cyTRON and cyTRON/JS: Two Cytoscape-Based Applications for the Inference of Cancer Evolution Models

Lucrezia Patruno[1], Edoardo Galimberti[2,3], Daniele Ramazzotti[4,8], Giulio Caravagna[5], Luca De Sano[1], Marco Antoniotti[1,7(✉)], and Alex Graudenzi[1,6,7(✉)]

[1] Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, Italy
{marco.antoniotti,alex.graudenzi}@unimib.it
[2] Department of Computer Science, University of Turin, Turin, Italy
[3] ISI Foundation, Turin, Italy
[4] Department of Pathology, Stanford University, Stanford, CA, USA
[5] The Institute of Cancer Research - ICR, London, UK
[6] Institute of Molecular Bioimaging and Physiology - IBFM of the National Research Council - CNR, Segrate, Milan, Italy
[7] Bicocca Bioinformatics Biostatistics and Bioimaging (B4) Center, University of Milano-Bicocca, Milan, Italy
[8] School of Medicine and Surgery, Università degli Studi di Milano-Bicocca, Monza, Italy

Abstract. The increasing availability of sequencing data of cancer samples is fueling the development of algorithmic strategies to investigate tumor heterogeneity and infer reliable models of cancer evolution. We here build up on previous works on cancer progression inference from genomic alteration data, to deliver two distinct `Cytoscape`-based applications, which allow to produce, visualize and manipulate cancer evolution models, also by interacting with public genomic and proteomics databases. In particular, we here introduce `cyTRON`, a standalone `Cytoscape` app, and `cyTRON/JS`, a web application which employs the functionalities of `Cytoscape/JS`.

`cyTRON` was developed in Java. `cyTRON/JS` was developed in JavaScript and R.

Keywords: Cancer · Cancer evolution · Mutational graphs · Tumor phylogenies · Cytoscape

1 Scientific Background

Cancer is a complex disease, whose development is caused by the accumulation of alterations in the genome. Some alterations may confer a selective advantage

E. Galimberti and L. Patruno—Equal contributors.

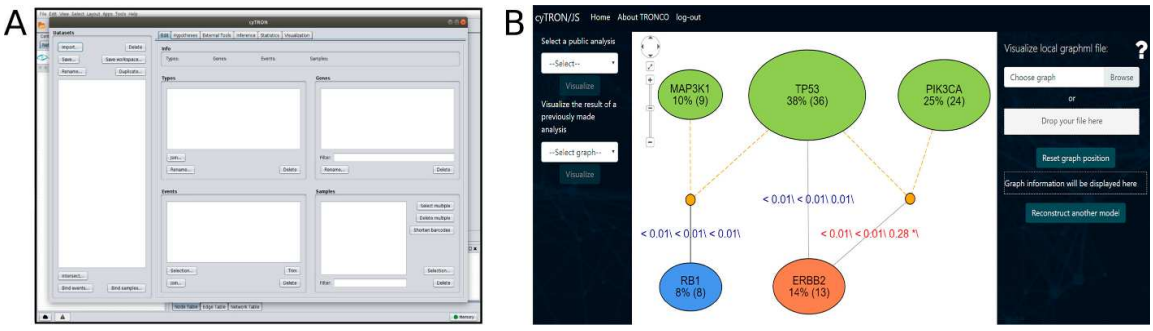

Fig. 1. We show in the figure a view of `cyTRON` workspace (A) and an example of output model by `cyTRON/JS` (B).

to cancer cells, and this may result in the expansion of cancer clones. In order to understand how cancer evolves, it is of great importance to understand how such *driver* alterations accumulate over time [5]. This goal can be pursued by reconstructing cancer evolution models, which are graphs that encode the evolutionary history of drivers and their temporal relationships. The reconstruction of such models is a complex task mainly because of two reasons: first, much of the data publicly available from initiatives such as `TCGA` [https://portal.gdc.cancer.gov/] comes from cross-sectional samples, and hence they lack of temporal information. The second main reason can be found in the heterogeneity of tumors [2,7].

