mm_radf

V.1.0.2

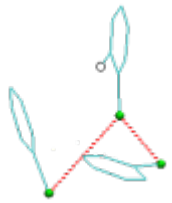
Generated by Doxygen 1.8.5

Fri Aug 30 2013 03:01:56

Contents

Chapter 1

mm_radf



1.1 Introduction

About this program:

- Program that calculates radial distribution function (RDF) or radial-angles distribution function

Developer:

- Evgeniy Alekseev aka arcanis

  <esalexeev (at) gmail (dot) com>

License:

- GPL

1.2 How to use

Usage:

```
mm_radf -i INPUT -s FIRST,LAST -c X,Y,Z -a ... -o OUTPUT [ -r MIN,MAX ] [ -rs R_STEP ]
                  [ -a MIN,MAX ] [ -as ANG_STEP ] [ -m ] [ -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-ATOM2' or 'A1,A2,A3-B1,B2,B3'
                 (will enable RDF calculation for center mass automaticaly)
   -o          - output file name
   -r          - minimal and maximal radii for analyze (float), A. Default is '2.0,15.0'
   -rs         - radius step for analyze (float), A. Default is '0.2'
   -a          - minimal and maximal angles for analyze (float), deg. Default is '0.0,90.0'
```

```
-as         - angle step for analyze (float), deg. This option will enable RADF
              calculation automaticaly
-m          - matrix output enable
-l          - log enable
-q          - quiet enable
-h          - show this help and exit
```

Chapter 2

Install

2.1 Requirements

The application mm_radf 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.1.0.2 (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 num_needed_at, const int *needed_at, const char *output, const int to)

  function that checks errors in input variables
- 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 printing_head (const char *output, const int log, const int quiet, const int matrix, const char *input, const int from, const int to, const float *cell, const int mode, const double r_min, const double r_max, const double r_step, const double ang_min, const double ang_max, const double ang_step, const int *needed_at)

  function that prints header in output file
- int set_defaults (float *ang_max, float *ang_min, float *ang_step, float *cell, int *from, char *input, int *log, int *matrix, float *r_max, float *r_min, float *r_step, char *output, int *to, 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 *num_needed_at*, const int * *needed_at*, const char * *output*, const int *to*)

function that checks errors in input variables

```
* error_checking (cell, from, input, num_needed_at, needed_at, output, to);
*
```

Parameters

cell	massive of cell size

from	first trajectory step
input	input file name
num_needed_at	number of needed atom types
needed_at	massive of number of needed atom types
output	output file name
to	last trajectory step

Returns

11 - error in 'cell'
12 - error in 'input'
13 - error in 'output'
14 - error in 'from' or 'to'
15 - error in 'num_needed_at'
16 - error in 'needed_at'
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 int *matrix*, const char * *input*, const int *from*, const int *to*, const float * *cell*, const int *mode*, const double *r_min*, const double *r_max*, const double *r_step*, const double *ang_min*, const double *ang_max*, const double *ang_step*, const int * *needed_at*)

function that prints header in output file

```
* printing_head (output, log, quiet, matrix, input, from, to, cell, mode, r_min,
*                 r_max, r_step, ang_min, ang_max, ang_step, needed_at);
*
```

Parameters

output	output file nams
log	status of log-mode

quiet	status of quiet-mode
matrix	status of matrix-mode
input	mask of trajectory files
from	first trajectory step
to	last trajectory step
cell	massive of cell size
mode	0 - if RDF, 1 - if RDF for center mass, 2 - if RADF
r_max	maximal radius
r_min	minimal radius
r_step	radius step
ang_max	maximal angle for RADF
ang_min	minimal angle for RADF
ang_step	anlge step for RADF
needed_at	massive of number of needed atom types

Returns

0 - exit without errors

5.1.1.4 int set_defaults (float * *ang_max*, float * *ang_min*, float * *ang_step*, float * *cell*, int * *from*, char * *input*, int * *log*, int * *matrix*, float * *r_max*, float * *r_min*, float * *r_step*, char * *output*, int * *to*, int * *quiet*)

function that sets default values of variables

```
* set_defaults (&ang_max, &ang_min, &ang_step, cell, &from, input, &log, &r_max,
*                &r_min, &r_step, output, &to, &quiet);
*
```

Parameters

ang_max	maximal angle for RADF
ang_min	minimal angle for RADF
ang_step	angle step
cell	massive of cell size
from	first trajectory step
input	mask of trajectory files
log	status of log-mode
matrix	status of matrix-mode
r_max	maximal radius
r_min	minimal radius
r_step	radius step
output	output file name
to	last trajectory step
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/main.c File Reference

