18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Emergent Models in a Multi-Level Biochemical Network Regulating Pea Flowering

Jacob Stolk [1] **and Jim Hanan** [2]

[1] *Dione Complex Systems, Auckland, New Zealand*
[2] *The University of Queensland, Centre for Biological Information Technology, Brisbane, Australia*
Email: Jacob@DioneComplexSystems.com

Abstract: The Emergent Models methodology (EM) is an adaptive computational method for discovering models of complex systems in computer simulations (Stolk 2005). EM uses machine learning and optimisation algorithms such as genetic programming. Stolk and Hanan (2007) used EM to discover genetic regulatory network models of branching in *Pisum sativum* (pea). Here EM is used to discover models of genetic and metabolic networks regulating flowering in pea. These models describe multiple levels and components in the whole plant complex system, including genes, intercellular signals, modules and phenotype.

Flowering in pea is determined by genes and mobile signals, mediating environmental influences such as photoperiod. Models of biochemical mechanisms explaining flowering time of pea studied here incorporate modules such as a circadian clock, signal processors and switching mechanisms. Each module is a combination of chemical reactions. Three hierarchical system levels are involved: the top level of the whole plant (phenotype); a middle level of modules; a bottom level of chemical reactions.

It was hypothesised that models describing each level could be automatically discovered by genetic programming, given data on the next higher level. Discovered models should predict experimental data on gene expression and flowering time of wild type and several mutant pea plants. The purpose of this research is a proof of principle of a computational method, so models were kept deliberately as simple as possible and no attempt was made to incorporate the latest biological insights.

In three experiments it was attempted to discover models of a biochemical reactions network leading to a circadian clock, i.e. a time evolution with a periodicity of about 24 hours of at least one chemical substance in the network. Nodes of possible networks were concentrations of chemical substances. Connections were reactions between these substances. Possible reactions included activation, inhibition, synthesis, transformation and degradation. It was shown that genetic programming can successfully reverse engineer a circadian chemical clock, but that the exact solution found depends on the flexibility allowed by imposed constraints.

In another experiment it was assumed that flowering time of the plant was determined by a network of modules. Nodes of possible networks were modules, each consisting of a set of chemical reactions performing a well-defined function. Modules were connected by reactions between chemical substances representing output and input signals. One module's input could be activated or inhibited by another module's output. Possible modules included: a circadian clock; a light sensing module; a long day signal module combining the clock and light signals to determine day length; a flowering signal module; a flowering switch module determining flowering time.

Genetic programming, using as inputs possible modules and connecting activation and inhibition functions, produced parameters for these functions leading to a model with a good fit to the experimental data on flowering time of different pea mutants in different photoperiod conditions.

In this research it has been shown how genetic programming can be used in a modular system at multiple levels. At the lowest level, possible mechanisms of a module can be inferred from the module's function. At a higher level, knowledge about the way modules fit together can be obtained from the behaviour of the whole system.

Keywords: *Emergent Models (EM), systems biology, computer simulation*

1. INTRODUCTION

The Emergent Models methodology (EM) (Stolk 2005) is an adaptive computational method for discovering models of complex systems (Bar-Yam 1997; Holland 1998) in computer simulations. EM uses machine learning and optimisation algorithms such as genetic programming (Koza 1992; Koza et al. 1999; Luke 2002). The present research aims to demonstrate application of EM to living organisms and is meant as a proof of principle, so models were kept deliberately as simple as possible and no attempt was made to incorporate the latest biological insights.

Living organisms consist of interacting objects such as organs, cells, and genes (Bower and Bolouri 2001; De Jong 2002; Ptashne and Gann 2002). Therefore, they can be considered complex systems and we have applied a complex systems simulation methodology to a living organism, focussing on the relationship between interacting genes, metabolic networks and phenotype. Stolk and Hanan (2007) used EM to discover genetic regulatory network models of branching in *Pisum sativum* (pea), given phenotypic data on mutant grafted plants. The plant was regarded as a two level system with the level of the genotype and the level of the phenotype.

