Network visualisation of synthetic biology designs

Matthew Crowther
m.crowther1@ncl.ac.uk
Newcastle University
Newcastle Upon Tyne, United Kingdom

Anil Wipat
anil.wipat@ncl.ac.uk
Newcastle University
Newcastle Upon Tyne, United Kingdom

Ángel Goñi-Moreno
angel.goni@upm.es
Universidad Politécnica de Madrid
Madrid, Spain

1 ABSTRACT

Visualising the complex information captured by synthetic biology designs is still a major challenge. The popular glyph approach where each genetic part is displayed on a linear sequence allows researchers to generate diagrams and visualise abstract designs [2], but only represents a single, static representation that results in visualisation that is not specific to the requirements of a user resulting in a one-size-fits-all visualisation. We developed a network visualisation technique that automatically turns all design information into a graph, displaying otherwise hidden data. The structure of the resulting graphs can be dynamically adjusted according to specific visualisation requirements, such as highlighting proteins, interactions or hierarchy. Since biological systems have an inherent affinity with network visualization [6], we advocate for adopting this approach to standardise and automate the representation of complex information.

2 RESULTS

Firstly, a NOR gate design (adapted from [8]) is used to showcase some fundamental visualization processes and the methods. Secondly, more complex regulatory circuits are used to illustrate the potential of the network visualisation approach to effectively display novel features.

Data. Before any visualisation can be realised, the underlying data representation must be considered. Without a rich data representation, most meaningful visualisation is not achievable simply due to the data not being encoded. Therefore, we will use the Synthetic Biology Open Language (SBOL) [5] as the data capture format. SBOL is a more formal and synthetic biology-centric approach to the design specification.

View. Visualizing unmodified data will produce an incomprehensible visualisation as the domain is too broad so the significance of connections is lost. As seen within Figure 1A, despite a small design due to the verbose nature of the underlying data very little can be inferred. Therefore, a view is defined as an aggregation of data to produce a graph that is focused on a specific aspect of the design. With a more concentrated domain focusing on a single aspect, visual complexity is reduced. For this introduction, a basic view that aggregates data into the overall design and constituent biological parts and entities is used as seen within Figure 1B.

Layout. A view of any meaningful size will produce an incoherent visualization when the position of nodes do not consider the data being represented. Layout pertains to the coordinate location of the nodes within the plot. Trivially, layouts can ensure the rendered nodes and edges do not overlap but more significant implementation can have a layout mirroring the intent behind the data being visualized. In the current working example (Figure 1A, despite a better visual, the network is incoherent as no positional data is encoded. However, as seen within Figure 1C, despite a relatively basic layout, the visual output is more clear.

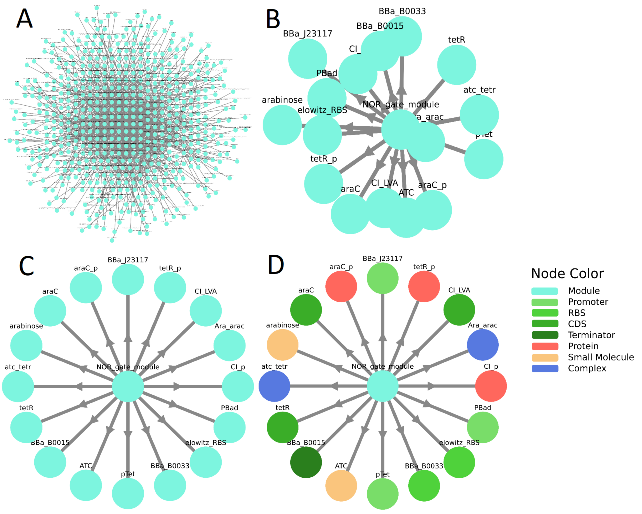

Figure 1: A) All encoded data is rendered. B) Simple view of constituent biological parts, proteins and non-genetic entities described within a design. C) A basic concentric layout - all constituent entities are positioned around the central node denoting the overall design. C) Addition of colour mapping with biological entities types and roles within the design.

Label reduction. Labels are added directly to the graph, connected to edges or nodes. However, screen space is finite and can become saturated. This issue is compounded when: nodes are closely positioned, the rendered text is long, and/or the graph is highly connected. Therefore, label reduction is the process of replacing labels with visual features to increase concision but still encode the information. With the working example, while the core focus on the view is comprehensible, information that may be desired is not present. As seen within Figure 1D, a user may want to visualise the role of each biological entity.

