Logic gene network design
a software tool based on modularity and standardization

van den Berg, Bastiaan; Reinders, Marcel; Bellomo, Domenico

Publication date
2009

Document Version
Final published version

Citation (APA)
van den Berg, B., Reinders, M., & Bellomo, D. (2009). *Logic gene network design: a software tool based on modularity and standardization*. 1. Poster session presented at NBIC Conference 2009, Lunteren, Netherlands.

Logic gene network design:
a software tool based on modularity and standardization

Bastiaan A van den Berg[1], Marcel JT Reinders[1,2], Domenico Bellomo[1,2,3]
[1] The Delft Bioinformatics Lab, Delft University of Technology, Delft
[2] Kluyver Centre for Genomics of Industrial Fermentation, Delft University of Technology, Delft
[3] Bioprocess Technology Section, Delft University of Technology, Delft

Summary

Synthetic biology aims at building biological systems for useful purposes. Relatively simple gene networks have been engineered, but the design process is limited. Many papers advocate the use of engineering concepts like standardization and modular design to simplify the design process and enable the design of more complex systems. We have developed a software tool (Figure 1) to show how standardization and modular design can be used for the design of logic gene networks, gene networks that implement a logic function.

The user can design a logic network template using modular logic gates. The software tool uses bioparts from an artificial bioparts database to build devices that implement the given template. The software converts each device to a model, which is used for stochastic simulation. Comparison of the simulation result with the user-defined desired result provides an evaluation score for each device.

