A knowledge base for generating patient-specific pathways for individualized treatment decisions in clinical applications

Dissertation

for the award of the degree

”Doctor rerum naturalium” (Dr.rer.nat.)

of the Georg-August-Universität Göttingen

within the doctoral program Environmental Informatics (PEI)

of the Georg-August University School of Science (GAUSS)

submitted by

Florian Johann Auer

from Cham, Germany

Göttingen, 2022

Thesis Committee

Prof. Dr. Tim Beißbarth
Department of Medical Bioinformatics
University Medical Centre Göttingen

Prof. Dr. Frank Kramer
IT-Infrastructure for Translational Medical Research
University of Augsburg

Prof. Dr. Stephan Waack
Institute of Computer Science
Georg August University Göttingen

Members of the Examination Board

1st Reviewer: Prof. Dr. Tim Beißbarth

2nd Reviewer: Prof. Dr. Frank Kramer

Further members of the Examination Board:

Prof. Dr. Stephan Waack
Institute of Computer Science
Georg August University Göttingen

Prof. Dr. Ulrich Sax
Department of Medical Informatics
University Medical Centre Göttingen

Prof. Dr. Burkhard Morgenstern
Department of Bioinformatics
University of Göttingen

Prof. Dr. Carsten Damm
Institute of Computer Science
Georg August University Göttingen

Date of the oral examination:

18.05.2022

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Florian J. Auer

Place and Date

Acknowledgments

I would like to thank everyone who supported me during the last years during the preparation of this thesis and beyond.

Thank you to my supervisors, Prof. Dr. Frank Kramer and Prof. Dr. Tim Beißbarth, for your patience, guidance, and support. I have benefited greatly from your wealth of knowledge. I am extremely grateful that you took me on as a student and continued to have faith in me over the years.

I would also like to thank my Prof. Dr. Stephan Waack who was of great help, by giving me valuable remarks and comments in the field.

Special thanks to my former and current coworkers both in Göttingen and Augsburg for their help, especially Zaynab Hammoud, Dominik Müller, Johann Frei, Hryhorii Chereda, Júlia Perera-Bel and Michaela Bayerlová. In them, I not only found wonderful colleagues but also close friends who provided me with unconditional physical and emotional support, food, and accommodation in times of need.

Last but not least I would like to thank my parents for their continuous support during the years of my lengthy academic endeavors.

I also acknowledge the generous financial support from the Federal Ministry of Education and Research (BMBF).

Table of contents

List of figures

List of tables

List of abbreviations

AJAX	Asynchronous JavaScript
API	Application Programming Interface
ATAC-seq	Assay for Transposase-accessible Chromatin Sequencing
BioPAX	Biological Pathway Exchange
CDN	Contend Delivery Networks
ChIP–seq	Chromatin Immunoprecipitation Based Sequencing
CNN	Convolutional Neural Networks
CRAN	Comprehensive R Archive Network
CRUD	Create, Read, Update, and Delete
CSS	Cascading Style Sheets
CX	Cytoscape Exchange format
DNA	Deoxyribonucleic Acid
DOI	Digital Object Identifier
DOM	Document Object Model
GLM	Graph Modeling Language
Graph-CNN	Graph Convolutional Neural Networks
HATEOAS	Hypermedia As The Engine Of Application State
HPRD	Human Protein Reference Database
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
KEGG	Kyoto Encyclopedia of Genes and Genomes
KGML	KEGG Markup Language
mRNA	messenger RNA
NDEx	Network Data Exchange
OAS	OpenAPI Specification
PID	Pathway Interaction Database
PPI	Protein-Protein interaction
PSI-MI	Proteomics Standard Initiative-Molecular Interaction
REST	Representational State Transfer
RNA-seq	RNA sequencing
RxJS	Reactive Extensions for JavaScript
SIF	Simple Interaction Format
SOAP	Simple Object Access Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universal Unique Identifier
XGMML	eXtensible Graph Markup and Modeling Language
XML	Extensible Markup Language

"Declare the past,
diagnose the present,
foretell the future."

—Hippocrates

Florian J. Auer

1 Introduction

1.1 Motivation

The miracle of life not only keeps us researchers fascinated and mesmerized, it drives our passion to understand the underlying molecular complexity that forms the basis of our human existence. In the same way it haunts us when the miracle tends to vanish, and death becomes the imminent, inevitable consequence. The changes in the molecular patterns that cause this fatal shift seem to remain a mystery, not eager to unfold and reveal themselves to us. Almost.

It has been almost 70 years since the helical structure of DNA (1) was uncovered, and close to 50 years later the first human genome was sequenced (2). Since then, next-generation sequencing technology altered the landscape of research sustainably and has had remarkable impact on the understanding of fundamental coherences in genetics and biology (3,4). Single-cell RNA sequencing (scRNA-Seq) techniques provide transcriptomic insight on tissue composition, transcriptional dynamics, regulatory relationships between gene, and cancer evolution (5,6). With that, experiments that were previously technically not feasible or affordable now pave their way into clinical diagnostics (7–9).

Meanwhile, cancer has remained a burden to mankind even before its first depiction in ancient Egyptian manuscripts around 3600 years ago (10), and yet was crowned “the emperor of all maladies” in literature (11). Cancer exceeded cardiovascular diseases as leading cause of premature death by noncommunicable diseases in many high-income countries, while both still are responsible for two-thirds of all premature deaths from noncommunicable diseases worldwide (12).

In the last decades, biomedical fields have seen an on-going and extensive rise in data availability. In particular, high-throuput technologies have allowed researchers to explore the huge collections on genetic variability (13). The data ranges from not only all types of omics data like genomics, transcriptomics, proteomics, and metabolomics, but also on annotations, and links between data types. The large amount of data offers to opportunity for powerful advancements in biological and medical research. Public databases provide access to this omics resources, and thus enable more advanced analyses to uncover relevant actors and subnetworks which otherwise remained hidden. Biomedical knowledge, like biological profiles, presence or harmful markers, or missing interactions between proteins, provide medically relevant information to clinicians and scientists.

However, the data access in biological databases is often non-standardized and highly heterogeneous. Furthermore, these databases reveal extensive inter-connectivity between each other which cause knowledge representations to be complex and difficult to achieve. Still, for data exploration, comprehension, or integration into other processes, standardized and well-documented instruments for data exchange are crucial. This issue hinders not only the interoperability between research institutes, but also the integration capabilities into clinical applications.

This work outlines my contributions on solving this issue: Through the following presented publications, it was possible to push forward the scientific fields of data integration and visualization, and providing interoperability for biological data in medicine (Figure 1). Thereby the utilized biomedical data and their representation through biological networks acts as the base for the generation or patient-specific subnetworks. Furthermore, my contribution to fields of reproducibility

on network data integration and interoperability with clinical systems, including their current open issues are elaborated.

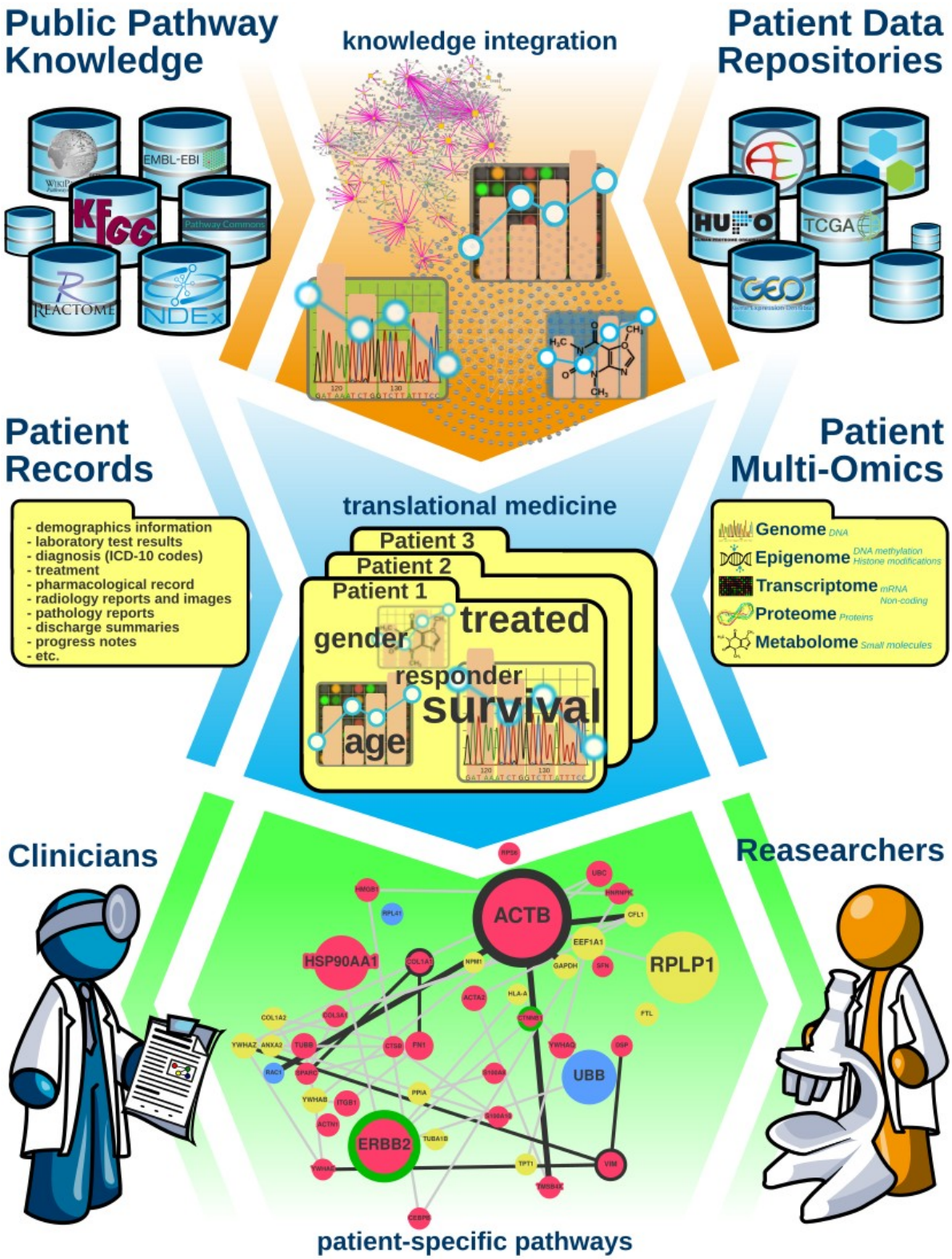


Figure 1: Integration of publicly available pathway knowledge and patient data to generate patient-specific pathways

1.2 Biological Data in Medicine

1.2.1 Gene Expression

Long before the invention of modern biotechnology, studies were done on observable characteristics of organism (also called phenotype). These early observation studies revealed that some kind of information is hidden in an organism which defines the outcome of the phenotype and is passed over generations, which was formulated as the principles of Mendelian inheritance. As later discovered, the origin of such characteristics is hidden in complements of the DNA (deoxyribonucleic acid), which are called genotypes. The process of a phenotype becoming observable is determined by specific environmental conditions influencing the genotype (14).

In the last 7 decades, various accomplishments in biology were achieved and set the foundation for modern genetics, ranging from the discovery of the helical structure of DNA to the industrialization of next-generation sequencing technology. These methods and theories altered the landscape of biological research and application. As a result, DNA, genes, mRNA and proteins were defined as key components in genetics. Whereas DNA, RNA and proteins were introduced as molecular units in a cell, genes established a basic concept of heredity through nucleotide encoding in the DNA which results in gene product synthesis represented by RNA or proteins. Thus, the term ‘gene expression’ can be defined as “production of an observable phenotype by a gene — usually by directing the synthesis of a protein”, according to Bruce Alberts et al. (15).

Gene expression is a complex mechanism that consists of multiple processes on different levels. A gene located on the DNA contains genetic instructions producing a gene product which can be responsible for development, functionality, growth or reproduction of a cell. These genetic instructions are transcribed into RNA by creating a complementary copy of the template strand. This process (also called transcription) results commonly in messenger RNA (mRNA) and represents a critical role of gene expression, protein coding, and regulation. Afterwards, mRNA is translated into a Protein. The protein translation is a complex process consisting of multiple steps as ribosome initiation and elongation, folding as well as translocation to the target localization in the cell. In consequence of translation, the final protein is assembled out of amino acids. The resulting protein reaction as enzyme, receptor, structure unit or transporter and its interaction with other proteins or molecules effect the phenotype.

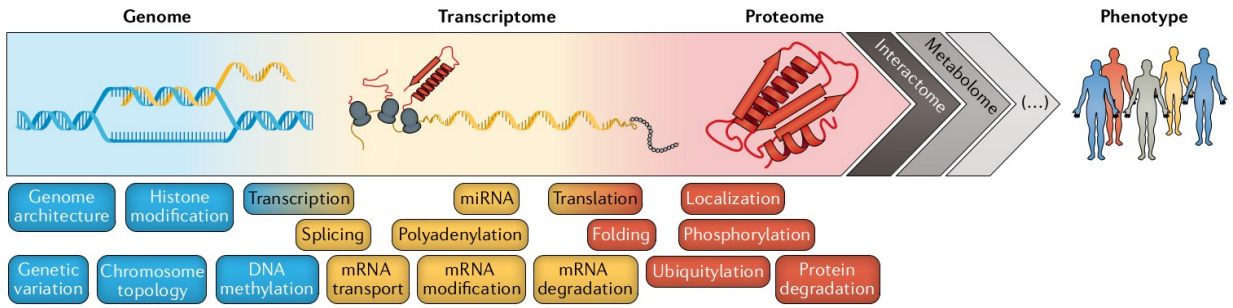


Figure 2: Overview of the gene expression pathway. Figure was obtained from Buccitelli et al. (14)

The gene expression can be measured on different levels providing distinct insights into the described multi-step process. The analysis of the transcriptome is the most widely performed procedure for gene expression quantification, which is why, gene expression is often treated as synonym for mRNA measurement in bioinformatical research (14). Consequently, the majority of

gene expression analysis studies are based exclusively on mRNA data. Even so, these studies presume for simplicity that a single gene results into a single gene product (14). However, this theory (called the central dogma of molecular biology) has been falsified decades ago, because a single gene can be the origin of various transcripts or proteins (16,17). Therefore, a comprehensive gene expression analysis requires not only mRNA quantification, but also genomic and protein measurement for obtaining insights on each level of the gene expression. Through next-generation sequencing techniques, the transcriptomic level is commonly profiled by RNA sequencing (RNA-seq) (18) and the genome by chromatin immunoprecipitation based sequencing (ChIP–seq) as well as assay for transposase-accessible chromatin sequencing (ATAC-seq)(19). Next to the quantification of gene expression intermediates, the localization can also be determined which allows additional information about spatial context (14).

Today, the improvements of omics technologies allow scientists gene expression analyses on various levels on an unprecedented scale. Despite that, the high-dimensional data provides not only unthinkable research and medical possibilities, but also creates new challenges in representation and integration through its exponentially increasing complexity.

1.2.2 Biological Networks

The ability for all living creatures to perform certain tasks depends on various interactions involved on different levels of the organism. The single cells of an organism constantly receive signals from the inside and outside and proteins and other molecules have to collaborate in processing the stimuli. The different entities must work together to perform properly, and slight alterations can effect the collapse of the whole system with diseases like cancer (20) or Alzheimer disease (21) as possible consequences.

The series of molecule involved in a process performing a certain task is referred to as biological pathway, with varying scope of the level of their involvement from metabolic interactions, gene regulation, or signal transduction just to name the most common. Signal transduction or cell signaling pathways are address the single steps involved in passing signals from the outside of cells to the inside to trigger particular action within cell. Receptors on the outside of the cell receive these signals and pass the message further using specialized proteins until a specific action is triggered or particular reactions are activated. This might be the activation of special genes to produce proteins which then again affect other actions. The mutual influence on gene activation and inhibition, and thus the change in the functions taking place through the encoded proteins is captured by gene regulatory pathways. Molecular pathways on the other hand portray the chemical reaction in which the proteins, and other molecules are involved.

However, the interaction across biological pathways are more complicated and need to broaden the focus to the entire biological networks containing the collection of all kinds of interactions proceeding within living systems (22). Curated in publicly available databases, biological networks comprise a valuable resource for capturing associations between any types of biological entities such as genes, transcripts, proteins, metabolites, ligands, diseases or drugs.

1.3 Integration of Biomedical Data

Biological networks are a powerful and flexible resource of biological knowledge and allow to express complex association valuable for subsequent analyses but only their integration with additional information allows their full potential to be exploited. Only the combination of biological network data and real-live patient information enables a deeper insight into underlying aberrations.

A large number of public databases make their biological knowledge available in domain-specific exchange formats. In subsequent analyses, these networks will be further enriched with heterogeneous data and therefore require a more flexible format for capturing their content. Albeit the many benefits of network formats, sharing, collaboration, and curation, including tracking of changes between different network versions, and detection different versions in general, still is a major problem in network biology.

Classically analysis workflows of transcriptomic data from high-throughput experiments to determine the gene expression, as well as subsequent analyses are often done in R. Integration of the results with biological networks calls for the establishment of a robust data model for storing and distribution the integrated results. Public online commons like the NDEx platform can thereby play the role of a knowledge base for the simple management of the results. However, this requires an equally strong interface to the database as well as seamless integration and support of the network models on both sides. For use in a clinical context, additional established standards for data exchange must also be taken into account (23).

At the same time, the visualization of this integrated data is absolutely necessary for communication and understanding of the findings, which makes it all the more important that visualization and integration go hand in hand. Instead of seeing the visualization only as an additional application to the results, it must itself be seen as part of the analysis and ideally become part of the integrated network data.

Goal of this work is to build a basis for handling the knowledge from various resources, integrate the the data, and subsequently generate patient-specific pathways. This knowledge base then can be used in clinical applications to provide patient-specific recommendation to clinicians supporting treatment decisions for better prognosis.

2 Materials and Methods

2.1 Web Standards and Development

Since its invention in 1989 the world wide web underwent great advancements in its technology and changed the communication of data and information, and ultimately mankind itself. In its beginning it was developed for the sharing and management of documentation (24) but rapidly evolved to distribute information of any kind. The underlying technology advanced with the new demands from static propagation of information as documents in the HyperText Markup Language (HTML) towards to enable interaction from the client side introducing forms and servers that could handle the requests. Cascading Style Sheets (CSS) separate the content from the visualization by applying generalized, hierarchical rules to the elements of the documents to adjust colors, fonts, and layouts.

An increasing amount of the interactive functionality began to move from server side processing to be handled within the web-browser. The retrieval of requested information as whole HTML documents changed the same way to supplying only updated fragments up to solely the data necessary to construct its representation. Also the perspective on the data changed in the way that it is seen as a resource that can be interacted with, modified, and linked with other resources, even on distributed servers.

Web development is a broad and constantly emerging field ranging from client-server communication, front- and back-end scripting, web design, and application and database development. This work depends on a variety of established web-standards and technologies, methods and frameworks for development. An overview of the concepts that are referenced within this work and essential for it understanding is provided in the following.

2.1.1 JavaScript

The programming language JavaScript is one of the core components of modern days web-technologies (25). While HTML provides the content and CSS define the visualization, JavaScript adds the functionality to the website. Although the name suggest a relation with the Java programming language both only share marginal similarities, mostly in syntax and standard libraries. One of the main differences is the weak and dynamic typing of JavaScript: The initial type of a variable can be reassigned with another type which leads to implicit typecasts. Therefore, type checks are required to be implemented to ensure the expected behavior.

JavaScript is a multi-paradigm language, meaning it integrates features from different programming paradigms: Mainly it follows an object-oriented principle where objects contain the data and functions form modification and interaction. Thereby, In contrast to the in Java used class and inheritance model, JavaScript follows the instance based approach in which objects serve as prototypes that are cloned and extended. Functions assigned to the prototype then are accessible to the instances, and in general there is no differentiation between static functions and object methods. Additionally, the functions can be passed to other functions illustrating the functional character of the programming language.

These feature are often used within external packages. The rather small set of functionality included by default library is mainly extended by third-party libraries which not necessarily need to be hosted at the same server. Specialized Contend Delivery Networks (CDN) are used to distribute JavaScript libraries and use them on several websites. Cashing these libraries in the browser supersedes the reload of these and thus reduces network traffic and speeds up website loading.

JavaScript also incorporates event driven features using events and callback functions to determine the script execution. This enables the parallel processing of user interaction and program input and output, for example mouse clicks can be handled while simultaneously data is retrieved from an online resource. The concurrency can be implemented either as promises that are resolved by callback functions or using an Async/await pattern.

The most important feature of JavaScript is the interaction with the Document Object Model (DOM), the tree representation of HTML document. Manipulation of the DOM not only allows the validation of contained form data, or animation of website components but also replacement of specific parts. Asynchronous calls can be used to retrieve HTML fragment without reloading of the whole website to update its content (Asynchronous JavaScript and XML, AJAX).

Web-development is generally divided in server- and client-side implementation of project logic, historically realized in different programming languages. This leads to the problems that the same parts have to be adapted several times, which raises costs, requires the programmers to be familiar with multiple programming languages, and is potentially error prone. Therefore, different attempts have been made using the same programming language for both back-end and front-end. In the Java world, as major language for the back-end, the framework Vaadin (26) enables the creation of HTML based front-ends in the same language. In R it is possible to build interactive web applications and HTML widgets with the Shiny (27) package. The JavaScript runtime node.js (28) goes the opposite way and brings the language to the web-server. The open-source, cross-platform software allows exchangeability of code and data models between both stacks and founds as base many current frameworks for web-development build upon.

2.1.2 JavaScript Object Notation

JavaScript Object Notation (JSON) is a file format derived from JavaScript data types and is an established standard for data exchange. Although its origin lies in its usage within scripts on web-pages it is widely adapted by many software and programming languages. It makes use of the standard JavaScript data types like strings, numbers and booleans, and allows with the help of lists and maps (called objects) the construction of complex nested data structures and serializable objects. An example of a real-life JSON object for a FHIR Patient resource used in Chapter 3.7 is shown in Code 1.

JSON provides a syntactic framework for the definition of data structures. Definition and validation of specific schemes for custom data structures can be implemented using vocabularies like JSON Schema (29). With that the expected data structure of applications can be specified and checked to ensure data quality. However, the most validation schemes and validators are limited to structural elements and cannot handle references across or within objects. Also for serialized objects the schemes require a specific order of the objects.

Code 1: Exemplary JSON object retrieved from a FHIR server for a Patient resource

```
{
"fullUrl": "http://localhost:8080/fhir/Patient/1",
"resource": {
  "resourceType": "Patient",
  "id": "1",
  "meta": {
      "versionId": "1",
      "lastUpdated": "2022-02-10T21:13:49.350+00:00",
      "source": "#waYE0EueRQyUFd0r"
  },
  "text": {
      "status": "generated",
      "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\"><div
class=\"hapiHeaderText\">Yong <b>HUEL </b></div></div>"
  },
  "identifier": [ {
      "system": "study_internal_id",
      "value": "Patient 1"
  } ],
  "name": [ {
      "family": "Huel",
      "given": [ "Yong" ]
  } ],
  "telecom": [ {
      "system": "phone",
      "value": "999-43-6884"
  } ],
  "gender": "female",
  "birthDate": "2007-10-28",
  "address": [ {
      "line": [ "1077 Zemlak Annex" ],
      "city": "Chelmsford",
      "district": "Middlesex County",
      "state": "Massachusetts",
      "postalCode": "1851"
  } ],
  "maritalStatus": {
      "coding": [ {
        "system": "http://terminology.hl7.org/CodeSystem/v3-
MaritalStatus",
        "code": "UNK"
      } ],
      "text": "unknown"
  }
}
}
```

With Extensible Markup Language (XML) there exists an other established and widely used data format besides JSON for data exchange. XML is more powerful in terms of expressiveness and mostly used in applications with more complex requirements than data interchange. However, both contributes to greater efforts in data handling and validation, thus XML is commonly used for the transmission of large scale data collections while JSON is preferred within web-applications.

2.1.3 TypeScript

TypeScript originated from the shortcomings of JavaScript is a syntactical superset of it, meaning it builds upon it and adds features to the language. The TypeScript code is usually not run directly but transpiled to JavaScript which then is deployed to server or web applications.

Code 2: TypeScript adaptation of the FHIR Patient resource

```
import {Identifier} from './identifier';

export class Patient {
  private id: number;
  private identifier: Identifier;
  private name: PatientName;
  private telecom: PatientTelecom;
  private gender: string;
  private birthDate: string;
  private address: PatientAddress;
  private maritalStatus: MaritalStatus | undefined;

  constructor(data: any){
    this.id = Number(data.id);
    this.identifier = new Identifier(data.identifier[0]);
    this.name = new PatientName(data.name[0]);
    this.gender = data.gender;
    this.birthDate = data.birthDate;
    this.telecom = new PatientTelecom(data.telecom[0]);
    this.address = new PatientAddress(data.address[0]);
    this.maritalStatus = new MaritalStatus(data.maritalStatus);
  }

  getName(): string {
    return this.name.given.join(' ') + ' ' + this.name.family;
  }
}
```

The dynamic typing and flexible inheritance mechanism can cause huge problems in development, especially in larger projects. Type safety cannot be ensured in JavaScript without rigorous type checks, which even worsens considering the data loaded via JSON. TypeScript implements strict typing of variables and enforces it on compile time. It introduces a class system similar to Java which can be used for the formal definition of data loaded in JSON format. To prove the validity of the above FHIR Patient data (Code 1) and be able to use it safely within application a corresponding class can be implemented as shown in Code 2. Objects of this class are created from the loaded data and type checks are performed automatically. Furthermore, default values can be set and the data modified in the constructor to fit the data best to its intended use case.

TypeScript was widely adapted by projects and frameworks for web development since it heavily reduces the effort for ensuring data, type and code consistency. Many third-party libraries originally developed in JavaScript provide TypeScript headers for their seamless integration. Even large frameworks like Angular were rewritten in TypeScript because of its advantages in expressiveness, maintainability, and usefulness (30).

2.1.4 Representational State Transfer

Web resources are the documents and files available on the World Wide Web and rely certain design and architectural style guidelines and constrains for practical usage. Network-based applications such as in client-server architectures build upon simple interfaces and a layer of abstraction decoupling usable entities from the underling implementation. A solution brings the Representational State Transfer (REST) (31), a widely accepted software architecture style with focus on scalability, simplicity, uniform interfaces, independent components, enforcing security, encapsulating legacy systems, and the creation of layered architectures (32). In contrast to other common approaches like the Simple Object Access Protocol (SOAP) REST is considered easier to use and implement contributing to it popularity.

APIs following the REST guideline have to conform to some criteria:

- Client-server design pattern to separate data storage from user interface concerns (separation of concerns)
- The server implementation and structure is hidden from the client (layered architecture)
- The requests of resources have to be managed by the Hypertext Transfer Protocol (HTTP)
- Each request is separate and unconnected, i. e. no client information is stored between requests (stateless client-server communication)
- Requested data is cacheable to reduce or even eliminate unnecessary client-server interactions
- Uniformity of the interface with standardized transmission, which includes:
  - Identification of the resources in requests
  - Manipulation of the resources through their representation (including modification and deletion)
  - Each message contains enough information about how to be processed (self-descriptiveness)
  - Hypermedia as the engine of application state (HATEOAS), meaning after access of a resource the client is able to use provided hyperlinks to dynamically discover all other available and required resources
- Optional: Temporal extension or customization of the client functionality (Code on demand) for example with JavaScript

Within network-based architectures the Application Programming Interface (API) provides a set of protocols and definitions for building and accessing information resources. APIs obeying the REST principles are commonly referred to as RESTful (33). It is noteworthy that APIs described as RESTful not always fulfill all constrains, especially the uniformity of the interface may lack in at least one constrain.

Web-services designed with HTTP-based RESTful APIs allow the interaction with the provided resources, or more precisely the representation of those. These resources must be reachable on web-servers by a Uniform Resource Identifier (URI), thereby, the Uniform Resource Locator (URL) contains the resource name and a unique identifier or query parameters (Figure 3). Basic persistence operations (as for databases) can be performed on the resources like create, read, update, and delete (CRUD). For each of those operations exists at least one corresponding HTTP method (Table 1) for programmatic interaction with the API. However, not all of these operations must be supported by all resources, only reading of the resource is essential.

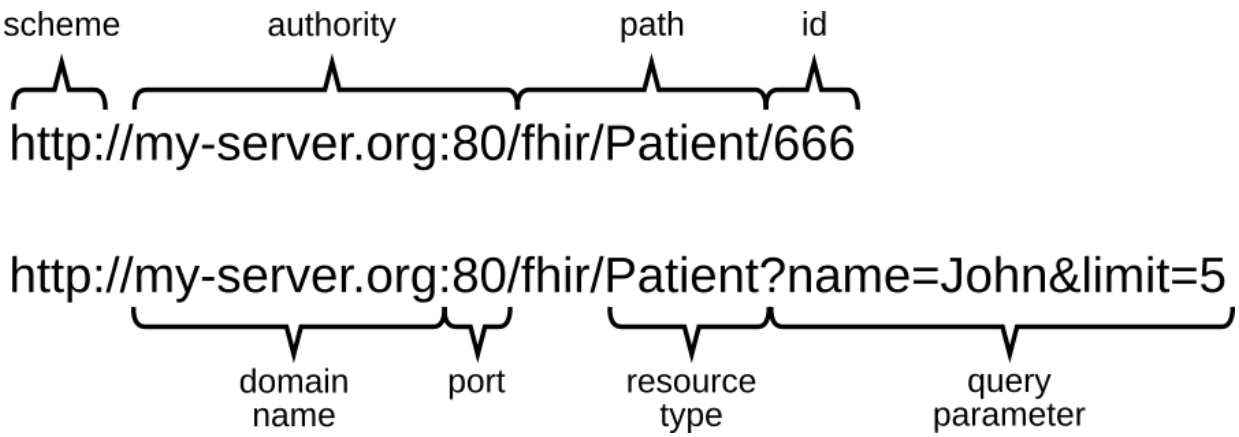


Figure 3: Composition of URLs illustrated in the example of a FHIR endpoint for a patient resource with ID or query parameters.

Working with resources on RESTful APIs generally works as follows: Firstly a GET request is performed to the resource endpoint, mostly including parameters for querying or pagination. The returned list contains either a list of resources or simply links to the single entities. In the latter case the included links, or only the IDs if necessary can be used to access specific resources. Those can either modified with PUT or PATCH, used to create new instances with POST, or simply deleted with the correspondingly named operation.

Table 1: Overview of CRUD and HTTP operations for RESTful APIs

CRUD	***HTTP***	***Description***
Read	GET	Retrieval of a single resource, or list of resources for queries
Create	POST	Creation of a new resource object
Update	PUT	Replace or create a resource with the defined state
	PATCH	Partially update a resource only with the defined properties
Delete	DELETE	Delete the targeted resource

RESTful web-interfaces have made great impact to the shape and interaction of current applications. Much of its success can be attributed to its simplicity and that it founds on established technology standards. Since the used methods are agnostic to the project specific programming languages it has fostered data-driven development in many fields and was adopted by major development frameworks.

2.1.5 Angular

The world wide web has a growing impact on our every days life which was also influence through the wide availability of mobile devices. This also lead to a shift in the development of applications from native software towards to be run within web browsers. With that also the complexity of these applications increased and larger projects face the same problems as classical software development. Specialized frameworks for the development of browser-based web-applications filled this growing gap, with Angular (34) as one of the most prominent representatives.

Angular is a component-based framework written in TypeScript for building modular web-applications. It provides built-in libraries for routing, client-server communication, and form management, as well as tools to develop, test and build scalable applications. It was developed by Google as a free and open-source software framework to build complex single-page web-applications and progressive web apps.

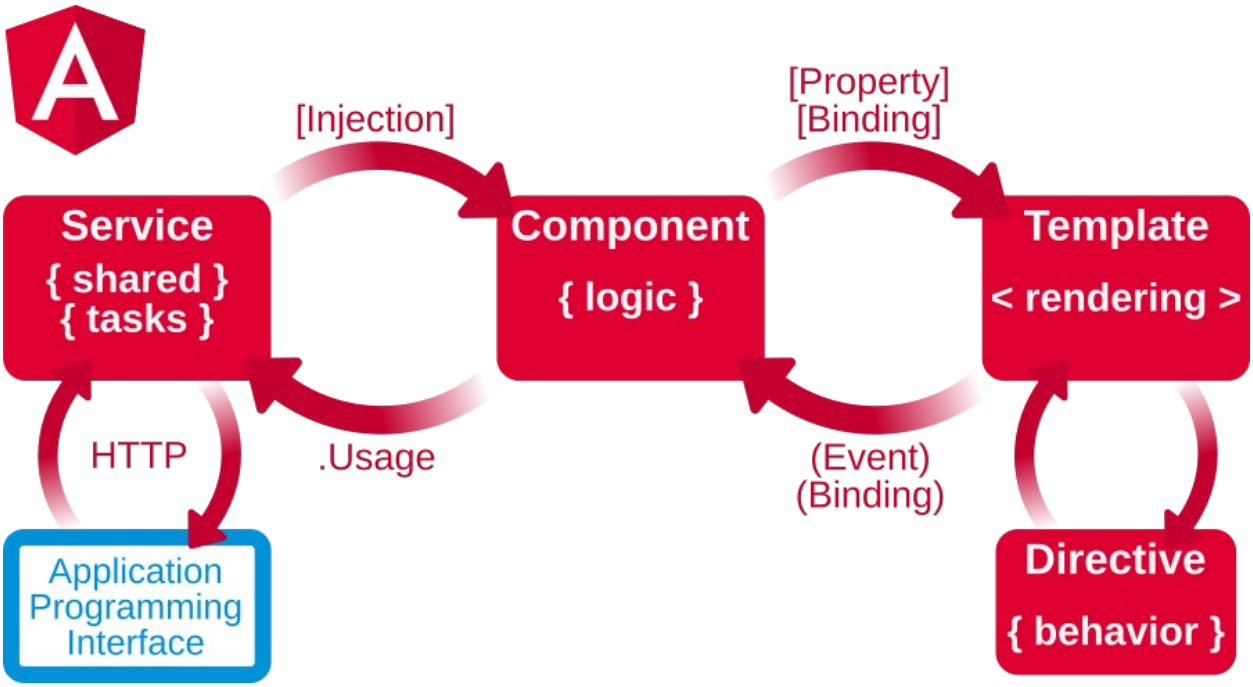


Figure 4: Architecture of Angular applications. The main building blocks are Components, Templates, Services, and Directives, all separated into individual modules.

A core concepts of the Angular architecture (Figure 4) is its modularity. The different modules are associated with specific views, meaning a certain part within the application that has a specific topic, functionality, and visual representation (rendering). The modules are typically arranged hierarchical, therefore encapsulating different parts of the application. On the one hand this promotes security because objects can only be shared with the children of the modules. On the other hand it enables reuse of the modules as templates in the classical sens, meaning as composition of Angular *components* and *templates*. Angular consist of the following connected entities with their individual purpose:

- *Components:* Contain the application data and logic
- *Templates:* HTML templates combined with Angular markup to render the data of the component (view)
- *Directives:* Functions that change the structure and attributes dynamically while rendering the templates
- *Services:* Data and logic that is shared across components and is therefore not associated with a specific view

Components and *templates* are linked to each other by a two-way binding, meaning that on the one side changes in the data, e. g. loaded data from a web-resource affects the rendering of the data.

On the other side, changes in the DOM, such as user input and actions impact the program data in the *components*.

Services help in structuring the application to separate common jobs from the specialized tasks of the *components*. Dependency injection provide the services when need, for example performing server requests, form validation, or logging. Moreover, the routing module provides a service to define the navigation path based on the URL to the different states of the application. It emulates the browser navigation and allows to assign sub-paths to a specific component hierarchy, including IDs similar to resource IDs in RESTful APIs.

Data retrieval from web-resources requires time to process, thus asynchronous handling of requests is necessary to avoid blocking and freezing of the application. Therefore, Angular implements the observer pattern as *Observables* with a *subject* that maintains a list of subscribed *observers* and notifies them on state changes. *Observables* are used extensively withing Angular and handle different types of data like literals, messages, or events. The application logic can focus on its actual tasks and only need to subscribe to retrieve the data and unsubscribe afterwards.

With the Reactive Extensions for JavaScript (RxJS) library (35) Angular makes use of the asynchronous programming paradigm and enables reactive programming using *Observables*. A practical example for its usage is the retrieval, handling, and processing of data from several web-resources: Asynchronous loaded data is steamed through different steps to be filtered, mapped, composed, or iterated over, and finally returned as *Observable*. Subscribe consumers are simply called after data is loaded and all processing steps are performed and can continue to work with the results.

Angular is a powerful framework which gained its popularity by providing a huge tool set facilitating the development of complex web-applications. Based on established methods an paradigms for web- and asynchronous development it simplifies structuring and implementing of large projects. Furthermore, it is extensible and customizable through a vast amount of available third-party libraries.

2.2 Data Structures and Formats in Network Biology

2.2.1 Adjacency Matrix

An adjacency matrix is a square matrix with both dimensions representing the nodes. The values of the matrix can simply indicating a connection between two nodes, or also weight the interaction. Self loops, i. e. an edge from one node to itself, can be possible, but multiple edges between two nodes (multi-edges) can be expressed to some extent. Since the nodes are present in both dimension it is possible to indicate and weight the edges in two directions.

2.2.2 Edge List

A graph can be described by simply enumerating the edges between the vertices. An edge list therefore is a list of pairs of nodes defining the start and the end of the edge. Self loops and multi-edges can be formulated. If the edge is weighted an additional number for the weight is given, but whether the edge is directed is dependent on the reader.

2.2.3 Simple Interaction Format

The simple interaction format (SIF) could be considered an extension to the edge list: It is composed of the source node, the interaction type, and a single or list of target edges. A simple example look as follows:

```
node1 relation node2 node3 node4
```

Self loops and multi-edges are possible in this format.

2.2.4 Graph Modeling Language

The Graph Modeling Language (GML) is a text based format for describing graphs with a simple syntax. The graph, nodes and edges are defined as objects with specific attributes (Code 3). There is an XML adaptation of GML called eXtensible Graph Markup and Modeling Language (XGMML), which simply expresses the objects as XML elements with its attributes.

Code 3: Exemplary definition of a network using the graph modeling language

```
graph  [
      node  [
        id  0
        label “node A”
    ]
    node [
        id  1
        label “node B”
    ]
    edge [
        source  1
        target  0
        label “connects”
    ]
]
```

2.2.5 Cytoscape Exchange Format

Previously presented network formats are primarily designed for the storage of network information. Therefore, sharing and usage of networks in this format brings the same issues as other file-based data management approaches. With the rise of web based technologies formats for network encoding underwent a re-thinking, away from storage and towards transmission of the data. Also the similarity of GML to the object focused style of JSON might have inspired the further development through the adaptation of established web technologies.

The result of these developmental advancements is the Cytoscape Exchange format (CX), a JSON-based format specifically tailored to the transmission of networks. It was developed by the Cytoscape consortium especially for this purpose to exchange networks from within the visualization software Cytoscape (see 2.4.1) with a web-based storage platform (see 2.3.2). It follows an aspect-oriented design, meaning the network is seen as individual interlinked components. A separation of the different concerns as single aspects provides a flexible framework for the incorporation of networks from any domain. Moreover, the segmentation of network data allows to request and dynamically load only specific excerpts of the the network, based on the intended usage. The format

was developed with thought to streamline the network data, meaning the processing of networks can be consecutively chained so that the output of one program can form the input of the next. Thereby, the different programs can focus on only specific aspects and pass through others leaving them to remain opaque to the processor.

Table 2: Official aspects defined for the Cytoscape Exchange (CX) format. *Optional aspects are marked in italic font, aspects with IDs are marked with an asterix.*

Meta information	***Core***	***Cytoscape***
metaData	nodes*	*cySubNetworks**
numberVerification	*edges**	*cyGroups**
status	*nodeAttributes*	*cyHiddenAttributes*
	edgeAttributes	*cyNetworkRelations*
	networkAttributes	*cyTableColumn*
	cartesianLayout	*cyVisualAttributes*

The CX format provides a collection of predefined aspects (Table 2) that can be categorized into three groups:

- core aspects define the structure, attributes and layout of the network
- aspects inherited from Cytoscape specifying the network visualization (e. g. color and size of nodes and edges depending on their attributes)
- meta information necessary for the transmission of the network data

The aspects consist of different mandatory and optional properties, even most of the aspects are optional for building valid networks. For example, a node consists of an unique ID, and optional name and reference (a simple gene name or link to external database, called *represents*). The IDs are simple integers, only unique within one aspect that can be used to reference the aspect's elements within other aspects and therefore link the different component of the network (Figure 5). This principle applies to all aspects of the network: the elements of the *edges* aspect connect two nodes and define the interaction between both, the *cartesianLayout* aspect positions the single

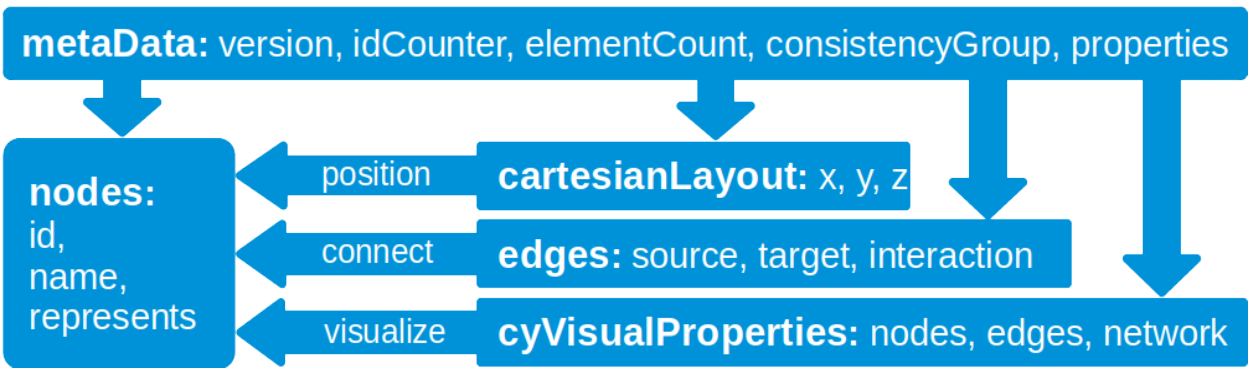


Figure 5: Relation between aspects in the CX format. *The nodes aspect is the main entity which is referenced by other aspects to assign a position (cartesianLayout) to the nodes, connect them (edges), or apply visual styles (cyVisualProperties). The metaData aspect aggregates information about the contained aspects in the CX network.*

nodes in two- or three-dimensional space, and the *cyVisualProperties* aspect applies a mapping of the contained data to create a visual representation. The latter is a special feature of the CX network format: the visualization not only applied to the network but even part of it and linked with its other aspects.

To construct valid networks only at least one node has to be defined and the aspects involved for transmission (Code 4), and those in a specific order. Thereby, *numberVerification* has to be the first aspect present, defining the maximal integer manageable by the client system, thus ensuring following data can be processed. Next aspect must be the *metaData* aspect, describing the in this network contained aspects, including their number of elements and, if applicable, the highest used id. Updating different aspects individually should be reflected by increasing the version number correspondingly. Eventually occurring deletions or changes of elements with own IDs are therefore marked and can responded to with checking the dependent (i. e. referencing) aspects, update them as the circumstances require, and adjust their version accordingly. In the case of streamlining this steps then can be delegated to the successors. The last element must be the *status* aspect which simply indicates the success of the transmission or processing of the network, or encloses an according error message.

Code 4: Example for a minimal valid network in CX format: The network only consists of one nodes and necessary meta data for transmission. The names of the aspects are highlighted.

```
[
      {"numberVerification": [{"longNumber": 281474976710655}]},
      {"metaData": [{
            "name": "nodes",
            "elementCount": 1,
            "idCounter": 1,
            "version": "1.0"
            }]},
      {"nodes": [{"@id": 0}]},
      {"status": [{"error": "", "success": true}]}
]
```

Additional data of the networks is integrated as attributes of either nodes, edges, or the network itself. The data follows a key-value pair structure, but with both provided as values for the predefined keys *n* (name) for the key and *v* (value) for the corresponding value respectively. Thereby, there is no distinction between the type of the data: By default the values are treated as strings, except it is stated otherwise explicitly as either boolean, integer, double, or a list of those. For example, Code 5 show the attributes for the network used to describe the RCX data structure (see Chapter 3.1). As by convention the NDEx platform (see Chapter 2.3.2) uses the attributes *name*, *description*, *author*, *rightsHolder*, and *version* as specially treated meta information for display on the web-application.

Additionally to the officially specified aspects it is possible to define own ones. This enable to include additional data if needed in a suitable format. NDEx and other applications will treat those aspects the same way opaquely as applications in the streamline process. With this the CX data structure provides a great option of extensibility. However, to be able to use this efficiently it is recommended to provide documentation of the extension or even an implementation for potential adoption in specialized clients.

Code 5: NodeAttributes taken as an excerpt from the network on NDEx with UUID ebdda4da-2ca5-11ec-b3be-0ac135e8bacf and visualized in Figure 13.


```
{"networkAttributes": [
      {"n": "name", "v": "RCX Data Structure"},
      {"n": "description", "v":"Figure 13: An RCX object (green) is
composed of several aspects (red), which themselves consist different
properties (blue). Some properties contain sub-properties (light red)
that also hold properties. Properties reference ID properties of other
aspects."},
      {"n": "version", "v": "1.0"},
      {"n": "author", "v": "Florian J. Auer"},
      {"n": "rightsHolder", "v": "Florian J. Auer"},
      {
            "n": "references",
            "v": [
                  "https://bioconductor.org/packages/RCX",
                  "https://github.com/frankkramer-lab/RCX",
                  "https://home.ndexbio.org/data-model/"
            ],
            "d": "list_of_string"
      },{
            "n": "referencesProvideSourceCode",
            "v": [
                  "true",
                  "true",
                  "false"
            ],
            "d": "list_of_boolean"
      }
]}
```


2.3 Databases for Biological Networks

2.3.1 Human Protein Reference Database

The Human Protein Reference Database (HPRD) provides literature derived protein annotations, including protein-protein interactions (PPI), posttranslational modifications, enzyme/substrate relationships, disease associations, tissue expression, and subcellular localization (36–38).

Contrary to other reference databases, HPRD focuses only on the human proteome, and therefore excludes entries from different species and references to those even if interaction with human proteins were verified. The database is available for download in tab delimited, XML, and Proteomics Standard Initiative-Molecular Interaction (PSI-MI) (39) format, with options for downloading solely the binary PPI data or additional protein features included such as post-translational modifications, tissue expression, subcellular localization.

The latest release of the database dates back to 2010, and therefore omits over ten years of scientific insights. Nevertheless, this manually curated information about 30,000 human proteins makes HPRD a comprehensive resource for studying the human proteome.

2.3.2 The Network Data Exchange

The Network Data Exchange (NDEx) platform (40–42) is an online commons for biological networks of any kind, where users can store and manage their networks, and share it with others. In simple terms, it can be described as Dropbox or Google Drive for networks, but with a specialized focus on the community concept in particular. Researchers, scientists and organizations are encouraged to contribute their network data to build up a shared collection of biological knowledge. The intention is not compete with existing pathway and interaction databases, but rather to establish an access point, platform and distribution channel for users and groups. NDEx also replaces the Pathway Interaction Database (PID) (43), a curated and peer-reviewed collection of human signaling and regulatory pathways and cellular processes. The original data was included into NDEx and is still available for analyses.

The web portal of the NDEx platform serves as contact point for the community and shows recent news and highlight in network biology. For first time visitors a selection of featured and exemplary networks are listed to facilitate to getting started, as well as an comprehensive guide to use the platform. There are several options to query the database: A simple search returns all networks for a given term that matches in the name or description. More sophisticated can be formulated using keywords, wildcard, logical conjunctions and ranges to limit the search spaced. For example the expression

```
name:bre* AND owner:ccmi AND nodeCount:[500 TO 600]
```

allows to search for networks, that start with “bre”, are owned by the user “ccmi” and contain between 600 and 600 nodes.

