

PLSA (baseline)

1.0.1

Generated by Doxygen 1.5.6

Sun Mar 7 16:16:27 2010

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	cooccur Struct Reference	5
3.1.1	Field Documentation	5
3.1.1.1	x	5
3.1.1.2	column	5
3.2	info Struct Reference	6
3.2.1	Field Documentation	7
3.2.1.1	verbose	7
3.2.1.2	debug	7
3.2.1.3	textio	7
3.2.1.4	rounding	7
3.2.1.5	no_output	7
3.2.1.6	seed	7
3.2.1.7	num_clusters	7
3.2.1.8	base_fn	7
3.2.1.9	maxiter	7
3.2.1.10	m	7
3.2.1.11	n	7
3.2.1.12	co_fn	8
3.2.1.13	cos	8
3.2.1.14	row_ids	8
3.2.1.15	column_ids	8
3.2.1.16	iter	8

3.2.1.17	probw1_z	8
3.2.1.18	probw2_z	8
3.2.1.19	probz	8
3.2.1.20	probz_w1w2	8
3.2.1.21	world_id	8
3.2.1.22	world_size	8
3.2.1.23	block_start	9
3.2.1.24	block_end	9
3.2.1.25	block_size	9
3.2.1.26	sigfpe_count	9
3.2.1.27	program_start	9
3.2.1.28	run_time	9
3.2.1.29	readCO_time	9
3.2.1.30	initEM_time	9
3.2.1.31	calculateML_time	9
3.2.1.32	applyEStep_time	9
3.2.1.33	applyMStep_time	9
3.2.1.34	normalizeProbs_time	9
3.2.1.35	printCoProbs_time	9
3.2.1.36	program_end	9
3.3	wmstruct Struct Reference	10
3.3.1	Field Documentation	10
3.3.1.1	ptr	10
3.3.1.2	size	10
3.3.1.3	file	10
3.3.1.4	line	10
3.3.1.5	next	10
4	File Documentation	11
4.1	config.h File Reference	11
4.1.1	Define Documentation	12
4.1.1.1	HAVE__BOOL	12
4.1.1.2	HAVE_FLOAT_H	12
4.1.1.3	HAVE_INTTYPES_H	12
4.1.1.4	HAVE_LIMITS_H	12
4.1.1.5	HAVE_MALLOC	12
4.1.1.6	HAVE_MEMORY_H	12

4.1.1.7	HAVE_REALLOC	12
4.1.1.8	HAVE_STDBOOL_H	12
4.1.1.9	HAVE_STDINT_H	12
4.1.1.10	HAVE_STDLIB_H	12
4.1.1.11	HAVE_STRING_H	12
4.1.1.12	HAVE_STRINGS_H	12
4.1.1.13	HAVE_SYS_STAT_H	12
4.1.1.14	HAVE_SYS_TYPES_H	12
4.1.1.15	HAVE_UNISTD_H	12
4.1.1.16	PACKAGE	12
4.1.1.17	PACKAGE_BUGREPORT	12
4.1.1.18	PACKAGE_NAME	12
4.1.1.19	PACKAGE_STRING	12
4.1.1.20	PACKAGE_TARNAME	12
4.1.1.21	PACKAGE_VERSION	12
4.1.1.22	STDC_HEADERS	12
4.1.1.23	VERSION	12
4.2	debug.c File Reference	13
4.2.1	Function Documentation	13
4.2.1.1	checkCoProb	13
4.2.1.2	debugCheckCo	13
4.2.1.3	handler_sigfpe	13
4.3	debug.h File Reference	14
4.3.1	Function Documentation	14
4.3.1.1	checkCoProb	14
4.3.1.2	debugCheckCo	14
4.3.1.3	handler_sigfpe	14
4.4	em-estep.c File Reference	15
4.4.1	Function Documentation	15
4.4.1.1	applyEStep	15
4.4.1.2	calculateML	15
4.4.1.3	initEM	15
4.5	em-estep.h File Reference	16
4.5.1	Function Documentation	16
4.5.1.1	applyEStep	16
4.5.1.2	calculateML	16

4.5.1.3	initEM	16
4.6	em-mstep.c File Reference	17
4.6.1	Function Documentation	17
4.6.1.1	applyMStep	17
4.6.1.2	normalizeProbs	17
4.7	em-mstep.h File Reference	18
4.7.1	Function Documentation	18
4.7.1.1	applyMStep	18
4.7.1.2	normalizeProbs	18
4.8	input.c File Reference	19
4.8.1	Function Documentation	19
4.8.1.1	initializePostInput	19
4.8.1.2	readCO	19
4.9	input.h File Reference	20
4.9.1	Function Documentation	20
4.9.1.1	initializePostInput	20
4.9.1.2	readCO	20
4.10	main.c File Reference	21
4.10.1	Function Documentation	21
4.10.1.1	main	21
4.11	output.c File Reference	22
4.11.1	Function Documentation	22
4.11.1.1	printCoProb	22
4.12	output.h File Reference	23
4.12.1	Function Documentation	23
4.12.1.1	printCoProb	23
4.13	parameters.c File Reference	24
4.13.1	Define Documentation	24
4.13.1.1	_GNU_SOURCE	24
4.13.2	Function Documentation	24
4.13.2.1	checkSettings	24
4.13.2.2	processOptions	24
4.13.2.3	usage	24
4.14	parameters.h File Reference	25
4.14.1	Function Documentation	25
4.14.1.1	checkSettings	25

