III). (C) Fatty acids metabolism pathway extended with non-coding RNAs involved in the regulation of the considered genes. Red circle indicates miRNAs discussed in the text. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

As an example, in Fig. 4 we used single nucleus RNA-seq results from a previous study [25]. We used genes significantly upregulated in the cluster of nuclei specific for the slow contracting myofibers (type 1 or myosin heavy chain 7; Myh7). Most enriched pathways correctly describe heart functions and the Ppar pathway (Fig. 4B). It is known that slow myofibers have isoforms of contractile proteins similar to heart, and peroxisome proliferator-activated receptor δ (PPARδ) induces a switch to form

increased numbers of type 1 myofibers [51]. Prevalently, the metabolism of type 1 myofibers is based on lipid oxidation [5] and the "Fatty acids metabolism" is one of the most enriched pathways (Supplementary Table S2). Interestingly in the pathway corresponding to "Fatty acid metabolism" among other miRNAs we identified miR-27b that is considered a hub in the lipid metabolism [52], miR-674, which is associated with circulating lipids [53], and miR-143, which is already known to regulate lipid metabolism [54] (Fig. 4C).

4.2. Data validations

To demonstrate the potentiality and the validity of data extracted from the MyoData resource, we performed four case studies focused on important aspects of the skeletal muscle physiology: i) the modulation of mitochondrial shape by miRNAs that may impact muscle metabolism; ii) the action of lncRNAs as miRNA sponges; iii) the co-participation of different non-coding RNAs in the regulation of myofiber functions; iv) the improvement of snRNA-seq information.

4.2.1. Case study 1: Identification of miRNAs impacting on the mitochondrial shape

MyoData outputs the expression of miRNAs in different myofiber types permitting users to hypothesize their function based on physiological differences of myofibers.

For example, by searching for miR-214, −142, −208b, −382, and let-7e in the MyoData, users will see that these miRNAs are not expressed in intermediate myofibers, the most oxidative ones [5]. The modulation of these miRNAs likely impacts the expression of proteins controlling metabolism in this type of muscle cells. As proof of principle, we tested this hypothesis by evaluating mitochondrial shape, which is a readily measurable phenotype and is important for skeletal muscle metabolism and functions [55–58]. In addition to the aforementioned miRNAs, we also included miR-301a, −29a, −143, −27a, −149, −378a, and let-7a because they target several genes coding for mitochondrial proteins (Supplementary Table S3). We tested the inhibition of miR-378a using antisense sequences since miR-378a knock-down was previously shown to induce the accumulation of abnormal mitochondria and apoptosis [59]. We confirmed that its inhibition indeed induced mitochondrial fragmentation, which is known to be a marker of apoptosis [60] (Fig. 5A). We obtained comparable results after the inhibition of miR-29a and let-7a confirming previous observations obtained in the heart [61] and HT29 cells [62]. However, the upregulation of miR-143, −382, −301a, and −208b did not change the conformation of the mitochondrial network (Fig. 5A). miR-208b is a miRNA highly expressed in slow myofibers and is involved in the specification of those types of myofibers via its blocking of Sox6 [63]. Slow oxidative myofibers are very rich in mitochondria, which correlates well with our experiments that show that the upregulation of miR-208b did not affect the mitochondrial network. miR-143 is particularly expressed in skeletal muscle and is associated with the

maintenance of the satellite cell population and with aging [64,65], similar to miR-382 [66]. Both miR-301a and −143 are upregulated in mice fed with high-fat diet [67] which impacts mitochondrial function but, according to our validation experiments, their upregulation did not affect mitochondrial conformation. The upregulation of the other tested miRNAs caused mitochondrial fission (miR-27a, −142, and let-7e) or fusion (miR-149, −214) (Fig. 5A). In summary, we confirmed that 8 out of 12 tested miRNAs altered mitochondrial shape, thereby potentially impacting the regulation of muscle metabolism. These results can be important starting points for researchers interested in studying the metabolic impact of tested miRNAs.

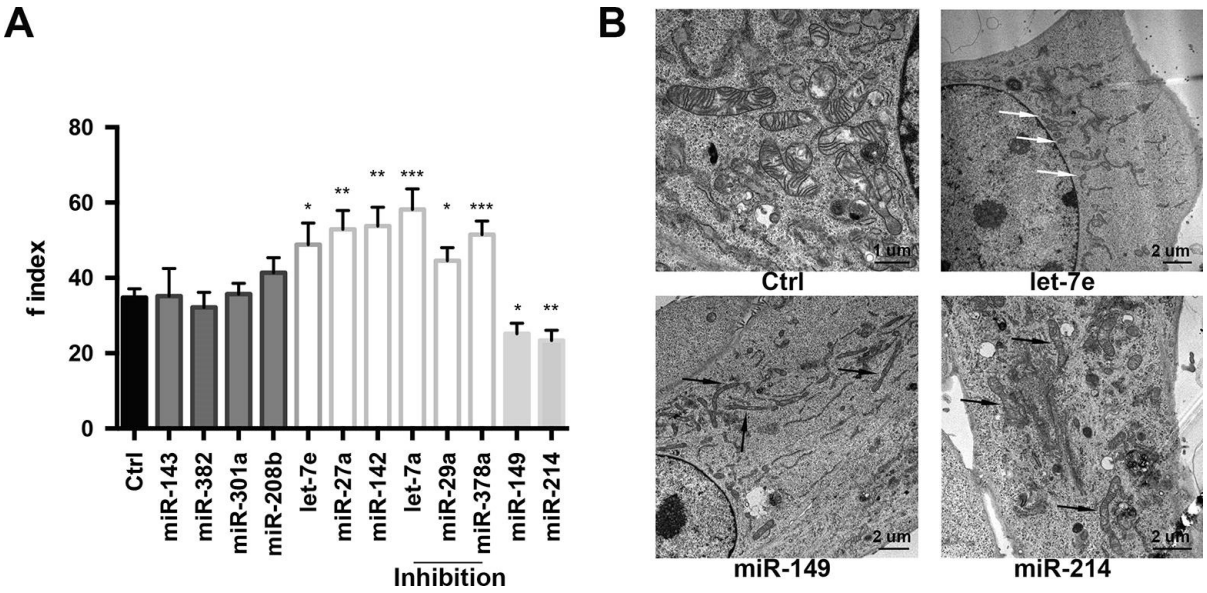


*Fig. 5. miRNAs regulate mitochondrial shape. (A) Quantification of the fragmentation index, f index, of mitochondrial networks after miRNA transfections; n = at least 20 mitochondria for each condition (three independent transfections per each miRNA). Dark grey bars represent f index associated with miRNAs not affecting mitochondrial shape; white bars represent f index associated with miRNAs that induce mitochondrial fragmentation; light grey bars represent f index associated with miRNAs that induce mitochondria fusion. Among miRNAs inducing mitochondrial fragmentation those that were inhibited are indicated. Significance was calculated using t-test between control and each treated sample considering unequal variance between samples. * P ≤ 0.05, ** P ≤ 0.005, *** P ≤ 0.0005. Indicated statistical significance is referred to the control (Ctrl). Error bars represent SEM. (B) Electron microscopy of C2C12 cells transfected with pCMV-MiR vectors to upregulate specific miRNAs. Black arrows indicate elongated mitochondria in cells overexpressing miR-149 and –214; white arrows indicate fragmented mitochondria in cells overexpressing let-7e.*

To confirm our previously described results, we also checked mitochondrial ultrastructure by electron microscopy. We previously showed the change in mitochondrial ultrastructure after the upregulation of miR-27a and −142 [5], therefore we tested let-7e, which according to the analysis of the f-index causes mitochondrial fission, and miR-149 and −214, which cause mitochondrial fusion, confirming in all cases previously described results (Fig. 5B).

4.2.2. Case study 2: lncRNA Pvt1 as a miRNA sponge

In the MyoData database, we integrated information on miRNA–mRNA and miRNA–lncRNA interactions. This allows for the identification of miRNA–mRNA–lncRNA network triangles that describe the miRNA sponge activity of lncRNAs. We used this information to experimentally validate the activity of the lncRNA plasmacytoma variant 1 (Pvt1) as a sponge for miR-27a. We previously demonstrated that Pvt1 is involved in muscle atrophy by regulating cMYC [12]. This is possible thanks to the cytoplasmic localization of Pvt1 [12] where it acts as a sponge for miR-200 family, miR-199a, −152, and −30a in different cancers [68–71].

The network associated with Pvt1 outputted from MyoData identifies Pvt1 as a central node regulating miR-101a, -22, -24, -26a, -27a, -322, and -532 (Fig. 6A). Network triangles Pvt1–miR-322–Rtcb, Pvt1–miR-532–Atl2, and Pvt1–miR-101–Ajm1 have been previously experimentally validated using the HITS-CLIP technique in C2C12 cells

(see edges in MyoData). To demonstrate if Pvt1 is able to act as a sponge for miR-22, –27a, -322, and -532, we evaluated the expression of the miRNA targets after Pvt1 silencing in C2C12 myotubes. We expected that the reduction of Pvt1 allows the release of miRNAs from the lncRNA, thereby permitting them to downregulate their targets. We showed that all considered targets were downregulated with the exception of RAR Related Orphan Receptor B (Rorb) that was upregulated (Fig. 6B). These data support the sponge activity of Pvt1 and its interaction with miR-532 and –322, whose relationship was derived from RNA-CLIP experiments (Fig. 6A), but do not demonstrate the direct interaction between Pvt1 and miR-27 or –22. We excluded miR-22 from experiments to validate Pvt1 interactions with miRNAs, since the miRNA target transcript Rorb was not downregulated in Pvt1-silenced cells. In the validation experiments carried out with luciferase assays, we were able to demonstrate the direct interaction of Pvt1 with miR-27a (Fig. 6C). To strengthen this result, we overexpressed miR-27a in C2C12 cells showing the downregulation of both Nnmt and Cdh8 genes. Nnmt and Cdh8 downregulation was instead attenuated in cells overexpressing the region of Pvt1 containing binding sites for miR-27a (Fig. 6D).

