C.0.1 Systems Biology: Biological Languages to Design Natural Genomes with the Cell Cycle Example

We explored the possibility of scaling-up our approach by designing a G2P language for a natural genome from a system biology perspective. Indeed, looking at the system biology model addressed most syntactic issues because we focus only on a set of molecules, often corresponding to discrete genetic areas of interest that are not overlapping.
In the previous sections, the GRN examples have illustrated the use of our semantic DNA compilation system in making biological constructs. We used attribute grammars that appear to be suitable for genotype to phenotype mapping. In order to show the possibility to design not only genetic constructs but genomes, we built an attribute grammar for the budding yeast genome and modeled the yeast cell cycle [Chen et al. 2004]: in a logical and structured fashion, information from genomic databases and mathematical models will be combined for the exploration of mutants.

For instance, the budding yeast cell cycle regulation system [Chen et al. 2004] draws heavily on experiments (131 mutants), but the model itself does not encompass genetic data. We develop a linguistic model of the yeast genome expressing how it encodes a complex regulatory network. First, the syntax of the cell cycle grammar must be defined. We used a systems biology approach to focus on essential elements. Therefore, by looking at the mathematical model, we can identify the protein of interest and use a genomic database, Saccharomyces Genome Database (SGD) [Cherry et al. 1998], to find the corresponding coding sequences and their locations in the genomes of the cell. A map can then be drawn, and the syntax of a grammar proposed C.2. The genome is broken down into its sixteen constituent chromosomes, and each chromosome can be further broken down into the functional elements of the cell cycle it carries.
Then using biological knowledge and the list of mutants and their impact on the mathematical model (`http://mpf.biol.vt.edu/research/budding_yeast_model/pp/mutant_list.php`), the declaration of parameters is made for each wild-type part. The semantic action of each chromosome may carry declarations for the equation of the proteins encoded and their declaration as species as well as rule to compute parameters values or for the formation of complexes.
Some elements of the mathematical model cannot be linked to a DNA sequence, and are either in the category Model or declared in the semantic action of the Start rule. For example, it is the case for the intermediary enzyme (IE) or for the four events. There is an additional category, Init, which contains all the initial conditions for the protein of the system. Init and Model parts offers to choose different sets of parameters (wild type in glucose, or wild type in galactose, etc.) but the parts in these categories are only here for the semantics: they do not have any DNA sequence.

By reusing the SBML library, we implemented in GenoCAD a straightforward version of the yeast genome that outputs the mathematical model for the wild type version in Tables C.0.1 of Appendix C and the GenoCAD and compiler generated files are at `http:`

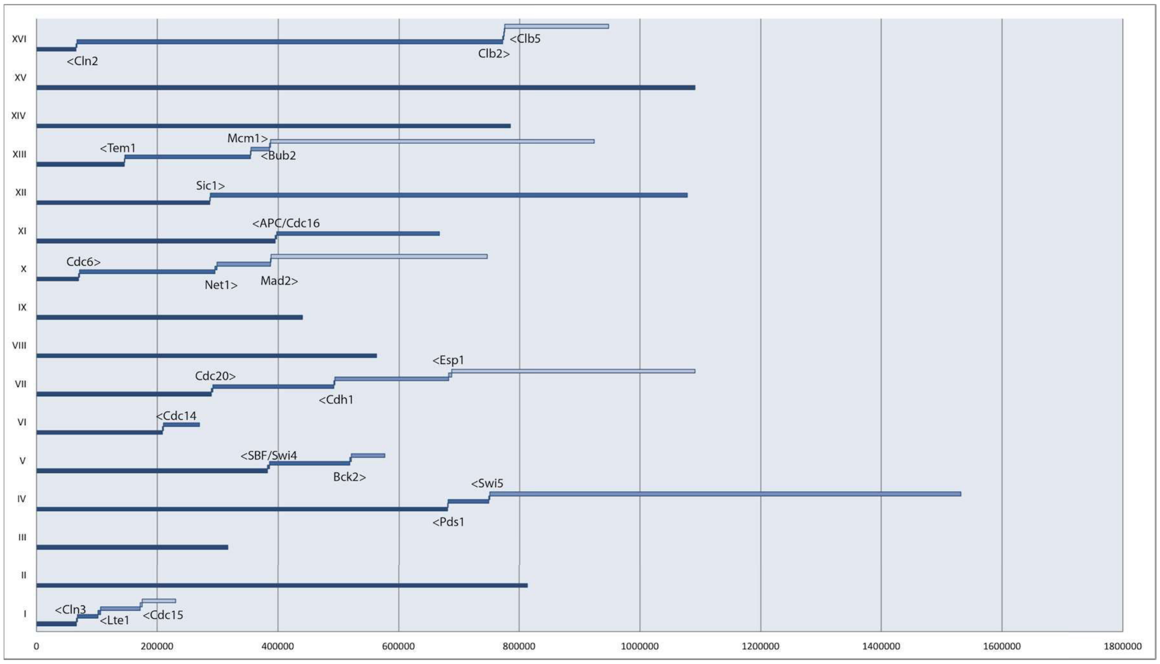

Figure C.2: Syntax proposed for the attribute grammar for the cell cycle of the Budding Yeast Genome. We listed the key proteins for the regulation of the cell cycle used in Chen's model and search for their coding sequences and locations on the yeast genome in the SGD database `www.yeastgenome.org/?`. We were then able to draw this map and derive the syntactic rules for the yeast genome. The 'Genome' is transformed into the 16 chromosomes. Each chromosome may have a rule that let the users choose the coding sequences of the genes located on this chromosome that are related to the regulation of the cell cycle.

