

CPPN-NEAT

Generated by Doxygen 1.8.6

Mon Mar 3 2014 13:26:31

Contents

1	CPPN-NEAT	1
2	Namespace Index	3
2.1	Namespace List	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	ANN_USM Namespace Reference	9
5.1.1	Enumeration Type Documentation	9
5.1.1.1	gene_type	9
6	Class Documentation	11
6.1	ANN_USM::connection_gene Class Reference	11
6.1.1	Detailed Description	11
6.1.2	Member Function Documentation	11
6.1.2.1	c_g	11
6.1.2.2	kill	12
6.1.3	Member Data Documentation	12
6.1.3.1	enable	12
6.1.3.2	exist	12
6.1.3.3	in	12
6.1.3.4	innovation	12
6.1.3.5	out	13
6.1.3.6	weight	13
6.2	Function Class Reference	13
6.2.1	Detailed Description	13
6.2.2	Constructor & Destructor Documentation	13
6.2.2.1	Function	13

6.2.2.2	Function	13
6.2.3	Member Function Documentation	13
6.2.3.1	eval	13
6.2.3.2	get_name	13
6.2.3.3	get_name	14
6.3	ANN_USM::Genetic_Encoding Class Reference	14
6.3.1	Detailed Description	14
6.3.2	Member Function Documentation	15
6.3.2.1	add_connection	15
6.3.2.2	add_connection	15
6.3.2.3	add_node	16
6.3.2.4	add_node	16
6.3.2.5	change_weight	16
6.3.2.6	eval	17
6.3.2.7	JSON	17
6.3.2.8	load	17
6.3.2.9	save	18
6.3.2.10	spread_final_result	19
6.3.3	Member Data Documentation	19
6.3.3.1	input_nodes	19
6.3.3.2	Lconnection_genes	19
6.3.3.3	Lnode_genes	19
6.3.3.4	niche	19
6.3.3.5	output_nodes	19
6.3.3.6	row_orderer_list	19
6.4	ANN_USM::Niche Class Reference	20
6.4.1	Detailed Description	20
6.4.2	Member Data Documentation	20
6.4.2.1	amount_of_offspring	20
6.4.2.2	exist	20
6.4.2.3	niche_champion_position	20
6.4.2.4	organism_position	20
6.4.2.5	total_fitness	20
6.4.2.6	years	20
6.5	ANN_USM::node_gene Class Reference	21
6.5.1	Detailed Description	21
6.5.2	Constructor & Destructor Documentation	21
6.5.2.1	node_gene	21
6.5.3	Member Function Documentation	22
6.5.3.1	decrease_incoming_connection	22

6.5.3.2	eval	22
6.5.3.3	get_final_result	22
6.5.3.4	increase_incoming_connection	22
6.5.3.5	is_ready	22
6.5.3.6	kill	22
6.5.3.7	n_g	23
6.5.4	Member Data Documentation	23
6.5.4.1	accumulative_result	23
6.5.4.2	counter	23
6.5.4.3	exist	23
6.5.4.4	final_result	23
6.5.4.5	function	24
6.5.4.6	incoming_connections	24
6.5.4.7	node	24
6.5.4.8	outgoing_connections	24
6.5.4.9	row	24
6.5.4.10	type	24
6.6	ANN_USM::Population Class Reference	24
6.6.1	Detailed Description	25
6.6.2	Constructor & Destructor Documentation	25
6.6.2.1	Population	25
6.6.3	Member Function Documentation	26
6.6.3.1	compatibility	26
6.6.3.2	crossover	26
6.6.3.3	epoch	27
6.6.3.4	mutation_change_weight	28
6.6.3.5	mutation_connection	28
6.6.3.6	mutation_node	29
6.6.3.7	obtain_historical_node	29
6.6.3.8	obtain_innovation	30
6.6.3.9	obtain_row	30
6.6.3.10	put_randoms_weight	31
6.6.3.11	save	31
6.6.3.12	save_all	32
6.6.3.13	spatiation	32
6.6.4	Member Data Documentation	32
6.6.4.1	best_organism	32
6.6.4.2	champion	32
6.6.4.3	current_niches	32
6.6.4.4	fitness_champion	32

6.6.4.5	historical_innovation	33
6.6.4.6	historical_nodes	33
6.6.4.7	historical_row	33
6.6.4.8	last_innovation	33
6.6.4.9	last_niche_id	33
6.6.4.10	last_node	33
6.6.4.11	last_row	33
6.6.4.12	length	33
6.6.4.13	organisms	33
6.6.4.14	prev_niches	33
6.6.4.15	prev_organisms	33
6.6.4.16	row_orderer_list	33
6.6.4.17	total_niches	34
7	File Documentation	35
7.1	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/CPPN-NEAT.hpp File Reference	35
7.1.1	Function Documentation	36
7.1.1.1	operator<<	36
7.2	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/fitness.hpp File Reference	37
7.2.1	Function Documentation	37
7.2.1.1	fitness	37
7.3	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/function.hpp File Reference	38
7.4	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/genetic_encoding.hpp File Reference	39
7.4.1	Function Documentation	40
7.4.1.1	operator<<	40
7.5	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/user_definitions.hpp File Reference	41
7.5.1	Macro Definition Documentation	41
7.5.1.1	DISTANCE_CONST_1	41
7.5.1.2	DISTANCE_CONST_2	42
7.5.1.3	DISTANCE_CONST_3	42
7.5.1.4	DISTANCE_CONST_4	42
7.5.1.5	DISTANCE_THRESHOLD	42
7.5.1.6	FUNCTION_NUM	42
7.5.1.7	GENERATIONS	42
7.5.1.8	LARGE_POPULATION_DISCRIMINATOR	42
7.5.1.9	LARGER_POPULATIONS_PROBABILITY_ADDING_NEW_CONNECTION	42
7.5.1.10	LARGER_POPULATIONS_PROBABILITY_ADDING_NEW_NODE	42
7.5.1.11	PERCENT_MUTATION_CONNECTION	42

7.5.1.12	PERCENTAGE_OFFSPRING_WITHOUT_CROSSOVER	42
7.5.1.13	POPULATION_MAX	42
7.5.1.14	PROBABILITY_CONNECTION_WEIGHT_MUTATING	43
7.5.1.15	PROBABILITY_INTERSPECIES_MATING	43
7.5.1.16	SMALLER_POPULATIONS_PROBABILITY_ADDING_NEW_CONNECTION . .	43
7.5.1.17	SMALLER_POPULATIONS_PROBABILITY_ADDING_NEW_NODE	43
7.6	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/README.md File Reference . . .	43
7.7	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/CPPN-NEAT.cpp File Reference	43
7.7.1	Macro Definition Documentation	44
7.7.1.1	CPPN_NEAT_CPP	44
7.7.2	Function Documentation	44
7.7.2.1	operator<<	44
7.8	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/fitness.cpp File Reference . .	44
7.8.1	Macro Definition Documentation	44
7.8.1.1	FITNESS_CPP	44
7.8.2	Function Documentation	45
7.8.2.1	fitness	45
7.9	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/function.cpp File Reference . .	45
7.10	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/genetic_encoding.cpp File Reference	46
7.10.1	Macro Definition Documentation	47
7.10.1.1	GENETIC_ENCODING_CPP	47
7.10.2	Function Documentation	47
7.10.2.1	operator<<	47
7.11	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/run_CPPN-NEAT.cpp File Reference	47
7.11.1	Function Documentation	48
7.11.1.1	main	48
7.11.1.2	print_usage	48
7.12	/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/test_genetic_encoding.cpp File Reference	48
7.12.1	Macro Definition Documentation	49
7.12.1.1	T_G_E_CPP	49
7.12.2	Function Documentation	49
7.12.2.1	main	49

Chapter 1

CPPN-NEAT

Compiling

Just execute the *makefile* in order to compile the program

```
"" $ cd CPPN-NEAT $ make clean $ make ""
```

Usage

We've incorporated both of the minimal actions related to evolving a network, i.e. training a certain network & using it.

Training a network

In order to train a network (by default *genetic_encodings/default_00.genetic_encoding*) you must use the flag **-t** followed by the file to train

```
"" $ cd CPPN-NEAT $ ./executables/CPPN-NEAT -t genetic_encodings/default_00.genetic_encoding ""
```

This will print in the console the

This will create a file in the same folder with the same name (with extension ***out.genetic_encoding***) which contain the network structure.

- We use JSON as a network structure in order to make a visual panel in the future using *html* and *javascript*.

Use a trained network

```
"" $ cd CPPN-NEAT $ $ # In order to train a network file (extension .genetic_encoding) $ ./executables/CPPN-NEAT -t genetic_encodings/default_00.genetic_encoding $ $ # In order to execute the trained network $ ./executables/CPPN-NEAT -x genetic_encodings/default_00.out.genetic_encoding -v 0 0 ""
```


Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

ANN_USM	9
-----------------------------------	---

Chapter 3

Class Index

3.1 Class List

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

ANN_USM::connection_gene	11
Function	13
ANN_USM::Genetic_Encoding	14
ANN_USM::Niche	20
ANN_USM::node_gene	21
ANN_USM::Population	24

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/CPPN-NEAT.hpp	35
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/fitness.hpp	37
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/function.hpp	38
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/genetic_encoding.hpp	39
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/user_definitions.hpp	41
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/CPPN-NEAT.cpp	43
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/fitness.cpp	44
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/function.cpp	45
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/genetic_encoding.cpp	46
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/run_CPPN-NEAT.cpp	47
/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/test_genetic_encoding.cpp	48

Chapter 5

Namespace Documentation

5.1 ANN_USM Namespace Reference

Classes

- class [Niche](#)
- class [Population](#)
- class [connection_gene](#)
- class [node_gene](#)
- class [Genetic_Encoding](#)

Enumerations

- enum [gene_type](#) { [INPUT](#), [HIDDEN](#), [OUTPUT](#) }

5.1.1 Enumeration Type Documentation

5.1.1.1 enum ANN_USM::gene_type

Enumerator

INPUT
HIDDEN
OUTPUT

Definition at line 20 of file genetic_encoding.hpp.