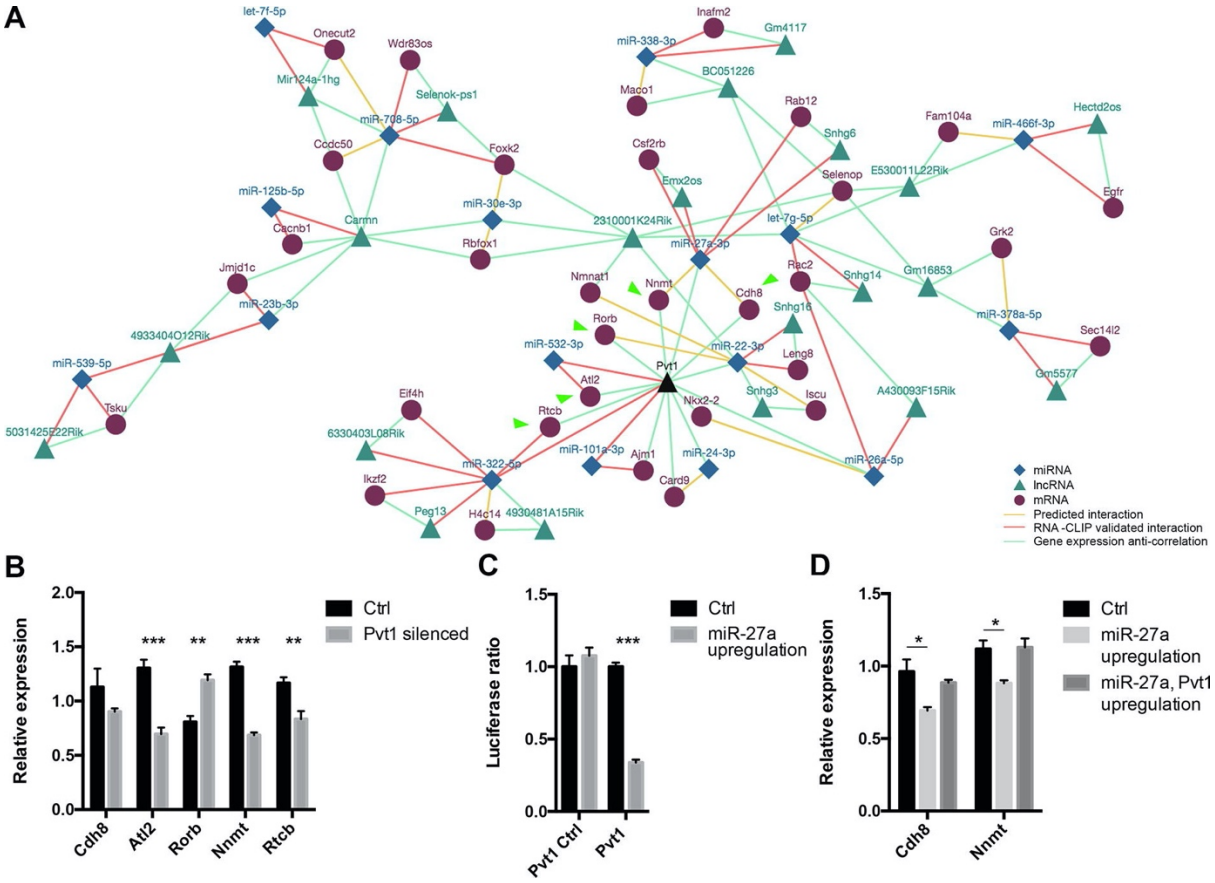


*Fig. 6. Pvt1 network. (A) Network associated to Pvt1 (black triangle) and single myofiber expression. Green arrows indicate mRNAs tested for their expression after the downregulation of Pvt1. (B) Histograms represent expression values relative to the average expression of the gene among samples. Tbp was used as control gene. At least four independent experiments were performed. Error bars indicate SEM. (C) Luciferase assays were performed to demonstrate the direct interaction between Pvt1 and miR-27a. Part of Pvt1 sequence containing the miRNA putative interaction site (or not containing; Pvt1 Ctrl) was cloned into pmirGLO vector. Firefly luciferase (reporter gene) and Renilla luciferase (control reporter for normalization) activities were measured after the transfection in C2C12 cells together with pCMV-MiR coding for miR-27a or empty pCMV-MiR (Ctrl). Data are expressed as the mean of at least five independent transfections. Error bars indicate SEM. (D) Histograms represent expression values relative to the average expression of the gene among samples. Tbp was used as control gene. Co-transfecting cells with pCMV-MiR vector coding for miR-27a and pmirGLO coding for the sequence part of Pvt1 with binding sites for miR-27a, Cdh8 and Nnmt expression were not affected. At least four biological replicates were performed. Error bars indicate SEM. For this entire figure, significance was calculated using t-test between control and each treated sample considering unequal variance between samples. * $P \leq 0.05$, ** $P \leq 0.01$, *** $P \leq 0.0005$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)*

4.2.3. Case study 3: The identification of miRNAs involved in myofiber type specification

MyoData allows for parallel searching for multiple entries. This may be useful, for example, to search if specific miRNAs influence the activity of genes coding for proteins that participate in the same cellular process or if they modulate the activity of co-regulated genes. We decided to use the database to evaluate if miR-206, −208b, and miR-214 can regulate genes involved in myofiber type specification. It was previously shown that loss of miR-214 expression in Zebrafish leads to a reduction of slow myofibers through the regulation of Su(fu) gene that participates in Hedgehog signaling. Su(fu) inhibition induces an increase in the number of slow myofibers [72]. miR-206 is predicted to regulate the expression of transcriptional repressors of the slow myosin heavy chain, such as Sox6, Purβ, and Sp3 [73].

By querying MyoData for miR-206, −208b, −214, Sox6, and Slc16a3 we retrieved the network described in Fig. 7A. The three miRNAs were selected because they are exclusively expressed in slow contracting myofibers [5] and probably impact specific functions in these myofibers. Sox6 was previously reported as an important transcription factor involved in the regulation of slow myosin heavy chain gene [74], while Slc16a3 (MCT3-M/MCT4), which codifies for a lactate transporter, may be involved in the metabolism of specific myofibers. In fact, it is much more abundantly expressed in fast-twitch oxidative and fast-twitch glycolytic muscles than in slow-

twitch oxidative muscles [75]. To validate this network and the suggested interactions between miRNAs and targets, we upregulated the expression of miR-208b or −214 in the C2C12 muscle cell line. In cells overexpressing miR-208b we found a clear downregulation of Sox6 (Fig. 7B and C), confirming previous evidence of this specific interaction [76]. Moreover, in C2C12 cells overexpressing miR-214, both Sox6 and Slc16a3 genes were downregulated (Fig. 7D and E). We confirmed the interaction between miR-214 and Sox6 and miR-214 and Slc16a3 via the luciferase assay (Fig. 7F) supporting the ability of miR-214 to regulate both Sox6 and Slc16a3 and its involvement in the modulation of slow myofiber functions.

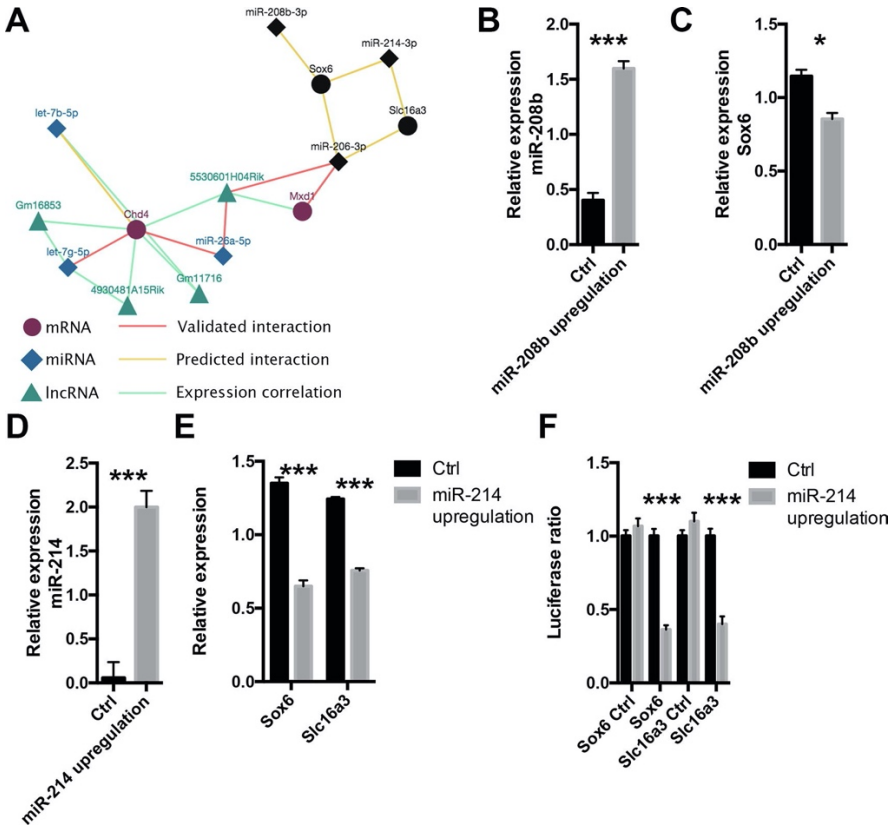


Fig. 7. Regulation of genes that present a myofiber type dependent expression. (A) Network resulted from multiple searching of mmu-miR-206-3p, –208b-3p, –214-3p, Sox6, and Slc16a3. Black nodes indicate user nodes. (B) Histograms represent expression values relative to the average expression of the gene among samples. U6 was used as control gene. Four biological replicates were performed. Error bars indicate SEM. (C) Histograms represent

*expression values relative to the average expression of the gene among samples. Txn1 was used as control gene. Four biological replicates were performed. Error bars indicate SEM. (D) Histograms represent expression values relative to the average expression of the gene among samples. U6 was used as control gene. Four biological replicates were performed. Error bars indicate SEM. (E) Histograms represent expression values relative to the average expression of the gene among samples. Txn1 was used as control gene. Four biological replicates were performed. Error bars indicate SEM. (F) Luciferase assays were performed to demonstrate the direct interaction between miR-214 and Sox6 and miR-214 and Slc16a3. Part of Sox6 and Slc16a3 sequences containing the miRNA putative interaction sites (or not containing; Sox6 Ctrl and Slc16a3 Ctrl) were cloned in pmirGLO vector. Firefly luciferase (reporter gene) and Renilla luciferase (control reporter for normalization) activities were measured after the transfection in C2C12 cells together with pCMV-MiR coding for miR-214 or empty pCMV-MiR (Ctrl). Data are expressed as the mean of at least four independent transfections. Error bars indicate SEM. For this entire figure, significance was calculated using t-test between control and treated samples considering unequal variance between samples. * $P \leq 0.05$, ** $P \leq 0.01$, *** $P \leq 0.0005$.*

4.2.4. Case study 4: Integration of single nucleus and single myofiber data: New perspectives to understand Spinal and bulbar muscular atrophy

Spinal and bulbar muscular atrophy (SBMA) is characterized by loss of motor neurons and sensory neurons, accompanied by atrophy of muscle fibers. This causes a glycolytic-to-oxidative fiber-type switch in fast-contracting skeletal muscles without a reduction of muscle mass in slow-contracting muscles (oxidative myofibers). Fast contracting muscles are also associated with a reduction of tetanic force while slow contracting muscles are not affected [77]. These observations suggest that oxidative myofibers are protected from the atrophy induced in SBMA patients. To better understand if non-coding RNAs participate in this protective mechanism we interrogated the MyoData database using the list of gene markers for slow nuclei described in [25]. The computed network is represented in Fig. 8A. Interestingly, a putative interaction of miR-27a with E2-ubiquitin ligase Ube2q1 is described. The miR-

27a and Ube2q1 couple is an interesting target because we previously showed that the upregulation of miR-27a induces the increase of oxidative myofibers [5] that may have a protective role in SBMA. miR-27a is expressed only in oxidative myofibers and silent in glycolytic myofibers [5], which are the most affected in muscles of SBMA patients. First, we asked if miR-27a can modulate marker genes for fast myofibers identified by snRNA-seq [25]. To respond to this question, we evaluated the network generated by gene markers of fast myofibers. Interestingly, miR-27a was predicted to regulate 55% (11 out of 20) of fast myonuclei markers (Fig. 8 B and Supplemental Table S4). In muscles overexpressing miR-27a we confirmed by qRT-PCR that ~ 82% (9 of 11 tested genes) of genes targeted by miR-27a were downregulated (Fig. 8C). This confirms the ability of miR-27a to inhibit the fast myofiber phenotype. We then experimentally validated the suggested interaction of miR-27a with Ube2q1 using the luciferase assay (Fig. 8D). Furthermore, we showed that following the upregulation of miR-27a in muscle cells, the expression of Ube2q1 significantly decreased (Fig. 8E). In summary, our experimental data support the ability of miR-27a to modulate genes specifically expressed in fast myofibers and to buffer the expression of Ube2q1 in oxidative myofibers but not in glycolytic myofibers. This evidence may be particularly important to modulate atrophic processes in SBMA muscle. Alternatively, the upregulation of Ube2q1 in SBMA muscles [78] may be associated with the inability of myoblasts to produce new myotubes in degenerating SBMA muscles. In fact, the upregulation of Ube2q1 is associated with enhanced cell proliferation in hepatocellular

carcinoma [79]. To produce myotubes, myoblasts have to withdraw from the cell cycle to fuse with each other. If withdrawal is prevented, myotubes cannot form. Finally, it is interesting to notice that Rocchi et al [77] described that a high-fat diet (HFD) ameliorates the phenotype of SBMA model mice. We showed that HFD induces the expression of miR-27a more in glycolytic than in oxidative muscles [5]. These two lines of evidence support the importance of miR-27a in SBMA and show how the database can be used to evaluate the impact of non-coding RNAs in the regulation of marker genes for specific myofibers identified by snRNA-seq experiments.