2 Materials and Methods

In order to produce meaningful evolution models, we developed a pipeline, `PICNIC` [2], which includes the following steps: *i*) identification of homogeneous sample subgroups (e.g., tumor subtypes), *ii*) identification of drivers (i.e., the nodes of the output model), *iii*) identification of mutually exclusive patterns, *iv*) inference of cancer evolution models via distinct algorithms (e.g., [4,6,8]). These algorithms perform the inference of probabilistic graphical models of cancer progression in two steps; first, the assessment of statistical associations between a set of genomic variants is performed and a partially order set (poset) among them is reconstructed by considering their frequencies, i.e., variants at higher frequency can precede variants at lower frequency, and their statistical dependency, i.e., arcs are allowed only among positively dependent variants. Then, a maximum likelihood directed acyclic graph [6] or a maximum score tree is inferred within such poset [4,8].

The `PICNIC` pipeline was implemented within the `TRONCO R` suite for TRanslational ONCOlogy [1,3], which was recently employed, for instance, to analyze the largest kidney cancer cohort currently available [10].

However, `TRONCO` presents two practical limitations: first, it requires at least some basic programming skills due to its underlying `R` infrastructure; second, `TRONCO` is not integrated with publicly available genomic databases, hence providing a non-interactive visualization of the output graphs.

Therefore, to improve the practicality, effectiveness, interactivity and diffusion of our framework, we integrated it within Cytoscape, a user-friendly open-source platform for the visualization and manipulation of complex networks [9]. We here present cyTRON, a stand-alone Cytoscape app, and cyTRON/JS a web application which employs the functionalities of Cytoscape/JS, both of which allow to produce, visualize and manipulate cancer evolution models, also by interacting with public genomic databases. Figure 1B shows an example of the output in cyTRON/JS, which exploits Cytoscape/JS to provide an interactive visualization of the evolution model.

cyTRON and cyTRON/JS were designed for two main purposes:

- Providing an *interactive* and *user-friendly* visualization of TRONCO models: while TRONCO R-based graph display is static, cyTRON and cyTRON/JS provide interactive views, which allow to directly retrieve information about genes involved in the study, by accessing widely-used public genome databases.
- Making TRONCO *accessible* to users unfamiliar with R programming: cyTRON and cyTRON/JS provide interfaces that enable the usage of TRONCO respectively from Cytoscape and a Web browser, thus removing the need for users to execute any code in order to complete a whole analysis.

The architecture of both tools can be conceptually defined as follows:

- The *front-end* side is composed of an interface that can be used to:
  1. Select input data for the TRONCO analysis [3]: the input files should be either MAF, GISTIC or user-defined Boolean matrices that contain information about the mutations observed in each sample.
  2. Set the parameters for the inference in order to access most TRONCO capabilities. Users can indicate which driver mutations to include in the analysis, which algorithm among those implemented in TRONCO to use for the reconstruction and the algorithm's corresponding parameters.
  3. Visualize cancer evolution models and dynamically interact with the result: for instance, by clicking on the genes of the output graph, it is possible to retrieve the information available on public genomic databases. cyTRON gives access to gene information through databases such as Ensembl[1] and Entrez[2], that are accessible through the Cytoscape interface.
     In cyTRON/JS the data displayed for each node are retrieved from the Entrez database Gene[3] using E-Utils, an API provided by the National Center for Biotechnology Information.
- The *back-end* side includes the communication channel with R. For cyTRON, a Java bridge with R is built by means of rJava. Instead, for cyTRON/JS it is based on js-call-r, a Node.js package which collects the data and parameters set by the user, encodes them in JSON and sends them to R.

[1] http://www.ensembl.org/index.html.
[2] https://www.ncbi.nlm.nih.gov/search/.
[3] https://www.ncbi.nlm.nih.gov/gene/.

Then, R commands are transparently executed in order to perform any specific step of the analysis by TRONCO.

Figure 1 shows a view of cyTRON workspace (left) and an example of output model by cyTRON/JS (right). In order to choose between the two tools, users should take into consideration the data and the type of analysis they need to carry out. In particular, since cyTRON/JS is a web application, it is readily accessible from any device and computations are carried out on the back-end side. This feature is useful in case a user needs to carry out a computational-expensive analysis. However, cyTRON is more complete with respect to all the functionalities implemented in TRONCO: for example, it implements also the option of testing hypothesis on mutations through the algorithm Capri [6].

