OmicsNet and OmicsNetR

a web server project

Mateusz Wiktorzak

320668, IAD, MiNI PW

Table of contents

1. Introduction

OmicsNet 2.0 is the newest version of OmicsNet web server platform, which the report is about. It allows users to easily build, visualize, and analyse multi-omics networks to study rich relationships among lists of omics features of interest.

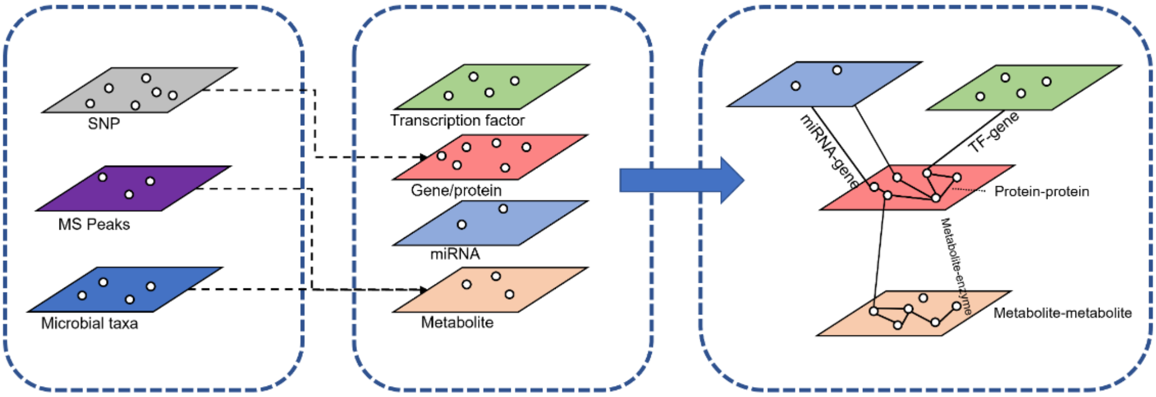
There is a growing realization that genetic variation only partially explains complex diseases such as common cancers, type 2 diabetes, heart diseases, etc. Recent years have seen increasing applications of multi-omics approaches to augment genomics with various other omics such as epigenomics, transcriptomics, proteomics, metabolomics, and microbiomics. The resulting heterogenous datasets generated from these studies have posed significant bioinformatics challenges for proper analysis, integration and interpretation. To address these needs, many different methods and tools have been developed in recent years. Biological networks such as protein–protein interaction (PPI) networks, gene regulatory networks, or biochemical reaction networks provide a conceptual and intuitive framework for integrating results from multi-omics studies. This approach involves two key procedures - network creation and network analysis. Building high-quality networks with intuitive visual presentation play a significant role in interpreting multi-omics data.

2. OmicsNet functioning

It is important to mention that omics relate to disciplines with „-omics” suffix, such as genomics, proteomics, metabolomics, metagenomics, phenomics, transcriptomics – studies about molecules living within living organisms.

