

mm_radf
V.1.0.2

Generated by Doxygen 1.8.5

Fri Aug 30 2013 15:16:49

Contents

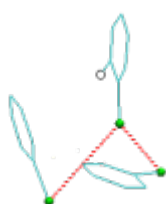
1	mm_radf	1
1.1	Introduction	1
1.2	How to use	1
2	Install	3
2.1	Requirements	3
2.2	How to install	3
2.2.1	Linux	3
2.2.2	Windows	3
3	Changelog	5
4	File Index	7
4.1	File List	7
5	File Documentation	9
5.1	src/add_main.c File Reference	9
5.1.1	Function Documentation	9
5.1.1.1	error_checking	9
5.1.1.2	print_message	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/main.c File Reference	13
5.3.1	Function Documentation	13
5.3.1.1	main	13
5.4	src/messages.c File Reference	13
5.4.1	Function Documentation	13
5.4.1.1	message	13
5.5	src/radf.c File Reference	14
5.5.1	Macro Definition Documentation	14

5.5.1.1	M_PI	14
5.5.2	Function Documentation	14
5.5.2.1	search_rdf	14
5.5.2.2	search_rdf	15
5.5.2.3	search_rdf_cent	15
5.6	src/rdf_proc.c File Reference	16
5.6.1	Macro Definition Documentation	16
5.6.1.1	M_PI	16
5.6.2	Function Documentation	16
5.6.2.1	print_result	16

Index	18
--------------	-----------

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 ]
```

Parameters:

- | | |
|-----|---|
| -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 automatically) |
| -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 \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.1.0.2 (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/ main.c	13
src/ messages.c	13
src/ radf.c	14
src/ radf_proc.c	16

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

<i>cell</i>	massive of cell size
-------------	----------------------

<i>from</i>	first trajectory step
<i>input</i>	input file name
<i>num_needed_at</i>	number of needed atom types
<i>needed_at</i>	massive of number of needed atom types
<i>output</i>	output file name
<i>to</i>	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

<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 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

<i>output</i>	output file nams
<i>log</i>	status of log-mode

<i>quiet</i>	status of quiet-mode
<i>matrix</i>	status of matrix-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>mode</i>	0 - if RDF, 1 - if RDF for center mass, 2 - if RADF
<i>r_max</i>	maximal radius
<i>r_min</i>	minimal radius
<i>r_step</i>	radius step
<i>ang_max</i>	maximal angle for RADF
<i>ang_min</i>	minimal angle for RADF
<i>ang_step</i>	angle step for RADF
<i>needed_at</i>	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

<i>ang_max</i>	maximal angle for RADF
<i>ang_min</i>	minimal angle for RADF
<i>ang_step</i>	angle step
<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>matrix</i>	status of matrix-mode
<i>r_max</i>	maximal radius
<i>r_min</i>	minimal radius
<i>r_step</i>	radius step
<i>output</i>	output file name
<i>to</i>	last trajectory step
<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/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

<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

5.5 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 searches molecule for rdf massive
- `int search_rdf_cent (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 searches 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 searches 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 searches 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

<i>num_atoms</i>	number of atoms
<i>type_atoms</i>	massive of atom types
<i>label_mol</i>	massive of numbers of molecule for atoms
<i>coords</i>	massive of coordinates
<i>r_min</i>	minimal radius
<i>r_max</i>	maximal radius
<i>r_step</i>	radius step
<i>ang_min</i>	minimal angle
<i>ang_max</i>	maximal angle
<i>ang_step</i>	angle step
<i>radf</i>	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 searches molecule for rdf massive

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

Parameters

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

Returns

- 0 - exit without errors

5.5.2.3 `int search_rdf_cent (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 searches molecule for rdf massive by centered coordinates

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

Parameters

<i>num_atoms</i>	number of atoms
<i>type_atoms</i>	massive of atom types
<i>label_mol</i>	massive of numbers of molecule for atoms
<i>coords</i>	massive of coordinates
<i>r_min</i>	minimal radius
<i>r_max</i>	maximal radius
<i>r_step</i>	radius step
<i>radf</i>	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

<i>output</i>	output file name
---------------	------------------

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

Returns

0 - exit without errors

Index

- add_main.c
 - error_checking, [9](#)
 - print_message, [10](#)
 - printing_head, [10](#)
 - set_defaults, [11](#)
- coords.c
 - reading_coords, [12](#)
- error_checking
 - add_main.c, [9](#)
- M_PI
 - radf.c, [14](#)
 - radf_proc.c, [16](#)
- main
 - main.c, [13](#)
- main.c
 - main, [13](#)
- message
 - messages.c, [13](#)
- messages.c
 - message, [13](#)
- print_message
 - add_main.c, [10](#)
- print_result
 - radf_proc.c, [16](#)
- printing_head
 - add_main.c, [10](#)
- radf.c
 - M_PI, [14](#)
 - search_radf, [14](#)
 - search_rdf, [15](#)
 - search_rdf_centr, [15](#)
- radf_proc.c
 - M_PI, [16](#)
 - print_result, [16](#)
- reading_coords
 - coords.c, [12](#)
- search_radf
 - radf.c, [14](#)
- search_rdf
 - radf.c, [15](#)
- search_rdf_centr
 - radf.c, [15](#)
- set_defaults
 - add_main.c, [11](#)
- src/add_main.c, [9](#)
- src/coords.c, [11](#)
- src/main.c, [13](#)
- src/messages.c, [13](#)
- src/radf.c, [14](#)
- src/radf_proc.c, [16](#)