

Genome Simulator

Dr.Itsik Peer^{1,*}, Dr.Pier Palamara² and Umang Patel²

¹Department of Computer Science, Columbia University.

²Department of Computer Science, Columbia University

ABSTRACT

Motivation: A Genome Simulator is a program to simulate sequences drawn from population . Simulator made at University of Michigan had no graphical user interface . So apart from developing graphical user interface we developed our own language to simulate sequences drawn from population .

Results: The code and manual for simulator can be found at <https://github.com/17patelumang/GenomeSimulator.git>

Contact:

itsik@cs.columbia.edu , pier@cs.columbia.edu , ujp2001@columbia.edu

admixtureGeneration is the name of generation , 15 is the value of generation and offset value is 2

2) node

SYNTAX: Node_name= node(generation_variable)

Each population in our model is represented by a node.

generation_variable - variable of generation as described above .

EXAMPLE: AfricanPopulation=node(admixtureGeneration);
AfricanPopulation is the name of node , admixtureGeneration is the name of generation variable

1 INTRODUCTION

A Genome Simulator is a program to simulate sequences drawn from population. The purpose of this program is to simulate sequences on the whole genome scale within practical time.

2 APPROACH

To make the simulator we designed our own language to describe various population and various events like migration , population split , population merging etc . To make it more user interactive we developed a graphical user interface . The simulator take file written in our syntax as input and produce a graph .

3 METHODS

The syntax and semantics of our language is as under . There are six key words .

- gen
- node
- size
- edge
- rate
- migration

1) gen

SYNTAX: generation_variable=gen(generation_value,offset_parameter)

This keyword is used to define generation.

Generation_value-value of generation ,
Offset_parameter - 1 or 2

EXAMPLE: admixtureGeneration = gen(15,2);

3) size

SYNTAX: size_variable= size(size_value,offset_parameter)

It is used to specify size of population .

size_value- value of size ,
offset_parameter - 1 or 2

EXAMPLE: size_of _Africanpopulation=size(100000,1);
size_of _Africanpopulation is the name of size variable , 15 is the value of size and offset value is 1

4) edge

SYNTAX: edgename= edge(startnode,endnode,startnode_size,endnode_size)

It is used to specify how one population interacts with other . An edge is always between two nodes .

startnode - node from which edge originates , endnode - node at which edge , startnode_size - size of starting node(can be size_variable or inline size declaration) , endnode_size - size of ending node(can be size_variable or inline size declaration)

EXAMPLE: africanTOamerican=edge(africanPopulation,americanPopulation , size(10000,1),size(98000,1));

africanTOamerican -is the name of edge , africanPopulation is the starting node variable name , americanPopulation is the ending node variable name , size(10000,1) or size(98000,1) are inline declaration of size variable

5) rate

SYNTAX: rate_name=rate(rate_number,offset_parameter)

It is used to specify the rate at which migration takes place . Do not give rate names as **R1** or **R2** as the are internally used by parser .

rate_number - it is a decimal value between 0.1 to 1 , offset_parameter - 1 or 2

EXAMPLE: rateAfricanToAmerican=rate(0.001,1);
rateAfricanToAmerican is the name of rate variable , 0.001 is the rate value
, offset is 1

6) migration

SYNTAX: migration(startingedge,endingedge,ratevariable)

It is used to specify how migration takes place from one population to another . Migration always occurs between edges

startingedge - edge name FROM which migration takes place , ending edge
- edge name TO which migration takes place , ratevariable - rate variable

EXAMPLE: migration(EuropeanPostAdmixture,AfricanPostAdmixture,rateAf-
AA);

EuropeanPostAdmixture is the starting edge name, AfricanPostAdmixture is
the ending edge name , rateAf-AA is the name of rate variable