Chapter 6

Class Documentation

6.1 ANN_USM::connection_gene Class Reference

```
#include <genetic_encoding.hpp>
```

Public Member Functions

- void `c_g` (int `innovation`, int `in`, int `out`, double `weight`, bool `enable`)
- void `kill` ()

Public Attributes

- int `innovation`
- int `in`
- int `out`
- bool `enable`
- double `weight`
- bool `exist`

6.1.1 Detailed Description

Definition at line 31 of file `genetic_encoding.hpp`.

6.1.2 Member Function Documentation

6.1.2.1 void `connection_gene::c_g` (int *innovation*, int *in*, int *out*, double *weight*, bool *enable*)

Definition at line 15 of file `genetic_encoding.cpp`.

6.1.3.5 int ANN_USM::connection_gene::out

Definition at line 40 of file genetic_encoding.hpp.

6.1.3.6 double ANN_USM::connection_gene::weight

Definition at line 42 of file genetic_encoding.hpp.

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

- /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/genetic_encoding.hpp
- /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/genetic_encoding.cpp

6.2 Function Class Reference

```
#include <function.hpp>
```

Public Member Functions

- [Function](#) ()
- [Function](#) (string)
- double [eval](#) (double)
- string [get_name](#) (int function)
- string [get_name](#) ()

6.2.1 Detailed Description

Definition at line 11 of file function.hpp.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Function::Function ()

Definition at line 5 of file function.cpp.

6.2.2.2 Function::Function (string *function_name*)

Definition at line 7 of file function.cpp.

6.2.3 Member Function Documentation

6.2.3.1 double Function::eval (double *input*)

Definition at line 36 of file function.cpp.

6.2.3.2 string Function::get_name (int *function*)

Definition at line 41 of file function.cpp.

Here is the caller graph for this function:



6.2.3.3 string Function::get_name ()

Definition at line 59 of file function.cpp.

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

- /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/function.hpp
- /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/function.cpp

6.3 ANN_USM::Genetic_Encoding Class Reference

```
#include <genetic_encoding.hpp>
```

Public Member Functions

- void [add_node](#) (int node, int row, [gene_type](#) type, string function)
- void [add_node](#) ([node_gene](#) node)
- void [add_connection](#) (int innovation, int in, int out, double weight, bool enable)
- void [add_connection](#) ([connection_gene](#) conn)
- void [change_weight](#) (int innovation, double weight)
- void [save](#) (char path[])
- void [load](#) (char path[])
- void [spread_final_result](#) (int node, double value)
- string [JSON](#) ()
- vector< double > [eval](#) (vector< double > inputs)

Public Attributes

- vector< [connection_gene](#) > [Lconnection_genes](#)
- vector< [node_gene](#) > [Lnode_genes](#)
- vector< int > [input_nodes](#)
- vector< int > [output_nodes](#)
- vector< int > [row_orderer_list](#)
- int [niche](#)

6.3.1 Detailed Description

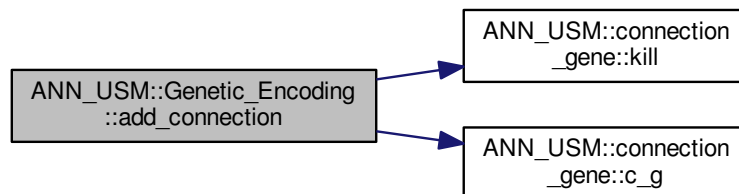
Definition at line 88 of file genetic_encoding.hpp.

6.3.2 Member Function Documentation

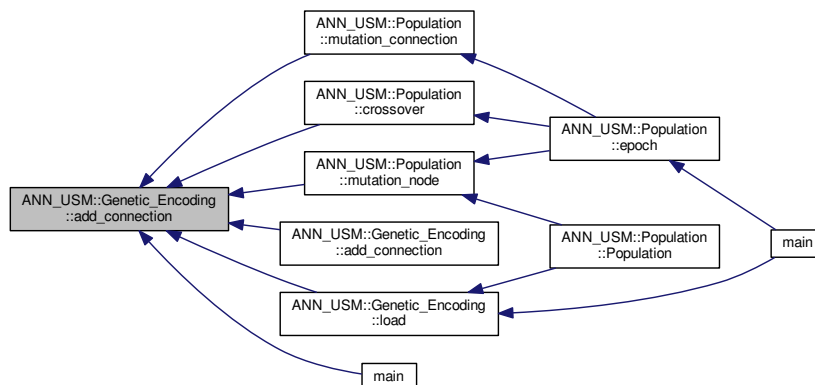
6.3.2.1 void Genetic_Encoding::add_connection (int *innovation*, int *in*, int *out*, double *weight*, bool *enable*)

Definition at line 100 of file genetic_encoding.cpp.

Here is the call graph for this function:



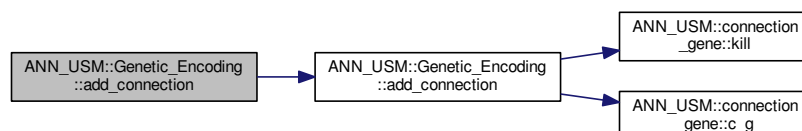
Here is the caller graph for this function:



6.3.2.2 void Genetic_Encoding::add_connection (connection_gene *conn*)

Definition at line 95 of file genetic_encoding.cpp.

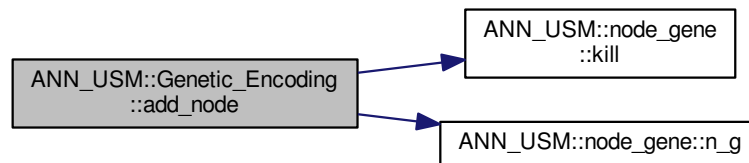
Here is the call graph for this function:



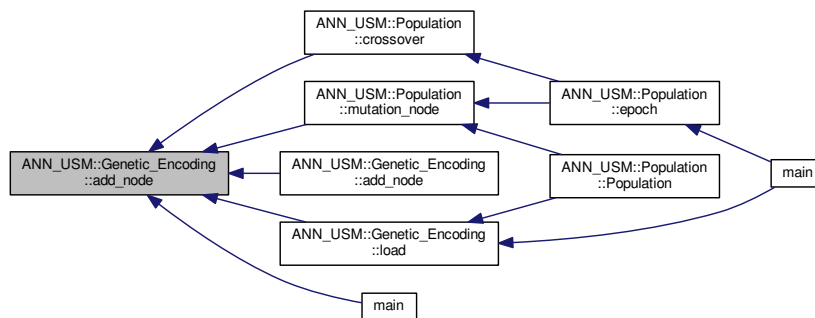
6.3.2.3 void Genetic_Encoding::add_node (int node, int row, gene_type type, string function)

Definition at line 145 of file genetic_encoding.cpp.

Here is the call graph for this function:



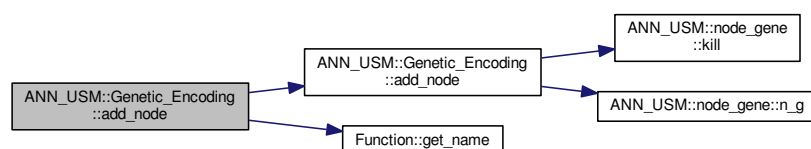
Here is the caller graph for this function:



6.3.2.4 void Genetic_Encoding::add_node (node_gene node)

Definition at line 140 of file genetic_encoding.cpp.

Here is the call graph for this function:

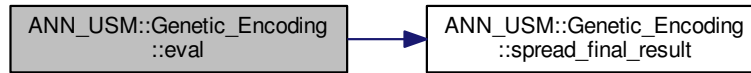


6.3.2.5 void ANN_USM::Genetic_Encoding::change_weight (int innovation, double weight)

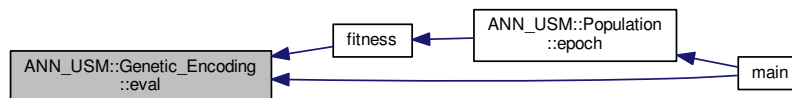
6.3.2.6 `vector< double > Genetic_Encoding::eval (vector< double > inputs)`

Definition at line 186 of file `genetic_encoding.cpp`.

Here is the call graph for this function:



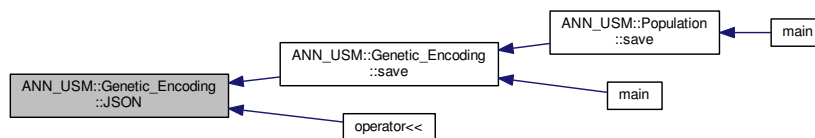
Here is the caller graph for this function:



6.3.2.7 `string Genetic_Encoding::JSON ()`

Definition at line 246 of file `genetic_encoding.cpp`.