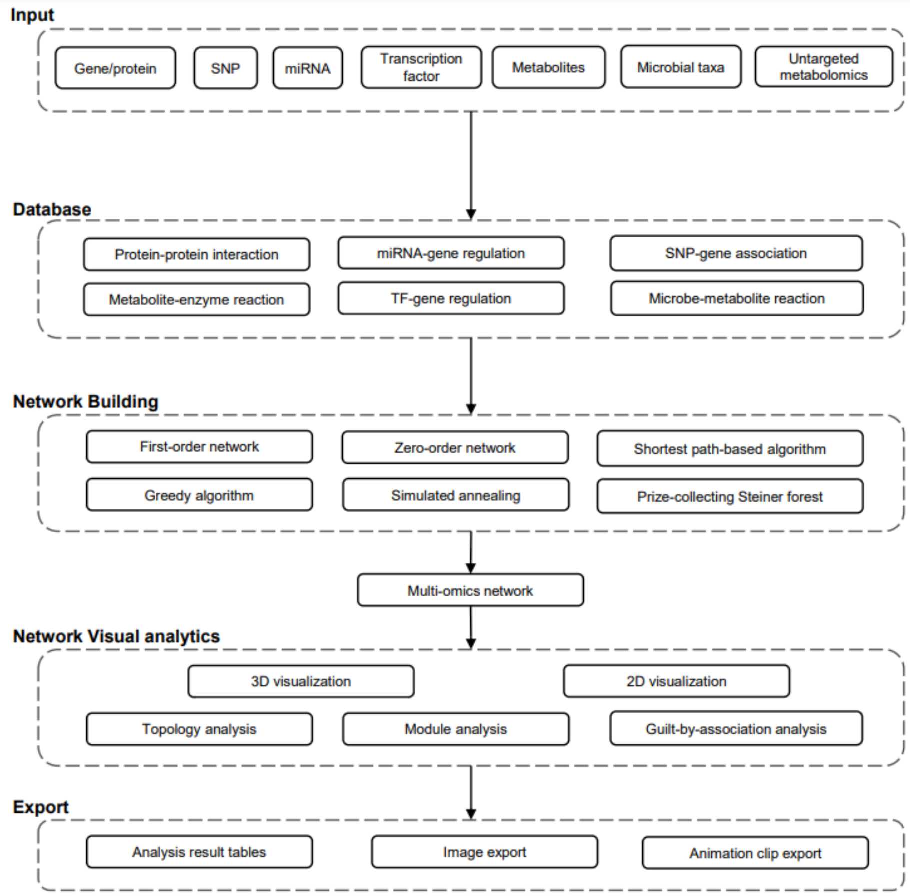
2.1. Potential and workflow of OmicsNet

OmicsNet is a powerful tool designed for the exploration of multi-omics data in the context of molecular interaction knowledge. It allows users to input one or multiple lists of mRNAs, transcription factors (TFs), miRNAs, metabolites, microbial taxa, MS peaks, or SNP data. By using various molecular interaction databases encompassing protein-protein interactions (PPI), TF-gene interactions, miRNA-mRNA interactions, and metabolite-protein interactions, OmicsNet enables the creation of diverse biological networks.



The workflow of OmicsNet can be summarized into four steps - data upload, database selection, network creation and network visual analytics. Users can upload as mentioned before, lists of genes, proteins, transcription factors, miRNAs, metabolites, LC-MS peaks, microbial taxa, or SNPs to search different molecular interaction databases or perform annotation. The results will be merged to create multi-omics networks which can be optionally customization using various methods.

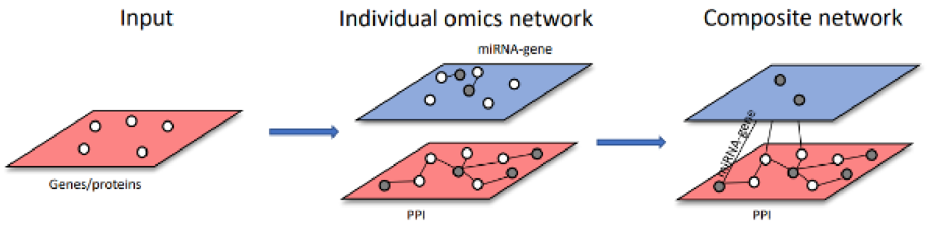
The subnetworks can be explored in 2D or 3D space with comprehensive built-in support for layouts, network analysis and functional analysis. Such visual representations aid in the exploration and interpretation of the generated results. OmicsNet empowers researchers and analysts to uncover meaningful relationships, interactions, and patterns within complex multi-omics datasets, ultimately facilitating a deeper understanding of molecular mechanisms and biological processes. Above process is shown on schema below.

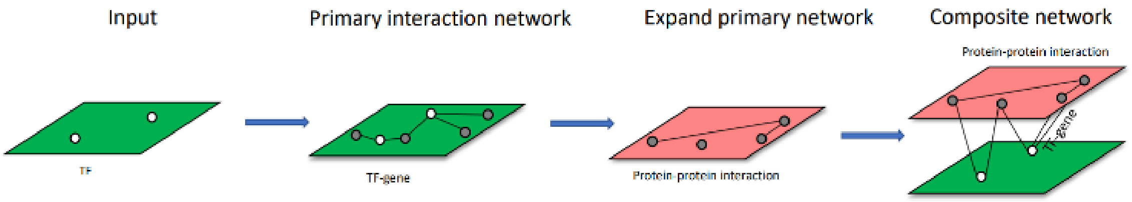

Input
Gene/protein
SNP
miRNA
Transcription factor
Metabolites
Microbial taxa
Untargeted metabolomics
Database
Protein-protein interaction
miRNA-gene regulation
SNP-gene association
Metabolite-enzyme reaction
TF-gene regulation
Microbe-metabolite reaction
Network Building
First-order network
Zero-order network
Shortest path-based algorithm
Greedy algorithm
Simulated annealing
Prize-collecting Steiner forest
Multi-omics network
Network Visual analytics
3D visualization
2D visualization
Topology analysis
Module analysis
Guilt-by-association analysis
Export
Analysis result tables
Image export
Animation clip export

2.2. Building of networks from lists.

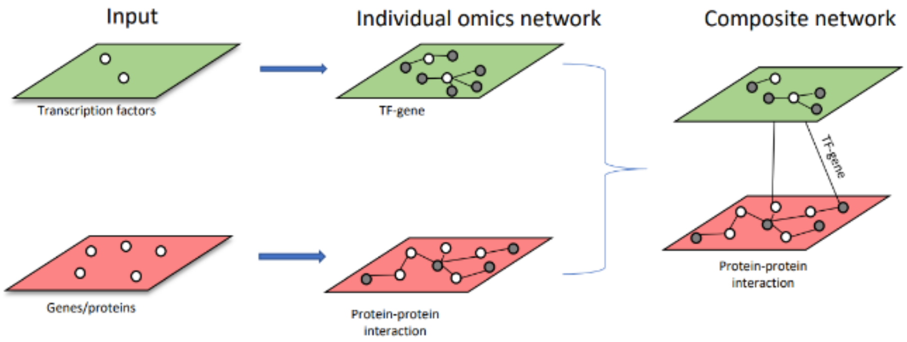
It is important to point out how network is built for different types of lists. When it comes to a single list, if gene/protein list is uploaded the seed gene/proteins can be used to query one or multiple interaction networks based on interaction type and database selected. The interaction networks will be merged into connected multi-omics subnetworks (picture on the top).