The sample input file is as shown below (Fig 1.)

```
admixtureGeneration = gen(15,2); |
nodeEuToAA = node(admixtureGeneration);
nodeAfToAA = node(admixtureGeneration);
nodeAA = node(admixtureGeneration);

sizeEuAtAdmixture = size(100000,1);
sizeAfAtAdmixture = size(70000,1);
ancestralSizeAA = size(10000,2);

EuropeansPostAdmixture = edge(node(G0), nodeEuToAA, size(1000000), size(90000,1));
AfricansPostAdmixture = edge(node(G0), nodeAfToAA, size(700000,1), size(62000,1));
AfricanAmericans = edge(node(G0), nodeAA, size(1000000,2), ancestralSizeAA);
migrationBranchEU = edge(nodeAA, nodeEuToAA, size(2000,2));
migrationBranchAf = edge(nodeAA, nodeAfToAA, size(8000,2));

nodeSplitAfEU = node(gen(400,1));
ancientBranchEU = edge(nodeEuToAA, nodeSplitAfEU, sizeEuAtAdmixture, size(50000,1));
ancientBranchAf = edge(nodeAfToAA, nodeSplitAfEU, sizeAfAtAdmixture, size(50000,1));

nodeAncestralAll = node(gen(600));
ancestralSize = size(100000);
edge(nodeSplitAfEU, nodeAncestralAll, size(100000,1), ancestralSize);
edge(nodeAncestralAll, node(Ginf), ancestralSize);

rateAf-AA=rate(0.001,1);
migration(EuropeansPostAdmixture, AfricansPostAdmixture, rateAf-AA );
```

Fig. 1. Sample Input File

4 DISCUSSION

Features of this new simulator includes:

- Read input file and produce graph
- Create Graph and Export it in syntax of new simulator
- Add node
- Add edge
- Add migration
- Output in old Genome Simulator format by University of Michigan.

1)Read input file and produce graph

- Click on File
- Click on Open
- Browse File

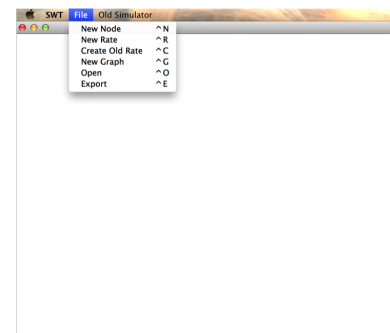


Fig. 2. Click on File

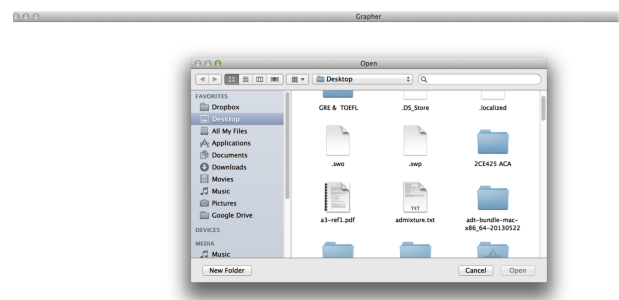


Fig. 3. Click on Open

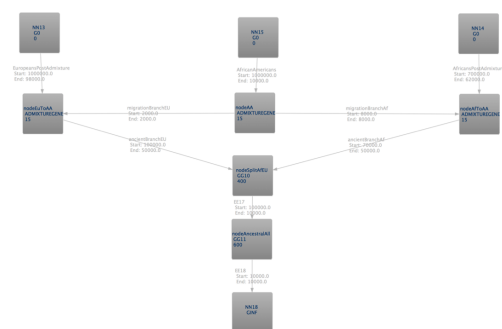
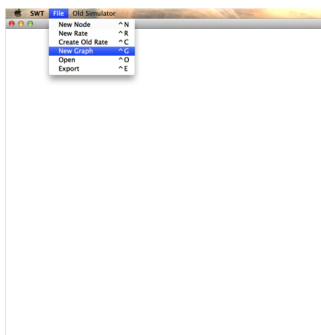
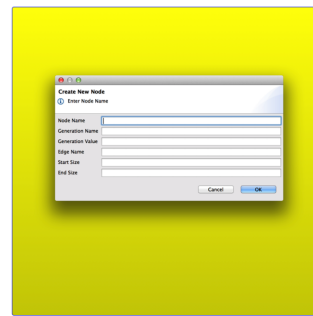
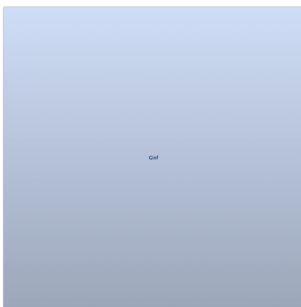
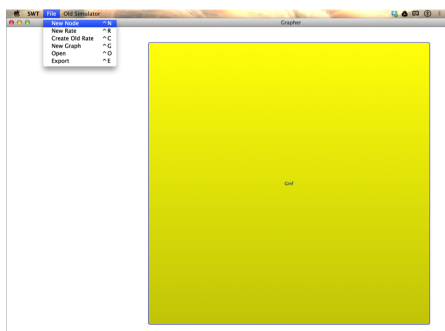
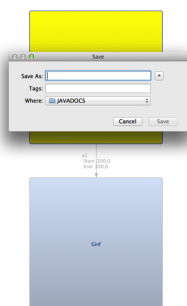
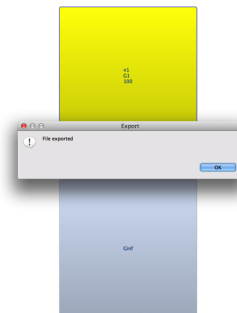