Visualising complex information via *presets*

The overview previously discussed is only one instance of producing a comprehensible visualisation. Here we use the term “preset” to denote a view combined with a collection of visual techniques that are complementary to said view such that the visual output focuses attention upon a specific and desired feature of a design. Below two presets are discussed, including intent and how visual modifications have an affinity with the view. However, this is not an exhaustive list and providing the information is encoded within the design data, any feature of a design can be visualised using a network/graph focused approach.

Hierarchy. The hierarchy of a design focuses on visualizing how the different perceived levels of biological entities are structured. This provides insight into each abstraction level and how the components of each level map to their neighbours. Furthermore, a hierarchical view can visualize a design of arbitrary depth which is beneficial since the levels of abstraction within a design increase as modules become larger (parts, devices, circuits, systems, consortia). Figure 2 displays a hierarchical view that allows not only a visualization of individual parts and constructs but also the makeup of larger modules.

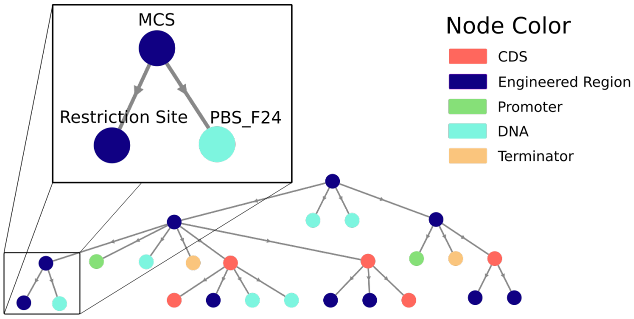

Figure 2: Visualising the *digitalizer* synthetic circuit from [3] with a hierarchical representation of genetic modules in 4 layers.

Interaction. In contrast with sequence-level visualisations, where intent and function are not explicitly described and non-genetic entities are often poorly represented, interaction networks provide an explicitly functional perspective. By using this view, non-genetic entities (e.g., proteins) are easily represented. Furthermore, the description of biochemical networks fit well into a graph-based approach since these are conceptualized as a set of interacting entities. We visualised a Boolean genetic circuit (Figure 3) by using only its interactions and non-genetic elements. Inputs, outputs and information flow are easily comprehended even at a glance. However, visualising the same relatively complex design using sequence-level information by deriving functional details would be more challenging.

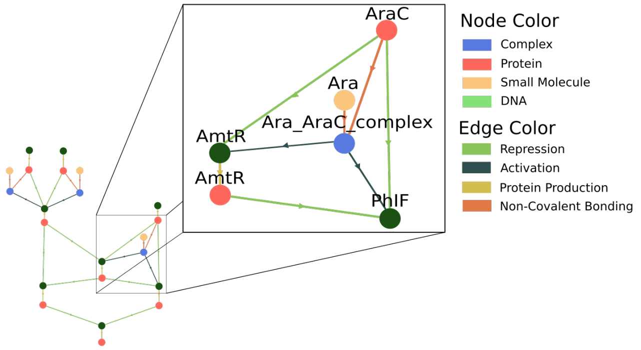

Figure 3: Visualising interactions (edges) and non-genetic elements (nodes) of the *0xC7* Boolean genetic circuit from [7].

Scaling abstraction. Very often, despite the focus on a specific design feature (e.g., non-genetic elements), issues of comprehension still arise due to the level of design details and annotations. The ability to visualize a higher level of abstraction allows a more granular, and more easily comprehensible output [4]. Figure 4 displays the same design as in Figure 3 but at a higher level of abstraction, which may be more adequate for a rapid design check.

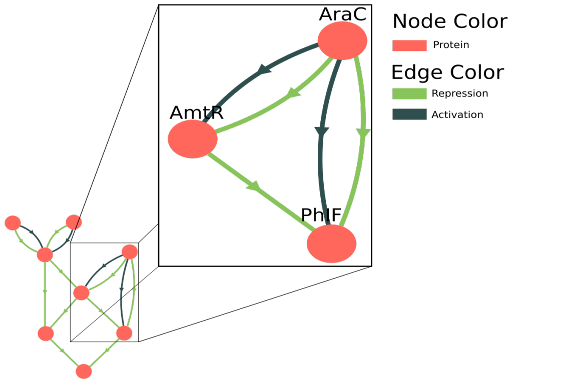

Figure 4: Visualising interactions (edges) and non-genetic elements (nodes) of the *0xC7* Boolean genetic circuit from [7] with *higher* abstraction.

