

Gene Network Analyze and Predict Tool

Version 2.1.5

iGEM USTC-Software

Chapter 1

USTC-Software 2013

We are USTC-Software, a team from University of Science and Technology of China. We will be competing in iGem 2013!

Introduction

Our application aims to simulate genetic networks. The application analyzes the stability of genetic networks after introduction of exogenous genes. Meanwhile, given the original network and specific purposes, the application traces the regulative process back and gives possible regulative patterns.

gNAP: Genetic Network Analyse and Predict

This software contains four parts, dealing with separate functions in forward and backward modeling of [GRN\(Genetic Regulatory Network\)](#) analyse.

1. Start
2. Monitor
3. Result
4. Display

Start

Start is used to prepare for the later analysis and prediction. In this part, users could input their database downloaded on Internet and sequences of exogenous gene which is needed to analyse. Also, if not input sequence in **Start**, users could also use the "Predict" function in next part.

Monitor

Monitor undertakes several functions of our software as the core methods of **gNAP**. First of them is **Analyse** function which figure out the network change when input an exogenous gene. In the same time a score presenting stability of new [GRN](#) by statist stable time and value variation for lots of times. **Analyse** result could be saw intuitively in **Result** part next. Secondly, **Predict** function use target gene exprssion to figure out possible interaction whose result could also receive in **Result**.

Result

Result is a output part which contains all results of operations used. It is easy to read each gene's information and changing consequence in this part. What's more, all gene information could be output in [SBOL](#).

Display

Display is the data visualization part of our software. To reach a more vivid output data, this part had been written in JAVA. There are three parts in **Display**: ShowRegulation, ShowChange and ShowNetwork.

This software can be built on Windows, Linux and MacOS operating platform.

For more information, please refer to our [wiki page](#).

Source Files

gNAP folder contains the command line source files in **Code** folder and GUI source files. The command line source files are written in C++ language and visualization parts are written in Java language. Both of them can be compiled across platforms.

The GUI source files are written in C++ language with Qt Creator, it can also be compiled across platforms using Qt 5.1.0, which can be found [here](#).

Database

The example database has been put into **data** folder and it can also be downloaded from RegulonDB, which can be found [here](#).

The data which used in **gNAP** is flexible. All database in those form could be read in our software.

Contacts

For any questions, feel free to contact:

Chenkun Wang(ustckun@gmail.com)

Jinyang Li(jinyangustc@gmail.com)

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

denci_tim	7
GeneIM	7
GetReady	11
GRN	15
ModleNetwork	18
PSO	19
RandomSequence	
Generate a random amino acid sequence at a specific length	22
SBOL	
Creat SBOL files outside based on gene information	23
Sequence	24

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

define.h	Define the class <code>define</code>	27
GeneIM.cpp	Statements of functions of the class <code>GeneIM</code>	29
GeneIM.h	Define the class <code>GeneIM</code>	30
GetReady.cpp	Statements of functions of the class <code>GetReady</code>	30
GetReady.h	Define the class <code>GetReady</code>	31
GRN.cpp	Statements of functions of the class <code>GRN</code>	32
GRN.h	Define the class <code>GRN</code>	32
ModleNetwork.h	??
PSO.cpp	Statements of functions of the class <code>PSO</code>	33
PSO.h	Define the class <code>PSO</code>	34
RandSeq.cpp	Statements of functions of class <code>RandSeq</code>	34
RandSeq.h	Define the class <code>RandSeq</code>	35
SBOL.cpp	Statements of functions of the class <code>SBOL</code>	35
SBOL.h	Define the class <code>SBOL</code>	36
Sequence.cpp	Statements of functions of the class <code>Sequence</code>	37
Sequence.h	Define the class <code>Sequence</code>	37
strlwr.h	??