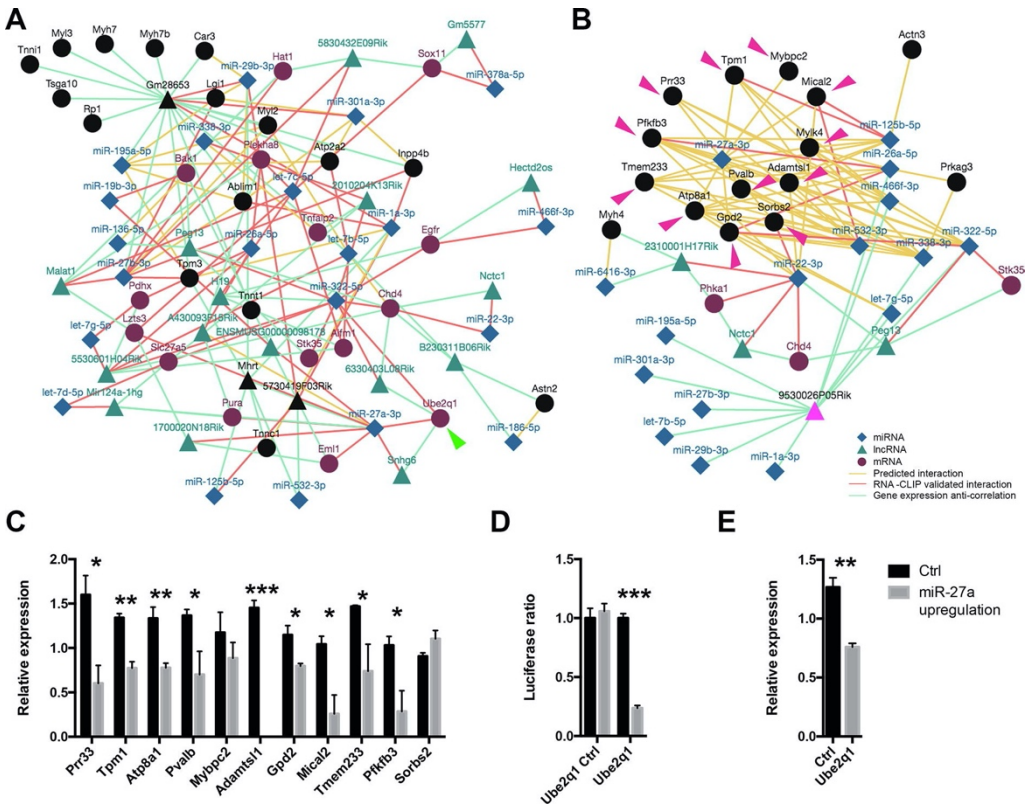


Fig. 8. Integration of single nucleus and single myofiber data. (A) Network resulted from the interrogation of MyoData with markers from a previous study [25] for slow myofibers. Green arrow indicates Ube2q1 gene. (B) Network resulted from the interrogation of MyoData with markers from a previous study [25] for fast myofibers. Pink arrows indicate predicted targets for miR-27a. The legend is for both part A and B of the figure. Black nodes in the networks indicate searched entries from the user. (C) Gene expression of predicted targets of miR-27a shown in part B of this figure. Histograms represent expression values relative to the average expression of the gene among samples. Txn1 was used as control gene. Four biological replicates were performed. Error bars indicate SEM. (D) Luciferase assays were performed to demonstrate the direct interaction between miR-27a and Ube2q1. Part of Ube2q1

*sequence containing the miRNA putative interaction sites (or not containing; Ube2q1 Ctrl) were cloned in pmirGLO vector. Firefly luciferase (reporter gene) and Renilla luciferase (control reporter for normalization) activities were measured after the transfection in C2C12 cells together with pCMV-MiR coding for miR-27a or empty pCMV-MiR (Ctrl). Data are expressed as the mean of four independent transfections. Error bars indicate SEM. (E) Relative expression of Ube2q1. Histograms represent expression values relative to the average expression of the gene among samples. Txn1 was used as control gene. Four biological replicates were performed. Error bars indicate SEM. For this entire figure, significance was calculated using t-test between control and treated samples considering unequal variance between samples. * P ≤ 0.05, ** P ≤ 0.01, *** P ≤ 0.0005. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)*

5. Conclusions

Gene regulation is a complex process where regulatory elements and their targets participate to form highly complex interactions thus affecting biological processes. Transcription factors (TFs) are the most well-known molecules involved in this process and several tools and databases have been published to evaluate TFs involved in the regulation of commonly altered genes [80–85]. Some databases have been already developed to explore the gene expression of skeletal muscle [86–89] and skeletal muscle after exercise [90,91] without considering the importance of non-coding RNAs in the post-transcriptional regulation of gene expression. Different databases integrate TFs and miRNAs to describe feed-forward regulatory circuits [92–94]. Improvements in RNA sequencing technologies have allowed for the identification of single cell and single nucleus gene expression and the consequent development of several web interfaces to query mRNA and lncRNA gene expression [95–100]. However, the integration of miRNA–lncRNA–mRNA networks at the single cell level has not been demonstrated. Such integration represents an important improvement in the comprehension of gene regulation by allowing for the identification of cell type-

specific expression (higher than coding genes) and the ability of lncRNAs to sponge miRNAs.

We took advantage of our genome-wide experiments on single myofibers to implement a database to describe hypothetical and experimentally-validated interactions among miRNAs, lncRNAs, and coding RNAs to dissect gene regulation in different myofiber types. The database can be used to evaluate the impact of a single gene or group of genes (both coding and non-coding genes) on the regulation of related genes (co-expressed or coding for proteins involved in a specific pathway). MyoData integrates miRNAs and lncRNAs in KEGG pathways thereby incorporating the information of the regulation of biological processes. Mature myofibers are derived from the fusion of multiple satellite cells and are therefore a syncytium containing hundreds of nuclei that can participate differently to the cumulative gene expression [25]. We therefore included the possibility of visualizing how the networks calculated in the database change considering clusters of nuclei based on their expression retrieved from snRNA-seq experiments. We integrated these clusters with our information on miRNA expression in single myofibers because it is not feasible with current techniques to recover mature miRNA expression from snRNA-seq. We showed that this approach may be useful to identify miRNAs that regulate coding genes involved in muscle atrophy. By evaluating specific miRNAs and lncRNAs, we experimentally demonstrated that the database can guide the discovery of novel functions of non-coding RNAs in skeletal muscle. Moreover, we showed that MyoData

is a valuable resource to integrate single myofiber and single nucleus gene expression information to investigate at a deeper level the molecular bases and regulations of physiology and pathology of such an abundant and complex organ as skeletal muscle.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors wish to thank Dr. Federico Caicci (BioImaging Facility of the Department of Biology, University of Padova) for the support in EM analysis. We also thank Dr. Matthieu Dos Santos for having shared with us multiplex immunohistochemistry of the skeletal muscle and Dr. Andreas C. Chai for proofreading the manuscript. This research was supported by CARIPLO Foundation [2016–1006 to S.C. and G.L.].

Authors statement

Davide Corso: developed the database and performed bioinformatic analyses; Francesco Chemello: wrote and edited the manuscript; Enrico Alessio: edited the manuscript; Ilenia Urso: performed bioinformatic analyses; Giulia Ferrarese: performed experiments; Martina Bazzega: performed experiments; Chiara Romualdi: reviewed and edited the manuscript; Gerolamo Lanfranchi: reviewed and edited the manuscript; Gabriele Sales: wrote original draft, supervised database construction, conceptualized the database, implemented the computer code; Stefano Cagnin:

conceptualized the database, prepared figures, wrote original draft, supervised experiments.

References

1. Giordani L, He GJ, Negroni E, Sakai H, Law JYC, Siu MM, et al. High-Dimensional Single-Cell Cartography Reveals Novel Skeletal Muscle-Resident Cell Populations. Mol Cell. 2019;74: 609-621.e6. doi:10.1016/j.molcel.2019.02.026
2. Dave HD, Shook M, Varacallo M. Anatomy, Skeletal Muscle. StatPearls. 2021.
3. Van Wessel T, De Haan A, Van Der Laarse WJ, Jaspers RT. The muscle fiber type-fiber size paradox: Hypertrophy or oxidative metabolism? Eur J Appl Physiol. 2010;110: 665–694. doi:10.1007/s00421-010-1545-0
4. Matsakas A, Patel K. Skeletal muscle fibre plasticity in response to selected environmental and physiological stimuli. Histol Histopathol. 2009;24: 611–629.
5. Chemello F, Grespi F, Zulian A, Cancellara P, Hebert-Chatelain E, Martini P, et al. Transcriptomic Analysis of Single Isolated Myofibers Identifies miR-27a-3p and miR-142-3p as Regulators of Metabolism in Skeletal Muscle. Cell Rep. 2019;26: 3784-3797.e8. doi:10.1016/j.celrep.2019.02.105
6. Mok GF, Lozano-Velasco E, Münsterberg A. microRNAs in skeletal muscle development. Semin Cell Dev Biol. 2017;72: 67–76. doi:10.1016/j.semcdb.2017.10.032
7. Alexander MS, Kunkel LM. Skeletal muscle MicroRNAs: Their diagnostic and therapeutic potential in human muscle diseases. J Neuromuscul Dis. 2015;2: 1–11. doi:10.3233/JND-140058

8. Wang S, Jin J, Xu Z, Zuo B. Functions and regulatory mechanisms of lncrnas in skeletal myogenesis, muscle disease and meat production. Cells. 2019;8. doi:10.3390/cells8091107

9. Sweta S, Dudnakova T, Sudheer S, Baker AH, Bhushan R. Importance of long non-coding RNAs in the development and disease of skeletal muscle and cardiovascular lineages. Front Cell Dev Biol. 2019;7: 1–19. doi:10.3389/fcell.2019.00228

10. Vacante F, Denby L, Sluimer JC, Baker AH. The function of miR-143, miR-145 and the MiR-143 host gene in cardiovascular development and disease. Vascul Pharmacol. 2019;112: 24–30. doi:10.1016/j.vph.2018.11.006

11. Martone J, Mariani D, Desideri F, Ballarino M. Non-coding RNAs Shaping Muscle. Front Cell Dev Biol. 2020;7. doi:10.3389/fcell.2019.00394

12. Alessio E, Buson L, Chemello F, Peggion C, Grespi F, Martini P, et al. Single cell analysis reveals the involvement of the long non-coding RNA Pvt1 in the modulation of muscle atrophy and mitochondrial network. Nucleic Acids Res. 2019;47: 1653–1670. doi:10.1093/nar/gkz007

13. Cabili M, Trapnell C, Goff L, Koziol M, Tazon-Vega B, Regev A, et al. Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. Genes Dev. 2011;25: 1915–1927. doi:10.1101/gad.17446611

14. Derrien T, Johnson R, Bussotti G, Tanzer A, Djebali S, Tilgner H, et al. The

GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. Genome Res. 2012;22: 1775–1789. doi:10.1101/gr.132159.111

15. Marques AC, Ponting CP. Catalogues of mammalian long noncoding RNAs: Modest conservation and incompleteness. Genome Biol. 2009;10. doi:10.1186/gb-2009-10-11-r124

16. Mercer TR, Dinger ME, Sunkin SM, Mehler MF, Mattick JS. Specific expression of long noncoding RNAs in the mouse brain. Proc Natl Acad Sci U S A. 2008;105: 716–721. doi:10.1073/pnas.0706729105

17. Vučićević D, Corradin O, Ntini E, Scacheri PC, Ørom UA. Long ncRNA expression associates with tissue-specific enhancers. Cell Cycle. 2015;14: 253–260. doi:10.4161/15384101.2014.977641

18. Filipowicz W, Bhattacharyya SN, Sonenberg N. Mechanisms of post-transcriptional regulation by microRNAs: Are the answers in sight? Nat Rev Genet. 2008;9: 102–114. doi:10.1038/nrg2290

19. Fernandes JCR, Acuña SM, Aoki JI, Floeter-Winter LM, Muxel SM. Long non-coding RNAs in the regulation of gene expression: Physiology and disease. Non-coding RNA. 2019;5. doi:10.3390/ncrna5010017

20. Paraskevopoulou MD, Hatzigeorgiou AG. Analyzing MiRNA–LncRNA interactions. Methods in Molecular Biology. 2016. pp. 271–286. doi:10.1007/978-1-4939-3378-5_21

21. Alessio E, Bonadio RS, Buson L, Chemello F, Cagnin S. A single cell but many different transcripts: A journey into the world of long non-coding RNAs. Int J Mol Sci. 2020;21. doi:10.3390/ijms21010302

22. Nomura S. Single-cell genomics to understand disease pathogenesis. J Hum Genet. 2021;66: 75–84. doi:10.1038/s10038-020-00844-3

23. Strzelecka PM, Ranzoni AM, Cvejic A. Dissecting human disease with single-cell omics: Application in model systems and in the clinic. DMM Dis Model Mech. 2018;11. doi:10.1242/dmm.036525

24. Chemello F, Wang Z, Li H, McAnally JR, Liu N, Bassel-Duby R, et al. Degenerative and regenerative pathways underlying Duchenne muscular dystrophy revealed by single-nucleus RNA sequencing. Proc Natl Acad Sci U S A. 2020;117: 29691–29701. doi:10.1073/pnas.2018391117

25. Dos Santos M, Backer S, Saintpierre B, Izac B, Andrieu M, Letourneur F, et al. Single-nucleus RNA-seq and FISH identify coordinated transcriptional activity in mammalian myofibers. Nat Commun. 2020;11. doi:10.1038/s41467-020-18789-8

26. Petrany MJ, Swoboda CO, Sun C, Chetal K, Chen X, Weirauch MT, et al. Single-nucleus RNA-seq identifies transcriptional heterogeneity in multinucleated skeletal myofibers. Nat Commun. 2020;11. doi:10.1038/s41467-020-20063-w

27. Vlachos IS, Paraskevopoulou MD, Karagkouni D, Georgakilas G, Vergoulis T,

Kanellos I, et al. DIANA-TarBase v7.0: Indexing more than half a million experimentally supported miRNA:mRNA interactions. Nucleic Acids Res. 2015;43: D153–D159. doi:10.1093/nar/gku1215

28. http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=tarbase/index. [accessed on 6th of July 2021].
29. Li J-H, Liu S, Zhou H, Qu L-H, Yang J-H. StarBase v2.0: Decoding miRNA-ceRNA, miRNA-ncRNA and protein-RNA interaction networks from large-scale CLIP-Seq data. Nucleic Acids Res. 2014;42. doi:10.1093/nar/gkt1248
30. http://starbase.sysu.edu.cn/. [accessed on 6th of July 2021].
31. Chen Y, Wang X. MiRDB: An online database for prediction of functional microRNA targets. Nucleic Acids Res. 2020;48: D127–D131. doi:10.1093/nar/gkz757
32. http://mirdb.org/. (Accessed on 6th of July 2021).
33. Vejnar CE, Zdobnov EM. MiRmap: Comprehensive prediction of microRNA target repression strength. Nucleic Acids Res. 2012;40: 11673–11683. doi:10.1093/nar/gks901
34. https://mirmap.ezlab.org/. [accessed on 6th of July 2021].
35. https://cm.jefferson.edu/rna22/. [accessed on 6th of July 2021].
36. https://genie.weizmann.ac.il/pubs/mir07/index.html. [accessed on 6th of July 2021].
37. Paraskevopoulou MD, Vlachos IS, Karagkouni D, Georgakilas G, Kanellos I,

Vergoulis T, et al. DIANA-LncBase v2: Indexing microRNA targets on non-coding transcripts. Nucleic Acids Res. 2016;44: D231–D238. doi:10.1093/nar/gkv1270

38. Sales G, Calura E, Romualdi C. Meta Graphite-a new layer of pathway annotation to get metabolite networks. Bioinformatics. 2019;35: 1258–1260. doi:10.1093/bioinformatics/bty719

39. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17: 261–272. doi:10.1038/s41592-019-0686-2

40. Seabold S, Perktold J. Statsmodels: Econometric and statistical modeling with python. Proc 9th Python Sci Conf. 2010; 57–61.