If features from omics type other than gene/protein are uploaded, network building is an iterative process where primary interaction network is used to expand the composite network. First primary interaction network is composed of seeds and its immediate interacting partners. Secondary interactions will query for interactions against molecules contained in the primary network, not only seeds. PPI as secondary or tertiary interaction will add edges to existing gene/proteins in the network by default. All other interaction types would add new edges and nodes (picture on the bottom).

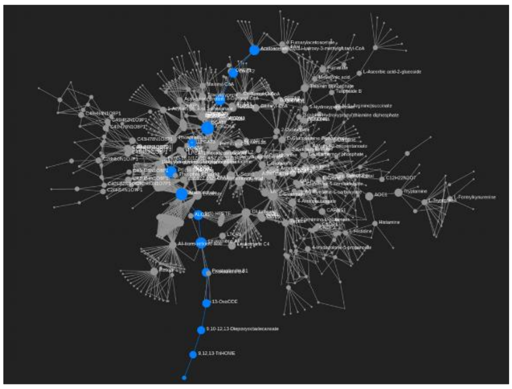



Input
Primary interaction network
Expand primary network
Composite network
Protein-protein interaction
TF
TF-gene
Protein-protein interaction
TF-gene

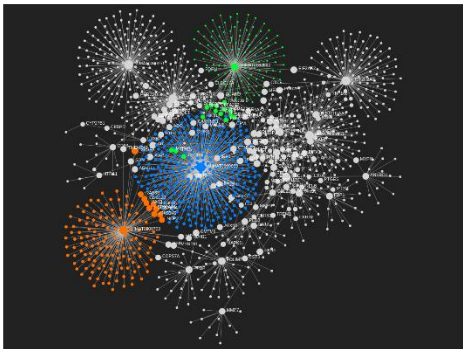
With multiple lists: each list of molecules are used to build individual omics interaction network. Then the individual omics networks are merged to form composite network through shared nodes. It is show on the picture below.



2.3. Analysis in OmicsNet



Highlighting the shortest path between two entities within the network can be a useful feature for understanding the potential interactions and relationships between them. The shortest path represents the most direct and efficient route connecting two nodes in the network. To highlight a shortest path, users typically start by selecting two entities of interest, such as genes, proteins etc. These entities can be identified either by their names or unique identifiers. Highlighting can be done by emphasizing the nodes and edges along the path, making it easier for users to identify and analyse the connections between the selected entities. This feature aids in understanding the potential functional or regulatory relationships that exist between the entities and can provide insights into the underlying biological mechanisms.

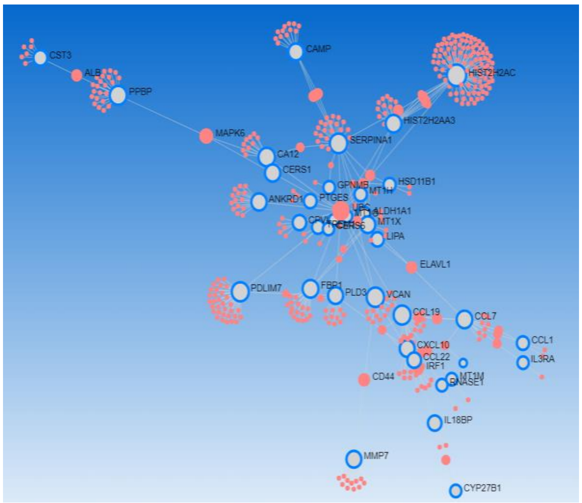


On the other hand, highlighting the top modules is a useful feature for identifying cohesive and functionally related groups of nodes within the network. Modules, also known as clusters or communities, are sets of nodes that exhibit strong interconnectivity and are often associated with shared biological functions or pathways. To highlight the top modules in OmicsNet, various graph clustering algorithms are typically employed. Additionally, OmicsNet often provides summary statistics or metrics for each module, such as enrichment scores or functional annotations, to provide insights into the biological significance of the modules. By highlighting the top modules in OmicsNet, users can effectively identify groups of nodes that are functionally related. This feature allows for a more focused and meaningful analysis of the network and aiding in the understanding of complex biological processes.

