

statgen

V.1.0.1

Generated by Doxygen 1.8.4

Sat Jul 27 2013 23:34:17



# Contents

<b>1</b>	<b>statgen</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	How to use . . . . .	1
<b>2</b>	<b>Install</b>	<b>3</b>
2.1	Requirements . . . . .	3
2.2	How to install . . . . .	3
2.2.1	Linux . . . . .	3
2.2.2	Windows . . . . .	3
<b>3</b>	<b>Changelog</b>	<b>5</b>
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>File Documentation</b>	<b>9</b>
5.1	about.dox File Reference . . . . .	9
5.2	src/add_main.c File Reference . . . . .	9
5.2.1	Function Documentation . . . . .	9
5.2.1.1	error_checking . . . . .	9
5.2.1.2	print_message . . . . .	10
5.2.1.3	printing_head . . . . .	10
5.2.1.4	set_defaults . . . . .	11
5.3	src/add_main.h File Reference . . . . .	11
5.3.1	Function Documentation . . . . .	12
5.3.1.1	error_checking . . . . .	12
5.3.1.2	print_message . . . . .	12
5.3.1.3	printing_head . . . . .	12
5.3.1.4	set_defaults . . . . .	13
5.4	src/coords.c File Reference . . . . .	13
5.4.1	Function Documentation . . . . .	14
5.4.1.1	reading_coords . . . . .	14
5.5	src/coords.h File Reference . . . . .	14

5.5.1	Function Documentation	14
5.5.1.1	reading_coords	14
5.6	src/graph.c File Reference	15
5.6.1	Function Documentation	16
5.6.1.1	check_cycle	16
5.6.1.2	check_cycle_size	17
5.6.1.3	check_tail	17
5.6.1.4	graph_analyze	17
5.7	src/graph.h File Reference	18
5.7.1	Function Documentation	18
5.7.1.1	check_cycle	18
5.7.1.2	check_cycle_size	18
5.7.1.3	check_tail	19
5.7.1.4	graph_analyze	19
5.8	src/int2char.c File Reference	19
5.8.1	Function Documentation	19
5.8.1.1	conv	19
5.9	src/int2char.h File Reference	20
5.9.1	Function Documentation	20
5.9.1.1	conv	20
5.10	src/main.c File Reference	20
5.10.1	Function Documentation	21
5.10.1.1	main	21
5.11	src/messages.c File Reference	21
5.11.1	Function Documentation	21
5.11.1.1	message	21
5.12	src/messages.h File Reference	21
5.12.1	Function Documentation	21
5.12.1.1	message	21
5.13	src/stat_print.c File Reference	22
5.13.1	Function Documentation	22
5.13.1.1	printing_agl	22
5.14	src/stat_print.h File Reference	23
5.14.1	Function Documentation	23
5.14.1.1	printing_agl	23
5.15	src/stat_select.c File Reference	24
5.15.1	Function Documentation	24
5.15.1.1	create_matrix	24
5.16	src/stat_select.h File Reference	25
5.16.1	Function Documentation	25

5.16.1.1	<a href="#">create_matrix</a>	25
5.17	<a href="#">src/stat_sort.c File Reference</a>	25
5.17.1	<a href="#">Function Documentation</a>	26
5.17.1.1	<a href="#">proc_matrix</a>	26
5.18	<a href="#">src/stat_sort.h File Reference</a>	26
5.18.1	<a href="#">Function Documentation</a>	26
5.18.1.1	<a href="#">proc_matrix</a>	26
5.19	<a href="#">src/summary_stat.c File Reference</a>	27
5.19.1	<a href="#">Function Documentation</a>	27
5.19.1.1	<a href="#">summary_statistic</a>	27
5.20	<a href="#">src/summary_stat.h File Reference</a>	28
5.20.1	<a href="#">Function Documentation</a>	28
5.20.1.1	<a href="#">summary_statistic</a>	28
<b>Index</b>		<b>29</b>



# Chapter 1

## statgen

### 1.1 Introduction

#### Developer:

- Evgeniy Alekseev aka arcanis

<esalexeev (at) gmail (dot) com>

#### License:

- GPL

### 1.2 How to use

#### Usage:

```
statgen -i INPUT -s FIRST, LAST -c X, Y, Z -a ... -r ... -o OUTPUT [ -g DEPTH ]  
[ -l LOGFILE ] [ -q ] [ -h ]
```

#### Parameters:

-i	- mask of input files
-s	- trajectory steps (integer)
-c	- cell size (float), Å
-a	- atom types (integer). Format: 'ATOM1' or 'ATOM1, ATOM2' or etc
-r	- criteria (float), Å. Format: '0-0:2.4, 0-1:3.0' means 0-0-interaction (<2.4 Å) and 0-1 (<3.0) are needed. This flag can be used multiple times
-o	- output file name
-g	- check graph isomorphism. DEPTH is max depth for check cycles (>= 3)
-l	- log enable
-q	- quiet enable
-h	- show this help and exit





# Chapter 2

## Install

### 2.1 Requirements

The application statgen requires the following external stuff:

- cmake  $\geq$  2.8
- gcc  $\geq$  4.8

### 2.2 How to install

#### 2.2.1 Linux

```
mkdir build && cd build
cmake -DCMAKE_INSTALL_PREFIX=/usr -DCMAKE_BUILD_TYPE=Release ../
make
make install
```

#### 2.2.2 Windows

```
create project file using 'cmake'
compile project
```

You may also download compiled executable file for Win\_x86.