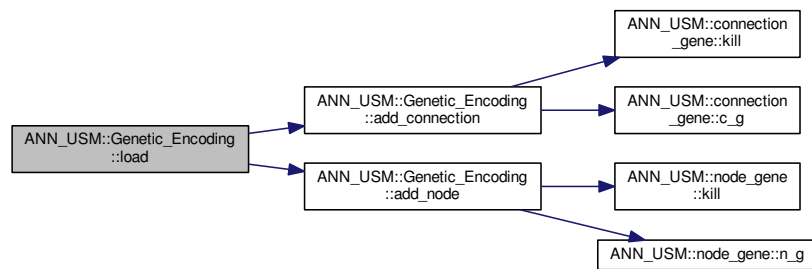
Here is the caller graph for this function:



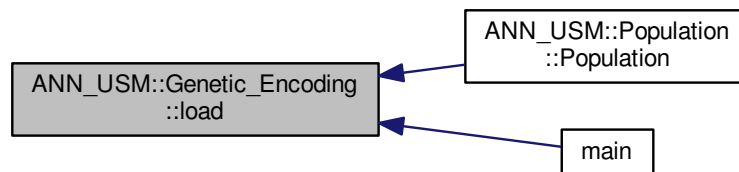
6.3.2.8 `void Genetic_Encoding::load (char path[])`

Definition at line 298 of file `genetic_encoding.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



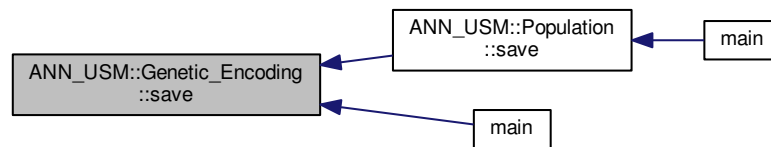
6.3.2.9 void Genetic_Encoding::save (char path[])

Definition at line 290 of file genetic_encoding.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.2.10 void Genetic_Encoding::spread_final_result (int node, double value)

Definition at line 224 of file `genetic_encoding.cpp`.

Here is the caller graph for this function:



6.3.3 Member Data Documentation

6.3.3.1 vector<int> ANN_USM::Genetic_Encoding::input_nodes

Definition at line 111 of file `genetic_encoding.hpp`.

6.3.3.2 vector<connection_gene> ANN_USM::Genetic_Encoding::Lconnection_genes

Definition at line 108 of file `genetic_encoding.hpp`.

6.3.3.3 vector<node_gene> ANN_USM::Genetic_Encoding::Lnode_genes

Definition at line 109 of file `genetic_encoding.hpp`.

6.3.3.4 int ANN_USM::Genetic_Encoding::niche

Definition at line 115 of file `genetic_encoding.hpp`.

6.3.3.5 vector<int> ANN_USM::Genetic_Encoding::output_nodes

Definition at line 112 of file `genetic_encoding.hpp`.

6.3.3.6 vector<int> ANN_USM::Genetic_Encoding::row_orderer_list

Definition at line 113 of file `genetic_encoding.hpp`.

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

- [/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/genetic_encoding.hpp](#)
- [/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/genetic_encoding.cpp](#)

6.4 ANN_USM::Niche Class Reference

```
#include <CPPN-NEAT.hpp>
```

Public Attributes

- int [years](#)
- int [niche_champion_position](#)
- int [amount_of_offspring](#)
- bool [exist](#)
- double [total_fitness](#)
- vector< int > [organism_position](#)

6.4.1 Detailed Description

Definition at line 18 of file CPPN-NEAT.hpp.

6.4.2 Member Data Documentation

6.4.2.1 int ANN_USM::Niche::amount_of_offspring

Definition at line 24 of file CPPN-NEAT.hpp.

6.4.2.2 bool ANN_USM::Niche::exist

Definition at line 26 of file CPPN-NEAT.hpp.

6.4.2.3 int ANN_USM::Niche::niche_champion_position

Definition at line 23 of file CPPN-NEAT.hpp.

6.4.2.4 vector<int> ANN_USM::Niche::organism_position

Definition at line 30 of file CPPN-NEAT.hpp.

6.4.2.5 double ANN_USM::Niche::total_fitness

Definition at line 28 of file CPPN-NEAT.hpp.

6.4.2.6 int ANN_USM::Niche::years

Definition at line 22 of file CPPN-NEAT.hpp.

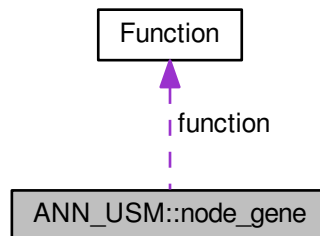
The documentation for this class was generated from the following file:

- [/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/CPPN-NEAT.hpp](#)

6.5 ANN_USM::node_gene Class Reference

```
#include <genetic_encoding.hpp>
```

Collaboration diagram for ANN_USM::node_gene:



Public Member Functions

- [node_gene](#) ()
- void [n_g](#) (int [node](#), int [row](#), [gene_type](#) type, string [function](#))
- void [kill](#) ()
- void [increase_incoming_connection](#) ()
- void [decrease_incoming_connection](#) ()
- void [eval](#) (double value)
- double [get_final_result](#) ()
- bool [is_ready](#) ()

Public Attributes

- int [row](#)
- int [node](#)
- int [incoming_connections](#)
- int [counter](#)
- bool [exist](#)
- double [accumulative_result](#)
- double [final_result](#)
- vector< int > [outgoing_connections](#)
- [gene_type](#) type
- [Function](#) * [function](#)

6.5.1 Detailed Description

Definition at line 50 of file genetic_encoding.hpp.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 node_gene::node_gene ()

Definition at line 34 of file genetic_encoding.cpp.

6.5.3 Member Function Documentation

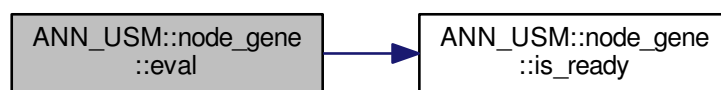
6.5.3.1 void node_gene::decrease_incoming_connection ()

Definition at line 71 of file genetic_encoding.cpp.

6.5.3.2 void node_gene::eval (double *value*)

Definition at line 76 of file genetic_encoding.cpp.

Here is the call graph for this function:



6.5.3.3 double node_gene::get_final_result ()

Definition at line 56 of file genetic_encoding.cpp.

6.5.3.4 void node_gene::increase_incoming_connection ()

Definition at line 66 of file genetic_encoding.cpp.

6.5.3.5 bool node_gene::is_ready ()

Definition at line 61 of file genetic_encoding.cpp.

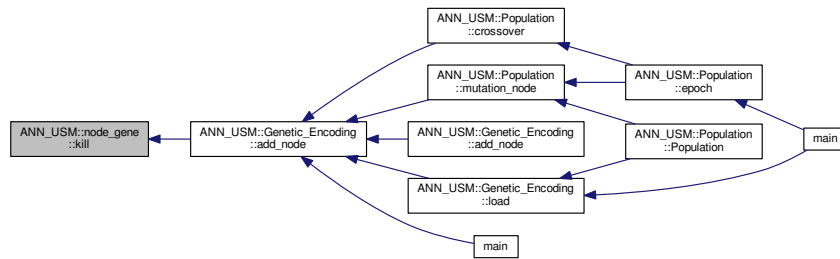
Here is the caller graph for this function:



6.5.3.6 void node_gene::kill ()

Definition at line 51 of file genetic_encoding.cpp.

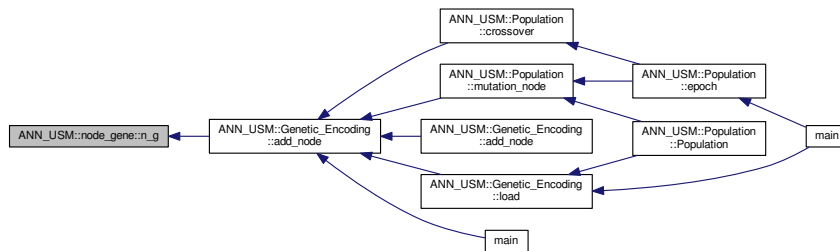
Here is the caller graph for this function:



6.5.3.7 void node_gene::n_g (int node, int row, gene_type type, string function)

Definition at line 42 of file genetic_encoding.cpp.

Here is the caller graph for this function:



6.5.4 Member Data Documentation

6.5.4.1 double ANN_USM::node_gene::accumulative_result

Definition at line 73 of file genetic_encoding.hpp.

6.5.4.2 int ANN_USM::node_gene::counter

Definition at line 69 of file genetic_encoding.hpp.

6.5.4.3 bool ANN_USM::node_gene::exist

Definition at line 71 of file genetic_encoding.hpp.

6.5.4.4 double ANN_USM::node_gene::final_result

Definition at line 74 of file genetic_encoding.hpp.

6.5.4.5 `Function*` `ANN_USM::node_gene::function`

Definition at line 81 of file `genetic_encoding.hpp`.

6.5.4.6 `int` `ANN_USM::node_gene::incoming_connections`

Definition at line 68 of file `genetic_encoding.hpp`.

6.5.4.7 `int` `ANN_USM::node_gene::node`

Definition at line 67 of file `genetic_encoding.hpp`.

6.5.4.8 `vector<int>` `ANN_USM::node_gene::outgoing_connections`

Definition at line 77 of file `genetic_encoding.hpp`.

6.5.4.9 `int` `ANN_USM::node_gene::row`

Definition at line 66 of file `genetic_encoding.hpp`.

6.5.4.10 `gene_type` `ANN_USM::node_gene::type`

Definition at line 79 of file `genetic_encoding.hpp`.