Chapter 4

Class Documentation

4.1 denci_tim Class Reference

Public Attributes

- double * **an**
- [denci_tim](#) * **next**

The documentation for this class was generated from the following file:

- ModleNetwork.cpp

4.2 GeneIM Class Reference

```
#include <GeneIM.h>
```

Public Member Functions

- void [getGeneInformation](#) (map< string, string > dict)
- void [getPromoterIF](#) (map< string, string > dict)
- string [getID](#) ()
- string [getGeneSequence](#) ()
- string [getPromoterSequence](#) ()
- string [getPromoterName](#) ()
- string [getGeneTrueName](#) ()
- int [getLeftPosition](#) ()
- int [getRightPosition](#) ()
- void [putName](#) ()
Put gene name into gene_name[10].
- void [putInPromoterName](#) (string promoter)
Put promoter name into promoter_name.
- int [getRNA](#) ()
Get those genes which are not expressed into amino acid but RNA.
- char * [getGeneName](#) ()
- string [getGeneDescription](#) ()

Public Attributes

- int `gene_number`
The number of gene in GRN.
- char * `name`
Name used to store name in TF-TF file temporary.
- char `gene_name` [10]

Private Attributes

- string `iD`
ID in RegulonDB.
- string `gene_sequence`
Gene sequence.
- int `left_position`
Gene left position.
- int `right_position`
Gene right position.
- string `gene_description`
Gene description which contains the gene expression products.
- string `promoter_name`
Promoter name.
- string `promoter_sequence`
Promoter sequence.
- string `true_name`
Gene name which distinct capital and small letter.
- int `RNA`
Represent to RNA or not by yes(1), no(0)

4.2.1 Detailed Description

A class which contain one gene information such as gene name, gene position, gene ID, gene sequence, gene description, promoter name and promoter sequence.

1. Get gene information.
Get gene information in map of gene info and input them into corresponding variable.
2. Get promoter information.
Get promoter information in map of promoter which is constructed by gene position in TU.
3. Find RNA gene.
Some genes are not expressed to amino acid but RNA or tRNA. Avoid of aligning the AAS of RNA sequence, we find out the RNA gene.

4.2.2 Member Function Documentation

4.2.2.1 string GeneIM::getGeneDescription ()

Get gene description

Returns

gene description

4.2.2.2 void GeneIM::getGeneInformation (map< string, string > *dict*)

Get gene information from map of gene info constructed in class [GetReady](#)

Parameters

<i>map</i>	of gene info
------------	--------------

See Also

[GetReady](#)

4.2.2.3 char * GenelM::getGeneName ()

Get gene name for finding

Returns

Gene name

4.2.2.4 string GenelM::getGeneSequence ()

Get gene sequence

Returns

gene sequence

4.2.2.5 string GenelM::getGeneTrueName ()

Get gene name which distinct capital and small letter

Returns

Gene name

4.2.2.6 string GenelM::getID ()

Get ID of gene in RegulonDB

Returns

ID of gene

4.2.2.7 int GenelM::getLeftPosition ()

Get gene left position which mean the position of gene

Returns

left position of gene

4.2.2.8 void GenelM::getPromoterIF (map< string, string > dict)

Get promoter information from map of promoter sequence also constructed in class [GetReady](#)

Parameters

<i>Map</i>	of promoter sequence
------------	----------------------

See Also

[GetReady](#)

4.2.2.9 string GeneIM::getPromoterName ()

Get promoter name

Returns

promoter name

4.2.2.10 string GeneIM::getPromoterSequence ()

Get promoter sequence

Returns

promoter sequence

4.2.2.11 int GeneIM::getRightPosition ()

Get gene right position

Returns

right position of gene

4.2.3 Member Data Documentation

4.2.3.1 char GeneIM::gene_name[10]

contain gene name which not distinct capital and small letter It is used to find the right gene in map

The documentation for this class was generated from the following files:

- [GeneIM.h](#)
- [GeneIM.cpp](#)

4.3 GetReady Class Reference

```
#include <GetReady.h>
```

Public Member Functions

- void [getRegulationMatrix](#) ([GeneIM](#) temp_gene_IM[], string TF_TF_address, string TF_Gene_address)
- int [getGeneAmount](#) ()
- int [getTFAmount](#) ()

- `map< string, string > mapTFIM` (string Gene_IM_address)
- `map< string, string > mapPromoter` (string promoters_address)
- `void readTUPosition` (string TU_position_address)
a vector contains the position of each TU
- `void getGenePromoter` (GeneIM temp_gene_IM[])
- `void inputUncertainGene` ()

Public Attributes

- `double ** originalGRN`
Original GRN matrix.
- `vector< int > TU_position`
a vector contains the promoter name of each promoter
- `vector< string > promoter_name_dict`
a vector contains the promoter name of each promoter

Private Member Functions

- `void readTFTF` (GeneIM temp_gene_IM[], double **old_GRN, string TF_TF_address)
- `void readTFGene` (GeneIM temp_gene_IM[], double **old_GRN, string TF_Gene_address)
- `void addTF` (GeneIM temp_gene_IM[], string TF_Gene_address)

Private Attributes

- `vector< int > uncertain_row`
Get row number of uncertain genes.
- `vector< int > uncertain_column`
Get column number of uncertain genes.
- `int gene_amount`
Gene number of GRN.
- `int TF_amount`
Transcription Factor number of GRN.
- `int unknow`
Uncertain gene number.
- `ofstream uncertain`
Output stream of uncertain genes.