41. https://pypi.org/project/python-rocksdb/. [accessed on 6th of July 2021].

42. Hagberg AA, Schult DA, Swart PJ. Exploring network structure, dynamics, and function using NetworkX. 7th Python in Science Conference (SciPy 2008). 2008.

43. https://reactjs.org. [accessed on 6th of July 2021].

44. https://fontawesome.com. [accessed on 6th of July 2021].

45. https://apexcharts.com/docs/react-charts/#. [accessed on 6th of July 2021].

46. https://react-table.tanstack.com. [accessed on 6th of July 2021].

47. Franz M, Lopes CT, Huck G, Dong Y, Sumer O, Bader GD. Cytoscape.js: A graph theory library for visualisation and analysis. Bioinformatics. 2016;32: 309–311. doi:10.1093/bioinformatics/btv557

48. https://github.com/cytoscape/cytoscape.js-cola. [accessed on 6th of July 2021].

49. Vowinckel J, Hartl J, Butler R, Ralser M. MitoLoc: A method for the simultaneous quantification of mitochondrial network morphology and membrane potential in single cells. Mitochondrion. 2015;24: 77–86. doi:10.1016/j.mito.2015.07.001

50. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, et al. Cytoscape: A software Environment for integrated models of biomolecular interaction networks. Genome Res. 2003;13: 2498–2504. doi:10.1101/gr.1239303

51. Wang Y-X, Zhang C-L, Yu RT, Cho HK, Nelson MC, Bayuga-Ocampo CR, et al. Regulation of muscle fiber type and running endurance by PPARδ. PLoS Biol. 2004;2. doi:10.1371/journal.pbio.0020294

52. Vickers KC, Shoucri BM, Levin MG, Wu H, Pearson DS, Osei-Hwedieh D, et al. MicroRNA-27b is a regulatory hub in lipid metabolism and is altered in dyslipidemia. Hepatology. 2013;57: 533–542. doi:10.1002/hep.25846

53. Coffey AR, Smallwood TL, Albright J, Hua K, Kanke M, Pomp D, et al. Systems genetics identifies a co-regulated module of liver microRNAs associated with plasma LDL cholesterol in murine diet-induced dyslipidemia. Physiol Genomics. 2017;49: 618–629. doi:10.1152/physiolgenomics.00050.2017

54. Wang T, Li M, Guan J, Li P, Wang H, Guo Y, et al. MicroRNAs miR-27a and miR-143 regulate porcine adipocyte lipid metabolism. Int J Mol Sci. 2011;12: 7950–7959. doi:10.3390/ijms12117950

55. McCarron JG, Wilson C, Sandison ME, Olson ML, Girkin JM, Saunter C, et al. From structure to function: Mitochondrial morphology, motion and shaping in vascular smooth muscle. J Vasc Res. 2013;50: 357–371. doi:10.1159/000353883

56. Rafelski SM. Mitochondrial network morphology: Building an integrative, geometrical view. BMC Biol. 2013;11. doi:10.1186/1741-7007-11-71

57. Putti R, Migliaccio V, Sica R, Lionetti L. Skeletal muscle mitochondrial bioenergetics and morphology in high fat diet induced obesity and insulin resistance: Focus on dietary fat source. Front Physiol. 2016;6. doi:10.3389/fphys.2015.00426

58. Romanello V, Sandri M. Mitochondrial quality control and muscle mass maintenance. Front Physiol. 2016;6. doi:10.3389/fphys.2015.00422

59. Li Y, Jiang J, Liu W, Wang H, Zhao L, Liu S, et al. MicroRNA-378 promotes autophagy and inhibits apoptosis in skeletal muscle. Proc Natl Acad Sci U S A. 2018;115: E10849–E10858. doi:10.1073/pnas.1803377115

60. Suen D-F, Norris KL, Youle RJ. Mitochondrial dynamics and apoptosis. Genes Dev. 2008;22: 1577–1590. doi:10.1101/gad.1658508

61. Caravia XM, Fanjul V, Oliver E, Roiz-Valle D, Morán-Álvarez A, Desdín-Micó G, et al. The microRNA-29/PGC1α regulatory axis is critical for metabolic control of cardiac function. PLoS Biol. 2018;16. doi:10.1371/journal.pbio.2006247

62. Geng L, Zhu B, Dai B-H, Sui C-J, Xu F, Kan T, et al. A let-7/Fas double-negative feedback loop regulates human colon carcinoma cells sensitivity to Fas-related

apoptosis. Biochem Biophys Res Commun. 2011;408: 494–499. doi:10.1016/j.bbrc.2011.04.074

63. McCarthy JJ. The myomiR network in skeletal muscle plasticity. Exerc Sport Sci Rev. 2011;39: 150–154. doi:10.1097/JES.0b013e31821c01e1

64. Du J, Zhang Y, Shen L, Luo J, Lei H, Zhang P, et al. Effect of MIR-143-3p on C2C12 myoblast differentiation. Biosci Biotechnol Biochem. 2016;80: 706–711. doi:10.1080/09168451.2015.1123604

65. Onodera Y, Teramura T, Takehara T, Itokazu M, Mori T, Fukuda K. Inflammation-associated MIR-155 activates differentiation of muscular satellite cells. PLoS One. 2018;13. doi:10.1371/journal.pone.0204860

66. Jung HJ, Lee K-P, Kwon K-S, Suh Y. MicroRNAs in Skeletal Muscle Aging: Current Issues and Perspectives. Journals Gerontol - Ser A Biol Sci Med Sci. 2019;74: 1008–1014. doi:10.1093/gerona/gly207

67. Chen G-Q, Lian W-J, Wang G-M, Wang S, Yang Y-Q, Zhao Z-W. Altered microRNA expression in skeletal muscle results from high-fat diet-induced insulin resistance in mice. Mol Med Rep. 2012;5: 1362–1368. doi:10.3892/mmr.2012.824

68. Paci P, Colombo T, Farina L. Computational analysis identifies a sponge interaction network between long non-coding RNAs and messenger RNAs in human breast cancer. BMC Syst Biol. 2014;8. doi:10.1186/1752-0509-8-83

69. Wang C, Han C, Zhang Y, Liu F. LncRNA PVT1 regulate expression of HIF1α

via functioning as ceRNA for miR-199a-5p in non-small cell lung cancer under hypoxia. Mol Med Rep. 2018;17: 1105–1110. doi:10.3892/mmr.2017.7962

70. Li T, Meng X-L, Yang W-Q. Long Noncoding RNA PVT1 Acts as a "Sponge" to Inhibit microRNA-152 in Gastric Cancer Cells. Dig Dis Sci. 2017;62: 3021–3028. doi:10.1007/s10620-017-4508-z

71. Wang L, Xiao B, Yu T, Gong L, Wang Y, Zhang X, et al. lncRNA PVT1 promotes the migration of gastric cancer by functioning as ceRNA of miR-30a and regulating Snail. J Cell Physiol. 2021;236: 536–548. doi:10.1002/jcp.29881

72. Flynt AS, Li N, Thatcher EJ, Solnica-Krezel L, Patton JG. Zebrafish miR-214 modulates Hedgehog signaling to specify muscle cell fate. Nat Genet. 2007;39: 259–263. doi:10.1038/ng1953

73. McCarthy JJ. MicroRNA-206: The skeletal muscle-specific myomiR. Biochim Biophys Acta - Gene Regul Mech. 2008;1779: 682–691. doi:10.1016/j.bbagrm.2008.03.001

74. Hagiwara N, Yeh M, Liu A. Sox6 is required for normal fiber type differentiation of fetal skeletal muscle in mice. Dev Dyn. 2007;236: 2062–2076. doi:10.1002/dvdy.21223

75. Bonen A. Lactate transporters (MCT proteins) in heart and skeletal muscles. Med Sci Sports Exerc. 2000;32: 778–789. doi:10.1097/00005768-200004000-00010

76. van Rooij E, Quiat D, Johnson BA, Sutherland LB, Qi X, Richardson JA, et al. A Family of microRNAs Encoded by Myosin Genes Governs Myosin Expression

and Muscle Performance. Dev Cell. 2009;17: 662–673. doi:10.1016/j.devcel.2009.10.013

77. Rocchi A, Milioto C, Parodi S, Armirotti A, Borgia D, Pellegrini M, et al. Glycolytic-to-oxidative fiber-type switch and mTOR signaling activation are early-onset features of SBMA muscle modified by high-fat diet. Acta Neuropathol. 2016;132: 127–144. doi:10.1007/s00401-016-1550-4

78. Rusmini P, Polanco MJ, Cristofani R, Cicardi ME, Meroni M, Galbiati M, et al. Aberrant Autophagic Response in the Muscle of A Knock-in Mouse Model of Spinal and Bulbar Muscular Atrophy. Sci Rep. 2015;5. doi:10.1038/srep15174

79. Chang R, Wei L, Lu Y, Cui X, Lu C, Liu L, et al. Upregulated expression of ubiquitin-conjugating enzyme E2Q1 (UBE2Q1) is associated with enhanced cell proliferation and poor prognosis in human hapatocellular carcinoma. J Mol Histol. 2015;46: 45–56. doi:10.1007/s10735-014-9596-x

80. Kreft L, Soete A, Hulpiau P, Botzki A, Saeys Y, De Bleser P. ConTra v3: A tool to identify transcription factor binding sites across species, update 2017. Nucleic Acids Res. 2017;45: W490–W494. doi:10.1093/nar/gkx376

81. Lee C, Huang CH. LASAGNA-search: An integrated web tool for transcription factor binding site search and visualization. Biotechniques. 2013;54. doi:10.2144/000113999

82. Sun K, Wang H, Sun H. MTFkb: A knowledgebase for fundamental annotation of mouse transcription factors. Sci Rep. 2017;7. doi:10.1038/s41598-017-02404-w

83. Cui X, Wang T, Chen HS, Busov V, Wei H. TF-finder: A software package for identifying transcription factors involved in biological processes using microarray data and existing knowledge base. BMC Bioinformatics. 2010;11. doi:10.1186/1471-2105-11-425

84. Gearing LJ, Cumming HE, Chapman R, Finkel AM, Woodhouse IB, Luu K, et al. CiIIder: A tool for predicting and analysing transcription factor binding sites. PLoS One. 2019;14. doi:10.1371/journal.pone.0215495

85. Roopra A. MAgIC: A tool for predicting transcription factors and cofactors driving gene sets using ENCODE data. PLoS Comput Biol. 2020;16. doi:10.1371/journal.pcbi.1007800

86. Asplund O, Rung J, Groop L, Rashmi Prasad B, Hansson O. MuscleAtlasExplorer: A web service for studying gene expression in human skeletal muscle. Database. 2020;2020. doi:10.1093/database/baaa111

87. https://www.proteinatlas.org/humanproteome/tissue/skeletal+muscle. [accessed on 6th of July 2021].

88. https://nicopillon.com/tools/muscle-atlas/. [accessed on 6th of July 2021].

89. http://yu-mbl-muscledb.com/NeuroMuscleDB/. [accessed on 6th of July 2021].