## Chapter 3

# Changelog

V.1.0.1 (2013-07-27)

- initial release



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">add_main.c</a>	9
src/ <a href="#">add_main.h</a>	11
src/ <a href="#">coords.c</a>	13
src/ <a href="#">coords.h</a>	14
src/ <a href="#">graph.c</a>	15
src/ <a href="#">graph.h</a>	18
src/ <a href="#">int2char.c</a>	19
src/ <a href="#">int2char.h</a>	20
src/ <a href="#">main.c</a>	20
src/ <a href="#">messages.c</a>	21
src/ <a href="#">messages.h</a>	21
src/ <a href="#">stat_print.c</a>	22
src/ <a href="#">stat_print.h</a>	23
src/ <a href="#">stat_select.c</a>	24
src/ <a href="#">stat_select.h</a>	25
src/ <a href="#">stat_sort.c</a>	25
src/ <a href="#">stat_sort.h</a>	26
src/ <a href="#">summary_stat.c</a>	27
src/ <a href="#">summary_stat.h</a>	28



## Chapter 5

# File Documentation

### 5.1 about.dox File Reference

### 5.2 src/add\_main.c File Reference

```
#include <stdio.h>
#include "messages.h"
```

#### Functions

- int [error\\_checking](#) (const float \*cell, const int from, const char \*input, const int max\_depth, const int num\_of\_inter, const char \*output, const int to, const int type\_inter)  
*function that checks errors in input variables*
- int [printing\\_head](#) (const char \*output, const int log, const int quiet, const char \*input, const int from, const int to, const float \*cell, const int type\_inter, const int \*label\_atom, const int num\_of\_inter, const float \*crit, const int max\_depth)  
*function that prints header in output file*
- int [print\\_message](#) (const int quiet, FILE \*std\_output, const int log, FILE \*f\_log, const int mode, const char \*str)  
*function that prints message in log and stdout*
- int [set\\_defaults](#) (float \*cell, int \*from, char \*input, int \*log, int \*max\_depth, int \*num\_of\_inter, char \*output, int \*to, int \*type\_inter, int \*quiet)  
*function for set default values of variables*

#### 5.2.1 Function Documentation

5.2.1.1 int [error\\_checking](#) ( const float \* *cell*, const int *from*, const char \* *input*, const int *max\_depth*, const int *num\_of\_inter*, const char \* *output*, const int *to*, const int *type\_inter* )

function that checks errors in input variables

```
error\_checking (cell, from, input, max_depth, num_of_inter, output, to,
                 type_inter);
```

## Parameters

<i>cell</i>	massive of cell size
<i>from</i>	first trajectory step
<i>input</i>	mask of trajectory files
<i>max_depth</i>	max depth for check cycles in graph analyze
<i>num_of_inter</i>	number of different interactions
<i>output</i>	output file name
<i>to</i>	last trajectory step
<i>type_inter</i>	number of atoms for interactions

## Returns

11 - error in 'type\_inter'  
 12 - error in 'cell'  
 13 - error in 'to' or 'from'  
 14 - error in 'num\_of\_inter'  
 15 - error in 'input'  
 16 - error in 'output'  
 19 - error in 'max\_depth'  
 0 - exit without errors

5.2.1.2 `int print_message ( const int quiet, FILE * std_output, const int log, FILE * f_log, const int mode, const char * str )`

function that prints message in log and stdout

```
print_message (quiet, stdout, log, f_log, 0, str);
```

## Parameters

<i>quiet</i>	status of quiet-mode
<i>std_output</i>	stdout
<i>log</i>	status of log-mode
<i>f_log</i>	log file
<i>mode</i>	number of message in "messages.c"
<i>str</i>	additional text in message

## Returns

0 - exit without errors

5.2.1.3 `int printing_head ( const char * output, const int log, const int quiet, const char * input, const int from, const int to, const float * cell, const int type_inter, const int * label_atom, const int num_of_inter, const float * crit, const int max_depth )`

function that prints header in output file

```
printing_head (output, log, quiet, input, from, to, cell, type_inter, label_atom,
               num_of_inter, crit, max_depth);
```