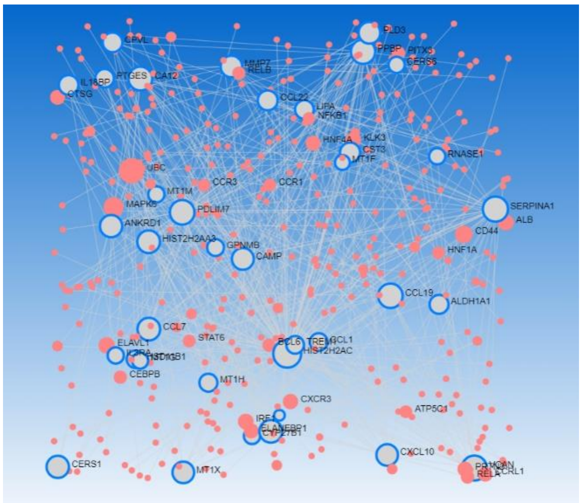
2.4. Types of layouts.

In OmicsNet 1.0 only 3D layouts were available. However OmicsNet 2.0 significantly enhanced overall network visual analytics by implementing 2D network visualization with 11 different graph layouts, together with a novel 3D network module layout. Some of these layouts are shown below.

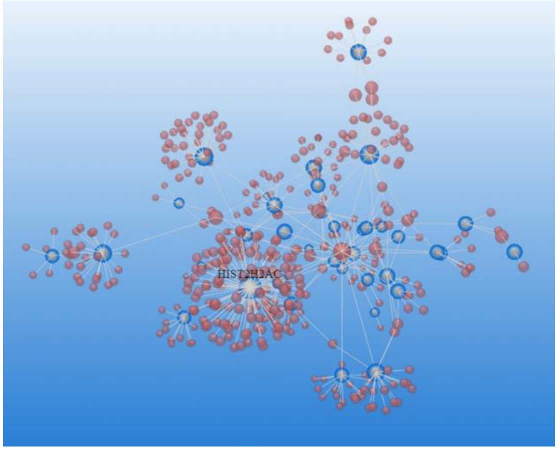
Auto Layout (2D)



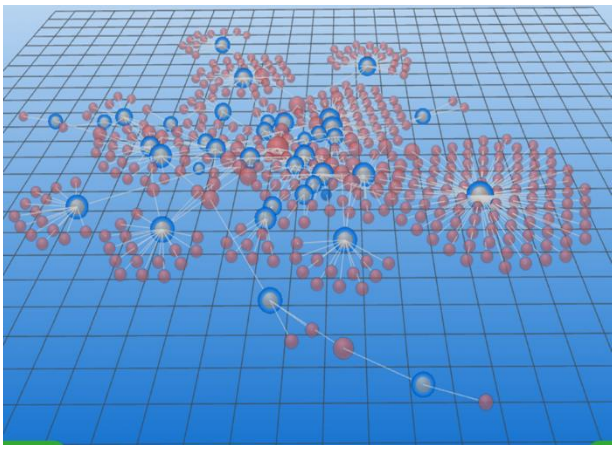
Random Layout (2D)



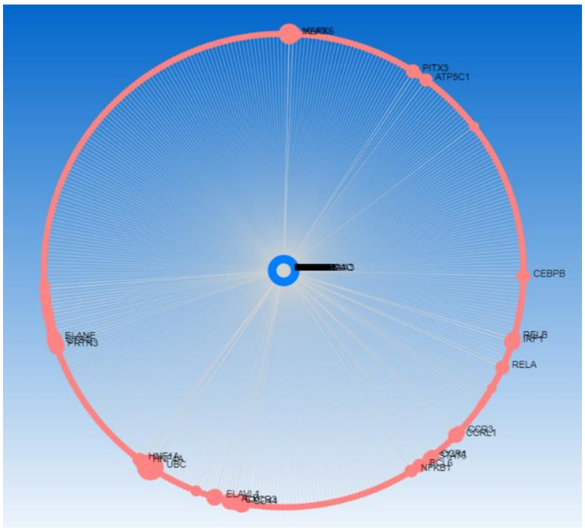
Standard (3D)



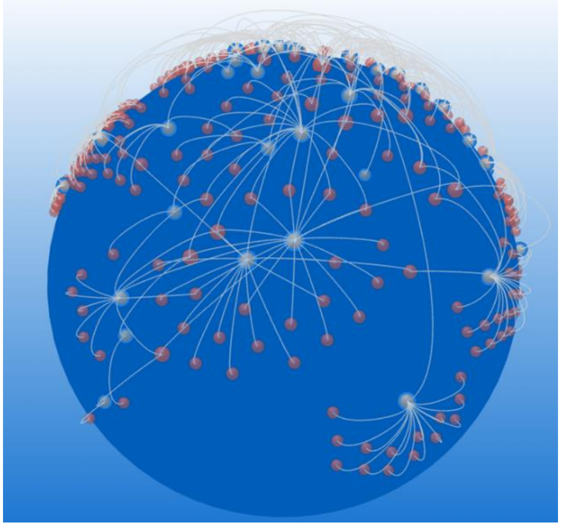
Perspective (3D)



Circular Bipartite (2D)



Spherical (3D)

