mm_statgen

V.1.1.1

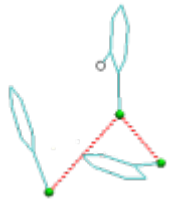
Generated by Doxygen 1.8.5

Tue Sep 3 2013 17:37:09

Contents

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 ]

Parametrs:
   -i          - mask of input files
   -s          - trajectory steps (integer)
   -c          - cell size (float), A
   -a          - atom types (integer). Format: 'ATOM1' or 'ATOM1,ATOM2' or etc
   -r          - criteria (float), A. Format: '0-0:2.4,0-1:3.0' means 0-0-interaction
                 (<2.4 A) 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 >= 2.8
- gcc >= 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.1.1.1 (2013-09-03)

- optimization

V.1.1.0 (2013-09-02)

- added help window
- added help docs
- small bug fixes

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:

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

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

quiet	status of quiet-mode
std_output	stdout
log	status of log-mode
f_log	log file
mode	number of message in "messages.c"
str	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

output	output file nams
log	status of log-mode
quiet	status of quiet-mode
input	mask of trajectory files
from	first trajectory step
to	last trajectory step
cell	massive of cell size
type_inter	number of atoms for interactions
label_atom	massive of atom types for interactions
num_of_inter	number of different interactions
crit	massive of criteria
max_depth	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

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

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

N	number of vertexes
pn	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

N	number of vertexes
matrix	connectivity matrix
depth	depth of search (maximum number of vertexes in cycle)
n_cycle	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

pn	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 isomorhic class

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

Parameters

N	number of vertexes
matrix	connectivity matrix
max_depth	maximum depth of search for check_cycle_size
iso	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

log	equal to 1 if print to logfile
mode	number of message
text	additional text
output	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

input	input file name
output	output file name
connect	connectivity graph for all molecules

num_mol	number of molecules
true_label_mol	massive of true numbers of molecule for atoms
num_mol_agl	massive of number of molecules in agglomerates
agl	massive of agglomerates
stat	massive of statistic
max_depth	maximum depth for check cycles in graph analyze
type_agl	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

num_mol	number of molecules
num_atoms	number of atoms
label_mol	massive of numbers of molecule for atoms
type_atoms	massive of atom types
coords	massive of coordinates
num_of_inter	number of different interactions
crit	massive of criteria
connect	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

num_mol	number of molecules
connect	connectivity graph for all molecules
num_mol_agl	massive of number of molecules in agglomerates
agl	massive of agglomerates

stat	massive of statistic
stat_all	massive of summary statistic

Returns

1 - memory error
0 - exit without errors

Work blocks

```
select non-bonded molecules

unwraping 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

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

Returns

0 - exit without errors

Index