90. Pillon NJ, Gabriel BM, Dollet L, Smith JAB, Sardón Puig L, Botella J, et al. Transcriptomic profiling of skeletal muscle adaptations to exercise and inactivity. Nat Commun. 2020;11. doi:10.1038/s41467-019-13869-w

91. Cao Q, Deng Z, Liu J, Li X. SGDB: A Sports Gene Database for Visualization of

Sports Effects on Human Skeletal Muscle Gene Expression. IEEE Access. 2020;8. doi:10.1109/ACCESS.2020.2968514

92. Friard O, Re A, Taverna D, De Bortoli M, Corá D. CircuitsDB: A database of mixed microRNA/transcription factor feed-forward regulatory circuits in human and mouse. BMC Bioinformatics. 2010;11. doi:10.1186/1471-2105-11-435

93. Tong Z, Cui Q, Wang J, Zhou Y. TransmiR v2.0: An updated transcription factor-microRNA regulation database. Nucleic Acids Res. 2019;47. doi:10.1093/nar/gky1023

94. Wang S, Li W, Lian B, Liu X, Zhang Y, Dai E, et al. ™REC: A database of transcription factor and miRNA regulatory cascades in human diseases. PLoS One. 2015;10. doi:10.1371/journal.pone.0125222

95. Franzén O, Gan LM, Björkegren JLM. PanglaoDB: A web server for exploration of mouse and human single-cell RNA sequencing data. Database. 2019;2019. doi:10.1093/database/baz046

96. https://singlecell.broadinstitute.org/single_cell. [accessed on 6th of July 2021].

97. https://bioinfo.uth.edu/scrnaseqdb/. [accessed on 6th of July 2021].

98. Zhao T, Lyu S, Lu G, Juan L, Zeng X, Wei Z, et al. SC2disease: A manually curated database of single-cell transcriptome for human diseases. Nucleic Acids Res. 2021;49. doi:10.1093/nar/gkaa838

99. https://www.ebi.ac.uk/gxa/sc/home. [accessed on 6th of July 2021].

100. Wang Z, Feng X, Li SC. SCDevDB: A database for insights into single-cell gene

expression profiles during human developmental processes. Front Genet. 2019;10. doi:10.3389/fgene.2019.00903

SpatialDE

1. Introduction and rationale

In biology, the knowledge of the spatial context allows models to describe a biological network in which every element is influenced by its surrounding environment. Indeed, the gene expression of each cell within a tissue microenvironment influences and is influenced by the cells around them [1]. In the microenvironment, positional information and gene expression are essential to understanding tissue functionality [2] and elucidating the context-dependent transcriptional regulation during development [3].

Recently, the development of high-throughput single-cell sequencing enabled the measurements of gene expression in complex biological systems [4] and provided unprecedented insights into the cellular diversity of tissues across diverse organisms [5]. It has allowed the building of comprehensive atlases useful to describe different cell expression profiles, including marker genes [6,7] characterizing rare cell types that

play a role in genetic diseases [8,9], and providing information for previously uncharacterized cell types.

Although this sequencing technology has revolutionized our understanding of gene expression at the single-cell level and became a standard approach to characterizing cell identity and state [10], it has the limitation of losing the positional information of the cells due to the including of a step of cell dissociation [11]. Recent advancements such as Spatial Resolved Transcriptomics (SRT) overcome this limitation by preserving spatial information. SRT can measure gene expression (from hundreds to thousands of genes) at different resolutions per spot, from several cells to less than one cell, along with spatial coordinates of the measurements [12]. It has the potential to significantly advance the biomedical research field [13] and to increase the knowledge of complex multicellular biological systems.

In this field, different technologies have been built with the common aim to combine gene expression data with spatial information [1].

SRT is divided into two categories:

- image-base methods, including in situ hybridization (ISH) and in situ sequencing (ISS), which require prior knowledge of genes of interest;
- capture-based methods, including laser capture microdissection (LCM) that are spatially barcoded on slides, or beads, able to capture the whole transcriptome from tissue sections.

Among image-capture technologies, the ISH visualizes RNA molecules directly in their original environment by hybridizing a fluorescently labeled complementary to the RNA target of interest. This strategy allows the detection of gene expression in fixed tissues [1]. Further development of this approach, named single-molecule RNA fluorescence in situ hybridization (smFISH), gives a higher and more robust signal enabling quantitative measurements of transcripts. smFISH provides high sensitivity but it has the limitation of targeting a few genes at the time.

The ISS technology performs RNA sequencing directly on the RNA content of the cell within the tissue. mRNA molecules are reverse transcribed and amplified by rolling circle amplification (RCA). The micrometer- or nanometer-sized RCA products are later sequenced and the barcodes, that are joined to the biological sequence, are decoded [1,13].

Regarding the capture-based methods, they are divided into three sub-categories [13]: directly cutting out a region of interest from the tissue by laser capture microdissection (LCM); using custom slides, or bead arrays to capture mRNAs by oligonucleotide-based spatial barcodes followed by NGS. Regarding the first category, in 2017 Geo-Seq [14] proposed an extended version in which LCM is combined with single-cell sequencing to profile the transcriptome of that region.

The Spatial Transcriptomics (ST) [2] technology analyzes the transcriptome of a particular tissue section by placing it over glass slides which are pre-arranged with a set of barcoded RT primers, specifying specific coordinates of the array. The tissue is

permeabilized and mRNAs hybridize to the barcoded RT primers. Reverse transcription is performed in situ, and the cDNA-mRNA complexes are extracted for library preparation and sequencing. Other technologies perform a procedure similar to the ST method but using beads in solution and then dispense them on a glass surface [1]. For example, Slide-seq [15] is a technology that uses 10μm-sized beads, and beads' positions are decoded by sequencing-by-ligation (SBL).

In general, in the previous years, different technological platforms have been developed, for example, Spatial Transcriptomics [2], 10x Genomics Visium (10x Genomics), Slide-seqV2 [16], seqFISH [17,18], each one with different features regarding the number of transcripts detected, cellular resolution, and size of the region captured [13,19].

One of the most common analyses that can be performed using this type of data is the identification of the spatially variable genes (SVGs), which are genes that have a spatial pattern of expression variation. SVGs can represent potential markers of biological processes and they can be used for downstream analyses [12]. This type of analysis is an example of how spatial information can be useful in addition to gene expression measures. Indeed, one of the frequent analyses performed on scRNA-seq data is the computation of highly variable genes (HVG), which is however performed ignoring the spatial information.

In recent years different approaches have been developed to compute spatially variable genes, and one of the first methods is SpatialDE [20], which is also able to

provide another type of analysis called *'automatic expression histology*' able to group SVGs with similar spatial expression patterns. SpatialDE is a Python method able to detect SVGs using a class of models initially developed in the context of geostatistics. My project aimed to create an R package to wrap the Python functions and methods of SpatialDE, allowing the use of such a statistical analysis in the R environment. This wrapper was created in response to one of the BiocSpatialChallenges (https://helenalc.github.io/BiocSpatialChallenges), proposed during the conference EuroBioc 2020. The resulting package has been published on the Bioconductor platform.

2. Materials and Methods

This R wrapper was developed following the Bioconductor package development guidelines (https://contributions.bioconductor.org/) which promote high-quality, well-documented, and interoperable software.

The aim of the project was to create a wrapper of the original Python SpatialDE methods, thus I didn't re-write all the Python code in the corresponding R environment. Instead, I have created a Python environment that can be called from R, able to perform the Python functions of SpatialDE. To build such a wrapper, I used two important packages:

- Basilisk: (https://www.bioconductor.org/packages/release/bioc/html/basilisk.html);
- Reticulate (https://github.com/rstudio/reticulate/).

Basilisk is a Bioconductor package, which is able to handle Python dependencies by automatically creating and managing a Conda environment from R, ensuring the availability of the required Python and system libraries. Furthermore, it is able to freeze those dependencies at a specific version. This package is well integrated with

Reticulate which provides a comprehensive set of tools for interoperability between Python and R. It allows to run of Python code from R in a variety of ways and translates Python objects to R (for example, between R and Python Pandas data frames, or between R matrices and Python NumPy arrays), as shown in Tab. 1. The combination of these two packages allows interoperating between the two languages.

Tab. 1: Type conversions between R and Python (adapted from https://rstudio.github.io/reticulate/#type-conversions).

R	**Python**	**Examples**
Single-element vector	Scalar	`1`, `1L`, `TRUE`, `"foo"`
Multi-element vector	List	`c(1.0, 2.0, 3.0)`, `c(1L, 2L, 3L)`
List of mutiple types	Tuple	`list(1L, TRUE, "foo")`
Named list	Dict	`list(a = 1L, b = 2.0)`, `dict(x = x_data)`
Matrix/Array	NumPy ndarray	`matrix(c(1,2,3,4), nrow = 2, ncol = 2)`
Data Frame	Pandas DataFrame	`data.frame(x = c(1,2,3), y = c("a", "b", "c"))`
Function	Python function	`function(x) x+1`
NULL, TRUE, FALSE	None, True, False	`NULL`, `TRUE`, `FALSE`

Example code of object conversion (adapted from https://rstudio.github.io/reticulate/articles/calling_python.html#object-conversion):

```
# import numpy and specify no automatic Python to R conversion
np <- import("numpy", convert = FALSE)

# do some array manipulations with NumPy
a <- np$array(c(1:4))
sum <- a$cumsum()

# convert to R explicitly at the end
py_to_r(sum)
```

This integration between Basilisk and Reticulate, in addition to the interconnection between the object of the two languages, was necessary also for the input and the output of the wrapper. Being an R package, this wrapper takes an input R objects, thus through Reticulate, I was able to convert objects from R to Python. By default, the Python method of SpatialDE provides a basic workflow in which the two main inputs data concern two pandas data frames:

- the first one referred to as "count matrix", is a classic count matrix embedded in a data frame, thus containing the expression measurement of the genes for all the sequenced spots;
- the second referred to as "coordinate", contains the 2-dimension spatial information described as coordinates X and Y of each spot.

The Python method SpatialDE is able to compute its workflow starting from these two types of objects. It is important to say that those can be considered as the basic and standard objects for SRT data, as they provide spatial location in addition to the standard count matrix of gene expression.

To ensure the correctness of the R input data, the wrapper provides various checks performed with the "*checkmate*" package, which allows for an assessment of the right type of object on specific functions and returns a documented message in case of type errors.

Moreover, the R wrapper was developed to support S4 objects. These represent a class system allowing the execution of the code within the paradigm of object-oriented programming. In particular, despite the Python SpatialDE run on given input of Pandas data frames, the R wrapper integrates the possibility to perform the SpatialDE workflow on SpatialExperiment (SpE) object. SpE is an infrastructure package that allows storing data from Spatially Resolved Transcriptomic data, independently from the source technology. It provides a robust infrastructure that simplifies operations on the data to the user. Furthermore, it is compatible with downstream analysis packages that use the SpatialExperiment or SingleCellExperiment class (which is the one being extended) [19].

Through, Basilisk and Reticulate, I was able to encapsulate in R all the python functions provided by the original SpatialDE, including the 'automatic expression histology' analysis, and provide all these functions able to accept input both as a data

frame or as an S4 class of SpatialExperiment. Furthermore, the wrapper provides functions to perform plots included in the publication of the method: '*FVS_sig*' which describes the fraction spatial variance versus the Q-value; '*multiGenePlots*' which allows displaying the spatial patterns of multiple genes.

3. Results and conclusions

This project was a collaboration with other two Ph.D. students (one from the Ghent University and another from the California Institute of Technology). The resulting R package wraps the Python functions of SpatialDE, one of the most popular methods for the identification of spatially variable genes. The repository is publicly available on GitHub (at the following link https://github.com/sales-lab/spatialDE). In addition, we chose to publish this package also on Bioconductor (at the following link https://www.bioconductor.org/packages/release/bioc/html/spatialDE.html), as one of its missions is to provide high-quality documentation packages to simplify integration for the user. The linked Bioconductor page contains detailed information on how to install and use the package. Specifically, a "vignette" is available which provides a tutorial of different workflows for the identification of spatially variable genes with different input type data. The page also reports the reference manual that describes the guideline of each function created with Roxygen. Roxygen is a tool that allows developers to build R software documenting the code as easily as possible. Although the package is a wrapper of Python functions in R language, the code of each function was developed with comprehensive documentation. In particular, Roxygen's

conventions were integrated concerning the documentation for the input parameters, the explanation of the outputs, the description of the function, and a minimal example. Following this code development design, the package contains a comprehensive guide of the provided functions, and it also helps both the user and the developer to correctly maintain and update the package during the time, as the Roxygen documentation helps both to understand the goal of a specific function and its structure.

In this project, I presented a wrapper package that allows users to execute in the R environment, a method originally designed for Python.