## Parameters



<i>output</i>	output file nams
<i>log</i>	status of log-mode
<i>quiet</i>	status of quiet-mode
<i>input</i>	mask of trajectory files
<i>from</i>	first trajectory step
<i>to</i>	last trajectory step
<i>cell</i>	massive of cell size
<i>type_inter</i>	number of atoms for interactions
<i>label_atom</i>	massive of atom types for interactions
<i>num_of_inter</i>	number of different interactions
<i>crit</i>	massive of criteria
<i>max_depth</i>	maximum depth for check cycles in graph analyze

#### Returns

0 - exit without errors

**5.2.1.4** `int set_defaults ( float * cell, int * from, char * input, int * log, int * max_depth, int * num_of_inter, char * output, int * to, int * type_inter, int * quiet )`

function for set default values of variables

```
set_defaults (cell, &from, input, &log, &max_depth, &num_of_inter, output, &to,
              &type_inter, &quiet);
```

#### Parameters

<i>cell</i>	massive of cell size
<i>from</i>	first trajectory step
<i>input</i>	mask of trajectory files
<i>log</i>	status of log-mode
<i>max_depth</i>	maximum depth for check cycles in graph analyze
<i>num_of_inter</i>	number of different interactions
<i>output</i>	output file name
<i>to</i>	last trajectory step
<i>type_inter</i>	number of atoms for interactions
<i>quiet</i>	status of quiet-mode

#### Returns

0 - exit without errors

## 5.3 src/add\_main.h File Reference

### Functions

- int `error_checking` (const float \*, const int, const char \*, const int, const int, const char \*, const int, const int)  
*function that checks errors in input variables*
- int `printing_head` (const char \*, const int, const int, const char \*, const int, const int, const float \*, const int, const int \*, const int, const float \*, const int)  
*function that prints header in output file*
- int `print_message` (const int, FILE \*, const int, FILE \*, const int, const char \*)  
*function that prints message in log and stdout*
- int `set_defaults` (float \*, int \*, char \*, int \*, int \*, int \*, char \*, int \*, int \*, int \*)  
*function for set default values of variables*

### 5.3.1 Function Documentation

**5.3.1.1** `int error_checking ( const float * cell, const int from, const char * input, const int max_depth, const int num_of_inter, const char * output, const int to, const int type_inter )`

function that checks errors in input variables

```
error_checking (cell, from, input, max_depth, num_of_inter, output, to,
               type_inter);
```

#### Parameters

<i>cell</i>	massive of cell size
<i>from</i>	first trajectory step
<i>input</i>	mask of trajectory files
<i>max_depth</i>	max depth for check cycles in graph analyze
<i>num_of_inter</i>	number of different interactions
<i>output</i>	output file name
<i>to</i>	last trajectory step
<i>type_inter</i>	number of atoms for interactions

#### Returns

11 - error in 'type\_inter'  
 12 - error in 'cell'  
 13 - error in 'to' or 'from'  
 14 - error in 'num\_of\_inter'  
 15 - error in 'input'  
 16 - error in 'output'  
 19 - error in 'max\_depth'  
 0 - exit without errors

**5.3.1.2** `int print_message ( const int quiet, FILE * std_output, const int log, FILE * f_log, const int mode, const char * str )`

function that prints message in log and stdout

```
print_message (quiet, stdout, log, f_log, 0, str);
```

#### Parameters

<i>quiet</i>	status of quiet-mode
<i>std_output</i>	stdout
<i>log</i>	status of log-mode
<i>f_log</i>	log file
<i>mode</i>	number of message in "messages.c"
<i>str</i>	additional text in message

#### Returns

0 - exit without errors

**5.3.1.3** `int printing_head ( const char * output, const int log, const int quiet, const char * input, const int from, const int to, const float * cell, const int type_inter, const int * label_atom, const int num_of_inter, const float * crit, const int max_depth )`

function that prints header in output file

```
printing_head (output, log, quiet, input, from, to, cell, type_inter, label_atom,
              num_of_inter, crit, max_depth);
```

## Parameters

<i>output</i>	output file nams
<i>log</i>	status of log-mode
<i>quiet</i>	status of quiet-mode
<i>input</i>	mask of trajectory files
<i>from</i>	first trajectory step
<i>to</i>	last trajectory step
<i>cell</i>	massive of cell size
<i>type_inter</i>	number of atoms for interactions
<i>label_atom</i>	massive of atom types for interactions
<i>num_of_inter</i>	number of different interactions
<i>crit</i>	massive of criteria
<i>max_depth</i>	maximum depth for check cycles in graph analyze

## Returns

0 - exit without errors

5.3.1.4 `int set_defaults ( float * cell, int * from, char * input, int * log, int * max_depth, int * num_of_inter, char * output, int * to, int * type_inter, int * quiet )`

function for set default values of variables

```
set_defaults (cell, &from, input, &log, &max_depth, &num_of_inter, output, &to,
              &type_inter, &quiet);
```