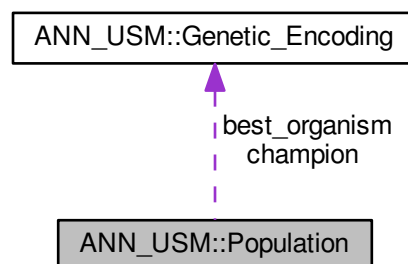
The documentation for this class was generated from the following files:

- [/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/genetic_encoding.hpp](#)
- [/home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/genetic_encoding.cpp](#)

6.6 `ANN_USM::Population` Class Reference

```
#include <CPPN-NEAT.hpp>
```

Collaboration diagram for `ANN_USM::Population`:



Public Member Functions

- [Population](#) (char path[])
- void [save_all](#) (char path[])
- void [save](#) (char path[])
- void [epoch](#) ()
- void [spatiation](#) ()
- int [obtain_historical_node](#) (int initial_in, int initial_out)
- int [obtain_innovation](#) (int in, int out)

Obtain the (historical) innovation number between two nodes.
- int [obtain_row](#) (int node, int node_initial_in, int node_initial_out)
- double [compatibility](#) ([Genetic_Encoding](#) orgm1, [Genetic_Encoding](#) orgm2)

Measure the distance between two organism.
- [Genetic_Encoding](#) [mutation_node](#) ([Genetic_Encoding](#) organism)
- [Genetic_Encoding](#) [mutation_connection](#) ([Genetic_Encoding](#) organism)
- [Genetic_Encoding](#) [mutation_change_weight](#) ([Genetic_Encoding](#) organism)
- [Genetic_Encoding](#) [put_randoms_weight](#) ([Genetic_Encoding](#) organism)
- [Genetic_Encoding](#) [crossover](#) ([Genetic_Encoding](#) orgm1, [Genetic_Encoding](#) orgm2)

Public Attributes

- int [lenght](#)
- int [total_niches](#)
- int [last_niche_id](#)
- int [last_innovation](#)
- int [last_node](#)
- int [last_row](#)
- double [fitness_champion](#)
- vector< [Genetic_Encoding](#) > [organisms](#)
- vector< [Genetic_Encoding](#) > [prev_organisms](#)
- vector< [Niche](#) > [current_niches](#)
- vector< [Niche](#) > [prev_niches](#)
- [Genetic_Encoding](#) * [best_organism](#)
- [Genetic_Encoding](#) [champion](#)
- vector< vector< int > > [historical_nodes](#)
- vector< vector< int > > [historical_innovation](#)
- vector< int > [historical_row](#)
- vector< int > [row_orderer_list](#)

6.6.1 Detailed Description

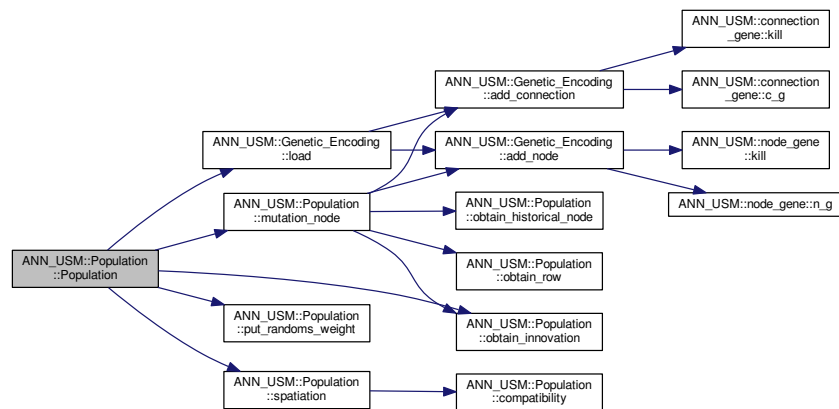
Definition at line 37 of file CPPN-NEAT.hpp.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 Population::Population (char path[])

Definition at line 13 of file CPPN-NEAT.cpp.

Here is the call graph for this function:



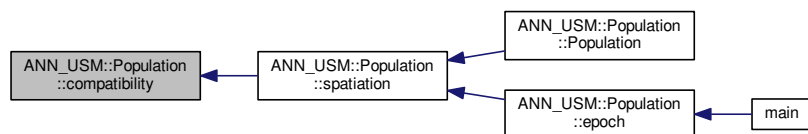
6.6.3 Member Function Documentation

6.6.3.1 `double Population::compatibility (Genetic_Encoding orgm1, Genetic_Encoding orgm2)`

Measure the distance between two organism.

Definition at line 341 of file CPPN-NEAT.cpp.

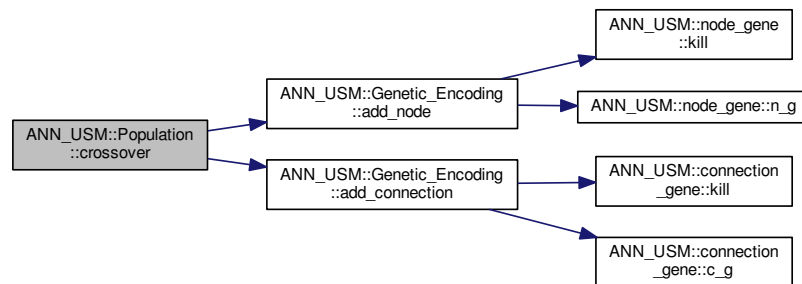
Here is the caller graph for this function:



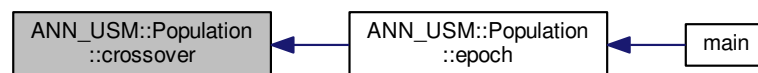
6.6.3.2 `Genetic_Encoding Population::crossover (Genetic_Encoding orgm1, Genetic_Encoding orgm2)`

Definition at line 370 of file CPPN-NEAT.cpp.

Here is the call graph for this function:



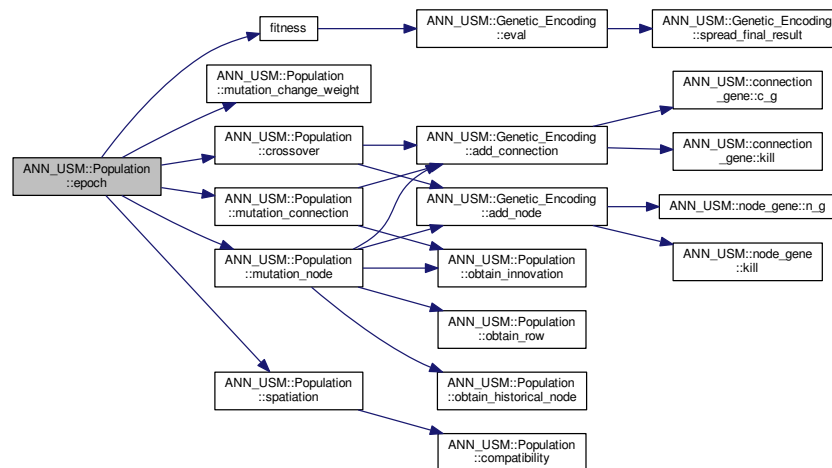
Here is the caller graph for this function:



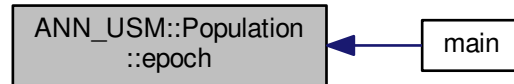
6.6.3.3 void Population::epoch ()

Definition at line 530 of file CPPN-NEAT.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.6.3.4 Genetic_Encoding Population::mutation_change_weight (Genetic_Encoding *organism*)

Definition at line 57 of file CPPN-NEAT.cpp.

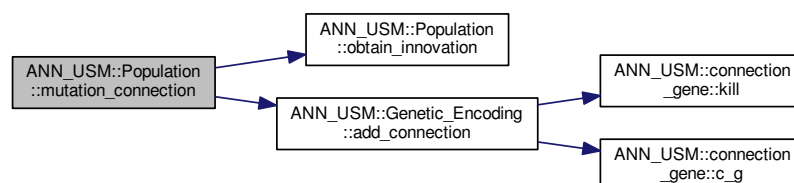
Here is the caller graph for this function:



6.6.3.5 Genetic_Encoding Population::mutation_connection (Genetic_Encoding *organism*)

Definition at line 241 of file CPPN-NEAT.cpp.

Here is the call graph for this function:



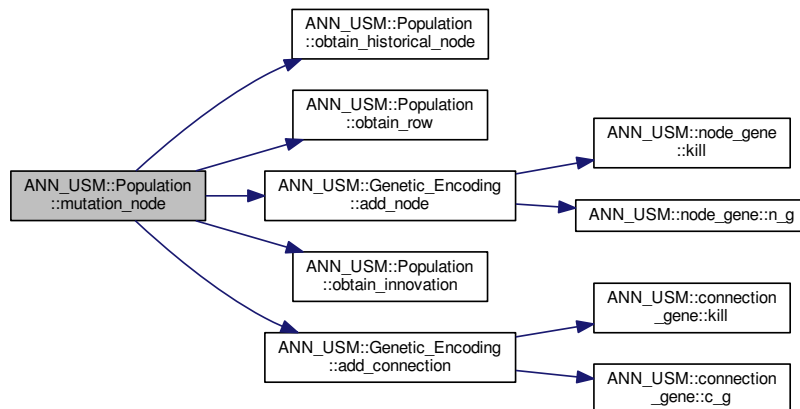
Here is the caller graph for this function:



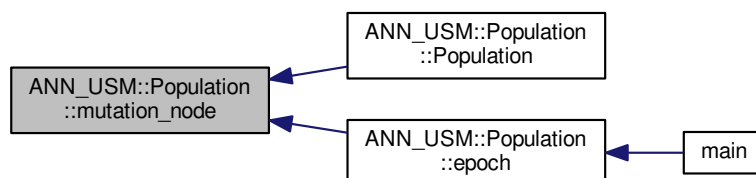
6.6.3.6 Genetic_Encoding Population::mutation_node (Genetic_Encoding organism)

Definition at line 68 of file CPPN-NEAT.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.6.3.7 int Population::obtain_historical_node (int initial_in, int initial_out)

Definition at line 193 of file CPPN-NEAT.cpp.

Here is the caller graph for this function:



6.6.3.8 int Population::obtain_innovation (int in, int out)

Obtain the (historical) innovation number between two nodes.

If both nodes were connected at any time in the past then they will have the very same innovation number they had earlier.

If they weren't connected at all then they will have a new innovation number, filling every vector in between with -1, i.e. not connected. If the inner node of the connection is greater than the vector size, fill with empty vectors until reach the desired node.

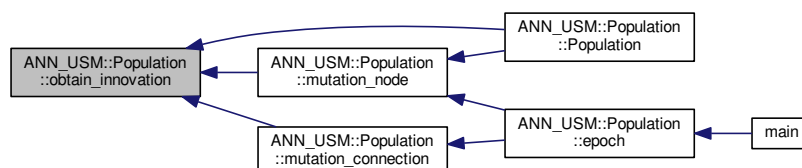
If the outgoing node of the connection is greater than the vector size at the inner position, fill with -1 until reach the desired node.