4.14.1.2	processOptions	25
4.14.1.3	usage	25
4.15	plsa-defn.h File Reference	26
4.15.1	Define Documentation	27
4.15.1.1	BLOCK_SIZE	27
4.15.1.2	DBL_LESS	27
4.15.1.3	DOEXP	27
4.15.1.4	DOLOG	27
4.15.1.5	DOLOG1PEXP	27
4.15.1.6	DOLOGONE	27
4.15.1.7	FCLOSE	27
4.15.1.8	FOPEN	27
4.15.1.9	GET_COS	27
4.15.1.10	GET_COS_POSITION	27
4.15.1.11	GET_PROBW1_Z	27
4.15.1.12	GET_PROBW2_Z	28
4.15.1.13	GET_PROBZ	28
4.15.1.14	GET_PROBZ_W1W2	28
4.15.1.15	LN_LIMIT	28
4.15.1.16	logSumsInline	28
4.15.1.17	MAINPROC	28
4.15.1.18	MIN_PROB	28
4.15.1.19	ML_DELTA	28
4.15.1.20	PROGRESS_MSG	28
4.15.1.21	RANDOM_FLOAT	29
4.15.1.22	ROUND_DIGITS	29
4.15.1.23	SET_COS	29
4.15.2	Typedef Documentation	29
4.15.2.1	COOCCUR	29
4.15.2.2	INFO	29
4.15.2.3	PROBNODE	29
4.16	run.c File Reference	30
4.16.1	Function Documentation	30
4.16.1.1	initialize	30
4.16.1.2	run	30
4.16.1.3	uninitialize	30

4.17	run.h File Reference	31
4.17.1	Function Documentation	31
4.17.1.1	initialize	31
4.17.1.2	run	31
4.17.1.3	uninitialize	31
4.18	wmalloc.c File Reference	32
4.18.1	Function Documentation	33
4.18.1.1	countFree	33
4.18.1.2	countMalloc	33
4.18.1.3	hash	33
4.18.1.4	initWMalloc	33
4.18.1.5	printInUseWMalloc	33
4.18.1.6	printWMalloc	33
4.18.1.7	wfree	33
4.18.1.8	wmalloc	33
4.18.1.9	wrealloc	33
4.18.2	Variable Documentation	33
4.18.2.1	inuse_malloc	33
4.18.2.2	max_malloc	33
4.18.2.3	tempstr	33
4.18.2.4	wm_array	33
4.19	wmalloc.h File Reference	34
4.19.1	Define Documentation	35
4.19.1.1	TEMPSTRLEN	35
4.19.1.2	WM_SIZE	35
4.19.2	Typedef Documentation	35
4.19.2.1	WMSTRUCT	35
4.19.3	Function Documentation	35
4.19.3.1	countFree	35
4.19.3.2	countMalloc	35
4.19.3.3	initWMalloc	35
4.19.3.4	printInUseWMalloc	35
4.19.3.5	printWMalloc	35
4.19.3.6	wfree	35
4.19.3.7	wmalloc	35
4.19.3.8	wrealloc	35

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

cooccur	5
info	6
wmstruct	10

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

config.h	11
debug.c	13
debug.h	14
em-estep.c	15
em-estep.h	16
em-mstep.c	17
em-mstep.h	18
input.c	19
input.h	20
main.c	21
output.c	22
output.h	23
parameters.c	24
parameters.h	25
plsa-defn.h	26
run.c	30
run.h	31
wmalloc.c	32
wmalloc.h	34

Chapter 3

Data Structure Documentation

3.1 cooccur Struct Reference

```
#include <plsa-defn.h>
```

Data Fields

- [PROBNODE x](#)
- unsigned int [column](#)

3.1.1 Field Documentation

3.1.1.1 PROBNODE cooccur::x

The co-occurrence count, as a log value

3.1.1.2 unsigned int cooccur::column

Column position of this value

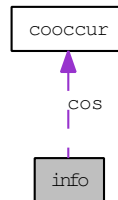
The documentation for this struct was generated from the following file:

- [plsa-defn.h](#)

3.2 info Struct Reference

```
#include <plsa-defn.h>
```

Collaboration diagram for info:



Data Fields

- bool [verbose](#)
- bool [debug](#)
- bool [textio](#)
- bool [rounding](#)
- bool [no_output](#)
- unsigned int [seed](#)
- unsigned int [num_clusters](#)
- char * [base_fn](#)
- unsigned int [maxiter](#)
- unsigned int [m](#)
- unsigned int [n](#)
- char * [co_fn](#)
- [COOCCUR](#) ** [cos](#)
- unsigned int * [row_ids](#)
- unsigned int * [column_ids](#)
- unsigned int [iter](#)
- [PROBNODE](#) * [probw1_z](#)
- [PROBNODE](#) * [probw2_z](#)
- [PROBNODE](#) * [probz](#)
- [PROBNODE](#) ** [probz_w1w2](#)
- signed int [world_id](#)
- signed int [world_size](#)
- unsigned int [block_start](#)
- unsigned int [block_end](#)
- unsigned int [block_size](#)
- unsigned int [sigfpe_count](#)
- time_t [program_start](#)
- double [run_time](#)
- double [readCO_time](#)
- double [initEM_time](#)
- double [calculateML_time](#)
- double [applyEStep_time](#)
- double [applyMStep_time](#)
- double [normalizeProbs_time](#)
- double [printCoProbs_time](#)
- time_t [program_end](#)

3.2.1 Field Documentation

3.2.1.1 bool info::verbose

Verbose output?

3.2.1.2 bool info::debug

Debugging output?

3.2.1.3 bool info::textio

Text I/O

3.2.1.4 bool info::rounding

Should the output values be rounded?