## Parameters

<i>cell</i>	massive of cell size
<i>from</i>	first trajectory step
<i>input</i>	mask of trajectory files
<i>log</i>	status of log-mode
<i>max_depth</i>	maximum depth for check cycles in graph analyze
<i>num_of_inter</i>	number of different interactions
<i>output</i>	output file name
<i>to</i>	last trajectory step
<i>type_inter</i>	number of atoms for interactions
<i>quiet</i>	status of quiet-mode

## Returns

0 - exit without errors

## 5.4 src/coords.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
```

## Functions

- `int reading_coords (const int mode, const char *filename, const int type_inter, const int *label_atom, const float *cell, int *num_mol, int *num_atoms, int *true_label_mol, int *label_mol, int *type_atoms, float *coords, char *ch_type_atoms)`  
*function that reads coordinates from special file format*

### 5.4.1 Function Documentation

**5.4.1.1** `int reading_coords ( const int mode, const char * filename, const int type_inter, const int * label_atom, const float * cell, int * num_mol, int * num_atoms, int * true_label_mol, int * label_mol, int * type_atoms, float * coords, char * ch_type_atoms )`

function that reads coordinates from special file format

```
reading_coords (0, filename, type_inter, label_atom, cell, &num_mol,
                &num_atoms, true_label_mol, label_mol, type_atoms,
                coords, ch_type_atoms);
```

#### Parameters

<i>mode</i>	mode of reading; '1' is statgen, '2' is enviro or frad, '3' is agl
<i>filename</i>	input file name
<i>type_inter</i>	number of needed atoms (number of needed molecules)
<i>label_atom</i>	massive of needed atom types (massive of needed molecules)
<i>cell</i>	massive of cell size
<i>num_mol</i>	number of molecules
<i>num_atoms</i>	number of atoms
<i>true_label_mol</i>	massive of true numbers of molecule for atoms
<i>label_mol</i>	massive of numbers of molecule for atoms
<i>type_atoms</i>	massive of atom types
<i>coords</i>	massive of coordinates
<i>ch_type_atoms</i>	massive of char atom types

#### Returns

- 1 - file \$filename does not exist
- 2 - unknown mode
- 0 - exit without errors

#### Work blocks

reading file

translation

free memory

## 5.5 src/coords.h File Reference

### Functions

- `int reading_coords (const int, const char *, const int, const int *, const float *, int *, int *, int *, int *, int *, float *, char *)`

*function that reads coordinates from special file format*

### 5.5.1 Function Documentation

**5.5.1.1** `int reading_coords ( const int mode, const char * filename, const int type_inter, const int * label_atom, const float * cell, int * num_mol, int * num_atoms, int * true_label_mol, int * label_mol, int * type_atoms, float * coords, char * ch_type_atoms )`

function that reads coordinates from special file format

```
reading_coords (0, filename, type_inter, label_atom, cell, &num_mol,
                &num_atoms, true_label_mol, label_mol, type_atoms,
                coords, ch_type_atoms);
```

#### Parameters

<i>mode</i>	mode of reading; '1' is statgen, '2' is enviro or frad, '3' is agl
<i>filename</i>	input file name
<i>type_inter</i>	number of needed atoms (number of needed molecules)
<i>label_atom</i>	massive of needed atom types (massive of needed molecules)
<i>cell</i>	massive of cell size
<i>num_mol</i>	number of molecules
<i>num_atoms</i>	number of atoms
<i>true_label_mol</i>	massive of true numbers of molecule for atoms
<i>label_mol</i>	massive of numbers of molecule for atoms
<i>type_atoms</i>	massive of atom types
<i>coords</i>	massive of coordinates
<i>ch_type_atoms</i>	massive of char atom types

#### Returns

- 1 - file \$filename does not exist
- 2 - unknown mode
- 0 - exit without errors

#### Work blocks

reading file

translation

free memory

## 5.6 src/graph.c File Reference

```
#include <math.h>
#include <stdlib.h>
```

#### Functions

- int [check\\_cycle](#) (const int N, const int \*pn)  
*function that calculates number of cycles in graph*
- int [check\\_cycle\\_size](#) (const int N, const int \*matrix, const int depth, int \*n\_cycle)  
*function that returns number of cycles different size*
- int [check\\_tail](#) (const int \*pn)  
*function that calculates number of tails*
- int [graph\\_analyze](#) (const int N, const int \*matrix, const int max\_depth, int \*iso)  
*function that analyzes graph isomorphic class*

### 5.6.1 Function Documentation

#### 5.6.1.1 `int check_cycle ( const int N, const int * pn )`

function that calculates number of cycles in graph

```
cycle = check_cycle (N, pn);
```

## Parameters

<i>N</i>	number of vertexes
<i>pn</i>	massive of number of vertexes with weight equals to i

## Returns

number of cycles

5.6.1.2 `int check_cycle_size ( const int N, const int * matrix, const int depth, int * n_cycle )`

function that returns number of cycles different size