Fig. 4. Graph Displayed

2)Create Graph and Export it in syntax of new simulator

- Click on File
- Click on New Graph
- Draw as You want
- Click on File
- Click on Export

**Fig. 5.** Click on New Graph**Fig. 8.** Fill Details of node**Fig. 6.** Default Node**Fig. 7.** Click on New Node**Fig. 9.** Graph Displayed**Fig. 10.** Click on Export**Fig. 11.** File Exported

```
G1=gen(100,2);

n1=node(G1);
gnode=node(Ginf);

e1=edge(n1,gnode,size(200));
```

Fig. 12. Code Generated



Fig. 14. Click on File , New Node

3)Add node

- Click on node above which you want to add node and select New Node from File
- To add final node give Generation name as G0 and Generation Value as 0
- Each node in generation just below G0 generation , should have individual nodes in G0 generation having generation name as G0 and generation value as 0

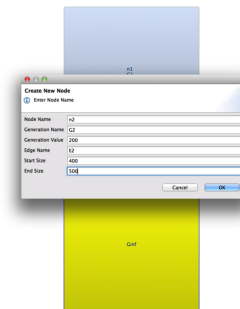


Fig. 15. Fill Details of node

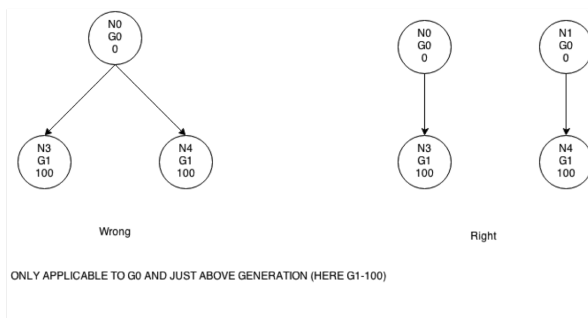


Fig. 13. Correct G0 node addition

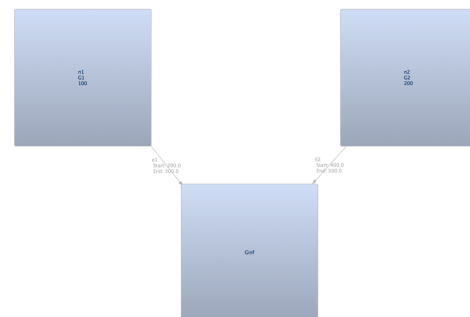


Fig. 16. Node Added

4)Add Edge

- Click on first node from which you want the edge to start
- Press CTRL
- Click on second node at which you want to end edge