3.2.1.5 bool info::no_output

Suppress output

3.2.1.6 unsigned int info::seed

Random seed

3.2.1.7 unsigned int info::num_clusters

Number of clusters

3.2.1.8 char* info::base_fn

Base filename for the output file

3.2.1.9 unsigned int info::maxiter

Maximum number of iterations

3.2.1.10 unsigned int info::m

Number of unique query terms

3.2.1.11 unsigned int info::n

Number of terms in the document collection

3.2.1.12 char* info::co_fn

Co-occurrence filename

3.2.1.13 COOCCUR info::cos**

Co-occurrence counts in a COOCCUR data structure

3.2.1.14 unsigned int* info::row_ids

List of row identifiers (m of them)

3.2.1.15 unsigned int* info::column_ids

List of column identifiers (m of them)

3.2.1.16 unsigned int info::iter

Iteration; only calculated by the main process and broadcasted to others

3.2.1.17 PROBNODE* info::probw1_z

$P(w1|z)$ of size $(k * m)$

3.2.1.18 PROBNODE* info::probw2_z

$P(w2|z)$ of size $(k * n)$

3.2.1.19 PROBNODE* info::probz

$P(z)$ of size (k) ; one-dimensional array does not need a pointer

3.2.1.20 PROBNODE info::probz_w1w2**

$P(z|w1w2)$ of size $(k * m * n)$

3.2.1.21 signed int info::world_id

ID of this process

3.2.1.22 signed int info::world_size

Number of processes total

3.2.1.23 unsigned int info::block_start

Starting block (cluster) for this process to handle

3.2.1.24 unsigned int info::block_end

Ending block (cluster) for this process to handle

3.2.1.25 unsigned int info::block_size

Size of the block for this process to handle

3.2.1.26 unsigned int info::sigfpe_count

Number of floating point exception errors

3.2.1.27 time_t info::program_start**3.2.1.28 double info::run_time****3.2.1.29 double info::readCO_time****3.2.1.30 double info::initEM_time****3.2.1.31 double info::calculateML_time****3.2.1.32 double info::applyEStep_time****3.2.1.33 double info::applyMStep_time****3.2.1.34 double info::normalizeProbs_time****3.2.1.35 double info::printCoProbs_time****3.2.1.36 time_t info::program_end**

The documentation for this struct was generated from the following file:

- [plsa-defn.h](#)

3.3 wmstruct Struct Reference

```
#include <wmalloc.h>
```

Collaboration diagram for wmstruct:



Data Fields

- void * [ptr](#)
- size_t [size](#)
- char * [file](#)
- unsigned int [line](#)
- struct [wmstruct](#) * [next](#)

3.3.1 Field Documentation

3.3.1.1 void* [wmstruct::ptr](#)

3.3.1.2 size_t [wmstruct::size](#)

3.3.1.3 char* [wmstruct::file](#)

3.3.1.4 unsigned int [wmstruct::line](#)

3.3.1.5 struct [wmstruct](#)* [wmstruct::next](#) [[read](#)]

The documentation for this struct was generated from the following file:

- [wmalloc.h](#)

Chapter 4

File Documentation

4.1 config.h File Reference

Defines

- #define [HAVE_FLOAT_H](#) 1
- #define [HAVE_INTTYPES_H](#) 1
- #define [HAVE_LIMITS_H](#) 1
- #define [HAVE_MALLOC](#) 1
- #define [HAVE_MEMORY_H](#) 1
- #define [HAVE_REALLOC](#) 1
- #define [HAVE_STDBOOL_H](#) 1
- #define [HAVE_STDINT_H](#) 1
- #define [HAVE_STDLIB_H](#) 1
- #define [HAVE_STRINGS_H](#) 1
- #define [HAVE_STRING_H](#) 1
- #define [HAVE_SYS_STAT_H](#) 1
- #define [HAVE_SYS_TYPES_H](#) 1
- #define [HAVE_UNISTD_H](#) 1
- #define [HAVE__BOOL](#) 1
- #define [PACKAGE](#) "plsa-base"
- #define [PACKAGE_BUGREPORT](#) "r.wan@aist.go.jp"
- #define [PACKAGE_NAME](#) "PLSA-BASE"
- #define [PACKAGE_STRING](#) "PLSA-BASE 1.0"
- #define [PACKAGE_TARNAME](#) "plsa-base"
- #define [PACKAGE_VERSION](#) "1.0"
- #define [STDC_HEADERS](#) 1
- #define [VERSION](#) "1.0"

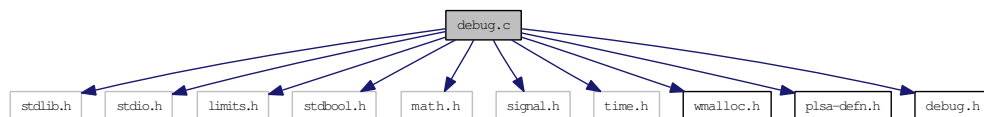
4.1.1 Define Documentation

- 4.1.1.1 `#define HAVE__BOOL 1`
- 4.1.1.2 `#define HAVE_FLOAT_H 1`
- 4.1.1.3 `#define HAVE_INTTYPES_H 1`
- 4.1.1.4 `#define HAVE_LIMITS_H 1`
- 4.1.1.5 `#define HAVE_MALLOC 1`
- 4.1.1.6 `#define HAVE_MEMORY_H 1`
- 4.1.1.7 `#define HAVE_REALLOC 1`
- 4.1.1.8 `#define HAVE_STDBOOL_H 1`
- 4.1.1.9 `#define HAVE_STDINT_H 1`
- 4.1.1.10 `#define HAVE_STDLIB_H 1`
- 4.1.1.11 `#define HAVE_STRING_H 1`
- 4.1.1.12 `#define HAVE_STRINGS_H 1`
- 4.1.1.13 `#define HAVE_SYS_STAT_H 1`
- 4.1.1.14 `#define HAVE_SYS_TYPES_H 1`
- 4.1.1.15 `#define HAVE_UNISTD_H 1`
- 4.1.1.16 `#define PACKAGE "plsa-base"`
- 4.1.1.17 `#define PACKAGE_BUGREPORT "r.wan@aist.go.jp"`
- 4.1.1.18 `#define PACKAGE_NAME "PLSA-BASE"`
- 4.1.1.19 `#define PACKAGE_STRING "PLSA-BASE 1.0"`
- 4.1.1.20 `#define PACKAGE_TARNAME "plsa-base"`
- 4.1.1.21 `#define PACKAGE_VERSION "1.0"`
- 4.1.1.22 `#define STDC_HEADERS 1`
- 4.1.1.23 `#define VERSION "1.0"`