4.3.1 Detailed Description

Input the database files downloaded and get ready to fulfill all information needed to calculate.

1. Get gene regulatory network.

Get gene regulatory network from gene to gene interaction files like TF-TF and TF-Gene database on RegulonDB. Build a matrix which contains active(1),repressive(-1),uncertain(2),unknow or no interaction(0).

2. Map genes' and promoters' information.

Use map function to build a map of genes' information and promoters' detail preparing for getting sequence, position and so on.

3. Ensure the uncertain genes.

When build regulatory matrix, there will be some uncertain interactions such as "ada->ada" which has both active and repressive interaction based on the environment outside. An "uncertain" is output for users to make sure those uncertain interactions as needed.

4.3.2 Member Function Documentation

4.3.2.1 void GetReady::addTF (GeneIM temp_gene_IM[], string TF_Gene_address) [private]

Add TF not included in TF-TF regulation

Some transcription factors are not included in TF-TF regulation but included in TF-Gene regulation.

Parameters

<i>array</i>	of GeneIM 's object
<i>file</i>	address of TF-Gene regulation file

4.3.2.2 int GetReady::getGeneAmount ()

Get the amount of all genes in [GRN](#)

Returns

number of genes

4.3.2.3 void GetReady::getGenePromoter (GeneIM temp_gene_IM[])

Get promoter name and sequence

Use gene position to confirm the TU which contains it. Search promoter name in promoter info map and get its sequence.

Parameters

<i>array</i>	of GeneIM 's object
--------------	-------------------------------------

See Also

[GeneIM](#)

4.3.2.4 void GetReady::getRegulationMatrix (GeneIM temp_gene_IM[], string TF_TF_address, string TF_Gene_address)

Build [GRN](#) matrix

Parameters

<i>array</i>	of GeneIM 's objects
<i>file</i>	address of TF-TF file
<i>file</i>	address of TF-Gene file

See Also

[GeneIM](#)
[readTFTF](#)
[readTFGene](#)
[addTF](#)

4.3.2.5 int GetReady::getTFAmount ()

Get the amount of TFs in [GRN](#)

Returns

number of transcription factors

4.3.2.6 void GetReady::inputUncertainGene ()

Ensure uncertain genes interaction

File named "uncertain" having been output in getRegulationMatrix function is read to change the original matrix. Users make sure the interaction and change those uncertain genes in that file.

4.3.2.7 map< string, string > GetReady::mapPromoter (string promoters_address)

Get transcription unit position

This position is used to ensure the promoter to each gene.

Parameters

<i>file</i>	address of TU info file
-------------	-------------------------

4.3.2.8 map< string, string > GetReady::mapTFIM (string Gene_IM_address)

Construct genes' information map

Parameters

<i>file</i>	address of gene info file
-------------	---------------------------

Returns

map of gene info whose flag is gene name

4.3.2.9 void GetReady::readTFGene (GeneIM temp_gene_IM[], double ** old_GRN, string TF_Gene_address) [private]

Build TF-Gene [GRN](#)

Parameters

<i>array</i>	of GeneIM 's object
<i>original</i>	GRN matrix
<i>file</i>	address of TF-Gene regulation file

4.3.2.10 void GetReady::readTFTF (GeneIM temp_gene_IM[], double ** old_GRN, string TF_TF_address) [private]

Build TF-TF [GRN](#) and get TF name

Parameters

<i>array</i>	of GeneIM 's object
<i>original</i>	GRN matrix

<i>file</i>	address of TF-TF regulation file
-------------	----------------------------------

The documentation for this class was generated from the following files:

- [GetReady.h](#)
- [GetReady.cpp](#)

4.4 GRN Class Reference

```
#include <GRN.h>
```

Public Member Functions

- void [initialize_GRN](#) (double **old_GRN, int num_row, int num_column)
- void [construct_new_GRN](#) ([Sequence](#) reg_unit[])
- double [AminoAcidSeqAlignment](#) (std::string query, int query_size, std::string subject, int subject_size)
- double [DNASeqAlignment](#) (std::string query, int query_size, std::string subject, int subject_size)
- void [load_matrix_BLOSUM](#) ()

Public Attributes

- double ** [new_GRN](#)
New GRN matrix.