Since R is a popular language for data analysis, providing a wrapper of one of the most popular tools for the identification of spatially variable genes may help users to perform such analysis on SRT data imported in R. Also, considering both the support for the S4 class of SpatialExperiment and the already integrated plots function, it may further increase the usability of this package to all the users without advanced technical expertise.

References

1. Asp M, Bergenstråhle J, Lundeberg J. Spatially Resolved Transcriptomes—Next Generation Tools for Tissue Exploration. BioEssays. 2020. doi:10.1002/bies.201900221
2. Ståhl PL, Salmén F, Vickovic S, Lundmark A, Navarro JF, Magnusson J, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. Science. 2016. doi:10.1126/science.aaf2403
3. Lee JH. Quantitative approaches for investigating the spatial context of gene expression. Wiley Interdiscip Rev Syst Biol Med. 2017;9. doi:10.1002/wsbm.1369
4. Lederer AR, La Manno G. The emergence and promise of single-cell temporal-omics approaches. Current Opinion in Biotechnology. 2020. doi:10.1016/j.copbio.2019.12.005
5. Kulkarni A, Anderson AG, Merullo DP, Konopka G. Beyond bulk: a review of single cell transcriptomics methodologies and applications. Current Opinion in Biotechnology. 2019. doi:10.1016/j.copbio.2019.03.001
6. Aizarani N, Saviano A, Sagar, Mailly L, Durand S, Herman JS, et al. A human liver cell atlas reveals heterogeneity and epithelial progenitors. Nature. 2019;572. doi:10.1038/s41586-019-1373-2
7. Consortium TM, coordination O, coordination L, processing O collection and,

sequencing L preparation and, analysis C data, et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature. 2018;562.

8. Montoro DT, Haber AL, Biton M, Vinarsky V, Lin B, Birket SE, et al. A revised airway epithelial hierarchy includes CFTR-expressing ionocytes. Nature. 2018;560. doi:10.1038/s41586-018-0393-7

9. Plasschaert LW, Žilionis R, Choo-Wing R, Savova V, Knehr J, Roma G, et al. A single-cell atlas of the airway epithelium reveals the CFTR-rich pulmonary ionocyte. Nature. 2018;560. doi:10.1038/s41586-018-0394-6

10. Savulescu AF, Jacobs C, Negishi Y, Davignon L, Mhlanga MM. Pinpointing Cell Identity in Time and Space. Front Mol Biosci. 2020;7. doi:10.3389/fmolb.2020.00209

11. Teves JM, Won KJ. Mapping cellular coordinates through advances in spatial transcriptomics technology. Molecules and Cells. 2020.

12. Weber LM, Saha A, Datta A, Hansen KD, Hicks SC. nnSVG: scalable identification of spatially variable genes using nearest-neighbor Gaussian processes. bioRxiv. 2022; 2022.05.16.492124. doi:10.1101/2022.05.16.492124

13. Lee J, Yoo M, Choi J. Recent advances in spatially resolved transcriptomics: challenges and opportunities. BMB Rep. 2022;55. doi:10.5483/BMBRep.2022.55.3.014

14. Chen J, Suo S, Tam PP, Han JDJ, Peng G, Jing N. Spatial transcriptomic analysis of cryosectioned tissue samples with Geo-seq. Nat Protoc. 2017;12.

doi:10.1038/nprot.2017.003

15. Rodriques SG, Stickels RR, Goeva A, Martin CA, Murray E, Vanderburg CR, et al. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. Science (80-). 2019;363. doi:10.1126/science.aaw1219
16. Stickels RR, Murray E, Kumar P, Li J, Marshall JL, Di Bella DJ, et al. Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2. Nat Biotechnol. 2021;39. doi:10.1038/s41587-020-0739-1
17. Lubeck E, Coskun AF, Zhiyentayev T, Ahmad M, Cai L. Single-cell in situ RNA profiling by sequential hybridization. Nature Methods. 2014. doi:10.1038/nmeth.2892
18. Shah S, Lubeck E, Zhou W, Cai L. In Situ Transcription Profiling of Single Cells Reveals Spatial Organization of Cells in the Mouse Hippocampus. Neuron. 2016;92. doi:10.1016/j.neuron.2016.10.001
19. Righelli D, Weber LM, Crowell HL, Pardo B, Collado-Torres L, Ghazanfar S, et al. SpatialExperiment: infrastructure for spatially-resolved transcriptomics data in R using Bioconductor. Bioinformatics. 2022;38: 3128–3131. doi:10.1093/bioinformatics/btac299
20. Svensson V, Teichmann SA, Stegle O. SpatialDE: Identification of spatially variable genes. Nat Methods. 2018;15. doi:10.1038/nmeth.4636

VoyageR

1. Introduction

Spatially-resolved transcriptomics (SRT) refers to recently developed technologies that measure gene expression along with spatial coordinates of the measurements [1]. In tissue, spatial locations and gene expression levels of cells play a critical role in diseases, and this sequencing technology has been used to study the spatial landscape of gene expression, for example, in the brain and cancer [1]. One of the most common analyses, using both the coordinates and the expression measurements, is the identification of spatially variable genes (SVGs), which are genes that have a spatial pattern of expression variation. For a more complete introduction to the SVGs, refer to the introduction of SpatialDE chapter.

The present project aims to conduct a benchmarking of existing methods designed to identify SVGs. A comprehensive benchmark is still lacking and could be helpful for users to choose the best suitable procedures for their use. Furthermore, this kind of

study will enable comparisons across these technologies, in particular with regard to detection sensitivity, specificity, and capture efficiency [2]. Here I list the methods I have selected in this project, together with a brief description of their approach and the programming language used:

- BOOST-GP (R) [3], employs the framework of gaussian process (GP) to capture the spatial correlation. It directly models count data using a negative binomial (NB) distribution to account for the over-dispersion observed in real sequencing data.
- Giotto (R) [4], propose new methods based on statistical enrichment of binarized expression data in neighboring cells within the spatial network. Such network can be created by connecting neighboring cells through a Delaunay triangulation network, by selecting the k-nearest neighbors, or by using a fixed distance cut-off. For each gene, expression values are binarized using K-means clustering or simple thresholding on rank.
- GLISS (Python) [5], utilizes a graph-based association measure to select and link genes that are spatially dependent in both data sources.
- GPcounts (Python) [6], implements GP regression methods for modeling counts data using a negative binomial likelihood function.
- JSTA (Python) [7], identifies SVGs by determining if the spatial expression pattern of a given gene was statistically different from a null distribution by permuting the gene expression values.

- MERINGUE (R) [8], uses spatial auto-correlation and cross-correlation analyses. Given a set of spatial positions such as those corresponding to single cells, MERINGUE first represents these cells as neighborhoods using Voronoi tessellation, then cells are considered adjacent if their neighborhoods share an edge. The computation of SVGs is performed using Moran's I statistic.
- nnSVG (R) [1], is based on statistical advances in computationally scalable parameter estimation in spatial covariance functions in GPs using nearest neighbor Gaussian process (NNGP) models.
- RayleighSelection (R) [9], utilizes the combinatorial Laplacian score to rank and disaggregate genes according to their spatial expression pattern.
- scGCO (Python) [10], a method based on fast optimization of Markov Random Fields with graph cuts. The authors describe that a crucial insight of their method is that identifying spatial genes is analogous to identifying objects from an image, which is typically optimally solved in computer vision problems.
- Seurat (R) [11], models spatial transcriptomics data as a mark point process and computes a variogram, which identifies genes whose expression level is dependent on their spatial location. Alternatively, Moran's I method can be used to identify SVGs, as spatially patterned genes also exhibit autocorrelation.
- SingleCellHyastack (R) [12], predicts DEGs (Differentially Expressed Genes) using the Kullback–Leibler divergence to find genes that are expressed in subsets of cells that are non-randomly positioned in a multidimensional space.

- SOMDE (Python) [13], uses a self-organizing map to cluster neighboring cells into nodes and then uses a Gaussian Process to fit the node-level spatial gene expression to identify SVGs.
- SPARK (R) [14], directly models spatial count data through a generalized linear spatial model (GLSM) with a variety of spatial kernels to accommodate count data generated. It relies on recently developed statistical formulas for hypothesis testing, providing effective control of type I errors and yielding high statistical power.
- SPARK-X (R) [15], builds upon a robust covariance test framework to model a wide variety of spatial transcriptomics data collected through different technologies. It relies on algebraic innovations for scalable computation as well as newly developed statistical formulas for hypothesis testing, producing well-calibrated p-values and yielding high statistical power. SPARK-X is highly computationally efficient and the only SE method scalable for the HDST (High-definition spatial transcriptomics) data.
- SpatialDE (Python) [16], uses a geostatistics class of models, testing whether gene expression levels at different locations covary in a manner that depends on their relative location, a thus spatially variable.
- Squidpy (Python) [17], provides an approach based on the well-known Moran's I statistics.

Moses and Pachter [18] recently published a curated review of literature on spatial transcriptomics, including a review of data analysis methods. However, no paper was published in any peer-reviewed journal performing a benchmark similar to the one developed in this study. For this reason, in this section of my Ph.D. thesis, I'm going to present materials and methods, and the first results obtained from this project (for both Python and R methods). I believe that the evaluation of the results' reliability, in addition to the report of the used memory resources, can be an important guide also for developers, as it allows them to compare their methods to the existing ones. Indeed, considering that the identification of the SVGs is one of the most popular analyses that can be performed on SRT data, this area of research will likely remain under active investigation for the near future.

To further simplify the comparison of different methods, we have designed the benchmark in a modular fashion. A user creating his own method could upload a minimal workflow to an ad-hoc system, run his pipeline and successively evaluate his approach both for the results and technical factors, like the computation time or RAM used, and compare his method to the ones already evaluated.

2. Materials and Methods

One of the main objectives of this project was to create a benchmarking system able to execute reproducible runs of methods that identify spatially variable genes.

To achieve this goal each package pipeline was built to be executed in closed container environments like Docker and Singularity. Indeed, both software makes it possible to create and run containers including software packages in a way that is portable and reproducible. Singularity was specifically designed to run containers in large HPC clusters. To build this closed environment with all the requirements, I created a Docker image of the Debian Linux system with R and Python libraries and the GitHub repository of the packages. Indeed, in the repository of this project (https://github.com/sales-lab/voyageR), there is a Dockerfile that allows the creation of the docker image.

Another important setup for conducting a benchmark is to define a common input data format to facilitate the execution on different datasets, as it is essential to evaluate pipelines on a diverse set of biological data. As previously described, many published methods perform the identification of SVGs with a different strategy, each with a specific input type too. For this reason, all the workflows are run starting from a

SpatialExperiment object (SpE) [19], even for Python packages. SpatialExperiment, as described in the SpatialDE project, is an infrastructure R package that allows storing data from Spatially Resolved Transcriptomic data, independently from the source technology. It provides a robust infrastructure that simplifies operations on the data to the user. Through the package *'zellkonverter'* [19], SpE can be also saved on the disk in H5AD format and successively loaded in Python as AnnData format [20], which can be considered as a Python infrastructure to handle annotated data matrices in memory and on disk.

Furthermore, Bioconductor has packages like *'TENxVisiumData'* or *'spatialLIBD'* that collect 10x Genomics Visium datasets prepared as objects of class SpatialExperiment, which makes SpE the perfect input format to be able to use method pipelines with different biological datasets, without changing code.

The benchmark is performed for the following packages:

- BOOST–GP (R)
- Giotto (R)
- GLISS (Python)
- GPcounts (Python)
- JSTA (Python)
- MERINGUE (R)
- nnSVG (R)
- RayleighSelection (R)

- scGCO (Python)
- Seurat (R)
- SingleCellHyastack (R)
- SOMDE (Python)
- Spark (R)
- Spark-X (R)
- SpatialDE (Python)
- Squidpy (Python)

For each of these methods, the corresponding workflow was executed starting from a common input format. In particular, for R packages the input format is the SpatialExperiment object, while Python methods take in input an AnnData, loaded from the SpE saved on disk as H5AD format.

However, *Seurat* and *SingleCellHaystack* packages run their methods using a SeuratObject, an alternative S4 class for single-cell genomic data and associated information, such as dimensionality reduction embeddings, nearest-neighbor graphs, and spatially-resolved coordinates. For this reason, for these two packages, the SpE object was converted to a SeuratObject by a script.

At the moment, the pipelines are run on the human brain dataset from Maynard et al [21]. It is provided by the R package 'spatialLIBD' directly as a SpE object and contains 12 samples. For each sample, a filtering procedure at a different threshold was applied,

to remove low-expressed genes and/or mitochondrial genes. In particular, filtering retains genes containing at least X expression counts in at least Y percent of the total number of spatial locations (spots), with the following values of X and Y:

- X = 3; Y = 0.1;
- X = 3; Y = 0.3;
- X = 3; Y = 0.5.