```
check_cycle_size (N, matrix, depth, n_cycle);
```

## Parameters

<i>N</i>	number of vertexes
<i>matrix</i>	connectivity matrix
<i>depth</i>	depth of search (maximum number of vertexes in cycle)
<i>n_cycle</i>	massive of number of cycle with number of vertexes equals to i

## Returns

1 - memory error  
0 - exit without errors

5.6.1.3 `int check_tail ( const int * pn )`

function that calculates number of tails

```
tails = check_tail (pn);
```

## Parameters

<i>pn</i>	massive of number of vertexes with weight equals to i
-----------	---

## Returns

number of tails

5.6.1.4 `int graph_analyze ( const int N, const int * matrix, const int max_depth, int * iso )`

function that analyzes graph isomorphich class

```
graph_analyze (N, matrix, max_depth, iso);
```

## Parameters

<i>N</i>	number of vertexes
----------	--------------------

<i>matrix</i>	connectivity matrix
<i>max_depth</i>	maximum depth of search for <code>check_cycle_size</code>
<i>iso</i>	isomorphism class

#### Returns

- 1 - memory error
- 0 - exit without errors

## 5.7 src/graph.h File Reference

### Functions

- `int graph_analyze (const int, const int *, const int, int *)`  
*function that analyzes graph isomorph class*
- `int check_cycle (const int, const int *)`  
*function that calculates number of cycles in graph*
- `int check_cycle_size (const int, const int *, const int, int *)`  
*function that returns number of cycles different size*
- `int check_tail (const int *)`  
*function that calculates number of tails*

### 5.7.1 Function Documentation

#### 5.7.1.1 `int check_cycle ( const int N, const int * pn )`

function that calculates number of cycles in graph

```
cycle = check_cycle (N, pn);
```

#### Parameters

<i>N</i>	number of vertexes
<i>pn</i>	massive of number of vertexes with weight equals to i

#### Returns

number of cycles

#### 5.7.1.2 `int check_cycle_size ( const int N, const int * matrix, const int depth, int * n_cycle )`

function that returns number of cycles different size

```
check_cycle_size (N, matrix, depth, n_cycle);
```

#### Parameters

<i>N</i>	number of vertexes
----------	--------------------



<i>matrix</i>	connectivity matrix
<i>depth</i>	depth of search (maximum number of vertexes in cycle)
<i>n_cycle</i>	massive of number of cycle with number of vertexes equals to i

**Returns**

- 1 - memory error
- 0 - exit without errors

**5.7.1.3 int check\_tail ( const int \* *pn* )**

function that calculates number of tails

```
tails = check_tail (pn);
```

**Parameters**

<i>pn</i>	massive of number of vertexes with weight equals to i
-----------	---

**Returns**

- number of tails

**5.7.1.4 int graph\_analyze ( const int *N*, const int \* *matrix*, const int *max\_depth*, int \* *iso* )**

function that analyzes graph isomorphich class

```
graph_analyze (N, matrix, max_depth, iso);
```

**Parameters**

<i>N</i>	number of vertexes
<i>matrix</i>	connectivity matrix
<i>max_depth</i>	maximum depth of search for check_cycle_size
<i>iso</i>	isomorphism class

**Returns**

- 1 - memory error
- 0 - exit without errors

**5.8 src/int2char.c File Reference****Functions**

- char `conv` (const int *fnumb*, const int *dig\_pos*)  
function that converts from integer to char

**5.8.1 Function Documentation****5.8.1.1 char conv ( const int *fnumb*, const int *dig\_pos* )**

function that converts from integer to char

```
char = conv (N, 0);
```

**Parameters**

<i>fnumb</i>	integer
<i>dig_pos</i>	position: ones=1, decimals=2, hundreds=3

**Returns**

char

**5.9 src/int2char.h File Reference****Functions**

- char `conv` (const int, const int)  
*function that converts from integer to char*

**5.9.1 Function Documentation****5.9.1.1 char conv ( const int *fnumb*, const int *dig\_pos* )**

function that converts from integer to char

```
char = conv (N, 0);
```

**Parameters**

<i>fnumb</i>	integer
<i>dig_pos</i>	position: ones=1, decimals=2, hundreds=3

**Returns**

char

**5.10 src/main.c File Reference**

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "add_main.h"
#include "coords.h"
#include "int2char.h"
#include "messages.h"
#include "stat_print.h"
#include "stat_select.h"
#include "stat_sort.h"
#include "summary_stat.h"
```

**Functions**

- int `main` (int argc, char \*argv[])

### 5.10.1 Function Documentation

5.10.1.1 `int main ( int argc, char * argv[] )`

## 5.11 src/messages.c File Reference

```
#include <stdio.h>
#include <time.h>
```

### Functions

- `int message (const int log, const int mode, const char *text, FILE *output)`  
*function that prints messages to output*

### 5.11.1 Function Documentation

5.11.1.1 `int message ( const int log, const int mode, const char * text, FILE * output )`

function that prints messages to output

```
message (log, mode, text, output);
```