Private Member Functions

- int [AminoAcidSequenceAlignScore](#) (char t, char s)
- int [DNASequenceAlignScore](#) (char t, char s)
- double [get_max_value](#) (double a, double b, double c)
- int [get_index_of_BLOSUM50](#) (char s)

Private Attributes

- int [number_row](#)
The number of rows of original GRN.
- int [number_column](#)
The number of columns of original GRN.
- int [BLOSUM](#) [20][20]
The substitution matrix.

4.4.1 Detailed Description

Calculate sequence simialrity and construct new [GRN](#).

1. Get original Gene Regulatory Network matrix.

Get original [GRN](#) matrix from the object of class [FIXME] and add a new row and column in the end to be filled in the new relationship.

2. Get sequence simialrity.

Get sequence similarity by sequence alignment using dynamic planning with the substituion matrix BLOSUM_50.

3. Predict exogenous gene regulatory behavior.

Using simialrity vector and regulatory vectors predict the behavior of exogenous gene. And fill the correlations in [GRN](#).

4.4.2 Member Function Documentation

4.4.2.1 `double GRN::AminoAcidSeqAlignment (std::string query, int query_size, std::string subject, int subject_size)`

Align amino acid sequence.

Parameters

<i>query</i>	The query amino acid sequence.
<i>query_size</i>	The length of query amino acid sequence.
<i>subject</i>	The subject amino acid sequence.
<i>subject_size</i>	The length of subject amino acid sequence.

Returns

Percentage similarity of the two amino acid sequences.

See Also

[DNASeqAlignment](#)

4.4.2.2 `int GRN::AminoAcidSequenceAlignScore (char t, char s) [private]`

Score a alignment of two amino acids.

One amino acid comes from the query sequence. Another comes from the subject sequence. The score will be filled in the score matrix of dynamic planning.

Parameters

<i>t</i>	An amino acid comes from the subject sequence.
<i>s</i>	An amino acid comes from the query sequence.

Returns

The score of the alignment.

See Also

[DNASequenceAlignScore](#)
[AminoAcidSeqAlignment](#)

Note

The alignment score is dependent on the substitution matrix.

4.4.2.3 `void GRN::construct_new_GRN (Sequence reg_unit[])`

Construct the new [GRN](#) with exogenous gene's row and column filled.

Parameters

<i>reg_unit</i>	The object array of class Sequence . Contains original RU sequences and the query sequences.
-----------------	--

See Also

[Sequence](#)

4.4.2.4 double GRN::DNASeqAlignment (std::string *query*, int *query_size*, std::string *subject*, int *subject_size*)

Align DNA sequence.

Parameters

<i>query</i>	The query DNA sequence.
<i>query_size</i>	The length of query DNA sequence.
<i>subject</i>	The subject DNA sequence.
<i>subject_size</i>	The length of subject DNA sequence.

Returns

Percentage simialrity of the two DNA sequence.

See Also

[AminoAcidSeqAlignment](#)

4.4.2.5 int GRN::DNASequenceAlignScore (char *t*, char *s*) [private]

Score a alignment of two DNAs. One DNA comes from the query sequence. Another comes from the subject sequence. The socre will be filled in the socre matrix of dynamic planning.

Parameters

<i>t</i>	A DNA comes from the subject sequence.
<i>s</i>	A DNA comes from the query sequence.

Returns

The score of the alignment.

See Also

[AminoAcidSequenceAlignScore](#)
[DNASeqAlignment](#)

4.4.2.6 int GRN::get_index_of_BLOSUM50 (char *s*) [private]

Get the index of BLOSUM_50.

Parameters

<i>s</i>	An amino acid.
----------	----------------

Returns

The index of the amino acid in BLOSUM_50.

4.4.2.7 `double GRN::get_max_value (double a, double b, double c) [private]`

Find the biggest value.

Returns

The biggest value of the input.

4.4.2.8 `void GRN::initialize_GRN (double ** old_GRN, int num_row, int num_column)`

Initialize the object.