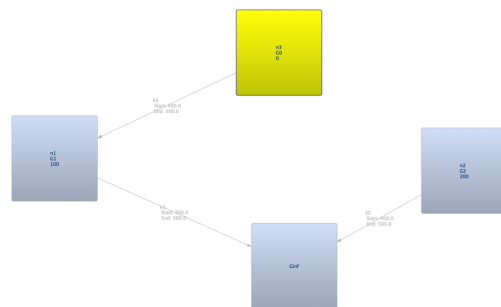


Fig. 17. Click on First node

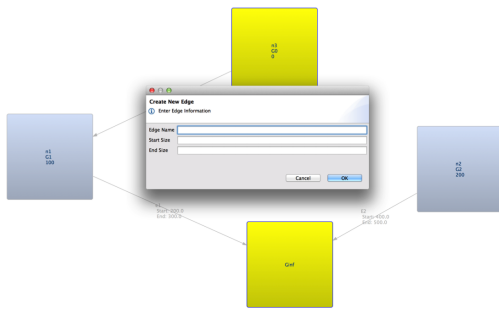


Fig. 18. Hold CTRL and Click on second node

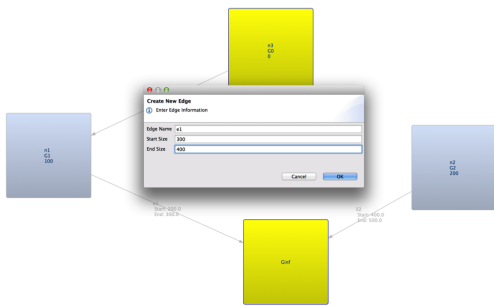


Fig. 19. Fill Details of edge

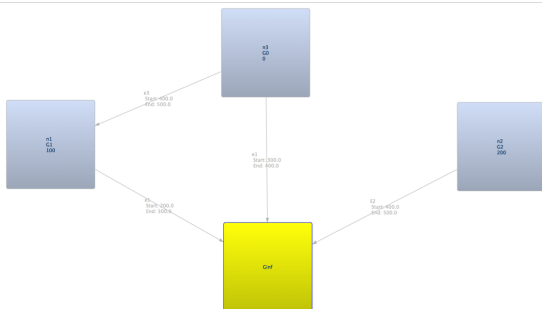


Fig. 20. Edge Added

5)Add Migration

MIGRATION CAN ADDED BETWEEN EDGES OF SAME GENERATION

- Click on File , Rate
- Enter Rate
- Click on the first edge FROM which you want to start migration
- Hold CTRL

- Select Rate from drop down list
- Click on the second edge TO which you want to end migration