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "add_main.h"
#include "coords.h"
#include "messages.h"
#include "radf.h"
#include "radf_proc.h"
```

Functions

- int main (int argc, char ∗argv[])

5.3.1 Function Documentation

5.3.1.1 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

5.4 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

5.4.1 Function Documentation

5.4.1.1 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

5.5 src/radf.c File Reference

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

Macros

- #define M_PI 3.14159265358979323846

Functions

- int search_rdf (const int num_atoms, const int ∗type_atoms, const int ∗label_mol, const float ∗coords, const double r_min, const double r_max, const double r_step, int ∗radf)

  function that searchs molecule for rdf massive
- int search_rdf_centr (const int num_atoms, const int ∗type_atoms, const int ∗label_mol, const float ∗coords, const double r_min, const double r_max, const double r_step, int ∗radf)

  function that searchs molecule for rdf massive by centered coordinates
- int search_radf (const int num_atoms, const int ∗type_atoms, const int ∗label_mol, const float ∗coords, const double r_min, const double r_max, const double r_step, const double ang_min, const double ang_max, const double ang_step, int ∗radf)

  function that searchs molecule for radf massive

5.5.1 Macro Definition Documentation

5.5.1.1 #define M_PI 3.14159265358979323846

5.5.2 Function Documentation

5.5.2.1 int search_radf (const int *num_atoms*, const int ∗ *type_atoms*, const int ∗ *label_mol*, const float ∗ *coords*, const double *r_min*, const double *r_max*, const double *r_step*, const double *ang_min*, const double *ang_max*, const double *ang_step*, int ∗ *radf*)

function that searchs molecule for radf massive

```
* search_radf (num_atoms, type_atoms, label_mol, coords, r_min, r_max, r_step,
*              ang_min, ang_max, ang_step, radf);
*
```

Parameters

num_atoms	number of atoms
type_atoms	massive of atom types
label_mol	massive of numbers of molecule for atoms
coords	massive of coordinates
r_min	minimal radius
r_max	maximal radius
r_step	radius step
ang_min	minimal angle
ang_max	maximal angle
ang_step	anlge step
radf	not normed RADF

Returns

0 - exit without errors
1 - error in set center (missing atoms)

5.5.2.2 int search_rdf (const int *num_atoms*, const int * *type_atoms*, const int * *label_mol*, const float * *coords*, const double *r_min*, const double *r_max*, const double *r_step*, int * *radf*)

function that searchs molecule for rdf massive

```
* search_rdf (num_atoms, type_atoms, label_mol, coords, r_min, r_max, r_step, radf);
*
```

Parameters

num_atoms	number of atoms
type_atoms	massive of atom types
label_mol	massive of numbers of molecule for atoms
coords	massive of coordinates
r_min	minimal radius
r_max	maximal radius
r_step	radius step
radf	not normed RDF

Returns

0 - exit without errors

5.5.2.3 int search_rdf_centr (const int *num_atoms*, const int * *type_atoms*, const int * *label_mol*, const float * *coords*, const double *r_min*, const double *r_max*, const double *r_step*, int * *radf*)

function that searchs molecule for rdf massive by centered coordinates

```
* search_rdf_centr (num_atoms, type_atoms, label_mol, coords, r_min, r_max, r_step,
*                   radf);
*
```

Parameters

num_atoms	number of atoms
type_atoms	massive of atom types
label_mol	massive of numbers of molecule for atoms
coords	massive of coordinates
r_min	minimal radius
r_max	maximal radius
r_step	radius step
radf	not normed RDF

Returns

0 - exit without errors
1 - error in set center (missing atoms)

5.6 src/radf_proc.c File Reference

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

Macros

- #define M_PI 3.14159265358979323846

Functions

- int print_result (const char *output, const int matrix, const int mode, const int step, const int num_atoms, const double r_min, const double r_max, const double r_step, const double ang_min, const double ang_max, const double ang_step, const float *cell, const int *radf)

  function that print result to output file

5.6.1 Macro Definition Documentation

5.6.1.1 #define M_PI 3.14159265358979323846

5.6.2 Function Documentation

5.6.2.1 int print_result (const char * *output*, const int *matrix*, const int *mode*, const int *step*, const int *num_atoms*, const double *r_min*, const double *r_max*, const double *r_step*, const double *ang_min*, const double *ang_max*, const double *ang_step*, const float * *cell*, const int * *radf*)

function that print result to output file

```
* print_result (output, matrix, mode, step, num_atoms, r_min, r_max, r_step, ang_min,
*               ang_max, ang_step, cell, radf);
*
```

Parameters

output	output file name

matrix	status of matrix-mode
mode	1 - if RDF, 2 - if RDF for center mass, 3 - if RADF
step	$(to - from + 1)
num_atoms	number of atoms
r_min	minimal radius
r_max	maximal radius
r_step	radius step
ang_min	minimal angle
ang_max	maximal angle
ang_step	angle step
cell	cell size
radf	not normed RADF

Returns

0 - exit without errors

Index