Here EM is used to discover models of genetic and metabolic networks regulating flowering in pea. These models describe multiple levels and components in the whole plant complex system, including genes, intercellular signals, modules and phenotype. The plant is regarded as a three level system, with the level of genetic and metabolic reactions, the level of modules of interactions related to specific functions of the organism, and the level of the whole plant. At each level models are discovered from assumed known behaviour or data at the next higher level.

2. FLOWERING IN PEA: GENES, MODULES AND PHENOTYPE

Flowering in pea is determined by genes and mobile signals, mediating the influence of, for example, photoperiod (Weller et al. 1997; Beveridge et al. 2003; Bell et al. 2003). Bell et al. (2003) have collected experimental data on gene expression and flowering time of wild type and several mutant pea plants. These data are reproduced by Stolk (2005, Appendix 2, Table 8). It is assumed there is a linear relationship between time and plant growth, expressed as the number of nodes produced. When growth starts, vegetative nodes are produced before the first floral node is produced, so flowering time can be estimated by the number of the first floral node, or node of floral initiation (NFI). The data reflect gene expression of genes Gigas (GI), Sterile Nodes (SN), and Late Flower (LF). Wild type expression level is assigned a value 1 by definition and other expression values are relative to the wild type for six mutants. Data were also collected on NFI values for each plant, for a photoperiod of 24 and 8 hours respectively.

In Arabidopsis it has been demonstrated that the circadian clock also plays a role in determining flowering time (Mouradov et al. 2002). Possible molecular mechanisms of circadian clocks have been elucidated in Drosophila (Goldbeter 1996) and in Arabidopsis (Zeilinger et al. 2006).

It was hypothesised that a model describing the genetic and biochemical mechanisms explaining flowering time of pea could incorporate several modules, such as a circadian clock, the effect of photoperiodism, and a switching mechanism (see Figure 1). Each module should function by a mechanism of chemical reactions. Thus, three levels of reality are involved: the top level of the whole plant (phenotype); a middle level of modules; a bottom level of chemical reactions.

It was also hypothesised that, given qualitative assumptions about the overall structure of the model, details of models describing each level could be automatically discovered by genetic programming, starting from the next higher level. The discovered models should predict the experimental results.

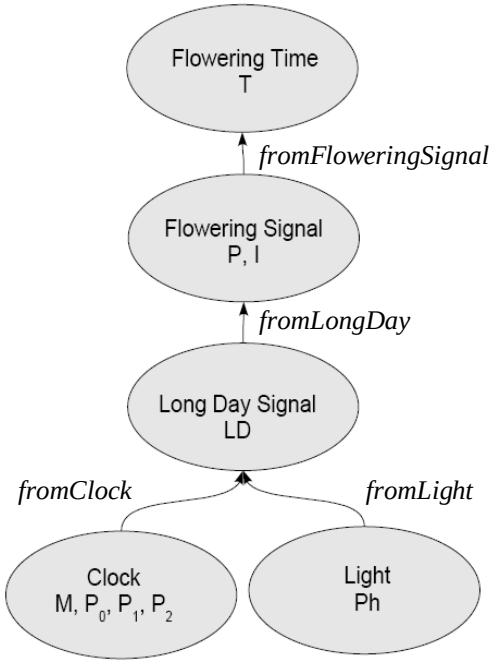

Figure 1. Modular model of flowering in pea. Variables are as in section 3.

3. COMPUTATIONAL EXPERIMENTS

In one series of computational experiments, described in section 3.1, it was attempted to discover models of biochemical reactions explaining the behaviour of a module, the circadian clock module. In other computational experiments, described in section 3.2, it was attempted to discover parameters of module interactions, given the behaviour of the whole plant.

3.1. Circadian Clock

It was attempted to reconstruct a network of chemical reactions leading to a circadian clock, i.e. a time evolution of at least one of the chemical substances in the network with a periodicity of about 24 hours. Nodes of possible networks were concentrations of chemical substances, and connections were reactions between these substances. Possible reactions included activation, inhibition, synthesis, transformation, and degradation. The genetic programming algorithm was given as input variables the concentrations of substances involved, as well as the possible reaction functions. Using these inputs, the algorithm was run to find a network of chemical reactions producing a time evolution of one of the substances approximating a given sine function with a periodicity of 24 hours.