The networks can also be searched for specific genes or proteins within the network by providing a list. A special feature is to automatically expand the list of gene and protein names to its synonyms, which leads to more relevant results and removes the need to do this manually. However, this feature only works for human genes and proteins, and only for networks that are indexed.

The results of the search are returned in three categories: networks, users and groups. For each category results provide more details about the single items in several columns that can be used to filter the results further. For networks, not only its name is provided, but also information about its owner, the number of nodes and edges it contains, its last date and time of modification, and if available the tissue it refers to, the disease it is involved, and a reference to a corresponding publication. Furthermore the description of the network can be displayed by clicking on the NDEx icon before its name, and it also can be downloaded directly.

The above advanced query returns the breast cancer protein-protein interaction network by Minkyu Kim (44) which can be opened in the web-application by clicking on its name. All networks uploaded to NDEx are assigned with an internal Universally Unique Identifier (UUID) by that the network is directly reachable. The web-application provides further information about the network and its properties along an interactive visualization of the network as shown in Figure 6. It is noteworthy that all displayed information, including the visual properties necessary for its visualization are included within the downloadable network. With one click, the network can also be opened directly in a running Cytoscape instance and modified there (see 2.4.1).

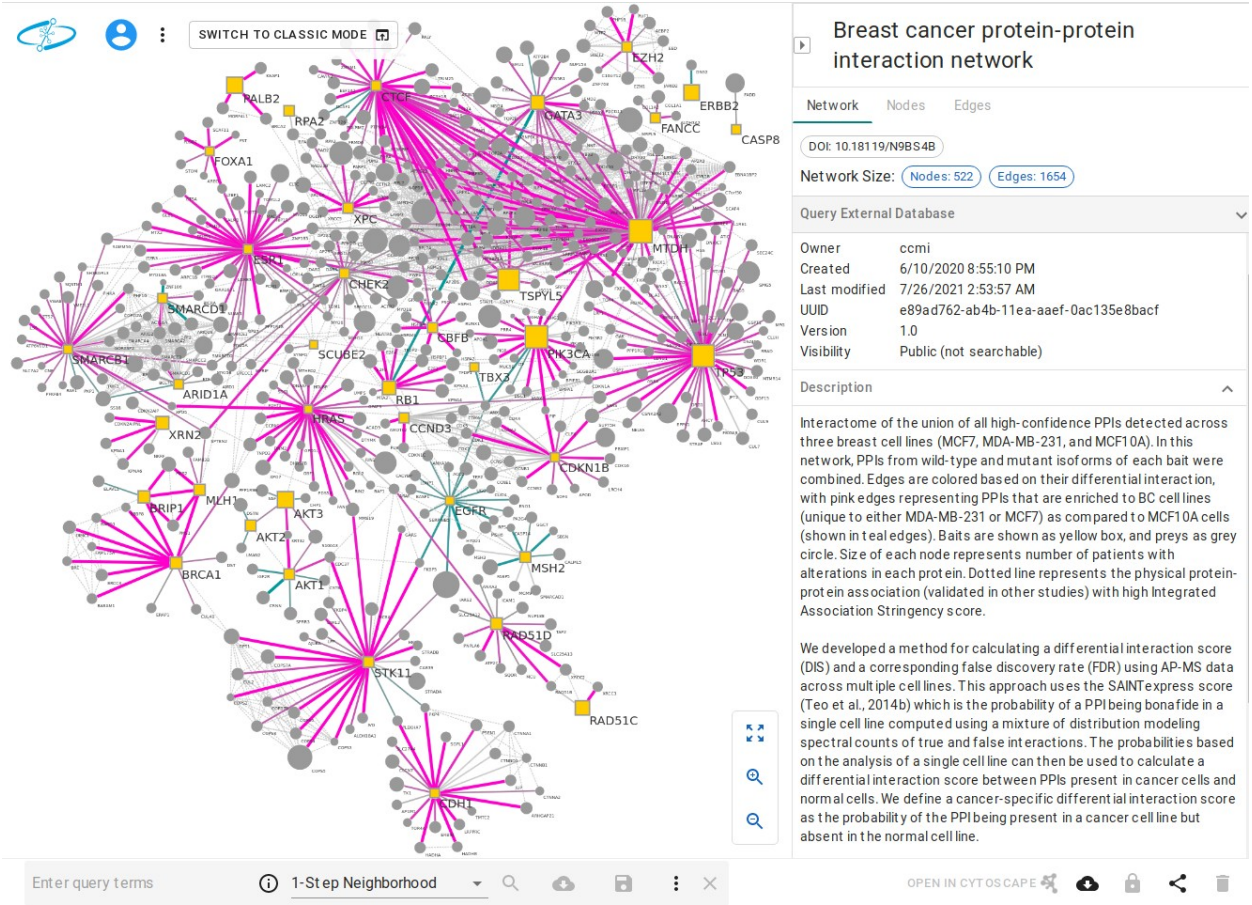


Figure 6: Breast cancer protein-protein interaction network visualized on NDEx. The network is reachable at https://doi.org/10.18119/N9BS4B

The interactive visualization allows to explore and rearrange the network to acquire more insight on the contained data. It can be explored by selecting several nodes and edges, for which the attributes will be displayed in the corresponding tab. The network also can be searched in the bottom with several options to define the sub-graph that should be returned. This can be simply be a graph spanned by the matching elements, their first or second step neighborhood or adjacency, or only the interconnected nodes. A visualization of the sub-graph then will be displayed bellow the actual network visualization.

NDEx is a community driven platform, which means that single users and groups can contribute with their own networks. After creating an account, users are gifted with an 10GB free storage for their network data. Uploading a network does not necessary mean, that it is intended to be available to the public, therefore different access and visibility options can be set. A typical workflow is illustrated in Figure 7:

A newly uploaded network is set to private by the user and can not be accessed by anyone else. To collaborate further on the network, it can be shared with other users or, even groups, whereat it can be specified if the network is only visible to them or can be modified. After the work is finalized, and the networks poses as supplemental information to a manuscript, a restricted link can be generated and provided in the submission, allowing only the reviewers to access the network. When the manuscript has been accepted, the network then is set to be public. For long term availability and

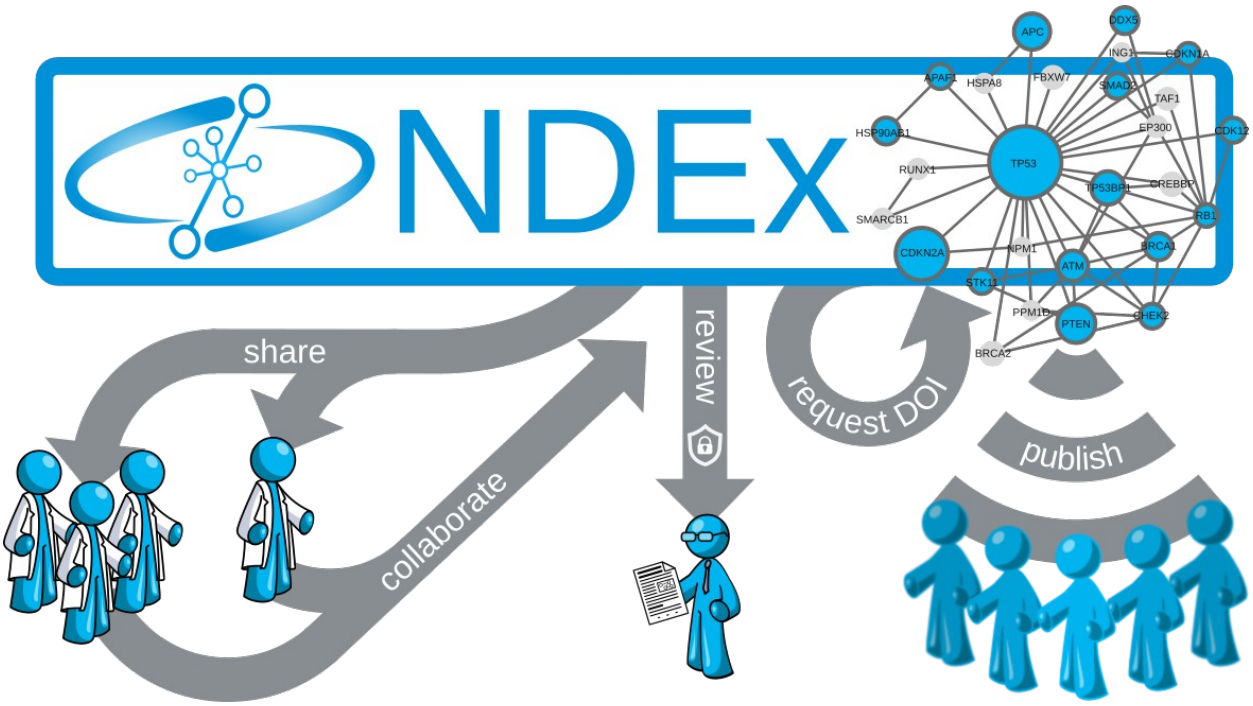


Figure 7: Workflow for sharing networks on the NDEx platform. Uploaded networks can be privately shared with certain people or groups to collaborate on. A protected link to the network can be provided to reviewers along a manuscript submission. Afterwards the network can be made public and a digital object identifier can be requested.

reproducibility of subsequent research based on this network, a Digital Object Identifier (DOI) can be requested, which then can be used for referencing the network in publications. The network becomes immutable by this, so that no further changes can be applied, with the only exception of adding the publication reference to the network.

Until this point, only the public instance of the NDEx platform was mentioned, but the NDEx consortium provides the whole platform in open-source to enable private installations. This is particularly useful in cases where NDEx is used on an institutional level and privacy concerns limit the use, or even block access of the public instance.

Besides the web front-end, private and public instances provide programmatic access using the RESTful API. The web front-end of the NDEx platform uses this interface to communicate with the data storage back-end and an official python client is also available. Additionally a comprehensive documentation of the API is provided on the NDEx website, along with a machine reachable OpenAPI Specification (OAS; previously known as Swagger) describing the service.

2.4 Tools for Framework Development

2.4.1 Cytoscape

Cytoscape (45) is the leading and most powerful software for network visualization. It is an Java based open-source software developed and maintained by the Cytoscape Consortium. Initially it was designed for the visualization, data integration and analysis of biological networks, but evolved into a general purpose platform for the handling of complex networks of any kind. It provides integrated interfaces to the major database for biological networks and additional molecular genetics data (Table 3). Additionally, integration with the NDEx platform is natively included.

Table 3: Cytoscape built-in import interface to public databases.

Database	***Description***
bhf-ucl	GO annotation of cardiovascular disease-relevant proteins and microRNAs
ChEMBL	Manually curated database of bioactive molecules with drug-like properties
HPIDb	Host-pathogen interactions
IMEx	Non-redundant set of physical molecular interactions
IntAct	Free and open-source molecular interaction database (EMBL-EBI)
iRefIndex	Index of protein interactions available in other primary interaction databases
MBInfo	Fundamental information on mechanobiology
MINT	Experimentally verified protein-protein interactions manually curated from scientific literature (ELIXIR)
MPIDB	Physical microbial protein interactions
NDEx	Online commons for biological networks
Reactome	Free, open-source, curated and peer-reviewed pathway database
UniProt	Protein sequence and function derived from literature
VirHostNet	Virus-host protein-protein interactions

In contrast to other visualization tools, Cytoscape nodes and edges are not styled individually, instead visualizations are generated by mapping the data to visual properties. These so called attribute-to-visual-mappings allow a general definition of the visual representation and therefore provide more flexibility to changing data. Once the visual properties are defined, they can easily applied to different networks. Instead of repeating the single visualization steps on every new network, simply the attributes on which the visual representation is base can be adjusted in the mapping.

The single attributes are accordingly organized in different tables for nodes, edges and the network itself. The applied mappings thereby depend on the contained data, thus enabling different types of

mappings of the data (Figure 8). Discrete mappings assign a new value to the different manifestations of the attribute, e. g. nodes that hold the attribute value “DNA” or “RNA” are assigned to represented by rhombuses and hexagons respectively. Missing values are caught by default settings that apply generally as circles in the here presented example.

Another simple mapping can be created by simply passing the attribute values to the visual properties. This can be done for color set within the attribute data bur also for explicit node size or the displayed node label, just to name a few. Continuous mappings are a more complex mapping in which several threshold and corresponding values can be set. The resulting values for the attribute then are interpolated accordingly. Thereby the mapping is not limited to numerical values but also colors, creating gradient between the set threshold color values.

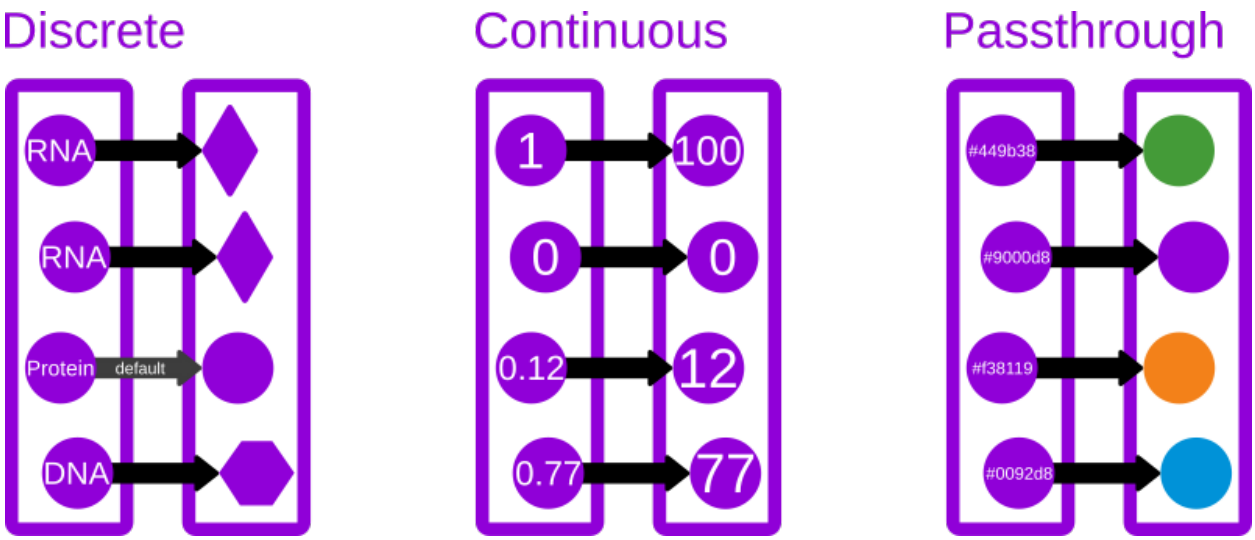


Figure 8: Attribute mapping types used in Cytoscape.

Cytoscape has a large set of layout algorithms built-in allowing the network to be presented using standard grid, circular, or hierarchical layouts, or using even more sophisticated, data-dependent algorithms like force-driven or heat diffusion based layouts. Also a comprehensive set of network analysis tools is provided, including investigation of subnetwork and pathway modules, or highly interconnected regions, or clustering algorithms.

A further feature of Cytoscape is the possibility to automate its usage through a provided REST API called CyREST (Figure 9). Using simple bash scripts or corresponding implementations in different programming languages allows to control Cytoscape to perform the same tasks as possible interactively. Furthermore, missing feature can be integrated into Cytoscape as apps developed by the community, from which already many are available at the integrated app store.

2.4.2 Cytoscape.js

For the web-based visualization of networks the Cytoscape consortium provides the JavaScript library Cytoscape.js (46), a successor of Cytoscape Web. The framework contains functions for network visualization and analysis that can easily integrated on websites or used for server-side rendering. Both, Cytoscape and Cytoscape.js share the same design concepts based on attribute-to-visual-mappings and can therefore easily adapted. Cytoscape even provides an export of network visualizations as interactive web application using Cytoscape.js.

However, although Cytoscape.js and the CX data format are related to Cytoscape, or even derived from the underlying data structure they are not natively compatible. Cytoscape.js is used to display the networks in CX format on the NDEx platform but to achieve an additional JavaScript library is necessary, which is provided in the official repositories (47).

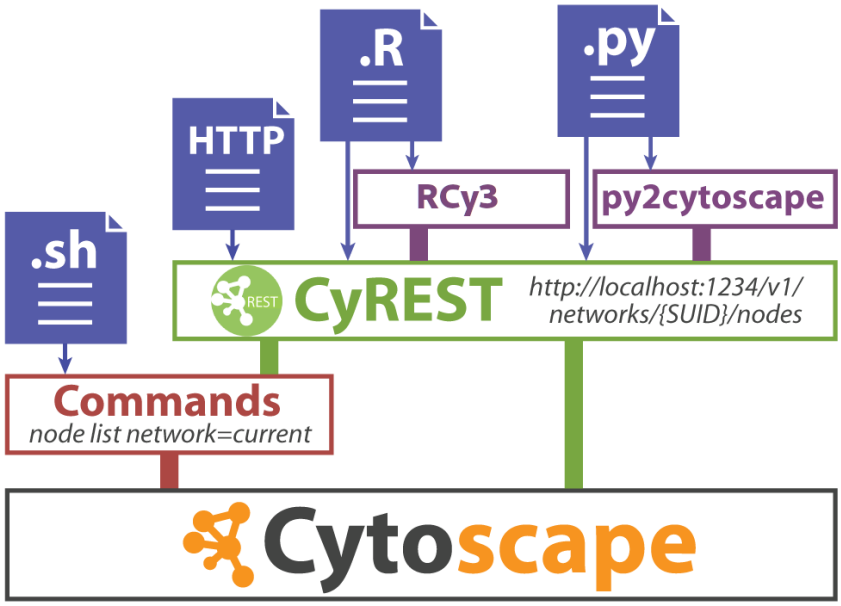


Figure 9: Interfaces for automated visualization in Cytoscape. Modified version from Cytoscape tutorials/train-the-trainers (77)

2.4.3 The R Statistical Programming Language

R (48) is a functional programming language widely adapted in the field of statistics and data science for data analysis and visualization, and statistical computing. In contrast to many other languages it evolves around the data and reflects this by it vector and table-based data types. A vast amount of freely available packages provided for extensions for specialized data structures, implementations state of the art statistical analyses, and interfaces to public data repositories. R provides an interface for the integration of C libraries which enables the usage of established and highly performant third-party libraries for native use within the language. Additional scripts and libraries can directly loaded from public code repositories, or from specialized and curated distribution systems like the Comprehensive R Archive Network (CRAN) (49) or Bioconductor (50) for the biological domain. Both require a comprehensive documentation and testing of the offered functionality, as well as active maintenance of the libraries. Because of this versatile software support researcher and data analysts use R as standard tool for data retrieval, cleaning, integration and visualization. Furthermore, various libraries offer visualization of their results using one of the standard libraries for graphs and networks.

a) igraph

The most prominent library for graph and network analysis is the *igraph* (51) collection which is natively written in C. It provides built-in methods for graph manipulation, analysis, and visualization and is supported by many additional packages extending its functionality. Its great advantages lie in

the comprehensible functions for graph analysis, while its capabilities for network visualization, although sufficient in most cases, could need improvements.

The visualization of networks using *igraph* follows the same scheme as plotting data in R in general (Figure 10). In contrast to Cytoscape the visualization of the graph depends on the manual definition of the visual representation for each node and edge individually. On the one hand, this simplifies the process because no abstraction is needed for its creation but therefore limits the application of the same visualization to different networks to sharing the corresponding source code or by providing custom functions implemented in specific packages.

b) Bioconductor graph

As complement of the igraph library in CRAN, Bioconductor provides its own native graph analysis and manipulation library called *graphNEL*, or simply *graph.* It is thought as addition to *igraph* with the purpose of enabling simple graph modeling within Bioconductor packages, but also provides functions for the lossless conversion between both. For admission to Bioconductor new packages are required to implement the *graph* model in addition to enclosed *igraph* functionality.

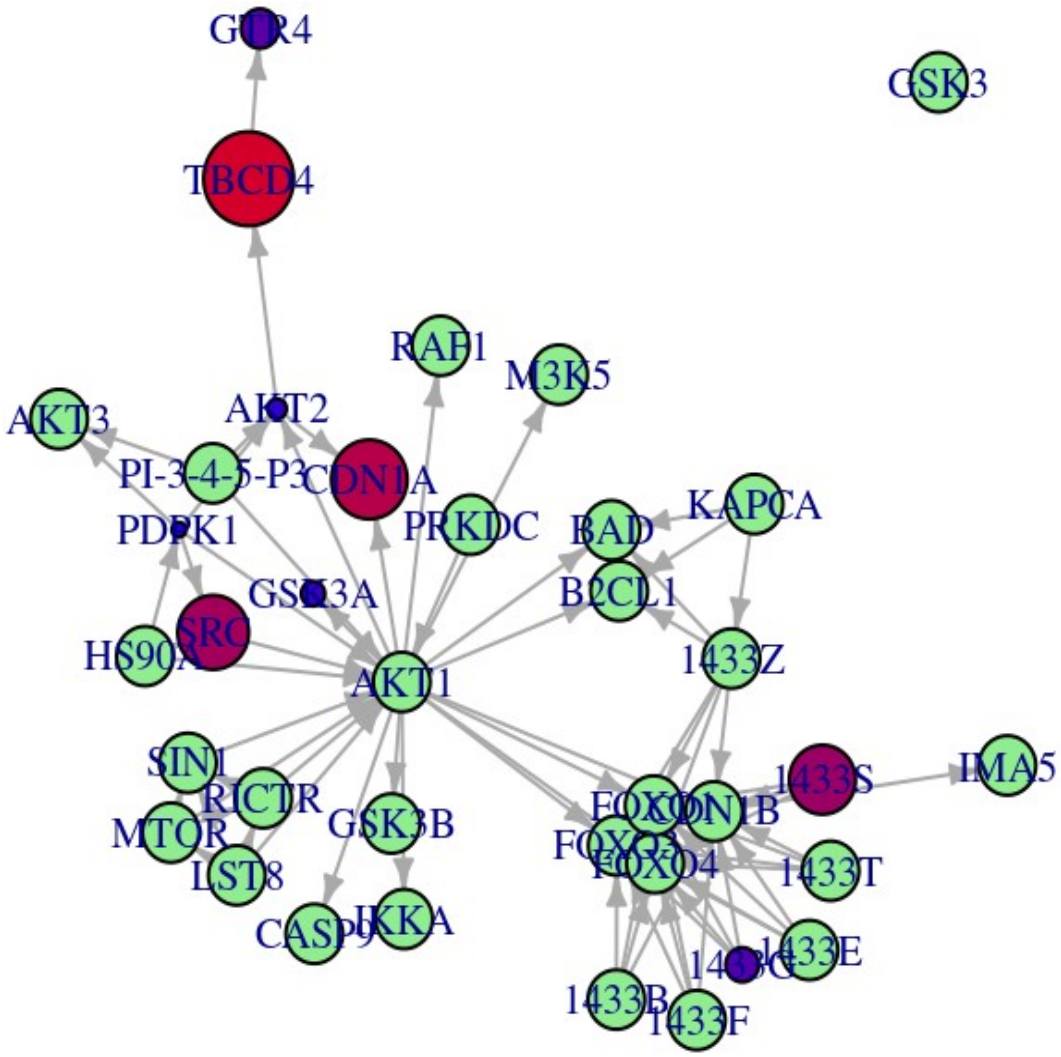


Figure 10: Network visualization with the plot function of the igraph packge.

c) RCy3

As before-mentioned, Cytoscape provides the CyREST interface for automation of the visualization tasks. The RCy3 (52) package provides and implementation on top of CyREST (Figure 9) and allows to control Cytoscape in this way. RCy3 is available on Bioconductor and thus implements both, *igraph* and *graph* for the network data exchange.

2.5 HL7 FHIR Standard

Beside the more recent efforts in interoperability within bioinformatics, the exchange of medical data in clinical environments always has been under strong aspiration. Commonly, the various modules and systems in a hospital communicate with “different languages”, varying in formats, or even concepts. Thus, integration of new tools into existing systems often involves a great effort and expense.

Out of this reason, the international organization HL7: Health Level Seven was founded in the year 1987, which focuses on the development for data exchange standards in health care (53). The aim of this organization is to standardize communication in hospitals and the entire healthcare system.

The popular HL7 FHIR standard, which is the abbreviation for Fast Healthcare Interoperability Resources, was released in 2018 and is strongly pushed into deployment from various leading health care organizations in the world (53). The standard is build on top of the previous HL7 versions with lessons learned from the flexibility but non-uniformness of version 2, and the over-complicated version 3. The principles behind FHIR are a strong focus on implementation simplicity, extensive tool sets composed of modular components, free-for-use specification, building on established web-technologies like REST, XML and JSON (Code 6), a human-readable serialization format, stability and concise as well as intuitive specifications (54). Furthermore, it follows the 80% rule for the wide variability caused by diverse healthcare processes, which can be defined as providing specification only for elements present in the most implementations (80%), whereas the remaining elements are covered by extensions (20%).

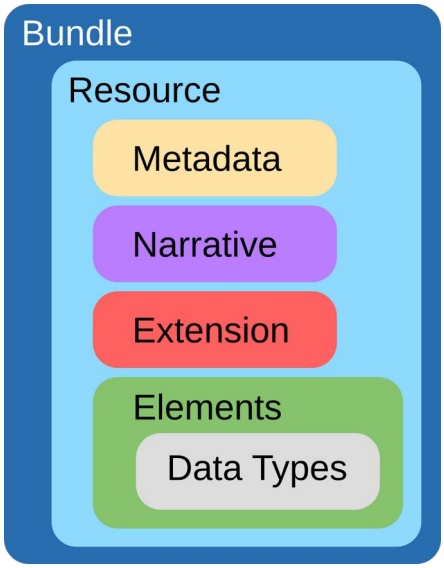


Figure 11: Composition of FHIR elements: All resources contain metadata about the resource, a human readable narrative, a declaration of implemented extensions, and the actual resource data using defined data types. Bundles collect several resources for retrieval.

The central building blocks of the FHIR standard are the FHIR resources. A FHIR resource is a compact, logically discrete unit for data exchange, which has a clearly defined structure (Figure 11), behavior and an unambiguous semantic. Also based on the 80% rule, the core specification of a resource consists only of data elements that are commonly prevalent. The data elements of a FHIR resource can range from structured data like simple values, modifiers or terminologies, to more dynamic narrative variables. Core examples for FHIR resources are patients, organizations, procedures and medications (Figure 12).

Code 6: Example of a FHIR Observation resource with Observation-genetics extension. The background is highlighted according to its association in Figure 11.

```
{
      "resourceType": "Observation",
      "id": "21289",
      "meta": {
            "versionId": "1",
            "lastUpdated": "2022-02-10T21:18:28.953+00:00",
            "source": "#87WWr5xOwKcSGDEk"
      },
      "text": "<div xmlns=\"http://www.w3.org/1999/xhtml\">
      <p><b>Observation genetics - Gene</b></p><p><b>id</b>: 21289</p>
      <p><b>code</b>: BAD </p>
      <p><b>system</b>: https://www.genenames.org </p>
      <p><b>subject</b>: Patient/1 </p>
      <p><b>specimen</b>: <a>Molecular Specimen ID: 14</a></p></div>",
      "extension": [ {
            "url": "http://hl7.org/fhir/StructureDefinition/observation-
geneticsGene",
            "valueCodeableConcept": {
            "coding": [ {
                  "system": "https://www.genenames.org",
                  "code": "BAD", "display": "BAD"
            } ] } } ],
      "identifier": [ {
            "system": "study_internal_id",
            "value": "Mucositis-BREN11:ENSG00000002330"
      } ],
      "status": "final",
      "code": {
            "coding": [ {
            "system": "http://hl7.org/fhir/ValueSet/observation-codes",
            "code": "48018-6",
            "display": "Gene studied"
            } ] },
      "subject": { "reference": "Patient/1" },
      "valueInteger": 6081077,
      "specimen": { "reference": "Specimen/14" },
      "derivedFrom": [ { "reference": "MolecularSequence/3487" } ]
}
```

For high-throughput biological data, FHIR provides the Genomics extension. Through the up-rising application of next-generation sequencing based tests in the clinical routine, the Genomics extension started to establish basic FHIR resources for encoding distinct genetic variants from sequence analyses (55). However, through the extension focus on only sequence variation representation, expression profiles are neglected, which prevents molecular profile interoperability like gene expression results as well as integration of clinical decision support systems.

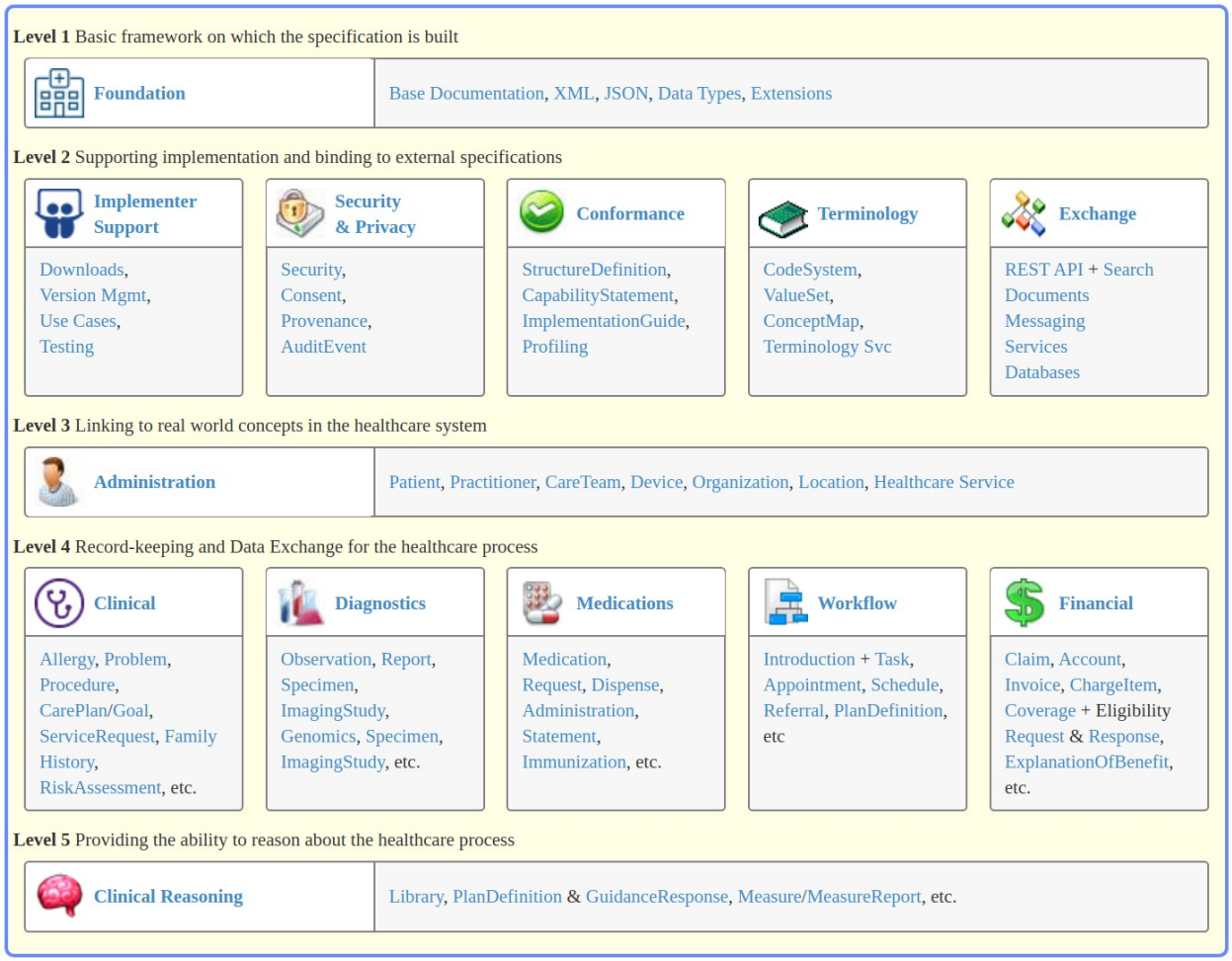


Figure 12: Overview of the FHIR specification including its different modules (Levels). Image retrieved from the official documentation (54).

3 Cumulative Publications

3.1 RCX – an R package adapting the Cytoscape Exchange format for biological networks

Florian Auer, Frank Kramer

This paper was published in *Bioinformatics Advances* in March 2022:

Bioinformatics Advances, Volume 2, Issue 1, 2022, vbac020
doi: https://doi.org/10.1093/bioadv/vbac020

Supplementary data:

– 01. RCX - an R package implementing the Cytoscape Exchange (CX) format
– 02. Creating RCX from scratch
– 03. Extending the RCX Data Model
– Appendix - The RCX and CX Data Model
– RCX Reference Manual
– RCX Cheat Sheet

Software availability:

Bioconductor: https://bioconductor.org/packages/RCX

Github: https://github.com/frankkramer-lab/RCX

3.1.1 Summary and discussion

Data formats, and their representation differ by the domain for which they were designed, and thus evoke incompatibilities in their usage. A clear example where this takes effect is when established web development patterns meet with data-centric perspective of R. JSON as standard transmission format focuses on a dynamic object-oriented structure with nested elements while R data structures consider data in a strict column- and table-like manner. With the continuous trend of integrating web-based resources and services into classical data science and statistical workflows major conflicts are inevitable.

This paper presents the R package *RCX* for the integration of the Cytoscape exchange format into the statistical programming language by adapting the data structure to standard R data types. The package therefore introduces the novel RCX data structure and provides functions for conversion, handling, validation, and visualization of the network data, thus overcoming the fundamental differences between data modeling in R and web-based data transmission.

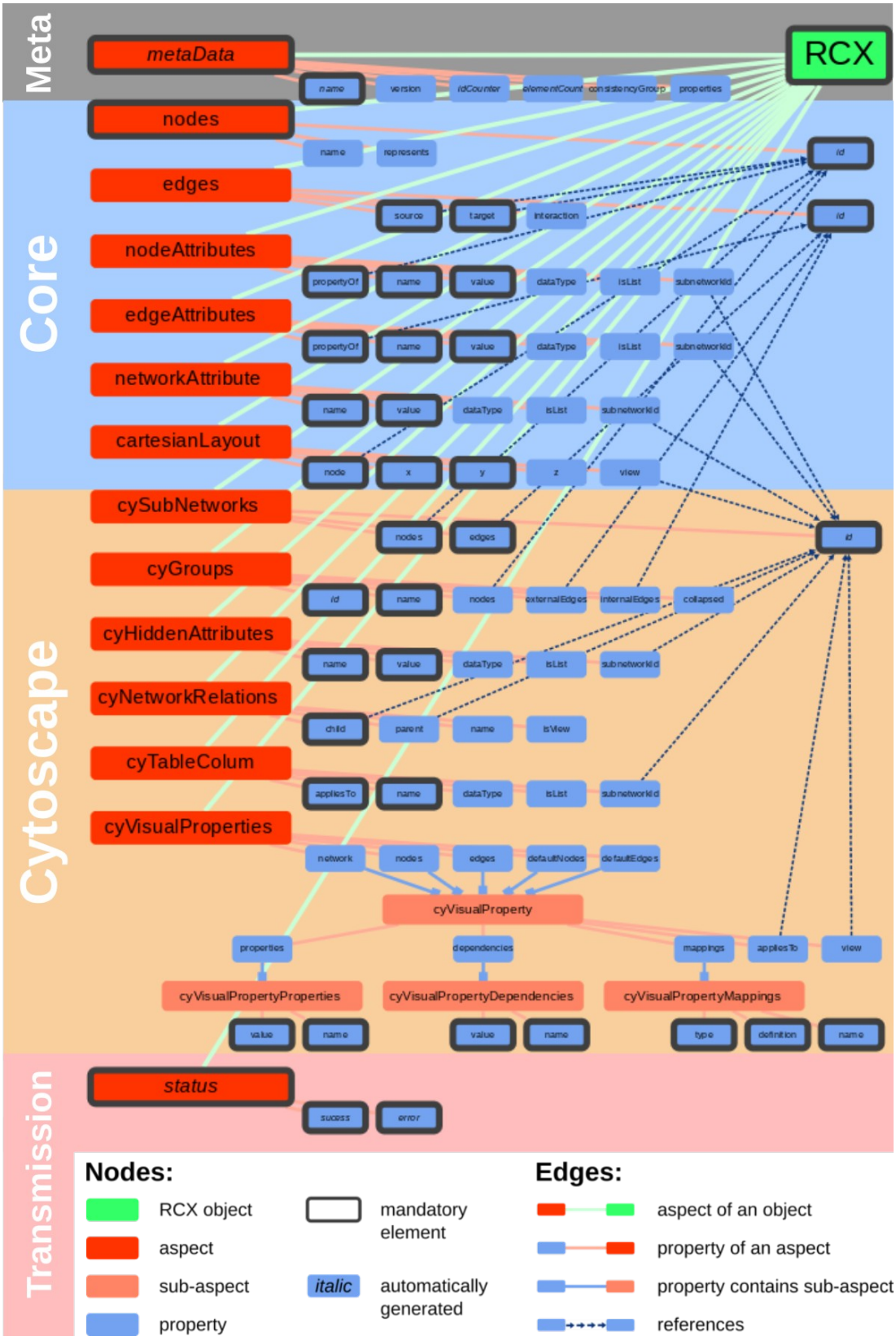


Figure 13: Structure of RCX with its aspects and corresponding properties. *The aspects are categorized as meta information, core and transmission aspects, and those derived from Cytoscape with corresponding sub-aspects. IDs, referencing, and optional properties, as well as automatically generated entities are highlighted. It is available on NDEx by the UUID ebdda4da-2ca5-11ec-b3be-0ac135e8bacf*

The *RCX* package implements an R data model of the same name composed of individual models for corresponding aspects of the CX data structure (Figure 13). However, due to the before-mentioned structural differences some adjustments have been made especially for aspects of the meta information and transmission categories, and the visual mappings defined by the *cyVisualProperties* aspect.

The meta information is generated automatically based on the contained data in the RCX object. Only the versioning and meta data properties can be updated manually to prevent inconsistencies in the converted CX model, and therefore rejection by other software like the NDEx platform. Similarly, aspects for transmission are created on-the-fly. The *cyVisualProperties* aspect had to be split into several linked sub-aspects (Code 7) to efficiently treat the data-dependent visual mappings.

Code 7: Class hierarchy of the implementation of the cyVisualProperties aspects in the RCX package

```
CyVisualProperties
 ├──network = CyVisualProperty
 ├──nodes = CyVisualProperty
 ├──edges = CyVisualProperty
 ├──defaultNodes = CyVisualProperty
 └──defaultEdges = CyVisualProperty

CyVisualProperty
 ├──properties = CyVisualPropertyProperties
 │                  ├──name
 │                  └──value
 ├──dependencies = CyVisualPropertyDependencies
 │                     ├──name
 │                     └──value
 ├──mappings = CyVisualPropertyMappings
 │                 ├──name
 │                 ├──type
 │                 └──definition
 ├──appliesTo = <optional reference to node or edge id>
 └──view = <optional reference to subnetwork id>
```

The package not only allows to define these visual properties, it also uses them to produce a visualization of the network consistent with Cytoscape and on the NDEx platform. They are implemented by integrating Cytoscape.js and the official NDEx JavaScript library (47) for translation of CX to Cytoscape.js compliant format into R functions. Those produce visualizations in Rstudio and external web-browsers, or export the visualization as single-page HTML visualizations that can be embedded in other applications. Usually visualization requires the positions of the nodes provided, that means individually defined in *cartesianLayout* aspect. The visualization functions also allow to apply the different layout algorithms provided by Cytoscape.js, and even bypass stipulated coordinates to circumvent this limitations and enable more flexibility in the network exploration.

To facilitate working with network in the RCX format the package includes functions to create and update the data model, its aspects, and sub-aspects. Thereby many of the parameters of the functions are optional and calculate the required properties automatically or derive them

from provided data. For example the IDs for respective aspects are attached automatically or continued when new data is added. All attribute aspects require a specification of the data type, including list version of those, which both is inferred from the R data types if not set explicitly.

The usefulness of the data models heavily depends on it interoperability with other data formats, mainly the CX data structure, but also with established libraries in R like *igraph* and *Bioconductor graph*. The *RCX* package implements for both functions for the lossless conversion between the different formats. The interchangeability with the CX format furthermore requires insurance of the data integrity, realized by validation functions on network and aspect level. Not only the data structure and types are verified but also important semantics like ID and property uniqueness, completeness of the contained elements, and references between the different components.

The packages makes great use of generic functions, especially for printing, summarizing and validating the network and aspects, or updating those. Moreover, the choice for this architectural decision was made regarding the dynamic aspect-oriented structure of CX, and hence the extensibility of RCX data model. All functions involved in creation, manipulation, representation, and conversion are implemented generic and use method dispatch to delegate the tasks to the appropriate functions of the RCX model or its aspects and subaspects. This way it is possible to extend the RCX data model to custom aspects by implementing functions for the appropriate generics (as demonstrated for MetaRelSubNetVis in Chapter 3.5) .

3.1.2 Declaration of contribution

Based on the CX format I developed the architecture of the package and implemented the package from scratch using the above listed external R and JavaScript libraries. Furthermore, I wrote the package documentation, accompanying vignettes, supplementary data to the manuscript, and created the on the NDEx platform available network illustrating the RCX data structure. Prof. Kramer offered guidance in software architectural ambiguities. The manuscript was written and revised by me, while Prof. Kramer provided feedback on the initial draft and approved the final version.

3.2 ndexr—an R package to interface with the network data exchange

Florian Auer, Zaynab Hammoud, Alexandr Ishkin, Dexter Pratt, Trey Ideker, Frank Kramer

This paper was published in *Bioinformatics* in October 2017:

Bioinformatics, Volume 34, Issue 4, 15 February 2018, Pages 716–717;
doi: https://doi.org/10.1093/bioinformatics/btx683

Supplementary data:

– A brief introduction to the data structures
– ndexr Reference Manual
– ndexr Cheat Sheet

Software availability:

Bioconductor: https://bioconductor.org/packages/ndexr

Github: https://github.com/frankkramer-lab/ndexr

3.2.1 Summary and discussion

Beside the availability of an adequate data format for conversion and handling of biological networks from online resources within R, a potent, easy to use, and reliable interface with those is equally essential. Modern web-resources offer communication capabilities farther from serving as simple online storage for retrieval. Accordingly, the possibilities to interact with the NDEx platform go beyond basic search and access of available networks. Integration of the full potential of the platform provided by its RESTful API enables to establish collaborative work in a programmatic manner.

The here presented paper exhibits the implementation of the above by *ndexr* — an R package to interface with the network data exchange. The package straightforwardly enables CRUD operations on networks in context with the NDEx platform but furthermore promotes associated administrative actions. To take full advantage of both authentication against the NDEx platform is required. This presupposes as user management, and NDEx also incorporates organization into groups, which all is manageable by function from the *ndexr* package.

ndexr allows to search the platform with the same capabilities as the web-interface including in functions incorporated search parameters. After users login at the NDEx platform with their credentials networks can be updated or even deleted, appropriate rights to the network presumed. For network a user owns those rights can be adjusted by assigning read or write permissions to individual users or groups. Moreover, even users and groups can be managed the same way as networks: right for participation can be customize and revoked, and both can be created on demand. The user management goes even one step further and allows to change and reset passwords by email.

Despite these functionality for administration are networks the central element of the platform and hence provide further configuration options. The public visibility of networks can be regulated to be

accessible by its UUID or listed in search results of different users. Also the network properties used for display in the NDEx web-application, that are also part of the network, e. g. name, description and version, can be changed and set without retrieving the network beforehand. Furthermore, the network ownership can be delegated to different users, all by functions provided by *ndexr*.

The initial network model included in *ndexr* proved in practical applications to be insufficient for broad adoption and lead to the development of the previously presented RCX data model. The complexity of both packages required efforts to assure maintainability, and to apply good developmental practices, both packages, RCX and *ndexr,* were further developed distinctly by separating data modeling from exchange accordingly.

3.2.2 Declaration of contribution

The project was initialized by Dexter Pratt, Trey Ideker, and Alexandr Ishkin and a draft of the package was started by Alexandr Ishkin. I took over the development of the package and replaced the majority of the existing source code with my own implementation and adjusted the remainder for consistency in function naming, used parameters, and usage of package internal functions. The supplementary data to the manuscript was also contributed by me. Zaynab Hammoud supported in R related questions on implementation details and gave feedback on the manuscript. The manuscript itself was written and revised by me, while Prof. Kramer provided feedback on the initial draft, approved and submitted the final version, and supervised the review process.

3.3 Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer

Hryhorii Chereda, Annalen Bleckmann, Kerstin Menck, Júlia Perera-Bel, Philip Stegmaier, Florian Auer, Frank Kramer, Andreas Leha, Tim Beißbarth

This paper was published in *Genome Medicine* in March 2021:

Genome Med. 2021 Mar 11;13(1):42;
doi: https://doi.org/10.1186/s13073-021-00845-7

Software availability:

GLRP:	http://mypathsem.bioinf.med.uni-goettingen.de/resources/glrp
	https://gitlab.gwdg.de/UKEBpublic/graph-lrp
MetaRelSubNetVis:	http://mypathsem.bioinf.med.unigoettingen.de/MetaRelSubNetVis

3.3.1 Summary and discussion

Cancer prognosis is generally difficult to predict and underlies various aspects, and understanding these relations allow the development of better treatment and thus foster patient survival. By measuring the molecular conditions we try to derive the genetic factors responsible for tumor progression and metastatic events. The high-dimensionality of measured gene expression makes it challenging to attribute the reason to certain genes. To improve the predictions, available background knowledge can be used to better understand the molecular relationships.

Deep learning methods have show great successes in all domains of science, and as well on learning features from complex biological data. Convolutional neural networks (CNNs) have proven their advantages on image data, have great potential for their application to molecular data, and gene expression in particular. Graph convolutional neural networks (Graph-CNNs) (56) furthermore are their extension to work on networks natively, and thus allow the integration of biological networks into the model as prior knowledge. However, deep learning models are generally considered as black-box-models, in which the underlying decisions contributing to the outcome of a prediction can not be back-traced to the those effecting entities. Estimation of the relevance of single genes to the predicted metastatic events is essential for understanding the molecular nature, and thus develop improved treatment strategies on this insights.

a) Generation of patient-specific gene relevance scores

The patient-specific relevance scores for the single genes are based on the deep learning model for predicting the cancer progression in the first place. A Graph-CNN is trained on the gene expression and molecular network data and the metastatic status predicted for every patient. Convolutional neural networks were developed for image data but gene expression data does not contain information of the connection between the genes. These linkages between neighboring pixels are emulated by their distance in HPRD PPI network. The architecture of the Graph-CNN consisted of

two graph convolutional layers containing 32 filters covering a neighborhood of 7 nodes, followed by a maximum pooling of 2, and two hidden fully connected layers with 512 and 128 nodes respectively.

To estimate the predictive performance of the GraphCNN model a 10-fold cross validation over a whole dataset was executed and compared to the performance of a Random Forest model without any prior knowledge as baseline, and a network-constrained sparse regression model using the HPRD PPI network (R package glmgraph, (57)). The results on the performance for the compared models is presented in Table 4 and shows similar great results on all methods with Graph-CNN leading.

Table 4: Performance of the different methods on predicting metastatic events in patients

Method	***AUC (%)***	***Accuracy (%)***	***F1-weighted (%)***
Graph-CNN	82.57±1.25	76.07±1.30	75.82±1.33
Random Forest	81.27±1.66	74.23±1.73	73.47±1.84
glmgraph with standardized gene expression	82.16±1.25	76.18±1.36	75.86±1.35

To determine the relevance of the genes for the predictions the GLRP algorithm was developed and applied to the patients' prediction. The relevances represent relevant walks from the input gene to the predicted outcome and are generated for each patient individually. To minimize the information flow from previous prediction, the gene expression dataset was randomly split in training (90%) and test (10%) set using manually selected hyperparameters from the previous 10-fold cross validation. From the 97 patients in the test set only 79 patients with matching predicted and reported metastasis were considered the generation of patient-specific subnetworks. Application of the GLRP method to those patients then produces the relevance score of the genes to the prediction.

b) Patient-specific subnetworks

The genes of the breast cancer data set were mapped onto the PPI network from HPRD consisting of a disconnected graph of 9,898 vertices. This decreased the network to 7,168 vertices in 207 connected components, from which only the main connected component was used for further analysis. This component contains 6,888 vertices, in contrast to the remaining components with only 1 to 4 vertices. The Graph-CNN algorithm requires a connected graph as input which was the initial reason for this pre-selection.