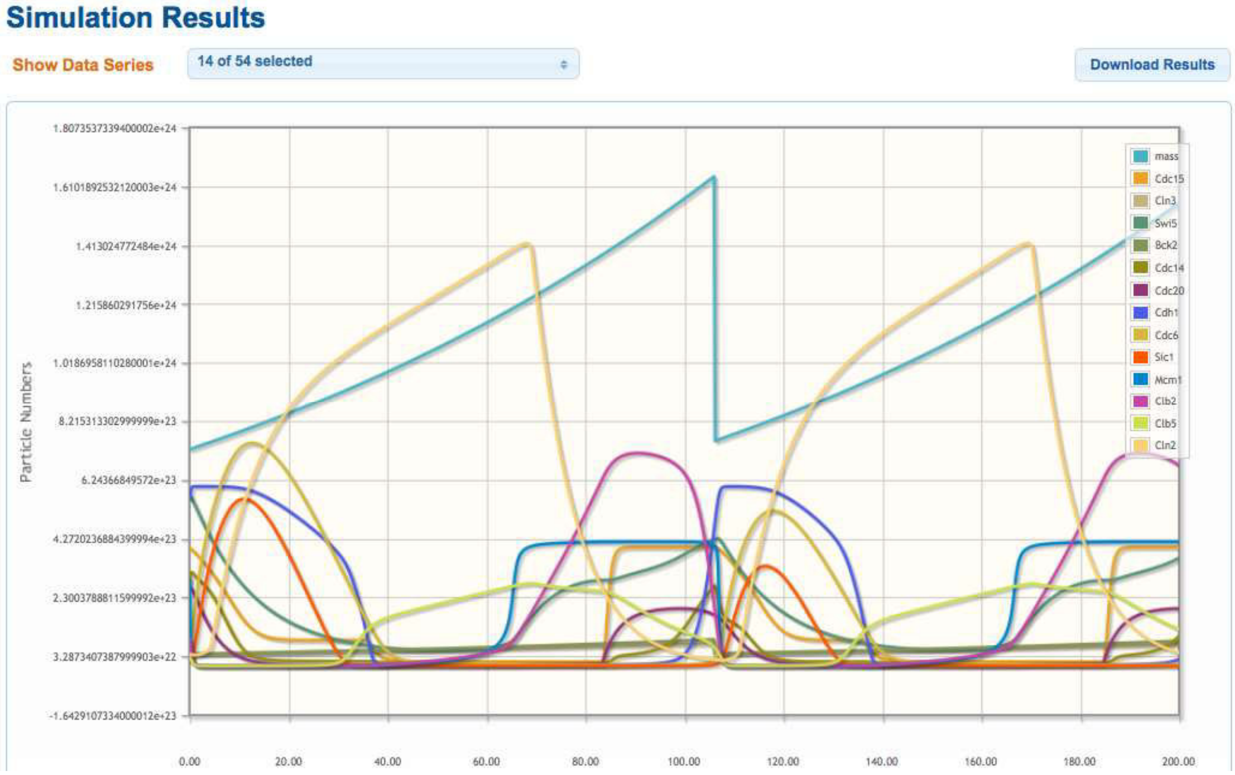

Figure C.3: Simulation results for the cell cycle of the Budding Yeast Genome Wild-Type design in GenoCAD. The time course simulation is ran for 200min. We can observe in blue the mass. The cell divides after about 110 minutes

`//web.figshare.com/download/file/1118850`. The semantics have been implemented according to the BioModel database [Le Novère et al. 2006] at `http://www.ebi.ac.uk/biomodels-main/BIOMD0000000056`. After designing the Wild-Type (WT) genome in GenoCAD, we ran the simulation and obtained the WT phenotype as expected, see Figure C.3.

Additionally,, we can introduce rules to design mutants. For instance, the deletion of cln2 may be syntactically represented by the rule 'Cln2 → []'. The semantics of this rule states that the parameter for the synthesis of cln2, kppsn2, is null. When a design uses this rule, the null value of ksppn2 is now used for the SBML model. The simulation of a *cln2*Δ mutant shows that the cell grows bigger and divides after about 160 minutes, as cln2 is responsible for bud emergence and its absence delays it.

References

[1] KC Katherine C Chen et al. “Integrative analysis of cell cycle control in budding yeast.” In: *Molecular biology of the cell* 15.8 (Aug. 2004), pp. 3841–3862. ISSN: 1059-1524. DOI: 10.1091/mbc.E03.

[2] JM Cherry et al. “SGD: Saccharomyces genome database”. In: *Nucleic acids* 26.1 (1998).

[3] Nicolas Le Novère et al. “BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems.” In: *Nucleic acids research* 34.Database issue (Jan. 2006), pp. D689–91. ISSN: 1362-4962. DOI: 10.1093/nar/gkj092.