The circadian clock was assumed to be described by a model similar to the period protein (PER) model for Drosophila (Goldbeter 1996), a simple model suitable for the proof of principle that is the aim of the present work. The original PER model was first written in a slightly different way to bring out the different kinds of chemical reactions, to be used as functions (building blocks) by the genetic programming algorithm. Reaction functions were defined as follows (all concentration variables are written in uppercase and all parameters in lowercase):

$$inhibition(v_i, k_i, C) = v_i \frac{{k_i}^n}{{k_i}^n + C^n}$$

$$synthesis(k_s, C) = k_s C$$

$$transformation(v_1, k_1, v_2, k_2, C, D) = -v_1 \frac{C}{k_1 + C} + v_2 \frac{D}{k_2 + D}$$

$$degradation(v_d, k_d, C) = v_d \frac{C}{k_d + C}$$

where:

C and D are concentrations of chemical substances; k_i and n are parameters of the Hill function used for inhibition, set to commonly used values of 1 and 4 respectively; all other lowercase symbols are parameters to be found.

Using these reaction functions, the PER model can be written as follows (omitting an equation for transport of P_2 to the cell nucleus, a process that is assumed given and not modelled here).

$$\frac{dM}{dt} = inhibition(v_s, K_I, P_N) - degradation(v_m, k_m, M)$$

$$\frac{dP_0}{dt} = synthesis(k_s, M) - transformation(v_1, k_1, v_2, k_2, P_0, P_1)$$

$$\frac{dP_1}{dt} = transformation(v_1, k_1, v_2, k_2, P_0, P_1) - transformation(v_3, k_3, v_4, k_4, P_1, P_2)$$

$$\frac{dP_2}{dt} = transformation(v_3, k_3, v_4, k_4, P_1, P_2) - degradation(v_d, k_d, P_2)$$

where:

M is the concentration of messenger RNA; P_0, P_1 and P_2 are concentrations of three proteins; t is time; and all other lowercase symbols are parameters of the model.

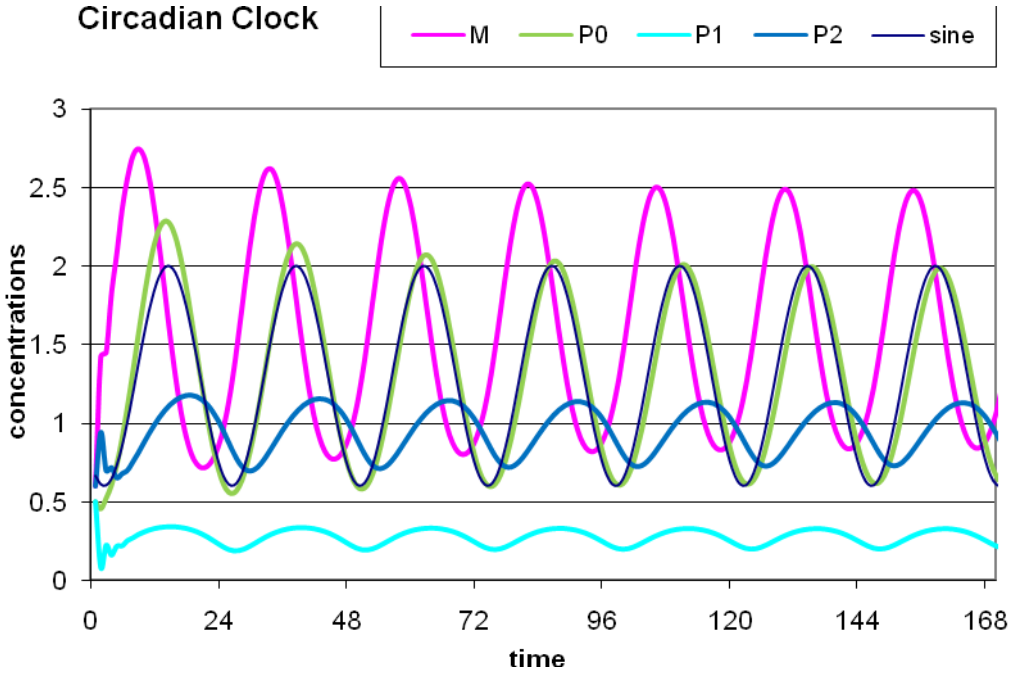

Figure 2. Circadian clock found by genetic programming.