The 140 most relevant genes were used for each of the 79 patients to induce intermediate patient-specific subnetworks. Those subnetworks were combined and again only the main connected component used for further processing, consisting of 407 nodes. The relevant genes from the therein contained patient-specific subnetworks were used for subsequent analysis.

Actionable genes withing the patient-specific relevant were identified using the a modified version of Molecular Tumor Board (MTB) report (referred to as "MTB report") (58) method. The gene expression levels served as proxy for gain and loss of function alterations by assuming high and low expression respectively for both. These were derived from the gene expression throughout the whole patient cohort with the 75% and 25% quantiles as boundaries for high, normal and low levels of expression.

The results of the MTB report were integrated into the combined subnetworks, together with the relevance scores, gene expression values and levels, and information of patients survival, cancer

types and subtypes. These network then served as basis for visualization in the newly for this purpose developed MetaRelSubNetVis web-app, and analysis of the results using this application.

c) Analysis of relevant genes

For evaluation of the gene relevance a pathway enrichment analyses was performed on annotated signal transduction pathways in the TRANSPATH® database (version 2020.1)(59) using the upstream analysis based on the Fisher's exact test (60) provided by the geneXplain platform (version 6.1)(61). The generated subnetworks contained common potential oncogenic drivers suggesting the extracted pathways to be fundamental to cancer. Their biological relevance is supported by findings in for specific subtypes: in estrogen receptor positive patients genes were associated with hormone receptor-positive breast cancer (e.g. CD36, ESR1, GLUL, IL6ST, RASA1), as well as LumA breast cancer genes associated with the basal-like subtype (e.g., AKT1, EGFR, SOX4, and high levels of HNRNPK). Additionally, a significant enrichment of pathways already associated with cancer disease mechanisms has been detected, such as the EGF, ER-alpha, p53, and TGFbeta pathways as well as Caspase and beta-catenin networks.

d) Patient-specific subnetwork visualization

MetaRelSubVis is an Angular based web-app for the visual exploration of the integrated patient-specific subnetworks. The main feature of the app is that the nodes of the single subnetworks remain positioned in place while different patients, thresholds, and data based styles for coloring and sizing are applied (Figure 14). Thereby the single nodes can be arranged and highlighted by selection. The application was also used to produce the publication-ready network visualization.

MetaRelSubNetVis is also used for the comparison of metastatic and non-metastatic patients within Basal and LumA breast cancer subtypes. For both comparisons are the nodes highlighted by relevance, gene expression and gene expression level (Figure 15). The position of the same genes is consistent in all subplots. Genes that are only present in one class of cancer progression are colored in gray with the corresponding half.

e) Findings of subnetwork investigation

MetaRelSubVis allows the exploration of the breast cancer subnetworks of all correctly predicted patients. It contributed to the investigation of the before-mentioned selected four patients and revealed interesting patterns in the gene expression. Figure 15 provides a visualization of the comparison of the metastatic versus non-metastatic patient in both breast cancer subtypes. Each comparison is highlighted by relevance, gene expression, and gene expression level to illustrate the different effect. The findings are marked in the single visualizations and correspond to the gene lists in Table 5.

Table 5: Genes relevant for cancer subtypes. The genes are listed by cancer subtype, namely BASAL and LumA. The table is arranged in the same order as Fig X. The gene lists are marked within the corresponding network visualizations.

	BASAL	**LumA**
	GSM519217 vs. GSM615695	**GSM615233 vs. GSM150990**
relevance	VIM	CAV1, PTPN11, FTL
expression	JUP, PCBP1, HMGN2	ESR1, VIM
level	MCL1, CTNNB1, EGFR, SOX4	RASA1, IL6ST, KRT19, RPS14

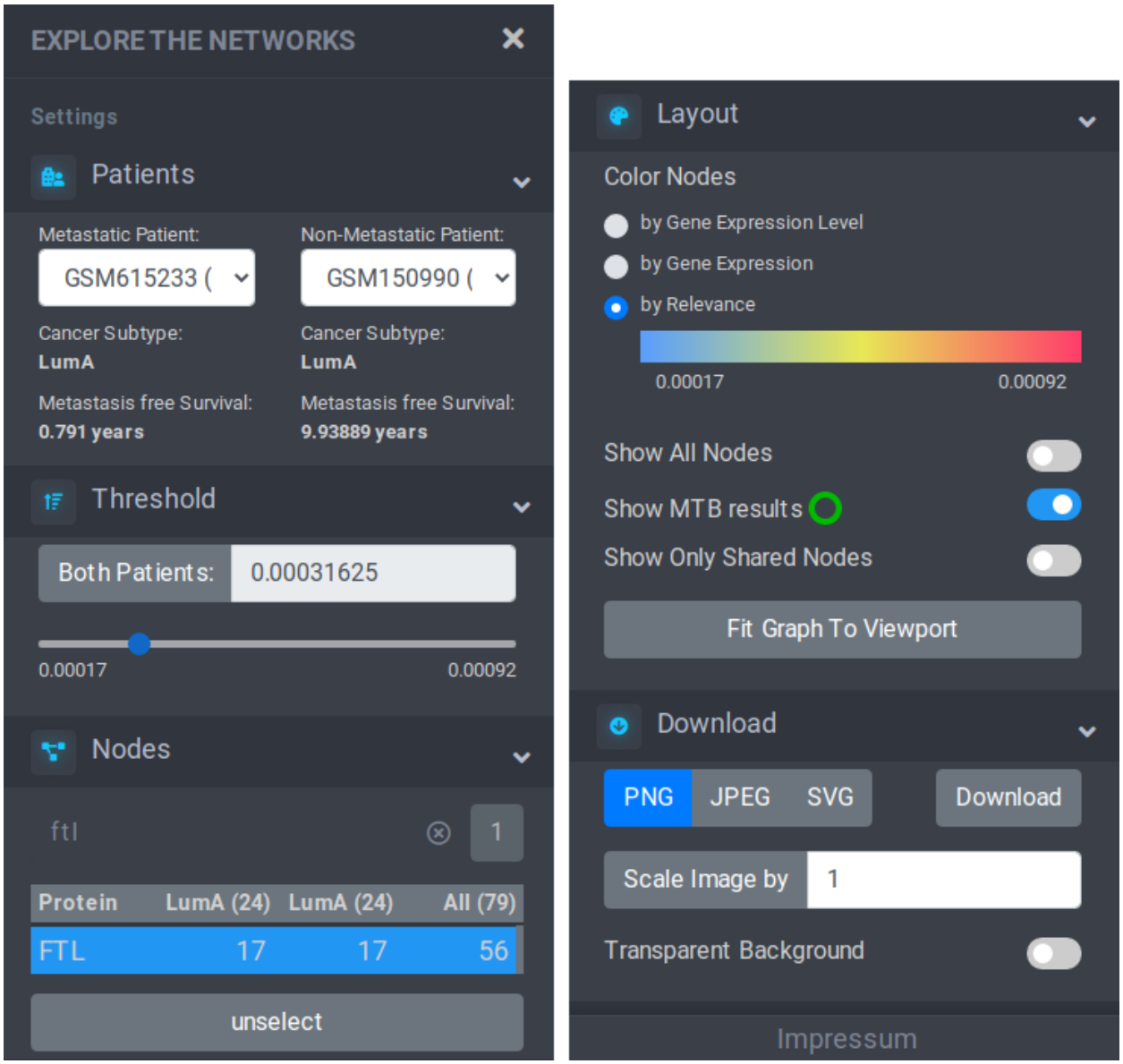


Figure 14: Customization options for the display of subnetworks within MetaRelSubNetVis.

The sole protein relevant to all 79 correctly predicted and the remaining 18 patient is the eukaryotic translation elongation factor EEF1A1, which protects tumor cells from proteotoxic stress. Figure 15 shows, that it is highly expressed and highly relevant all four patients, and is generally overexpressed in a majority of breast cancers (62). Another commonly shared gene between all patients is VIM, which is an important marker for epithelial-to-mesenchymal-transition (EMT)(63). Its high expression correlates with a motile, mesenchymal-like cancer cell state, and thus drives metastasis (64) This can be observed by the high expression in both selected metastatic patients, and low expression in the selected non-metastatic patients, while being relevant for the prediction for all four. The LumA subtype considered as estrogen receptor positive (65), which is consistent with the appearance of ESR1 in both patients of this subtype. In this subtype also holds the genes CAV1, FTL and PTPN11, which are known to be involved in aggressive tumor growth or therapy resistance (66–68).

Comparing the two cancer subtypes to each other reveals, that the generated subnetworks are capable to capture subtype specific features, meaning they only appear in one of the four selected patients: The genes EGFR, CTNNB1, MCL1 and SOX4 are often associated with poor prognosis in Basal subtypes, and only found there for the metastatic patient. On the other side, genes linked with better prognosis of the same subtype, namely JUP, HMGN2 and PCBP1 (69–71), and IL6ST, KRT19, and RASA1, RPS14 (72–75) for the LumA subtype are only found for the non-metastatic patient.

The biological relevance of the subnetworks is supported by findings in for specific subtypes: in estrogen receptor positive patients genes were associated with hormone receptor-positive breast cancer (e.g. CD36, ESR1, GLUL, IL6ST, RASA1), as well as LumA breast cancer genes associated with the basal-like subtype (e.g., AKT1, EGFR, SOX4, and high levels of HNRNPK). This suggests that our method successfully identified relevant key players with a general role in breast tumor genesis. Nevertheless, resistance mechanisms in breast cancer targeted therapies still found a major challenge due to the high variability of the interconnections within the network and thus of signaling pathways, that circumvent therapeutic target.

3.3.2 Declaration of contribution

Hryhorii Chereda and Prof. Beißbarth designed the study with Prof. Bleckmann, Prof. Kramer, and Philip Stegmaier providing major contributions. Hryhorii Chereda implemented, trained and evaluated the graph-based deep learning, as well as developed the graph layer-wise relevance propagation, and applied both to the breast cancer data. He also evaluated the performance of GCNN on MNIST dataset. Thereby, Andreas Leha provided guidance on machine learning machine learning methods and its validation. Prof. Beißbarth supervised the development and answered occurring questions on machine learning or related to the underlying biology within the study.

Pathway analysis with TRANSPATH database way performed by Philip Stegmaier form geneXplain. He also compared and evaluated the patient-specific subnetworks with results from the weighted gene co-expression network analysis.

I processed the results from the GCNN and consecutive GLRP and integrated them with the source networks, and the breast cancer patient and gene expression data. From those I generated the patient-specific subnetworks for individual patients and combined those to a connected network. Júlia Perera-Bel adapted the MTB report analysis to transcriptomic data and applied it to a by me provided list of relevant genes. The results then were added by me to the single and combined subnetworks, for which I generated the visualizations used in the publication. I also implemented MetaRelSubNetVis for the representation and investigation of the network based results. Based on this application Kerstin Menck investigated and evaluated the gene results for their genetic and biological significance under supervision of Prof. Bleckmann.

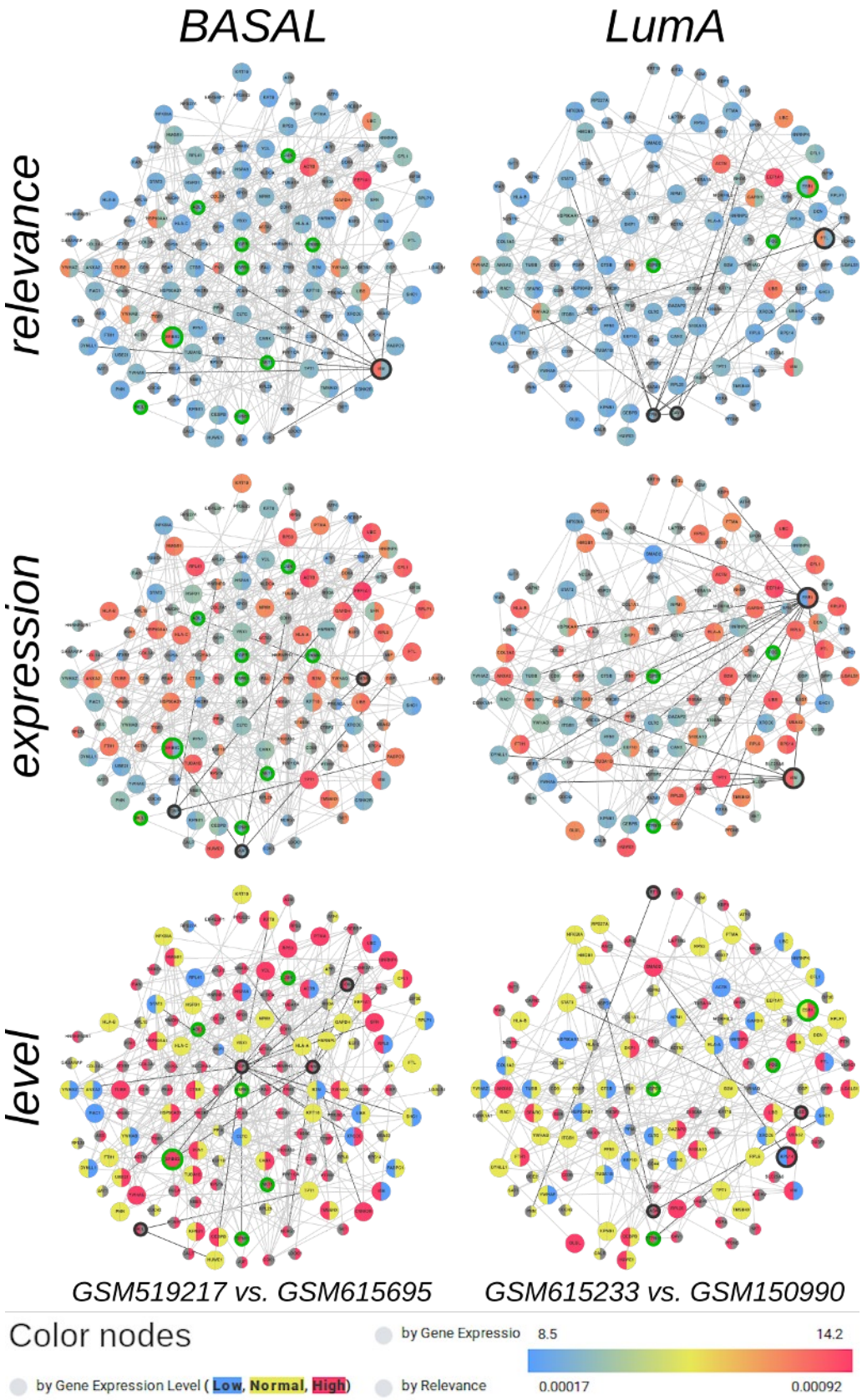


Figure 15: Comparison of the subnetworks of short vs. long survival of patients with BASAL (GSM519217 vs. GSM615695) and LumA (GSM615233 vs. GSM150990) breast cancer subtypes visualized by relevance score, and gene expression value and level.

3.4 Data-dependent visualization of biological networks in the web-browser with NDExEdit

Florian Auer, Simone Mayer, Frank Kramer

This paper was published in *PLOS Computational Biology* in June 2022:

PLOS Computational Biology 18(6): e1010205.
doi: https://doi.org/10.1371/journal.pcbi.1010205

Software availability:

source code: https://github.com/frankkramer-lab/NDExEdit

live version: https://frankkramer-lab.github.io/NDExEdit

3.4.1 Summary and discussion

In this paper, an application for the browser-based visualization of biological networks is introduced. Networks in the CX format can be loaded directly from the NDEx platform and their visual attributes adjusted, depending on the in the network included data. Integrated networks can contain complex relations that is associated with the individual nodes and edges, and can go far beyond name and type. Complex visualization require powerful tools to represent this information an easily comprehensible manner.

Application of visualizations to networks usually requires additional software to be installed like Cytoscape. This is often not possible for security reasons or when internet access is restricted limiting quick changes. The web application can be deployed on own servers to circumvent those limitations. NDExEdit is a simple to use, but no less powerful tool to quickly visualize biological networks within the web-browser. NDExEdit enables to enhance plain networks for presentation and publication by applying *attribute-to-visual-mappings* and explore the necessary data beforehand. The statistics view assists by selecting appropriate attributes from provided distribution and coverage charts for each and allows investigation of the network, without applying any changes. Additionally, matching criteria on the data can applied to be highlighted within the network visualization.

All network visualization can be exported directly to the NDEx platform, as CX-file, or as an image in PNG- or JPEG-format. The NDExEdit application illustrates the great potential of web-based visualization solutions for integrated and collaborative workflows in biological research (Figure 16). It narrows the gap between desktop clients to create, edit and beautify a network, and platforms to distribute them. The aim is not to replace established visualization software like Cytoscape but rather contribute to the community by closing this gap in working with biological networks.

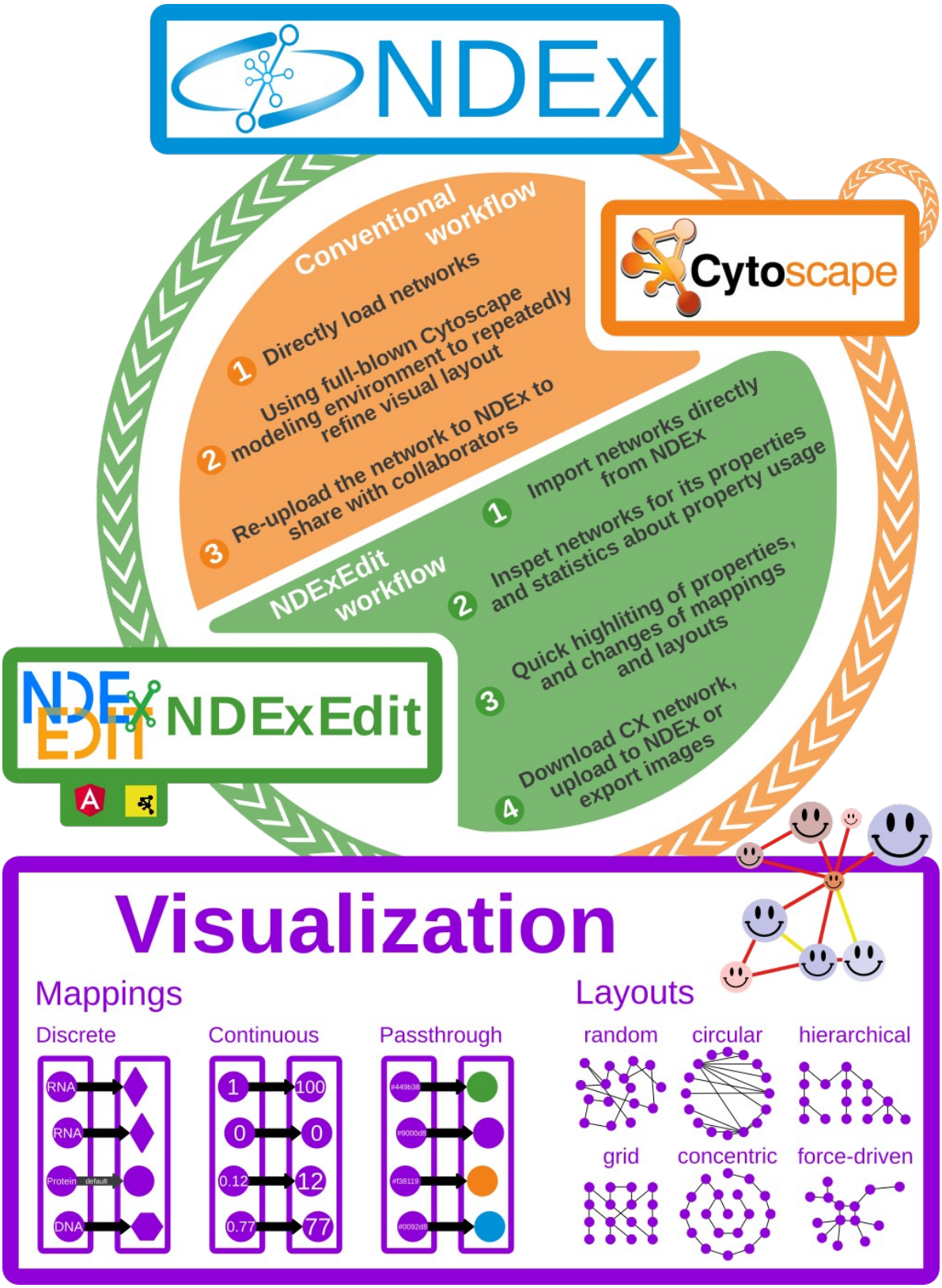


Figure 16: Comparison of the visualization workflow in NDExEdit and Cytoscape. The different visualization options for mappings and layouts are visually demonstrated.

3.4.2 Declaration of contribution

Simone Mayer implemented a first version of the application within the practical part of her Bachelor thesis and improved it afterwards in her role as student assistant, both under my supervision. I contributed for the implementation detailed insight on the CX data structure, the NDEx API, the definition of the Cytoscape derived mappings, and the layout algorithms. Simone Mayer was guided by me for the implementation of which she took over the majority, whereas I further took care of testing, error correction and code review, and lead the incremental releases.

The manuscript was written by me with feedback from Simone Mayer, while Prof. Kramer approved the final version.

3.5 MetaRelSubNetVis: Referenceable network visualizations based on integrated patient data with group-wise comparison

Florian Auer, Simone Mayer, Frank Kramer

A preprint version is available at bioRxiv:

bioRxiv 2022.04.18.488628;
doi: https://doi.org/10.1101/2022.04.18.488628

Software availability:

source code: https://github.com/frankkramer-lab/MetaRelSubNetVis

live version: https://frankkramer-lab.github.io/MetaRelSubNetVis

3.5.1 Summary and discussion

MetaRelSubNetVis is a web application that allows an interactive and comparative visualization of networks integrated with data from several groups and individuals. Users can investigate the results of preceding analysis with options to highly customize the view, as well as the selection options of those. The group-specific information can be inspected, visualized, and finally exported as images, or shared using custom links. A major aspect of MetaRelSubNetVis is that all nodes remain at the same position regardless the selected patients, layouts, thresholds, or highlighting. Furthermore, results for different groups within the network can be investigated side by side, thus foster a more comprehensible visual comparison of the contained data (Figure 17).

The networks can be directly loaded from the NDEx platform, not only promoting collaborative workflows through this platform but also prevent problems of incompatibilities due to differing data formats or finding individual hosting solutions for the created networks. By sharing a link to a specific visualization with set layout configuration facilitates collaboration, communication and exchange of network visualizations furthermore (Figure 17). These option also include specifications to hide specific parts of the sidebar or even the sidebar in total. This allows to embed a network visualizations within other web applications sparing the users of the development of proprietary visualization applications.

Individual visualization of networks require the definition of the contained data and its attributes for representation of the corresponding features within the application. This information is contained in a custom aspect for the CX data model (see Code 8) which will be treated opaquely by the NDEx platform. For usage of the custom aspect within other applications, a corresponding implementation extending the RCX data model (Chapter 3.1) is provided along the application.

MetaRelSubNetVis has already proven its great potential by its application in Chapter 3.3, where it was successfully used for the exploration, interpretation and visualization of the created patient-specific subnetworks. Since its introduction it underwent further progression to simplified its usage and allowing simple adoption for distinct integrated networks. Implementation of referable

visualizations provides a key stone for collaborative work on integrated networks and a powerful option to integrate network visualizations in further applications.

Code 8: Costom CX aspect metaRelSubNetVis with its own structure and properties

```
{“metaRelSubNetVis”: [{
    “highlight" : "#000000",
    "properties" : [
        {"property": "Occurrence", "label":"Occurrence",
         "type":"continuous", "threshold":true,
         "mapping":{"1":"#c7c7c7", "79":"#388eff"}},
        {"property":"qvalue", “label":"q-value",
         "type":"continuous", "threshold":true,
         "mapping":{"1.0":"#c7c7c7", "0.0":"#ff0000"}},
        {"property":"significant", "label":"significantly DE",
         "type":"boolean", "mapping":{"true":"#00ff00"}}
        ],
    "individual_properties" : [
        {"property":"GE", "label":"Gene Expression",
         "type":"continuous", "threshold":true,
         "mapping":{"8.5":"#599eff",
                    "11.4":"#e8e857",
                    "14.2":"#ff3d6a"}},
        {"property":"GE_Level", "label":"Gene Expression Level",
         "type":"discrete",
         "mapping":{"LOW":"#599eff",
                    "NORMAL":"#e8e857",
                    "HIGH":"#ff3d6a"}},
        {"property":"Score", "label":"Relevance",
         "type":"continuous", "threshold":true,
         "mapping":{"0.000298":"#599eff",
                    "0.00061":"#e8e857",
                    "0.000922":"#ff3d6a"}},
        {"property":"MTB", "label":"MTB results",
         "type":"boolean", "mapping":{"true":"#00bb00"}}
      ]
}]}
```

3.5.2 Declaration of contribution

The first implementation of MetaRelSubNetVis was done by me in the course of my contribution to the work presented in Chapter 3.3. Simone Mayer helped in implementing the improvements made afterwards in her role as student assistant under my supervision. For the advancements I contributed for the implementation detailed insight on the CX data structure, the NDEx API, and the specialized concentric layout algorithm. Furthermore, I developed the custom CX aspect and implemented its extension for the RCX data model. Simone Mayer was guided by me for her implementation tasks, and additionally I took care of testing, and code review, and lead the incremental releases. The manuscript was written by me with feedback from Simone Mayer, while Prof. Kramer approved the final version.

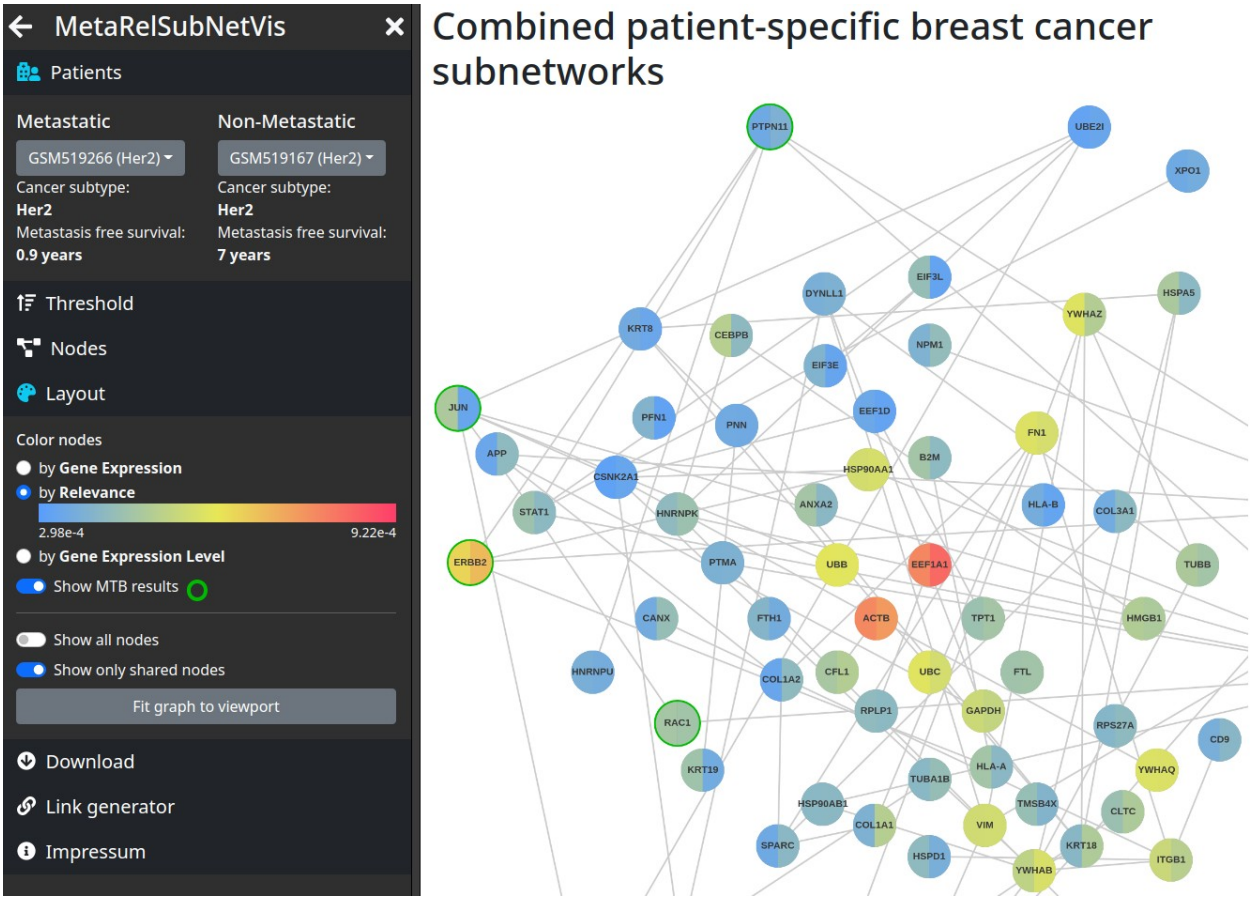


Figure 17: Comparative visualization of the patients GSM519266 and GSM519167 from the Combined patient-specific breast cancer subnetworks reachable through https://frankkramer-lab.github.io/MetaRelSubNetVis?uuid=a420aaee-4be9-11ec-b3be-0ac135e8bacf&pa=GSM519266&pb=GSM519167&th_GE=8.532345888264551&th_Score=0.00029828155&col=Score&size=GE&all=false&shared=true&bool=MTB&sb=0&cP=1&cT=1&cN=1&cL=0&cD=1&cG=1&cIm=1

3.6 Reproducible data integration and visualization of biological networks in R

Florian Auer, Hryhorii Chereda, Júlia Perera-Bel, Frank Kramer

A preprint version is available at bioRxiv:

bioRxiv 2022.04.15.488519
doi: https://doi.org/10.1101/2022.04.15.488519

Software availability:

source code: https://github.com/frankkramer-lab/reproducible-network-visualization

documentation: https://frankkramer-lab.github.io/reproducible-network-visualization

3.6.1 Summary and discussion

This manuscript demonstrates the in Chapter 3.3 performed workflow for the generation of patient-specific subnetworks for a large breast cancer data set in more detail, along with a collection of tools to perform network data integration and customized visualization of group- and network-wise attributes. To enable reproducibility of the all necessary steps the network model was updated to the *RCX* data structure (Chapter 3.1) and the intermediate and final results were stored on the NDEx platform using the *ndexr* package (Chapter 3.2). Also several software tools for the visualization were explained in detail with their individual usage or in combination to produce equivalent results. Those mainly evolve around the NDEx platform which allows the documentation of the performed integration steps and serves as interface for the used web-based solutions. The inclusion of the visualization within the networks contributes to the comprehensibility of the results as well as fosters compatibility and flexibility across the different tools.

The illustrated tools represent a broad range of possibilities of network visualization. A standard R visualization of networks is shown with the *igraph* package, and more complex options are demonstrated with the RCX package which natively includes the feature to store the visualization within the networks. The visualization is based on the *attribute-to-visual-mappings* used by Cytoscape and the NDEx platform in a congruent manner. The usage of Cytoscape to produce the same visualization by being remotely controlled from within R with *RCy3* is also demonstrated based on the *RCX* data model. A equivalent visualization is also demonstrated using NDExEDIT (Chapter 3.4) with the NDEx platform as source for the networks and to store the styled results. Finally, the fully integrated network stored on the NDEx platform can be used directly within MetaRelSubNetVis (Chapter 3.5) to enable interactive comparison and visual exploration of the enriched network, as well as to provide and platform for sharing specific visualizations of the data. Only the combination of the presented software tools, platforms and packages promotes an integrated environment for reproducibility of integration, exploration, and visualization of integrated network data.

3.6.2 Declaration of contribution

Hryhorii Chereda provided insight on the results of the preceding deep learning and relevance propagation. I processed the results, integrated the data and generated lists of relevant genes for each patient, and the patient-specific subnetworks for individual patients as well as a combined network. Júlia Perera-Bel adapted the MTB report analysis to transcriptomic data and applied it to the generated list of relevant genes. Based on the collected results I created their different visualizations and the documentation of them. However, for publication I also re-wrote the MTB report scripts to work without downloading the additional required data, improved the performance, and added documentation.

3.7 Adaptation of HL7 FHIR for the exchange of patients' gene expression profiles

Florian Auer, Zhibek Abdykalykova, Dominik Müller, Frank Kramer

This paper was published in *Studies in Health Technology and Informatics* in June 2022:

Stud Health Technol Inform. 2022 Jun 29;295:332-335.
doi: https://doi.org/10.3233/shti220730

Software availability:

source code: https://github.com/frankkramer-lab/gene-expression-on-fhir

live version: https://frankkramer-lab.github.io/gene-expression-on-fhir/

3.7.1 Summary and discussion

For clinical application integration of the result into healthcare systems is as important as the performed analysis itself. Therefore, this paper illustrates the exchange of gene expression profiles using FHIR resources as established standard for sharing clinical information. Although the FHIR definition already includes options to share patients' genomic features, possibilities to represent additional molecular omics data within the standard is currently missing. Adoption of this standard for transcriptomic information allows to demonstrates its feasibility for usage in a clinical setting, thus promoting its application within clinical decision support systems or for patient assessment. This work aims for closing this gap and enabling patient stratification through transcriptomic profiling in multi-center clinical trials across health care institutions.

The work was demonstrated on the gene expression analysis data set that examines a dose-limiting side effect in patients diagnosed with acute myeloid leukemia (AML) undergoing chemotherapy (76). The choice for this data set was dependent on its comparably small size, making it suitable for demonstrating the application in FHIR, in contrast to the large breast cancer dataset. A custom HAPI FHIR server was deployed using docker container to provide an adequate FHIR endpoint.

A collection of FHIR resources were used to capture patient information, derived samples, and their molecular conditions, namely *Patient*, *Specimen*, and *Condition* respectively.The analysis was based on the GRCh38.p13 reference genome, which was included entirely as *MolecularSequence* FHIR resources, which then could be used as references for the single gene expression values. Those were realized as Observation resources with the Observation-genetics extension and link patients and their different samples with the corresponding genes. In total translated the data to 252,684 resources stored on our FHIR server, while for performance issues not all gene from the reference genome (21,055 from 60,617 ensemble entries in total) were encoded in FHIR but only those available in the gene expression data.

To demonstrate the practical application of the FHIR resources a web application was implemented in Angular using the created resources directly form the FHIR server. The different resources were retrieved from the FHIR server, linked and combined into a visual representation of the gene expression across the patient samples. The different patients, their samples, and corresponding

expression profiles can be investigated individually, or as combined heatmap representation for all or selected sets of genes.

3.7.2 Declaration of contribution

Zhibek Abdykalykova developed a first proof-of-concept including *Patient*, *MolecularSequence*, and *Observation* resources in her Bachelor thesis under my supervision. Afterwards I extended the python scripts with additional background and study related information from the data set, and further cross-references to external databases, thereby adding *Specimen* and *Condition* resources. I also added the bash scripts and configuration files for setting, and replaced the initial JQuery based website with an own implementation as single-page web-application written with Angular.

Dominik Müller supported the supervision of the Bachelor thesis and implementation in python related issues, and assisted in the preparation of the manuscript. The manuscript was written by myself with feedback from Dominik Müller, and approval by Prof. Kramer.

4 Summary and Conclusion

The previous sections presented publications introducing a collection of tools, methodologies, and workflows for the integration, analysis, and visualization of biological networks within a biomedical context. The integration of omics and network data from various sources to generate patient-specific subnetworks is the superordinate theme of this thesis, along with a combination of the distinct resources to establish a comprehensive knowledge base documenting the reproducibility and further investigations. In the following, the focus rests on the relation between the single chapters and illustrates the interplay of the presented instruments to foster individualized treatment decisions in a clinical setting (Figure 18).

Chapter 3.1 introduces the *RCX* package, which forms the cornerstone around which the following network data integration and visualization evolves. The basic CX focuses on the exchange of biological networks through the web and therefore follows a dynamic object-oriented structure for encoding its content. This contrasts with the vector and table-centric view on data within R, a shortcoming resolved by the functions for the lossless conversion between both realms provided by this package. Furthermore, the included *RCX* data model also captures the relations between the different components of the network, which might easily be corrupted otherwise. To ensure the correctness of the data model and its internal relations the package provides functions for their validation, either for a network as a whole or at aspect level. This is particularly useful as automatically included validation step during the modification of networks or while creating within the package included functions.

A simple adaption to standard R data types would still be insufficient for its usability with well-established data models for graph and network analysis and visualization in R. Therefore the *RCX* package includes functions for the conversion to and from *iGraph* and *Bioconductor graph*, and as a result integrates with existing workflows. However, in contrast to these network data formats is the visualization included within the *RCX* network and shared along with it. As a consequence is the visual representation of the network congruent across the NDEx platform, Cytoscape, NDExEdit, and the *RCX* package included visualization.

The RCX data model is also used within the in chapter 3.2 presented ndexr package for the seamless data exchange with the NDEx platform. The package methods enable interaction with the platform API of public and private instances. The NDEx platform can be queried and available networks, their single aspects, metadata, or additional network information retrieved from within R. It also allows the upload of own networks and adjustment of their visibility on the platform and provides an option for sharing them with certain users or groups. The *RCX* and the *ndexr* package were implemented independently to separate the data model from the interface dealing with the server communication, and thereby ease maintainability and promote robustness against changes in both.

Networks created and enriched with additional information using the RCX package and uploaded to the NDEx platform, or already there publicly available networks do not necessarily contain a proper visualization. Moreover, included visualizations may not be sufficient to reflect all present information. Even minor changes in the visualization require an additional tool for the adjustments and, more importantly, profound knowledge of the contained data, its coverage of the network, and

the distribution of its values. The in chapter 3.4 introduced web application *NDExEdit* integrates with the NDEx platform and simplifies data exploration and adjustments to the visualization. Its data-centric perspective on networks, together with its online availability and no requirement for installation distinguishes NDExEdit from other common tools like Cytoscape. Furthermore, the linkage to NDEx allows quick sharing of the results with collaborators, as well as the simple re-use of the prepared visual properties within R using the *RCX* and *ndexr* packages.

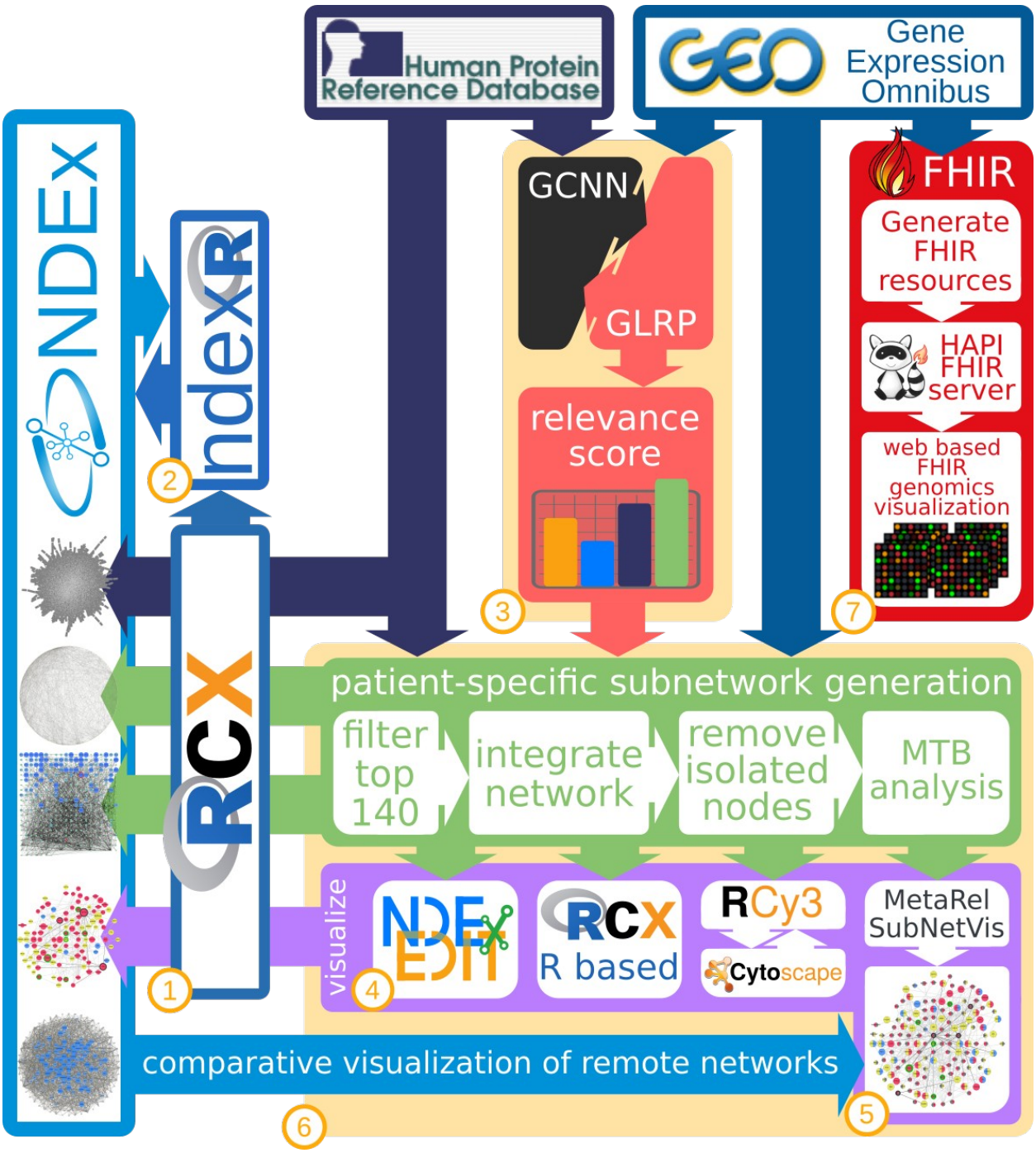


Figure 18: Interplay of the tools presented in this thesis. The tools are marked with the number of the corresponding sub-chapter they have been introduced.

All the previous tools for network data handling, integration, and visualization have firstly been applied for the generation of patient-specific subnetworks in breast cancer patients. Chapter 3.3 demonstrates the application of graph-based deep learning algorithms for the prognosis of patient survival and introduces a novel method for the attribution of the predictions to the individual molecular conditions. The first part was realized by adapting convolutional neural networks to take graphs as an additional input layer to the learning algorithm and allow in this way the usage of biological networks for training and prediction. These graph convolution neural networks not only take gene expression as the base for the prediction but also consider the context of the single genes in the form of the interactions between them. Thereby the paper demonstrates that the performance of the model is comparable to state-of-the-art machine learning algorithms like random forests.

Besides the patients' prognosis for the occurrence of distant metastases, the information about which genes promote these events is more significant for clinical applications and subsequently for individualized treatment decisions. Extension of layer-wise relevance propagation technique to the graph domain allows deriving the contribution of the single genes to a prediction. Application of the GCNN and GLRP approach to the breast cancer data set revealed both subtype- and patient-specific genes. Furthermore, these results could be shown to be in concordance with genes known from the literature to be relevant for tumor progression and as targets in established therapies.

The deducted relevance score was also used to induce patient-specific subnetworks by only considering the most relevant genes to the prediction. Therefore, the web-based tool *MetaRelSubNetVis* was developed to investigate the subnetworks in an interactive manner (Chapter 3.5). Besides exploring the patient-specific subnetworks the *MetaRelSubNetVis* enables comparisons between the two groups of patients with and without metastasis. It allows the setting of node size and color by the gene expression value and level, and relevance score while keeping the node positions consistent among all changes. Together with the adjustment of the threshold for relevant genes it enables a more in-depth investigation of the patient-specific subnetworks and facilitates their interpretability in a biological context on the patient level.

The integration of omics data with biological networks, and concurrent visualizations need to be documented simultaneously to enable reproducibility of the workflows. Chapter 3.6 examines the in the previous chapter performed steps in detail: Starting with the HPRD protein interaction network and the breast cancer gene expression data set, including information about patient metastatic status, and the relevance scores as a result of the GLRP analysis, data integration is performed incrementally and interim stages are captured in a reproducible manner. The visualization of the integrated network data at the various steps is crucial for transparency, quality control, and reproducibility, and is built upon the tools presented in this thesis.

NDEx provides capabilities important for collaboration on intermediate results and sharing of the final outcomes. Therefore, it serves as the central repository and acts as a knowledge base for the performed tasks. Biological networks from public databases used as an initial resource for the analysis are stored and published on NDEx. Subsequent integration and visualization steps are documented as liked snapshots of the enriched networks and are available as a supplement in publications.

Creating network visualizations manually within R is a complex task, especially because of the structure and its dependencies used for storing the visual properties within the network. The desktop software Cytoscape, from which the format is derived, provides with *RCy3* an R package to remotely control the application for the programmatic creation of visualizations. The paper demonstrates the different tools to achieve the same visualization of the final results and highlights the advantages

and drawbacks of the different approaches: a web-based visualization using NDExEdit, controlling Cytoscape with RCy3, manually creation of the visual properties in *RCX,* and preparation of the final network for the utilization in *MetaRelSubNetVis*.

The in chapter 3.7 presented work demonstrates the usage of FHIR resources for the exchange of gene expression profiles within a clinical setting. Integration of the created resources within an Angular web application illustrates their potential application in decision support systems. This work contributes to closing this gap for patient stratification through transcriptomic profiling across health care institutions and within multi-center clinical trials.

The further incorporation of currently missing genomic features into the FHIR standard offers additional opportunities to develop a standard for the aggregation of various molecular genetics data. Currently there is no possibility to encode biological networks, or networks in general, within the existing FHIR standard natively. A possible solution would entail the following: The inclusion of networks as primitive data type, including nodes and edges, the definition of a *FHIR-BiologicalNetwork* resource, and profiling of an extension of the *Observation* resource pointing to this new resource. In general it is feasible and enabled within the FHIR specification to define custom resources and their usage at private servers. Some FHIR server, like the Firely server, can even handle these resources when the appropriate structure definition is provided. However, the usage then is limited to these servers, and the exchange of resources outside ones programming control boundaries cannot be ensured. A wide-spread adoption among working groups and institutions could establish consistency but would cause a tremendous effort in maintenance and coordination. A slight change in the data structure definition would effect incompatibility between servers with different versions and thereby ruin the to the standard eponymous interoperability. Ultimately, only the integration into the FHIR core specification can guarantee consistency and long term adaptation, but this is also a long-lasting process even within fast pace FHIR community.

This thesis illustrates, how the presented tools and techniques can be combined to establish and maintain a knowledge base for the examination of enriched biological networks composted from various sources. Since not only the single resources for molecular interactions and patients' omics data are publicly available but also documented along each performed analysis step, the here presented work significantly contributes to the reproducibility in network biology. The interface to clinical systems using the FHIR standard for sharing patients' omics data within a hospital setting further fosters interoperability between bioinformatics and medical informatics approaches. Moreover, it facilitates the traceability of findings of diagnostic and therapeutic importance by narrowing the gap between both domains. With the here presented work, individualized treatment decisions become to some extent more widely and easily available, and thereby support clinicians in providing improved patient treatment.

