BBSI Project Proposal

IDE for BioNetGen

Yao Sun, Computer Science, University of Pittsburgh

Abstract

In this project, we will try to build an IDE for the BioNetGen. It is necessary since currently there are only web-based(e.g. GetBonNie) and graphic-based(e.g. RuleBuilder) interface for building a BioNetGen model, and they all have some limitations. In addition, we will try to introduce another visualization method of reaction models – influence map – into our rule based networks to facilitate biologists' modeling process.

Introduction

Up to now, the research in the biology fields are still mostly based on the experiments. Experiments have their own advantages, but they also have their disadvantages. They are expensive and time-consuming. So as a result, we may consider such a problem. Is there a way to model and simulate the biological system systematically? So that we can predict some results through the system and make some controls on them. Rule based modeling of biochemical systems is a good solution. It uses rules to represent and simulate the interactions of molecules in biochemical regulatory networks. BioNetGen is such a language or tool that builds and simulates such modelings. The input of the BioNetGen is a program source code like text file (Fig 1), which describes all the aspects of the biology network model to be built, such as the molecules, proteins, reaction rules and so on. BioNetGen can then builds and simulates the wanted reaction network through the specified rules and gives some simulation results.

In order to let the biologists to have a full view of the molecules and rules in the text specified model and facilitate them with understanding and analyzing the model when modeling, contact map is created. Contact map (Fig 2) is a map that depicts all the molecules and rules within the specified model. Every model has its own contact map. It is generated automatically by program.

Motivation

It seems perfect of BioNetGen and the current contact map system. But while in the modeling process, people will find that it's a pain to write a model in text without instant feedback, they will not know where there's a mistake in the model until after it is executed. It's annoying to debug a model through looking into the intermediate .net file. And the simulation results need to be imported to other programs to view. So a comfortable developing environment is necessary.

Currently there are web-based(e.g. GetBonNie) and graphic-based(e.g. RuleBuilder) interface for building a BioNetGen model, but they all have some limitations. GetBonNie works on a remote server, it is a step-by-step developing environment, thus it works slow. RuleBuilder also has some limitations in presenting the rules satisfactorily. So there's actually a need to have a comfortable IDE for BioNetGen.

Methodology and What Has Been Done

In order to achieve the previously stated goals, that is to facilitate the modelers' modeling processes, we will develop an IDE including interactive contact maps/influence maps to give instant feedbacks to modelers.

Currently we have successfully built a prototype of the IDE with interactive contact map. An outline of the prototype is shown in Fig 3.

The system mainly has 3 areas: text editing area(left), contact map area(middle) and related information area(right). And the related information area has 2 sub areas, which are rule information area(the upper subwindow) and protein information area(the lower subwindow).

When we read in an input file from the menu or directly input model specifications in the text editing area, then the file is saved and the "Show Map" button is pressed, a dynamically generated contact map will show up in the middle contact map area. This contact map is clickable, so when a particular molecule of the contact map is clicked, the 2d amino acids from the UniProt database will show up in the protein information area, and when a particular rule is clicked, the graphical and textual representation of the rules are shown in the rule information area. And also the preconditions of the selected rule are highlighted in the contact map.

More Work

In this summer, we have 2 parts of work to do.
(1) Add more features to the prototype of the IDE.
We will have to have a more comfortable IDE to be developed.

- We will have to make the prototype compatible with all the BNGL inputs.

- The Interface of the program should be more customizable.
- Context highlighting when corresponding rule/molecule/components is clicked in the contact map.
- To support the debug information and simulation process
- To support the view of the results after simulation

(2) Extend the influence map to the rule-based model.
Influence graph is in fact an abstraction of complex reaction networks, it represents the activation and inhibition effects within the reaction models, it is first introduced for the analysis of gene expression in the setting of gene regulatory networks. Our first step would be to formalize the influence map within the rule-based models. Given the definitions, we then would be able to visualize it for the corresponding models. Finally we hope to integrate the influence map into the IDE. Then we will have two alternating views of the reaction models. Thus the users can choose from different aspects of views of the models to have better understanding of the models.

Summary

In summary, our project aims to build an IDE for the BioNetGen to help people building their models through giving instant feedback and integrate all the working process into one interface. We believe with different aspects of visualization of the models, modelers would be more comfortable with what they have been building and found errors or mistakes more easily.