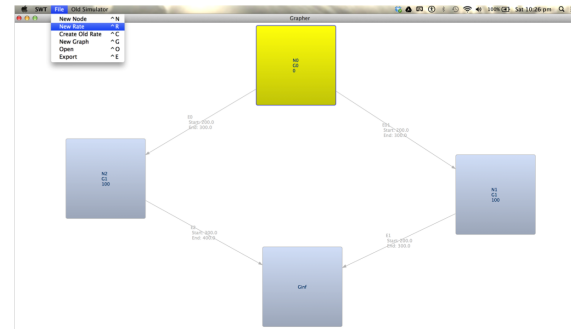


Fig. 21. Click on File , Rate

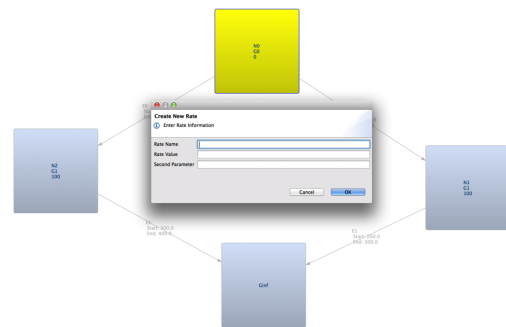


Fig. 22. Enter Rate Details

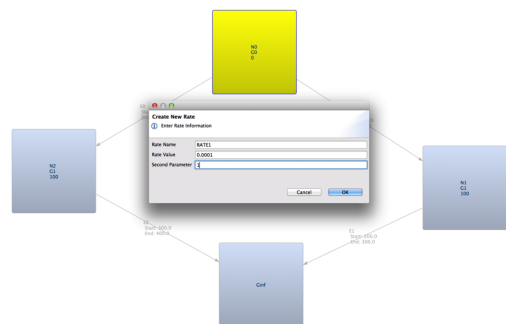


Fig. 23. Details Filled

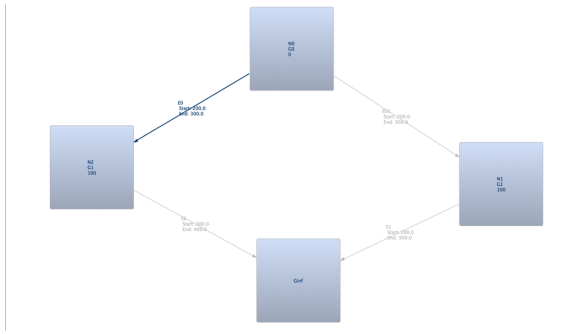


Fig. 24. Click on First Edge

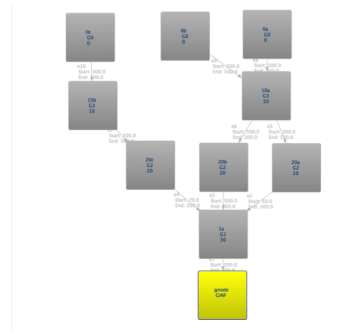


Fig. 27. Sample Graph which is to be outputted in old genome simulator format

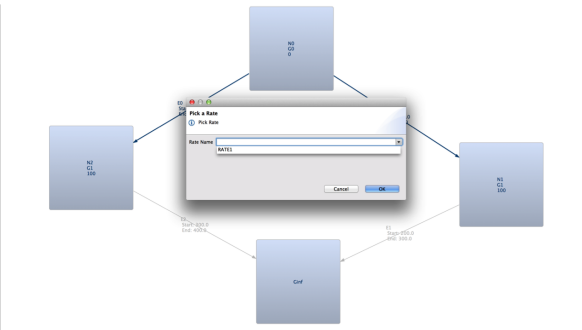


Fig. 25. Hold CTRL , Click on second Edge , Select Rate from dropdownlist

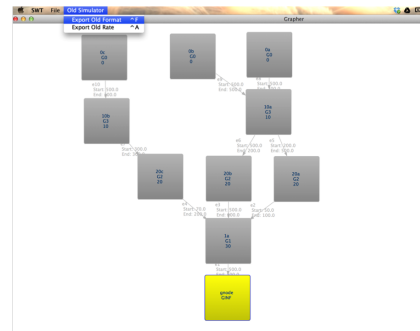


Fig. 28. Click on Old Simulator , Export Old Format

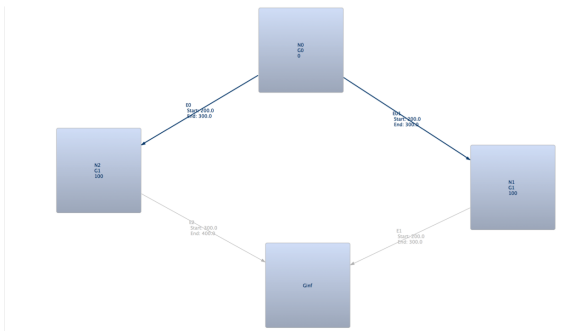


Fig. 26. Migration Added

6)Output in old Genome Simulator format by University of Michigan.

- Click on Old Simulator
- Click on Export Old Format
- Give name of file

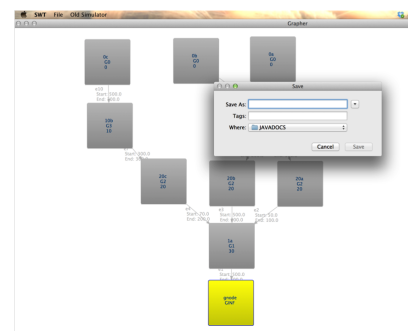


Fig. 29. Give Name of File

```
0 500 500 500
1-1 2-2 3-2
10 300 700
1-1 2-2 2-3
20 300 500 200
1-1 2-1 3-1
30 500
```

Fig. 30. Old Genome Simulator Format File

5 CONCLUSION

Using above described features any sequences can be simulated from population.

ACKNOWLEDGEMENT

I would like to thank Dr. Itsik Peer and Dr. Pier Palamara for their constant guidance.

REFERENCES

<http://www.sph.umich.edu/csg/liang/genome/GENOME-manual.pdf>