4.2 debug.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>
#include <math.h>
#include <signal.h>
#include <time.h>
#include "wmalloc.h"
#include "plsa-defn.h"
#include "debug.h"
```

Include dependency graph for debug.c:



Functions

- void [handler_sigfpe](#) ()
- void [debugCheckCo](#) (INFO **info*)
- void [checkCoProb](#) (INFO **info*)

4.2.1 Function Documentation

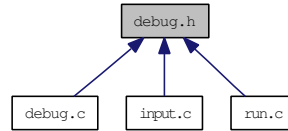
4.2.1.1 void [checkCoProb](#) (INFO **info*)

4.2.1.2 void [debugCheckCo](#) (INFO **info*)

4.2.1.3 void [handler_sigfpe](#) ()

4.3 debug.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [handler_sigfpe](#) ()
- void [debugCheckCo](#) (INFO **info*)
- void [checkCoProb](#) (INFO **info*)

4.3.1 Function Documentation

4.3.1.1 void [checkCoProb](#) (INFO **info*)

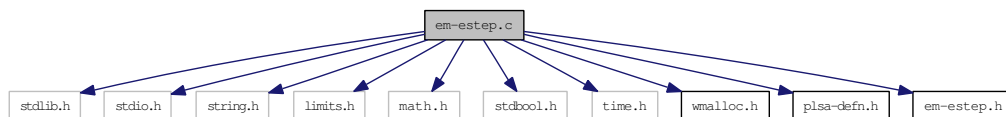
4.3.1.2 void [debugCheckCo](#) (INFO **info*)

4.3.1.3 void [handler_sigfpe](#) ()

4.4 em-estep.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <time.h>
#include "wmalloc.h"
#include "plsa-defn.h"
#include "em-estep.h"
```

Include dependency graph for em-estep.c:



Functions

- void [initEM](#) ([INFO](#) *[info](#))
- void [applyEStep](#) ([INFO](#) *[info](#))
- [PROBNODE](#) [calculateML](#) ([INFO](#) *[info](#))

4.4.1 Function Documentation

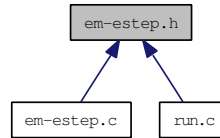
4.4.1.1 void [applyEStep](#) ([INFO](#) * *info*)

4.4.1.2 [PROBNODE](#) [calculateML](#) ([INFO](#) * *info*)

4.4.1.3 void [initEM](#) ([INFO](#) * *info*)

4.5 em-estep.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [initEM](#) (INFO **info*)
- void [applyEStep](#) (INFO **info*)
- [PROBNODE calculateML](#) (INFO **info*)

4.5.1 Function Documentation

4.5.1.1 void [applyEStep](#) (INFO * *info*)

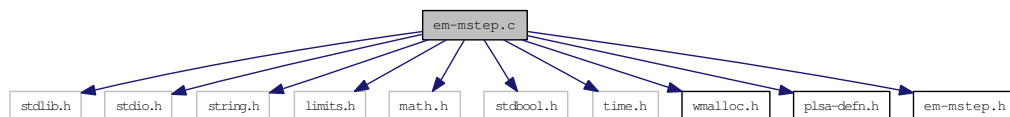
4.5.1.2 [PROBNODE calculateML](#) (INFO * *info*)

4.5.1.3 void [initEM](#) (INFO * *info*)

4.6 em-mstep.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <time.h>
#include "wmalloc.h"
#include "plsa-defn.h"
#include "em-mstep.h"
```

Include dependency graph for em-mstep.c:



Functions

- void [applyMStep](#) (INFO *info)
- void [normalizeProbs](#) (INFO *info)

4.6.1 Function Documentation

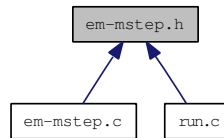
4.6.1.1 void [applyMStep](#) (INFO *info)

4.6.1.2 void [normalizeProbs](#) (INFO *info)

Normalize probabilities

4.7 em-mstep.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [applyMStep](#) (INFO **info*)
- void [normalizeProbs](#) (INFO **info*)

4.7.1 Function Documentation

4.7.1.1 void [applyMStep](#) (INFO * *info*)

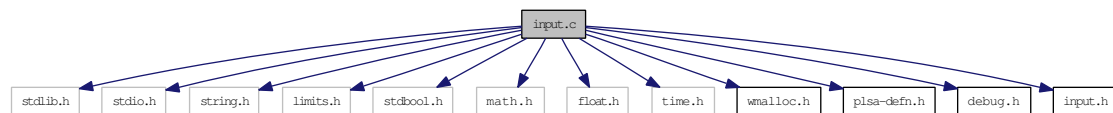
4.7.1.2 void [normalizeProbs](#) (INFO * *info*)

Normalize probabilities

4.8 input.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <limits.h>
#include <stdbool.h>
#include <math.h>
#include <float.h>
#include <time.h>
#include "wmalloc.h"
#include "plsa-defn.h"
#include "debug.h"
#include "input.h"
```

Include dependency graph for input.c:



Functions

- void [initializePostInput](#) (INFO *info)
- bool [readCO](#) (INFO *info)

4.8.1 Function Documentation

4.8.1.1 void initializePostInput (INFO * info)

Initialization that depends on the input file or parameters

4.8.1.2 bool readCO (INFO * info)

Read the co-occurrence data from file. The format of the file is:

```
[rows][columns][row id+][column id+][w1 cos_count (w21 c21) ... (w2n c2n)]+**
```

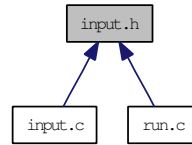
row and column ids are integer values that map to the original vocabulary. The number of values should be (info->m) and (info->n), respectively.