Parameters

<i>old_GRN</i>	Original GRN .
<i>num_row</i>	The numbers of rows of original GRN .
<i>num_column</i>	The numbers of column of original GRN .

See Also

[FIXME]

4.4.2.9 `void GRN::load_matrix_BLOSUM ()`

Read the substitution matrix BLOSUM_50.

The documentation for this class was generated from the following files:

- [GRN.h](#)
- [GRN.cpp](#)

4.5 ModleNetwork Class Reference

Public Member Functions

- void **Network_1** (double **ReguMatrix, int nx, int ny)
- void **Network_2** (double **Matr, int nx, int ny)

Public Attributes

- double ** **MaxMa**
- double * **value**

Private Member Functions

- void **RandMatrix** (double **a, double **b, const int nx, const int ny)
- double **FaNexVal** (double **Matr, double a[], const int nx, const int i, double p[], double q[], double nn[], double r[])

Private Attributes

- double * **p**
- double * **q**
- double * **r**
- double * **nn**

The documentation for this class was generated from the following files:

- ModleNetwork.h
- ModleNetwork.cpp

4.6 PSO Class Reference

```
#include <PSO.h>
```

Public Member Functions

- [PSO](#) ([ModleNetwork](#) New, int row, int column)
- void [getPrediction](#) ([ModleNetwork](#) New, int row, int column)
- void [getRange](#) (int row, int column, [ModleNetwork](#) cal)
- void [Filter](#) (int row, int column)

Public Attributes

- double [target](#) [[GENEAM](#)]
Target gene which needed to change.
- vector< double > [toPick](#)
- vector< double > [edPick](#)
- double [random_max](#) [[GENEAM](#)]
- double [random_min](#) [[GENEAM](#)]

Private Member Functions

- int [getMinLine](#) (double A[[GENEAM](#)], int column)
- double [getFitness](#) (vector< double > row_column_matrix, [ModleNetwork](#) New, int row, int column)
- double [getVariance](#) (double A[[GENEAM](#)])
- double [random](#) (double min, double max)

Private Attributes

- double ** [temp_GRN](#)
Store [GRN](#) in this vector and easy using.

4.6.1 Detailed Description

Use [PSO](#) to predict interactions between gene needed to put into [GRN](#) and original network.

[PSO](#) is Particle Swarm Optimization which is be used to find the best regulation fitting to users' goal.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 `PSO::PSO (ModleNetwork New, int row, int column)`

Initialize the [PSO](#) object Using class [ModleNetwork](#) to figure out the starting value of each genes.

Parameters

<i>an</i>	object of class ModleNetwork
<i>row</i>	number of GRN
<i>column</i>	number of GRN

See Also

[ModleNetwork](#)

4.6.3 Member Function Documentation

4.6.3.1 `void PSO::Filter (int row, int column)`

Filt predicted regualtion Classify the interactions to different degrees.

Parameters

<i>row</i>	number of GRN
<i>column</i>	number of GRN

4.6.3.2 `double PSO::getFitness (vector< double > row_column_matrix, ModleNetwork New, int row, int column)` [private]

Get fitness for each new regulation

Parameters

<i>a</i>	vector which contains the interactions between new gene and original genes
<i>an</i>	object of class ModleNetwork
<i>row</i>	number of GRN
<i>column</i>	number of GRN

Returns

the variance of prediction

See Also

[getVariance](#)

4.6.3.3 `int PSO::getMinLine (double A[GENEAM], int column)` [private]

Find out the minimum number in an array

Parameters

<i>variance</i>	for different particles in PSO method
<i>particle</i>	number

Returns

this minimum line number

4.6.3.4 void PSO::getPrediction ([ModleNetwork New](#), int *row*, int *column*)

Main function which use [PSO](#) method to predict interactions This function using [getMinLine\(\)](#), [getFitness\(\)](#), [getVariance\(\)](#) and [random\(\)](#).

Parameters

<i>an</i>	object of class ModleNetwork
<i>row</i>	number of GRN
<i>column</i>	number of GRN

See Also

[ModleNetwork](#)
[getMinLine](#)
[getFitness](#)
[getVariance](#)
[random](#)

4.6.3.5 void PSO::getRange (int *row*, int *column*, [ModleNetwork cal](#))

Get range of each gene's strength of expression Use random regulation to figure out the Maximum and Minimum expression strength. Those range have been put into `random_max` and `random_min`.