#### Parameters

<i>log</i>	equal to 1 if print to logfile
<i>mode</i>	number of message
<i>text</i>	additional text
<i>output</i>	output file (may be stdout)

#### Returns

0 - exit without errors

## 5.12 src/messages.h File Reference

### Functions

- `int message (const int, const int, const char *, FILE *)`  
*function that prints messages to output*

### 5.12.1 Function Documentation

5.12.1.1 `int message ( const int log, const int mode, const char * text, FILE * output )`

function that prints messages to output

```
message (log, mode, text, output);
```

## Parameters

<i>log</i>	equal to 1 if print to logfile
<i>mode</i>	number of message
<i>text</i>	additional text
<i>output</i>	output file (may be stdout)

## Returns

0 - exit without errors

## 5.13 src/stat\_print.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "graph.h"
```

## Functions

- `int printing_agl (const char *input, const char *output, const int *connect, const int num_mol, const int *true_label_mol, const int *num_mol_agl, const int *agl, const int *stat, const int max_depth, int *type_agl)`  
function that prints agglomerates to output file

## 5.13.1 Function Documentation

5.13.1.1 `int printing_agl ( const char * input, const char * output, const int * connect, const int num_mol, const int * true_label_mol, const int * num_mol_agl, const int * agl, const int * stat, const int max_depth, int * type_agl )`

function that prints agglomerates to output file

```
printing_agl (input_file, output_file, number_of_molecules,
             true_label_molecules, num_of_molecules_in_agglomerates, agglomerates,
             statistic, max_depth, type_of_agglomerate);
```

## Parameters

<i>input</i>	input file name
<i>output</i>	output file name
<i>connect</i>	connectivity graph for all molecules
<i>num_mol</i>	number of molecules
<i>true_label_mol</i>	massive of true numbers of molecule for atoms
<i>num_mol_agl</i>	massive of number of molecules in agglomerates
<i>agl</i>	massive of agglomerates
<i>stat</i>	massive of statistic
<i>max_depth</i>	maximum depth for check cycles in graph analyze
<i>type_agl</i>	massive of number of agglomerate types

## Returns

1 - memory error  
0 - exit without errors

## Work blocks

```
print header
```

```

print body

    creating connectivity graph

graph topology analyze

    free memory

free memory

```

## 5.14 src/stat\_print.h File Reference

### Functions

- int [printing\\_agl](#) (const char \*, const char \*, const int \*, const int, const int \*, const int \*, const int \*, const int \*, const int \*, const int, int \*)  
*function that prints agglomerates to output file*

#### 5.14.1 Function Documentation

5.14.1.1 int [printing\\_agl](#) ( const char \* *input*, const char \* *output*, const int \* *connect*, const int *num\_mol*, const int \* *true\_label\_mol*, const int \* *num\_mol\_agl*, const int \* *agl*, const int \* *stat*, const int *max\_depth*, int \* *type\_agl* )

function that prints agglomerates to output file

```

printing\_agl (input_file, output_file, number_of_molecules,
               true_label_molecules, num_of_molecules_in_agglomerates, agglomerates,
               statistic, max_depth, type_of_agglomerate);

```

#### Parameters

<i>input</i>	input file name
<i>output</i>	output file name
<i>connect</i>	connectivity graph for all molecules
<i>num_mol</i>	number of molecules
<i>true_label_mol</i>	massive of true numbers of molecule for atoms
<i>num_mol_agl</i>	massive of number of molecules in agglomerates
<i>agl</i>	massive of agglomerates
<i>stat</i>	massive of statistic
<i>max_depth</i>	maximum depth for check cycles in graph analyze
<i>type_agl</i>	massive of number of agglomerate types

#### Returns

- 1 - memory error
- 0 - exit without errors

#### Work blocks

```

print header

print body

    creating connectivity graph

```

```
graph topology analyze
```

```
free memory
```

```
free memory
```

## 5.15 src/stat\_select.c File Reference

```
#include <math.h>
#include <stdlib.h>
```

### Functions

- int [create\\_matrix](#) (const int num\_mol, const int num\_atoms, const int \*label\_mol, const int \*type\_atoms, const float \*coords, const int num\_of\_inter, const float \*crit, int \*connect)

*function that creates connectivity matrix*

#### 5.15.1 Function Documentation

**5.15.1.1** int [create\\_matrix](#) ( const int *num\_mol*, const int *num\_atoms*, const int \* *label\_mol*, const int \* *type\_atoms*, const float \* *coords*, const int *num\_of\_inter*, const float \* *crit*, int \* *connect* )

function that creates connectivity matrix