Every value is an unsigned integer in binary format, unless textmode is TRUE – if so, values are in text, separated by white space (tab).

Note: i indexes for rows (w1); j indexes for columns (w2)

4.9 input.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void `initializePostInput (INFO *info)`
- bool `readCO (INFO *info)`

4.9.1 Function Documentation

4.9.1.1 void initializePostInput (INFO * info)

Initialization that depends on the input file or parameters

4.9.1.2 bool readCO (INFO * info)

Read the co-occurrence data from file. The format of the file is:

```
[rows][columns][row id+][column id+][w1 cos_count (w21 c21) ... (w2n c2n)]+**
```

row and column ids are integer values that map to the original vocabulary. The number of values should be (`info` -> `m`) and (`info` -> `n`), respectively.

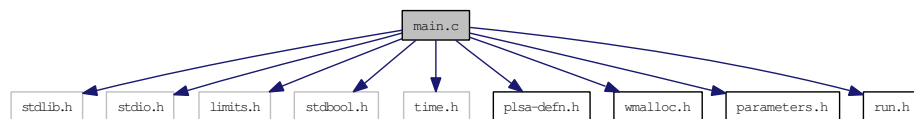
Every value is an unsigned integer in binary format, unless `textmode` is `TRUE` – if so, values are in text, separated by white space (tab).

Note: `i` indexes for rows (`w1`); `j` indexes for columns (`w2`)

4.10 main.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>
#include <time.h>
#include "plsa-defn.h"
#include "wmalloc.h"
#include "parameters.h"
#include "run.h"
```

Include dependency graph for main.c:



Functions

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

4.10.1 Function Documentation

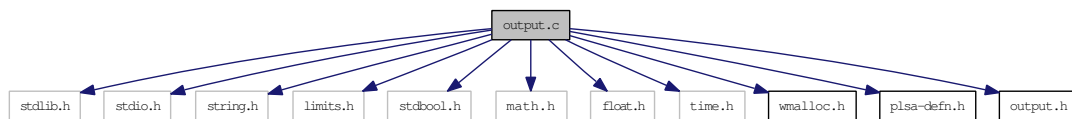
4.10.1.1 `int main (int argc, char * argv[])`

Main function

4.11 output.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <limits.h>
#include <stdbool.h>
#include <math.h>
#include <float.h>
#include <time.h>
#include "wmalloc.h"
#include "plsa-defn.h"
#include "output.h"
```

Include dependency graph for output.c:



Functions

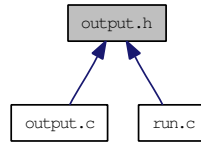
- void [printCoProb](#) (INFO *info)

4.11.1 Function Documentation

4.11.1.1 void [printCoProb](#) (INFO * *info*)

4.12 output.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [printCoProb](#) (INFO **info*)

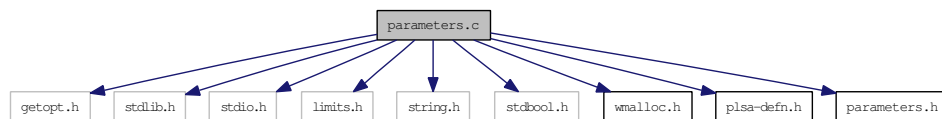
4.12.1 Function Documentation

4.12.1.1 void [printCoProb](#) (INFO **info*)

4.13 parameters.c File Reference

```
#include <getopt.h>
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <string.h>
#include <stdbool.h>
#include "wmalloc.h"
#include "plsa-defn.h"
#include "parameters.h"
```

Include dependency graph for parameters.c:



Defines

- #define [_GNU_SOURCE](#)

Functions

- void [usage](#) (char *progrname)
- bool [checkSettings](#) (INFO *info)
- bool [processOptions](#) (int argc, char *argv[], INFO *info)

4.13.1 Define Documentation

4.13.1.1 #define _GNU_SOURCE

4.13.2 Function Documentation

4.13.2.1 bool checkSettings (INFO * info)

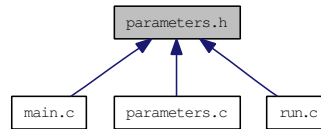
4.13.2.2 bool processOptions (int argc, char * argv[], INFO * info)

4.13.2.3 void usage (char * progrname)

Print out usage information

4.14 parameters.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [usage](#) (char *progrname)
- bool [checkSettings](#) ([INFO](#) **info*)
- bool [processOptions](#) (int argc, char *argv[], [INFO](#) **info*)

4.14.1 Function Documentation

4.14.1.1 bool [checkSettings](#) ([INFO](#) * *info*)

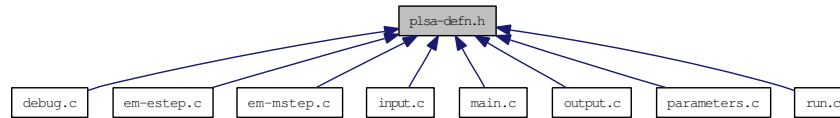
4.14.1.2 bool [processOptions](#) (int *argc*, char * *argv*[], [INFO](#) * *info*)