References

1. Watson JD, Crick FHC. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. Nature. 1953 Apr;171(4356):737–8.
2. Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, et al. The sequence of the human genome. Science. 2001 Feb 16;291(5507):1304–51.
3. Jalili R. Next Generation Sequencing Methods and Its Impacts on Genomics and Clinical Applications. 2013 [cited 2015 Jun 3]; Available from: http://171.65.20.140/biochem158/Final%20Papers%202013/Jallili.pdf
4. Mardis ER. The impact of next-generation sequencing technology on genetics. Trends in Genetics. 2008 Mar 1;24(3):133–41.
5. Kolodziejczyk AA, Kim JK, Svensson V, Marioni JC, Teichmann SA. The Technology and Biology of Single-Cell RNA Sequencing. Molecular Cell. 2015 May 21;58(4):610–20.
6. Suvà ML, Tirosh I. Single-Cell RNA Sequencing in Cancer: Lessons Learned and Emerging Challenges. Molecular Cell. 2019 Jul 11;75(1):7–12.
7. Voelkerding KV, Dames SA, Durtschi JD. Next-Generation Sequencing: From Basic Research to Diagnostics. Clinical Chemistry. 2009 Apr 1;55(4):641–58.
8. Meldrum C, Doyle MA, Tothill RW. Next-Generation Sequencing for Cancer Diagnostics: a Practical Perspective. Clin Biochem Rev. 2011 Nov;32(4):177–95.
9. Sikkema-Raddatz B, Johansson LF, de Boer EN, Almomani R, Boven LG, van den Berg MP, et al. Targeted Next-Generation Sequencing can Replace Sanger Sequencing in Clinical Diagnostics. Human Mutation. 2013;34(7):1035–42.
10. Breasted JH. The Edwin Smith Surgical Papyrus: Published in Facsimile and Hieroglyphic Transliteration With Translation and Commentary in Two Volumes. Chicago, Ill: University of Chicago Press; 1930.
11. Mukherjee S. The emperor of all maladies: A biography of cancer. New York: Scribner; 2010.
12. Bray F, Laversanne M, Cao B, Varghese C, Mikkelsen B, Weiderpass E, et al. Comparing cancer and cardiovascular disease trends in 20 middle- or high-income countries 2000–19: A pointer to national trajectories towards achieving Sustainable Development goal target 3.4. Cancer Treatment Reviews [Internet]. 2021 Nov 1 [cited 2021 Nov 6];100. Available from: https://www.cancertreatmentreviews.com/article/S0305-7372(21)00138-9/fulltext
13. Massard C, Michiels S, Ferté C, Le Deley MC, Lacroix L, Hollebecque A, et al. High-Throughput Genomics and Clinical Outcome in Hard-to-Treat Advanced Cancers: Results of the MOSCATO 01 Trial. Cancer Discovery. 2017 Jun 1;7(6):586–95.
14. mRNAs, proteins and the emerging principles of gene expression control | Nature Reviews Genetics [Internet]. [cited 2022 Apr 25]. Available from: https://www.nature.com/articles/s41576-020-0258-4
15. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. Molecular biology of the cell. New York: Garland Science; 2002.
16. Aebersold R, Agar JN, Amster IJ, Baker MS, Bertozzi CR, Boja ES, et al. How many human proteoforms are there? Nat Chem Biol. 2018 Mar;14(3):206–14.
17. Isoform-resolved correlation analysis between mRNA abundance regulation and protein level degradation. Molecular Systems Biology. 2020 Mar;16(3):e9170.

18. Stark R, Grzelak M, Hadfield J. RNA sequencing: the teenage years. Nat Rev Genet. 2019 Nov;20(11):631–56.

19. Goodwin S, McPherson JD, McCombie WR. Coming of age: ten years of next-generation sequencing technologies. Nat Rev Genet. 2016 Jun;17(6):333–51.

20. Eroles P, Bosch A, Alejandro Pérez-Fidalgo J, Lluch A. Molecular biology in breast cancer: Intrinsic subtypes and signaling pathways. Cancer Treatment Reviews. 2012 Oct 1;38(6):698–707.

21. Calabrò M, Rinaldi C, Santoro G, Crisafulli C. The biological pathways of Alzheimer disease: a review. AIMS Neurosci. 2020 Dec 16;8(1):86–132.

22. Alm E, Arkin AP. Biological networks. Current Opinion in Structural Biology. 2003 Apr 1;13(2):193–202.

23. Huser V, Sastry C, Breymaier M, Idriss A, Cimino JJ. Standardizing data exchange for clinical research protocols and case report forms: An assessment of the suitability of the Clinical Data Interchange Standards Consortium (CDISC) Operational Data Model (ODM). Journal of Biomedical Informatics. 2015 Oct 1;57:88–99.

24. Berners-Lee T. The WorldWideWeb [Internet]. [cited 2022 Apr 19]. Available from: https://www.w3.org/People/Berners-Lee/1991/08/art-6484.txt

25. Flanagan D. JavaScript: the definitive guide [Internet]. Beijing; Farnham: O'Reilly; 2011 [cited 2022 Apr 19]. Available from: http://public.eblib.com/choice/publicfullrecord.aspx?p=686420_0

26. Vaadin - An open platform for building web apps in Java [Internet]. [cited 2022 Apr 19]. Available from: https://vaadin.com

27. Shiny [Internet]. [cited 2022 Apr 19]. Available from: https://shiny.rstudio.com/

28. Node.js. Node.js [Internet]. Node.js. [cited 2022 Apr 19]. Available from: https://nodejs.org/en/

29. JSON Schema [Internet]. JSON Schema. [cited 2022 Apr 19]. Available from: https://json-schema.org/

30. Savkin V. Angular: Why TypeScript? [Internet]. Medium. 2019 [cited 2022 Apr 19]. Available from: https://vsavkin.com/writing-angular-2-in-typescript-1fa77c78d8e8

31. Fielding RT, Taylor RN. Principled design of the modern Web architecture. ACM Trans Internet Technol. 2002 May 1;2(2):115–50.

32. Giessler P, Gebhart M, Sarancin D, Steinegger R, Abeck S. Best Practices for the Design of RESTful Web Services. 2015;7.

33. Richardson L, Amundsen M, Ruby S. RESTful Web APIs: Services for a Changing World. O'Reilly Media; 2013. 406 p.

34. Jain N, Mangal P, Mehta D. AngularJS: A Modern MVC Framework in JavaScript. Journal of Global Research in Computer Science [Internet]. 2014 [cited 2021 Nov 3]; Available from: https://www.semanticscholar.org/paper/AngularJS%3A-A-Modern-MVC-Framework-in-JavaScript-Jain-Mangal/92f02a5409407a3732ebd747822f860ea91654fd

35. RxJS [Internet]. [cited 2022 Apr 21]. Available from: https://rxjs.dev/

36. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, et al. Human Protein Reference Database—2009 update. Nucleic Acids Research. 2009 Jan 1;37(suppl_1):D767–72.

37. Mishra GR, Suresh M, Kumaran K, Kannabiran N, Suresh S, Bala P, et al. Human protein reference database—2006 update. Nucleic Acids Research. 2006 Jan 1;34(suppl_1):D411–4.

38. Peri S, Navarro JD, Amanchy R, Kristiansen TZ, Jonnalagadda CK, Surendranath V, et al. Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Res. 2003 Oct;13(10):2363–71.

39. Kerrien S, Orchard S, Montecchi-Palazzi L, Aranda B, Quinn AF, Vinod N, et al. Broadening the horizon – level 2.5 of the HUPO-PSI format for molecular interactions. BMC Biol. 2007 Oct 9;5(1):44.

40. Pratt D, Chen J, Welker D, Rivas R, Pillich R, Rynkov V, et al. NDEx, the Network Data Exchange. cels. 2015 Oct 28;1(4):302–5.

41. Pillich RT, Chen J, Rynkov V, Welker D, Pratt D. NDEx: A Community Resource for Sharing and Publishing of Biological Networks. Methods Mol Biol. 2017;1558:271–301.

42. Pratt D, Chen J, Pillich R, Rynkov V, Gary A, Demchak B, et al. NDEx 2.0: A Clearinghouse for Research on Cancer Pathways. Cancer Res. 2017 Nov 1;77(21):e58–61.

43. Schaefer CF, Anthony K, Krupa S, Buchoff J, Day M, Hannay T, et al. PID: the Pathway Interaction Database. Nucleic Acids Research. 2009 Jan 1;37(suppl_1):D674–9.

44. Kim M, Park J, Bouhaddou M, Kim K, Rojc A, Modak M, et al. A protein interaction landscape of breast cancer. Science. 2021 Oct;374(6563):eabf3066.

45. 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 Nov;13(11):2498–504.

46. Franz M, Lopes CT, Huck G, Dong Y, Sumer O, Bader GD. Cytoscape.js: a graph theory library for visualisation and analysis. Bioinformatics. 2016 Jan 15;32(2):309–11.

47. ndexbio/ndex-javascript [Internet]. NDEx; 2017 [cited 2022 Apr 22]. Available from: https://github.com/ndexbio/ndex-javascript

48. R Development Core Team. R: A Language and Environment for Statistical Computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2008 [cited 2017 May 9]. Available from: http://www.R-project.org

49. Hornik K. The Comprehensive R Archive Network. WIREs Computational Statistics. 2012;4(4):394–8.

50. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, et al. Bioconductor: open software development for computational biology and bioinformatics. Genome Biol. 2004 Sep 15;5(10):R80.

51. Csardi G, Nepusz T. The igraph software package for complex network research. InterJournal. 2006;Complex Systems:1695.

52. Gustavsen JA, Pai S, Isserlin R, Demchak B, Pico AR. RCy3: Network biology using Cytoscape from within R. F1000Res. 2019 Dec 4;8:1774.

53. Health Level Seven International - Homepage | HL7 International [Internet]. [cited 2021 Sep 25]. Available from: https://www.hl7.org

54. FHIR Foundation, HL7.org. Welcome to FHIR v4.0.1 [Internet]. [cited 2021 May 31]. Available from: https://www.hl7.org/fhir/

55. Alterovitz G, Heale B, Jones J, Kreda D, Lin F, Liu L, et al. FHIR Genomics: enabling standardization for precision medicine use cases. npj Genom Med. 2020 Mar 18;5(1):1–4.

56. Defferrard M, Bresson X, Vandergheynst P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In: Advances in Neural Information Processing Systems [Internet]. Curran Associates, Inc.; 2016 [cited 2022 Apr 24]. Available from: https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html

57. Chen L, Liu H, Kocher JPA, Li H, Chen J. glmgraph: an R package for variable selection and predictive modeling of structured genomic data. Bioinformatics. 2015 Dec 15;31(24):3991–3.

58. Perera-Bel J, Hutter B, Heining C, Bleckmann A, Fröhlich M, Fröhling S, et al. From somatic variants towards precision oncology: Evidence-driven reporting of treatment options in molecular tumor boards. Genome Medicine. 2018 Mar 15;10(1):18.

59. Krull M. TRANSPATH(R): an integrated database on signal transduction and a tool for array analysis. Nucleic Acids Research. 2003 Jan 1;31(1):97–100.

60. Fisher RA. On the Interpretation of $\chi 2$ from Contingency Tables, and the Calculation of P. Journal of the Royal Statistical Society. 1922;85(1):87–94.

61. Koschmann J, Bhar A, Stegmaier P, Kel AE, Wingender E. “Upstream Analysis”: An Integrated Promoter-Pathway Analysis Approach to Causal Interpretation of Microarray Data. Microarrays. 2015 Jun;4(2):270–86.

62. Lin CY, Beattie A, Baradaran B, Dray E, Duijf PHG. Contradictory mRNA and protein misexpression of EEF1A1 in ductal breast carcinoma due to cell cycle regulation and cellular stress. Sci Rep. 2018 Sep 17;8(1):13904.

63. Mendez MG, Kojima SI, Goldman RD. Vimentin induces changes in cell shape, motility, and adhesion during the epithelial to mesenchymal transition. FASEB J. 2010 Jun;24(6):1838–51.

64. Sharma P, Alsharif S, Fallatah A, Chung BM. Intermediate Filaments as Effectors of Cancer Development and Metastasis: A Focus on Keratins, Vimentin, and Nestin. Cells. 2019 May;8(5):497.

65. Perou CM, Sørlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, et al. Molecular portraits of human breast tumours. Nature. 2000 Aug;406(6797):747–52.

66. Qian XL, Pan YH, Huang QY, Shi YB, Huang QY, Hu ZZ, et al. Caveolin-1: a multifaceted driver of breast cancer progression and its application in clinical treatment . OTT. 2019 Feb 27;12:1539–52.

67. Aceto N, Sausgruber N, Brinkhaus H, Gaidatzis D, Martiny-Baron G, Mazzarol G, et al. Tyrosine phosphatase SHP2 promotes breast cancer progression and maintains tumor-initiating cells via activation of key transcription factors and a positive feedback signaling loop. Nat Med. 2012 Apr;18(4):529–37.

68. Chekhun VF, Lukyanova NY, Burlaka AP, Bezdenezhnykh NA, Shpyleva SI, Tryndyak VP, et al. Iron metabolism disturbances in the MCF-7 human breast cancer cells with acquired resistance to doxorubicin and cisplatin. International Journal of Oncology. 2013 Nov 1;43(5):1481–6.

69. Bailey CK, Mittal MK, Misra S, Chaudhuri G. High Motility of Triple-negative Breast Cancer Cells Is Due to Repression of Plakoglobin Gene by Metastasis Modulator Protein SLUG *. Journal of Biological Chemistry. 2012 Jun 1;287(23):19472–86.

70. Fan B, Shi S, Shen X, Yang X, Liu N, Wu G, et al. Effect of HMGN2 on proliferation and apoptosis of MCF-7 breast cancer cells. Oncology Letters. 2019 Jan 1;17(1):1160–6.

71. Shi H, Li H, Yuan R, Guan W, Zhang X, Zhang S, et al. PCBP1 depletion promotes tumorigenesis through attenuation of p27Kip1 mRNA stability and translation. Journal of Experimental & Clinical Cancer Research. 2018 Aug 7;37(1):187.

72. Liu Y, Liu T, Sun Q, Niu M, Jiang Y, Pang D. Downregulation of Ras GTPase-activating protein 1 is associated with poor survival of breast invasive ductal carcinoma patients. Oncology Reports. 2015 Jan 1;33(1):119–24.

73. Saha SK, Kim K, Yang GM, Choi HY, Cho SG. Cytokeratin 19 (KRT19) has a Role in the Reprogramming of Cancer Stem Cell-Like Cells to Less Aggressive and More Drug-Sensitive Cells. International Journal of Molecular Sciences. 2018 May;19(5):1423.

74. Zhou X, Hao Q, Liao J ming, Liao P, Lu H. Ribosomal Protein S14 Negatively Regulates c-Myc Activity *. Journal of Biological Chemistry. 2013 Jul 26;288(30):21793–801.

75. Mathe A, Wong-Brown M, Morten B, Forbes JF, Braye SG, Avery-Kiejda KA, et al. Novel genes associated with lymph node metastasis in triple negative breast cancer. Sci Rep. 2015 Nov 5;5(1):15832.

76. Mougeot JLC, Bahrani-Mougeot FK, Lockhart PB, Brennan MT. Microarray analyses of oral punch biopsies from acute myeloid leukemia (AML) patients treated with chemotherapy. Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology. 2011 Oct 1;112(4):446–52.

77. cytoscape/cytoscape-tutorials [Internet]. Cytoscape Consortium; 2017 [cited 2017 Nov 30]. Available from: https://github.com/cytoscape/cytoscape-tutorials

Appendix

Appendix A - RCX

Publication

RCX – an R package adapting the Cytoscape Exchange format for biological networks

Auer F, Kramer F

Bioinformatics Advances, Volume 2, Issue 1, 2022, vbac020
doi: https://doi.org/10.1093/bioadv/vbac020

Bioinformatics Advances, 2022, 1–2
https://doi.org/10.1093/bioadv/vbac020
Advance Access Publication Date: 31 March 2022
Application Note

Databases and ontologies

RCX—an R package adapting the Cytoscape Exchange format for biological networks

Florian Auer * and Frank Kramer

Department of IT-Infrastructure for Translational Medical Research, Faculty of Applied Computer Science, University of Augsburg, Augsburg 86159, Germany

*To whom correspondence should be addressed.

Associate Editor: Marieke Kuijjer

Received on November 8, 2021; revised on March 18, 2022; editorial decision on March 23, 2022

Abstract

Motivation: The Cytoscape Exchange (CX) format is a JSON-based data structure designed for the transmission of biological networks using standard web technologies. It was developed by the network data exchange, which itself serves as online commons to share and collaborate on biological networks. Furthermore, the Cytoscape software for the analysis and visualization of biological networks contributes structure elements to capture the visual layout within the CX format. However, there is a fundamental difference between data handling in web standards and R. A manual conversion requires detailed knowledge of the CX format to reproduce and work with the networks.

Results: Here, we present a software package to create, handle, validate, visualize and convert networks in CX format to standard data types and objects within R. Networks in this format can serve as a source for biological knowledge and also capture the results of the analysis of those while preserving the visual layout across all platforms. The RCX package connects the R environment for statistical computing with outside platforms for storage and collaboration, as well as further analysis and visualization of biological networks.

Availability: RCX is a free and open-source R package, available on Bioconductor from release 3.15 (https://bioconductor.org/packages/RCX) and via GitHub (https://github.com/frankkramer-lab/RCX).

Contact: florian.auer@informatik.uni-augsburg.de

Supplementary information: Supplementary data are available at *Bioinformatics Advances* online.

1 Introduction

Biological networks are a common and widely used resource to capture associations between any types of biological entities such as genes, transcripts, proteins, metabolites, ligands, diseases or drugs. Furthermore, the data formats used for encoding the network information differ heavily depending on the contained data and their intended use.

A variety of public databases provide their biological knowledge in domain-specific exchange formats. In subsequent analyses, those networks are further enriched with heterogeneous data and therefore require a more flexible format for capturing their content. Additionally, the layout and visualization of networks are often not considered as part of the network and omitted.

The Cytoscape exchange (CX) format covers the above shortcomings by following an aspect-based design: the network is split into independent modules (aspects) with specific schemes for the information they contain. For example, the *edges* aspect comprises the interactions between the nodes defined in the *nodes* aspect, and the *cartesianLayout* aspect provides the coordinates to position the nodes in space. The CX format was developed by the Network Data Exchange (NDEx), an online commons for biological networks (Pratt *et al.*, 2015). The schemes for aspects responsible for storing visual attributes are derived from Cytoscape (Shannon *et al.*, 2003), one of the most popular open-source software tools for the analysis and visualization of biomedical networks. Both Cytoscape and NDEx use the CX format for the exchange of the networks between their platforms with consistent visualizations.

Users of the statistical programming language R (R Development Core Team, 2008) can use existing packages like rBiopaxParser (Kramer *et al.*, 2013) to retrieve biological knowledge from public databases and conduct further analyses. The ndexr package (Auer *et al.*, 2018) interfaces with the NDEx platform to store subsequent results. However, the included data model is a simple conversion of the JSON structure that conflicts with the table-oriented approach in R. Consequently, constructing valid networks requires advance knowledge and hinders the adoption of the software by a large user base. The RCX package implements the aspect-oriented design while allowing the user to focus on the networks instead of the underlying data structure.

2 Features

The RCX package provides custom functions for the creation and modification of aspects and networks in RCX format. Additional functions are provided to validate data types, aspects properties and references between the aspects, even after manual editing. The CX format

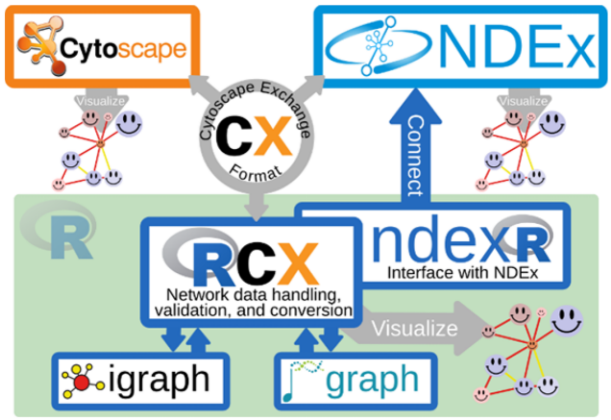


Fig. 1. The RCX package connects the CX transmission format with established analysis libraries in R. The CX format is shared between RCX, Cytoscape and NDEx, while the consistent visual representation is preserved

also requires a meta-data aspect that provides an overview of the included aspects (e.g. number of elements within the aspects). Within RCX objects, this meta-data is created and updated automatically.

The RCX package not only provides accessibility of networks in CX format, but it also provides conversion to and from objects of iGraph (Csardi and Nepusz, 2006) and Bioconductor graph (Gentleman *et al.*, 2021), both widely used libraries for graph manipulation and network analysis (Fig. 1).

The R-based visualization of the networks is congruent with its representation in both the NDEx platform and Cytoscape. It also can be exported as an HTML file for further use. Since the visual representation is saved as an aspect within the network, it can easily be reused to layout additional networks in the same style without modification.

A key feature of the aspect-oriented design of the CX format is to allow the definition of custom aspects. Therefore, the RCX package was designed with a focus on extensibility with additional functions for the creation, modification, conversion and validation of custom aspects.

Detailed documentation and examples can be found in the package manual and vignettes. Supplementary Materials contain code examples for working with RCX networks and their creation.

3 Implementation

The RCX package builds upon several R packages for data processing and graph representation. The CX networks are read and written with the readr package. The obtained JSON is transformed and further processed using the jsonlite (Ooms, 2014) and tidyr packages (Wickham, 2011).

The visualization was realized with the JavaScript library cytoscape.js (Franz *et al.*, 2016) and a custom script to map the visual properties between the in CX used Cytoscape properties to cytoscape.js compatible layout definitions.

4 Conclusion

The RCX package is a freely available R software tool that enables the lossless conversion between the object-oriented JSON format of the CX data structure and the table-like paradigm of data in R. The data model was designed to enhance usability and enrich functionality by a better adjustment to fundamental R data structures and adding high-level functions for data manipulation.

Integrated conversion to igraph and Bioconductor compatible graph objects fosters the accessibility to advanced network analysis tools. Furthermore, extensibility was increased by facilitating the creation of custom aspects that cover specialized extensions to the CX data model.

By implementing this software, we ease the task of handling network data available via NDEx within the R Framework for Statistical Computing. Enriched networks as results of investigations and their visualizations can be easily created and translated to the CX format that connects analysis, visualization and collaboration.

Funding

This work is a part of the Multipath project and was supported by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) [FKZ01ZX1508].

Conflict of Interest: none declared.

Availability of data: There are no new data associated with this article.

References

Auer,F. *et al.* (2018) ndexr—an R package to interface with the network data exchange. *Bioinformatics*, **34**, 716–717.

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, complex systems*, *1695*, 1–9.

Franz,M. *et al.* (2016) Cytoscape.Js: a graph theory library for visualisation and analysis. *Bioinformatics*, **32**, 309–311.

Gentleman,R. *et al.* (2021). *Graph: A Package to Handle Graph Data Structures*. R Package Version 1.70.0.

Kramer,F. *et al.* (2013). RBiopaxParser – an R package to parse, modify and visualize BioPAX data. *Bioinformatics*, **29**, 520–522.

Ooms,J. (2014). The Jsonlite package: a practical and consistent mapping between JSON data and R objects. ArXiv:1403.2805 [Cs, Stat], March. http://arxiv.org/abs/1403.2805.

Pratt,D. *et al.* (2015) NDEx, the network data exchange. *Cell Syst.*, **1**, 302–305.

R Development Core Team. (2008). *R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing*. http://www.R-project.org.

Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.

Wickham,H. (2011) The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software, **40**, 1–29. https://doi.org/10.18637/jss.v040.i01.

Appendix A – RCX – Publication A - RCX

Supplement

RCX Cheat Sheet

CHEAT SHEET

createRCX(

nodes,
createNodes(id=NULL, name=NULL, represents=NULL)

edges,
createEdges(id=NULL, source, target, interaction=NULL)

nodeAttributes,
createNodeAttributes(propertyOf, name, value, dataType=NULL, isList=NULL, subnetworkId=NULL)

edgeAttributes,
createEdgeAttributes(propertyOf, name, value, dataType=NULL, isList=NULL, subnetworkId=NULL)

networkAttributes,
createNetworkAttributes(name, value, dataType=NULL, isList=NULL, subnetworkId=NULL)

cartesianLayout,
createCartesianLayout(node, x, y, z=NULL, view=NULL)

cySubNetworks,
createCySubNetworks(id, nodes=NULL, edges=NULL)

cyGroups,
createCyGroups(id=NULL, name, nodes=NULL, externalEdges=NULL, internalEdges=NULL, collapsed=NULL)

cyHiddenAttributes,
createCyHiddenAttributes(name, value, dataType=NULL, isList=NULL, subnetworkId=NULL)

cyNetworkRelations,
createCyNetworkRelations(child, parent=NULL, name=NULL, isView=FALSE)

cyTableColumn,
createCyTableColumn(appliesTo, name, dataType=NULL, isList=NULL, subnetworkId=NULL)

cyVisualProperties,
createCyVisualProperties(network=NULL, nodes=NULL, edges=NULL, defaultNodes=NULL, defaultEdges=NULL)

checkReferences=TRUE
)

readCX(file, verbose=FALSE, aspectClasses=NULL)

updateMetaData(
x,
version = NULL,
consistencyGroup = NULL,
properties=NULL,
aspectClasses=NULL
)

createCyVisualProperty(properties=NULL, dependencies=NULL, mappings=NULL, appliesTo=NULL, view=NULL)

createCyVisualPropertyMappings(
type,
definition,
name=NULL
)

createCyVisualPropertyDependencies(
value,
name=NULL
)

createCyVisualPropertyProperties(
value,
name=NULL
)

validate(x, verbose=TRUE)

visualize(
x,
layout=NULL,
openExternal=FALSE
)

writeCX(
x,
file,
verbose=FALSE,
pretty=FALSE
)

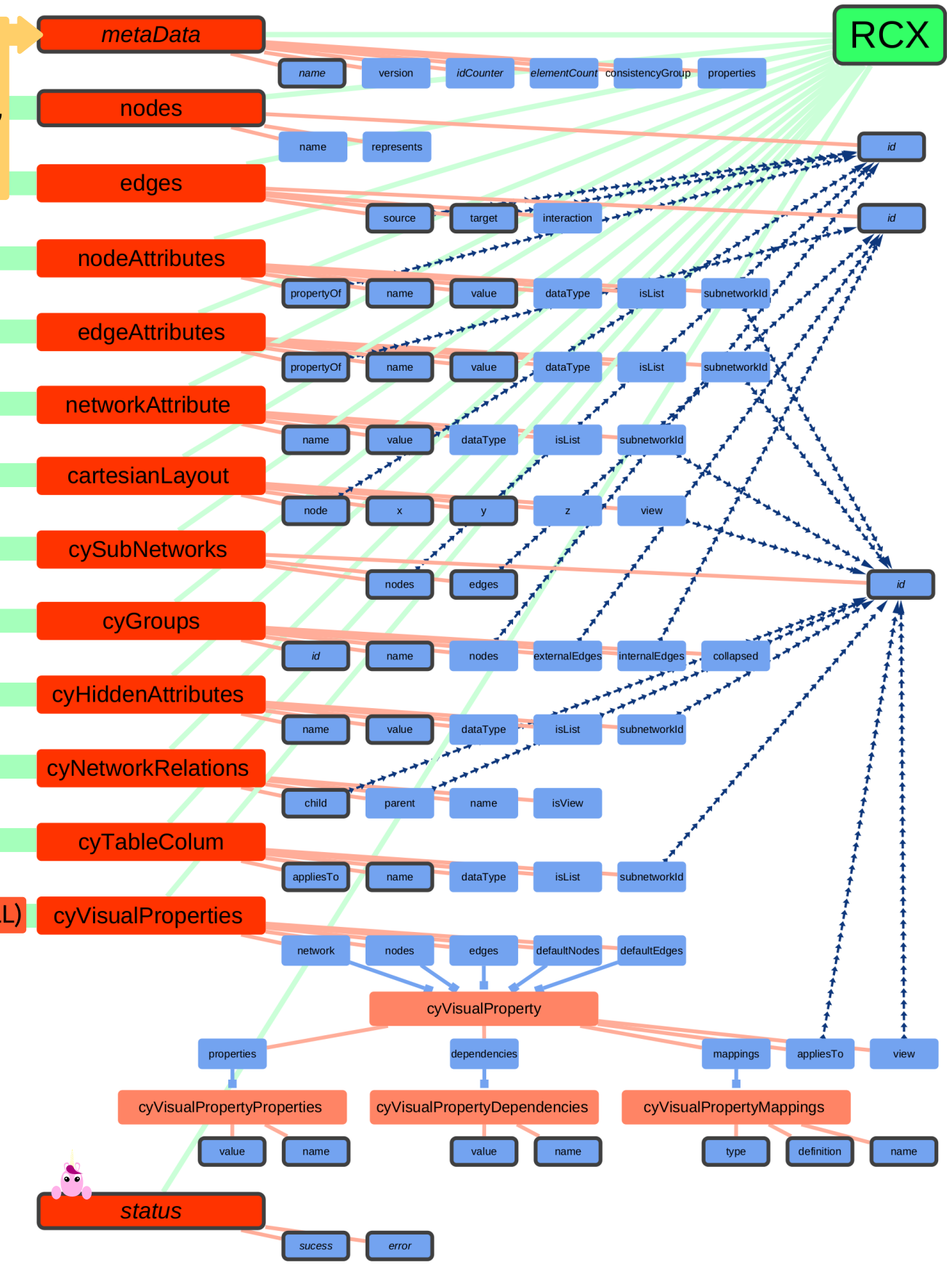
toGraphNEL(rcx, directed=FALSE)

toIgraph(rcx, directed=FALSE)

graphNEL

igraph

DATA MODEL STRUCTURE



Nodes:

- RCX object
- aspect
- sub-aspect
- property
- mandatory element
- *italic* automatically generated

Edges:

- aspect of an object
- property of an aspect
- property contains sub-aspect
- references

Appendix B - NDExR

Publication

ndexr—an R package to interface with the network data exchange

Auer F, Hammoud Z, Ishkin A, Pratt D, Ideker T, Kramer F

Bioinformatics, Volume 34, Issue 4, 15 February 2018, Pages 716–717;
doi: https://doi.org/10.1093/bioinformatics/btx683

Bioinformatics, 34(4), 2018, 716–717
doi: 10.1093/bioinformatics/btx683
Advance Access Publication Date: 26 October 2017
Applications Note

Databases and ontologies

ndexr—an R package to interface with the network data exchange

Florian Auer[1,*], Zaynab Hammoud[1], Alexandr Ishkin[2], Dexter Pratt[3], Trey Ideker[3,4] and Frank Kramer[1]

[1]Department of Medical Statistics, University Medical Center Göttingen, Göttingen 37099, Germany, [2]Discovery Science, Clarivate Analytics, Boston, MA 02210, USA, [3]Department of Medicine, University of California San Diego, La Jolla, CA 92093, USA and [4]Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92093, USA

*To whom correspondence should be addressed.

Associate Editor: Janet Kelso

Received on May 30, 2017; revised on October 5, 2017; editorial decision on October 23, 2017; accepted on October 25, 2017

Abstract

Motivation: Seamless exchange of biological network data enables bioinformatic algorithms to integrate networks as prior knowledge input as well as to document resulting network output. However, the interoperability between pathway databases and various methods and platforms for analysis is currently lacking. The Network Data Exchange (NDEx) is an open-source data commons that facilitates the user-centered sharing and publication of networks of many types and formats.
Results: Here, we present a software package that allows users to programmatically connect to and interface with NDEx servers from within R. The network repository can be searched and networks can be retrieved and converted into igraph-compatible objects. These networks can be modified and extended within R and uploaded back to the NDEx servers.
Availability and implementation: ndexr is a free and open-source R package, available via GitHub (https://github.com/frankkramer-lab/ndexr) and Bioconductor (http://bioconductor.org/packages/ndexr/).
Contact: florian.auer@med.uni-goettingen.de
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Networks are a convenient representation of complex relations and interactions and are commonly used in a wide range of fields in biology. The information represented in networks can range from knowledge on molecular interactions and cellular pathways (Hucka *et al.*, 2003; Kohn, 1999) to the results of bioinformatic methods for network reconstruction (Margolin *et al.*, 2006).

Public databases Reactome (Fabregat *et al.*, 2016) provide access to the rapidly increasing body of biological pathway knowledge and are a well-established source for hypothesis generation and testing. Existing packages, like rBiopaxParser (Kramer *et al.*, 2013), already enable users of the statistical programming language R (R Development Core Team, 2008) to work with biological pathway data.

Researchers, however, still face challenges in dealing with network complexity, diverse data formats, the integration of network analysis tools and methods to share and collaborate on pathway data.

2 Network data exchange

The network data exchange (NDEx) is an open-source software framework to manipulate, store and exchange networks of various types and formats (Pratt *et al.*, 2015).

The NDEx can be used to upload, share and distribute network data, facilitating the creation and curation of networks by users and communities. It can serve as both a source for networks consumed by applications and a destination for the networks that they produce.

The server can be accessed by end users via a web interface and by programs using a relational state transfer application programming interface (REST API) (Fielding and Taylor, 2002).

3 Features

This package provides an interface to NDEx installations from within R and enables a seamless transition from data acquisition to statistical analysis.

Using the NDEx REST API, this package provides an interface to the public NDEx server, as well as private installations, enabling programs to upload, download or modify biological networks. The package also provides classes to implement the cytoscape cyberinfrastructure (CX) format, a flexible, modular and extensible data structure for the transmission of networks. Furthermore, it provides conversion to objects of the *iGraph* package (Csardi and Nepusz, 2006), a widely used library for graph manipulation and network analysis.

A typical workflow illustrating the most important features of this package is described in Figure 1 and might include following steps:

Browse, search and query NDEx to find a network of interest, either as an authenticated user or as an anonymous visitor.

Download a network into R and convert it to built-in data structures resembling the CX structure or to an iGraph object.

1. Perform some network analysis (3a) or apply a typical bioinformatics workflow by integrating additional data and subsequently selecting a subnetwork (3b).
2. Upload the newly created network to the NDEx server.
3. Share the preliminary network only with certain people (5a) or groups of persons (5b).

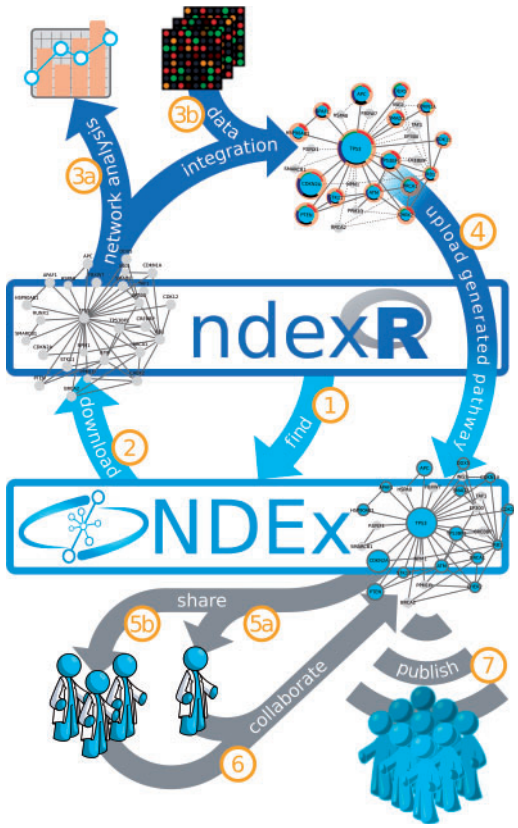


Fig. 1. Workflow using the NDEx server and the ndexr package. The co-operation of both software tools enables a faster and more efficient way to acquire, alter, share and publish networks

4. Collaborate with other people or groups to amend and complete the network.

Reveal the network to the public and/or provide it as supplement along a publication.

Detailed documentation and examples can be found in the package vignettes and manual. The Supplementary Materials contain code examples for common procedures to interact with the public NDEx server.

4 Implementation

The *ndexr* package is based on several R packages for data processing, internet communication and graph representation. Network connections to the NDEx server APIs are handled using the *httr* package. Network data is de- and encoded, as well as transformed into the different interchangeable data structures using the *jsonlite* package and the *plyr* and *tidyr* packages (Wickham, 2011).

This package also provides classes specifically tailored to cope with NDEx and analysis: The package supports NDEx versions 1.3 and 2, and new features and API specifications will be included in regular package updates. Additionally, it is possible to switch and modify the API configuration manually.

5 Conclusion

The ndexr package is a freely available R software tool which enables users to connect to and interact with NDEx servers. Package methods enable networks to be found via queries, retrieved and be converted into igraph compatible graph objects. These graphs can be used and modified within R and uploaded to the NDEx servers. By implementing this software, we ease the task of retrieving and using network data available via NDEx within the R Framework for Statistical Computing.

Funding

This work was supported by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) grants FKZ01ZX1508 and FKZ031L0024A.

Conflict of Interest: none declared.

References

Csardi,G. and Nepusz,T. (2006) The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.

Fabregat,A. *et al.* (2016) The reactome pathway knowledgebase. *Nucleic Acids Res.*, **44**, D481–D487.

Fielding,R.T. and Taylor,R.N. (2002) Principled design of the modern web architecture. *ACM Trans. Internet Technol.*, **2**, 115–150.

Hucka,M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Kohn,K.W. (1999) Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Mol. Biol. Cell*, **10**, 2703–2734.

Kramer,F. *et al.* (2013) rBiopaxParser—an R package to parse, modify and visualize BioPAX data. *Bioinformatics*, **29**, 520–522.

Margolin,A.A. *et al.* (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7**, S7.

Pratt,D. *et al.* (2015) NDEx, the network data exchange. *Cell Syst.*, **1**, 302–305.

R Development Core Team. (2008) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Wickham,H. (2011) The split-apply-combine strategy for data analysis. *J. Stat. Softw.*, **40**, 1–29.

Appendix B - NDExR

Vignette

NDExR Cheat Sheet

ndexR Cheat Sheet

Installation and Connection

```
# via Bioconductor
source("https://bioconductor.org/biocLite.R")
biocLite("ndexr")
# load library
library(ndexr)
# connect anonymously
ndexcon <- ndex_connect()
```

```
# via GitHub using devtools R package
require(devtools)
install_github("frankkramer-lab/ndexr")
```

```
# ...or using credentials
ndexcon <- ndex_connect("username", "password")
```

1 Find networks

```
# browse & search for a specific
networks <- ndex_find_networks(ndexcon)
networks <- ndex_find_networks(ndexcon, "EGFR")
networks <- ndex_find_networks(ndexcon, accountName="username")
# get the UUID of the first search result
networkId <- networks[1,"externalId"]
```

2 Download networks

```
# ...complete networks
rcx <- ndex_get_network(ndexcon, networkId)
# ...summary, meta-data, aspects & provenance
networkSummary <- ndex_network_get_summary(ndexcon, networkId)
networkMetadata <- ndex_network_get_metadata(ndexcon, networkId)
aspect <- ndex_network_get_aspect(ndexcon, networkId, "nodeAttributes")
aspectMetadata <- ndex_network_aspect_get_metadata(ndexcon, networkId, "nodeAttributes")
provenance = ndex_network_get_provenance(ndexcon, networkId)
```

3 Work with the networks

```
# remove NDEx artefacts from network
rcx <- rcx_asNewNetwork(rcx)
# ...do some fancy stuff with the network, then update the meta-data
rcx <- rcx_updateMetaData(rcx)
# work with NGraph instead
ngraph = ngraph_fromRCX(rcx)
rcx = ngraph_toRCX(ngraph)
# or start from scratch
rcx = rcx_new( data.frame( '@id'=c(1,2,3),
                           n=c("Some Name","And another name",NA),
                           r=c("HGNC:Symbol",NA,"UniProt:C3P0"),
                           check.names=FALSE))
# print the RCX object
print(rcx)
```

4 Upload a network to the NDEx server

```
# upload network as a new network
networkId <- ndex_create_network(ndexcon, rcx)
# update the network
ndex_update_network(ndexcon, rcx)
ndex_update_network(ndexcon, rcx, networkId)
```

5a Share networks with people

```
# find people
users <- ndex_find_users(ndexcon, "user")
userId <- users$externalId
user <- ndex_find_user_byName(ndexcon, "username")
user <- ndex_find_user_byId(ndexcon, userId)
# manage accounts
user <- ndex_create_user( ndexcon, userName="UserName", password="SecretPassword",
                          emailAddress="A@bc.de", isIndividual=TRUE,
                          displayName="J.D.", firstName="John", lastName="Doe",
                          website="www.abc.de", description="Nothing to see here..")
ndex_verify_user(ndexcon, userId, "4ct1v4t10n-C0d3")
ndex_update_user(ndexcon, userId, firstName = "Max", lastName = "Power")
ndex_user_change_password(ndexcon, userId, "SuperSaveNewPassword")
ndex_user_forgot_password(ndexcon, userId)
ndex_user_mail_password(ndexcon, userId)
ndex_delete_user(ndexcon, userId)
networkPermissions <- ndex_user_list_permissions(ndexcon, userId)
networkPermission <- ndex_user_show_permission( ndexcon, userId, networkId,
                                                directonly=TRUE)
showcase = ndex_user_get_showcase(ndexcon, userId)
networkSummary <- ndex_user_get_networksummary(ndexcon, userId)
```

5b Share networks with groups

```
# find groups
groups <- ndex_find_groups(ndexcon, "Ideker Lab")
groupId <- groups$externalId
group <- ndex_get_group(ndexcon, groupId)
# manage groups
group <- ndex_create_group( ndexcon, "SomeGroupName", image="http://bit.ly/1M3NoQZ",
                            website="www.gidf.com", description="Special group..")
ndex_update_group(ndexcon, groupId, description="A really nice group!")
ndex_delete_group(ndexcon,groupId)
ndex_group_set_membership(ndexcon, groupId, userId, type="MEMBER")
users <- ndex_group_list_users (ndexcon, groupId, type="ADMIN", start=0, size=10)
networkPermissions <- ndex_group_list_networks(ndexcon, groupId, permission="READ")
networkPermission <- ndex_group_network_get_permission(ndexcon, groupId, networkId)
groups <- ndex_user_list_groups(ndexcon, userId)
group <- ndex_user_show_group(ndexcon, userId, groupId)
```

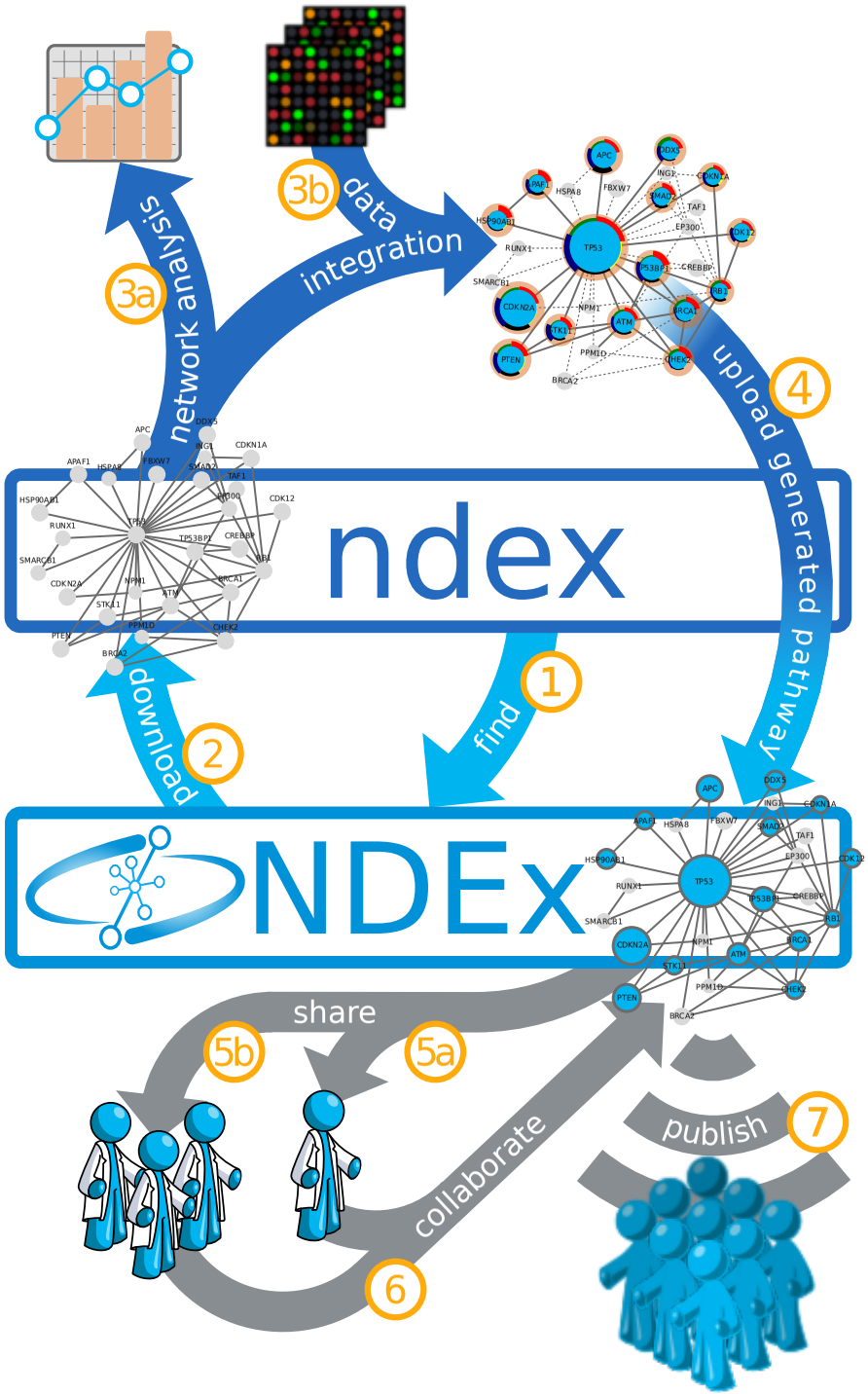
ndexR Cheat Sheet

6 Collaborate on networks

```
# control user permissions to a network
permissions = ndex_network_get_permission(ndexcon, networkId, "user")
permissions = ndex_network_get_permission(ndexcon, networkId, "user", "READ")
ndex_network_update_permission(ndexcon, networkId, user=userId, "WRITE")
ndex_network_delete_permission(ndexcon, networkId, user=userId)
# control group permissions to a network
permissions = ndex_network_get_permission(ndexcon, networkId, "group")
ndex_network_update_permission(ndexcon, networkId, group=groupId, "READ")
ndex_network_delete_permission(ndexcon, networkId, group=groupId)
# change network properties
ndex_network_update_profile(ndexcon, networkId, name="Some fancy name for the network")
ndex_network_update_profile(ndexcon, networkId, description="Description of the network")
ndex_network_update_profile(ndexcon, networkId, version="1.2.3.4")
# realize, you did bad things, so better delete the network
ndex_delete_network(ndexcon, networkId)
```

7 Publish networks

```
# make the network visible to everyone
ndex_network_set_systemProperties(ndexcon, networkId, readOnly=TRUE)
ndex_network_set_systemProperties(ndexcon, networkId, showcase=TRUE)
ndex_network_set_systemProperties(ndexcon, networkId, visibility="PUBLIC")
# publish the link to the network
http://www.ndexbio.org/#/newNetwork/9ed0cd55-9ac0-11e4-9499-000c29202374
```



Appendix C – GCNN and GLRP

Publication

Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer

Chereda H, Bleckmann A, Menck K, Perera-Bel J, Stegmaier P, Auer F, Kramer F, Leha A, Beißbarth T

Genome Med. 2021 Mar 11;13(1):42;
doi: https://doi.org/10.1186/s13073-021-00845-7

Chereda *et al. Genome Medicine* (2021) 13:42
https://doi.org/10.1186/s13073-021-00845-7

Genome Medicine

RESEARCH Open Access

Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer

Hryhorii Chereda[1], Annalen Bleckmann[2], Kerstin Menck[2], Júlia Perera-Bel[3], Philip Stegmaier[4], Florian Auer[5], Frank Kramer[5], Andreas Leha[6] and Tim Beißbarth[1,7*]

Abstract

Background: Contemporary deep learning approaches show cutting-edge performance in a variety of complex prediction tasks. Nonetheless, the application of deep learning in healthcare remains limited since deep learning methods are often considered as non-interpretable black-box models. However, the machine learning community made recent elaborations on interpretability methods explaining data point-specific decisions of deep learning techniques. We believe that such explanations can assist the need in personalized precision medicine decisions via explaining patient-specific predictions.