A Note on How cyTRON/JS *Handles Uploads.* In order to perform its analysis, cyTRON/JS requests a pro-forma registration on the site. This is only to identify the datasets subsequently loaded. cyTRON/JS does not require, or record any email or other identifying information. The datasets are uploaded by the user under his responsibility; that is, cyTRON/JS assumes that the user has all the rights to upload and share the data, which is used "as-is" and not modified by cyTRON/JS. In other words, cyTRON/JS does not claim to be GDPR-compliant or to follow the *data stewardship* guidelines put forth by FAIR [11].

3 Case Study

We present a case study on mutation data of prostate cancer, which can be downloaded from cyTRON/JS's GitHub repository[4]. This dataset was downloaded from TCGA, and it is composed of a MAF file which contains data about SNPs, Insertions and Deletions and a GISTIC file which contains data about CNVs.

For the first part of the analysis, both cyTRON and cyTRON/JS provide an interface to select the input files and, optionally, to choose which genes and which samples to consider in the analysis. The last two options can be exploited to carry out an analysis over a specific cancer subtype and to restrict the inference only to driver mutations. For this case study we selected only those samples that present a mutation on the ERG gene, and we restricted the analysis to driver mutations.

Finally, we ran the inference with CAPRI algorithm: through the user interface it is possible to indicate the heuristic search to be performed, enable the estimation of error rates through bootstrap and specify the regularization criterion for the likelihood estimation. Once the inference was performed, we visualized the result in cyTRON/JS: we present an example of how cyTRON/JS can serve as a means to identify associations among genes. Indeed, by clicking on a node of the graph users can read additional information about the corresponding gene and they can exploit this information to find which genes are involved in the same processes.

[4] https://github.com/lucreziaPatruno/cyTRON_JS/tree/master/examples.

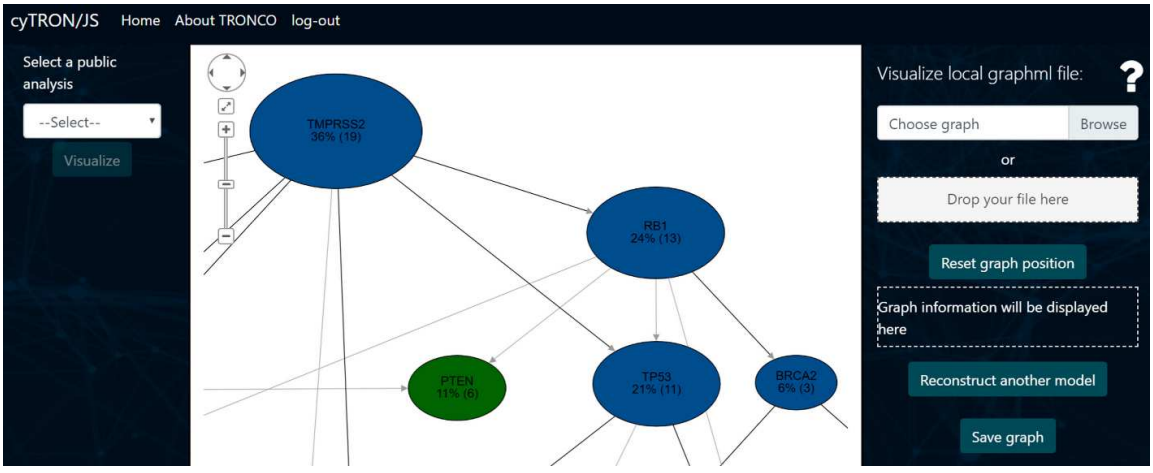

Fig. 2. Example of the visualization provided by `cyTRON/JS`: this figure displays a portion of the progression model reconstructed from prostate cancer mutation data.