4.14.1.3 void [usage](#) (char * *progrname*)

Print out usage information

4.15 plsa-defn.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [cooccur](#)
- struct [info](#)

Defines

- #define [LN_LIMIT](#) 23.02585093
- #define [MIN_PROB](#) (1.0E-24)
- #define [DOLOG](#)(X) (logf (X))
- #define [DOEXP](#)(X) (expf (X))
- #define [DOLOGONE](#)(X) (log1pf (X))
- #define [DOLOGIPEXP](#)(x) DOLOGONE(DOEXP(x))
- #define [RANDOM_FLOAT](#) (([PROBNODE](#))rand () / (([PROBNODE](#))RAND_MAX + ([PROBNODE](#))1.0))
- #define [DBL_LESS](#)(A, B) ((B - A) > DBL_EPSILON)
- #define [ML_DELTA](#) 0.001
- #define [MAINPROC](#) 0
- #define [ROUND_DIGITS](#) 100000000
- #define [BLOCK_SIZE](#)(id, p, n) (BLOCK_LOW ((id) + 1, p, n)-BLOCK_LOW(id, p, n))
- #define [PROGRESS_MSG](#)(A)
- #define [FOPEN](#)(FILENAME, FP, MODE)
- #define [FCLOSE](#)(FP) (void) fclose (FP);
- #define [SET_COS](#)(W, X, Y, Z)
- #define [GET_COS](#)(W, X) ([info](#) → cos[W][X].x)
- #define [GET_COS_POSITION](#)(W, X) ([info](#) → cos[W][X].column)
- #define [GET_PROBW1_Z](#)(X, Y) ([info](#) → probw1_z[X * [info](#) → m + Y])
- #define [GET_PROBW2_Z](#)(X, Y) ([info](#) → probw2_z[X * [info](#) → n + Y])
- #define [GET_PROBZ](#)(X) ([info](#) → probz[X])
- #define [GET_PROBZ_W1W2](#)(W, X, Y) ([info](#) → probz_w1w2[W][X * [info](#) → n + Y])
- #define [logSumsInline](#)(A, B)

Typedefs

- typedef double [PROBNODE](#)
- typedef struct [cooccur](#) [COOCCUR](#)
- typedef struct [info](#) [INFO](#)

4.15.1 Define Documentation

4.15.1.1 #define BLOCK_SIZE(id, p, n) (BLOCK_LOW ((id) + 1, p, n)-BLOCK_LOW(id, p, n))

4.15.1.2 #define DBL_LESS(A, B) ((B - A) > DBL_EPSILON)

Test if two double values are close to each other

4.15.1.3 #define DOEXP(X) (expf (X))

Macro to perform the exp function

4.15.1.4 #define DOLOG(X) (logf (X))

Macro to perform a log

4.15.1.5 #define DOLOG1PEXP(x) DOLOGONE(DOEXP(x))

Macro to perform log (1 + expt(x))

4.15.1.6 #define DOLOGONE(X) (log1pf (X))

Macro to perform log (1 + x)

4.15.1.7 #define FCLOSE(FP) (void) fclose (FP);

4.15.1.8 #define FOPEN(FILENAME, FP, MODE)

Value:

```
FP = fopen ((char*) FILENAME, MODE); \
if (FP == NULL) { \
    fprintf (stderr, "Error %s %s.\n", (strcmp (MODE, "w") == 0) ? "creating" : "opening", FILENAME); \
    exit (EXIT_FAILURE); \
}
```

4.15.1.9 #define GET_COS(W, X) (info → cos[W][X].x)

Function to retrieve the position from the cooccurrence array

4.15.1.10 #define GET_COS_POSITION(W, X) (info → cos[W][X].column)

Function to retrieve the cooccurrence count from the cooccurrence array

4.15.1.11 #define GET_PROBW1_Z(X, Y) (info → probw1_z[X * info → m + Y])

Function to retrieve from $P(w_1|z)$; translate 2D to 1D co-ordinates

4.15.1.12 #define GET_PROBW2_Z(X, Y) (info → probw2_z[X * info → n + Y])

Function to retrieve from $P(w_2|z)$; translate 2D to 1D co-ordinates

4.15.1.13 #define GET_PROBZ(X) (info → probz[X])

Function to retrieve from $P(z)$

4.15.1.14 #define GET_PROBZ_W1W2(W, X, Y) (info → probz_w1w2[W][X * info → n + Y])

Function to retrieve from $P(z|w_1w_2)$; translate 3D to 1D co-ordinates

4.15.1.15 #define LN_LIMIT 23.02585093

Accuracy of floating point values as a log (base e) value, multiplied by -1

4.15.1.16 #define logSumsInline(A, B)

Value:

```
{
    register PROBNODE x, y; \
    if (A > B) { \
        x = A; y = B; \
    } \
    else { \
        x = B; y = A; \
    } \
    \
    /* a > b */ \
    \
    A = (fabs (y - x) > LN_LIMIT) ? x : x + DOLOG1PEXP (y - x); \
}
```

4.15.1.17 #define MAINPROC 0

ID of the main processor is always 0

4.15.1.18 #define MIN_PROB (1.0E-24)

Minimum probability

4.15.1.19 #define ML_DELTA 0.001

Minimum difference between two maximum likelihoods

4.15.1.20 #define PROGRESS_MSG(A)

Value:

```
if (info -> verbose) { \
    fprintf (stderr, "=="\t%s\n", A); \
}
```