Methods: Layer-wise Relevance Propagation (LRP) is a technique to explain decisions of deep learning methods. It is widely used to interpret Convolutional Neural Networks (CNNs) applied on image data. Recently, CNNs started to extend towards non-Euclidean domains like graphs. Molecular networks are commonly represented as graphs detailing interactions between molecules. Gene expression data can be assigned to the vertices of these graphs. In other words, gene expression data can be structured by utilizing molecular network information as prior knowledge. Graph-CNNs can be applied to structured gene expression data, for example, to predict metastatic events in breast cancer. Therefore, there is a need for explanations showing which part of a molecular network is relevant for predicting an event, e.g., distant metastasis in cancer, for each individual patient.

Results: We extended the procedure of LRP to make it available for Graph-CNN and tested its applicability on a large breast cancer dataset. We present Graph Layer-wise Relevance Propagation (GLRP) as a new method to explain the decisions made by Graph-CNNs. We demonstrate a sanity check of the developed GLRP on a hand-written digits dataset and then apply the method on gene expression data. We show that GLRP provides patient-specific molecular subnetworks that largely agree with clinical knowledge and identify common as well as novel, and potentially druggable, drivers of tumor progression.

Conclusions: The developed method could be potentially highly useful on interpreting classification results in the context of different omics data and prior knowledge molecular networks on the individual patient level, as for example in precision medicine approaches or a molecular tumor board.

Keywords: Gene expression data, Explainable AI, Personalized medicine, Precision medicine, Classification of cancer, Deep learning, Prior knowledge, Molecular networks

*Correspondence: tim.beissbarth@bioinf.med.uni-goettingen.de
[1]Medical Bioinformatics, University Medical Center Göttingen, Göttingen, Germany
[7]Campus-Institute Data Science (CIDAS), University of Göttingen, Göttingen, Germany
Full list of author information is available at the end of the article

Background

Gene expression profiling by microarrays or next-generation sequencing has played a significant role in identifying predictive gene signatures and discovering individual biomarkers in cancer prognosis [1]. High-throughput sequencing produces huge amounts of gene expression data that can potentially be used for deriving clinical predictors (e.g., predicting occurrence of metastases) and identifying novel drug targets. Breast cancer is one of the paradigmatic examples of the utility of high-throughput data to derive prognostic molecular signatures (PAM50, MammaPrint, OncotypeDX) [2, 3] that predict clinical outcome. Based on the expression of 50 genes, the PAM50 classifier is widely used to divide breast cancers into four main molecular subtypes: luminal A, luminal B, triple-negative/basal-like, and HER2-enriched [4]. While the two luminal subtypes are characterized by high hormone receptor expression and generally have a better prognosis, the basal-like breast cancers are a heterogeneous group of hormone receptor- and HER2-negative breast cancers that are highly proliferative and often metastasize early. MammaPrint and OncotypeDX are 70- and 21-gene expression signatures that stratify patients according to the likelihood of metastasis. Although molecular signatures have prognostic impact, a more complete analysis of the molecular characteristics in the individual patient is required for personalized breast cancer therapy [2]. We hypothesize that molecular signatures can differ from one patient to another due to the heterogeneity of breast cancers. Such molecular signatures can be depicted as patient-specific subnetworks that are parts of a molecular network representing background knowledge about biological mechanisms. Presenting interpretable patient-specific subnetworks to clinicians and researchers enables better interpretability of the data for further medical and pharmaceutical insights, and possibly, for extended treatment options.

From a machine learning (ML) perspective, the prediction of a clinical outcome is a classification task, and molecular signatures can be identified as discriminative features. One drawback is that the search for molecular signatures is based on high-dimensional gene expression datasets, where the number of genes is much higher than the number of patients. The "curse of dimensionality" leads to instability in the feature selection process across different datasets. Stability can be improved including prior knowledge of molecular networks (e.g., pathways) into ML approaches [5]. ML methods benefit from pathway knowledge since neighboring genes are not treated as independent but instead similarities among adjacent genes, which should have similar expression profiles, are captured [6].

The essence of our classification task is to predict an occurrence of distant metastasis based on gene expression data structured by a molecular network (encoded as a graph) representing connections between genes. The patients are represented as graph signals (gene expression data) on a single graph. Since each vertex of a molecular network has a corresponding gene expression value as an attribute, we perform a graph signal classification task. Patients' gene expression profiles create different graph signal patterns that can be learned by the means of deep learning.

In recent years, deep learning has been widely applied on image data using convolutional neural networks (CNNs). The CNNs exploit the grid-like structure of images and cannot directly process data structured in non-Euclidean domains. Examples of non-Euclidean data domains include networks in social sciences and molecular networks in biology. Recently, deep learning methods extended to domains like graphs and manifolds [7]. Graph-CNN [8] learns graph signal patterns and can be applied to our graph signal classification task.

Deep neural networks are able to model complex interactions between the input and output variables. This complexity does not allow to track what role a particular input feature plays in the output; thus, a neural network itself as a black-box ML model does not provide interpretable insights.

On the other hand, decisions proposed by neural networks have to be explained before they can be taken into account in the clinical domain [9]. The European Union's recent General Data Protection Regulation (GDPR) restricted automated decision making produced by algorithms [10]. Article 13 of [10] specifies that clinics should provide patients with "meaningful information about the logic involved". Article 22 of [10] states that a patient shall have the right not to be subject to an automated decision unless the patient gives a consent with it (paragraph 2.c). Therefore, the explainability of deep neural networks becomes an imperative for clinical applications.

Explanation methods aim at making classification decisions of complex ML models interpretable in terms of input variables. These methods use one of two available approaches [11]: functional or message passing. The first group of methods produces explanations out of local analysis of a prediction. It includes the sensitivity analysis, Taylor series expansion, and the model agnostic approaches LIME [12] and SHAP [13]. The second group [14, 15] provides explanations by running a backward pass in a computational graph, which generates a prediction as its output. The Layer-Wise Relevance Propagation (LRP) method [15] combines through the framework of deep Taylor decomposition [11] functional and message passing approaches to generate relevances of each input feature. For a fixed input feature, the relevance shows how much this feature influences the classifier's decision.

The relevances are generated for each data point (in our application each patient) individually.

In image data, LRP exhibited promising results and has been applied in cancer research to identify prognostic biomarkers: Klauschen et al. [16] applied LRP for visual scoring of tumor-infiltrating lymphocytes (TIL) on hematoxylin and eosin breast cancer images. Binder et al. [17] used LRP to identify spatial regions (cancer cell, stroma, TILs) on morphological tumor images that explained predictions of molecular tumor properties (like protein expression).

There are also some interpretation methods specialized for Graph Neural Networks (GNN). In [18–20], the authors provided explanation methods that are exclusively based on and crafted only for Graph Convolutional Network [21] utilizing a convolutional architecture which is a simplified version of that of Graph-CNN [8] we use. Ying et al. [22] suggested the model-agnostic GNNExplainer that is suitable for node classification, link prediction, and graph classification, but the authors did not consider an application of their approach to graph signal classification [23, 24], which is the problem at hand. The GNN-LRP method [25] proposes explanations in the form of scored sequences of edges on the input graph (i.e., relevant walks). Such a sequence represents a path extracted from the input to the output of GNN that brings insights for GNN's decision strategy. This is useful especially for graph classification tasks, where each data point is represented as an individual graph. In our task, patients are represented as graph signals on a single graph, so that this method is not applicable.

Hence, there is still a lack of methods explaining individualized predictions in the context of graph signal classification task. Here, we adapted an existing LRP technique to graph convolutional layers of Graph-CNN [8] incorporating prior knowledge of a molecular network. Our approach generates explanations in the form of relevant subgraphs for each data point and allows to provide interpretable molecular subnetworks that are individual for each patient. According to the knowledge of the authors, an explanation method that benefits from prior knowledge and provides patient-specific subnetworks has not been shown before. The novelty of our work consists of two parts. First, we present the Graph Layer-wise Relevance Propagation (GLRP) method delivering data point-specific explanations for Graph-CNN [8]. Second, we train Graph-CNN on a large breast cancer dataset to predict an occurrence of distant metastasis and show how patient-specific molecular subnetworks assist in personalized precision medicine decisions: We interpret the classifier's predictions by patient-specific subnetworks that explain the differential clinical outcome and identify therapeutic vulnerabilities.

Methods

Gene expression data and molecular network

Protein-protein interaction network

We used the Human Protein Reference Database (HPRD) protein-protein interaction (PPI) network [26] as the molecular network to structure the gene expression data. The database contains protein-protein interaction information based on yeast two-hybrid analysis, in vitro and in vivo methods. The PPI network is an undirected graph with binary interactions between pairs of proteins. The graph is not connected.

Breast cancer data

We applied our methods to a large breast cancer patient dataset that we previously studied and preprocessed [27]. That data is compiled out of 10 public microarray datasets measured on Affymetrix Human Genome HG-U133 Plus 2.0 and HG-U133A arrays. The datasets are available from the Gene Expression Omnibus (GEO) [28] data repository (accession numbers GSE25066, GSE20685, GSE19615, GSE17907, GSE16446, GSE17705, GSE2603, GSE11121, GSE7390, GSE6532). The RMA probe-summary algorithm [29] was used to process each of the datasets, and only samples with metadata on metastasis-free survival were selected and combined together on the basis of HG-U133A array probe names. Quantile normalization was applied over all datasets. In the case of several probes mapping to one gene, only the probe with the highest average value was considered. After pre-processing the dataset contained 12,179 genes in 969 patients. The patients were assigned to one of two classes: 393 patients with distant metastasis within the first 5 years and 576 patients without metastasis having the last follow-up between 5 and 10 years. Breast cancer molecular subtypes for the patient samples were predicted in [27] utilizing *genefu* R-package [30].

After mapping of 12,179 genes to the vertices of the PPI, the resulting PPI graph consisted of 7168 vertices (mapped genes) in 207 connected components. The main connected component had 6888 vertices, and each of the other 206 components had from 1 to 4 vertices. For further analyses, we utilized only the main connected component since the Graph-CNN requires the graph to be connected. The preprocessed data is provided in [31].

Expression data of HUVECs before and after TNFα stimulation

For validation purposes, we analyzed gene expression data from human umbilical vein endothelial cells (HUVECs) treated or not treated with tumor necrosis factor alpha TNFα [32]. The data, provided by the same authors (GEO database series: GSE144803), containing 39 sample pairs (treated and untreated), were suitable for a binary classification task and balanced. The expression data were

quantile normalized and mapped to vertices of HPRD PPIs resulting in 7798 genes in the main connected component.

Problem formulation

We focus on explaining classifier decisions of Graph-CNN adapting existing LRP approaches for graph convolutional layers. LRP should be applied as a postprocessing step to a model already trained for the ML task. The task is formulated as a binary classification of gene expression data $X \in R^{n \times m}$ to a target variable $Y \in \{0,1\}^n$. n is the number of data points (patients) and m is the number of features (genes). The information of the molecular network is presented as an undirected weighted graph $G = (V, E, A)$, where V and E denote the sets of vertices and edges respectively and A denotes the adjacency matrix. The Graph-CNN was designed to work with weighted graphs. We define weighted adjacency matrix A of dimensionality $m \times m$ since in general molecular networks can be weighted. For the unweighted HPRD PPI network, the matrix A has only "0s" and "1s" as its elements. A row x of the gene expression matrix X contains data from one data point (patient) and can be mapped to the vertices of the graph G. In such a way, values of x are interpreted as a graph signal.

A trained neural network can be represented as a function $f : R_+^m \rightarrow [0,1]$ mapping the input to the probability of the output class. The input x is a set of gene expression values $x = \{x_g\}$ where g denotes a particular gene. The function $f(x)$ computes the probability that a certain pattern of gene expression values is present w.r.t to the output class. LRP methods apply propagation rules from the output of the neural network to the input in order to quantify the relevance score $R_g(x)$ for each gene g. These relevances show how much gene g influences the prediction $f(x)$:

$$\forall x : f(x) = \sum_g R_g(x). \tag{1}$$

Equation (1) [11] demonstrates that the relevance scores are calculated w.r.t every input data point x.

Graph Convolutional Neural Network and Layer-wise Relevance propagation

Usual CNNs learn data representations on grid-like structures. The Graph-CNN [8] as a deep learning technique is designed to learn features on weighted graphs. The convolution on graphs is used to capture localized patterns of a graph signal. This operation is based on spectral graph theory. The main operator to investigate the spectrum of a graph is the graph Laplacian $L = D - A$, where D is a weighted degree matrix, and A is a weighted adjacency matrix. L is a real symmetric positive semidefinite matrix that can be diagonalized such that $L = U \Lambda U^T$, where $\Lambda = diag([\lambda_1, \ldots, \lambda_m])$ is a diagonal non-negative real valued matrix of eigenvalues, matrix U is composed of eigenvectors. Matrices U and U^T define the Fourier and the inverse Fourier transform respectively. According to the convolution theorem, the operation of graph convolution can be viewed as a filtering operation:

$$y = h_\theta(L)x = h_\theta\left(U \Lambda U^T\right) x = U h_\theta(\Lambda) U^T x, \tag{2}$$

where $x, y \in R^m$, and the filter $h_\theta(\Lambda)$ is a function of eigenvalues (graph frequencies). To localize filters in space, the authors in [8] decided to use a polynomial parametrization

$$h_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k, \tag{3}$$

where $\theta \in R^k$ is a vector of parameters. The order of the polynomial, which is equal to $K-1$, specifies the local $K-1$ hop neighborhood. The neighborhood is determined by the shortest path distance. The polynomial filter can be computed recursively, as a Chebyshev expansion, which is commonly used in graph signal processing to approximate kernels [33]. The Chebyshev polynomial $T_k(x)$ of order k is calculated as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$. The Chebyshev expansion applies for values that lie in $[-1, 1]$; therefore, the diagonal matrix of eigenvalues Λ has to be derived from a rescaled Laplacian $L = (D - A)/\lambda_{max} - I_n$. Thus, the filtering operation can be rewritten as

$$y = h_\theta(\Lambda)x = \sum_{k=0}^{K-1} \theta_k T_k(L)x = [\bar{x}_0, \ldots, \bar{x}_{K-1}]\,\theta, \tag{4}$$

where $\bar{x}_k = 2L\bar{x}_{k-1} - \bar{x}_{k-2}$ with $\bar{x}_0 = x$ and $x_1 = Lx$. The transition in Eq. 4 is done according to the observation $\left(U \Lambda U^T\right)^k = U \Lambda^k U^T$. The filtering at the convolutional layer boils down to an efficient sequence of $K-1$ sparse matrix-vector multiplications and one dense matrix-vector multiplication [8].

LRP is based on the theoretical framework of deep Taylor decomposition. The function $f(x)$ from Eq. (1) can be decomposed in terms of the Taylor expansion at some chosen root point x^* so that $f(x^*) = 0$. The first order Taylor expansion of f(x) is:

$$\begin{aligned} f(x) &= f(x^*) + \sum_{g=1}^{m} \frac{\partial f}{\partial x}\Big|_{x=x^*} \cdot \left(x_g - x_g^*\right) + \epsilon \\ &= 0 + \sum_{g=1}^{m} R_g(x) + \epsilon \end{aligned} \tag{5}$$

where the relevances $R_g(x)$ are the partial differentials of the function $f(x)$. The details of how to choose a good root point are described in [11]. The $f(x)$ represents an output neuron of a neural network which consists of multiple layers and each layer consists of several neurons. A

neuron receives a weighted sum of its inputs and applies a nonlinear activation function. The idea of the deep Taylor decomposition is to perform a first order Taylor expansion at each neuron of the neural network. These expansions allow to produce relevance propagation rules that compute relevances at each layer in a backward pass. The rules redistribute the relevance from layer to layer starting from output until the input is reached. The value of the output represents the model's decision which is equal to the total relevance detected by the model.

LRP is commonly applied to deep neural networks consisting of layers with rectified linear units (ReLU) nonlinearities. In our experiments, we use only this activation function. Let i and j be single neurons at two consecutive layers at which the relevance should be propagated from j to i. The activation function has this form:

$$a_j = max\left(0, \sum_i a_i w_{ij} + b_j\right) \tag{6}$$

where a_i, a_j are neurons' values, w_{ij} are weights, and b_j is bias. Noticeably, the layers of this type always have non-negative activations. The relevance propagation rule is the following:

$$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+ + \epsilon} R_j, \tag{7}$$

where w_{ij}^+ corresponds to the positive weights w_{ij} and ϵ stabilizes numerical computations [9]. We set ϵ to 1^{-10}. Equation (7) depicts the z^+ rule coming from deep Taylor decomposition [11]. The z^+ rule is commonly applied to the convolutional and fully connected layers. It favors the effect of only positive contributions to the model decisions. The first input layer can have other propagation rules that are specific to the domain [34]. In our work, we used the rule (7) for the input layer as well since the gene expression data has positive values.

In order to propagate relevance through the filtering (4), we rewrite it as follows:

$$y = \sum_{k=0}^{K-1} \theta_k T_k(L)x = \left[\bar{L}_0, \ldots, \bar{L}_{K-1}\right]\theta x = Wx, \tag{8}$$

where matrix $W \in R^{m\times m}$ connects nodes y and x. The computation of matrix W is done as: $W = \left[\bar{L}_0, \ldots, \bar{L}_{K-1}\right]\theta$, where $\bar{L}_k = 2L\bar{L}_{k-1} - \bar{L}_{k-2}$ with $\bar{L}_0 = I$ and $\bar{L}_1 = L$ are the Chebyshev polynomials of the Laplacian matrix.

Each convolutional layer has F_{in} channels

$$\left[x_1, \ldots, x_{F_{in}}\right] \in R_+^{m\times F_{in}} \tag{9}$$

in the input feature map and F_{out} channels

$$\left[y_1, \ldots, y_{F_{out}}\right] \in R^{m\times F_{out}} \tag{10}$$

of the output feature map. We consider the values of output feature maps before applying ReLU non-linearities on them. The $F_{in} \times F_{out}$ vectors of the Chebyshev coefficients $\theta_{i,j} \in R^k$ are the layer's trainable parameters. The input feature map can be transformed into a vector $\hat{x} = \left[x_1^T, \ldots, x_{F_{in}}^T\right]^T \in R_+^{m\cdot F_{in}}$. We adapt Eq. (8) to compute the j^{th} channel of the output feature map based on the input feature map:

$$\begin{aligned} y_j &= \left[\bar{L}_0, \ldots, \bar{L}_{K-1}\right] \cdot \left[\theta_{1,j}, \ldots, \theta_{F_{in}j}\right] \cdot \left[x_1^T, \ldots, x_{F_{in}}^T\right]^T \\ &= \left[\hat{L}_{1,j}, \ldots, \hat{L}_{F_{in},j}\right] \cdot \left[x_1^T, \ldots, x_{F_{in}}^T\right]^T \\ &= \hat{W}_j \times \hat{x} \in R^m \end{aligned} \tag{11}$$

where $\hat{L}_{i,j} = \left[\bar{L}_0, \ldots, \bar{L}_{K-1}\right]\theta_{i,j} \in R^{m\times m}$, $\hat{W}_j = \left[\hat{L}_{1,j}, \ldots, \hat{L}_{F_{in},j}\right] \in R^{m\times m\cdot F_{in}}$

Since the j^{th} channel of the output feature map is connected through the matrix-vector multiplication with the input feature map, $\hat{W}_j$ can be treated as a matrix of weights joining two fully connected layers. Therefore, the relevance $R_y^j \in R_+^m$ from the j^{th} output channel can be propagated to the input feature map relevance $R_{\hat{x}}^j \in R_+^{m\cdot F_{in}}$ according to the rule (7). Overall, the relevance propagated from the output feature map to the input feature map is:

$$R_{\hat{x}} = \sum_{j=1}^{F_{out}} R_{\hat{x}}^j \in R_+^{m\cdot F_{in}}. \tag{12}$$

For running LRP on graph convolutional layers, one needs to compute huge and dense matrices $\hat{W}_j$. It requires $K-2$ sparse matrix-matrix multiplications and one sparse to dense matrix-matrix multiplication. The computations for relevance propagation are heavier and much more memory demanding compared to the filtering (4). The code implementing our GLRP approach is available in [35].

GLRP on gene expression data

To demonstrate the utility of GLRP, the Graph-CNNs were trained on two gene expression datasets described in the "Gene expression data and molecular network" section. In our previous study [23], the gene expression data were standardized for the training. But in this paper, we did not standardize the data. The argument for it is the following. For the non-image data, to standardize the input features is the usual practice. However, in case of standardization, the input features are treated independently. For an image, the neighboring pixels are highly correlated. If the pixels as features are standardized across the dataset, then this can distort the pattern of the image quite

significantly and lead to misinterpretation. Analogically, feature wise standardization of microarray data changes expression patterns of genes located in the same neighborhood of a molecular network (HPRD PPI in our case). This might affect the explainability of the Graph-CNN that we aim at. Therefore, we trained the Graph-CNN directly on the quantile normalized data avoiding the additional standardization step. Instead, we subtracted the minimal value (5.84847) of the data from each cell of the gene expression matrix to keep the gene expression values non-negative. If initially, GE data was lying in [5.84847, 14.2014], now it is in the interval [0.0, 8.3529]. This transformation allows Graph-CNN to converge faster, to apply the LRP propagation rule (7) suitable for non-negative input values, and to preserve original gene expression patterns in local neighborhoods of the PPI network.

For each of the two gene expression datasets structured by the same prior knowledge (HPRD PPI), we used a 10-fold cross validation over a whole dataset to estimate the predictive performance of Graph-CNN. The hyperparameters such as the number of filters, the presence of pooling, the learning rate, and decay were tweaked manually on this 10-fold cross validation.

The architecture of the Graph-CNN trained on the HUVECs dataset and its performance are given in the "Comparison of subnetworks derived by GLRP to gene-coexpression networks identified by WGCNA" section.

For the breast cancer dataset, the Graph-CNN architecture consisted of two graph convolutional layers following maximum pooling of size 2, and two hidden fully connected layers with 512 and 128 units respectively. Each graph convolutional layer contained 32 filters covering the vertex' neighborhood of size 7. For the performance comparison, we trained a "glmgraph" method [36] implementing network-constrained sparse regression model using HPRD PPI network, and Random Forest without any prior knowledge as baselines. The results on 10-fold cross validations are presented in the "GLRP to deliver patient-specific subnetworks" section.

Further we generated the patient-specific (data point specific) subnetworks via GLRP. For that, each of the gene expression datasets was randomly split again: 90% training and 10% test. We retrained the Graph-CNN on 90% of data using manually selected hyperparameters from 10-fold cross validation, and propagated relevances on test data which was not "seen" by the model during training to make it more challenging. Since the LRP rule (7) propagates only positive contributions, our Graph-CNN had two output neurons for binary classification tasks that showed the probability of these two classes. For each patient in the test set, relevance was propagated by GLRP from the predicted output neuron to the input neurons representing genes (vertices) of the underlying molecular network. The workflow to deliver the patient-specific subnetworks is depicted on Fig. 1. A patient-specific subnetwork explaining the prediction was constructed from the 140 most relevant genes. Selecting more than 140 top relevant vertices entailed visualization issues. The single-

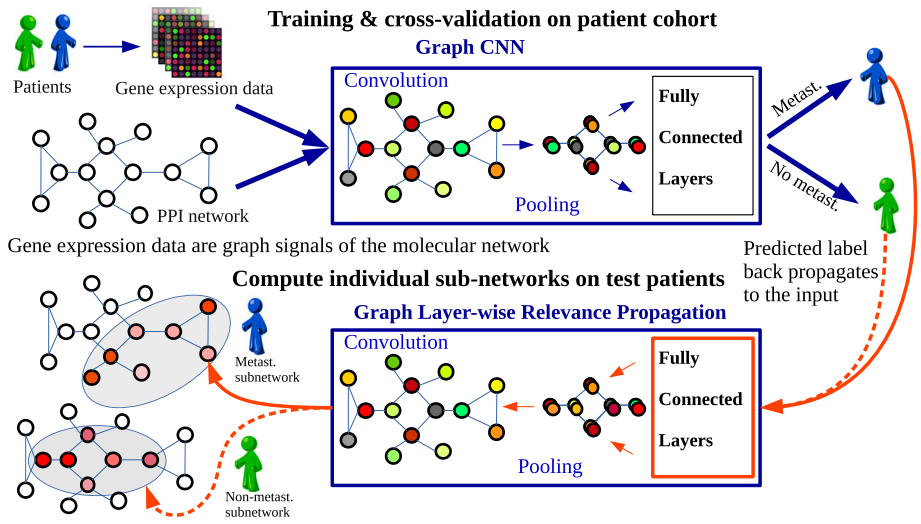


Fig. 1 The workflow to obtain a data point-specific subnetwork. For clarity, a data point represented by a gene expression profile of a patient from the breast cancer dataset. The molecular network (HPRD PPI) structures the genes and is the same for every patient. Patient's gene-expression values are assigned to every vertex of the HPRD PPI so that the patient is represented as a graph signal. Trained Graph-CNN performs graph convolutions and as output classifies the patient as metastatic or non-metastatic. GLRP is applied as a post hoc processing, propagating the relevance from the predicted label up to the input features (vertices of the molecular network). Top 140 highly relevant vertices constitute a molecular subnetwork. Molecular subnetworks differ from one patient to another

tons were deleted so that the subnetwork consisted mainly of around 130 vertices. The same workflow was applied to generate data-point-specific subnetworks for the data described in the "Expression data of HUVECs before and after TNFα stimulation" section.

Pathway analysis

Enrichment of signal transduction pathways annotated in the TRANSPATH® database version 2020.1 [37] in genes prioritized by GLRP were analyzed using the geneXplain platform version 6.1 [38]. The analysis based on the Fisher's exact test [39] was carried out for gene sets obtained for individual patients from the breast cancer dataset as well as for their combination into subtype gene sets.

The following calculations were applied to investigate differences in pathway hits. Let P denote a set of pathway genes and S_i and S_k two subnetwork gene sets, so that $P_i = P \cap S_i$ and $P_k = P \cap S_k$ are the sets of pathway genes matched by the two subnetworks. The difference $\triangle P_{i,k}$ in matched pathway genes was then calculated as $|(P_i \cup P_k) \setminus (P_i \cap P_k)|/|P_i \cup P_k|$ with $|P_i \cup P_k| > 0$. For each selected pathway, we calculated $\triangle P_i, k$ for each pair of subnetworks and reported the median of examined pairs.

Comparison of subnetworks derived by GLRP to gene-coexpression networks identified by WGCNA

To further examine the biological relevance of subnetwork genes prioritized by GLRP and for the purpose of comparison to an already available method that uses expression and network information to prioritize gene sets, we analyzed the gene expression data described in "Expression data of HUVECs before and after TNFα stimulation" section. We compared gene sets identified in our subnetworks to gene modules and differentially expressed genes in response to TNFα identified by Rhead et al. [32]. Rhead et al. [32] reported gene modules obtained by weighted gene co-expression network analysis (WGCNA). The method has been applied in many studies and constructs a gene network based on expression measurements from which it can derive modules of co-expressed genes [40]. We trained a Graph-CNN on the gene expression data to classify the TNFα treatment status of HUVECs. The Graph-CNN architecture consisted of 2 convolutional layers with 4 and 8 filters respectively followed by one hidden fully connected layer with 128 nodes. The vertex's neighborhood covered by graph convolutions was of size 7. No pooling was used. The performance of the Graph-CNN in 10-fold cross validation: mean 100*AUC, accuracy, and F1-weighted were 99.49, 96.25% and 96.06%, respectively. A random forest achieved the same performance. We generated the subnetworks according to the "GLRP on gene expression data" section, retrained the Graph-CNN on 70 randomly selected samples, and applied GLRP on 8 test samples (4 treated and 4 not treated). The test samples were predicted correctly. For each of the 8 test samples, we constructed a subnetwork. Associations between subnetwork genes sets and 16 gene modules defined by Rhead et al. [32] as well as 589 upregulated genes (log-fold change > 0.5, FDR < 0.01), 425 downregulated genes (log-fold change < -0.5, FDR < 0.01), and the combined set of 1014 DE genes were analyzed using the *Functional classification* tool of the geneXplain platform [41]. Fisher test calculations were carried out with a total contingency table count corresponding to the number of genes in [32, file S1 of] after mapping to Ensembl [42] gene ids (10022 genes). Rhead et al. [32] assigned a color code to the 16 gene co-expression modules and denoted them as *black*, *blue*, *brown*, *cyan*, *green*, *greenyellow*, *grey*, *magenta*, *midnightblue*, *pink*, *purple*, *red*, *salmon*, *tan*, *turquoise*, and *yellow* which is maintained in results reported here.

Results

Sanity check of the implemented graph LRP

To initially validate our implemented LRP, we applied Graph-CNN on the MNIST dataset [43] in the same way as described in the paper [8]. The MNIST dataset contains 70,000 images of hand-written digits each having a size of 28 by 28 pixels. To apply Graph-CNN on the image data, we constructed an 8 nearest-neighbors graph similarly to the schema proposed in [8], with the exception that all the weights are equal to 1. The weight 1 is more natural for the graph connecting neighboring image pixels. Thus, each image is a graph signal represented by node attributes—pixel values. We achieved high classification accuracy (99.02%) on the test set for the Graph-CNN, which is comparable to the performance of classical CNN (99.33%) reported in [8]. The number of parameters was the same for both methods.

Usually, to manage box-constrained pixel values, the special pixel-specific LRP rule is applied for the input layer. This pixel-specific rule highlights not only the digits itself, but also the contours of the digits [34, Fig. 13 of]. In contrast, the rule (7) highlights only those positively relevant parts of the image where the signal of the digit is present. We kept the propagation rule (7) for the input and all other layers in all our experiments. Further, we visually compared on the same digits how the heatmaps generated by implemented GLRP correspond to the heatmaps generated by usual LRP procedure applied on classical CNN (Fig. 2).

The heatmaps were rendered only for the classes predicted by classical CNN and Graph-CNN. In this case, the classes are "6" and "3". For the Graph-CNN, a bigger part of the digit is relevant for the classification since the covered neighborhood can be expanded up to 24 hops. Graph-CNN's filters are isotropic; thus, they tend to cover

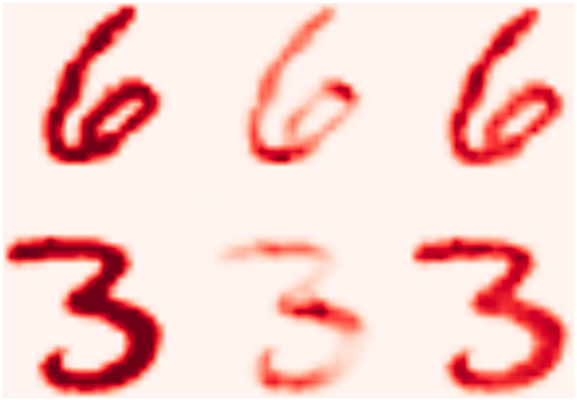


Fig. 2 From left to right: initial image, LRP on classical CNN and GLRP on Graph-CNN

roundish areas that concern rounded patterns (curves) of the digit (Additional file 1: Fig. S1).

Genes selected by GLRP correlate with modules identified by gene co-expression network analysis

In the analysis of TNF-induced gene expression changes in HUVECs, our procedure prioritized in total 168 genes of which 105 genes were found in subnetworks of all eight test samples (Additional file 2). Remarkably, the *green* gene module, which was the most strongly correlated one with TNFα upregulation [32], showed significant association (adjusted p value < 0.05) with the combined set of subnetwork genes, with genes found in the majority of subnetworks and also with 5 of the 8 subnetworks (Additional file 2). At the same significance level, the *turquoise* gene module described in [32] was strongly associated with 2 of 8 subnetworks and with genes found in all 8 subnetworks. In addition, both the *green* and the *turquoise* modules showed moderate association (adjusted p value < 0.1) with the majority of gene sets defined on the basis of the test subnetworks. Furthermore, we found strong (adjusted p value < 0.05) or moderately (adjusted p value < 0.1) significant overlap between upregulated genes and some subnetwork gene sets. The gene modules *cyan*, *greenyellow*, and *midnightblue* did not overlap with GLRP-derived subnetworks. These results demonstrate partial agreement between gene sets suggested by GLRP, another gene network analysis and classical differential expression analysis. Hence, the GLRP-based subnetworks gathered biologically meaningful genes and may even complement other approaches in revealing important properties of the underlying biological systems. Additionally, another two gene sets were compared with WGCNA modules: the intersection of subnetworks genes and genes that occurred in more than in 4 test samples subnetworks. Notably, the individual subnetworks shared more genes with the *green* and *turquoise* WGCNA modules than those described gene sets, pointing out the ability of GLRP to identify sample-specific genes.

GLRP to deliver patient-specific subnetworks

We applied the GLRP to the Graph-CNN trained on gene expression data from the "Breast cancer data" section. The gene expression data was structured by a protein-protein interaction network. The standardization of features was not performed as described in the "GLRP on gene expression data" section. The prediction task performed by the Graph-CNN was to classify patients into 2 groups, metastatic and non-metastatic. The results of a 10-fold cross validation are depicted in Table 1. While Graph-CNN and glmgraph utilized the HPRD PPI network topology, a random forest did not use any prior knowledge. glmgraph was not evaluated on non-standardized data, since it had convergence issues in this case. The metrics were averaged over folds and the standard errors of their means were calculated.

The GLRP was applied as described in the"GLRP on gene expression data" section. We retrained the Graph-CNN on 872 patients and generated relevances for 97 test patients. The relevances were propagated from the Graph-CNN's output node corresponding to the correctly predicted class. The most frequently selected features are summarized in Additional file 1: Table S1. The eukaryotic translation elongation factor EEF1A1, which is overexpressed in the majority of breast cancers and protects tumor cells from proteotoxic stress [44], was the sole factor that was selected in all of the 97 test set patients. Other frequently selected features in both non-metastatic as well as metastatic patients included genes such as the epithelial-to-mesenchymal-transition (EMT)-related gene VIM (46/58 non-metastatic, 30/39 metastatic patients), the extracellular matrix protein FN1 (43/58 non-metastatic, 22/39 metastatic patients), the actin cytoskeleton regulator CFL1 (7/58 non-metastatic, 7/39 metastatic

Table 1 Performance of Graph-CNN on metastatic event prediction, depending on normalization

Method	Std	100*AUC	Accuracy, %	F1-weighted, %
Graph-CNN	-	82.57±1.25	76.07±1.30	75.82±1.33
Random Forest	-	81.27±1.66	74.23±1.73	73.47±1.84
Graph-CNN	+	82.16±1.25	76.18±1.36	75.86±1.35
Random Forest	+	81.40±1.76	74.74±1.67	74.00±1.82
glmgraph	+	80.88±1.37	75.14±1.30	74.73±1.39

Std stands for standardization of features (genes)

patients), and the estrogen receptor ESR1 28/58 non-metastatic, 10/39 metastatic patients) that are all known to be linked with breast cancer development and progression [45–48]. This indicates that our method successfully identified relevant key players with a general role in breast tumorigenesis.

Additionally, we show individualized PPI subnetworks delivered for four correctly predicted breast cancer patients (Table 2) from the microarray data set. Two of them had been assigned with the most common subtype luminal A (LumA), while the other two suffered from the highly aggressive basal-like subtype. In each group, one patient with early metastasis was picked and one who did not develop any within at least 5 years of follow-up.

The generated PPI subnetworks are displayed in Fig. 3. The sequence of pictures in order ABCD is the same as in the table.

Interestingly, the networks of both LumA patients contained ESR1 which fits well since this subtype is considered as estrogen receptor positive [49]. In contrast, genes often associated with the basal-like subtype and a poor prognosis such as MCL1, CTNNB1, EGFR, or SOX4 were found in the basal-like patient GSM519217 suggesting that the generated networks are capable of extracting breast cancer subtype-specific features. The comparison of the subnetworks of the non-metastatic and the metastatic patients furthermore revealed some patient-specific genes which might give valuable information about specific mechanisms of tumorigenesis and therapeutic vulnerabilities in the respective patient. In general, it seemed that the subnetworks of the non-metastatic patients contained more genes that have been linked to better prognostic outcomes such as JUP, PCBP1, and HMGN2 in GSM615695 [50–52] or RASA1, IL6ST, KRT19, and RPS14 in GSM150990 [53–56], while the networks of both metastatic patients harbored genes that are known to be involved in aggressive tumor growth or therapy resistance which might explain the early metastatic spread in these patients. Some examples are CDK1, SFN, and XPO1 in GSM519217 [57–59] or CAV1, PTPN11, and FTL in GSM615233 [60–62].

Table 2 Patients that the PPI subnetworks are generated for

Patient's ID	Subtype	Metastatic event	Time of metastases, years	Last follow-up, years
GSM519217	Basal	1	0.9	-
GSM615233	LumA	1	0.79	-
GSM615695	Basal	0	-	5.38
GSM150990	LumA	0	-	9.93

However, not only the presence of specific genes might be important, but also their overall expression level. Our analyses identified, e.g., the EMT-related gene VIM as one of the most relevant nodes in the subnetworks of both metastatic patients in which the gene was highly expressed (> 75% quantile based on the gene expression throughout the whole patient cohort). In contrast, VIM was also present in the subnetworks of the two non-metastatic patients, however, with a lower relevance and a particularly low expression (< 25% quantile). VIM is an important marker for EMT and high expression levels correlate with a motile, mesenchymal-like cancer cell state, thus making VIM an essential effector of metastasis [45].

A comparison of subnetwork genes of 79 correctly predicted test set patients to a database of signal transduction pathways confirmed significant enrichment of pathways that have previously been associated with cancer disease mechanisms such as the EGF, ER-alpha, p53, and TGFbeta pathways as well as Caspase and beta-catenin networks. Comparisons were performed for each patient as well as for subtype gene sets formed by combining subnetwork genes of patients associated with a breast cancer subtype. Results for the 238 signaling pathways from the TRANSPATH® database that were significantly enriched with subtype genes are visualized in Fig. 4. Differences in enrichment significance may suggest that the importance of some signaling pathways detected this way is subtype-specific, e.g., for YAP ubiquitination or the VE-cadherin network (orange heatmap, Fig. 4, see also Additional file 1: Table S2 for details). The pattern of enrichment found on the level of cancer subtypes coincided well with the findings for subnetwork genes of individual patients revealing several molecular networks with elevated significance in both subtype and patient gene sets such as the EGF pathway, although the patient-level visualization did not suggest subtype-specific enrichment (green heatmap, Fig. 4). One source of these observations can be that patient subnetworks tend to be associated with certain pathways but cover different pathway components (genes). We therefore compared pathway genes in pairs of patient subnetworks for the 33 largest pathways. In 18 pathways, the median pair of patient subnetworks differed in 33% or more of the genes matched within a pathway (see also Additional file 1: Table S3 for details). These results demonstrate that the subnetworks obtained by

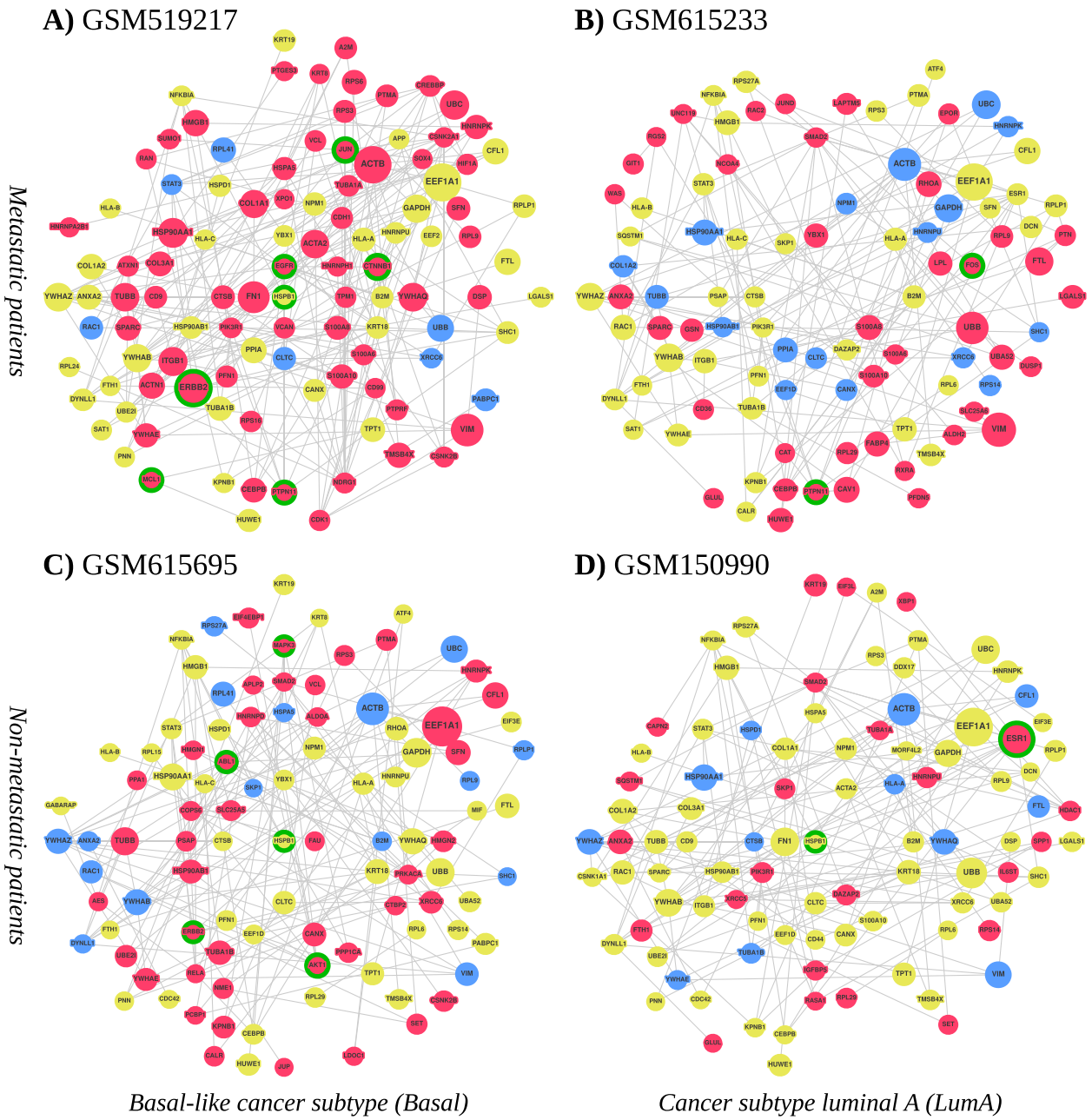


Fig. 3 The PPI subnetworks for (1) metastatic patients **a** (GSM519217) and **b** (GSM615233) and (2) non-metastatic patients **c** (GSM615695) and **d** (GSM150990). The coloring of the node is based on gene expression levels by 25% and 75% quantiles (blue=LOW, yellow=NORMAL, red=HIGH), based on the gene expression throughout the whole patient cohort. The size of vertices corresponds to the relevance scores within one subnetwork. All the subnetworks are highly relevant compared to the rest of the PPI network. Green circles highlight targetable genes

Graph-CNN were enriched with common signaling pathways relevant for the respective disease and can assign patient-specific priorities to pathway components.

Finally, we tested whether the subnetworks can also be used for finding potentially targetable genetic vulnerabilities that could open new options for personalized treatment decisions. We applied the "MTB report" methodology described in [63] to identify actionable genes present in the subnetworks. For that, we extended the algorithm to match high expression with gain of function alterations, and low expression with loss of function alterations. The results are summarized in Table 3.

Although information about the presence of actionable genetic variants is missing from our patient microarray

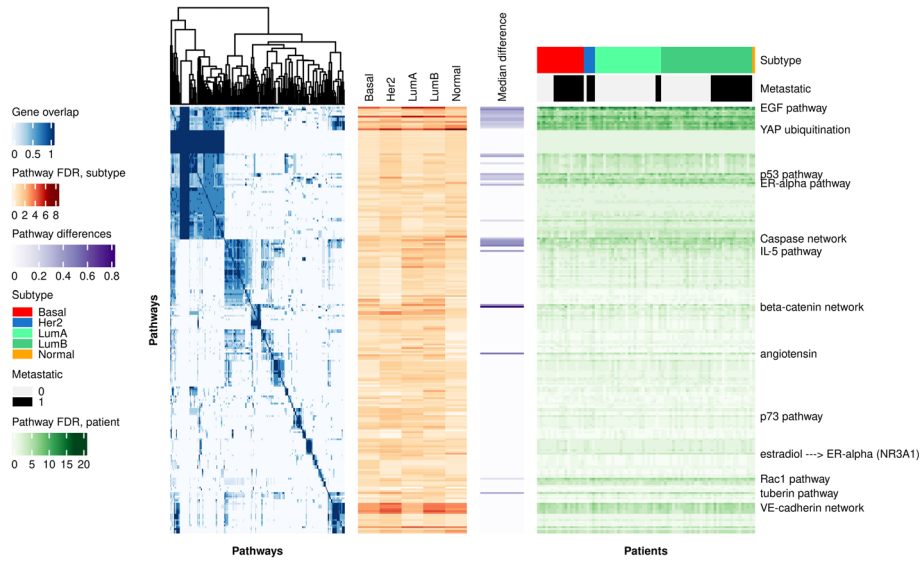


Fig. 4 Signal transduction pathway analysis of subnetwork genes reported for 79 patients in 5 subtypes. (From left to right) Blue heatmap: 238 signaling pathways clustered according to proportion of shared subnetwork genes; Orange heatmap: Enrichment significance of pathways in subnetwork genes combined from patients of given subtype. Darker orange indicates higher significance; Purple heatmap: Median difference in matched pathway genes observed in pairwise comparisons of subnetwork gene sets from patients mapped to 33 pathways. Darker purple indicates higher tendency of pairs of subnetwork gene sets to coincide with different pathway genes; Green heatmap: Enrichment significance of pathways in subnetwork genes of 79 patients. Darker green indicates higher significance. Corresponding subtypes and metastatic status are shown by the annotation above the heatmap. A detailed version of this figure capturing pathway and sample names is provided in Additional file 1: Fig. S2

data, the information generated by the PPI subnetworks could be used to define specific panels for subsequent sequencing. Indeed, the MTB reports highlighted specific genes that could be targeted therapeutically in each of the four patients: In the non-metastatic LumA patient GSM150990 ESR1 was proposed as therapeutic target which is in line with current treatment regimens that use hormone therapy as the main first-line treatment of choice for this patient subgroup. In contrast, in the metastatic LumA patient GSM615233 FOS and PTPN11 were identified as novel actionable alterations. In the often rapidly relapsing basal-like patients HSPB1 and ERBB2 were

Table 3 Actionable genes identified by the MTB report workflow

Patient	Gene	Expression	Known Var	Predicts
615695	HSPB1	Normal	expression	Response to gemcitabine
	ABL1	High	GoF	Response to ABL TK inhibitors (imatinib, desatininb, ponatinib, regorafenib. . .)
	AKT1	High	GoF	Response to PI3K, AKT, MTOR inhibitors; resistance to BRAF inhibitors
	ERBB2	High	GoF	Response to ERBB2, EGFR, MTOR, AKT inhibitors
	MAPK3	High	GoF	Resistance to EGFR inhibition
519217	HSPB1	Normal	expression	Response to gemcitabine
	CTNNB1	High	GoF	Response to everolimus + letrozole; resistance to Tankyrase inhibitors
	EGFR	High	GoF	Response to EGFR, ERBB2, HSP90 and MEK inhibitors
	ERBB2	High	GoF	Response to ERBB2, EGFR, MTOR, AKT inhibitors
	JUN	High	overexpr	Response to irbesartan (angiotensin II antagonist)
	MCL1	High	GoF	Resistance to anti-tubulin agents
	PTPN11	High	GoF	Response to MEK inhibitors
615233	FOS	High	overexpr	Response to irbesartan (angiotensin II antagonist)
	PTPN11	High	GoF	Response to MEK inhibitors
150990	HSPB1	Normal	expression	Response to gemcitabine
	ESR1	High	GoF	Response to novel ER degraders, fulvestrant, tamoxifen

Genes from the PPI subnetworks were matched to known genomic alterations (Known Var) that predict either response or resistance to drugs (Predicts). High and low gene expression were matched to gain of function (GoF) and loss of function (LoF) genomic variants, respectively

identified as common targets as well as MAPK3, AKT1, and ABL1 for the non-metastatic patient GSM615695 or EGFR, MCL1, CTNNB1, PTPN11, and JUN for the metastatic patient GSM519217, thereby suggesting novel possibilities for combinatory or alternative treatments. Taken together, GLRP provides subnetworks centered around known oncogenic drivers that seem reasonable in the context of cancer biology and can help to identify patient-specific cancer dependencies and therapeutic vulnerabilities in the context of precision oncology.