Since a ground truth about spatially variable genes is not available for all datasets, it is important to evaluate the reliability of the results using statistical measures and plots which are independent from such information.

Each sample was thus duplicated, and its coordinates were shuffled to remove any spatial pattern of expression. In particular, before shuffling the locations, these datasets were duplicated and filtered with new threshold values, in addition to the ones previously shown:

- X = 2; Y = 0.1;
- X = 2; Y = 0.3;
- X = 4; Y = 0.1;
- X = 4; Y = 0.3.

This procedure was performed to produce ten replicates of each filtered dataset, and thus for each sample, with the expectation that spatial patterns of expression were removed.

All the method's pipelines were run on all these datasets.

Another interesting benchmark parameter is the evaluation of the time computing of the methods, as a faster pipeline even with non-optimal results may be a good trade-off in specific use cases. Thus, both the computation time and memory efficiency are parameters that are going to be included in the results of this project. For this purpose, all the workflows were controlled by the software 'GNU time' which allows for running other programs and then displays information about the resources used and collected by the system while the program was running.

An example verbose (`-v`) of the collected output of 'GNU time' is shown as follows, performed on four seconds of sleep of the system:

```
$ time -v sleep 4
        Command being timed: "sleep 4"
        User time (seconds): 0.00
        System time (seconds): 0.05
        Percent of CPU this job got: 1%
        Elapsed (wall clock) time (h:mm:ss or m:ss): 0:04.26
        Average shared text size (kbytes): 36
        Average unshared data size (kbytes): 24
        Average stack size (kbytes): 0
        Average total size (kbytes): 60
        Maximum resident set size (kbytes): 32
        Average resident set size (kbytes): 24
        Major (requiring I/O) page faults: 3
```

```
Minor (reclaiming a frame) page faults: 0

Voluntary context switches: 11

Involuntary context switches: 0

Swaps: 0

File system inputs: 3

File system outputs: 1

Socket messages sent: 0

Socket messages received: 0

Signals delivered: 1

Page size (bytes): 4096

Exit status: 0
```

Two parameters that will be integrated into the project results are the 'Maximum resident set size (kbytes)' and the 'Elapsed (wall clock) time (h:mm:ss or m:ss)', which describes the maximum RAM used in kilobytes and the execution time of the program, respectively.

Each method was run on all the created datasets using Singularity containers.

3. Preliminary results and conclusions

This project is a collaboration with the Department of Statistics, at the University of Padova, and it is still a work in progress. We nonetheless have already gathered some preliminary results concerning technical aspects, essentially to build the scientifically reproducible benchmarking system.

After some initial test runs, I excluded from the comparisons the methods BOOST-GP, GPcounts, JSTA, and RayleighSelection, due to their very high computation time. None of the above methods was able to process a single sample in less than 4 days, compared to some hours for the other methods.

We then started to evaluate the results by measuring the False Positive Rate of SVG calls, producing "box-plots" where each box describes the distribution of the percentage of the SVGs with a p-value lower than the threshold on the x-axis. Fig. 3.1 shows the chart for results on normal datasets (thus without shuffled locations) with a filter to retain genes containing at least 3 expression counts in at least 0.5 percent of the total number of spatial locations (spots). Some methods were able to recognize a

reasonable number of SVGs, while others like Giotto (binSpect k-means), Seurat (Moran's I), or SPARK-X identified the majority of SVGs with a p-value lower than the threshold, suggesting an unreliable control of type I errors.

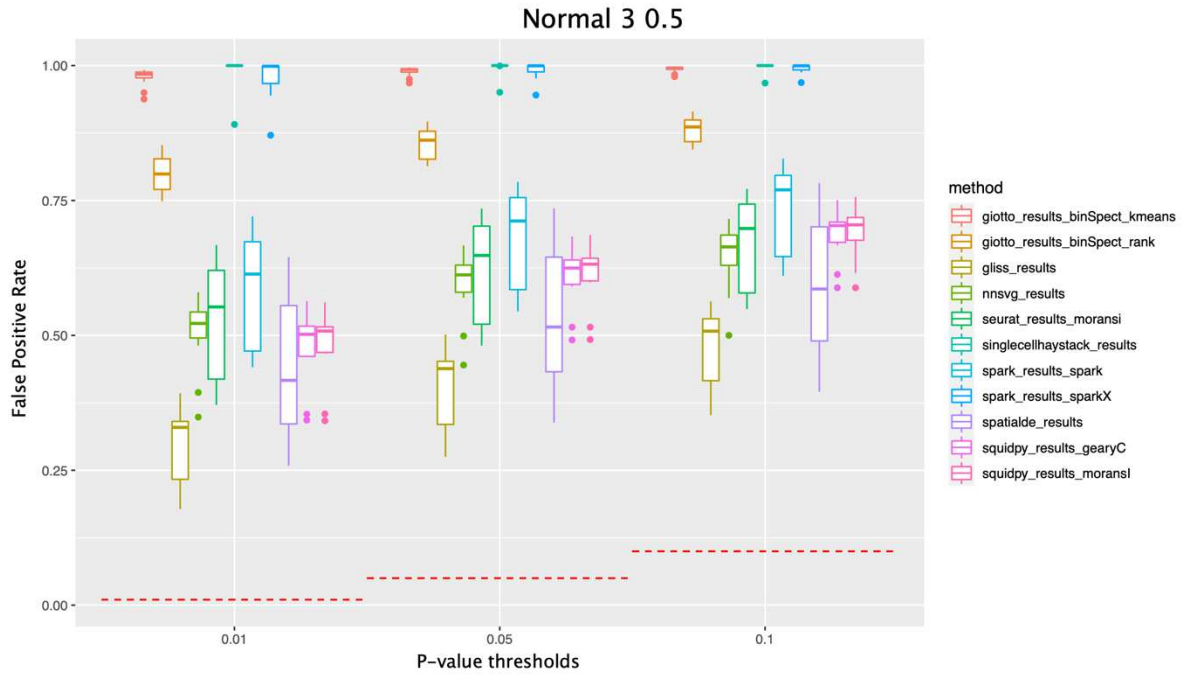


Fig. 3.1: Box-plot of the results on normal datasets filtered to retain genes containing at least 3 expression counts in at least 0.5 percent of the total number of spatial locations (spots).

This hypothesis was supported also by the box plots obtained from the shuffle datasets with the same filtering threshold. We stress that in this dataset the locations were shuffled to remove any spatial pattern of expression. It is thus expected that no SVGs are present in these samples. The distribution of p-values for reliable methods should appear in the following plots as close as possible to the selected threshold.

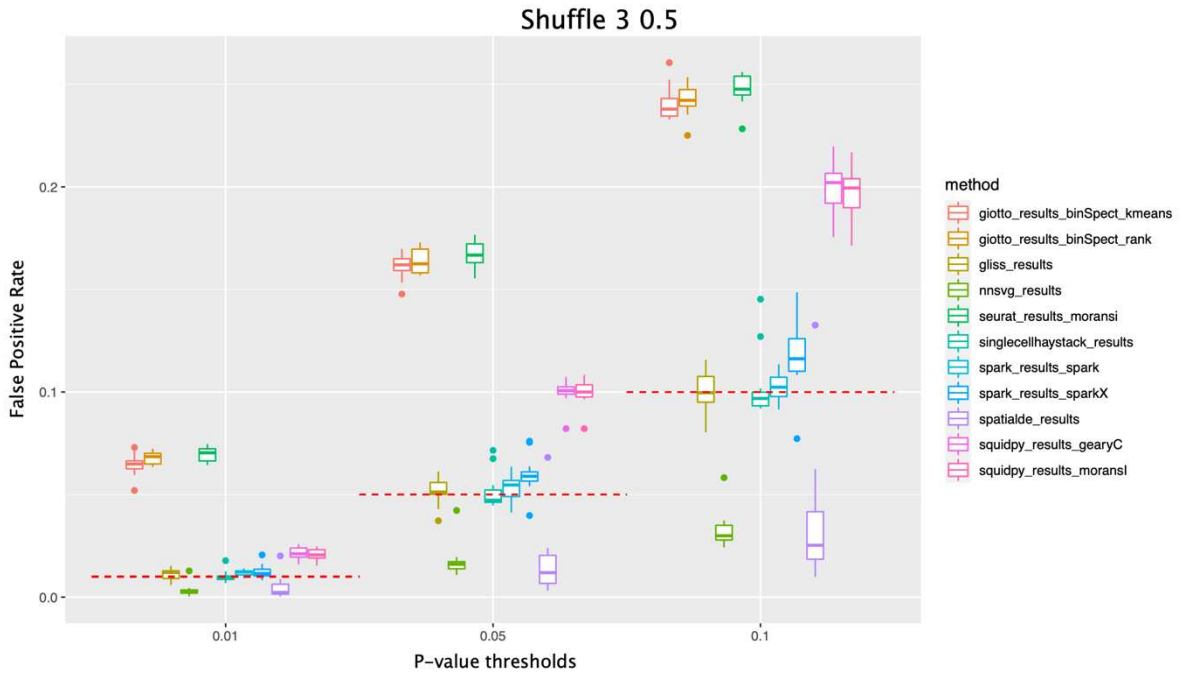


Fig. 3.2: Box-plot of the results on shuffle datasets filtered to retain genes containing at least 3 expression counts in at least 0.5 percent of the total number of spatial locations (spots).

This characteristic was recognized by nnSVG and SpatialDE, which identified a few genes with a p-value lower than the threshold. Indeed, these two methods maintained a false positive rate below all the critical values and controlled the type-I error. Especially for nnSVG, this aspect was also observed in plots of the shuffle datasets with different filtering values (shown below), while SpatialDE seems to lose it with less stringent filtering.

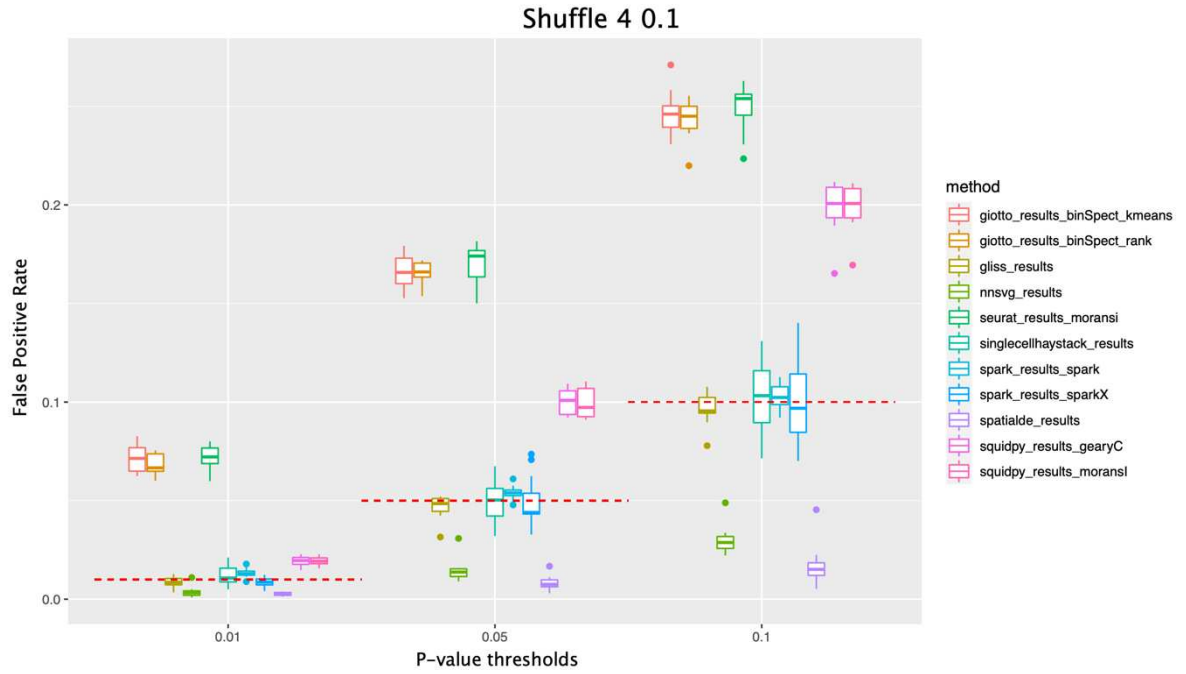


Fig. 3.3: Box-plot of the results on shuffle datasets filtered to retain genes containing at least 4 expression counts in at least 0.1 percent of the total number of spatial locations (spots).