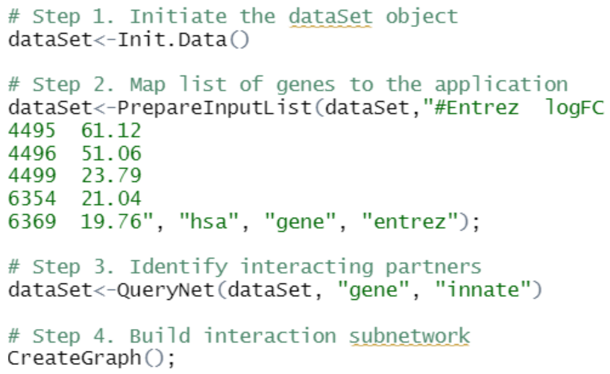


2.5. OmicsNetR

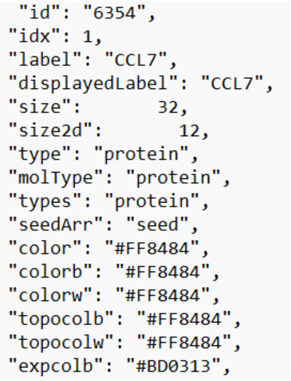
OmicsNetR is the R package that is in sync with the OmicsNet web server, providing a powerful platform for network-based integration and interpretation of multi-omics data at a systems level. This package consists of essential R functions required for creating, trimming, and analyzing networks.

By installing and loading OmicsNetR, users can replicate the results obtained from the web server on their local machines. They can access the R command history downloaded from the OmicsNet website, enabling them to run the R functions and gain more flexibility and reproducibility in their analyses. Utilizing OmicsNetR allows users to harness the capabilities of the R programming language, leveraging its extensive ecosystem of statistical and computational tools for in-depth exploration and interpretation of multi-omics networks.

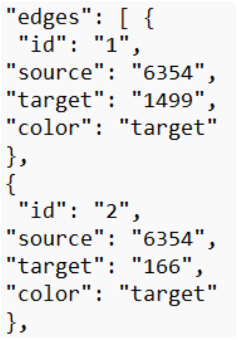
Below are shown sequentially sample code, part of node attributes and edges.



```
# Step 1. Initiate the dataSet object
dataSet<-Init.Data()

# Step 2. Map list of genes to the application
dataSet<-PrepareInputList(dataSet,"#Entrez  logFC
4495   61.12
4496   51.06
4499   23.79
6354   21.04
6369   19.76", "hsa", "gene", "entrez");

# Step 3. Identify interacting partners
dataSet<-QueryNet(dataSet, "gene", "innate")

# Step 4. Build interaction subnetwork
CreateGraph();
```



```
 "id": "6354",
"idx": 1,
"label": "CCL7",
"displayedLabel": "CCL7",
"size":       32,
"size2d":       12,
"type": "protein",
"molType": "protein",
"types": "protein",
"seedArr": "seed",
"color": "#FF8484",
"colorb": "#FF8484",
"colorw": "#FF8484",
"topocolb": "#FF8484",
"topocolw": "#FF8484",
"expcolb": "#BD0313",
```

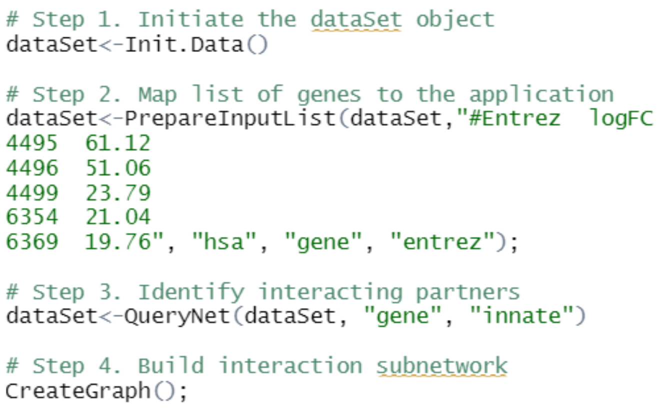


```
"edges": [ {
 "id": "1",
"source": "6354",
"target": "1499",
"color": "target"
},
{
 "id": "2",
"source": "6354",
"target": "166",
"color": "target"
},
```

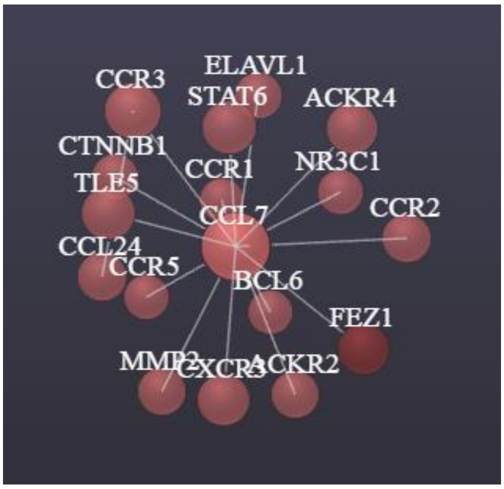
2.6. Examples

A few OmicsNet visualizations as well as code required to produce them, are shown below to have an idea what OmicsNet is creating.

