

mm_statgen

V.1.1.0

Generated by Doxygen 1.8.5

Mon Sep 2 2013 17:07:29

Contents

1	mm_statgen	1
1.1	Introduction	1
1.2	How to use	1
2	Install	3
2.1	Requirements	3
2.2	How to install	3
2.2.1	Linux	3
2.2.2	Windows	3
3	Changelog	5
4	File Index	7
4.1	File List	7
5	File Documentation	9
5.1	src/add_main.c File Reference	9
5.1.1	Function Documentation	9
5.1.1.1	error_checking	9
5.1.1.2	print_message	10
5.1.1.3	printing_head	10
5.1.1.4	set_defaults	11
5.2	src/coords.c File Reference	11
5.2.1	Function Documentation	12
5.2.1.1	reading_coords	12
5.3	src/graph.c File Reference	12
5.3.1	Function Documentation	13
5.3.1.1	check_cycle	13
5.3.1.2	check_cycle_size	13
5.3.1.3	check_tail	13
5.3.1.4	graph_analyze	13
5.4	src/main.c File Reference	14
5.4.1	Function Documentation	14

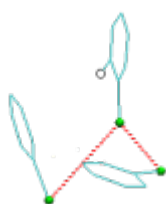
5.4.1.1	main	14
5.5	src/messages.c File Reference	14
5.5.1	Function Documentation	15
5.5.1.1	message	15
5.6	src/stat_print.c File Reference	15
5.6.1	Function Documentation	15
5.6.1.1	printing_agl	15
5.7	src/stat_select.c File Reference	16
5.7.1	Function Documentation	16
5.7.1.1	create_matrix	16
5.8	src/stat_sort.c File Reference	17
5.8.1	Function Documentation	17
5.8.1.1	proc_matrix	17
5.9	src/summary_stat.c File Reference	18
5.9.1	Function Documentation	18
5.9.1.1	summary_statistic	18

Index

19

Chapter 1

mm_statgen



1.1 Introduction

About this program:

- Program that analyzes molecular dynamic trajectories using topological analysis

Developer:

- Evgeniy Alekseev aka arcanis

<esalexeev (at) gmail (dot) com>

License:

- GPL

1.2 How to use

Usage:

```
mm_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 mm_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.1.0.3 (2013-08-30)

- Bug fixes

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/ add_main.c	9
src/ coords.c	11
src/ graph.c	12
src/ main.c	14
src/ messages.c	14
src/ stat_print.c	15
src/ stat_select.c	16
src/ stat_sort.c	17
src/ summary_stat.c	18

Chapter 5

File Documentation

5.1 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 that sets default values of variables

5.1.1 Function Documentation

5.1.1.1 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

5.1.1.2 `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

5.1.1.3 `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

5.1.1.4 `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 that sets 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

5.2 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

5.2.1 Function Documentation

5.2.1.1 `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

5.3 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

5.3.1 Function Documentation

5.3.1.1 `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

5.3.1.2 `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

5.3.1.3 `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

5.3.1.4 `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

5.4 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 "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[])

5.4.1 Function Documentation**5.4.1.1 int main (int argc, char * argv[])****Returns**

- 1 - error in error_checking
- 2 - input file does not exist
- 3 - memory error
- 4 - unknown flag
- 0 - exit without errors

5.5 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

5.5.1 Function Documentation

5.5.1.1 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

1 - unknown mode
0 - exit without errors

5.6 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

5.6.1 Function Documentation

5.6.1.1 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

```

5.7 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

5.7.1 Function Documentation

5.7.1.1 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

5.8 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

5.8.1 Function Documentation

5.8.1.1 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

```

5.9 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

5.9.1 Function Documentation

5.9.1.1 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

- add_main.c
 - error_checking, [9](#)
 - print_message, [10](#)
 - printing_head, [10](#)
 - set_defaults, [11](#)
- check_cycle
 - graph.c, [13](#)
- check_cycle_size
 - graph.c, [13](#)
- check_tail
 - graph.c, [13](#)
- coords.c
 - reading_coords, [12](#)
- create_matrix
 - stat_select.c, [16](#)
- error_checking
 - add_main.c, [9](#)
- graph.c
 - check_cycle, [13](#)
 - check_cycle_size, [13](#)
 - check_tail, [13](#)
 - graph_analyze, [13](#)
- graph_analyze
 - graph.c, [13](#)
- main
 - main.c, [14](#)
- main.c
 - main, [14](#)
- message
 - messages.c, [15](#)
- messages.c
 - message, [15](#)
- print_message
 - add_main.c, [10](#)
- printing_agl
 - stat_print.c, [15](#)
- printing_head
 - add_main.c, [10](#)
- proc_matrix
 - stat_sort.c, [17](#)
- reading_coords
 - coords.c, [12](#)
- set_defaults
 - add_main.c, [11](#)
- src/add_main.c, [9](#)
- src/coords.c, [11](#)
- src/graph.c, [12](#)
- src/main.c, [14](#)
- src/messages.c, [14](#)
- src/stat_print.c, [15](#)
- src/stat_select.c, [16](#)
- src/stat_sort.c, [17](#)
- src/summary_stat.c, [18](#)
- stat_print.c
 - printing_agl, [15](#)
- stat_select.c
 - create_matrix, [16](#)
- stat_sort.c
 - proc_matrix, [17](#)
- summary_stat.c
 - summary_statistic, [18](#)
- summary_statistic
 - summary_stat.c, [18](#)