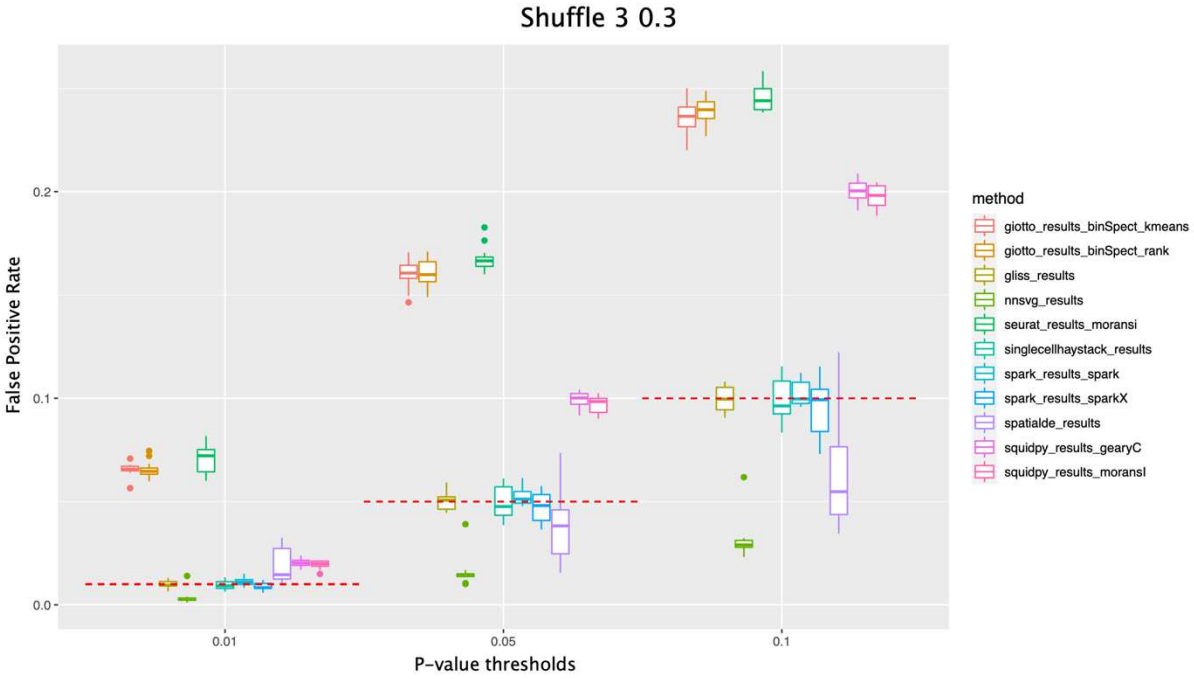


Fig. 3.4: Box-plot of the results on shuffle datasets filtered to retain genes containing at least 3 expression counts in at least 0.3 percent of the total number of spatial locations (spots).

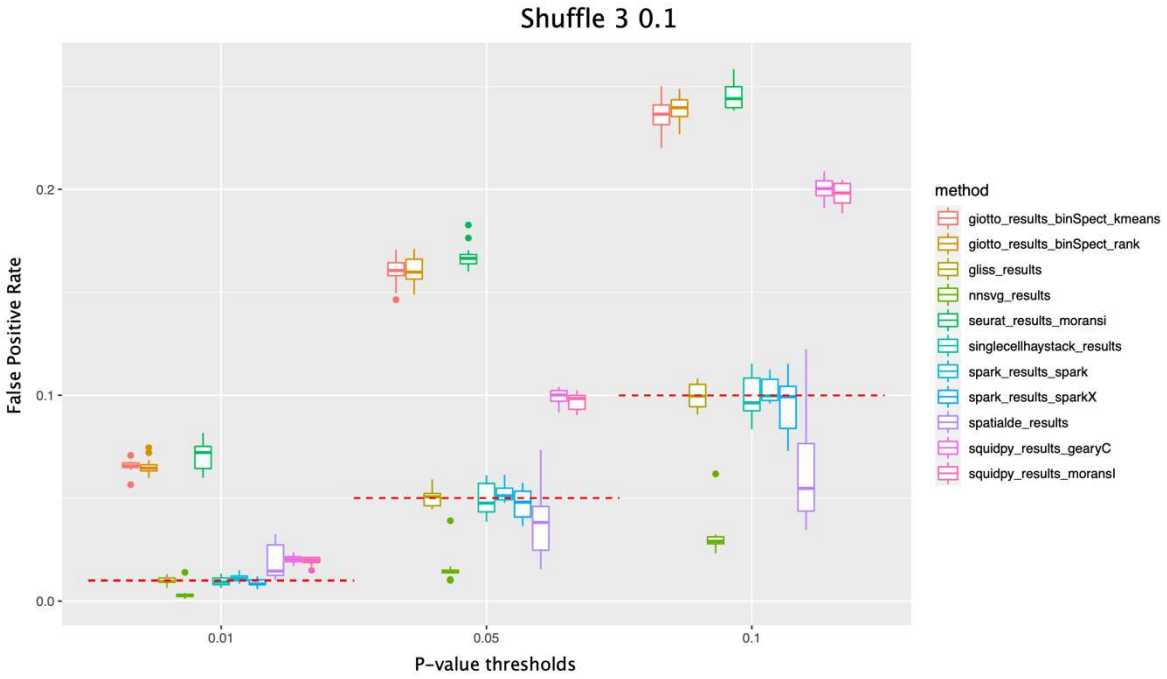


Fig. 3.5: Box-plot of the results on shuffle datasets filtered to retain genes containing at least 3 expression counts in at least 0.1 percent of the total number of spatial locations (spots).

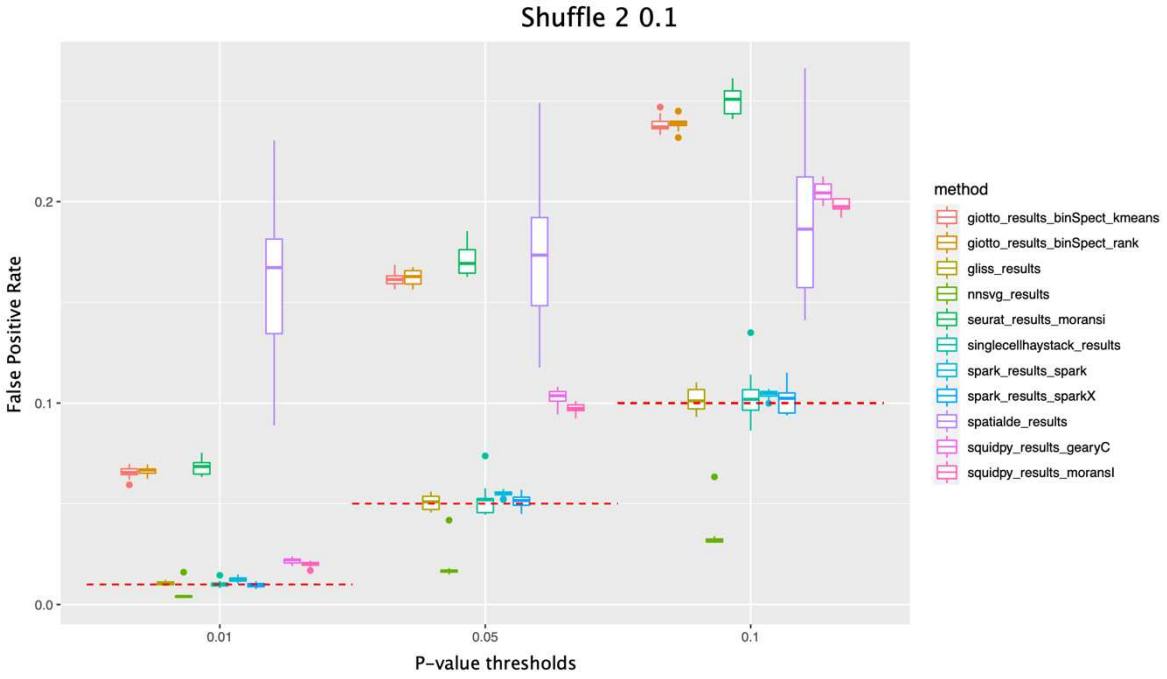


Fig. 3.6: Box-plot of the results on shuffle datasets filtered to retain genes containing at least 2 expression counts in at least 0.1 percent of the total number of spatial locations (spots).

At the time of writing, we are still fixing technical issues with the MERINGUE pipeline, which made it impossible to run it on all samples.

As described, even though technical aspects are the majority of the results of this project, we already conducted some statistical measures to evaluate the reliability of the results. Other analyzes, plots, and statistical measurements will be needed, however, some methods have already shown an interesting signal regarding the reliability of identifying spatially variable genes. Moreover, even if we are comparing various pipelines, each one with a different strategy, the made-up benchmarking systems allow the possibility to easily include in future new methods e comparing the results and the performance with the workflows already included. All the methods were run saving in a file performance information about the computation time, CPU, and RAM used. After completing the statistical evaluation of the results, we will start to compare the memory and resource usage.

References

1. Weber LM, Saha A, Datta A, Hansen KD, Hicks SC. nnSVG: scalable identification of spatially variable genes using nearest-neighbor Gaussian processes. bioRxiv. 2022; 2022.05.16.492124. doi:10.1101/2022.05.16.492124
2. Atta L, Fan J. Computational challenges and opportunities in spatially resolved transcriptomic data analysis. Nature Communications. 2021. doi:10.1038/s41467-021-25557-9
3. Li Q, Zhang M, Xie Y, Xiao G. Bayesian modeling of spatial molecular profiling data via Gaussian process. Bioinformatics. 2021. doi:10.1093/bioinformatics/btab455
4. Dries R, Zhu Q, Dong R, Eng CHL, Li H, Liu K, et al. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. Genome Biol. 2021;22. doi:10.1186/s13059-021-02286-2
5. Zhu J, Sabatti C. Integrative Spatial Single-cell Analysis with Graph-based Feature Learning. bioRxiv. 2020.
6. BinTayyash N, Georgaka S, John ST, Ahmed S, Boukouvalas A, Hensman J, et al. Non-parametric modelling of temporal and spatial counts data from RNA-seq experiments. Bioinformatics. 2021;37. doi:10.1093/bioinformatics/btab486
7. Littman R, Hemminger Z, Foreman R, Arneson D, Zhang G, Gómez-Pinilla F, et al. Joint cell segmentation and cell type annotation for spatial transcriptomics.

Mol Syst Biol. 2021;17. doi:10.15252/msb.202010108

8. Miller BF, Bambah-Mukku D, Dulac C, Zhuang X, Fan J. Characterizing spatial gene expression heterogeneity in spatially resolved single-cell transcriptomic data with nonuniform cellular densities. Genome Res. 2021;31. doi:10.1101/gr.271288.120

9. Govek KW, Yamajala VS, Camara PG. Clustering-independent analysis of genomic data using spectral simplicial theory. PLoS Comput Biol. 2019;15. doi:10.1371/journal.pcbi.1007509

10. Zhang K, Feng W, Wang P. Identification of spatially variable genes with graph cuts. bioRxiv. 2018.

11. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell. 2021;184. doi:10.1016/j.cell.2021.04.048

12. Vandenbon A, Diez D. A clustering-independent method for finding differentially expressed genes in single-cell transcriptome data. Nat Commun. 2020;11. doi:10.1038/s41467-020-17900-3

13. Hao M, Hua K, Zhang X. SOMDE: a scalable method for identifying spatially variable genes with self-organizing map. Bioinformatics. 2021;37. doi:10.1093/bioinformatics/btab471

14. Sun S, Zhu J, Zhou X. Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies. Nat Methods. 2020;17.

doi:10.1038/s41592-019-0701-7

15. Zhu J, Sun S, Zhou X. SPARK-X: non-parametric modeling enables scalable and robust detection of spatial expression patterns for large spatial transcriptomic studies. Genome Biol. 2021;22. doi:10.1186/s13059-021-02404-0

16. Svensson V, Teichmann SA, Stegle O. SpatialDE: Identification of spatially variable genes. Nat Methods. 2018;15. doi:10.1038/nmeth.4636

17. Palla G, Spitzer H, Klein M, Fischer D, Schaar AC, Kuemmerle LB, et al. Squidpy: a scalable framework for spatial omics analysis. Nat Methods. 2022;19. doi:10.1038/s41592-021-01358-2

18. Moses L, Pachter L. Museum of spatial transcriptomics. Nature Methods. 2022. doi:10.1038/s41592-022-01409-2

19. Righelli D, Weber LM, Crowell HL, Pardo B, Collado-Torres L, Ghazanfar S, et al. SpatialExperiment: infrastructure for spatially-resolved transcriptomics data in R using Bioconductor. Bioinformatics. 2022;38: 3128–3131. doi:10.1093/bioinformatics/btac299

20. Virshup I, Rybakov S, Theis FJ, Angerer P, Wolf FA. anndata: Annotated data. bioRxiv. 2021; 2021.12.16.473007. doi:10.1101/2021.12.16.473007

21. Maynard KR, Collado-Torres L, Weber LM, Uytingco C, Barry BK, Williams SR, et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. Nat Neurosci. 2021;24. doi:10.1038/s41593-020-00787-0