- List of genes

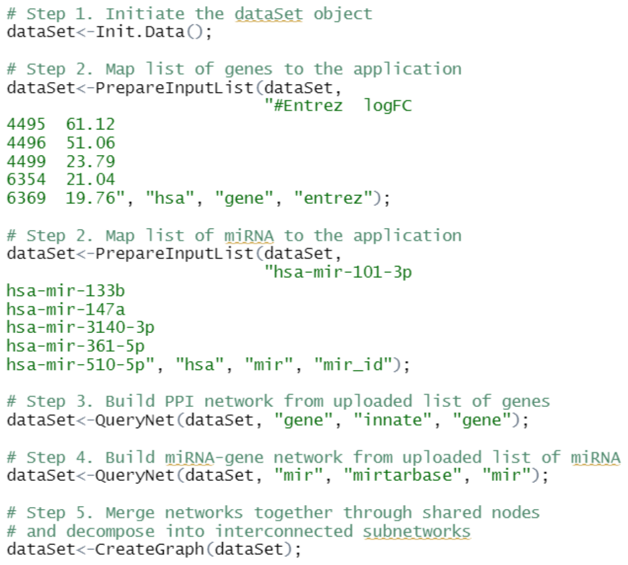


```
# Step 1. Initiate the dataSet object
dataSet<-Init.Data()

# Step 2. Map list of genes to the application
dataSet<-PrepareInputList(dataSet,"#Entrez  logFC
4495  61.12
4496  51.06
4499  23.79
6354  21.04
6369  19.76", "hsa", "gene", "entrez");

# Step 3. Identify interacting partners
dataSet<-QueryNet(dataSet, "gene", "innate")

# Step 4. Build interaction subnetwork
CreateGraph();
```

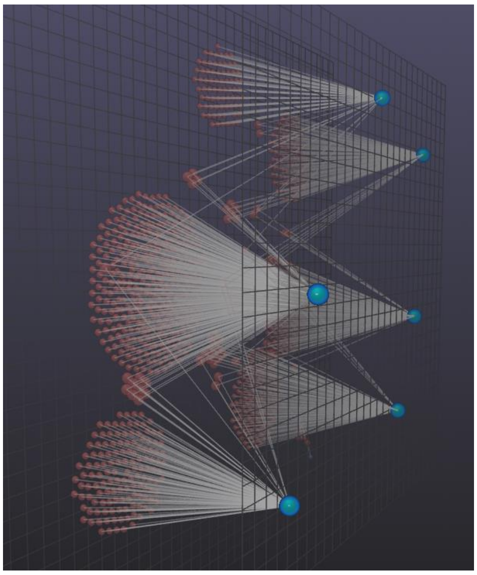


That is the simplest network that can be made in OmicsNet. As an input serves list of genes, then OmicsNet builds edges in between them. Finally JSON file with such graph is generated, so that it can be visualized on the platform. Visualization is presented in Standard 3D Layout.

- Integration of genes and miRNA

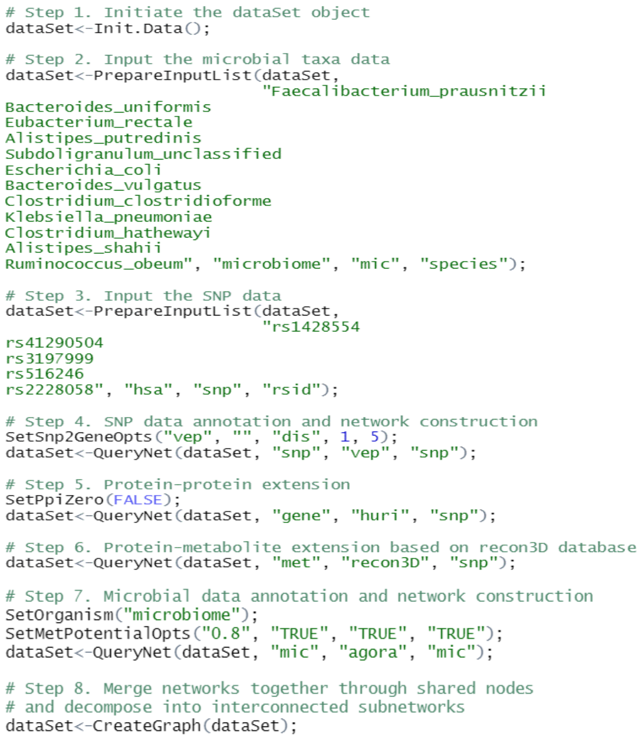


```
# Step 1. Initiate the dataSet object
dataSet<-Init.Data();

# Step 2. Map list of genes to the application
dataSet<-PrepareInputList(dataSet,
                          "#Entrez  logFC
4495  61.12
4496  51.06
4499  23.79
6354  21.04
6369  19.76", "hsa", "gene", "entrez");

# Step 2. Map list of miRNA to the application
dataSet<-PrepareInputList(dataSet,
                          "hsa-mir-101-3p
hsa-mir-133b
hsa-mir-147a
hsa-mir-3140-3p
hsa-mir-361-5p
hsa-mir-510-5p", "hsa", "mir", "mir_id");

# Step 3. Build PPI network from uploaded list of genes
dataSet<-QueryNet(dataSet, "gene", "innate", "gene");

# Step 4. Build miRNA-gene network from uploaded list of miRNA
dataSet<-QueryNet(dataSet, "mir", "mirtarbase", "mir");

# Step 5. Merge networks together through shared nodes
# and decompose into interconnected subnetworks
dataSet<-CreateGraph(dataSet);
```