Goldbeter (1996) showed mathematically that, for certain parameter values, this model produces oscillations of M, P_0, P_1 and P_2 with a period of about 24 hours. We are showing here that a genetic programming algorithm can find models with circadian clock behaviour by searching a space of possible models.

Inputs of the genetic programming algorithm were the reaction functions $inhibition$, $synthesis$, $transformation$, and $degradation$, variables M, P_0, P_1 and P_2, and all parameters except K_I, assumed to be a constant value of 1. Fitness was defined as the sum of deviations of the target protein concentration from a sine function with a period of 24 hours. The space of possible models searched by the genetic programming algorithm included not only models with different parameters, but also rearrangements of concentration variables and reaction functions. The search space was limited to realistic models by imposing constraints. For example, a substance could only exert inhibition or synthesis effects on a different substance, not itself; substances could only be transformed into different substances; a substance could only degrade itself; unrealistic solutions with negative concentrations were excluded by fitness penalties.

The following set of equations was obtained in a run of 10 000 individuals during 100 generations (a total of 1 000 000 individuals were evaluated).

$$\frac{dM}{dt} = inhibition(1.45, P_2) - degradation(1.66, 1.59, M)$$

$$\frac{dP_0}{dt} = synthesis(0.22, M) - transformation(0.66, 0.50, 0.62, 1.67, P_0, P_1)$$

$$\frac{dP_1}{dt} = transformation(0.66, 0.50, 0.62, 1.67, P_0, P_1) - transformation(2.43, 0.045, 1.51, 1.90, P_1, P_2)$$

$$\frac{dP_2}{dt} = transformation(2.43, 0.045, 1.51, 1.90, P_1, P_2) - degradation(1.37, 2.67, P_2)$$

Fitness, or the total deviation of modelled P_0 concentration over 300 time steps, was 16.35, with 261 hits (modelled results practically equal to the objective function) out of a possible 300. The time evolution of P_0 produced by these equations was close to the sine objective function, as shown in Figure 2.

This example demonstrates how a metabolic reaction model can be automatically discovered to produce circadian periodic behaviour. Automatic discovery without appropriate constraints easily leads to unrealistic

models. In this example, once sufficient constraints were imposed to exclude unrealistic solutions, the discovery algorithm found a model rather similar to the original model formulated by Goldbeter (1996). Applying an automatic discovery algorithm like the genetic programming algorithm used here makes it necessary to formulate explicit constraints on the space of possible models and clarifies which possible models, if any, are consistent with the imposed constraints and with available data.

3.2. Modules

In this experiment it was assumed that flowering time of the plant was determined by a network of modules, schematically shown in Figure 1. The use of these modules was inspired by the literature on determination of flowering time in plants, such as Weller et al. (1997), Mouradov (2002), Bernier and Périlleux (2005), and Weller (2005; 2007). Nodes of possible networks were modules, each consisting of a set of chemical reactions performing a well-defined function. The modules were connected by reactions between chemical substances representing output and input signals. One module's input could be activated or inhibited by another module's output. Possible modules included:

- a circadian clock;
- a light sensing module (in this case there were no explicitly modelled chemical reactions, but just an external light input; module output can be assumed to be a concentration of a light sensitive substance such as phytochrome);
- a long day signal module combining the clock and light signals to determine day length;
- a flowering signal module with flowering promotion and inhibition substances;
- a flowering switch module determining flowering time.