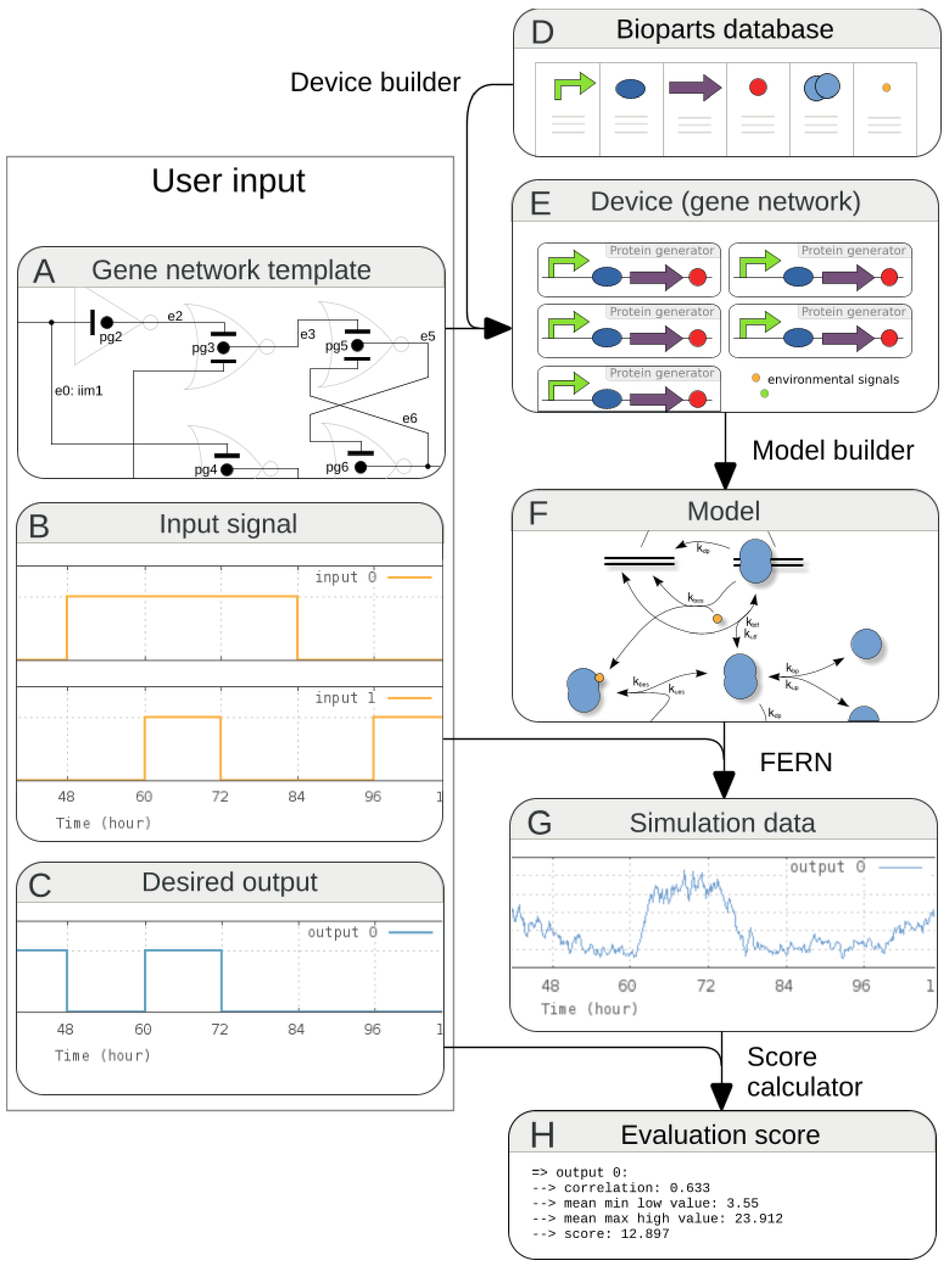


Figure 1 | **Software overview**
A) Using modular gene network templates, a user can simply connect logic gates to build a logic network. **B,C)** The user has to specify the input signal and the desired output signal. **D,E)** The software converts a template into so called devices, using standard biological parts from an artificial bioparts database. **F)** The software converts each device to a model. **G)** The model is used to run a stochastic simulation which results in a simulated output signal. **H)** The performance of a device is evaluated through comparisson of the simulated output signal with the user-defined desired output signal.

Results

We have used the software to design three logic networks: a demultiplexer, a decoder, and a DLatch. Figure 2 shows the design of the DLatch which constists of four NOT and four NOR gates.

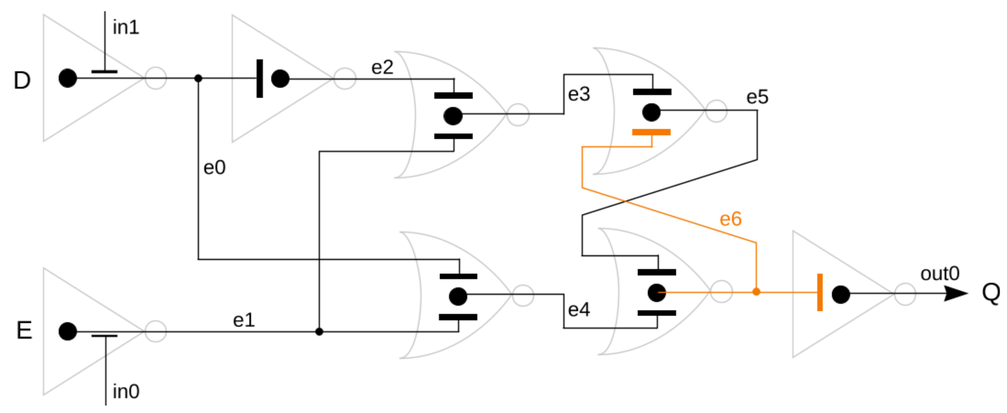


Figure 2 | **Design of a DLatch gene network**

With a small artificial bioparts database, the conversion from the template (Figure 2) to all possible devices provided 702 different devices. Figure 3 shows the desired output signal at the top and the simulation results of four out of the 702 devices below, ordered from worst performance to best performance.

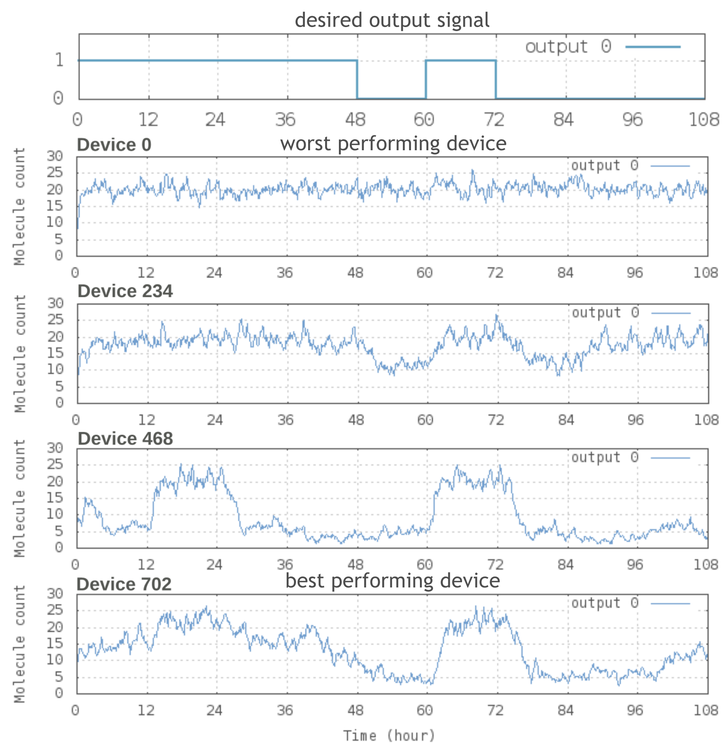


Figure 3 | **Desired and four simulated output signals**

Analysis of the simulation results shows a preference for certain biological parts at certain locations in the gene network. For example, Figure 4 shows that most well performing DLatch devices use a tetrameric transcription factor for edge *e6* (the orange edge in Figure 2).