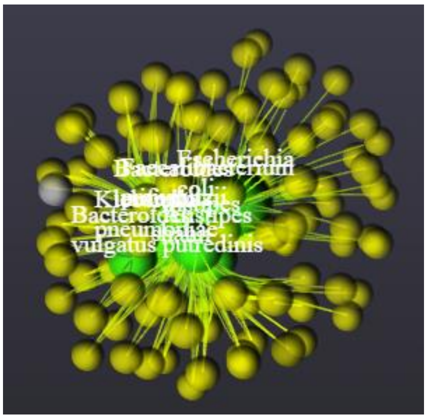


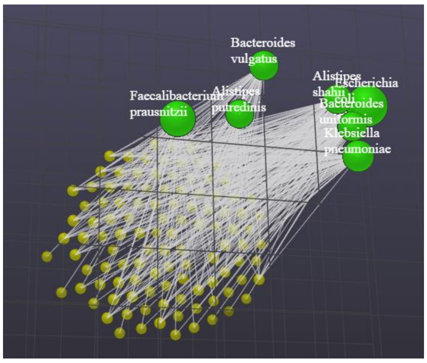
In this example first list of genes is mapped to the application, then list of miRNAs. Later two single list networks: PPI network and miRNA-gene networks, are created. Both are merged together to one composite network and the output is JSON graph file. Visualization is presented in Perspective 3D Layout with 2 layers.

- Integration of SNPs and Microbial Taxa



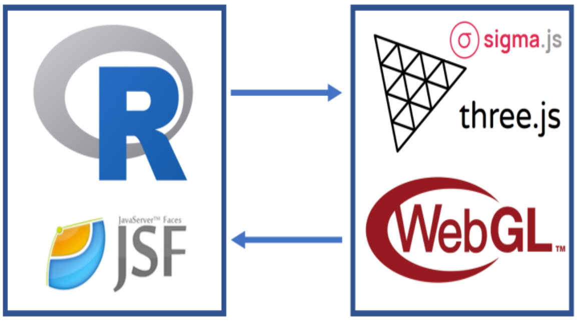
```
# Step 1. Initiate the dataSet object
dataSet<-Init.Data();

# Step 2. Input the microbial taxa data
dataSet<-PrepareInputList(dataSet,
                          "Faecalibacterium_prausnitzii
Bacteroides_uniformis
Eubacterium_rectale
Alistipes_putredinis
Subdoligranulum_unclassified
Escherichia_coli
Bacteroides_vulgatus
Clostridium_clostridioforme
Klebsiella_pneumoniae
Clostridium_hathewayi
Alistipes_shahii
Ruminococcus_obeum", "microbiome", "mic", "species");

# Step 3. Input the SNP data
dataSet<-PrepareInputList(dataSet,
                          "rs1428554
rs41290504
rs3197999
rs516246
rs2228058", "hsa", "snp", "rsid");

# Step 4. SNP data annotation and network construction
SetSnp2GeneOpts("vep", "", "dis", 1, 5);
dataSet<-QueryNet(dataSet, "snp", "vep", "snp");

# Step 5. Protein-protein extension
SetPpiZero(FALSE);
dataSet<-QueryNet(dataSet, "gene", "huri", "snp");

# Step 6. Protein-metabolite extension based on recon3D database
dataSet<-QueryNet(dataSet, "met", "recon3D", "snp");

# Step 7. Microbial data annotation and network construction
SetOrganism("microbiome");
SetMetPotentialOpts("0.8", "TRUE", "TRUE", "TRUE");
dataSet<-QueryNet(dataSet, "mic", "agora", "mic");

# Step 8. Merge networks together through shared nodes
# and decompose into interconnected subnetworks
dataSet<-CreateGraph(dataSet);
```





First Microbial Taxa is passed to the application alongside with SNP. Then both inputs' networks are made, protein-protein and protein-metabolite extensions are applied. Two networks that had been extended and improved through knowledgebase are merged together to build a composite network. Later JSON graph file would be generated. The upper picture shows Standard 3D Layout and the lower one is Perspective 3D Layout.

2.7. OmicsNet as a web server – implementation



OmicsNet is implemented as a web-based tool that integrates multiple technologies to provide its functionality. The implementation of OmicsNet involves the use of several programming languages and frameworks, including R, Java, WebGL, Three.js, and Sigma.js.