While increasing the abstraction level can produce a visual output that is more comprehensible in terms of function, the reduced granularity can lead to more ambiguous visualisations concerning mechanistic details. Therefore, lowering the level of abstraction (thus visualising more details) may be beneficial in some cases, for instance, when building a mathematical model of a genetic design. Figure 5 is a more detailed view compared to Figure 3 and despite a considerable complexity increase, interactions are broken down into a number reactions thus providing more detailed information.

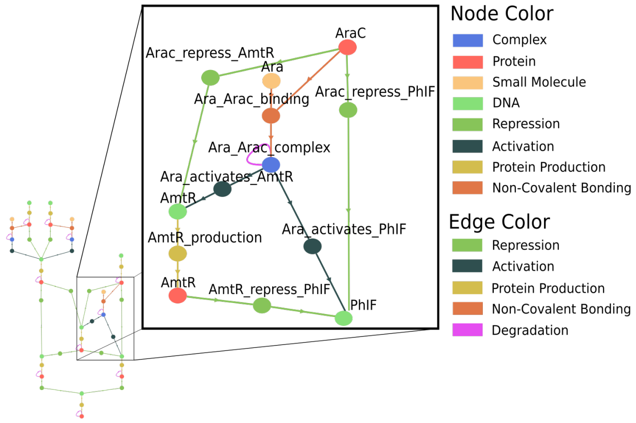

Figure 5: Visualising interactions (edges) and non-genetic elements (nodes) of the *0xC7* Boolean genetic circuit from [7] with *lower* abstraction.

3 DISCUSSION & FUTURE WORK

We present a visualisation method that offers an alternative approach to a conventional genetic-parts-based glyph. Designs can be automatically produced at differing levels of abstraction, as defined by the user. This approach promises to help understand complex designs more easily, and to scale better for large designs, such as chromosomes.

Future efforts will focus on four aspects. Firstly, the fact that networks are generated automatically and adjusted dynamically paves the way to develop powerful user interaction tools. Secondly, we will exploit the full potential of graphs for mathematical analysis through a wealth of graph theory methods. Indeed, networks are not only useful for visualisation purposes but mathematical structures for studying data. Thirdly, networks allow representing any type of data, not just gene design information. Therefore, specific visualisation networks of every stage throughout a standardised DBTL lifecycle [1] will be coupled into layered graphs that will include from automation to characterisation to modelling information. Finally, most current visualisation techniques will not scale to large designs. Therefore, exploring how network visualisation that has precedence with large data visualisation can be applied to designs of extreme size.

Visualization is complementary to the development of data standards. Here, we use designs encoded with the SBOL since this captures richer information than GenBank or FASTA formats.

REFERENCES

[1] Beal, J., Goñi-Moreno, A., Myers, C., Hecht, A., de Vicente, M. d. C., Parco, M., Schmidt, M., Timmis, K., Baldwin, G., Friedrichs, S., et al. The long journey towards standards for engineering biosystems: Are the molecular biology and the biotech communities ready to standardise? *EMBO reports 21*, 5 (2020), e50521.

[2] Beal, J., Nguyen, T., Gorochowski, T. E., Goñi-Moreno, A., Scott-Brown, J., McLaughlin, J. A., Madsen, C., Aleritsch, B., Bartley, B., Bhakta, S., et al. Communicating structure and function in synthetic biology diagrams. *ACS synthetic biology 8*, 8 (2019), 1818–1825.

[3] Calles, B., Goñi-Moreno, Á., and de Lorenzo, V. Digitalizing heterologous gene expression in gram-negative bacteria with a portable on/off module. *bioRxiv* (2019).

[4] Heinemann, M., and Panke, S. Synthetic biology—putting engineering into biology. *Bioinformatics 22*, 22 (09 2006), 2790–2799.

[5] Madsen, C., Moreno, A. G., Umesh, P., Palchick, Z., Roehner, N., Atallah, C., Bartley, B., Choi, K., Cox, R. S., Gorochowski, T., et al. Synthetic biology open language (sbol) version 2.3. *Journal of integrative bioinformatics 16*, 2 (2019).

[6] Miele, V., Matias, C., Robin, S., and Dray, S. Nine quick tips for analyzing network data. *CoRR abs/1904.05334* (2019).

[7] Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., Ross, D., Densmore, D., and Voigt, C. A. Genetic circuit design automation. *Science 352*, 6281 (2016).

[8] Tamsir, A., Tabor, J. J., and Voigt, C. A. Robust multicellular computing using genetically encoded nor gates and chemical 'wires'. *Nature 469*, 7329 (Jan 2011), 212–215.