Parameters

<i>row</i>	number of GRN
<i>column</i>	number of GRN
<i>an</i>	object of class ModleNetwork

See Also

[ModleNetwork](#)

4.6.3.6 double PSO::getVariance (double *A[GENEAM]*) [private]

Figure out the variance between target and prediction

Parameters

<i>gene</i>	expression strength array
-------------	---------------------------

Returns

the variance between prediction and users' goal

4.6.3.7 double PSO::random (double *min*, double *max*) [private]

Produce a random "double" figure from "min" to "max"

Parameters

<i>Random's</i>	lower limit
<i>Random's</i>	higher limit

Returns

random figure

4.6.4 Member Data Documentation**4.6.4.1 vector<double> PSO::edPick**

New gene is interacted by genes in original [GRN](#) This vector contain the strength of interaction

4.6.4.2 double PSO::random_max[GENEAM]

Max expression value of genes in original [GRN](#) These value is used to set the users' target genes which need high expression.

4.6.4.3 double PSO::random_min[GENEAM]

Min expression value of genes in original [GRN](#) These value is used to set the users' target genes which need low expression.

4.6.4.4 vector<double> PSO::toPick

New gene interact to genes in original [GRN](#) This vector contain the strength of interaction

The documentation for this class was generated from the following files:

- [PSO.h](#)
- [PSO.cpp](#)

4.7 RandomSequence Class Reference

Generate a random amino acid sequence at a specific length.

```
#include <RandSeq.h>
```

Public Member Functions

- void **generate_random_amino_acid_sequence** (int length)

Public Attributes

- std::string **random_amino_acid_sequence**

Private Member Functions

- char **GenerateRandomAminoAcid** ()

4.7.1 Detailed Description

Generate a random amino acid sequence at a specific length.

The documentation for this class was generated from the following files:

- [RandSeq.h](#)
- [RandSeq.cpp](#)

4.8 SBOL Class Reference

Creat [SBOL](#) files outside based on gene information.

```
#include <SBOL.h>
```

Public Member Functions

- void [CreatSBOL](#) (string gene_name, string ID, string left, string right, string description, string seq)

Private Member Functions

- string [Combine](#) (string title, string detail)
- string [FormartStart](#) (string a)
- string [FormartEnd](#) (string b)

Private Attributes

- string [head](#)
head of [SBOL](#) files

4.8.1 Detailed Description

Creat [SBOL](#) files outside based on gene information.

4.8.2 Member Function Documentation

4.8.2.1 string SBOL::Combine (string title, string detail) [private]

Combine [SBOL](#) detail and its lable

Parameters

<i>lable</i>	of info
<i>lable</i>	of detail about lable

Returns

string in the formart of lable and detail

4.8.2.2 void SBOL::CreatSBOL (string gene_name, string ID, string left, string right, string description, string seq)

Create [SBOL](#) files named by gene name

Parameters

<i>string</i>	of gene name
<i>string</i>	of RegulonDB ID
<i>string</i>	of left position
<i>string</i>	of right position
<i>string</i>	of gene description
<i>string</i>	of gene sequence

4.8.2.3 string SBOL::FormartEnd (string b) [private]

Formart of End lable

Parameters

<i>string</i>	of lable in each line
---------------	-----------------------

Returns

string contain both lable and end form

4.8.2.4 string SBOL::FormartStart (string a) [private]

Formart of Start lable

Parameters

<i>string</i>	of lable in each line
---------------	-----------------------

Returns

string contain both lable and start form

The documentation for this class was generated from the following files:

- [SBOL.h](#)
- [SBOL.cpp](#)

4.9 Sequence Class Reference

```
#include <Sequence.h>
```

Public Member Functions

- void [initialize_Sequence](#) (int RU_number, std::string promoter, int p_size, std::string gene, int g_size)
- void [Translation](#) ()

Public Attributes

- std::string [gene_sequence](#)
The protein coding sequence(DNA) of an regulation unit(RU).
- std::string [promoter_sequence](#)
The promoter sequence of the regulation unit(RU).

- `std::string amino_acid_sequence`
The translation product(amino acid sequence) of the RU.
- `int regulation_unit_number`
Number of the RU.
- `int gene_size`
The length of protein coding DNA sequence.
- `int promoter_size`
The length of promoter sequence.
- `int amino_acid_sequence_size`
The length of amino acid sequence.