R is a statistical programming language widely used in bioinformatics and data analysis. In the case of OmicsNet, the R language is utilized to develop the underlying OmicsNetR package. This package contains the necessary functions and algorithms for network creation, trimming, and analysis. It provides the computational backbone for OmicsNet's functionality.

Java is a multipurpose programming language known for its platform independence and robustness. In OmicsNet, Java is used for backend development and server-side operations. It handles tasks such as data processing, network calculations, and communication with the R functions in OmicsNetR. Java helps ensure the efficiency and reliability of OmicsNet's operations.

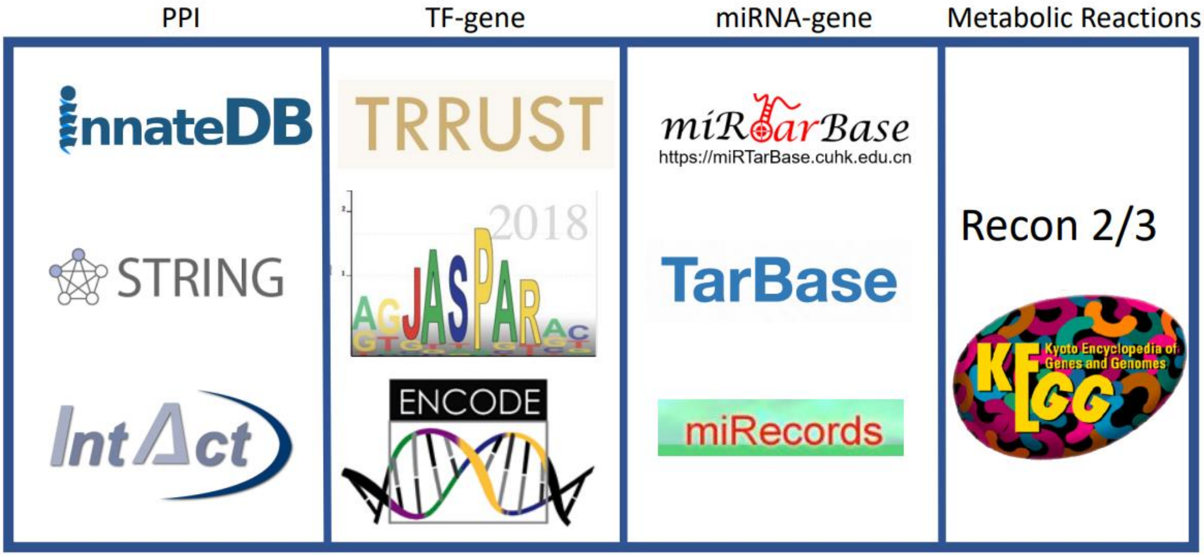
WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 2D and 3D graphics within web browsers. OmicsNet uses WebGL to enable the 3D force-directed layout visualization of the networks. It allows for interactive exploration and manipulation of the network in a visually appealing manner, enhancing the user experience.

Three.js is a JavaScript library that simplifies the creation and display of 3D graphics on the web. In OmicsNet, Three.js works in conjunction with WebGL to provide the framework for rendering the 3D network visualization. It handles the scene creation, camera movement, lighting, and rendering of the 3D objects in the network, bringing the visual representation to life.

Sigma.js is another JavaScript library specifically designed for network visualization. It offers a lightweight and efficient solution for displaying and interacting with graphs in web applications. OmicsNet utilizes Sigma.js to generate the 2D perspective and spherical layouts of the network. It enables smooth navigation, node highlighting, and other interactive features in the 2D and spherical visualizations.

The implementation of OmicsNet combines R for statistical analysis and network algorithms, Java for server-side operations, WebGL and Three.js for the 3D force-directed layout visualization, and Sigma.js for the 2D perspective and spherical layouts. Together, these technologies create a powerful and user-friendly platform for network-based multi-omics integration and visualization.

Also many databases are used in OmicsNet. they are used to store and retrieve curated biological knowledge and molecular interactions. They provide valuable resources for establishing connections between different biological entities, assigning functional annotations, and incorporating e.g. disease information. By using these databases, OmicsNet enhances the analysis and interpretation of multi-omics data within the context of existing biological knowledge. Some of them are shown below:



3. Summary

OmicsNet 2.0 is a network-based multi-omics analysis platform supporting both 2D and 3D network visual exploration. It enables support for three unordinary omics data inputs (SNPs, microbial taxa, and LC-MS peaks). It can reveal meaningful patterns, connections and functions. OmicsNet 2.0 addresses the need for easy-to-use web-based tools to support analysis of experimentally derived multi-omics data in their wider molecular context defined by our prior knowledge.

4. Bibliography

1. G. Zhou, Z.Pang, Y. Lu, J. Ewald, J. Xia; OmicsNet 2.0: a web-based platform for multi-omics integration and network visual analytics; Nucleic Acids Research, Volume 50, Issue W1, 5 July 2022 (https://academic.oup.com 2023), Pages W527–W533.

2. OmicsNet, Overview, omicsnet.ca, 2023.

3. OmicsNet 2.0 program, omicsnet.ca 2023.