```
create\_matrix (number_of_molecules, number_of_atoms, label_molecule,
               type_atoms, coords, number_of_interactions, criteria, connect_matrix);
```

#### Parameters

<i>num_mol</i>	number of molecules
<i>num_atoms</i>	number of atoms
<i>label_mol</i>	massive of numbers of molecule for atoms
<i>type_atoms</i>	massive of atom types
<i>coords</i>	massive of coordinates
<i>num_of_inter</i>	number of different interactions
<i>crit</i>	massive of criteria
<i>connect</i>	connectivity graph for all molecules

#### Returns

- 1 - memory error
- 0 - exit without errors

#### Work blocks

```
creating initial connectivity matrix
```

```
processing of initial connectivity matrix
```

```
free memory
```

## 5.16 src/stat\_select.h File Reference

### Functions

- int [create\\_matrix](#) (const int, const int, const int \*, const int \*, const float \*, const int, const float \*, int \*)  
*function that creates connectivity matrix*

#### 5.16.1 Function Documentation

- 5.16.1.1 int [create\\_matrix](#) ( const int *num\_mol*, const int *num\_atoms*, const int \* *label\_mol*, const int \* *type\_atoms*, const float \* *coords*, const int *num\_of\_inter*, const float \* *crit*, int \* *connect* )

function that creates connectivity matrix

```
create_matrix (number_of_molecules, number_of_atoms, label_molecule,
               type_atoms, coords, number_of_interactions, criteria, connect_matrix);
```

#### Parameters

<i>num_mol</i>	number of molecules
<i>num_atoms</i>	number of atoms
<i>label_mol</i>	massive of numbers of molecule for atoms
<i>type_atoms</i>	massive of atom types
<i>coords</i>	massive of coordinates
<i>num_of_inter</i>	number of different interactions
<i>crit</i>	massive of criteria
<i>connect</i>	connectivity graph for all molecules

#### Returns

- 1 - memory error
- 0 - exit without errors

#### Work blocks

```
creating initial connectivity matrix
```

```
processing of initial connectivity matrix
```

```
free memory
```

## 5.17 src/stat\_sort.c File Reference

```
#include <stdlib.h>
```

### Functions

- int [proc\\_matrix](#) (const int num\_mol, const int \*connect, int \*num\_mol\_agl, int \*agl, int \*stat, int \*stat\_all)  
*function that processes connectivity matrix*

### 5.17.1 Function Documentation

5.17.1.1 `int proc_matrix ( const int num_mol, const int * connect, int * num_mol_agl, int * agl, int * stat, int * stat_all )`

function that processes connectivity matrix

```
proc_matrix (number_of_molecules, connect_matrix,
num_of_molecules_in_agglomerates, agglomerates, statistic, summary_statistic);
```

#### Parameters

<i>num_mol</i>	number of molecules
<i>connect</i>	connectivity graph for all molecules
<i>num_mol_agl</i>	massive of number of molecules in agglomerates
<i>agl</i>	massive of agglomerates
<i>stat</i>	massive of statistic
<i>stat_all</i>	massive of summary statistic

#### Returns

- 1 - memory error
- 0 - exit without errors

#### Work blocks

```
select non-bonded molecules

unwrapping of connectivity matrix

filling statistic array

free memory
```

## 5.18 src/stat\_sort.h File Reference

### Functions

- `int proc_matrix (const int, const int *, int *, int *, int *, int *)`  
*function that processes connectivity matrix*

### 5.18.1 Function Documentation

5.18.1.1 `int proc_matrix ( const int num_mol, const int * connect, int * num_mol_agl, int * agl, int * stat, int * stat_all )`

function that processes connectivity matrix

```
proc_matrix (number_of_molecules, connect_matrix,
num_of_molecules_in_agglomerates, agglomerates, statistic, summary_statistic);
```

#### Parameters



<i>num_mol</i>	number of molecules
<i>connect</i>	connectivity graph for all molecules
<i>num_mol_agl</i>	massive of number of molecules in agglomerates
<i>agl</i>	massive of agglomerates
<i>stat</i>	massive of statistic
<i>stat_all</i>	massive of summary statistic

#### Returns

- 1 - memory error
- 0 - exit without errors

#### Work blocks

```

select non-bonded molecules

unwrapping of connectivity matrix

filling statistic array

free memory

```

## 5.19 src/summary\_stat.c File Reference

```
#include <stdio.h>
```

#### Functions

- int [summary\\_statistic](#) (const char \*filename, const int step, const int num\_mol, const int max\_depth, const int \*type\_agl, const int \*stat\_all)  
*function that prints summary statistic*

#### 5.19.1 Function Documentation

5.19.1.1 int [summary\\_statistic](#) ( const char \* *filename*, const int *step*, const int *num\_mol*, const int *max\_depth*, const int \* *type\_agl*, const int \* *stat\_all* )

function that prints summary statistic