It was attempted to derive parameters of the connections between these modules by genetic programming. In this example no attempt was made to discover an unknown module network structure. Activation and inhibition functions used were defined as follows.

$$activation(k_A, S) = k_A(1 - \frac{1}{1 + S^4})$$

$$inhibition(k_I, S) = k_I \frac{1}{1 + S^4})$$

$$timedactivation(k_A, k_H, S, T) = k_A(1 - \frac{(k_H/S)^4}{(k_H/S)^4 + T^4})$$

$$timedinhibition(k_I, k_H, S, T) = k_I \frac{(k_H/S)^4}{(k_H/S)^4 + T^4}$$

where:

k_A, k_I and k_H are constants; S is the concentration of an activating or inhibiting substance; T is time; the functions $timedactivation$ and $timedinhibition$ model a threshold effect of time elapsed on the strength of a hypothetical signal for flowering time, disregarding units of physical mechanisms; more elaborate models would take into account physical units and relate the flowering signal to concentrations of biochemical substances; Hill functions with powers of 4 are used for activation and inhibition threshold effects.

The following expressions for the signals between the different modules were obtained in a run of 100 individuals during 51 generations (a total of 5 100 individuals were evaluated).

$$fromClock = activation(1.70, C)$$

$$fromLight = activation(1.74, Ph)$$

$$fromFloweringSignalP = timedactivation(0.72, 804.4, P, T)$$

$$fromLongDay = activation(1.46, LD)$$

$$fromFloweringSignalI = timedinhibition(1.49, 947.9, I, T)$$

where:

C is the concentration of one of the substances of the chemical clock; Ph is the concentration of a light-sensitive substance; LD is the concentration of a long days indicating signal; P is the concentration of a substance promoting flowering; I is the concentration of a substance inhibiting flowering; T is time.

These signals were used by the modules, in addition to an internally used degradation function defined as:

$$degradation(k_D, k_M, S) = k_D \frac{S}{k_M + S})$$

where:

k_D and k_M are constants; S is the concentration of the degrading substance.

Modules updated concentrations of substances as follows.

Long Day Signal module:

$$\frac{dLD}{dt} = 0.76 \times fromClock \times fromLight - degradation(1.0, 0.5, LD)$$

Flowering Signal module:

$$\frac{dP}{dt} = fromLongDay + inhibition(1.0, I) - degradation(1.0, 0.5, P)$$

Flowering Time module:

$$FS = fromFloweringSignalP \times fromFloweringSignalI$$

where:

FS is the flowering signal determining flowering time.

Flowering time according to genotype and photoperiod predicted by this model was close to the experimental data. Fitness, or the total deviation of modelled flowering time from observed data, was 25.82, with 9 hits (modelled results practically equal to the objective function) out of a possible 14.

4. DISCUSSION AND CONCLUSIONS

In the circadian clock experiments we used a genetic programming algorithm to find a set of reactions that could produce a circadian time evolution of chemical substances similar to Goldbeter's (1996) PER model, using combinations of inhibition, transformation, synthesis and degradation to obtain a time evolution of P_0 with a period of approximately 24 hours and a good fit to the sine target function.

In one experiment no constraints were imposed on possible concentration values and the genetic programming algorithm, oblivious to physical reality, took advantage of this to come up with a solution exhibiting physically unrealistic negative concentration values for M. Also, transformation functions could be used without constraints in this experiment, so, for example, P_1 could be transformed into P_2 without decreasing its own concentration, and transformation of P_1 into P_0 increased P_1. Suitable constraints were introduced on the use of transformation functions in the equations to avoid such effects and a model with a good fit was found. Concentration values were constrained to be positive, and a solution was found with only positive values for all substances. The model still allowed some strange phenomena to occur, for example transformation of a substance (P_0 in the first equation) into itself, and a substance (P_1 in the third equation) decreasing as a result of promoting its own synthesis.

To obtain more realistic solutions, further constraints were introduced to exclude such phenomena. This produced a model similar to the original model formulated by Goldbeter (1996). It is obvious that genetic programming can successfully reverse engineer a circadian chemical clock, but that the exact solution found depends on the flexibility allowed by imposed constraints. The model studied here has the minimal characteristics needed to produce oscillatory behaviour, so it is not surprising that the genetic programming algorithm finds a solution similar to the original model. When more complex models are studied, the genetic

programming algorithm can produce several structural alternatives producing circadian clock behaviour. For example, while Zeilinger et al. (2006) have used evolution strategies to optimise model parameters, our approach could be used to explore alternative model formulations as well.