If the desired pair of nodes was not connected in the past, give the new innovation number to the connection and increase in one the innovation number.

If it was connected then skip the if statement and return the innovation number of the pair of nodes.

Definition at line 215 of file CPPN-NEAT.cpp.

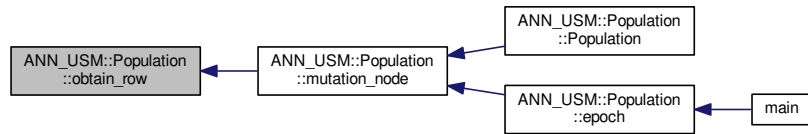
Here is the caller graph for this function:



6.6.3.9 int Population::obtain_row (int node, int node_initial_in, int node_initial_out)

Definition at line 126 of file CPPN-NEAT.cpp.

Here is the caller graph for this function:



6.6.3.10 Genetic_Encoding Population::put_randoms_weight (Genetic_Encoding organism)

Definition at line 49 of file CPPN-NEAT.cpp.

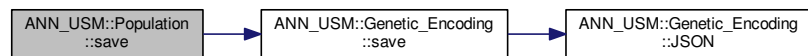
Here is the caller graph for this function:



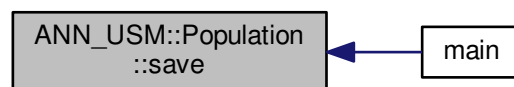
6.6.3.11 void Population::save (char path[])

Definition at line 334 of file CPPN-NEAT.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.6.3.12 void Population::save_all (char *path*[])

Definition at line 326 of file CPPN-NEAT.cpp.

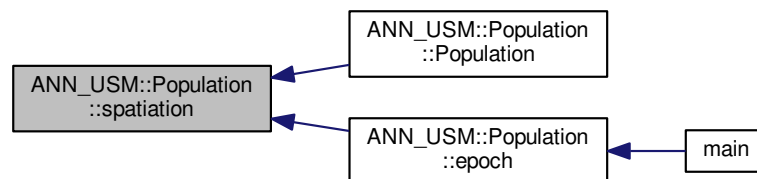
6.6.3.13 void Population::spatiation ()

Definition at line 469 of file CPPN-NEAT.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.6.4 Member Data Documentation

6.6.4.1 Genetic_Encoding* ANN_USM::Population::best_organism

Definition at line 75 of file CPPN-NEAT.hpp.

6.6.4.2 Genetic_Encoding ANN_USM::Population::champion

Definition at line 76 of file CPPN-NEAT.hpp.

6.6.4.3 vector<Niche> ANN_USM::Population::current_niches

Definition at line 72 of file CPPN-NEAT.hpp.

6.6.4.4 double ANN_USM::Population::fitness_champion

Definition at line 67 of file CPPN-NEAT.hpp.

6.6.4.5 `vector< vector<int> > ANN_USM::Population::historical_innovation`

Definition at line 79 of file CPPN-NEAT.hpp.

6.6.4.6 `vector< vector<int> > ANN_USM::Population::historical_nodes`

Definition at line 78 of file CPPN-NEAT.hpp.

6.6.4.7 `vector<int> ANN_USM::Population::historical_row`

Definition at line 81 of file CPPN-NEAT.hpp.

6.6.4.8 `int ANN_USM::Population::last_innovation`

Definition at line 63 of file CPPN-NEAT.hpp.

6.6.4.9 `int ANN_USM::Population::last_niche_id`

Definition at line 62 of file CPPN-NEAT.hpp.

6.6.4.10 `int ANN_USM::Population::last_node`

Definition at line 64 of file CPPN-NEAT.hpp.

6.6.4.11 `int ANN_USM::Population::last_row`

Definition at line 65 of file CPPN-NEAT.hpp.

6.6.4.12 `int ANN_USM::Population::lenght`

Definition at line 60 of file CPPN-NEAT.hpp.

6.6.4.13 `vector<Genetic_Encoding> ANN_USM::Population::organisms`

Definition at line 69 of file CPPN-NEAT.hpp.

6.6.4.14 `vector<Niche> ANN_USM::Population::prev_niches`

Definition at line 73 of file CPPN-NEAT.hpp.

6.6.4.15 `vector<Genetic_Encoding> ANN_USM::Population::prev_organisms`

Definition at line 70 of file CPPN-NEAT.hpp.

6.6.4.16 `vector<int> ANN_USM::Population::row_orderer_list`

Definition at line 82 of file CPPN-NEAT.hpp.

6.6.4.17 int ANN_USM::Population::total_niches

Definition at line 61 of file CPPN-NEAT.hpp.

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

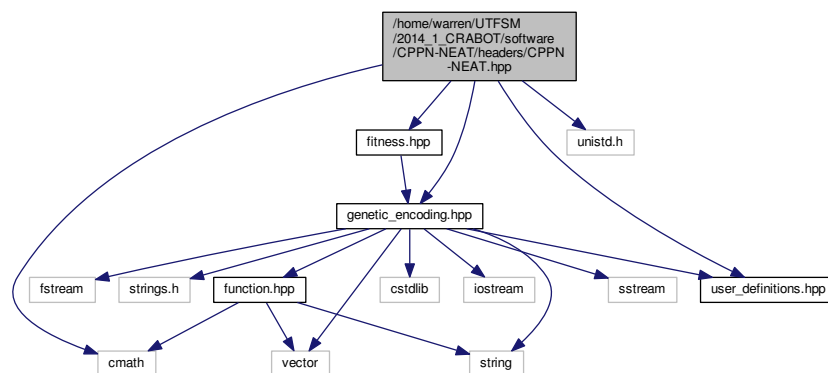
- /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/CPPN-NEAT.hpp
- /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/CPPN-NEAT.cpp

Chapter 7

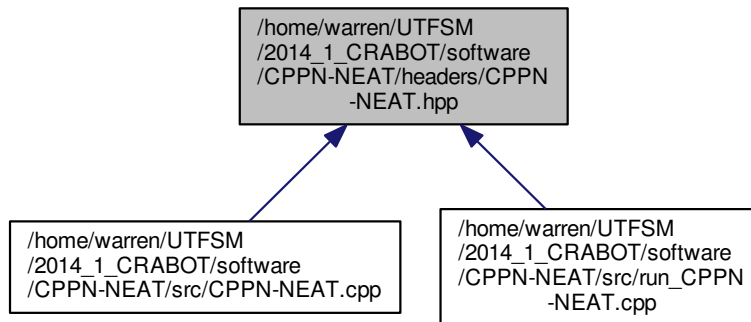
File Documentation

7.1 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/CPPN-NEAT.hpp File Reference

```
#include <cmath>
#include <unistd.h>
#include "fitness.hpp"
#include "genetic_encoding.hpp"
#include "user_definitions.hpp"
Include dependency graph for CPPN-NEAT.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [ANN_USM::Niche](#)
- class [ANN_USM::Population](#)

Namespaces

- [ANN_USM](#)

Functions

- ostream & [operator<<](#) (ostream &o, [ANN_USM::Population](#) &pop)

7.1.1 Function Documentation

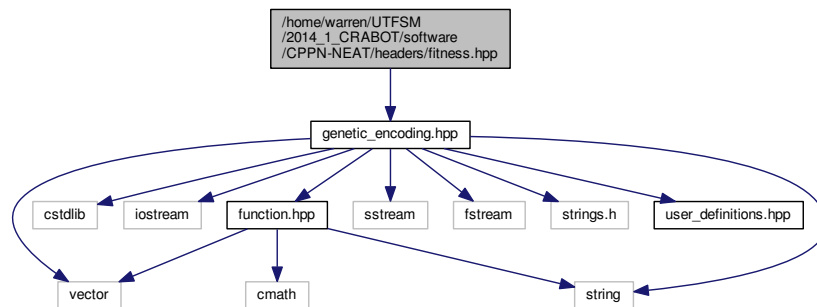
7.1.1.1 ostream& operator<< (ostream & o, ANN_USM::Population & pop)

Definition at line 315 of file CPPN-NEAT.cpp.

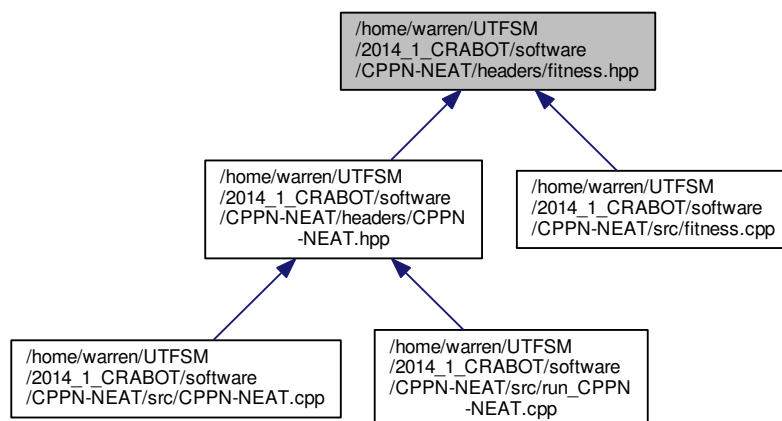
7.2 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/fitness.hpp File Reference