Appendix

```
begin reaction rules
# Ligand-receptor binding
1 EGFR(L,CR1) + EGF(R) <-> EGFR(L!1,CR1).EGF(R!1) kp1, km1
# Receptor-aggregation
2 EGFR(L!+,CR1) + EGFR(L!+,CR1) <-> EGFR(L!+,CR1!1).EGFR(L!+,CR1!1) kp2,km2
# Transphosphorylation of EGFR by RTK
3 EGFR(CR1!+,Y1068~U) -> EGFR(CR1!+,Y1068~P) kp3
# Dephosphorylation
4 EGFR(Y1068~P) -> EGFR(Y1068~U) km3
# Grb2 binding to pY1068
5 EGFR(Y1068~P) + Grb2(SH2) <-> EGFR(Y1068~P!1).Grb2(SH2!1) kp4,km4
# Grb2 binding to Sos1
6 Grb2(SH3) + Sos1(PxxP) <-> Grb2(SH3!1).Sos1(PxxP!1) kp5,km5
# Receptor dimer internalization/degradation
7 EGF(R!1).EGF(R!2).EGFR(L!1,CR1!3).EGFR(L!2,CR1!3) -> Trash() deg \
DeleteMolecules
end reaction rules
```

Fig 1 Part of Sample BioNetGen Input File

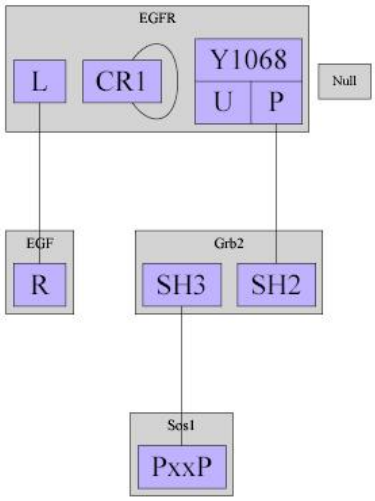

Fig2 Contact Map of EGFR Model

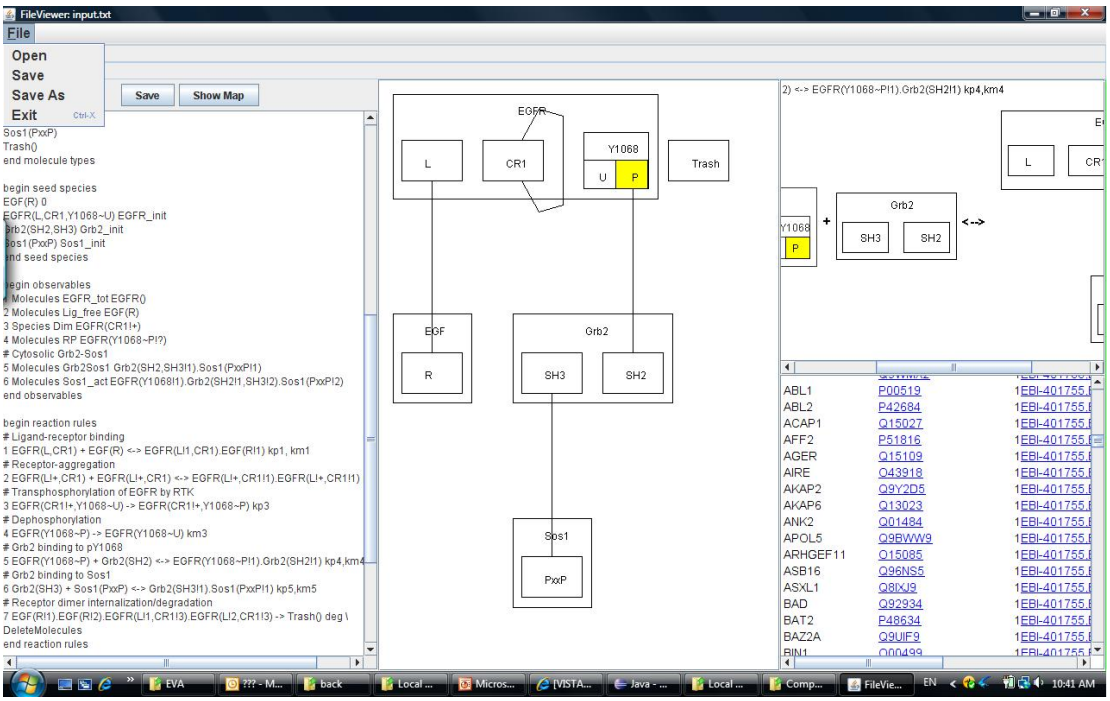

Fig 3. Outline of the Prototype