Private Member Functions

- `int Translate (char s)`

4.9.1 Detailed Description

Store promoter and protein coding sequence and construct regulation unit.

An object of class [Sequence](#) is a "regualtion unit". It contains a promoter sequence, a protein coding sequence, the corresponding amino acid sequence, and thier lengths. An RU is identified by a number which is also stored in the object.

4.9.2 Member Function Documentation

4.9.2.1 `void Sequence::initialize_Sequence (int RU_number, std::string promoter, int p_size, std::string gene, int g_size)`

Initializes an object.

Initialize an object and translates the gene sequence into amino acid sequence. Get the length of the amino acid sequence.

Parameters

<i>RU_number</i>	The number of the RU.
<i>promoter</i>	The promoter sequence of the RU.
<i>p_size</i>	The length of the promoter sequence.
<i>gene</i>	The protein coding sequence.
<i>g_size</i>	The length of the protein coding sequence.

See Also

[GRN](#)

4.9.2.2 `void Sequence::Translation ()`

Translates gene sequence into amino acid sequence.

Some explain of the transcription and translation process:

1. Actually, the protein expression process is:
DNA → mRNA (i.e. transcription); mRNA → protein (i.e. translation).
2. DNA has double strands, but only one strand takes part in transcription.
3. Codons are the sequence messages carried by mRNA;

Take initiation codon "AUG" for example:

—ATG— : DNA strand which doesn't take part in transcription process;

—TAC— : DNA strand which exactly takes part in transcription process;

—AUG— : mRNA strand which carries codons. In this case, it carries initiation codon, i.e. "AUG";

—TAC— : tRNA which also carries amino acid Methionine(M);

4. Owing to the DNA sequences that our database provided are the UNEXPRESSION strands, the translation process of the program can just use DNA sequence without the simulation of transcription process.

The documentation for this class was generated from the following files:

- [Sequence.h](#)
- [Sequence.cpp](#)

Chapter 5

File Documentation

5.1 define.h File Reference

Define the class define.

Macros

- #define **TFScale** 220
- #define **GENEAM** 1800
The maximum gene amount which could contain in database.
- #define **NN** 100
- #define **PETS** 128
- #define **STEP** (1.0/**PETS**)
- #define **MAXTIME** 100
- #define **INITIALVALUE** 2.5
- #define **PARTICLENUM** 30
- #define **minAccu** 0.01
- #define **Pmin** -1
- #define **Pmax** 1
- #define **Vmin** -0.01
- #define **Vmax** 0.01

5.1.1 Detailed Description

Define the class define. COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

This .h file is used to define some statistic value of factors in most command line.

The maximum TF amount which could contain in database

5.1.2 Macro Definition Documentation**5.1.2.1 #define INITIALVALUE 2.5**

Initial value of each gene in modle

See Also

[ModleNetwork](#)

5.1.2.2 #define MAXTIME 100

Interactions of [PSO](#)

See Also

[PSO](#)

5.1.2.3 #define minAccu 0.01

Minimum accuracy of [PSO](#)

See Also

[PSO](#)

5.1.2.4 #define NN 100

Interactions of [ModleNetwork](#)'s score

See Also

[ModleNetwork](#)

5.1.2.5 #define PARTICLENUM 30

Partical number of [PSO](#) method

See Also

[PSO](#)

5.1.2.6 #define PETS 128

Pets of solving differential equations

See Also

[ModleNetwork](#)

5.1.2.7 #define Pmax 1

Maximum position value of each partical in [PSO](#)

See Also

[PSO](#)

5.1.2.8 #define Pmin -1

Minimum position value of each partical in [PSO](#)

See Also

[PSO](#)

5.1.2.9 #define STEP (1.0/PETS)

Step of solving differential equations

See Also

[ModleNetwork](#)

5.1.2.10 #define Vmax 0.01

Maximum velocity value of each partical in [PSO](#)

See Also

[PSO](#)

5.1.2.11 #define Vmin -0.01

Minimum velocity value of each partical in [PSO](#)

See Also

[PSO](#)

5.2 GeneIM.cpp File Reference

Statements of funcions of the class [GeneIM](#).

```
#include "GeneIM.h"
```

5.2.1 Detailed Description

Statements of funcions of the class [GeneIM](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.3 GeneIM.h File Reference

Define the class [GeneIM](#).