```
#include "genetic_encoding.hpp"
```

Include dependency graph for fitness.hpp:



This graph shows which files directly or indirectly include this file:



Functions

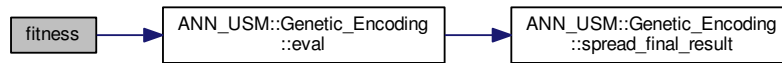
- double `fitness` (`Genetic_Encoding` organism)

7.2.1 Function Documentation

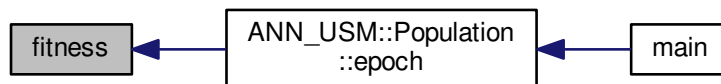
7.2.1.1 double `fitness` (`Genetic_Encoding` *organism*)

Definition at line 8 of file `fitness.cpp`.

Here is the call graph for this function:



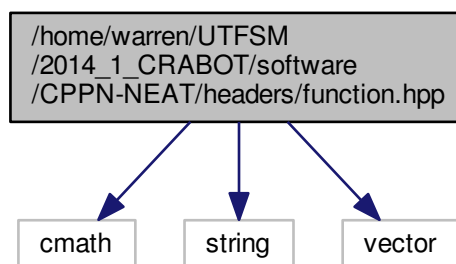
Here is the caller graph for this function:



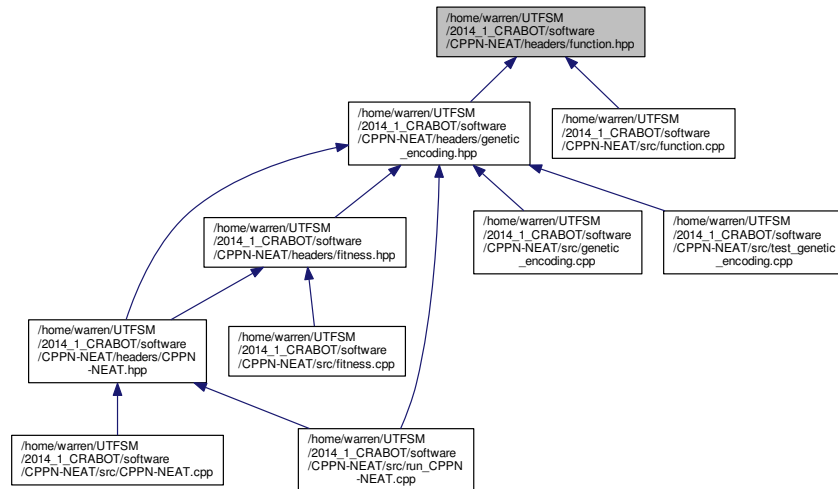
7.3 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/function.hpp File Reference

```
#include <cmath>
#include <string>
#include <vector>
```

Include dependency graph for function.hpp:



This graph shows which files directly or indirectly include this file:



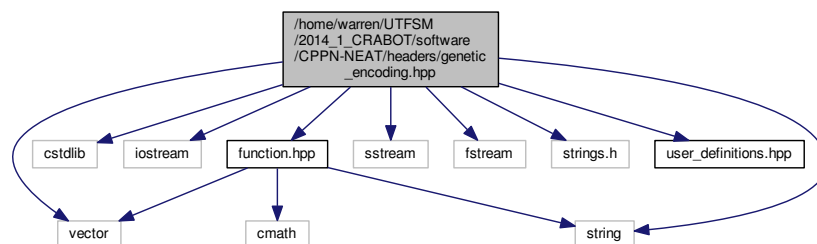
Classes

- class [Function](#)

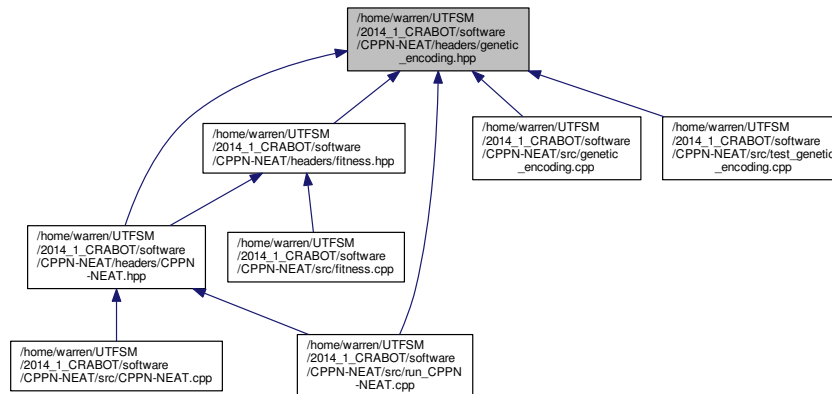
7.4 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/genetic_encoding.hpp File Reference

```
#include <vector>
#include <cstdlib>
#include <iostream>
#include <string>
#include <sstream>
#include <fstream>
#include <strings.h>
#include "user_definitions.hpp"
#include "function.hpp"
```

Include dependency graph for genetic_encoding.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [ANN_USM::connection_gene](#)
- class [ANN_USM::node_gene](#)
- class [ANN_USM::Genetic_Encoding](#)

Namespaces

- [ANN_USM](#)

Enumerations

- enum [ANN_USM::gene_type](#) { [ANN_USM::INPUT](#), [ANN_USM::HIDDEN](#), [ANN_USM::OUTPUT](#) }

Functions

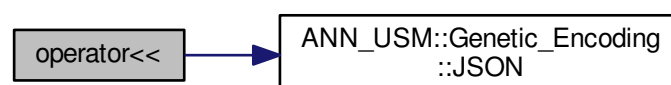
- ostream & [operator<<](#) (ostream &o, [ANN_USM::Genetic_Encoding](#) &encoding)

7.4.1 Function Documentation

7.4.1.1 ostream& operator<< (ostream & o, ANN_USM::Genetic_Encoding & encoding)

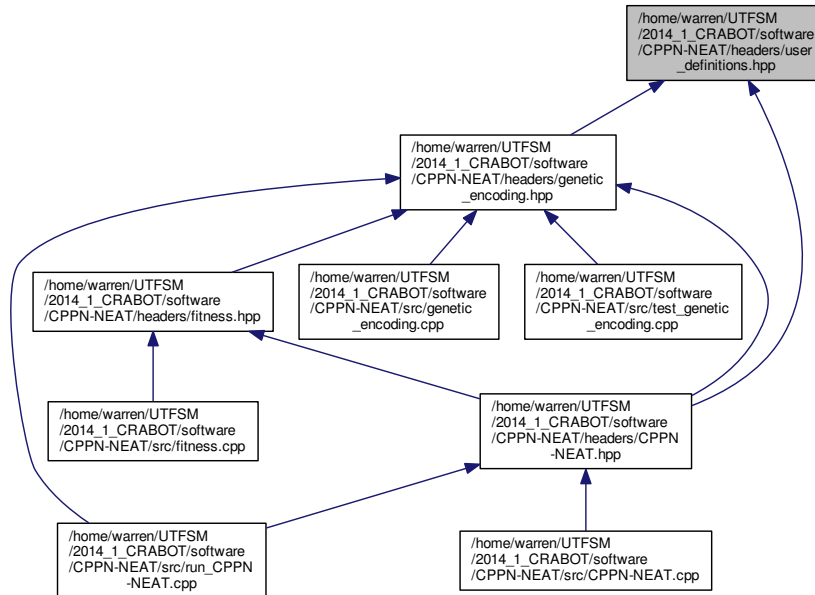
Definition at line 391 of file genetic_encoding.cpp.

Here is the call graph for this function:



7.5 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/headers/user_definitions.hpp File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define `POPULATION_MAX` 150
- #define `DISTANCE_CONST_1` 1.0
- #define `DISTANCE_CONST_2` 0.4
- #define `DISTANCE_CONST_3` 1.0
- #define `DISTANCE_CONST_4` 1.0
- #define `DISTANCE_THRESHOLD` 3.0
- #define `PERCENT_MUTATION_CONNECTION` 0.25
- #define `PERCENTAGE_OFFSPRING_WITHOUT_CROSSOVER` 25
- #define `PROBABILITY_INTERSPECIES_MATING` 0.01
- #define `SMALLER_POPULATIONS_PROBABILITY_ADDING_NEW_NODE` 0.07
- #define `SMALLER_POPULATIONS_PROBABILITY_ADDING_NEW_CONNECTION` 0.05
- #define `LARGER_POPULATIONS_PROBABILITY_ADDING_NEW_NODE` 0.01
- #define `LARGER_POPULATIONS_PROBABILITY_ADDING_NEW_CONNECTION` 0.3
- #define `PROBABILITY_CONNECTION_WEIGHT_MUTATING` 80
- #define `LARGE_POPULATION_DISCRIMINATOR` (POPULATION_MAX/10.0)
- #define `FUNCTION_NUM` 6
- #define `GENERATIONS` 50

7.5.1 Macro Definition Documentation

7.5.1.1 #define `DISTANCE_CONST_1` 1.0

Definition at line 6 of file user_definitions.hpp.

7.5.1.2 #define DISTANCE_CONST_2 0.4

Definition at line 7 of file user_definitions.hpp.

7.5.1.3 #define DISTANCE_CONST_3 1.0

Definition at line 8 of file user_definitions.hpp.

7.5.1.4 #define DISTANCE_CONST_4 1.0

Definition at line 9 of file user_definitions.hpp.

7.5.1.5 #define DISTANCE_THRESHOLD 3.0

Definition at line 10 of file user_definitions.hpp.

7.5.1.6 #define FUNCTION_NUM 6

Definition at line 22 of file user_definitions.hpp.

7.5.1.7 #define GENERATIONS 50

Definition at line 23 of file user_definitions.hpp.

7.5.1.8 #define LARGE_POPULATION_DISCRIMINATOR (POPULATION_MAX/10.0)

Definition at line 20 of file user_definitions.hpp.

7.5.1.9 #define LARGER_POPULATIONS_PROBABILITY_ADDING_NEW_CONNECTION 0.3

Definition at line 18 of file user_definitions.hpp.

7.5.1.10 #define LARGER_POPULATIONS_PROBABILITY_ADDING_NEW_NODE 0.01

Definition at line 17 of file user_definitions.hpp.

7.5.1.11 #define PERCENT_MUTATION_CONNECTION 0.25

Definition at line 12 of file user_definitions.hpp.

7.5.1.12 #define PERCENTAGE_OFFSPRING_WITHOUT_CROSSOVER 25

Definition at line 13 of file user_definitions.hpp.

7.5.1.13 #define POPULATION_MAX 150

Definition at line 5 of file user_definitions.hpp.

7.5.1.14 `#define PROBABILITY_CONNECTION_WEIGHT_MUTATING 80`

Definition at line 19 of file `user_definitions.hpp`.

7.5.1.15 `#define PROBABILITY_INTERSPACIES_MATING 0.01`

Definition at line 14 of file `user_definitions.hpp`.

7.5.1.16 `#define SMALLER_POPULATIONS_PROBABILITY_ADDING_NEW_CONNECTION 0.05`

Definition at line 16 of file `user_definitions.hpp`.

7.5.1.17 `#define SMALLER_POPULATIONS_PROBABILITY_ADDING_NEW_NODE 0.07`

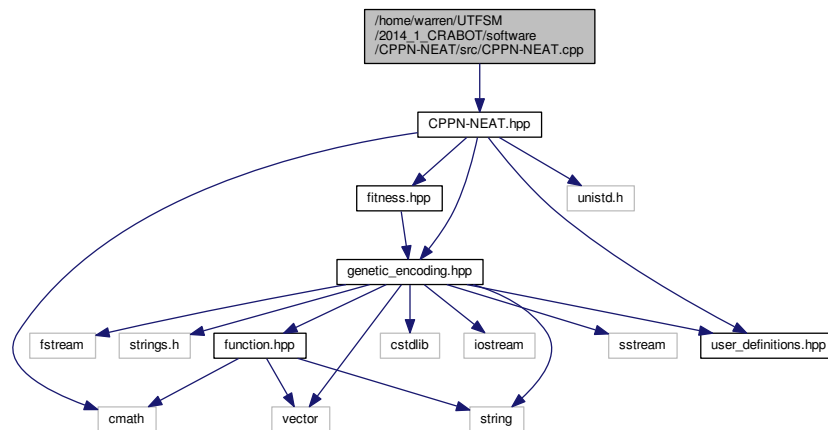
Definition at line 15 of file `user_definitions.hpp`.

7.6 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/README.md File Reference

7.7 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/CPPN-NEAT.cpp File Reference