Discussion

In our work, we focused on the interpretability of a deep learning method utilizing molecular networks as prior knowledge. We implemented LRP for Graph-CNN and provided the sanity check of the developed approach on the MNIST dataset. Essentially, the main aim of the paper was to explain the prediction of metastasis for breast cancer patients by providing an individual molecular subnetwork specific for each patient. The patient-specific subnetworks provided interpretability of the deep learning method and demonstrated clinically relevant results on the breast cancer dataset.

Supposedly, the performance of Graph-CNN can be improved. The batch normalization technique [64] that is used to accelerate the training of deep neural networks is not seen to be available for the Graph-CNN, so this can be the way to enhance its performance. The LRP rule for batch normalization layers is yet another procedure to be adapted for Graph-CNN.

Another possibility to identify genes (and construct subnetworks out of them) influencing classifier decisions is to apply model-agnostic SHAP and LIME explanation methods. LIME method provides explanations of a data point based on feature perturbations. The method samples perturbations from a Gaussian distribution, ignoring correlations between features. It leads to the instability of explanations that is not favorable for personalized medicine. SHAP provides Shapley values for each feature of a data point as well but does not have such an issue, so we attempted to derive patients-specific subnetworks applying TreeExplainer and KernelExplainer from SHAP python module on Random Forest and Graph-CNN respectively. The subnetworks were build on the basis of HPRD PPI utilizing positive Shapley values, which were pushing prediction to a higher probability of corresponding class (metastatic or non-metastatic). The subnetworks obtained were mostly consisting from single vertices. In contrast, the subnetworks from GLRP and Graph-CNN were mostly connected. The SHAP's DeepExplainer approach suitable for convenient deep learning models is not applicable for Graph-CNN. The model-agnostic KernelExplainer computes SHAP values out of a debiased lasso regression. Reevaluating the model happens several thousands numbers of times specified by a user as well as a small background dataset is needed for integrating out features. Hence, the KernelExplainer is not scalable and application of it on Graph-CNN resulted in not connected subnetworks as well.

Furthermore, the sensitivity of Graph-CNN to the changes of prior knowledge is still to be investigated. Authors in [8] showed that for the MNIST images a random graph connecting pixels significantly decreases the performance destroying local connectivity. In our case, the permutation of the vertices of the PPI network does not influence the classifier performance on standardized gene expression data. Yet, PPI network is a small world network and its degree distribution fits to the power law with the exponent $\alpha = 2.70$. It implies great connectivity between proteins and means that any two nodes are separated by less than six hops. The filters of convolutional layers cover a 7-hop neighborhood of each vertex, so we assume it still might be enough to capture the gene expression patterns. In our future work, we will investigate how the properties of the prior knowledge influence the performance and explainability of Graph-CNN.

The subnetworks generated by GLRP contained common potential oncogenic drivers which indicates that they can extract the essential cancer pathways. Indeed, our analyses identified genes associated with hormone receptor-positive breast cancer (e.g. ESR1, IL6ST, CD36, GLUL, RASA1) in the networks from the patients with estrogen receptor positive, LumA breast cancer and genes associated with the basal-like subtype (e.g., EGFR, SOX4, AKT1 as well as high levels of HNRNPK) in the basal-like patients, underlining the biological relevance of the networks. Next to subtype-specific genes, the networks contained several oncogenes that were found in all four patients and could thus represent common drivers of breast cancer initiation and progression. One example is the actin-binding protein cofilin (CFL1) that regulates cancer cell motility and invasiveness [46]. Another interesting candidate is STAT3 which is activated in more than 40% of breast cancers and can cause deregulated cell proliferation and epithelial-to-mesenchymal transition (EMT) [65]. Our graphs not only displayed patient-specific PPI subnetworks, but also concisely visualized the relevance of each node and its expression levels. This information is potentially relevant to judge the biological significance of the gene in a patient-specific context.

Next to the common genes found in all four networks, each network was characterized by several special, cancer-associated genes which are of high interest because they might represent patient-specific central signaling nodes and therapeutic vulnerabilities. Some examples are PTPN11 that is known to activate a transcriptional program associated with cancer stem cells or the EMT-related genes SOX4 or VIM that might be responsible for the high

invasive capacity of the tumors and their early metastasis formation [45, 61, 66, 67]. Interestingly, the network of the metastatic patient GSM615233 harbored the genes FABP4 and LPL which both have been shown to interact with CD36, another highly expressed node in the network, to support cell proliferation and counteract apoptosis [68–70]. In contrast, in the non-metastatic patient GSM150990 especially the interleukin receptor IL6ST and the Ras GTPase-activating protein 1 (RASA1) seem to be interesting because for both high expression levels have been linked with a favorable prognosis [53, 54]. In the other non-metastatic patient GSM615695 high levels of HMGN2 and PCBP1 were identified which both have been shown to be able to inhibit cell proliferation [51, 52]. Although the experimental validation for the networks is still missing, it is tempting to speculate that these genes might contribute to the benign phenotype of the tumor in these patients.

All patient-specific subnetworks contained relevant drug targets that have been largely studied in breast cancer (e.g., ERBB2, ESR1, EGFR, AKT1). Yet, resistance mechanisms in breast cancer targeted therapies represent a big challenge; many of the identified therapeutic approaches have failed [71] due to the highly interconnected nature of signaling pathways and potential circumvents. A promising way forward could involve the molecular characterization of the tumor with transcriptomics and a parallel culture of patient-derived organoids. PPI networks could elucidate the right combination strategy by identifying central signaling nodes. Different therapeutic strategies could be tested on organoids and confirm the best strategy that synergistically blocks cancer cell escape routes and minimizes the emergence of survival mechanisms. Only the identification of relevant mechanisms of action for cell survival as well as of the factors involved in resistance for each patient, together with a more precise and personalized characterization of each cancer phenotype, may provide useful improvements in current therapeutic approaches.

Conclusions

We present a novel Graph-CNN-based feature selection method that benefits from prior knowledge and provides patient-specific subnetworks. We adapted the existing Layer-wise Relevance Propagation technique to the Graph-CNN, demonstrated it on MNIST data, and showed its applicability on a large breast cancer dataset. Our new approach generated individual patient-specific molecular subnetworks that influenced the model's decision in the given context of a classification problem. The subnetworks selected by the developed method utilizing general prior knowledge are relevant for prediction of metastasis in breast cancer. They contain common as well as subtype-specific cancer genes that match the clinical subtype of the patients, together with patient-specific genes that could potentially be linked to aggressive/benign phenotypes. In the context of a breast cancer dataset GLRP provides patient-specific explanations for the Graph-CNN that largely agree with clinical knowledge, include oncogenic drivers of tumor progression, and can help to identify therapeutic vulnerabilities. We therefore conclude that our method GLRP in combination with Graph-CNN is a new, useful, and interpretable ML approach for high-dimensional genomic data-sets. Generated classifiers rely on prior knowledge of molecular networks and can be interpreted by patient-specific subnetworks driving the individual classification result. These subnetworks can be visualized and interpreted in a biomedical context on the individual patient level. This approach could thus be useful for precision medicine approaches such as for example the molecular tumorboard.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13073-021-00845-7.

Additional file 1: Contains Supplementary Tables S1-S3 and Supplementary Figures S1, S2.

Additional file 2: Subnetwork genes obtained for 8 test samples and analysis of their association with gene modules reported by [32] as well as differentially expressed (DE) genes.
Worksheet *Subnetwork genes 8 samples* provides identifiers and gene symbols of 167 subnetwork genes, in how many and in which samples they were selected. Worksheet *Gene module enrichment* presents results of Fisher test calculations comparing subnetwork gene sets to gene modules and DE gene sets. Each row contains data for a DE gene set or a gene module consisting of the total group size and column tripletts with *p*-value, adjusted *p*-value as well as the number of hits, respectively, observed in comparisons to the union of genes from 8 subnetworks, the set of genes occurring in the majority, the set of genes found in all of the subnetworks and each of the 8 samples. Highlighted are rows corresponding to *green* and *turquoise* gene modules, which were most often significantly associated with subnetwork gene sets (grey), adjusted *p*-values below 0.05 (red) and between 0.05 and 0.1 (yellow).

Abbreviations

ML: Machine learning; LRP: Layer-wise Relevance Propagation; GNN: Graph Neural Network; CNN: Convolutional Neural Network; GLRP: Graph Layer-wise Relevance Propagation; WGCNA: Weighted gene co-expression network analysis; GDPR: General Data Protection Regulation; GEO: Gene Expression Omnibus; HPRD: Human Protein Reference Database; PPI: Protein-protein interaction; ReLU: Rectified linear unit; HUVEC: Human umbilical vein endothelial cells; EMT: Epithelial-to-mesenchymal transition

Acknowledgements

We would like to acknowledge Michaela Bayerlová, Mark Gluzman, and Vladyslav Yushchenko for fruitful discussions. HC is a member of the International Max Planck Research School for Genome Science, part of the Göttingen Graduate Center for Neurosciences, Biophysics, and Molecular Biosciences. TB is a member of the Göttingen Campus Institute Data Science.

Authors' contributions

HC and TB designed the study. HC developed and implemented the approach and performed the computational experiments. AB, FK, and PS provided major contributions to the study design. AB and KM provided clinical insights as well as JPB and PS provided biological insights, performing analyses of patient-specific subnetworks. FA developed the web-site to visualize the subnetworks.

TB and AL provided machine learning insights. HC, TB, AL, KM, and PS wrote the manuscript. All authors read and approved the final manuscript.

Funding
This work was funded by the German Ministry of Education and Research (BMBF) e:Med project *MyPathSem* (031L0024) and the project *MTB-Report* by the big data initiative of the Volkswagenstiftung. KM was supported by German Research Foundation (DFG) project 424252458. We acknowledge support by the Open Access Publication Funds of the Göttingen University. Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials
The utilized breast cancer datasets are accessible from Gene Expression Omnibus (GEO) [28] data repository (accession numbers GSE25066, GSE20685, GSE19615, GSE17907, GSE16446, GSE17705, GSE2603, GSE11121, GSE7390, GSE6532). The HUVECs gene expression data [32] is available in GEO database (GSE144803). The HPRD PPI network can be found in [26]. The preprocessed breast cancer data, the adjacency matrix of the HPRD PPI network, and the code of the GLRP method are provided in http://mypathsem.bioinf.med.uni-goettingen.de/resources/glrp [31] and https://gitlab.gwdg.de/UKEBpublic/graph-lrp [35]. The web-site to explore patient-specific subnetworks is in http://mypathsem.bioinf.med.unigoettingen.de/MetaRelSubNetVis/ [72].

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
PS is an employee of geneXplain GmbH, Germany. The remaining authors declare that they have no competing interests.

Author details
[1]Medical Bioinformatics, University Medical Center Göttingen, Göttingen, Germany. [2]Dept. of Medicine A (Hematology, Oncology, Hemostaseology and Pulmonology), University Hospital Münster, Münster, Germany. [3]Hospital del Mar Medical Research Institute (IMIM), Barcelona, Spain. [4]geneXplain GmbH, Wolfenbüttel, Germany. [5]IT Infrastructure for Translational Medical Research, University of Augsburg, Augsburg, Germany. [6]Medical Statistics, University Medical Center Göttingen, Göttingen, Germany. [7]Campus-Institute Data Science (CIDAS), University of Göttingen, Göttingen, Germany.

Received: 26 August 2020 Accepted: 5 February 2021
Published online: 11 March 2021

References

1. Perera-Bel J, Leha A, Beißbarth T. In: Badve S, Kumar GL, editors. Bioinformatic methods and resources for biomarker discovery, validation, development, and integration. Cham: Springer; 2019, pp. 149–64. https://doi.org/10.1007/978-3-319-95228-4_11.
2. Rivenbark AG, O'Connor SM, Coleman WB. Molecular and cellular heterogeneity in breast cancer: challenges for personalized medicine. Am J Pathol. 2013;183(4):1113–24. https://doi.org/10.1016/j.ajpath.2013.08.002.
3. Sørlie T. Molecular classification of breast tumors: toward improved diagnostics and treatments. In: Target Discovery and Validation Reviews and Protocols. Totowa: Humana Press; 2007. p. 91–114. https://doi.org/10.1385/1-59745-165-7:91.
4. Fragomeni SM, Sciallis A, Jeruss JS. Molecular subtypes and local-regional control of breast cancer. Surg Oncol Clin N Am. 2018;27(1):95–120. https://doi.org/10.1016/j.soc.2017.08.005.
5. Porzelius C, Johannes M, Binder H, Beißbarth T. Leveraging external knowledge on molecular interactions in classification methods for risk prediction of patients. Biom J. 2011;53(2):190–201. https://doi.org/10.1002/bimj.201000155, Accessed 01 Dec 2020.
6. Johannes M, Brase JC, Fröhlich H, Gade S, Gehrmann M, Fälth M, Sültmann H, Beissbarth T. Integration of pathway knowledge into a reweighted recursive feature elimination approach for risk stratification of cancer patients. Bioinformatics. 2010;26(17):2136–44. https://doi.org/10.1093/bioinformatics/btq345.
7. Monti F, Boscaini D, Masci J, Rodola E, Svoboda J, Bronstein MM. Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 5115–24.
8. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS); 2016. p. 3844–52.
9. Yang Y, Tresp V, Wunderle M, Fasching PA. Explaining therapy predictions with layer-wise relevance propagation in neural networks. In: 2018 IEEE International Conference on Healthcare Informatics (ICHI); 2018. p. 152–62. https://doi.org/10.1109/ICHI.2018.00025.
10. Parliament and C. of the European Union. General data protection regulation. 2016. https://gdpr-info.eu/.
11. Montavon G, Lapuschkin S, Binder A, Samek W, Müller K-R. Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recogn. 2017;65:211–22. https://doi.org/10.1016/j.patcog.2016.11.008.
12. Ribeiro MT, Singh S, Guestrin C. "Why Should I Trust You?": explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD); 2016. p. 1135–44.
13. Lundberg S, Lee S-I. A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS); 2017. p. 4768–77.
14. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer Vision – ECCV. Cham: Springer; 2014. p. 818–33. https://doi.org/10.1007/978-3-319-10590-1_53.
15. Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE. 2015;10(7):0130140. https://doi.org/10.1371/journal.pone.0130140.
16. Klauschen F, Müller K-R, Binder A, Bockmayr M, Hägele M, Seegerer P, Wienert S, Pruneri G, de Maria S, Badve S, Michiels S, Nielsen TO, Adams S, Savas P, Symmans F, Willis S, Gruosso T, Park M, Haibe-Kains B, Gallas B, Thompson AM, Cree I, Sotiriou C, Solinas C, Preusser M, Hewitt SM, Rimm D, Viale G, Loi S, Loibl S, Salgado R, Denkert C. Scoring of tumor-infiltrating lymphocytes: from visual estimation to machine learning. Semin Cancer Biol. 2018;52:151–7. https://doi.org/10.1016/j.semcancer.2018.07.001. Immuno-oncological biomarkers.
17. Binder A, Bockmayr M, Hägele M, Wienert S, Heim D, Hellweg K, Stenzinger A, Parlow L, Budczies J, Goeppert B, Treue D, Kotani M, Ishii M, Dietel M, Hocke A, Denkert C, Müller K-R, Klauschen F. Towards computational fluorescence microscopy: machine learning-based integrated prediction of morphological and molecular tumor profiles. arXiv:1805.11178 [cs]. 2018.
18. Xie S, Lu M. Interpreting and understanding graph convolutional neural network using gradient-based attribution method. arXiv:1903.03768 [cs]. 2019. Accessed 12 July 2020.
19. Schwarzenberg R, Hübner M, Harbecke D, Alt C, Hennig L. Layerwise relevance visualization in convolutional text graph classifiers. arXiv:1909.10911 [cs]. 2019. Accessed 06 Nov 2020.
20. Pope PE, Kolouri S, Rostami M, Martin CE, Hoffmann H. Explainability methods for graph convolutional neural networks. In: 2019 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 10764–73. https://doi.org/10.1109/CVPR.2019.01103. ISSN: 2575-7075.
21. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907 [cs, stat]. 2016. Accessed 09-01-2017.
22. Ying R, Bourgeois D, You J, Zitnik M, Leskovec J. GNNExplainer: generating explanations for graph neural networks. Adv Neural Inf Process Syst. 2019;32:9240–51.
23. Chereda H, Bleckmann A, Kramer F, Leha A, Beissbarth T. Utilizing molecular network information via graph convolutional neural networks to predict metastatic event in breast cancer. Stud Health Technol Inform. 2019;267:181–6. https://doi.org/10.3233/SHTI190824.
24. Rhee S, Seo S, Kim S. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization; 2018. p. 3527–34. https://doi.org/10.24963/ijcai.2018/490. https://www.ijcai.org/proceedings/2018/490.

25. Schnake T, Eberle O, Lederer J, Nakajima S, Schütt KT, Müller K-R, Montavon G. XAI for graphs: explaining graph neural network predictions by identifying relevant walks. arXiv:2006.03589 [cs, stat]. 2020. Accessed 29 Oct 2020.
26. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A, Balakrishnan L, Marimuthu A, Banerjee S, Somanathan DS, Sebastian A, Rani S, Ray S, Harrys Kishore CJ, Kanth S, Ahmed M, Kashyap MK, Mohmood R, Ramachandra YL, Krishna V, Rahiman BA, Mohan S, Ranganathan P, Ramabadran S, Chaerkady R, Pandey A. Human protein reference database?2009 update. Nucleic Acids Res. 2009;37:767–72. https://doi.org/10.1093/nar/gkn892.
27. Bayerlová M, Menck K, Klemm F, Wolff A, Pukrop T, Binder C, Beißbarth T, Bleckmann A. Ror2 signaling and its relevance in breast cancer progression. Front Oncol. 2017;7:135. https://doi.org/10.3389/fonc.2017.00135.
28. Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, Marshall KA, Phillippy KH, Sherman PM, Holko M, Yefanov A, Lee H, Zhang N, Robertson CL, Serova N, Davis S, Soboleva A. NCBI GEO: archive for functional genomics data sets–update. Nucleic Acids Res. 2013;41(Database issue):991–5. https://doi.org/10.1093/nar/gks1193.
29. Irizarry RA, Hobbs B, Collin F, Beazer–Barclay YD, Antonellis KJ, Scherf U, Speed TP. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics. 2003;4(2):249–64. https://doi.org/10.1093/biostatistics/4.2.249.
30. Gendoo DMA, Ratanasirigulchai N, Schröder MS, Paré L, Parker JS, Prat A, Haibe-Kains B. Genefu: an R/Bioconductor package for computation of gene expression-based signatures in breast cancer. Bioinformatics. 2016;32(7):1097–9. https://doi.org/10.1093/bioinformatics/btv693.
31. Bayerlová M, Chereda H. Preprocessed breast cancer data. 2020. http://mypathsem.bioinf.med.uni-goettingen.de/resources/glrp.
32. Rhead B, Shao X, Quach H, Ghai P, Barcellos LF, Bowcock AM. Global expression and CpG methylation analysis of primary endothelial cells before and after TNFa stimulation reveals gene modules enriched in inflammatory and infectious diseases and associated DMRs. PLoS ONE. 2020;15(3):0230884. https://doi.org/10.1371/journal.pone.0230884.
33. Hammond DK, Vandergheynst P, Gribonval R. Wavelets on graphs via spectral graph theory. Appl Comput Harmon Anal. 2011;30(2):129–50. https://doi.org/10.1016/j.acha.2010.04.005.
34. Montavon G, Samek W, Müller K-R. Methods for interpreting and understanding deep neural networks. Digit Signal Process. 2018;73:1–15. https://doi.org/10.1016/j.dsp.2017.10.011.
35. Chereda H. Graph layer-wise relevance propagation (GLRP). Gitlab. 2020. https://gitlab.gwdg.de/UKEBpublic/graph-lrp.
36. Chen L, Liu H, Kocher J-PA, Li H, Chen J. glmgraph: an R package for variable selection and predictive modeling of structured genomic data. Bioinformatics. 2015;31(24):3991–3. https://doi.org/10.1093/bioinformatics/btv497.
37. Krull M, Voss N, Choi C, Pistor S, Potapov A, Wingender E. TRANSPATH ®: an integrated database on signal transduction and a tool for array analysis. Nucleic Acids Res. 2003;31(1):97–100. http://dx.doi.org/10.1093/nar/gkg089. https://academic.oup.com/nar/article-pdf/31/1/97/7127458/gkg089.pdf.
38. Koschmann J, Bhar A, Stegmaier P, Kel A, Wingender E. "Upstream analysis": an integrated promoter-pathway analysis approach to causal interpretation of microarray data. Microarrays. 2015;4(2):270–86. https://doi.org/10.3390/microarrays4020270.
39. Fisher RA. On the interpretation of χ^2 from contingency tables, and the calculation of P. J R Stat Soc. 1922;85(1):87–94. https://doi.org/10.2307/2340521.
40. Zhang B, Horvath S. A general framework for weighted gene co-expression network analysis. Stat Appl Genet Mol Biol. 2005;4:17. https://doi.org/10.2202/1544-6115.1128.
41. Kolpakov F, Poroikov V, Selivanova G, Kel A. GeneXplain–identification of causal biomarkers and drug targets in personalized cancer pathways. J Biomol Tech. 2011;22(Suppl):16.
42. Yates AD, Achuthan P, Akanni W, Allen J, Allen J, Alvarez-Jarreta J, Amode MR, Armean IM, Azov AG, Bennett R, Bhai J, Billis K, Boddu S, Marugán JC, Cummins C, Davidson C, Dodiya K, Fatima R, Gall A, Giron CG, Gil L, Grego T, Haggerty L, Haskell E, Hourlier T, Izuogu OG, Janacek SH, Juettemann T, Kay M, Lavidas I, Le T, Lemos D, Martinez JG, Maurel T, McDowall M, McMahon A, Mohanan S, Moore B, Nuhn M, Oheh DN, Parker A, Parton A, Patricio M, Sakthivel MP, Abdul Salam AI, Schmitt BM, Schuilenburg H, Sheppard D, Sycheva M, Szuba M, Taylor K, Thormann A, Threadgold G, Vullo A, Walts B, Winterbottom A, Zadissa A, Chakiachvili M, Flint B, Frankish A, Hunt SE, Ilsley G, Kostadima M, Langridge N, Loveland JE, Martin FJ, Morales J, Mudge JM, Muffato M, Perry E, Ruffier M, Trevanion SJ, Cunningham F, Howe KL, Zerbino DR, Flicek P. Ensembl 2020. Nucleic Acids Res. 2020;48(D1):682–8. https://doi.org/10.1093/nar/gkz966.
43. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324. https://doi.org/10.1109/5.726791.
44. Lin C-Y, Beattie A, Baradaran B, Dray E, Duijf PHG. Contradictory mRNA and protein misexpression of EEF1A1 in ductal breast carcinoma due to cell cycle regulation and cellular stress. Sci Rep. 2018;8(1):13904. https://doi.org/10.1038/s41598-018-32272-x.
45. Sharma P, Alsharif S, Fallatah A, Chung BM. Intermediate filaments as effectors of cancer development and metastasis: a focus on keratins, vimentin, and nestin. Cells. 2019;8(5):497. https://doi.org/10.3390/cells8050497.
46. Wang W, Eddy R, Condeelis J. The cofilin pathway in breast cancer invasion and metastasis. Nat Rev Cancer. 2007;7(6):429–40. https://doi.org/10.1038/nrc2148.
47. Lin T-C, Yang C-H, Cheng L-H, Chang W-T, Lin Y-R, Cheng H-C. Fibronectin in cancer: Friend or foe. Cells. 2019;9(1):27. https://doi.org/10.3390/cells9010027.
48. Feng Y, Spezia M, Huang S, Yuan C, Zeng Z, Zhang L, Ji X, Liu W, Huang B, Luo W, Liu B, Lei Y, Du S, Vuppalapati A, Luu HH, Haydon RC, He T-C, Ren G. Breast cancer development and progression: Risk factors, cancer stem cells, signaling pathways, genomics, and molecular pathogenesis. Genes Dis. 2018;5(2):77–106. https://doi.org/10.1016/j.gendis.2018.05.001.
49. Perou CM, Sørlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA, Fluge Ø., Pergamenschikov A, Williams C, Zhu SX, Lønning PE, Børresen-Dale A-L, Brown PO, Botstein D. Molecular portraits of human breast tumours. Nature. 2000;406(6797):747–52. https://doi.org/10.1038/35021093.
50. Bailey CK, Mittal MK, Misra S, Chaudhuri G. High motility of triple-negative breast cancer cells is due to repression of plakoglobin gene by metastasis modulator protein SLUG. J Biol Chem. 2012;287(23):19472–86. https://doi.org/10.1074/jbc.m112.345728.
51. Shi H, Li H, Yuan R, Guan W, Zhang X, Zhang S, Zhang W, Tong F, Li L, Song Z, Wang C, Yang S, Wang H. PCBP1 depletion promotes tumorigenesis through attenuation of p27 Kip1 mRNA stability and translation. J Exp Clin Cancer Res. 2018;37(1):187. https://doi.org/10.1186/s13046-018-0840-1.
52. Fan B, Shi S, Shen X, Yang X, Liu N, Wu G, Guo X, Huang N. Effect of HMGN2 on proliferation and apoptosis of MCF-7 breast cancer cells. Oncol Lett. 2018;17(1):1160–6. https://doi.org/10.3892/ol.2018.9668.
53. Liu Y, Liu T, Sun Q, Niu M, Jiang Y, Pang D. Downregulation of Ras GTPase-activating protein 1 is associated with poor survival of breast invasive ductal carcinoma patients. Oncol Rep. 2014;33(1):119–24. https://doi.org/10.3892/or.2014.3604.
54. Mathe A, Wong-Brown M, Morten B, Forbes JF, Braye SG, Avery-Kiejda KA, Scott RJ. Novel genes associated with lymph node metastasis in triple negative breast cancer. Sci Rep. 2015;5(1):15832. https://doi.org/10.1038/srep15832.
55. Saha S, Kim K, Yang G-M, Choi H, Cho S-G. Cytokeratin 19 (KRT19) has a role in the reprogramming of cancer stem cell-like cells to less aggressive and more drug-sensitive cells. Int J Mol Sci. 2018;19(5):1423. https://doi.org/10.3390/ijms19051423.
56. Zhou X, Hao Q, Liao J-M, Liao P, Lu H. Ribosomal protein S14 negatively regulates c-Myc activity. J Biol Chem. 2013;288(30):21793–801. https://doi.org/10.1074/jbc.m112.445122.
57. Alexandrou S, George S, Ormandy C, Lim E, Oakes S, Caldon C. The proliferative and apoptotic landscape of basal-like breast cancer. Int J Mol Sci. 2019;20(3):667. https://doi.org/10.3390/ijms20030667.
58. Neve RM, Chin K, Fridlyand J, Yeh J, Baehner FL, Fevr T, Clark L, Bayani N, Coppe J-P, Tong F, Speed T, Spellman PT, DeVries S, Lapuk A, Wang NJ, Kuo W-L, Stilwell JL, Pinkel D, Albertson DG, Waldman FM, McCormick F, Dickson RB, Johnson MD, Lippman M, Ethier S, Gazdar A, Gray JW. A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. Cancer Cell. 2006;10(6):515–27. https://doi.org/10.1016/j.ccr.2006.10.008.

59. Taylor J, Sendino M, Gorelick AN, Pastore A, Chang MT, Penson AV, Gavrila EI, Stewart C, Melnik EM, Chavez FH, Bitner L, Yoshimi A, Lee SC-W, Inoue D, Liu B, Zhang XJ, Mato AR, Dogan A, Kharas MG, Chen Y, Wang D, Soni RK, Hendrickson RC, Prieto G, Rodriguez JA, Taylor BS, Abdel-Wahab O. Altered nuclear export signal recognition as a driver of oncogenesis. Cancer Discov. 2019;9(10):1452–67. https://doi.org/10.1158/2159-8290.cd-19-0298.
60. Qian X-L, Pan Y-H, Huang Q-Y, Shi Y-B, Huang Q-Y, Hu Z-Z, Xiong L-X. Caveolin-1: a multifaceted driver of breast cancer progression and its application in clinical treatment. OncoTargets Ther. 2019;12:1539–52. https://doi.org/10.2147/ott.s191317.
61. Aceto N, Sausgruber N, Brinkhaus H, Gaidatzis D, Martiny-Baron G, Mazzarol G, Confalonieri S, Quarto M, Hu G, Balwierz PJ, Pachkov M, Elledge SJ, van Nimwegen E, Stadler MB, Bentires-Alj M. Tyrosine phosphatase SHP2 promotes breast cancer progression and maintains tumor-initiating cells via activation of key transcription factors and a positive feedback signaling loop. Nat Med. 2012;18(4):529–37. https://doi.org/10.1038/nm.2645.
62. Chekhun VF, Lukyanova NY, Burlaka AP, Bezdenezhnykh NA, Shpyleva SI, Tryndyak VP, Beland FA, Pogribny IP. Iron metabolism disturbances in the MCF-7 human breast cancer cells with acquired resistance to doxorubicin and cisplatin. Int J Oncol. 2013;43(5):1481–6. https://doi.org/10.3892/ijo.2013.2063.
63. Perera-Bel J, Hutter B, Heining C, Bleckmann A, Fröhlich M, Fröhling S, Glimm H, Brors B, Beißbarth T. From somatic variants towards precision oncology: Evidence-driven reporting of treatment options in molecular tumor boards. Genome Med. 2018;10(1):18. https://doi.org/10.1186/s13073-018-0529-2.
64. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, PMLR. 2015;37:448–456. http://proceedings.mlr.press/v37/ioffe15.html.
65. Banerjee K, Resat H. Constitutive activation of STAT 3 in breast cancer cells: a review. Int J Cancer. 2015;138(11):2570–8. https://doi.org/10.1002/ijc.29923.
66. Bentires-Alj M, Paez JG, David FS, Keilhack H, Halmos B, Naoki K, Maris JM, Richardson A, Bardelli A, Sugarbaker DJ, Richards WG, Du J, Girard L, Minna JD, Loh ML, Fisher DE, Velculescu VE, Vogelstein B, Meyerson M, Sellers WR, Neel BG. Activating mutations of the noonan syndrome-associated SHP2/PTPN11 gene in human solid tumors and adult acute myelogenous leukemia. Cancer Res. 2004;64(24):8816–20. https://doi.org/10.1158/0008-5472.can-04-1923.
67. Zhang J, Liang Q, Lei Y, Yao M, Li L, Gao X, Feng J, Zhang Y, Gao H, Liu D-X, Lu J, Huang B. SOX4 induces epithelial?mesenchymal transition and contributes to breast cancer progression. Cancer Res. 2012;72(17):4597–608. https://doi.org/10.1158/0008-5472.can-12-1045.
68. Guaita-Esteruelas S, Bosquet A, Saavedra P, Gumà J, Girona J, Lam EW-F, Amillano K, Borràs J, Masana L. Exogenous FABP4 increases breast cancer cell proliferation and activates the expression of fatty acid transport proteins. Mol Carcinog. 2016;56(1):208–17. https://doi.org/10.1002/mc.22485.
69. Liang Y, Han H, Liu L, Duan Y, Yang X, Ma C, Zhu Y, Han J, Li X, Chen Y. CD36 plays a critical role in proliferation, migration and tamoxifen-inhibited growth of ER-positive breast cancer cells. Oncogenesis. 2018;7(12):98. https://doi.org/10.1038/s41389-018-0107-x.
70. Kuemmerle NB, Rysman E, Lombardo PS, Flanagan AJ, Lipe BC, Wells WA, Pettus JR, Froehlich HM, Memoli VA, Morganelli PM, Swinnen JV, Timmerman LA, Chaychi L, Fricano CJ, Eisenberg BL, Coleman WB, Kinlaw WB. Lipoprotein lipase links dietary fat to solid tumor cell proliferation. Mol Cancer Ther. 2011;10(3):427–36. https://doi.org/10.1158/1535-7163.mct-10-0802.
71. Nakai K, Hung MC, Yamaguchi H. A perspective on anti-EGFR therapies targeting triple-negative breast cancer. Am J Cancer Res. 2016;6(8):1609–23.
72. Auer F. Patient specific molecular sub-networks responsible for metastasis in breast cancer. 2020. http://mypathsem.bioinf.med.uni-goettingen.de/MetaRelSubNetVis.

Publisher's Note

Appendix D - NDExEdit

Publication

Data-dependent visualization of biological networks in the web-browser with NDExEdit

Auer F, Mayer S, Kramer F

PLOS Computational Biology 18(6): e1010205.
doi: https://doi.org/10.1371/journal.pcbi.1010205

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Data-dependent visualization of biological networks in the web-browser with NDExEdit

Florian Auer*, Simone Mayer, Frank Kramer

Department of IT-Infrastructure for Translational Medical Research, Faculty of Applied Computer Science, University of Augsburg, Augsburg, Germany

* florian.auer@informatik.uni-augsburg.de

OPEN ACCESS

Citation: Auer F, Mayer S, Kramer F (2022) Data-dependent visualization of biological networks in the web-browser with NDExEdit. PLoS Comput Biol 18(6): e1010205. https://doi.org/10.1371/journal.pcbi.1010205

Editor: Dina Schneidman-Duhovny, Hebrew University of Jerusalem, ISRAEL

Received: November 8, 2021

Accepted: May 15, 2022

Published: June 8, 2022

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1010205

Data Availability Statement: A live demo is hosted on GitHub Pages at https://frankkramer-lab.github.io/NDExEdit and the corresponding source code for deploying own instances is provided at https://github.com/frankkramer-lab/NDExEdit.

Abstract

Networks are a common methodology used to capture increasingly complex associations between biological entities. They serve as a resource of biological knowledge for bioinformatics analyses, and also comprise the subsequent results. However, the interpretation of biological networks is challenging and requires suitable visualizations dependent on the contained information. The most prominent software in the field for the visualization of biological networks is Cytoscape, a desktop modeling environment also including many features for analysis.

A further challenge when working with networks is their distribution. Within a typical collaborative workflow, even slight changes of the network data force one to repeat the visualization step as well. Also, just minor adjustments to the visual representation not only need the networks to be transferred back and forth. Collaboration on the same resources requires specific infrastructure to avoid redundancies, or worse, the corruption of the data. A well-established solution is provided by the NDEx platform where users can upload a network, share it with selected colleagues or make it publicly available.

NDExEdit is a web-based application where simple changes can be made to biological networks within the browser, and which does not require installation. With our tool, plain networks can be enhanced easily for further usage in presentations and publications. Since the network data is only stored locally within the web browser, users can edit their private networks without concerns of unintentional publication. The web tool is designed to conform to the Cytoscape Exchange (CX) format as a data model, which is used for the data transmission by both tools, Cytoscape and NDEx. Therefore the modified network can be directly exported to the NDEx platform or saved as a compatible CX file, additionally to standard image formats like PNG and JPEG.

Author summary

Relations in biological research are often visualized as networks. For instance, if two proteins interact with each other during a certain process, the corresponding network would show two nodes connected by one edge. But the fact that the interaction between the two exists, may not be enough. With established software solutions like Cytoscape we can add

Funding: This work was supported by the Multipath project (https://www.sys-med.de/en/junior-research-groups/multipath/) funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) through grants to FK (grant FKZ01ZX1508). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

all the information we have about our nodes and their interaction to our data foundation. Furthermore, we can change the visual appearance of our nodes and their interaction based on this information.

For example, if our network contains 20 nodes, that all interact with each other, but the strength of these interactions each range between 0 and 1, we can illustrate that by making the edges wider for strong interactions and slimmer for weak interactions. Thus, our visualization is enriched with valuable information. As of now these data-dependent modifications can only be made with a desktop client.

We introduce NDExEdit, a web-based solution for visualization changes to networks that conform to the CX data format. It allows us to import networks directly from the NDEx platform and apply changes to the visualization—including all types of mappings, one of which was briefly described above.

This is a *PLOS Computational Biology* Software paper.

Introduction

Networks are well-established in a wide range of fields in biology [1–3], and are often used, either as a source or result, in biological research. Information associated with the individual nodes or edges can go far beyond name and type, thus increasing its complexity. Within common bioinformatics workflows data integration, network analysis, and visualization accompany each other [4, 5], and comprise fundamental challenges of combining various tools.

The information-rich data contained in biological networks provide the opportunity for comprehensive visualization but requires powerful tools to achieve. Cytoscape [6] is the most prominent desktop software for biological network analysis and visualization. It employs a data-dependent visualization strategy by applying so-called "attribute-to-visual-mappings", where a node's or edge's attribute translates to its visual representation. Besides its support for large networks and its rich set of features, Cytoscape comes with overhead for quick results and a steep learning curve.

A major challenge when working with networks is their distribution. Collaboration on the same resources requires specific infrastructure to avoid redundancies, or worse, the corruption of the data. A well-established solution is provided by the NDEx platform [7, 8] where users can upload a network, share it with selected colleagues or make it publicly available. NDEx also holds the feature to provide your private networks solely to the reviewers of a submitted paper, to protect the data until publication.

NDEx is tightly connected to Cytoscape, which reveals itself in the mutual integration of both platforms. For the transmission of the networks the Cytoscape Exchange (CX) data structure [9] was developed, which not only includes the structural information of the networks but also instructions for its visual representation.

There is a recent trend in software development towards web-based solutions. Desktop applications require individual installations, which is not possible in all cases for various reasons and also brings further expense for maintenance. Furthermore, accessibility across different devices grows in importance, while web-based applications provide secure access to centralized data. In the following, we illustrate how our lightweight web application NDExEdit implements current web technologies and thereby facilitates the data-dependent visualization of biological networks.

Design and implementation

Network data model

CX is a JSON (JavaScript Object Notation) based data structure designed for the transmission of biological networks between web applications and servers. The different types of information within a network are organized into single aspects of the network. These modular components separate the basic network structure from additional information and thus enable to only load the parts of the network that are of interest for an application. Since CX is designed as a transmission format, this reduces the amount of data needed to be transferred, but still combines all data as one coherent network.

The aspects have a defined scheme for the elements they can contain that must be followed. This includes definitions for core aspects, concerning the network topology and attributes, and aspects contributed by Cytoscape handling the visual representation. They link to each other by referencing the internal ID used in the aspects, for example, refer edges the IDs of the nodes aspects they are connecting. Furthermore, it is possible to include own custom aspects without a strict definition, that will be stored at the NDEx platform, but not processed or validated.

Implementation details

The client-side visualization of networks is realized using Cytoscape.js [10]. It is a JavaScript library for browser and server-based graph rendering, including layout algorithms for positioning nodes. One of its key features is the separation of data and its representation: stylesheets are used to data-dependently select network elements and assign visual properties to them.

Cytoscape.js does not natively support the handling of networks in CX format but is used in the front-end of the NDEx platform to visualize the CX networks. Their mapping script was incorporated into NDExEdit to assure a consistent visual representation in all software tools, including Cytoscape. Therefore, modifications of the script were necessary to enable highlighting and export of the networks.

The functionality of NDExEdit rests upon the Angular [11] platform, an open-source framework for building single-page web applications. It follows the Model-View-Controller (MVC) design pattern which reduces the code required for implementing the web application. Angular is based on TypeScript [12] as the programming language, which brings advantages for development in form of static typing and support of class-based object-oriented programming (OOP).

The layout of the web application is realized using the Bootstrap [13] framework. It is an open-source CSS framework for front-end development, containing design templates for interface components.

Results

NDExEdit simplifies the visual adjustment of networks and illustrates the great potential of web-based solutions for biological research: Users with any operating system can work with NDExEdit without a requirement for installation or account. Since the installation of desktop clients is often restricted due to security concerns, web-based applications can close this gap and provide access through mobile devices. It runs only in the web browser, without any supporting backend infrastructure, which ensures data privacy while still providing flexibility in the visualization workflow. Those concerns can even be reduced further by setting up private installations and securing their accessibility.

The web application provides a lightweight interface to explore the contents of networks and facilitates the quick defining of custom visualizations dependent on the data. Networks can be layouted using a variety of built-in algorithms, and refined manually. With compliance to the Cytoscape Exchange format, the network data and its visualization is contained within the same resource, which representation also remains consistent between all tools. NDExEdit narrows the gap between desktop software to create and edit a network, and web-based platforms to decorate and distribute them.

Web-application

A typical workflow within NDExEdit starts with the import of networks, for which several options are provided: The user can browse and query the publicly available, or by supplying personal credentials also the own private networks on the NDEx platform, and load selected ones directly into the app. Alternatively, networks can be loaded from a provided NDEx UUID or URL, or a local CX file. All successfully imported networks become accessible in the overview list and are ready for modification. The home button of any subordinate page leads back to this page to be able to switch between networks.

By default, the breast cancer protein-protein interaction network by Minkyu Kim [14] is provided for demonstration purposes. The network contains the interactome of all high-confidence PPIs detected across the three breast cell lines MCF7, MDA-MB-231, and MCF10A. Besides the valuable information contained in this network, it is also a great example of how the visual representation (Fig 1) supports the comprehension of the underlying data. Therefore, it will be used in the following to demonstrate the capabilities of NDExEdit to define and edit the attribute mappings dependent on the network data.

When accessing a network in NDExEdit, general information about it will be shown next to its visualization. This view can be customized by toggling the sides or moving the separating border in any direction. The general information panel provides an overview of all node, edge, and network attributes of the network. While the network attributes can be edited directly, the remaining attributes can be explored for their distribution and the coverage of the nodes and edges by this attribute. Additionally, the network can be inspected by creating rules on the values of the node and edge attributes to be highlighted in the graph.

The visualization of the network is interactive, which means that it can be zoomed and shifted, and also the nodes and edges can be selected and moved. Detailed information about the selected elements appears on top in the information panel to be able to compare its content. With the available buttons, the graph can be fit to the viewport and for better overview and performance improvements, the labels in the network can be hidden.

Attribute mappings

A key feature within the data-dependent visualization in Cytoscape is the so-called "attribute-to-visual-mappings" where the values of an attribute are processed by a specific function to generate a new value for the visual representation. Thereby one attribute (or property in the CX context) can be mapped to several visual properties. Cytoscape and the CX-file format distinguish between three kinds of mapping types that can be applied to nodes as well as edges: discrete, continuous, and pass-through.

The values of a property can vary in its data type, which limits the types of mappings that can be applied. For example, for string values, it is not possible to apply a continuous mapping, since by its nature only discrete manifestations are given without any order.

On the other hand, the visual properties vary by type of the value to which they are mapped:

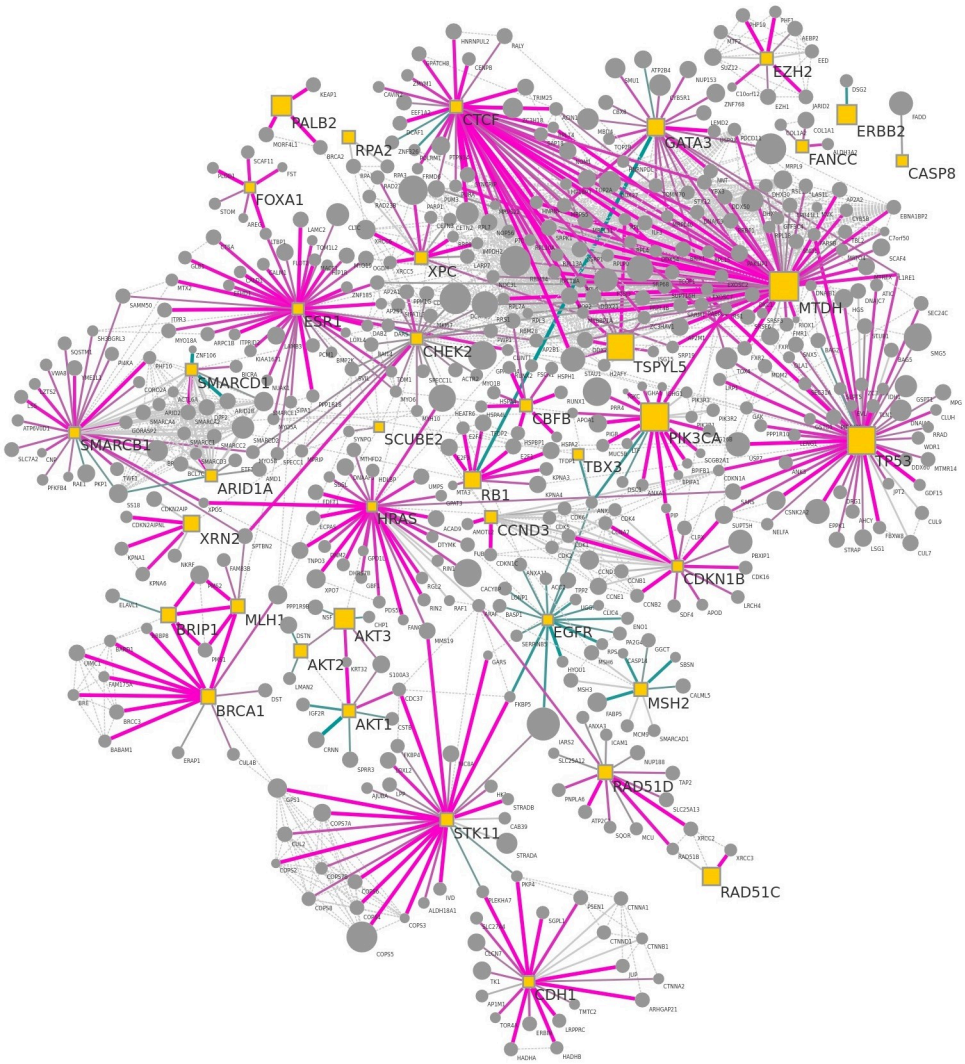


Fig 1. Breast cancer protein-protein interaction network used as example network on NDExEdit. It shows the interactome of the union of all high-confidence PPIs detected across breast cell lines MCF7, MDA-MB-231, and MCF10A. This network is available on NDEx by the UUID: e89ad762-ab4b-11ea-aaef-0ac135e8bacf.

https://doi.org/10.1371/journal.pcbi.1010205.g001

- numerical values for example "NODE_SIZE" or "EDGE_WIDTH"
- colors (in hexadecimal format) for example "NODE_FILL_COLOR"
- string values as in "ELLIPSE" for a "NODE_SHAPE"
- font declarations, including font-family, -style, and -size, are used for example for "NODE_LABEL_FONT_FACE"

NDExEdit limits the choices to select for visual mapping properties to only the applicable types to assure, that only valid mappings can be created. Custom selection tools for colors and fonts are included as well to facilitate the creation of new mappings. The highlighting of attributes and modification of the mappings does not take effect immediately to prevent disruptive errors in the data model and the visualization. Instead, the modification of other attributes is

locked and visually indicated by warning signs on the superior elements and a surrounding frame.

The mappings themselves are stored within the network in the "cyVisualProperties" aspect. This ensures a consistent visual representation of the network on all three platforms, namely NDExEdit, Cytoscape, and NDEx. Furthermore, the modification of the mappings can be continued on either NDEx or Cytoscape.

Discrete mapping. Discrete mappings are the most straightforward type of mappings: to one discrete value of a property, a corresponding mapping value is explicitly assigned. This way, all manifestations of the property can be set individually, but also left blank if no or a default value should be used. Fig 2 shows the discrete mappings of the provided sample network for the properties "Bait" and "BaitBoolean". It shows that each property has only one possible value with already several mappings to visual properties of different data types.

The mapping for the "Bait" property is shown in editing mode with an additional visual property already added using the green plus symbol next to it. The missing mapping value can easily be added using the gray plus symbol or removed with the red "X" button. Also, the visual properties can be removed or restored to the initial value before editing via the provided buttons.

The applied changes can be tested by temporarily showing their effects in the graph by using the magic wand button. All made adjustments can be omitted through the red "X" at the bottom, which leads back to the network overview. Only by actively accepting the changes the new mapping is applied and saved for export.