```
summary_statistic (filename, number_of_step, number_of_molecules, max_depth,
                  type_of_agglomerate, summary_statistic);
```

#### Parameters

<i>filename</i>	output file name
<i>step</i>	number of steps
<i>num_mol</i>	number of molecules
<i>max_depth</i>	maximum depth for check cycles in graph analyze

<i>type_agl</i>	massive of number of agglomerate types
<i>stat_all</i>	massive of summary statistic

#### Returns

0 - exit without errors

## 5.20 src/summary\_stat.h File Reference

### Functions

- int [summary\\_statistic](#) (const char \*, const int, const int, const int, const int \*, const int \*)  
*function that prints summary statistic*

#### 5.20.1 Function Documentation

5.20.1.1 int [summary\\_statistic](#) ( const char \* *filename*, const int *step*, const int *num\_mol*, const int *max\_depth*, const int \* *type\_agl*, const int \* *stat\_all* )

function that prints summary statistic

```
summary\_statistic (filename, number_of_step, number_of_molecules, max_depth,  
                    type_of_agglomerate, summary\_statistic);
```

#### Parameters

<i>filename</i>	output file name
<i>step</i>	number of steps
<i>num_mol</i>	number of molecules
<i>max_depth</i>	maximum depth for check cycles in graph analyze
<i>type_agl</i>	massive of number of agglomerate types
<i>stat_all</i>	massive of summary statistic

#### Returns

0 - exit without errors

# Index

- about.dox, [9](#)
- add\_main.c
  - error\_checking, [9](#)
  - print\_message, [10](#)
  - printing\_head, [10](#)
  - set\_defaults, [11](#)
- add\_main.h
  - error\_checking, [12](#)
  - print\_message, [12](#)
  - printing\_head, [12](#)
  - set\_defaults, [13](#)
- check\_cycle
  - graph.c, [16](#)
  - graph.h, [18](#)
- check\_cycle\_size
  - graph.c, [17](#)
  - graph.h, [18](#)
- check\_tail
  - graph.c, [17](#)
  - graph.h, [19](#)
- conv
  - int2char.c, [19](#)
  - int2char.h, [20](#)
- coords.c
  - reading\_coords, [14](#)
- coords.h
  - reading\_coords, [14](#)
- create\_matrix
  - stat\_select.c, [24](#)
  - stat\_select.h, [25](#)
- error\_checking
  - add\_main.c, [9](#)
  - add\_main.h, [12](#)
- graph.c
  - check\_cycle, [16](#)
  - check\_cycle\_size, [17](#)
  - check\_tail, [17](#)
  - graph\_analyze, [17](#)
- graph.h
  - check\_cycle, [18](#)
  - check\_cycle\_size, [18](#)
  - check\_tail, [19](#)
  - graph\_analyze, [19](#)
- graph\_analyze
  - graph.c, [17](#)
  - graph.h, [19](#)
- int2char.c
  - conv, [19](#)
- int2char.h
  - conv, [20](#)
- main
  - main.c, [21](#)
- main.c
  - main, [21](#)
- message
  - messages.c, [21](#)
  - messages.h, [21](#)
- messages.c
  - message, [21](#)
- messages.h
  - message, [21](#)
- print\_message
  - add\_main.c, [10](#)
  - add\_main.h, [12](#)
- printing\_agl
  - stat\_print.c, [22](#)
  - stat\_print.h, [23](#)
- printing\_head
  - add\_main.c, [10](#)
  - add\_main.h, [12](#)
- proc\_matrix
  - stat\_sort.c, [26](#)
  - stat\_sort.h, [26](#)
- reading\_coords
  - coords.c, [14](#)
  - coords.h, [14](#)
- set\_defaults
  - add\_main.c, [11](#)
  - add\_main.h, [13](#)
- src/add\_main.c, [9](#)
- src/add\_main.h, [11](#)
- src/coords.c, [13](#)
- src/coords.h, [14](#)
- src/graph.c, [15](#)
- src/graph.h, [18](#)
- src/int2char.c, [19](#)
- src/int2char.h, [20](#)
- src/main.c, [20](#)
- src/messages.c, [21](#)
- src/messages.h, [21](#)
- src/stat\_print.c, [22](#)
- src/stat\_print.h, [23](#)
- src/stat\_select.c, [24](#)

- src/stat\_select.h, [25](#)
- src/stat\_sort.c, [25](#)
- src/stat\_sort.h, [26](#)
- src/summary\_stat.c, [27](#)
- src/summary\_stat.h, [28](#)
- stat\_print.c
  - printing\_agl, [22](#)
- stat\_print.h
  - printing\_agl, [23](#)
- stat\_select.c
  - create\_matrix, [24](#)
- stat\_select.h
  - create\_matrix, [25](#)
- stat\_sort.c
  - proc\_matrix, [26](#)
- stat\_sort.h
  - proc\_matrix, [26](#)
- summary\_stat.c
  - summary\_statistic, [27](#)
- summary\_stat.h
  - summary\_statistic, [28](#)
- summary\_statistic
  - summary\_stat.c, [27](#)
  - summary\_stat.h, [28](#)