```
#include "CPPN-NEAT.hpp"
```

Include dependency graph for `CPPN-NEAT.cpp`:



Macros

- `#define` [CPPN_NEAT_CPP](#)

Functions

- `ostream & operator<<` (`ostream &o`, [ANN_USM::Population](#) &pop)

7.7.1 Macro Definition Documentation

7.7.1.1 #define CPPN_NEAT_CPP

Definition at line 3 of file CPPN-NEAT.cpp.

7.7.2 Function Documentation

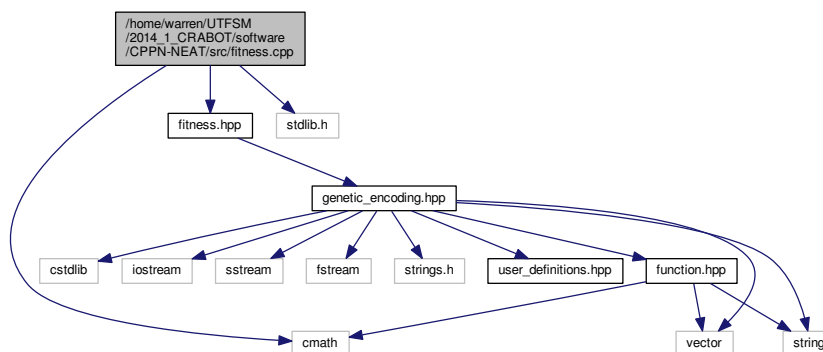
7.7.2.1 ostream& operator<< (ostream & o, ANN_USM::Population & pop)

Definition at line 315 of file CPPN-NEAT.cpp.

7.8 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/fitness.cpp File Reference

```
#include "fitness.hpp"
#include <stdlib.h>
#include <cmath>
```

Include dependency graph for fitness.cpp:



Macros

- #define `FITNESS_CPP`

Functions

- double `fitness` (`Genetic_Encoding` organism)

7.8.1 Macro Definition Documentation

7.8.1.1 #define FITNESS_CPP

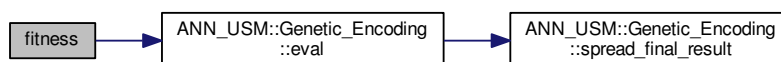
Definition at line 2 of file `fitness.cpp`.

7.8.2 Function Documentation

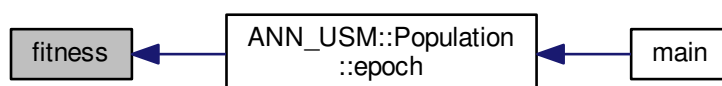
7.8.2.1 double fitness (Genetic_Encoding organism)

Definition at line 8 of file fitness.cpp.

Here is the call graph for this function:



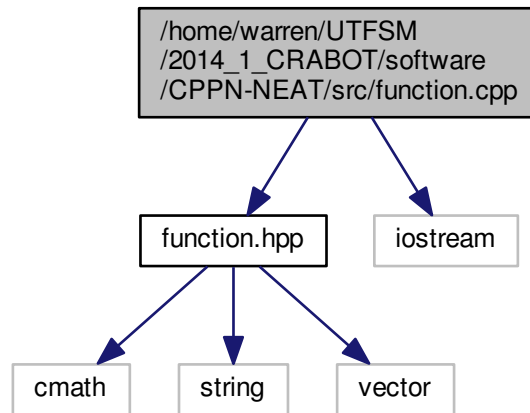
Here is the caller graph for this function:



7.9 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/function.cpp File Reference

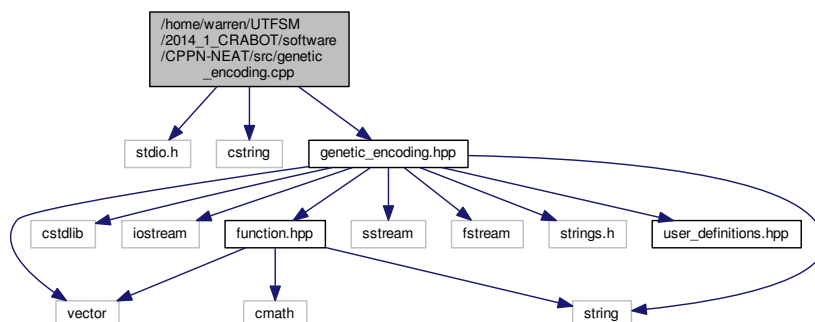
```
#include "function.hpp"
#include <iostream>
```

Include dependency graph for function.cpp:



7.10 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/genetic_encoding.cpp File Reference