Define'd function to indicate program progress

4.15.1.21 **#define RANDOM_FLOAT ((PROBNODE)rand () / ((PROBNODE)RAND_MAX + (PROBNODE)1.0))**

Generate a random number between [0, 1); cast to floating point first to prevent overflow

4.15.1.22 **#define ROUND_DIGITS 100000000**

Number of digits to round; used when outputting to binary only

4.15.1.23 **#define SET_COS(W, X, Y, Z)**

Value:

```
{ \
    info -> cos[W][X].column = Y; \
    info -> cos[W][X].x = Z; \
}
```

Function to retrieve from the cooccurrence array

4.15.2 Typedef Documentation

4.15.2.1 **typedef struct cooccur COOCCUR**

4.15.2.2 **typedef struct info INFO**

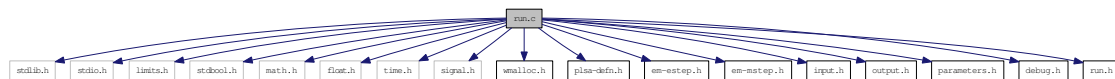
4.15.2.3 **typedef double PROBNODE**

Data type to use for probabilities

4.16 run.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>
#include <math.h>
#include <float.h>
#include <time.h>
#include <signal.h>
#include "wmalloc.h"
#include "plsa-defn.h"
#include "em-estep.h"
#include "em-mstep.h"
#include "input.h"
#include "output.h"
#include "parameters.h"
#include "debug.h"
#include "run.h"
```

Include dependency graph for run.c:



Functions

- **INFO** * [initialize](#) ()
- void [uninitialize](#) (**INFO** *[info](#))
- bool [run](#) (**INFO** *[info](#))

4.16.1 Function Documentation

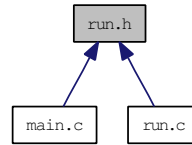
4.16.1.1 **INFO*** [initialize](#) ()

4.16.1.2 bool [run](#) (**INFO** * *info*)

4.16.1.3 void [uninitialize](#) (**INFO** * *info*)

4.17 run.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `INFO * initialize ()`
- `void uninitialize (INFO *info)`
- `bool run (INFO *info)`

4.17.1 Function Documentation

4.17.1.1 `INFO* initialize ()`

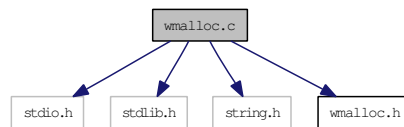
4.17.1.2 `bool run (INFO * info)`

4.17.1.3 `void uninitialize (INFO * info)`

4.18 wmalloc.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "wmalloc.h"
```

Include dependency graph for wmalloc.c:



Functions

- void * [wmalloc](#) (size_t y_arg)
- void * [wrealloc](#) (void *x_arg, size_t y_arg)
- void [wfree](#) (void *x_arg)
- static unsigned int [hash](#) (char *v, signed int M)
- void [initWMalloc](#) ()
- void [printWMalloc](#) ()
- void [printInUseWMalloc](#) (void)
- void [countMalloc](#) (void *ptr, size_t amount, const char *file, const unsigned int line)
- void [countFree](#) (void *ptr)

Variables

- static unsigned int [inuse_malloc](#) = 0
- static unsigned int [max_malloc](#) = 0
- static [WMSTRUCT](#) ** [wm_array](#)
- static char * [tempstr](#)

4.18.1 Function Documentation

4.18.1.1 void countFree (void * *ptr*)

4.18.1.2 void countMalloc (void * *ptr*, size_t *amount*, const char * *file*, const unsigned int *line*)

4.18.1.3 static unsigned int hash (char * *v*, signed int *M*) [static]

4.18.1.4 void initWMalloc (void)

4.18.1.5 void printInUseWMalloc (void)

4.18.1.6 void printWMalloc (void)

4.18.1.7 void wfree (void * *x_arg*)

4.18.1.8 void* wmalloc (size_t *y_arg*)

4.18.1.9 void* wrealloc (void * *x_arg*, size_t *y_arg*)

4.18.2 Variable Documentation

4.18.2.1 unsigned int inuse_malloc = 0 [static]

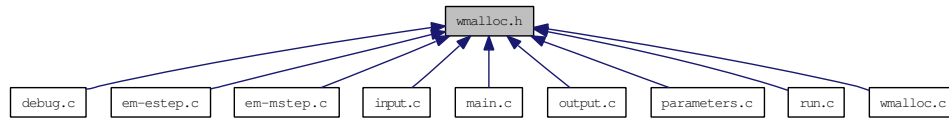
4.18.2.2 unsigned int max_malloc = 0 [static]

4.18.2.3 char* tempstr [static]

4.18.2.4 WMSTRUCT** wm_array [static]

4.19 wmalloc.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [wmstruct](#)

Defines

- #define [WM_SIZE](#) 65536
- #define [TEMPSTRLEN](#) 80

Typedefs

- typedef struct [wmstruct](#) [WMSTRUCT](#)

Functions

- void * [wmalloc](#) (size_t y_arg)
- void * [wrealloc](#) (void *x_arg, size_t y_arg)
- void [wfree](#) (void *x_arg)
- void [initWMalloc](#) (void)
- void [printWMalloc](#) (void)
- void [printInUseWMalloc](#) (void)
- void [countMalloc](#) (void *ptr, size_t amount, const char *file, unsigned int line)
- void [countFree](#) (void *ptr)

4.19.1 Define Documentation

4.19.1.1 `#define TEMPSTRLEN 80`

4.19.1.2 `#define WM_SIZE 65536`

4.19.2 Typedef Documentation

4.19.2.1 `typedef struct wmstruct WMSTRUCT`

4.19.3 Function Documentation

4.19.3.1 `void countFree (void * ptr)`

4.19.3.2 `void countMalloc (void * ptr, size_t amount, const char * file, unsigned int line)`