When the genetic programming algorithm was given as inputs possible modules and their connecting activation and inhibition functions, it produced parameters for these functions leading to a model with a good fit to the experimental data on flowering time of different pea mutants in different photoperiod conditions.

In this work we have shown how genetic programming can be used in a modular system at several levels. At the lowest level, possible mechanisms of a module can be inferred from the module's function. At a higher level, knowledge about the way modules fit together can be obtained from the behaviour of the whole system. The present research has established a proof of principle of application of the Emergent Models computational methodology to a three level living system using search spaces allowing only very simple possible models. In future research the methodology can be applied to more realistic biological systems, allowing more complex models in the search space.

ACKNOWLEDGMENTS

The original research for this paper was funded by the Advanced Computational Modelling Centre and the ARC Centre for Complex Systems of the University of Queensland. The principal author also thanks Kevin Gates for his much appreciated contribution as PhD advisor.

REFERENCES

Bar-Yam Y. (1997), *Dynamics of Complex Systems*, Perseus Books, Reading, MA.

Bell P., Parmenter K., Beveridge C. and Hanan J. (2003), *Hypothesis Driven Mathematical Modelling of the Flowering Network in Pea*. Project Report, University of Queensland, Brisbane, Australia.

Bernier G. and Périlleux C. (2005), A physiological overview of the genetics of flowering time control. *Plant Biotechnology Journal* 3, 3-16.

Beveridge C.A., Weller J.L., Singer S.R. and Hofer J.M.I. (2003). Axillary meristem development: budding relationships between networks controlling flowering, branching, and photoperiod responsiveness. *Plant Physiology* 131, 927-934.

Bower J.M. and Bolouri H. (eds) (2001), *Computational Modeling of Genetic and Biochemical Networks*, MIT Press, Cambridge, MA.

De Jong H. (2002), Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology* 9(1), 67–103.

Goldbeter A. (1996), *Biochemical Oscillations and Cellular Rhythms – the molecular bases of periodic and chaotic behaviour*. Cambridge University Press, Cambridge, UK.

Holland J.H. (1998), *Emergence: from chaos to order*. Oxford University Press, Oxford.

Koza J.R. (1992), *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA.

Koza J.R., Bennet III F.H., Andre D. and Keane M.A. (1999), *Genetic Programming III: Darwinian invention and problem solving*. Morgan Kaufmann, San Francisco, CA.

Luke S. (2002), *ECJ: an evolutionary computation and genetic programming system*. Retrieved 2 January 2005, from http://cs.gmu.edu/~eclab/projects/ecj/docs/.

Mouradov A., Cremer F. and Coupland G. (2002), Control of flowering time: interacting pathways as a basis for diversity. *The Plant Cell*, supp. 2002, S111-S130, American Society of Plant Biologists.

Ptashne M. and Gann A. (2002), *Genes and Signals*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

Stolk H.J. (2005), *Emergent Models in Hierarchical and Distributed Simulation of Complex Systems*. PhD Thesis, University of Queensland, Brisbane, Australia, available at http://www.DioneComplexSystems.com/EmergentModels.html .

Stolk H.J. and Hanan J. (2007), Discovering genetic regulatory network models in *Pisum sativum*. *MODSIM07*, Christchurch, New Zealand.

Weller J.L. (2005), Mobile flowering signals in pea. *Flowering Newsletter* 40, 39-42.

Weller J.L. (2007), Update on the genetics of flowering. *Pisum Genetics* 39, 1-8.

Weller J.L., Reid J.B., Taylor S.A. and Murfet I.C. (1997), The genetic control of flowering in pea. *Trends in Plant Science* 2, 412-418.

Zeilinger M.N., Farré E.M., Taylor S.R., Kay S.A. and Doyle III F.J. (2006), A novel computational model of the circadian clock in *Arabidopsis* that incorporates PRR7 and PRR9. *Molecular Systems Biology*, article number 58, EMBO and Nature Publishing Group.