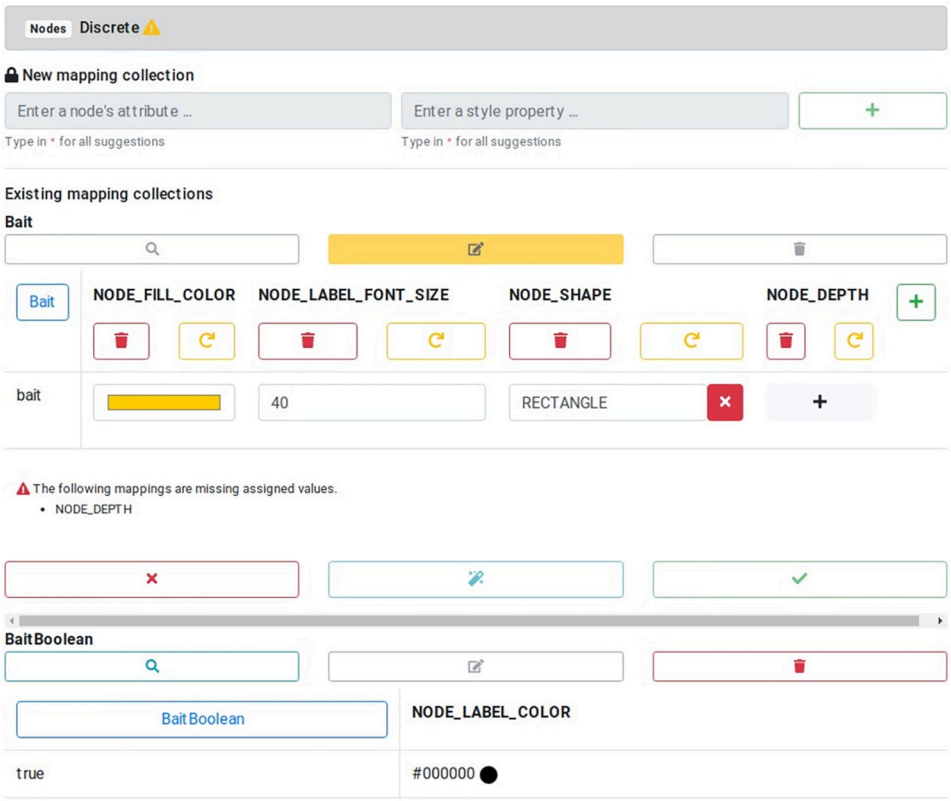


Fig 2. Discrete mapping for node properties. New discrete mappings can be created, existing mappings are shown for the "Bait" property of several visual properties. This includes mappings to colors, numerical and concrete string values.

https://doi.org/10.1371/journal.pcbi.1010205.g002

Continuous mapping. Defining a discrete mapping for continuous values would be tedious since for every value occurring in the attribute a corresponding value for the visual property would be needed. Continuous mappings relieve one from this burden by defining a function on which basis the values for the visual properties are generated. This function is simply characterized by thresholds for the attribute values with corresponding values for the visual property. All values between two thresholds are then mapped linearly in-between.

Continuous mappings can be defined in NDExEdit similarly as discrete mappings, only that the thresholds have to be defined first. Fig 3 shows the continuous mapping of the "diff_score" attribute to two visual properties of the edges. Although several thresholds are defined,

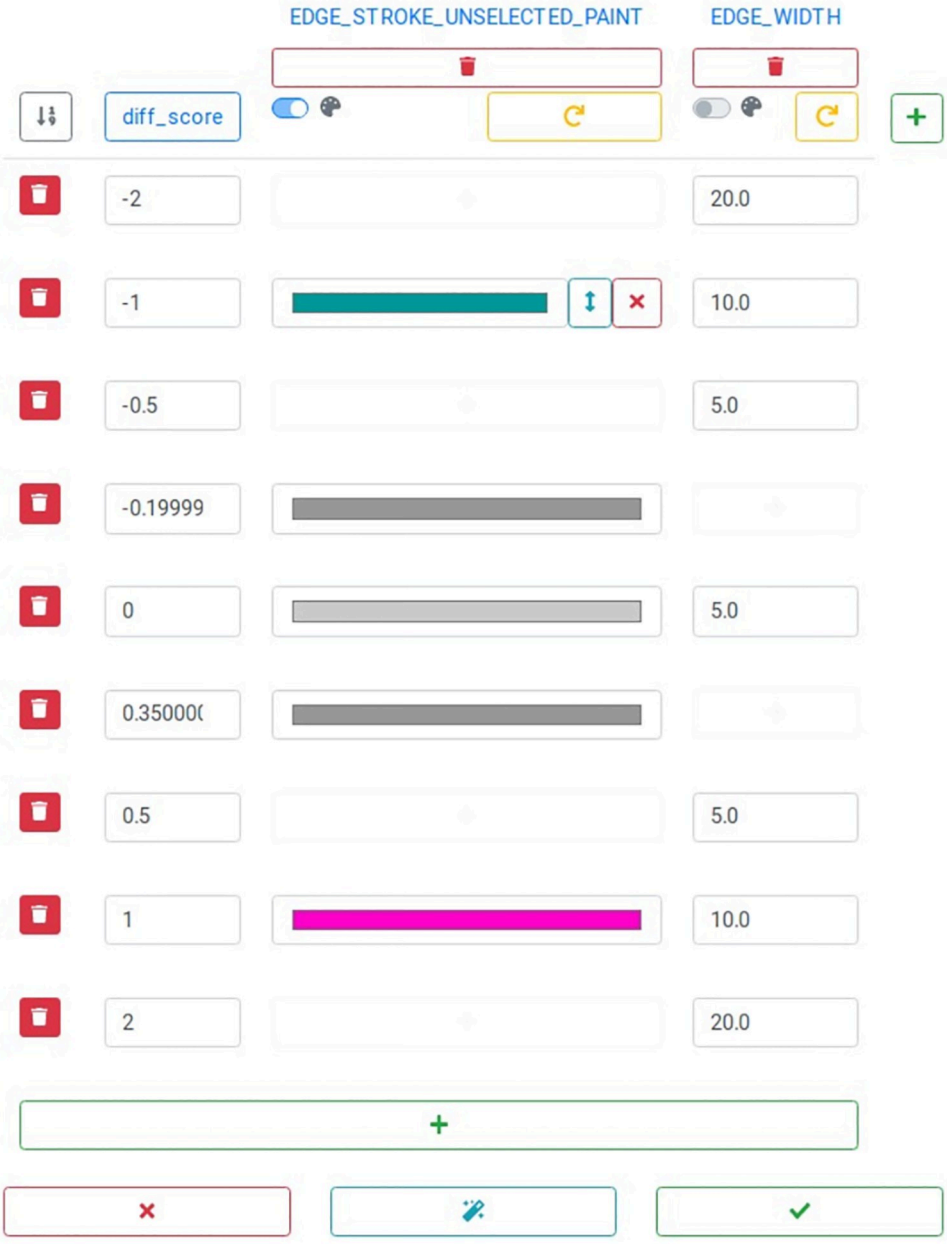


Fig 3. Edit continuous mappings. The score values of the edges are mapped using boundary values, to which colors and numeric values can be assigned. Mapping properties and boundaries can be deleted or new ones added.

https://doi.org/10.1371/journal.pcbi.1010205.g003

the visual properties may not specify all for every visual property. New mapped values can be added using a gray plus button, which appears on moving the cursor over a blank field. Existing ones can not only be deleted but also moved within the visual property by the double-sided arrow next to it.

New thresholds can be added with the green plus button at the bottom. This will lead to the new value being attached at the end of the list, therefore the thresholds can be sorted by value. The single thresholds, and corresponding mapping values, can be deleted by the trash bin button next to it. The addition and removal of the visual properties work as for discrete mappings.

To facilitate the definition of continuous mappings for an attribute, a histogram of the contained data is displayed along with the editing form, as shown in Fig 4A) for the "diff_score" attribute. It can be seen, that the values lie in the range of -1 and +1. The bin size can be adjusted to get a better overview of the data. This histogram is also shown when the creation of the mapping is finished. Additionally, the different visual properties can be selected to display the resulting mapping. For mappings to colors this shows the corresponding color gradient with marked thresholds (Fig 4B), while for numerical values a graph of the mapping function is displayed (Fig 4C).

Pass-through mapping. Pass-through mappings, as the name suggests, only pass the values of a property through to the mapping attribute. A relatable example is the labels of nodes that are displayed along. Although this mapping could be used to set other mapping properties, such as the node size, this way, in most cases it would be more appropriate to create a continuous mapping, which grants more flexibility afterward.

Default properties. Mappings can only be created based on the data, which limits the visual representation of the network to the available data. Furthermore, general visual features need to be defined, like the background color of the network. For nodes, edges, and networks those properties can be set there, and then are consequently used as default values to decorate the networks. They also serve as a fallback when nodes and edges are not covered by the data used for the mappings.

Graph layout

Cytoscape saves the coordinates of the nodes within the network in a dedicated aspect. However, this aspect is only optional, and even not all nodes must have coordinates provided. NDExEdit provides a variety of layout algorithms (Fig 5) to apply to a network, each with a special focus on the networks:

- **random**: nodes are distributed randomly across the viewport which enables to roughly explore the network and its content
- **grid**: nodes are arranged in a grid sorted by the node ids, which puts focus on the nodes
- **circular**: nodes are arranged in a circle so that the focus lies on the edges between the nodes
- **concentric**: nodes are arranged in concentric circles which is a more dense representation than the circular layout
- **hierarchical**: breadth-first arrangement of the network illustrates the topology of the network
- **force-driven**: cose (Compound Spring Embedder) layout [15] uses a physics simulation to determine node distances and produces a more dense representation of the network topology
- **preset**: initial layout saved within the network allows its restoration

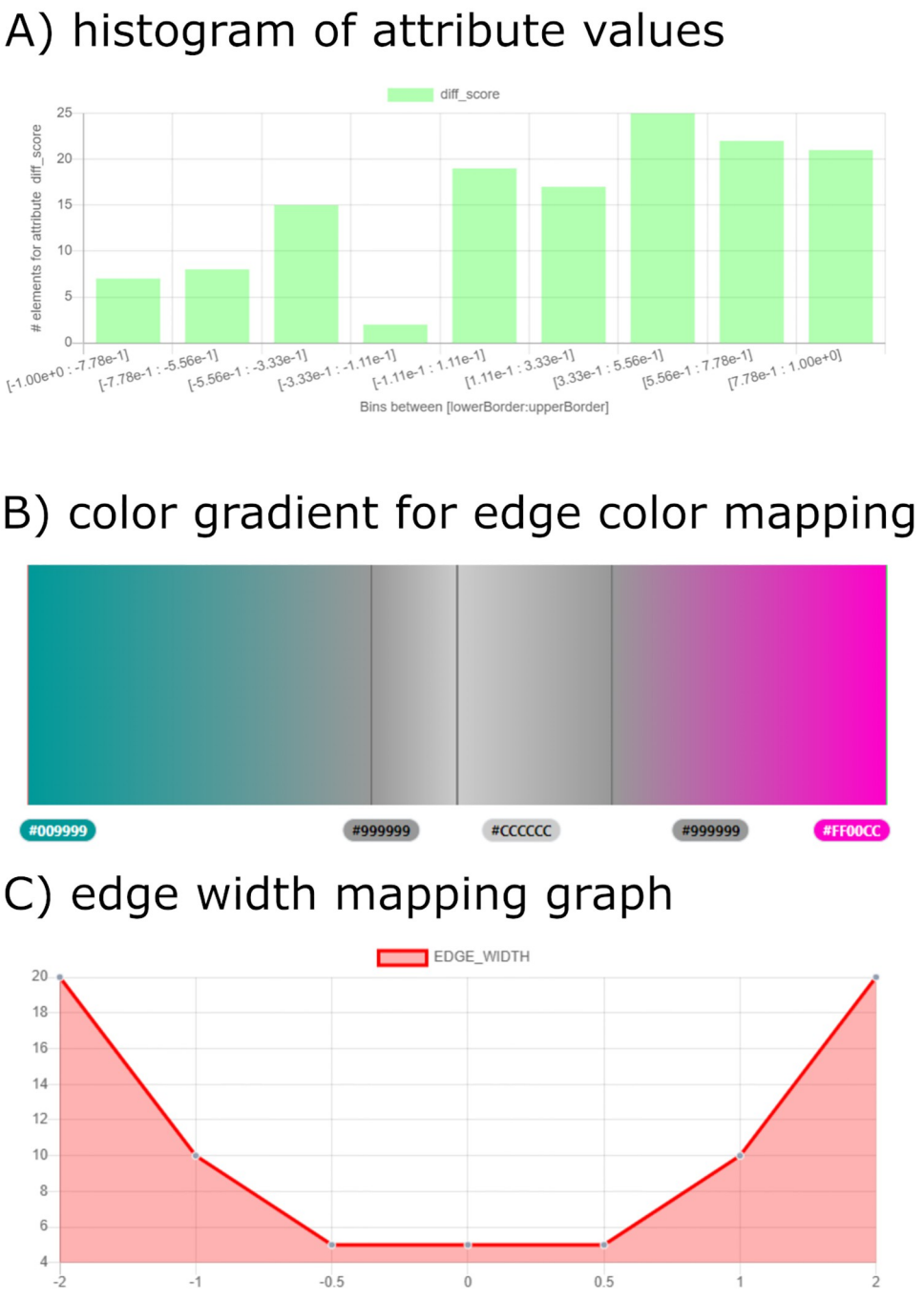


Fig 4. Continuous mapping of values of the edge attribute "diff_score". A) Histogram for the "diff_score" attribute values. B) Continuous mapping of the values to a color gradient with marked boundary values. C) Mapping graph for "diff_score" values to edge width.

https://doi.org/10.1371/journal.pcbi.1010205.g004

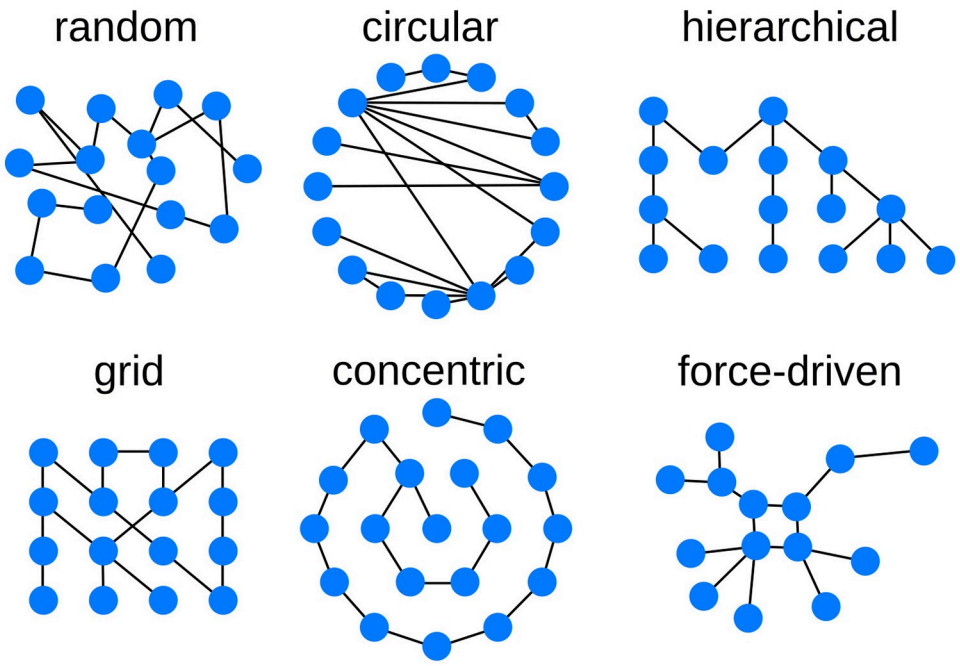


Fig 5. Available graph layout options. Different layout algorithms determine the position of each node, and therefore the overall representation of the network.

https://doi.org/10.1371/journal.pcbi.1010205.g005

The final network layout, including manual refinements, is saved in the cartesian-layout aspect of the exported CX file and is therefore available for the subsequent usage of the network.

Export

NDExEdit provides an option to export the modified networks, including their layout, and visual properties, and mappings as a compatible CX file. With provided credentials the networks can be directly exported to the NDEx platform, either creating a new network or updating an existing one. Also, the network can be exported as a compatible CX file that can be used by other applications.

Additionally, images in standard formats like PNG and JPEG can be created including a scaling factor to produce more detailed versions than a simple screen capture would allow. Also, the exported image can be set to only capture the viewport, or limited in its dimensions. For images in PNG format, it is also possible to change the background color or leave it transparent.

Differentiation to Cytoscape

Cytoscape not only is a software tool for the visualization of networks, but moreover, it is a platform for data integration and analysis, supported by many third-party plugins. The focus of NDExEdit lies instead on the quick and simple visualization of networks based on the contained data. After an analysis workflow, the networks typically contain all the integrated information, and NDExEdit enables to explore its distribution and apply data-dependent mappings to create different visualizations.

Before mentioned workflows are often performed by processing, analyzing, and integrating the data in different tools, or programming languages like R or Python. Especially in the latter

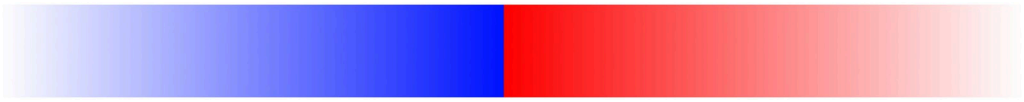


Fig 6. Threshold mapping. A rapid change of the color at the threshold is only hardly possible with the current mapping types.

https://doi.org/10.1371/journal.pcbi.1010205.g006

visualizing the networks is tedious to perform programmatically. NDExEdit, therefore, offers a lightweight interface to generate visualizations. Furthermore, with the NDEx platform as a repository for the networks, collaborators can contribute and refine the final layout simply in the web browser.

Like many software, Cytoscape needs to be installed on local machines, which either requires the administrative rights of the user or has to be managed by the administrator of the institution, along with its software dependencies. This always causes security risks and vulnerabilities, if not handled carefully. An alternative provides web-based solutions, which also can be managed in a centralized manner. NDExEdit runs only in the web browser, without the need for any backend for data processing. This simplifies maintenance of private installations, which are indispensable within systems with limited internet access.

Future directions

While inspecting several public networks missing features for mappings in general appeared: currently there is no elegant way of defining a mapping, that changes the color at a threshold (Fig 6). Currently, networks resemble this feature by defining a continuous mapping with two close, or even identical values as thresholds. The latter implicates further issues in the validation of the mapping.

On NDExEdit the specified mappings apply to the whole network, while it would be useful to restrict the mapping to certain sub-networks. Consequently, different mappings could be defined in general and switched on demand by the user. In the CX-format, as well as Cytoscape there already exists a possibility to manage different mappings for sub-networks and views. However, adaption on NDExEdit would require drastic adjustments to the used library for mapping the CX-format to Cytoscape.js.

Taking the idea of managing different mappings even further, would be the possibility to import existing mappings from other networks. This is possible in general, simply by manually editing the CX file and switching the "cyVisualProperties" aspect, but to be able to do it within NDExEdit would further improve the application. This also can be extended to an option to apply predefined visualization templates, such as SBGN [16], STRING [17], or Reactome [18, 19] layouts to a network.

While NDExEdit is intended to be a web application to easily change the visualization of the network dependent on the data, occasionally it would be beneficial to create additional data. For example, if the node degree is not provided as a property, it must be created with other tools to be available for mappings. More general, importing additional attributes from tabular data, or even the option to create whole networks from it can further decrease the barrier to create data-dependent visualizations of network data.

Author Contributions

Conceptualization: Florian Auer, Frank Kramer.

Funding acquisition: Frank Kramer.

Methodology: Florian Auer, Simone Mayer.

Project administration: Frank Kramer.

Software: Florian Auer, Simone Mayer.

Supervision: Frank Kramer.

Validation: Florian Auer.

Visualization: Florian Auer, Simone Mayer.

Writing – original draft: Florian Auer.

Writing – review & editing: Frank Kramer.

References

1. Califano A, Butte AJ, Friend S, Ideker T, Schadt E. Leveraging models of cell regulation and GWAS data in integrative network-based association studies. Nature Genetics. 2012; 44(8):841–847. https://doi.org/10.1038/ng.2355 PMID: 22836096
2. Chung SS, Pandini A, Annibale A, Coolen ACC, Thomas NSB, Fraternali F. Bridging topological and functional information in protein interaction networks by short loops profiling. Scientific Reports. 2015; 5:8540. https://doi.org/10.1038/srep08540 PMID: 25703051
3. Oulas A, Minadakis G, Zachariou M, Sokratous K, Bourdakou MM, Spyrou GM. Systems Bioinformatics: increasing precision of computational diagnostics and therapeutics through network-based approaches. Briefings in Bioinformatics. 2017.
4. Gligorijević V, Pržulj N. Methods for biological data integration: perspectives and challenges. Journal of The Royal Society Interface. 2015; 12(112):20150571. https://doi.org/10.1098/rsif.2015.0571 PMID: 26490630
5. Pavlopoulos GA, Malliarakis D, Papanikolaou N, Theodosiou T, Enright AJ, Iliopoulos I. Visualizing genome and systems biology: technologies, tools, implementation techniques and trends, past, present and future. GigaScience. 2015; 4(1):1–27. https://doi.org/10.1186/s13742-015-0077-2
6. 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 Research. 2003; 13 (11):2498–2504. https://doi.org/10.1101/gr.1239303 PMID: 14597658
7. Pratt D, Chen J, Welker D, Rivas R, Pillich R, Rynkov V, et al. NDEx, the Network Data Exchange. Cell Systems. 2015; 1(4):302–305. https://doi.org/10.1016/j.cels.2015.10.001 PMID: 26594663
8. Pillich RT, Chen J, Rynkov V, Welker D, Pratt D. NDEx: A Community Resource for Sharing and Publishing of Biological Networks. Methods in Molecular Biology (Clifton, NJ). 2017; 1558:271–301. https://doi.org/10.1007/978-1-4939-6783-4_13 PMID: 28150243
9. Consortium TC. NDEx Network Data Model; 2021. Available from: http://www.home.ndexbio.org/data-model/.
10. 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(2):309–311.
11. Jain N, Mangal P, Mehta D. AngularJS: A Modern MVC Framework in JavaScript. Journal of Global Research in Computer Science. 2014;.
12. Bierman G, Abadi M, Torgersen M. Understanding typescript. In: European Conference on Object-Oriented Programming. Springer; 2014. p. 257–281.
13. Otto M. Bootstrap from Twitter; 2021. Developer Blog. Available from: https://blog.twitter.com/developer/en_us/a/2011/bootstrap-twitter.
14. Kim M, Park J, Bouhaddou M, Kim K, Rojc A, Modak M, et al. A protein interaction landscape of breast cancer. Science. 2021; 374(6563):eabf3066. https://doi.org/10.1126/science.abf3066 PMID: 34591612
15. Dogrusoz U, Giral E, Cetintas A, Civril A, Demir E. A layout algorithm for undirected compound graphs. Information Sciences: an International Journal. 2009; 179(7):980–994. https://doi.org/10.1016/j.ins.2008.11.017
16. Novère NL, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, et al. The Systems Biology Graphical Notation. Nature Biotechnology. 2009; 27(8):735–741. https://doi.org/10.1038/nbt.1558 PMID: 19668183

17. Snel B, Lehmann G, Bork P, Huynen MA. STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. Nucleic Acids Research. 2000; 28(18):3442–3444. https://doi.org/10.1093/nar/28.18.3442 PMID: 10982861

18. Jassal B, Matthews L, Viteri G, Gong C, Lorente P, Fabregat A, et al. The reactome pathway knowledgebase. Nucleic Acids Research. 2020; 48(D1):D498–D503. https://doi.org/10.1093/nar/gkz1031 PMID: 31691815

19. Sidiropoulos K, Viteri G, Sevilla C, Jupe S, Webber M, Orlic-Milacic M, et al. Reactome enhanced pathway visualization. Bioinformatics (Oxford, England). 2017; 33(21):3461–3467. https://doi.org/10.1093/bioinformatics/btx441 PMID: 29077811

Appendix E – MetaRelSubNetVis

Publication

MetaRelSubNetVis: Reference-able network visualizations based on integrated patient data with group-wise comparison

Auer F, Mayer S, Kramer F

bioRxiv 2022.04.18.488628;
doi: https://doi.org/10.1101/2022.04.18.488628

Application Note

Preprint

MetaRelSubNetVis: Referenceable network visualizations based on integrated patient data with group-wise comparison

Florian Auer[1,*], Simone Mayer[1] and Frank Kramer[1]

[1]Department of IT-Infrastructure for Translational Medical Research, Faculty of Applied Computer Science, University of Augsburg, Augsburg, Germany

*To whom correspondence should be addressed.

Abstract

Motivation: Networks are a common data structure to describe relations among biological entities. Enriched with information to specify the entities or their connections, they provide a solid foundation for data-dependent visualization. When such annotations overlap, for example in a protein-protein interaction network that is enriched with patient-specific expressions, visualization is reliant on user interaction. Thereby, effective and reliable exchange of visualization parameters between collaborators is crucial to the communication within workflows.

Results: Here, we introduce MetaRelSubNetVis, a web-based tool that allows users to interactively apply group-wise visualizations to networks augmented with patient data. Our application can visually reflect patient-specific attributes for single patients or in a comparative context. Furthermore, we improved upon the exchange of network visualizations by providing unambiguous links that result in the same visual markup. Our work provides new prospects in interacting with and collaborating on network data, especially with respect to the exchange and integration of network visualizations.

Contact: florian.auer@informatik.uni-augsburg.de

1 Introduction

Networks are a well-established data structure in systems biology and often enhanced with annotations and integrated with additional data for their use in clinical applications (Heo et al., 2021). Visual exploration of these enriched networks is crucial for the interpretability of the contained information. Moreover, comparing different properties of a network, or even different networks based on the same properties remains an ongoing issue. Those networks may be composed of several patient-specific subnetworks based on preceding comparative analysis. An interactive investigation of these networks provides a more direct access to the information in contrast with static visualizations, the main purpose of which is mainly to communicate the results. Especially within a collaborative workflow individual investigations are required to gain necessary insights on the contained data. In turn this exchange of network data again requires specific infrastructure.

One well-established platform, where users can upload a network, share it with selected colleagues or make it publicly available, is the NDEx platform (Pratt et al., 2015). For stored networks, an interface is provided to integrate NDEx related services into third-party applications. One of those is Cytoscape (Shannon et al., 2003), a well-established network analysis and visualization software focusing on biological applications, and enables the exchange of networks with the platform. Cytoscape enables the visual exploration by defining mappings based on attributes of the integrated data but is impractical for quick changes between patient groups and properties. Furthermore, in a collaborative workflow the communication of the used visualization features is determining to the reproducibility and referenceability of the correct network visualization.

With MetaRelSubNetVis, we introduce a tool for the interactive group-wise visualization and comparison of integrated networks. In the following we elaborate on the example of patient-specific subnetworks, how our web-based application can facilitate the investigation of enriched networks in combination with the exchange of referenceable network visualizations.

2 Methods

MetaRelSubNetVis works with networks that are stored on the NDEx platform and can be referenced by its unique UUID and loaded in their proprietary Cytoscape Exchange (CX) format. CX is a JSON based data structure and was specifically designed for the data transmission. It originated from NDEx's close connection to Cytoscape, where CX is used for the exchange of visualized networks.

The NDEx platform can be searched with MetaRelSubNetVis and the graph for the retrieved networks is rendered using the Cytoscape.js (Franz et al., 2016) library. The networks original layout and visualization are discarded during import and instead, a concentric layout is calculated and applied to provide a neutral visual markup. MetaRelSubNetVis is built upon Angular, an open-source framework for building single-page web-applications. The user interface was designed using Bootstrap, a well-established CSS-framework that provides a large set of front-end building blocks.

A CX network is composed of multiple aspects, each of which relates to a specific property of the network, for instance, nodes and edges, and accompanying attributes, meta information, layouts, and visual styles. Integrated data and annotations are conventionally found in the *nodeAttributes* and *edgeAttributes* aspects, while the description of the origin and composition of the data is contained within the *networkAttributes* aspect.

MetaRelSubNetVis requires detailed information about the enclosed integrated data to be able to use it for the creation of the data-dependent visualization and corresponding selectable options. This crucial information about patient data is stored within the *networkAttributes* aspect and involves their pseudonyms, group and subgroup affiliation, and for the exemplary network also patient survival details.

The web-application allows node coloring, sizing, and filtering based on the integrated data stored in the *nodeAttributes* for network wide and group wise properties. This can be for example the number of occurrences of relevant genes across all patients and the relevance scores for the genes in one specific patient, respectively. MetaRelSubNetVis relies on the definition of these visualization options within a for this purpose created non-standard aspect of the same name.

The formal requirements of the *metaRelSubNetVis* aspect are specified on the website, and additional scripts and documentation is provided as extension to the RCX library (Auer & Kramer, 2022) for the creation and handling within the statistical programming language R (R Development Core Team, 2008). The aspect allows the definition of continuous, discrete, and boolean mappings with their thresholds and corresponding color values. Furthermore, the included sample network hosted on the NDEx platform (UUID a420aaee-4be9-11ec-b3be-0ac135e8bacf) contains an implementation of the aspect and can provide guidance.

The sample network resulted from the analysis of a breast cancer data set for metastasis prediction and generation of patient-specific subnetworks by Chereda et al. Thereby, the complete protein-protein interaction network from the Human Protein Reference Database (HPRD) (Keshava Prasad et al., 2009) was used together with a large breast cancer dataset (Bayerlová et al., 2017) to predict for single patients the occurrence of a metastatic event and calculate a gene-wise score of its relevance for the prediction. The 140 most relevant genes of each patient were used to induce subnetworks, which were further combined to a single network and integrated with the gene expression values, levels, relevance scores, and subsequent Molecular Tumor Board report (MTB) analysis (Perera-Bel et al., 2018). MetaRelSubNetVis was then used for the investigation and visualization of the combined network within the original publication.

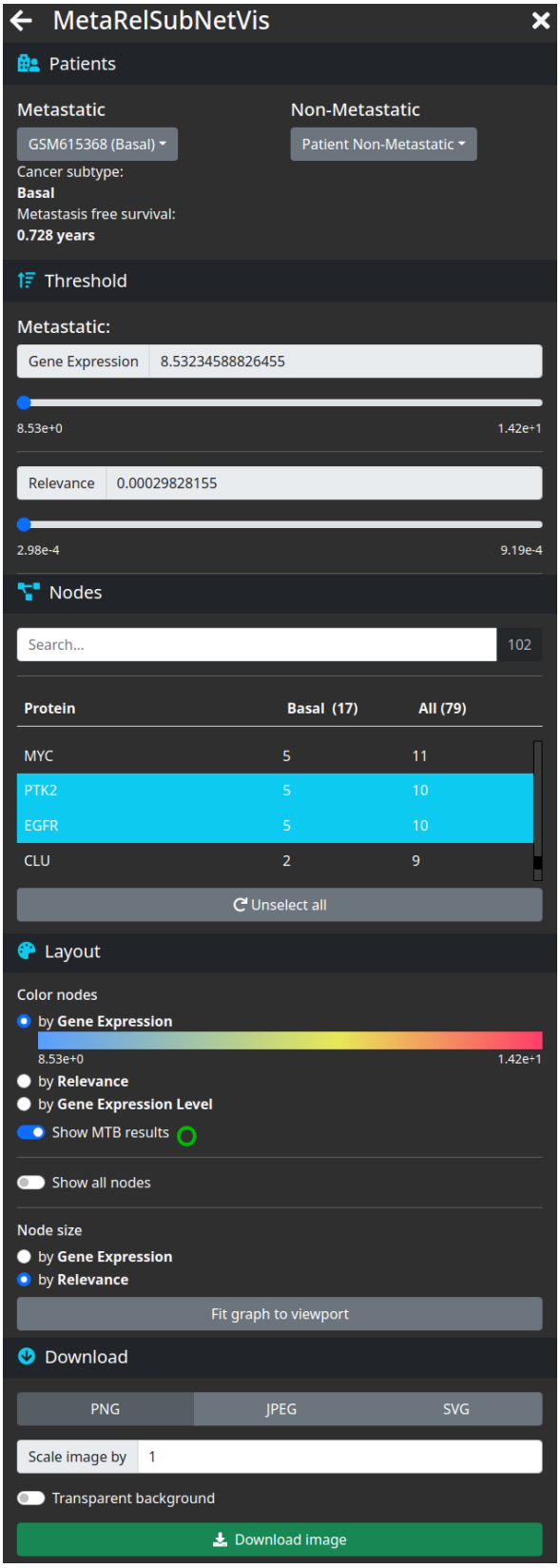


Fig. 1 Selection and visualization options in MetaRelSubNetVis. For a selected patient the threshold can be adjusted, specific nodes selected, coloring and size of nodes chosen, and the visualization exported.

3 Results

3.1 Group-wise selection and network visualization

On the main page of MetaRelSubNetVis the user can select a network by searching the NDEx platform or continue with the provided sample network. The selected network is rendered with the default concentric layout and can be explored interactively by re-arranging its nodes. The position of the nodes will be kept even when the patient selection changes, or different layout setting are applied to facilitate the visual comparison of the subnetworks.

In the sidebar the user can adjust multiple settings (Fig. 1), while group-wise selection is one of the key aspects for this application. In the following the visualization options and its settings will be illustrated based on the provided sample network. In the patient dropdowns the user can select one sample per group for which the rendered network is updated with the corresponding subnetwork. Simultaneous selection of two patients leads to a comparative visualization in which the nodes are split and display the values according to the side of the group (Fig. 3).

With a selected patient, the user can now adjust the thresholds of nodes to show. Adjusting this setting will hide nodes with lower values for those attributes. The nodes setting contains a list of all currently visible nodes and is searchable and interactive: Selecting nodes within the list will mark the respective node within the network. When one or two patients are selected, the list is augmented with information about the selected patient's cancer subtype and occurrence of the genes.

The layout tab allows users to apply the in the *metaRelSubNetVis* aspect defined visualization options for the network. They can choose one of the predefined properties like gene expression level, gene expression or relevance score, to modify the coloring of the nodes. If only one patient is selected, they can adjust the size of each node with defined continuous mappings. Boolean properties as the MTB results allow to highlight corresponding nodes with a colored border.
MetaRelSubNetVis offers options to export the visualized network as image in three available data formats, namely PNG, JPEG and SVG. The image can be scaled with a factor up to 10 for non-SVG images and allows setting a transparent background for the PNG export.

3.2 Sharable visualization link

One significant aspect of MetaRelSubNetVis is the ability to quickly share network visualizations via a custom URL (Fig. 3). In the link generator tab users can highly customize the view, they want to share (Fig. 2). The table at the beginning provides a summary of the previously defined visualization options, such as the UUID of the network, selected patients, defined threshold, and marked and highlighted nodes.

There may not be the need to inspect a network in the browser but rather continue working directly with an image of the network. In that case the user can decide to use the generated link to immediately trigger an image download in the specified format.

The rendering behavior of a network within MetaRelSubNetVis also includes customization of the sidebar. Each tab's visibility, including the back button, and even the visibility of the whole sidebar can be defined in different stages. That opens exceptional possibilities, such as integrating a specific visualization of the network within an iFrame on a different web page. Hiding sidebar components that are not relevant or even the whole sidebar can be highly beneficial to direct the user's focus on a particular aspect of the network.

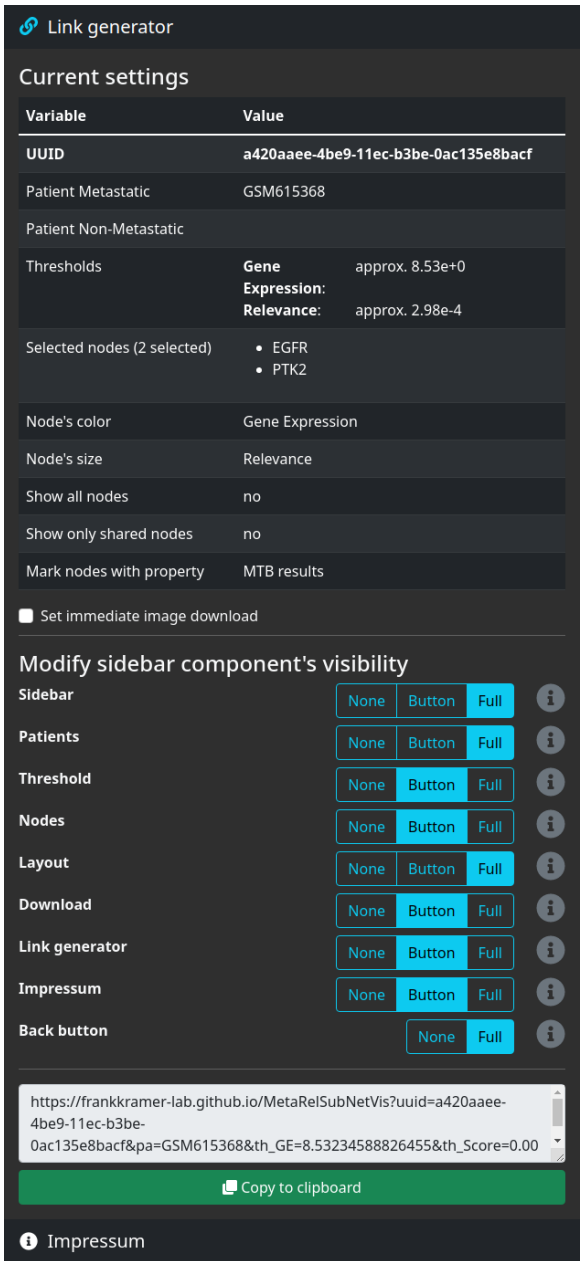


Fig. 2 Options for creating a sharable link. Current settings for the visualization are listed and the behavior of the sidebar and its elements can be adjusted.

4 Discussion

MetaRelSubNetVis was designed with focus on the visualization of a group-wise comparison between the integrated data. Yet, this approach is limited to two groups and for more than this it would also be difficult to be represented in an easily comprehensible manner. Additionally, the visualization is limited to the visualization of the same property within both groups. A potential extension could also be a comparative visualization of two properties for one selected patient. However, this would tremendously increase the dependency between visualization options and the required definitions for the visualization properties that it would be incompatible with the aim for simplicity of this application.

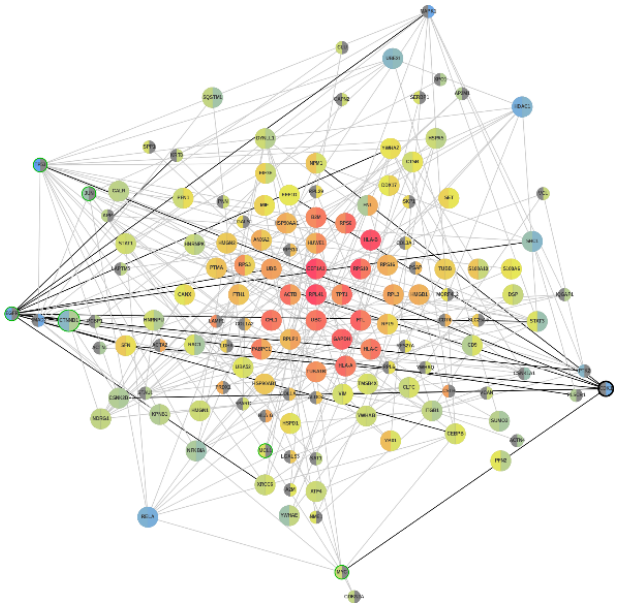


Fig. 3 Visualization of gene expression in metastatic patients GSM615368 and GSM50093. The network visualization by MetaRelSubNetVis is available at https://frankkramer-lab.github.io/MetaRelSubNetVis?uuid=a420aaee-4be9-11ec-b3be-0ac135e8bacf&pa=GSM615368&pb=GSM50093&th_GE=8.53234588826455&th_Score=0.00029828155&sel=2319,3406&col=GE&size=Score&all=false&shared=false&bool=MTB&sb=0&cP=0&cT=1&cN=1&cL=0&cD=1&cG=1&cIm=1&bb=true

On a related note, MetaRelSubNetVis only considers varying attributes relating to nodes. Patient-specific differences in edges are not yet respected and would be a challenge to render, especially for comparative visualizations. Splitting a node visually to show the expressions for the two respective patients is intuitively comprehensible, while a split or duplicated edge is hardly attributable to the corresponding node and may be perceived as confusing.

Providing information about the different groups and the data used for the integration of the network, as well as the definition of the visualization properties might be an obstacle to some users. However, the inclusion of meta-information should be a generally applied principle in network biology. Since the integration step already requires specialized knowledge, the effort required for the definition of the presumed visualization properties is negligible.

5 Conclusion

MetaRelSubNetVis is a web application that allows users to load a network enriched with group-specific information, inspect it and finally export or share the network. Retrieving the networks directly from NDEx not only promotes collaborative workflows through this platform but also circumvents the problems of finding individual hosting solutions for the used networks or incompatibilities due to different data formats.

Throughout the user's visualization efforts, the positions of the single nodes remain consistent and thus improve comparability of the different enclosed properties of an integrated network. The group-wise comparison of network attributes allows a more comprehensible investigation of the results of preceding, already comparative analyses.

The communication and exchange of network visualizations is simplified with MetaRelSubNetVis by sharing a link to a specific layout configuration, facilitating collaboration furthermore. The options to hide specific parts of the sidebar or even the sidebar in total proves to be invaluable when embedding a network's visualization within other web applications: Developers can highly customize the view without the implementation of an own proprietary network visualization.

MetaRelSubNetVis has already proven its potential by its application to the results of Chereda et al., where it was successfully used for the exploration, interpretation and visualization of the created patient-specific subnetworks.

6 Availability

A live version is hosted on GitHub Pages at https://frankkramer-lab.github.io/MetaRelSubNetVis and the corresponding source code for deploying own instances is provided on https://github.com/frankkramer-lab/MetaRelSubNetVis.

7 Funding

This work is a part of the Multipath project funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) grant FKZ01ZX1508.

Conflict of Interest: none declared.

8 References

Auer, F., & Kramer, F. (2022). RCX – an R package adapting the Cytoscape Exchange format for biological networks. *Bioinformatics Advances*, vbac020. https://doi.org/10.1093/bioadv/vbac020

Bayerlová, M., Menck, K., Klemm, F., Wolff, A., Pukrop, T., Binder, C., Beißbarth, T., & Bleckmann, A. (2017). Ror2 Signaling and Its Relevance in Breast Cancer Progression. *Frontiers in Oncology*, *7*. https://www.frontiersin.org/article/10.3389/fonc.2017.00135

Chereda, H., Bleckmann, A., Menck, K., Perera-Bel, J., Stegmaier, P., Auer, F., Kramer, F., Leha, A., & Beißbarth, T. (2021). Explaining decisions of graph convolutional neural networks: Patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. *Genome Medicine*, *13*(1), 42. https://doi.org/10.1186/s13073-021-00845-7

Franz, M., Lopes, C. T., Huck, G., Dong, Y., Sumer, O., & Bader, G. D. (2016). Cytoscape.js: A graph theory library for visualisation and analysis. *Bioinformatics*, *32*(2), 309–311. https://doi.org/10.1093/bioinformatics/btv557

Heo, Y. J., Hwa, C., Lee, G.-H., Park, J.-M., & An, J.-Y. (2021). Integrative Multi-Omics Approaches in Cancer Research: From Biological Networks to Clinical Subtypes. *Molecules and Cells*, *44*(7), 433–443. https://doi.org/10.14348/molcells.2021.0042

Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., Balakrishnan, L., Marimuthu, A., Banerjee, S., Somanathan, D. S., Sebastian, A., Rani, S., Ray, S., Harrys Kishore, C. J., Kanth, S., … Pandey, A. (2009). Human Protein Reference Database—2009 update. *Nucleic Acids Research*, *37*(suppl_1), D767–D772. https://doi.org/10.1093/nar/gkn892

Perera-Bel, J., Hutter, B., Heining, C., Bleckmann, A., Fröhlich, M., Fröhling, S., Glimm, H., Brors, B., & Beißbarth, T. (2018). From somatic variants towards precision oncology: Evidence-driven reporting of treatment options in molecular tumor boards. *Genome Medicine*, *10*(1), 18. https://doi.org/10.1186/s13073-018-0529-2

Pratt, D., Chen, J., Welker, D., Rivas, R., Pillich, R., Rynkov, V., Ono, K., Miello, C., Hicks, L., Szalma, S., Stojmirovic, A., Dobrin, R., Braxenthaler, M., Kuentzer, J., Demchak, B., & Ideker, T. (2015). NDEx, the Network Data Exchange. *Cell Systems*, *1*(4), 302–305. https://doi.org/10.1016/j.cels.2015.10.001

R Development Core Team. (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. http://www.R-project.org

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, *13*(11), 2498–2504. https://doi.org/10.1101/gr.1239303

Appendix F – Reproducible Data Integration

Publication

Reproducible data integration and visualization of biological networks in R

Auer F, Chereda H, Perera-Bel J, Kramer F

bioRxiv 2022.04.15.488519
doi: https://doi.org/10.1101/2022.04.15.488519

Preprint

Reproducible data integration and visualization of biological networks in R

Florian Auer[1,*], Hryhorii Chereda[2], Júlia Perera-Bel[3] and Frank Kramer[1]

[1]Department of IT-Infrastructure for Translational Medical Research, Faculty of Applied Computer Science, University of Augsburg, Augsburg, Germany

[2]Department of Medical Bioinformatics, University Medical Center Göttingen, Göttingen, Germany

[3]Department of Medical Oncology, IMIM-Hospital del Mar Medical Research Institute, Hospital del Mar, Barcelona, Spain

*To whom correspondence should be addressed.

Abstract

Motivation: Collaborative workflows in network biology not only require the documentation of the performed analysis steps but also of the network data on which the decisions were based. However, replication of the entire workflow or tracking of the intermediate networks used for a particular visualization remains an intricate task. Also, the amount and heterogeneity of the integrated data requires instruments to explore and thus comprehend the results.

Results: Here we demonstrate a collection of software tools and libraries for network data integration, exploration, and visualization to document the different stages of the workflow. The integrative steps are performed in R, and the entire process is accompanied by an interchangeable toolset for data exploration and network visualization.

Availability: The source code of the performed workflow is available as R markdown scripts at https://github.com/frankkramer-lab/reproducible-network-visualization. A compiled HTML version is also hosted on Github pages at https://frankkramer-lab.github.io/reproducible-network-visualization.

Contact: florian.auer@informatik.uni-augsburg.de

1 Introduction

Scientific research faces the problem of the reproducibility of the methods used in its various domains, with the effect that the reported results could not be reconstructed (Goodman et al., 2016). Using a steadily growing number of tools and working in multidisciplinary teams increasingly complicates replication, and necessitates to report not only the results but rather the entire workflow (Committee, 2021).

Analyses performed in network biology are no exception and require networks and their visualizations not to be seen only as input or result of the process. Networks are gradually enriched with additional data from various sources and the attention is mainly set on the applied methods. Intermediate networks are omitted although those could be utilized to document the progress and illuminate the contained information.

Furthermore, the visualizations of the integrated networks in any stage face the problem of choosing the appropriate attributes to focus on. Especially in a collaborative process this hampers the communication of relevant aspects in different steps of the workflow. Also, visual representations are subjective to the creator and may hide important features crucial for the understanding for collaborators and subsequent decisions for progression. An interactive exploration of the enriched networks can help to expose otherwise invisible characteristics but requires a seamless integration into the process.

Here we present a collection of tools to construct a reproducible workflow for network data integration and visualization, including the documentation of intermediate and final results. We demonstrate the workflow on previous results for the generation of patient-specific subnetworks for a large breast cancer data set (Chereda et al., 2021). Thereby we point out several options of software tools for the visualiza-

tion that can be used individually or in combination to foster the reproducibility of the workflow.