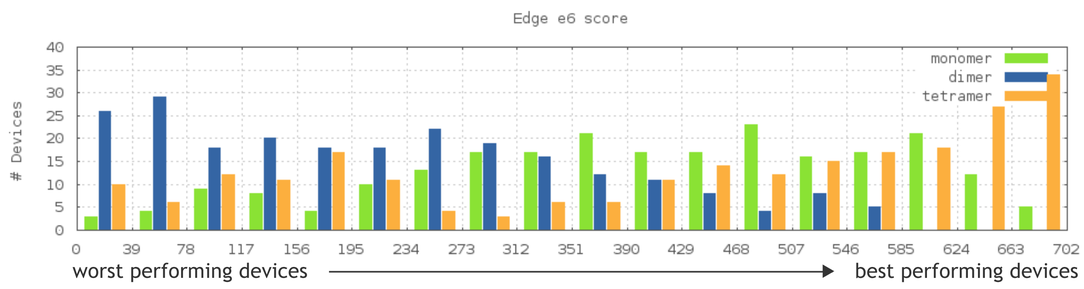
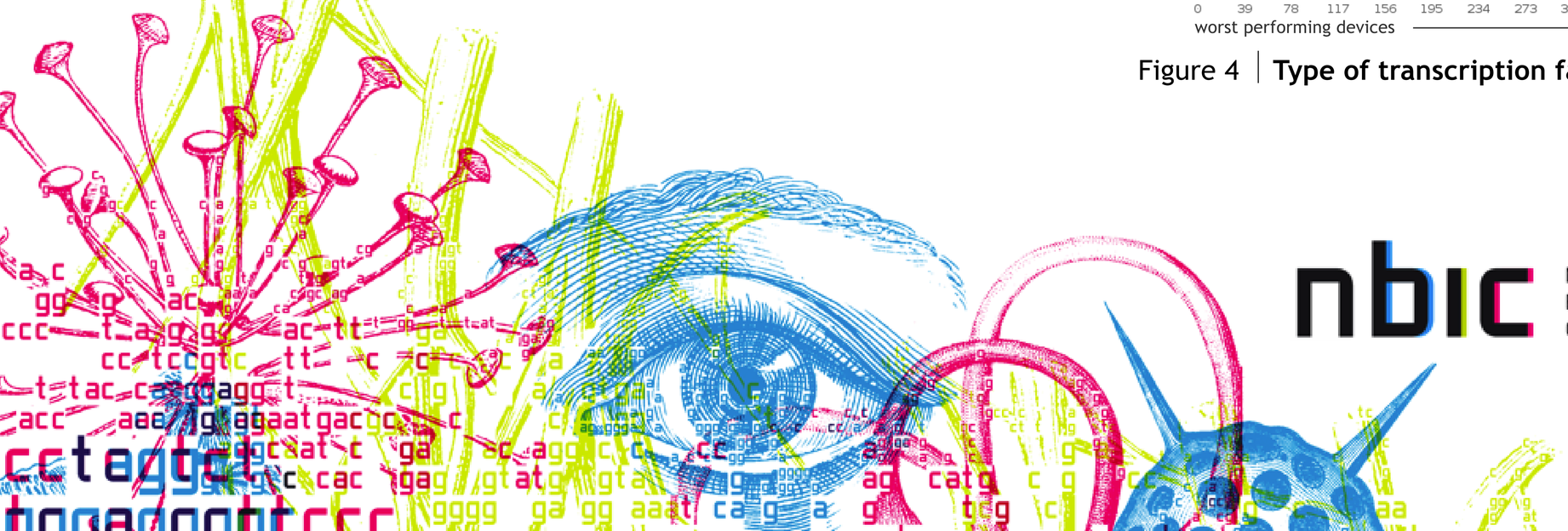


Figure 4 | **Type of transcription factor used for edge *e6* in Figure 2**



nbic Netherlands Bioinformatics Centre | TU Delft