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <map>
#include "define.h"
```

Classes

- class [GeneIM](#)

5.3.1 Detailed Description

Define the class [GeneIM](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.4 GetReady.cpp File Reference

Statements of functions of the class [GetReady](#).

```
#include "GeneIM.h"
#include "GetReady.h"
```

5.4.1 Detailed Description

Statements of functions of the class [GetReady](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.5 GetReady.h File Reference

Define the class [GetReady](#).

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>
#include <map>
#include <cstring>
#include "define.h"
#include "strlwr.h"
```

Classes

- class [GetReady](#)

5.5.1 Detailed Description

Define the class [GetReady](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.6 GRN.cpp File Reference

Statements of functions of the class [GRN](#).

```
#include "GRN.h"
#include "RandSeq.h"
#include <vector>
#include <string>
#include <fstream>
#include <ctime>
#include <cmath>
#include <stdlib.h>
```

Macros

- `#define GAP -8`
Linear gap penalty of amino acid sequence alignment.
- `#define GAP_2 -1`
Linear gap penalty of DNA sequence alignment.
- `#define RAND_SCALE 100`
The number of generated random sequences.
- `#define SIGMA_NUM 0.2`
Filter control determines the range of similarity to be filtered.

5.6.1 Detailed Description

Statements of functions of the class [GRN](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Li Jinyang

Date

July 26, 2013

5.7 GRN.h File Reference

Define the class [GRN](#).

```
#include <iostream>
#include <vector>
#include <fstream>
#include <cmath>
#include "Sequence.h"
```

Classes

- class [GRN](#)

5.7.1 Detailed Description

Define the class [GRN](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Li Jinyang

Date

July 26, 2013

5.8 PSO.cpp File Reference

Statements of functions of the class [PSO](#).

```
#include "PSO.h"
```

5.8.1 Detailed Description

Statements of functions of the class [PSO](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.9 PSO.h File Reference

Define the class [PSO](#).

```
#include <vector>
#include <cstdlib>
#include "define.h"
#include "ModleNetwork.h"
```

Classes

- class [PSO](#)

5.9.1 Detailed Description

Define the class [PSO](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.10 RandSeq.cpp File Reference

Statements of functions of class RandSeq.

```
#include "RandSeq.h"
#include <iostream>
#include <ctime>
#include "stdlib.h"
```

5.10.1 Detailed Description

Statements of functions of class RandSeq. COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Li Jinyang

Date

Aug. 9, 2013

5.11 RandSeq.h File Reference

Define the class RandSeq.

```
#include <iostream>
```

Classes

- class [RandomSequence](#)
Generate a random amino acid sequence at a specific length.

5.11.1 Detailed Description

Define the class RandSeq. COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Generate a random amino acid sequence at a specific length.

Version

1.0

Author

Li Jinyang

Date

Aug. 9, 2013

5.12 SBOL.cpp File Reference

Statements of functions of the class [SBOL](#).

```
#include "SBOL.h"
```

5.12.1 Detailed Description

Statements of functions of the class [SBOL](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.13 SBOL.h File Reference

Define the class [SBOL](#).

```
#include <fstream>
#include <iostream>
#include <string>
```

Classes

- class [SBOL](#)

Creat [SBOL](#) files outside based on gene information.

5.13.1 Detailed Description

Define the class [SBOL](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Wang Chenkun

Date

September 2nd, 2013

5.14 Sequence.cpp File Reference

Statements of functions of the class [Sequence](#).

```
#include "Sequence.h"
```

5.14.1 Detailed Description

Statements of functions of the class [Sequence](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Version

1.0

Author

Li Jinyang

Date

July 26, 2013

5.15 Sequence.h File Reference

Define the class [Sequence](#).

```
#include <iostream>
```

Classes

- class [Sequence](#)

5.15.1 Detailed Description

Define the class [Sequence](#). COPYRIGHT NOTICE

Distribute under BSD License

Copyright (c) 2013, iGEM Software Team of University of Science and Technology of China

All rights reserved.

Achieve the construction of a regulation unit. The object contains a promoter sequence, the length of the promoter sequence, a protein coding sequence, the length of the promoter sequence, the amino acid sequence translated from the protein coding sequence, and the length of the amino acid sequence. The regulation unit is identified by a number which is also stored in the object.

Version

1.0

Author

Li Jinyang

Date

July 26, 2013