2 Methods

2.1 Gene expression and molecular networks data

2.1.1 Breast cancer data set

The studied breast cancer data set is composed from 10 microarray data sets publicly available at the Gene Expression Omnibus (GEO) (Barrett et al., 2013) repository by the accession numbers GSE25066, GSE20685, GSE19615, GSE17907, GSE16446, GSE17705, GSE2603, GSE11121, GSE7390, GSE6532. The expression data was measured on Affymetrix Human Genome HG-U133 Plus 2.0 and HG-U133A arrays and previously preprocessed and studied (Bayerlová et al., 2017), including the prediction of the molecular subtypes for the Breast cancer samples. Sample selection, filtering, combination and normalization was performed according to previous work (Chereda et al., 2021) and resulted in a data set containing 12,179 genes in 969 patients. The patients` metastatic status were derived from the occurrence of distant metastasis within the first 5 years (393 patients) or absence with the last follow-up between 5 and 10 years (576 patients).

2.1.2 Protein interaction networks

The protein-protein interaction (PPI) network from the Human Protein Reference Database (HPRD) (Keshava Prasad et al., 2009; Mishra et al., 2006; Peri et al., 2003) was used as basis for capturing the relations between the expressed genes. The information contained in molecular network is based on evidence from in vitro and in vivo yeast two-hybrid analyses and constitutes of undirected binary interactions between pairs of proteins.

The disconnected graph consists of 9,898 vertices which decreased after mapping the genes of the breast cancer data set onto the PPI network to 7,168 vertices in 207 connected components. For further analysis only the main connected component was used, which consisted of 6,888 vertices, while the remaining components only contained 1 to 4 vertices. The main reason for this choice was that the Graph-CNN algorithm requires a connected graph as input.

2.2 Data processing

2.2.1 Relevance score

The computation of a patient-specific relevance score is a two-step process: Firstly, a Graph Convolutional Neural Network (Graph-CNN) is trained on the gene expression and molecular network data to predict the metastatic status for a patient. Secondly, the Graph Layer-wise Relevance propagation (GLRP) algorithm is applied to a patient's prediction to determine the relevance of the genes to the predictive outcome. (Chereda et al., 2021)

The Graph-CNNs were trained on the gene expression dataset with the HPRD PPI network as prior knowledge with a 10-fold cross validation over a whole dataset to estimate the predictive performance of Graph-CNN. For the generation of the relevance scores, the gene expression dataset was randomly split in training (90%) and test (10%) set. The Graph-CNN was trained using manually selected hyperparameters from 10-fold cross validation, and subsequently used to predict metastatic events for the test set consisting of 97 patients.

For the subsequent analysis, meaning the generation of patient-specific subnetworks, only 79 patients were considered with matching predicted and reported metastasis. The GLRP method was applied for those patients to determine the relevance of the genes to the prediction, thus called relevance score.

2.2.2 Molecular tumor board report analysis

Actionable genes present in the patient-specific subnetworks were identified using the Molecular Tumor Board (MTB) report (referred to as "MTB report") methodology described in Perera-Bel et al. Therefore, the algorithm was extended by inferring gain of function alterations from high expression, and loss of function alterations from low expression respectively. The gene expression levels were derived based on the gene expression throughout the whole patient cohort with the 25% and 75% quantiles as boundaries for low, normal and high levels of expression.

Although information about specific gene variants is not present in the breast cancer gene expression data set due to the used quantification method, the results can be used to define specific panels for subsequent sequencing.

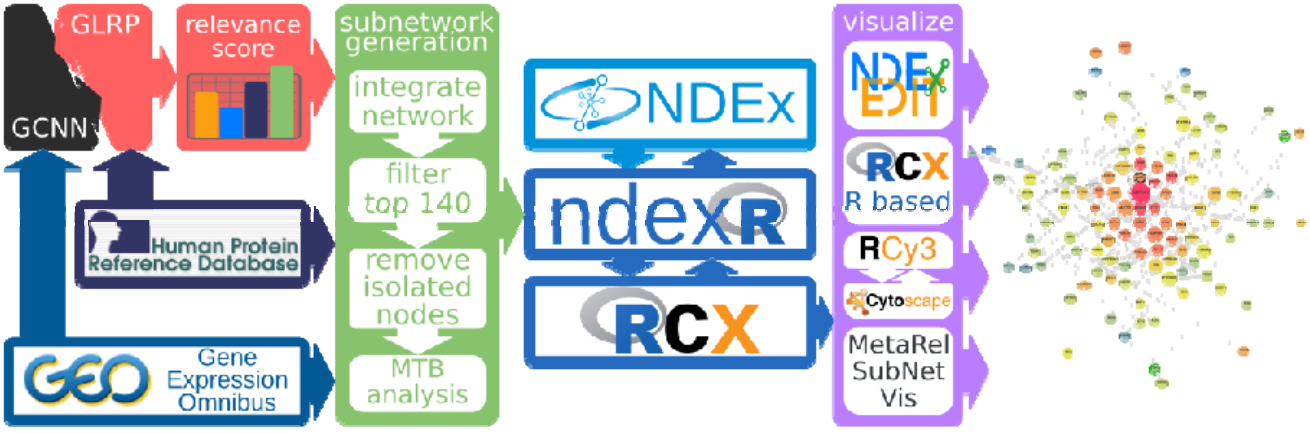


Fig. 1 Network data integration and visualization workflow. HPRD and GEO serve as raw data resource for the generation of patient-specific subnetworks and used for calculation of the relevance scores. The networks are stored on the NDEx platform using *ndexr* and handled by *RCX* which was also used for the visualization, additionally to *NDExEdit*, *MetaRelSubNetVis*, and RCy3 and Cytoscape.

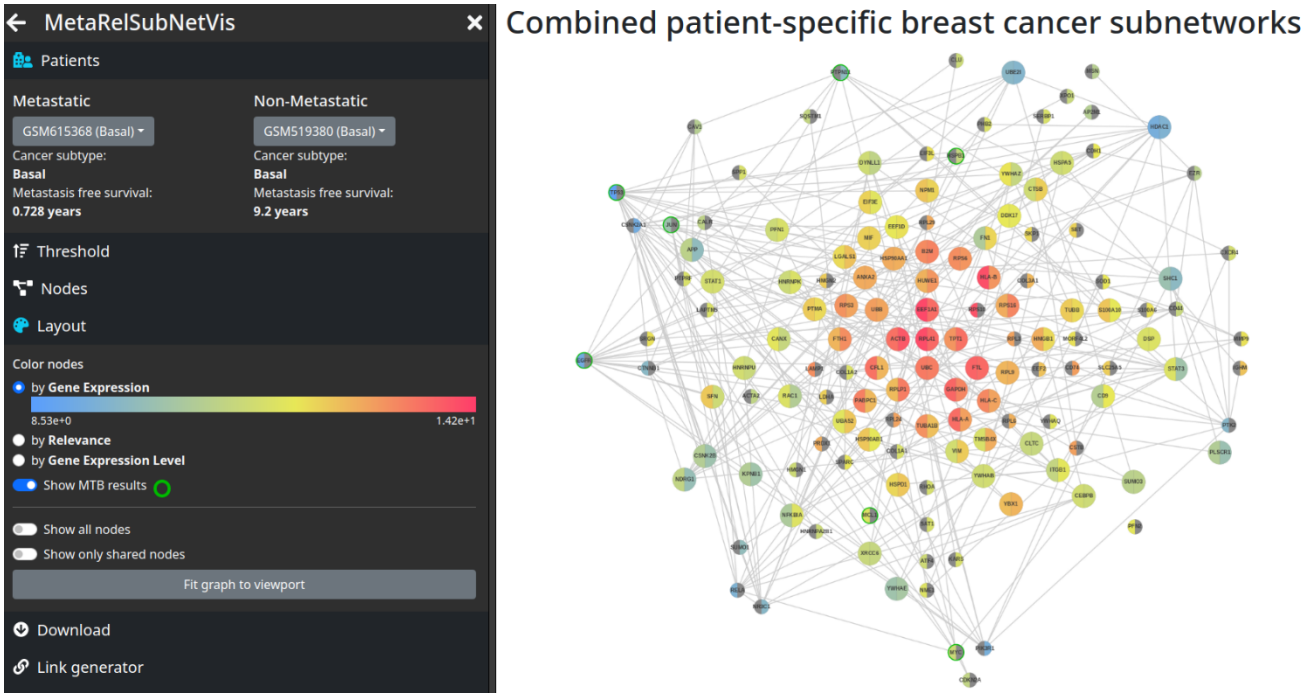


Fig. 2 Comparative visualization of patient GSM615368 and GSM615184 on MetaRelSubNetVis colored by gene expression. The visualization network is available at https://frankkramer-lab.github.io/MetaRelSubNetVis?uuid=a420aaee-4be9-11ec-b3be-0ac135e8bacf&pa=GSM615368&pb=GSM519380&th_GE=8.53234588826455&th_Score=0.00029828155&col=GE& size=GE&all=false&shared=false&bool=MTB&sb=0&cP=0&cT=1&cN=1&cL=0&cD=1&cG=1&cIm=1&bb=true

2.3 Data integration and modeling

2.3.1 The NDEx platform

The distribution of biological networks is an important aspect in collaborative workflows. Working on the same data basis can be challenging to be arranged, be it providing the networks used as resource for the initial analysis or sharing intermediate and final results.

The Network Data Exchange (NDEx) (Pillich et al., 2017; Pratt et al., 2015) is an online commons specifically designed for the exchange of and collaboration on biological networks. Networks can be uploaded and shared with individual persons or groups and remain visible only to this peer to prevent premature disclosure of this valuable information.

For the publication of the results, NDEx then can be used to propagate the network data supplementary to manuscripts, and as possible resources for further analyses. Furthermore, a comprehensible collection of networks is publicly available on the platform as for example the NCI Pathway Interaction Database (PID) (Schaefer et al., 2009) from which NDEx initially originated.

2.3.2 ndexr

Programmatic interaction with the NDEx platform from within R (R Development Core Team, 2008) is provided by the *ndexr* package (Auer et al., 2018). The package enables the search of public and private networks on the platform as well as the exchange of networks with the platform. Moreover, it provides functions to adjust the accessibility and visibility of the networks as well as options for sharing with specific users and groups.

In this work we use *ndexr* to document the progress of the integrative network analysis. The HPRD PPI network available at the NDEx platform was retrieved, and the intermediate stages from integration of the gene expression with the network to the creation of the single patient-specific subnetworks are saved using this package.

2.3.3 RCX

The NDEx platform uses for the exchange of the network data their proprietary Cytoscape Exchange (CX) data format. It is an aspect oriented and JSON based data structure tailored to the transmission of biological networks. It utilized established web standards for the transmission and thereby encapsulates the different components of the network (i.e., nodes, edges, layout and visual representation, and associated attributes) into separated modules (aspects). The different aspects are independent by itself but can refer to each other, if necessary, for example edges refer to the nodes they connect, and the cartesian layout to the nodes to which they assign the position.

CX originated from the cooperation with the Cytoscape consortium and consequently inherited aspects dedicated to capture the visual representation of the network. Moreover, one remarkable feature of CX in contrast to other network formats is that the visual representation is a part of the network itself.

The *RCX* package (Auer & Kramer, 2022) includes functions and models to facilitate working with biological networks in CX format within R. The RCX data model, including separate models for the single aspects, is thereby the adaptation of CX to standard R data types and structures. Due to the fundamental differences between the table-based view on data in R and the object-oriented composition in JSON, and hence CX, the *RCX* package offers specialized functions for conversion and handling of the networks. Besides the lossless conversion to the CX format also *igraph* (Csardi & Nepusz, 2006) and Bioconductor *graph* (Gentleman et al., 2021) are supported, both established libraries for graph analysis and visualization. Furthermore, *RCX* includes functions for the creation, modification and validation of networks in this format to facilitate usability.

2.3.4 Generation of patient-specific subnetworks

The *ndexr* package uses *RCX* as data model for the integration of the gene expression values and levels, and relevance scores with the HPRD PPI network to generate the subnetworks, as well as to capture and store the visualizations (Fig. 1). The resulting integrated network is available on the NDEx (UUID 833b1cee-42f6-11ec-b3be-0ac135e8bacf) and forms the basis for the subnetwork generation.

For each of the 79 patients the 140 most relevant genes were used to induce the intermediate patient-specific subnetworks. Those subnetworks then were combined and again only the main connected component consisting of 407 nodes used. For these final patient-specific subnetworks the MTB report analysis was applied and the results integrated into the combined network (UUID a420aaee-4be9-11ec-b3be-0ac135e8bacf), which then was used as basis for the different visualization approaches. The patient-specific subnetworks are also individually accessible at the NDEx platform within a network collection (UUID 5d308fbb-42da-11ec-b3be-0ac135e8bacf).

Since the single integration steps build upon each other it is necessary to track the preceding networks for reproducibility of the integration steps. Therefore, the source networks are listed by their UUID within the network attributes of the current network model.

3 Results

3.1 MetaRelSubNetVis

MetaRelSubNetVis is a web-based tool for the interactive exploration of integrated patient subnetworks and comparison of those networks between patient groups. The combined integrated subnetwork is directly loaded from NDEx and visualized within the web-browser with a concentric layout applied by default. The visual representation of the subnetworks can be adjusted to base on the integrated data. The included genes can be sized and colored using a gradient for expression or relevance score values. Alternatively, different colors for the expression levels can be set. Additionally, the results of the MTB analysis can be highlighted within the graph as well as selected nodes.

The different patient groups, i.e., metastatic and non-metastatic, can be visualized and compared within the same graph. For a selected patient of each group the nodes are split and colored by the value of the patient of the corresponding group (Fig. 3). Shared genes are sized greater than individual ones and to put even more emphasis on common nodes only those can be shown.

The visualization can be explored interactively by moving nodes or adjusting the thresholds for relevance scores or gene expression to display the nodes. Thereby, the position of the nodes is preserved between visualization and patient selection changes to facilitate identification of the same entities across the different subnetworks.

Furthermore, custom links can be generated that lead directly to the selected patients and their visualization. These links can be used to communicate and reference specific findings within the subnetworks. They can also be provided along publication for illustration and be embedded on websites for reference or interactive exploration (Fig. 2).

3.2 NDExEdit

The visualization of networks, even only simple ones, often requires additional software to be installed on the local machine. *NDExEdit* is a web-based approach for the data-dependent visualization of networks where those can be loaded directly from the NDEx platform. For a load-

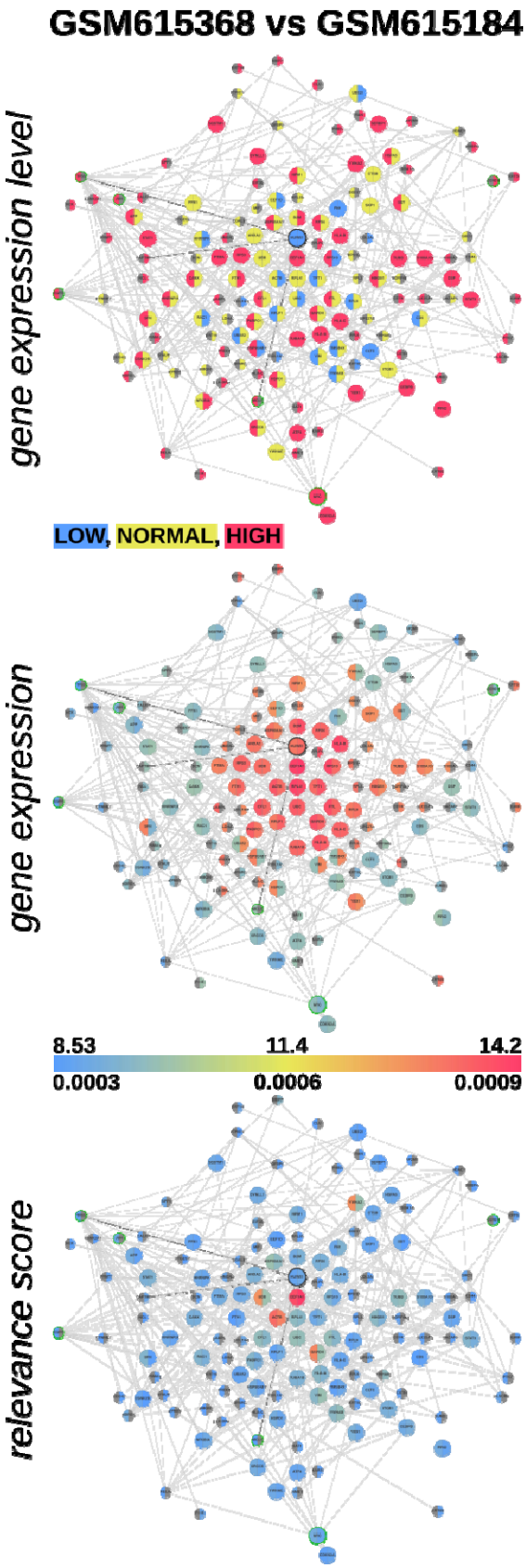


Fig. 3 Comparative visualization of patient GSM615368 and GSM615184 colored by gene expression level and value, and relevance score.

ed network the single attributes, and their distribution, can be explored and visual attributes applied based on the contained data. The network can be arranged, different layouts applied, and the results saved within the network. This can be used to highlight for example the number of occurrences of the different genes across the combined network (Fig. 4), or to define the same visual styles for a single patient as demonstrated with *MetaRelSubNetVis*. Finally, the resulting networks with included visualization can directly exported to the NDEx platform, downloaded as CX file, or exported as publication-ready images in various formats.

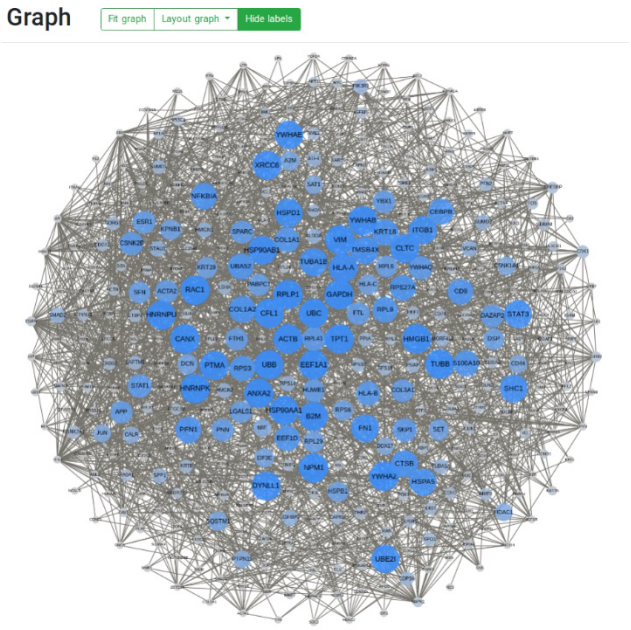


Fig. 4 Combined patient-specific subnetworks. The nodes are colored and sized by the number of occurence across the 79 patients. The network is available on the NDEx platform by the UUID a420aaee-4be9-11ec-b3be-0ac135e8bacf.

3.3 R based visualization

3.3.1 igraph

Within the R environment the most prominent software package for graph manipulation, analysis, and visualization is the *igraph* library. The package thereby follows the typical R methodology of defining the visual attributes for each node and edge individually and explicitly before plotting the graph. This way similar visualizations can be created as with the above tools, but the attribute values for single nodes (and edges) must be set individually. Also, the sharing and exploration of networks in this format is rather inconvenient and especially the latter requires expertise with appropriate tools.

3.3.2 Cytoscape and RCy3

One of the most widely used software tools for the visualization of biological networks is Cytoscape (Shannon et al., 2003). It allows the import of networks in various formats, including CX from file or directly from the NDEx platform. Besides the simple definition of visual properties Cytoscape offers plenty of tools for network analysis, that can be extended even further by custom plugins.

In contrast to the *igraph* package where individual values are assigned to the nodes and edges, Cytoscape defines mappings based on attributes of those. This not only allows a more generalized definition of the visual properties, but also promotes the reuse of the created visual styles for different networks.

Cytoscape provides a REST API which can be used with the R package *RCy3* (Gustavsen et al., 2019) to access the software in a programmatic manner. This allows to remotely control Cytoscape to reproduce the visualization of the patient-specific subnetworks with the same representation as in NDExEdit. Since both tools allow the export of the visualized networks to the NDEx platform the visualizations can be continued or refined in both tools interchangeably.

3.3.3 RCX

The *RCX* package was not only used for handling of the network data and integration steps, but also to define and apply layouts and visual attributes. Therefore, the package includes functions to produce visualizations of the networks consistent with those on the NDEx platform, on NDExEdit and within Cytoscape. This consistency is based on the aspect-oriented structure of the RCX data model which includes the properties for the visual representation. We show how the aspect for the visual representation can be created from scratch, including the necessary properties, mappings and dependencies for the nodes and edges.

However, for users unfamiliar with the Cytoscape visual properties this approach is arduous. Therefore, we demonstrate a simpler strategy by reusing the visualization created with *NDExEdit* for a single patient. The visual properties of the downloaded network are adjusted for the remaining patients for mapping the corresponding patient data. Subsequently the patient-specific subnetworks including their visualizations are exported the NDEx platform.

4 Discussion

When working on data integration with biological networks in R the most straight forward approach is to use the most established *igraph* library, especially if it requires methods for graph and network analysis. However, visualization and distribution of the integrated network models is rather limited. Tools like Cytoscape simplify the visualization process of the created networks, but still require a solution for network import and distribution.

The NDEx platform offers a solution for management and collaboration and is already integrated into Cytoscape. Together with the *RCy3* package they provide an option to load, visualize and store and share the networks. Nevertheless, this may constitute an unnecessary detour, especially if the visualization is performed by another party. The usage of the *RCX* package for network integration, or through conversion from the *igraph* models provides in combination with *ndexr* a shortcut to the NDEx platform.

The RCX package itself can be used for the visualization of networks but instead of manually defining the visual properties its greater benefit lies in the reuse and adjustment of those. Again, Cytoscape can be used for the creation of visual properties but NDExEdit on the contrary does not require installation of the software. Furthermore, it provides options to explore the contained data and adjust the visual mappings accordingly which supports especially users unfamiliar with the network. This though comes with the cost of NDExEDIT relying on the NDEx platform due to missing support of import options for other network formats.

Although the NDEx platform promotes sharing and referencing the resulting networks it still has its shortcomings in terms of the representa-

tion of integrated network data. *MetaRelSubNetVis* offers an addition to NDEx by allowing to interactively explore the networks with a consistent network structure, a property-based visualization, and a group-wise comparison. Furthermore, the specific visualization settings can be shared additionally to the network. However, this requires the information about the visualization parameters to be included within the network.

5 Conclusion

Here we presented different approaches for the reproducible network data integration and visualization. The presented tools constitute of established software and libraries each with its own advantages and use-cases. They mainly evolve around the NDEx platform which enables storage and distribution of results of an analysis and allows the documentation of the performed steps. Together with the inclusion of the visualization within the networks it not only contributes to the comprehensibility of the results but also fosters their reproducibility.

The application for the generation of patient-specific subnetworks illustrates its applicability in a typical bioinformatics workflow. The proposed solutions are not exclusive but rather complementary to established methods and demonstrate their benefits especially through flexibility in their usage. The visualization of intermediated network results brings additional insights to the performed integration steps. Only the combination of the here discussed software tools, platforms and packages promotes an environment for the reproducibility network data integration and accompanying visualization.

Funding

This work is a part of the Multipath project funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) grant FKZ01ZX1508.

Conflict of Interest: none declared.

References

Auer, F., Hammoud, Z., Ishkin, A., Pratt, D., Ideker, T., & Kramer, F. (2018). ndexr—An R package to interface with the network data exchange. Bioinformatics, 34(4), 716–717. https://doi.org/10.1093/bioinformatics/btx683

Auer, F., & Kramer, F. (2022). RCX – an R package adapting the Cytoscape Exchange format for biological networks. Bioinformatics Advances, vbac020. https://doi.org/10.1093/bioadv/vbac020

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., Yefanov, A., Lee, H., Zhang, N., Robertson, C. L., Serova, N., Davis, S., & Soboleva, A. (2013). NCBI GEO: Archive for functional genomics data sets—update. Nucleic Acids Research, 41(D1), D991–D995. https://doi.org/10.1093/nar/gks1193

Bayerlová, M., Menck, K., Klemm, F., Wolff, A., Pukrop, T., Binder, C., Beißbarth, T., & Bleckmann, A. (2017). Ror2 Signaling and Its Relevance in Breast Cancer Progression. Frontiers in Oncology, 7. https://www.frontiersin.org/article/10.3389/fonc.2017.00135

Chereda, H., Bleckmann, A., Menck, K., Perera-Bel, J., Stegmaier, P., Auer, F., Kramer, F., Leha, A., & Beißbarth, T. (2021). Explaining decisions of graph convolutional neural networks: Patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. Genome Medicine, 13(1), 42. https://doi.org/10.1186/s13073-021-00845-7

Committee, U. R. N. S. (2021). From grassroots to global: A blueprint for building a reproducibility network. PLOS Biology, 19(11), e3001461. https://doi.org/10.1371/journal.pbio.3001461

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695.

Gentleman, R., Whalen, E., Huber, W., & Falcon, S. (2021). Graph: A package to handle graph data structures. R package version 1.70.0.

Goodman, S. N., Fanelli, D., & Ioannidis, J. P. A. (2016). What does research reproducibility mean? Science Translational Medicine, 8(341), 341ps12-341ps12. https://doi.org/10.1126/scitranslmed.aaf5027

Gustavsen, J. A., Pai, S., Isserlin, R., Demchak, B., & Pico, A. R. (2019). RCy3: Network biology using Cytoscape from within R. F1000Research, 8, 1774. https://doi.org/10.12688/f1000research.20887.3

Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., Balakrishnan, L., Marimuthu, A., Banerjee, S., Somanathan, D. S., Sebastian, A., Rani, S., Ray, S., Harrys Kishore, C. J., Kanth, S., … Pandey, A. (2009). Human Protein Reference Database—2009 update. Nucleic Acids Research, 37(suppl_1), D767–D772. https://doi.org/10.1093/nar/gkn892

Mishra, G. R., Suresh, M., Kumaran, K., Kannabiran, N., Suresh, S., Bala, P., Shivakumar, K., Anuradha, N., Reddy, R., Raghavan, T. M., Menon, S., Hanumanthu, G., Gupta, M., Upendran, S., Gupta, S., Mahesh, M., Jacob, B., Mathew, P., Chatterjee, P., … Pandey, A. (2006). Human protein reference database—2006 update. Nucleic Acids Research, 34(suppl_1), D411–D414. https://doi.org/10.1093/nar/gkj141

Perera-Bel, J., Hutter, B., Heining, C., Bleckmann, A., Fröhlich, M., Fröhling, S., Glimm, H., Brors, B., & Beißbarth, T. (2018). From somatic variants towards precision oncology: Evidence-driven reporting of treatment options in molecular tumor boards. Genome Medicine, 10(1), 18. https://doi.org/10.1186/s13073-018-0529-2

Peri, S., Navarro, J. D., Amanchy, R., Kristiansen, T. Z., Jonnalagadda, C. K., Surendranath, V., Niranjan, V., Muthusamy, B., Gandhi, T. K. B., Gronborg, M., Ibarrola, N., Deshpande, N., Shanker, K., Shivashankar, H. N., Rashmi, B. P., Ramya, M. A., Zhao, Z., Chandrika, K. N., Padma, N., … Pandey, A. (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Research, 13(10), 2363–2371. https://doi.org/10.1101/gr.1680803

Pillich, R. T., Chen, J., Rynkov, V., Welker, D., & Pratt, D. (2017). NDEx: A Community Resource for Sharing and Publishing of Biological Networks. Methods in Molecular Biology (Clifton, N.J.), 1558, 271–301. https://doi.org/10.1007/978-1-4939-6783-4_13

Pratt, D., Chen, J., Welker, D., Rivas, R., Pillich, R., Rynkov, V., Ono, K., Miello, C., Hicks, L., Szalma, S., Stojmirovic, A., Dobrin, R., Braxenthaler, M., Kuentzer, J., Demchak, B., & Ideker, T. (2015). NDEx, the Network Data Exchange. Cell Systems, 1(4), 302–305. https://doi.org/10.1016/j.cels.2015.10.001

R Development Core Team. (2008). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. http://www.R-project.org

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., & Buetow, K. H. (2009). PID: The Pathway Interaction Database. Nucleic Acids Research, 37(Database issue), D674–D679. https://doi.org/10.1093/nar/gkn653

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. Genome Research, 13(11), 2498–2504. https://doi.org/10.1101/gr.1239303

Appendix G – Gene Expression of FHIR

Publication

Adaptation of HL7 FHIR for the exchange of patients' gene expression profiles

Auer F, Abdykalykova Z, Müller D, Frank Kramer F

Stud Health Technol Inform. 2022 Jun 29;295:332-335.
doi: https://doi.org/10.3233/shti220730

Advances in Informatics, Management and Technology in Healthcare
J. Mantas et al. (Eds.)

doi:10.3233/SHTI220730

Adaptation of HL7 FHIR for the Exchange of Patients' Gene Expression Profiles

Florian AUER[a,1], Zhibek ABDYKALYKOVA[a], Dominik MÜLLER[a] and Frank KRAMER[a]
[a] *IT-Infrastructure for Translational Medical Research, University of Augsburg, Germany*

Abstract. High-throughput technologies, especially gene expression analyses can accurately capture the molecular state in patients under different conditions. Thus, their application in clinical routine gains increasing relevance and fosters patient stratification towards individualized treatment decisions. Electronic health records already evolved to capture genomic data within clinical systems and standards like FHIR enable sharing within, and even between institutions. However, FHIR only provides profiles tailored to variations in the molecular sequence, while expression patterns are neglected although being equally important for decision making. Here we provide an exemplary implementation of gene expression profiles of a microarray analysis of patients with acute myeloid leukemia using an adaptation of the FHIR genomics extension. Our results demonstrate how FHIR resources can be facilitated in clinical systems and thereby pave the way for usage for the aggregation and exchange of transcriptomic data in multi-center studies.

Keywords. FHIR, interoperability, omics, gene expression

1. Introduction

Measuring the gene expression in patient samples provides detailed insights into the molecular conditions of the underlying disease. Over the years, high-throughput technologies have evolved to be used in routine clinical diagnostics and foster individualized treatment. At the same time digitalization in healthcare systems advanced to electronic health records (EHR) capturing also genomic data. Interoperability and data sharing between systems and institutions gain more importance with commonly accepted standards like Fast Healthcare Interoperability Resources (FHIR) [1] as a foundation.

FHIR divides the information into modular and extensible components, as well as adapts widely established web standards and the RESTful architecture principle for the sharing of EHRs. Included extensions for genomics data are tailored to cover only variations in the molecular sequence while expression patterns are neglected. Moreover, recommendations for the realization of gene expression results in FHIR are lacking. Nevertheless, these insights are important for decision support and translational research.

Here we provide a feasible FHIR implementation for gene expression profiles from microarray analyses and demonstrate the interoperability of the resulting FHIR resources within an interactive web application.

[1] Corresponding Author: Florian Auer, IT Infrastructure for Translational Medical Research, Alter Postweg 101, 86159 Augsburg, Germany; E-mail: florian.auer@informatik.uni-augsburg.de.

2. Methods

2.1. Gene expression data

The data set examines a dose-limiting side effect in patients diagnosed with acute myeloid leukemia (AML) that are treated with chemotherapy [2]. Mucositis, DNA damage within the oral mucosa caused by the chemotherapy is investigated based on the derived gene expression profiles. The samples are collected from punch buccal biopsies from five AML patients pre- and post-chemotherapy, and three healthy controls for comparison. Microarray analysis was performed using Human Genome U133 Plus 2.0 Array (Affymetrix, Santa Clara, CA) with GRCh38.p13 (Genome Reference Consortium Human Build 38, Ensembl release 99) as a reference, followed by a Robust Multichip Average (RMA) normalization of the raw data. The authors made the data available at the EBI Expression Atlas [3] portal by the ID E-GEOD-10746.

We chose this gene expression data set because the conducted analysis represents a typical bioinformatics workflow resulting in several gene expression profiles from the same and different individuals that enable disease classification and patient stratification into risk groups [4].

2.2. Adaption in FHIR resources

The central element within FHIR to capture real-world concepts is the *Patient* resource: A study evolves around patient treatment therefore all subsequent patient-specific results, and resources implementing those refer to this base element. Detailed information about the sample donors was not included in the original data set to preserve the anonymity of the participants, instead, we used artificially generated data using Synthea™ [5] to create *Patient* resources as reference. The medical condition of the AML patients was captured by the *Condition* resource to distinguish them from the healthy donors. The single samples are captured by the *Specimen* resource and serve as a link to distinguish between samples collected from the same patient, namely pre- and post-chemotherapy.

The gene expression values are generated based on the GRCh38.p13 reference genome and were measured for each sample. Since all gene expression profiles use the same reference, the single genes contained in the reference genome were included as *MolecularSequence* resources. The actual expression values are treated as single measurements realized as *Observation* resources with the *Observation-geneticsGene* extension referring to the corresponding gene symbol. Although the *Patient* resource is referenced directly within the *Observation* resource, the *Specimen* resource is still

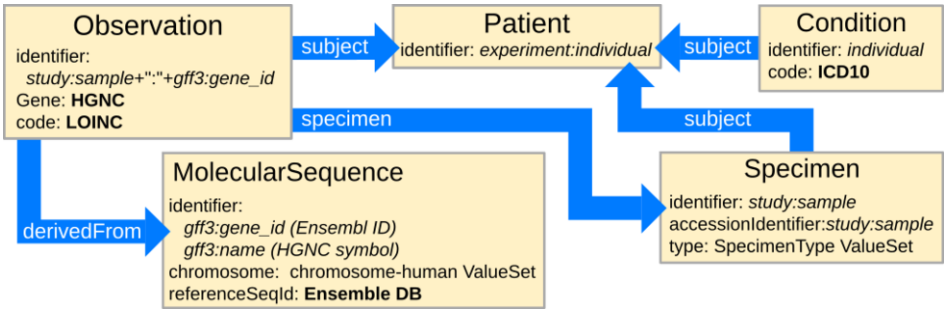


Figure 1. FHIR resources and their references between each other as well as to external databases.

required to differentiate between the different samples of the same patient. Using the *MolecularSequence* resource as a reference avoids redundancy of shared genomic information and simplifies the retrieval of the gene expression values for one particular gene across the different samples. An overview of the resources and their links to public databases, as well as references between the resources is shown in figure 1.

An in-house installation of the dockerized HAPI FHIR server [6] was used for storing the created resources. Additionally, we developed a web application that uses the FHIR REST API to retrieve and display the FHIR resources to demonstrate a minimalistic decision support system.

2.3. Data and material availability

All necessary software to reproduce the results is publicly available on GitHub at https://frankkramer-lab/gene-expression-on-fhir. This includes scripts to download the data sets from the official platforms, set up the dockerized HAPI FHIR server and import the data, host the web application, and corresponding source code (GPL-3.0 License). Additionally, a demonstration of the web application with hard-coded excerpts of the FHIR resource data is hosted as a static service using the GitHub pages functionality which can be accessed at https://frankkramer-lab.github.io/gene-expresssion-on-fhir.

3. Results

The original data translated to 252,684 resources stored on our FHIR server. For performance improvements, not all gene ids in the reference genome (60,617 ensemble entries) were encoded in FHIR but only those present in the gene expression data. A detailed overview of the created resources and the time requirements is shown in table 1.

Table 1. Summary of FHIR resources and time required to upload to the FHIR server.

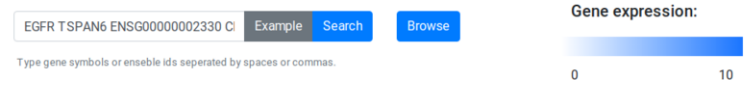
FHIR resource	Number of objects	Time for creation
Patient	8	~1sec
Condition	5	~1sec
Specimen	11	~2sec
MolecularSequence	21,055	~5min
Observation	231,605	~45min

The web application demonstrates the usage of the created resources: Those are obtained directly from the FHIR server, then linked and assembled into a visual representation of the gene expression across the patient samples (figure 2).

4. Discussion

Through our contribution to the FHIR Genomics extension, we were able to include genomic profiling) data. Since only excerpts of the molecular data are necessary for detailed investigation, FHIR encoded gene expression profiles are suitable for usage in web-based applications. Furthermore, we were able to demonstrate the integration capabilities of FHIR encoded genomic profiles in decision support systems. Further improvements could consist of consolidation of the outcome of the analyses, e.g.,

significantly differentially expressed genes between samples, as *DiagnosticReport* resources.



			None			Acute myeloid leukaemia, not elsewhere classified by criteria of other types							
			none			pre-chemotherapy				two days post-chemotherapy			
#	ENSEMBL	Gene symbol	Boyce Greenholt	Darcie Hyatt	Bruce Reichert	Yong Huel	Pedro Jacobs	Carina Sanford	Carmen Kertzmann	Pedro Jacobs	Bryon Ebert	Carina Sanford	Carmen Kertzmann
3487	ENSG00000002330	BAD	6.213432	6.038769	6.160183	6.081077	6.091502	6.156229	6.046596	5.909643	6.302064	6.053074	6.107919
18077	ENSG00000146648	EGFR	8.418005	8.188606	8.826504	9.954438	9.051215	8.806058	9.460211	9.137021	9.00577	8.616357	8.94842
19594	ENSG00000147889	CDKN2A	5.388622	5.252077	5.295755	5.838683	5.558717	6.040631	5.523091	6.105364	6.267255	6.139984	5.473827
19596	ENSG00000147883	CDKN2B	8.815149	8.851738	8.830046	8.779058	8.96846	9.114393	8.661207	8.43376	8.834632	8.771359	9.172349
20505	ENSG00000126752	SSX1	2.28402	2.426147	2.266226	2.28402	2.302603	2.28402	2.28402	2.28402	2.340258	2.331636	2.061351
20717	ENSG00000000003	TSPAN6	9.093082	9.252337	9.42373	9.298296	9.227221	9.482802	9.287413	8.598412	9.287413	9.620169	9.756805

Figure 2. Web application using the created FHIR resources to show the gene expression between the samples of different patients. The heatmap is colored by the corresponding expression intensity.

5. Conclusions

Our results demonstrate how FHIR resources can be facilitated for the clinical exchange of expression profiles. The usage of the adopted resources within our web application demonstrates its feasibility for usage in decision support systems or patient assessment. The further incorporation of genomic features into the FHIR standard offers the opportunity to establish the currently missing standard for the aggregation of various molecular genetics data in a clinical setting. This work contributes to closing this gap and paves the way towards patient stratification through transcriptomic profiling even across health care institutions and within multi-center clinical trials.

References

[1] Bender D, Sartipi K. HL7 FHIR: An agile and RESTful approach to healthcare information exchange. In: Proceedings of CBMS 2013 - 26th IEEE International Symposium on Computer-Based Medical Systems. 2013. p. 326–31.

[2] Mougeot JLC, Bahrani-Mougeot FK, Lockhart PB, Brennan MT. Microarray analyses of oral punch biopsies from acute myeloid leukemia (AML) patients treated with chemotherapy. Oral Surg Oral Med Oral Pathol Oral Radiol Endodontology. 2011 Oct 1;112(4):446–52.

[3] Kapushesky M, Emam I, Holloway E, Kurnosov P, Zorin A, Malone J, Rustici G, Williams E, Parkinson H, Brazma A. Gene expression atlas at the European bioinformatics institute. Nucleic acids research. 2010 Jan 1;38(suppl_1):D690-8.

[4] Döhner H, Estey EH, Amadori S, Appelbaum FR, Büchner T, Burnett AK, Dombret H, Fenaux P, Grimwade D, Larson RA, Lo-Coco F. Diagnosis and management of acute myeloid leukemia in adults: recommendations from an international expert panel, on behalf of the European LeukemiaNet. Blood, The Journal of the American Society of Hematology. 2010 Jan 21;115(3):453-74.

[5] Walonoski J, Kramer M, Nichols J, Quina A, Moesel C, Hall D, Duffett C, Dube K, Gallagher T, McLachlan S. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. Journal of the American Medical Informatics Association. 2018 Mar 1;25(3):230-8.

[6] Sánchez YKR, Demurjian S, Baihan MS. Achieving RBAC on RESTful APIs for Mobile Apps Using FHIR. 2017 5th IEEE Int Conf Mob Cloud Comput Serv Eng MobileCloud. 2017.

Curriculum Vitae

Personal information

Date of birth: April 22nd 1985

Place of birth: Cham, Germany

florian.auer@informatik.uni-augsburg.de
(+49) 0821- 598 3748
linkedin.com/in/florian-auer-09a28983
https://orcid.org/0000-0002-5320-8900

Education

2016 – 2022	Phd Student at the Georg-August-Universität Göttingen
2006 – 2014	Diploma study in Bioinformatics at the Ludwig-Maximilians-Universität and the Technische Universität München Diploma thesis: “Meta Analysis of Larger Scale Protein Expression Profiles” Overall grade: 2,1

Professional Experience

Since November 2018	Research And Teaching Assistant at University of Augsburg Faculty of Applied Computer Science Department of IT Infrastructure for Translational Medical Research
April 2016 – October 2018	Research And Teaching Assistant at University Medical Center Göttingen Department of Medical Statistics/Statistical Bioinformatics
April 2012	Student assistant at Rostlab under the supervision of Prof. Burkhard Rost, Technische Universität München
2003 – 2004	Founder of a student company for the implementation of fire and rescue plans

Achievements

2022	Founding Member “Förderer und Angehörige der Medizininformatik e.V.”
2021	Teaching certificate for Bavarian Universities
2017	e:Med Poster Award of Systems Medicine
2016	Winner of the SBGN design competition at the COMBINE conference

Publications

PLOS Computational Biology 2022, 18 (6), e1010205	Data-dependent visualization of biological networks in the web-browser with NDExEdit *F Auer, S Mayer, F Kramer* https://doi.org/10.1371/journal.pcbi.1010205
Studies in Health Technology and Informatics 2022;295:332–335	Adaptation of HL7 FHIR for the Exchange of Patients' Gene Expression Profiles *F Auer, Z Abdykalykova, D Müller, F Kramer* https://doi.org/10.3233/SHTI220730
Studies in Health Technology and Informatics 2022; 290:912-916	Perspective on Code Submission and Automated Evaluation Platforms for University Teaching *F Auer, J Frei, D Müller, F Kramer* https://doi.org/10.3233/SHTI220212
Studies in Health Technology and Informatics 2022; 294:33-37	MISeval: A Metric Library for Medical Image Segmentation Evaluation *D Müller, D Hartmann, P Meyer, F Auer, J Frei, I Soto-Rey, F Kramer* https://doi.org/10.3233/SHTI220391
Bioinformatics Advances 2022;, vbac020	RCX – an R package adapting the Cytoscape Exchange format for biological networks, Bioinformatics Advances *F Auer, F Kramer* doi: 10.1093/bioadv/vbac020
Frontiers in Pharmacology vol. 12, p. 3393, 2021	Comprehensive Analysis of Chemical Structures That Have Been Tested as CFTR Activating Substances in a Publicly Available Database CandActCFTR *M. M. Nietert, L. Vinhoven, F. Auer, S. Hafkemeyer, and F. Stanke* doi: 10.3389/fphar.2021.689205
Genome medicine **13**, Article number: 42 (2021)	Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer *H Chereda, A Bleckmann, K Menck, J Perera-Bel, P Stegmaier, F Auer et al.* doi: 10.1186/s13073-021-00845-7
Bioinformatics 2017 Oct 26.	ndexr - an R package to interface with the Network Data Exchange. *Auer F, Hammoud Z, Ishkin A, Pratt D, Ideker T, Kramer F.* doi: 10.1093/bioinformatics/btx683
Nature Methods 2013; **10**, 221-227	A large-scale evaluation of computational protein function prediction *Radivojac P, Clark WT, Oron TR et al.* doi: 10.1038/nmeth.2340
BMC Bioinformatics 2013; **14** (Suppl 3):S7	Homology-based inference sets the bar high for protein function prediction *Hamp T, Kassner R, Seemayer S, Vicedo E, Schaefer C, Achten D, Auer F, Boehm A, Braun T, Hecht M, Heron M, Hönigschmid P, Hopf T A, Kaufmann S, Kiening M, Krompass D, Landerer C, Mahlich Y, Roos M, Rost B* doi: 10.1186/1471-2105-14-S3-S7

Cell Reports
2013; Vol. 4, Issue 3 pp. 609-620

Global Proteome Analysis of the NCI-60 Cell Line Panel
Moghaddas Gholami A, Hahne H, Wu Z, Auer FJ, Meng C, Wilhelm M, Kuster B
doi: 10.1016/j.celrep.2013.07.018

Preprints

bioRxiv
April 18, 2022

MetaRelSubNetVis: Referenceable network visualizations based on integrated patient data with group-wise comparison
F Auer, S Mayer, F Kramer
https://doi.org/10.1101/2022.04.18.488628

bioRxiv
April 16, 2022

Reproducible data integration and visualization of biological networks in R
F Auer, H Chereda, J Perera-Bel, F Kramer
doi: 10.1101/2022.04.15.488519

Presentations

Talk
July 2022
online

Adaptation of HL7 FHIR for the exchange of patients' gene expression profiles
International Conference on Informatics, Management and Technology in Healthcare (ICIMTH)

Poster
May 2022
Nice, France

Implementation of gene expression profiles in the HL7 FHIR standard
Medical Informatics Europe (MIE)

Talk
October 2021
online

Perspective on Code Submission and Automated Evaluation Platforms for University Teaching
MedInfo

Poster
September 2021
online

NDExEdit: A web tool for biological network visualization
GMDS & TMF Jahrestagung

Talk & Poster
November 2020
Bonn, online

Bringing signaling pathway knowledge into clinical systems using the HL7 FHIR standard
e:Med Kick-off Meeting

Talk
October 2019
Erlangen

A knowledge base for generating patient-specific pathways for individualized treatment decisions in clinical applications
GMDS Doktorandensymposium

Poster
September 2019
Dortmund

Using signaling pathway knowledge in a hospital setting: Extending the FHIR standard for health care data exchange
GMDS Jahrestagung

Poster
July 2019
Basel, Switzerland

The RCX data model: An R adaptation of the Cytoscape exchange format for biological networks
Intelligent Systems for Molecular Biology & European Conference on Computational Biology (ISMB/ECCB)

Presentations

Talk
October 2018
Heidelberg

A knowledge base for generating patient-specific pathways for individualized treatment decisions in clinical applications
GMDS Doktorandensymposium

Poster
September 2018
Vienna, Austria

Bringing Pathway Knowledge to Systems Medicine Approaches
German Conference on Bioinformatics (GCB)

Talk & Poster
September 2018
Osnabrück

Integration of biological networks into healthcare systems with FHIRgraph
GMDS Jahrestagung

Talk & Poster
September 2018
Osnabrück

NDExR and Cytoscape: interactive and automated visualization of biological networks using R
GMDS Jahrestagung

Session chair & Talk & Poster
July 2018
Chicago, USA

Composing a dockerized Ecosystem for the Exchange and Visualization of Biological Networks
Intelligent Systems for Molecular Biology (ISMB)

Talk
March 2018
Regensburg

NDExR and Cytoscape: Interactive and automated visualization of biological networks using R
Workshop on Computational Models in Biology and Medicine

Poster
November 2017
Göttingen

ndexr - an R package to interface with the network data exchange
e:Med Kick-off Meeting

Talk
October 2017
Braunschweig

A knowledge base for generating patient-specific pathways for individualized treatment decisions in clinical applications
GMDS Doktorandensymposium

Poster
September 2017
Tübingen

Composing a dockerized Ecosystem for the Exchange and Visualization of Biological Networks
German Conference on Bioinformatics (GCB)

Poster
July 2017
Prague, Czech Republic

Composing a dockerized Ecosystem for the Exchange and Visualization of Biological Networks
Intelligent Systems for Molecular Biology & European Conference on Computational Biology (ISMB/ECCB)

Poster
September 2016
Newcastle, UK

Bringing Pathway Knowledge to Systems Medicine Approaches
Combine

Poster
September 2016
Berlin

Bringing Pathway Knowledge to Systems Medicine Approaches
German Conference on Bioinformatics (GCB)