Figure 2 displays a portion of the output graph that resulted from the case study. The observations that stood out while exploring the progression model are the following:

- There is an oriented edge which connects RB1 to TP53. The protein encoded by the former serves as a negative regulator for cell cycle, and the one encoded by the latter has a role in inducing cell cycle arrest, apoptosis and senescence.
- Another oriented edge connects RB1 to PTEN. The latter encodes a protein which is a negative regulator of the AKT/PKB signaling pathway.

Therefore through the `cyTRON/JS` interface it was possible to identify two connected nodes that represent genes which both have a role in cell cycle.

Even though these might be trivial associations, this is an example of how this tool can be exploited to better understand the role of genes that are connected in the graph, possibly finding interesting associations.

Given the fact that some researchers may want to access `cyTRON/JS` only to explore some already constructed model, the web application contains a section which allows users to access without authenticating, in order to visualize one of their local `graphml` files.

4 Conclusion and Future Work

`TRONCO` is an `R` package that implements state-of-the-art algorithms for the inference of cancer evolution models with the ultimate goal of understanding the evolutionary trajectories driving tumor evolution. In such a multidisciplinary domain, where computer scientists actively cooperate with biologists, being capable of visually understanding the data is crucial to both parties. In order to effectively allow the use of `TRONCO`, we here presented `cyTRON` and `cyTRON/JS`, two `Cytoscape`-based applications which translate many of `TRONCO` functionalities

into Cytoscape. Our effort aims at designing user-friendly and accessible tools to support the user in the task of exploring cancer genomic data.

As the TRONCO functionalities are extended, these two new tools need to be kept up to date. Thus, future work will be focused on integrating TRONCO's new algorithms for analyzing single cell datasets [8]. In addition to this, cyTRON/JS needs to be extended with the hypothesis testing functionality, in order to enable users to carry out more complex analysis.

Availability

The code for cyTRON is available at https://github.com/BIMIB-DISCo/cyTRON and on the Cytoscape App Store http://apps.cytoscape.org/apps/cytron.

The source code for cyTRON/JS is available at https://github.com/BIMIB-DISCo/cyTRON-js and the tool itself is accessible from https://bimib.disco.unimib.it/cytronjs/welcome.

Acknowledgments. cyTRON and cyTRON/JS were developed within the Google summer of Code program, in collaboration with the National Resource for Network Biology (NRNB – www.nrnb.org); we thank Alexander Pico and Kristina Hanspers of Gladstone Institutes of San Francisco, CA, USA, for their support. The authors declare no conflicts of interest.

References

1. Antoniotti, M., et al.: Design of the TRONCO bioconductor package for translational oncology. R J. **8**(2), 39–59 (2016)
2. Caravagna, G., et al.: Algorithmic methods to infer the evolutionary trajectories in cancer progression. Proc. Nat. Acad. Sci. **113**(28), E4025–E4034 (2016)
3. De Sano, L., et al.: TRONCO: an R package for the inference of cancer progression models from heterogeneous genomic data. Bioinformatics **32**(12), 1911–1913 (2016)
4. Loohuis, L.O., et al.: Inferring tree causal models of cancer progression with probability raising. PLoS ONE **9**(10), e108358 (2014)
5. Nowell, P.C.: The clonal evolution of tumor cell populations. Science **194**(4260), 23–28 (1976)
6. Ramazzotti, D., et al.: CAPRI: efficient inference of cancer progression models from cross-sectional data. Bioinformatics **31**(18), 3016–3026 (2015)
7. Ramazzotti, D., Graudenzi, A., Caravagna, G., Antoniotti, M.: Modeling cumulative biological phenomena with Suppes-Bayes causal networks. Evol. Bioinform. **14**, 1176934318785167 (2018)
8. Ramazzotti, D., Graudenzi, A., De Sano, L., Antoniotti, M., Caravagna, G.: Learning mutational graphs of individual tumour evolution from single-cell and multi-region sequencing data. BMC Bioinformatics **20**(1), 210 (2019)
9. Shannon, P., et al.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. **13**(11), 2498–2504 (2003)
10. Turajlic, S., et al.: Deterministic evolutionary trajectories influence primary tumor growth: TRACERx renal. Cell **173**(3), 595–610 (2018)
11. Wilkinson, M.D., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data **3**(1), 160018 (2016)