```
#include <stdio.h>
#include <cstring>
#include "genetic_encoding.hpp"
Include dependency graph for genetic_encoding.cpp:
```



Macros

- `#define` [GENETIC_ENCODING_CPP](#)

Functions

- `ostream & operator<<` (`ostream &o`, [Genetic_Encoding](#) &encoding)

7.10.1 Macro Definition Documentation

7.10.1.1 #define GENETIC_ENCODING_CPP

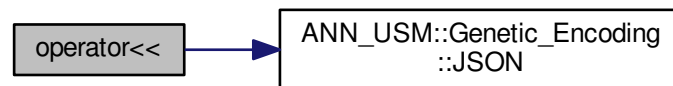
Definition at line 2 of file genetic_encoding.cpp.

7.10.2 Function Documentation

7.10.2.1 ostream& operator<< (ostream & o, Genetic_Encoding & encoding)

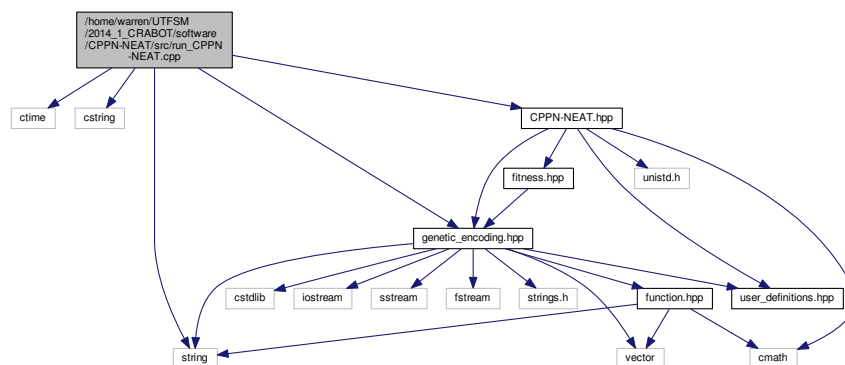
Definition at line 391 of file genetic_encoding.cpp.

Here is the call graph for this function:



7.11 /home/warren/UTFSM/2014_1_CRABOT/software/CPPN-NEAT/src/run_CPPN-NEAT.cpp File Reference

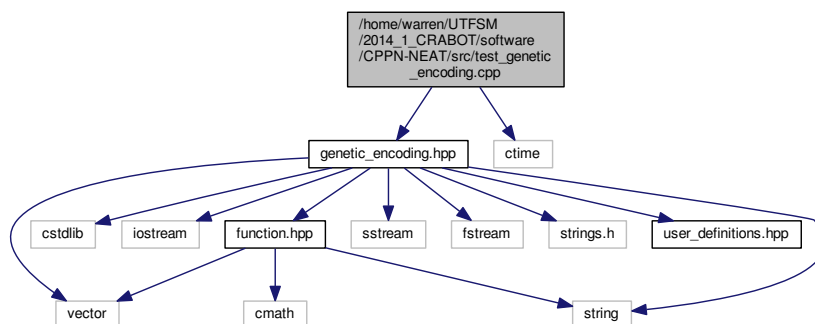
```
#include <ctime>
#include <cstring>
#include <string>
#include "CPPN-NEAT.hpp"
#include "genetic_encoding.hpp"
Include dependency graph for run_CPPN-NEAT.cpp:
```



Functions

- void [print_usage](#) (string proc)
- int [main](#) (int argc, char **argv)

Include dependency graph for test_genetic_encoding.cpp:



Macros

- `#define T_G_E_CPP`

Functions

- `int main()`

7.12.1 Macro Definition Documentation

7.12.1.1 `#define T_G_E_CPP`

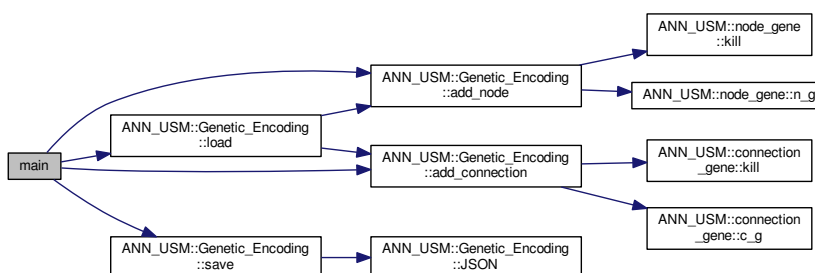
Definition at line 2 of file test_genetic_encoding.cpp.

7.12.2 Function Documentation

7.12.2.1 `int main()`

Definition at line 9 of file test_genetic_encoding.cpp.

Here is the call graph for this function:



Index

ANN_USM
 HIDDEN, 9
 INPUT, 9
 OUTPUT, 9
ANN_USM, 9
 gene_type, 9
ANN_USM::Genetic_Encoding, 14
 add_connection, 15
 add_node, 15, 16
 change_weight, 16
 eval, 16
 input_nodes, 19
 JSON, 17
 Lconnection_genes, 19
 Lnode_genes, 19
 load, 17
 niche, 19
 output_nodes, 19
 row_orderer_list, 19
 save, 18
 spread_final_result, 19
ANN_USM::Niche, 20
 amount_of_offspring, 20
 exist, 20
 niche_champion_position, 20
 organism_position, 20
 total_fitness, 20
 years, 20
ANN_USM::Population, 24
 best_organism, 32
 champion, 32
 compatibility, 26
 crossover, 26
 current_niches, 32
 epoch, 27
 fitness_champion, 32
 historical_innovation, 32
 historical_nodes, 33
 historical_row, 33
 last_innovation, 33
 last_niche_id, 33
 last_node, 33
 last_row, 33
 length, 33
 mutation_change_weight, 28
 mutation_connection, 28
 mutation_node, 29
 obtain_historical_node, 29
 obtain_innovation, 30
 obtain_row, 30
 organisms, 33
 Population, 25
 prev_niches, 33
 prev_organisms, 33
 put_randoms_weight, 31
 row_orderer_list, 33
 save, 31
 save_all, 31
 spatiation, 32
 total_niches, 33
ANN_USM::connection_gene, 11
 c_g, 11
 enable, 12
 exist, 12
 in, 12
 innovation, 12
 kill, 12
 out, 12
 weight, 13
ANN_USM::node_gene, 21
 accumulative_result, 23
 counter, 23
 decrease_incoming_connection, 22
 eval, 22
 exist, 23
 final_result, 23
 function, 23
 get_final_result, 22
 incoming_connections, 24
 increase_incoming_connection, 22
 is_ready, 22
 kill, 22
 n_g, 23
 node, 24
 node_gene, 21
 outgoing_connections, 24
 row, 24
 type, 24
accumulative_result
 ANN_USM::node_gene, 23
add_connection
 ANN_USM::Genetic_Encoding, 15
add_node
 ANN_USM::Genetic_Encoding, 15, 16
amount_of_offspring
 ANN_USM::Niche, 20
best_organism
 ANN_USM::Population, 32

- c_g
 - ANN_USM::connection_gene, 11
- CPPN-NEAT.cpp
 - operator<<, 44
- CPPN-NEAT.hpp
 - operator<<, 36
- CPPN_NEAT_CPP
 - CPPN-NEAT.cpp, 44
- champion
 - ANN_USM::Population, 32
- change_weight
 - ANN_USM::Genetic_Encoding, 16
- compatibility
 - ANN_USM::Population, 26
- counter
 - ANN_USM::node_gene, 23
- crossover
 - ANN_USM::Population, 26
- current_niches
 - ANN_USM::Population, 32
- DISTANCE_CONST_1
 - user_definitions.hpp, 41
- DISTANCE_CONST_2
 - user_definitions.hpp, 41
- DISTANCE_CONST_3
 - user_definitions.hpp, 42
- DISTANCE_CONST_4
 - user_definitions.hpp, 42
- DISTANCE_THRESHOLD
 - user_definitions.hpp, 42
- decrease_incoming_connection
 - ANN_USM::node_gene, 22
- enable
 - ANN_USM::connection_gene, 12
- epoch
 - ANN_USM::Population, 27
- eval
 - ANN_USM::Genetic_Encoding, 16
 - ANN_USM::node_gene, 22
 - Function, 13
- exist
 - ANN_USM::connection_gene, 12
 - ANN_USM::Niche, 20
 - ANN_USM::node_gene, 23
- FITNESS_CPP
 - fitness.cpp, 44
- FUNCTION_NUM
 - user_definitions.hpp, 42
- final_result
 - ANN_USM::node_gene, 23
- fitness
 - fitness.cpp, 45
 - fitness.hpp, 37
- fitness.cpp
 - FITNESS_CPP, 44
 - fitness, 45
- fitness.hpp
 - fitness, 37
- fitness_champion
 - ANN_USM::Population, 32
- Function, 13
 - eval, 13
 - Function, 13
 - get_name, 13, 14
- function
 - ANN_USM::node_gene, 23
- GENERATIONS
 - user_definitions.hpp, 42
- gene_type
 - ANN_USM, 9
- genetic_encoding.cpp
 - operator<<, 47
- genetic_encoding.hpp
 - operator<<, 40
- get_final_result
 - ANN_USM::node_gene, 22
- get_name
 - Function, 13, 14
- HIDDEN
 - ANN_USM, 9
- historical_innovation
 - ANN_USM::Population, 32
- historical_nodes
 - ANN_USM::Population, 33
- historical_row
 - ANN_USM::Population, 33
- INPUT
 - ANN_USM, 9
- in
 - ANN_USM::connection_gene, 12
- incoming_connections
 - ANN_USM::node_gene, 24
- increase_incoming_connection
 - ANN_USM::node_gene, 22
- innovation
 - ANN_USM::connection_gene, 12
- input_nodes
 - ANN_USM::Genetic_Encoding, 19
- is_ready
 - ANN_USM::node_gene, 22
- JSON
 - ANN_USM::Genetic_Encoding, 17
- kill
 - ANN_USM::connection_gene, 12
 - ANN_USM::node_gene, 22
- last_innovation
 - ANN_USM::Population, 33
- last_niche_id
 - ANN_USM::Population, 33
- last_node

- ANN_USM::Population, 33
- last_row
 - ANN_USM::Population, 33
- Lconnection_genes
 - ANN_USM::Genetic_Encoding, 19
- length
 - ANN_USM::Population, 33
- Lnode_genes
 - ANN_USM::Genetic_Encoding, 19
- load
 - ANN_USM::Genetic_Encoding, 17
- main
 - run_CPPN-NEAT.cpp, 48
 - test_genetic_encoding.cpp, 49
- mutation_change_weight
 - ANN_USM::Population, 28
- mutation_connection
 - ANN_USM::Population, 28
- mutation_node
 - ANN_USM::Population, 29
- n_g
 - ANN_USM::node_gene, 23
- niche
 - ANN_USM::Genetic_Encoding, 19
- niche_champion_position
 - ANN_USM::Niche, 20
- node
 - ANN_USM::node_gene, 24
- node_gene
 - ANN_USM::node_gene, 21
- OUTPUT
 - ANN_USM, 9
- obtain_historical_node
 - ANN_USM::Population, 29
- obtain_innovation
 - ANN_USM::Population, 30
- obtain_row
 - ANN_USM::Population, 30
- operator<<
 - CPPN-NEAT.cpp, 44
 - CPPN-NEAT.hpp, 36
 - genetic_encoding.cpp, 47
 - genetic_encoding.hpp, 40
- organism_position
 - ANN_USM::Niche, 20
- organisms
 - ANN_USM::Population, 33
- out
 - ANN_USM::connection_gene, 12
- outgoing_connections
 - ANN_USM::node_gene, 24
- output_nodes
 - ANN_USM::Genetic_Encoding, 19
- POPULATION_MAX
 - user_definitions.hpp, 42
- Population
 - ANN_USM::Population, 25
- prev_niches
 - ANN_USM::Population, 33
- prev_organisms
 - ANN_USM::Population, 33
- print_usage
 - run_CPPN-NEAT.cpp, 48
- put_randoms_weight
 - ANN_USM::Population, 31
- row
 - ANN_USM::node_gene, 24
- row_orderer_list
 - ANN_USM::Genetic_Encoding, 19
 - ANN_USM::Population, 33
- run_CPPN-NEAT.cpp
 - main, 48
 - print_usage, 48
- save
 - ANN_USM::Genetic_Encoding, 18
 - ANN_USM::Population, 31
- save_all
 - ANN_USM::Population, 31
- spatiation
 - ANN_USM::Population, 32
- spread_final_result
 - ANN_USM::Genetic_Encoding, 19
- T_G_E_CPP
 - test_genetic_encoding.cpp, 49
- test_genetic_encoding.cpp
 - main, 49
 - T_G_E_CPP, 49
- total_fitness
 - ANN_USM::Niche, 20
- total_niches
 - ANN_USM::Population, 33
- type
 - ANN_USM::node_gene, 24
- user_definitions.hpp
 - DISTANCE_CONST_1, 41
 - DISTANCE_CONST_2, 41
 - DISTANCE_CONST_3, 42
 - DISTANCE_CONST_4, 42
 - DISTANCE_THRESHOLD, 42
 - FUNCTION_NUM, 42
 - GENERATIONS, 42
 - POPULATION_MAX, 42
- weight
 - ANN_USM::connection_gene, 13
- years
 - ANN_USM::Niche, 20