4.19.3.3 `void initWMalloc (void)`

4.19.3.4 `void printInUseWMalloc (void)`

4.19.3.5 `void printWMalloc (void)`

4.19.3.6 `void wfree (void * x_arg)`

4.19.3.7 `void* wmalloc (size_t y_arg)`

4.19.3.8 `void* wrealloc (void * x_arg, size_t y_arg)`

Index

- [_GNU_SOURCE](#)
 - [parameters.c](#), [24](#)
- [applyEStep](#)
 - [em-estep.c](#), [15](#)
 - [em-estep.h](#), [16](#)
- [applyEStep_time](#)
 - [info](#), [9](#)
- [applyMStep](#)
 - [em-mstep.c](#), [17](#)
 - [em-mstep.h](#), [18](#)
- [applyMStep_time](#)
 - [info](#), [9](#)
- [base_fn](#)
 - [info](#), [7](#)
- [block_end](#)
 - [info](#), [9](#)
- [BLOCK_SIZE](#)
 - [plsa-defn.h](#), [27](#)
- [block_size](#)
 - [info](#), [9](#)
- [block_start](#)
 - [info](#), [8](#)
- [calculateML](#)
 - [em-estep.c](#), [15](#)
 - [em-estep.h](#), [16](#)
- [calculateML_time](#)
 - [info](#), [9](#)
- [checkCoProb](#)
 - [debug.c](#), [13](#)
 - [debug.h](#), [14](#)
- [checkSettings](#)
 - [parameters.c](#), [24](#)
 - [parameters.h](#), [25](#)
- [co_fn](#)
 - [info](#), [7](#)
- [column](#)
 - [cooccur](#), [5](#)
- [column_ids](#)
 - [info](#), [8](#)
- [config.h](#), [11](#)
 - [HAVE__BOOL](#), [12](#)
 - [HAVE_FLOAT_H](#), [12](#)
 - [HAVE_INTTYPES_H](#), [12](#)
 - [HAVE_LIMITS_H](#), [12](#)
 - [HAVE_MALLOC](#), [12](#)
 - [HAVE_MEMORY_H](#), [12](#)
 - [HAVE_REALLOC](#), [12](#)
 - [HAVE_STDBOOL_H](#), [12](#)
 - [HAVE_STDINT_H](#), [12](#)
 - [HAVE_STDLIB_H](#), [12](#)
 - [HAVE_STRING_H](#), [12](#)
 - [HAVE_STRINGS_H](#), [12](#)
 - [HAVE_SYS_STAT_H](#), [12](#)
 - [HAVE_SYS_TYPES_H](#), [12](#)
 - [HAVE_UNISTD_H](#), [12](#)
 - [PACKAGE](#), [12](#)
 - [PACKAGE_BUGREPORT](#), [12](#)
 - [PACKAGE_NAME](#), [12](#)
 - [PACKAGE_STRING](#), [12](#)
 - [PACKAGE_TARNAME](#), [12](#)
 - [PACKAGE_VERSION](#), [12](#)
 - [STDC_HEADERS](#), [12](#)
 - [VERSION](#), [12](#)
- [COOCCUR](#)
 - [plsa-defn.h](#), [29](#)
- [cooccur](#), [5](#)
 - [column](#), [5](#)
 - [x](#), [5](#)
- [cos](#)
 - [info](#), [8](#)
- [countFree](#)
 - [wmalloc.c](#), [33](#)
 - [wmalloc.h](#), [35](#)
- [countMalloc](#)
 - [wmalloc.c](#), [33](#)
 - [wmalloc.h](#), [35](#)
- [DBL_LESS](#)
 - [plsa-defn.h](#), [27](#)
- [debug](#)
 - [info](#), [7](#)
- [debug.c](#), [13](#)
 - [checkCoProb](#), [13](#)
 - [debugCheckCo](#), [13](#)
 - [handler_sigfpe](#), [13](#)
- [debug.h](#), [14](#)
 - [checkCoProb](#), [14](#)

- debugCheckCo, [14](#)
 - handler_sigfpe, [14](#)
- debugCheckCo
 - debug.c, [13](#)
 - debug.h, [14](#)
- DOEXP
 - plsa-defn.h, [27](#)
- DOLOG
 - plsa-defn.h, [27](#)
- DOLOG1PEXP
 - plsa-defn.h, [27](#)
- DOLOGONE
 - plsa-defn.h, [27](#)
- em-estep.c, [15](#)
 - applyEStep, [15](#)
 - calculateML, [15](#)
 - initEM, [15](#)
- em-estep.h, [16](#)
 - applyEStep, [16](#)
 - calculateML, [16](#)
 - initEM, [16](#)
- em-mstep.c, [17](#)
 - applyMStep, [17](#)
 - normalizeProbs, [17](#)
- em-mstep.h, [18](#)
 - applyMStep, [18](#)
 - normalizeProbs, [18](#)
- FCLOSE
 - plsa-defn.h, [27](#)
- file
 - wmstruct, [10](#)
- FOPEN
 - plsa-defn.h, [27](#)
- GET_COS
 - plsa-defn.h, [27](#)
- GET_COS_POSITION
 - plsa-defn.h, [27](#)
- GET_PROBW1_Z
 - plsa-defn.h, [27](#)
- GET_PROBW2_Z
 - plsa-defn.h, [27](#)
- GET_PROBZ
 - plsa-defn.h, [28](#)
- GET_PROBZ_W1W2
 - plsa-defn.h, [28](#)
- handler_sigfpe
 - debug.c, [13](#)
 - debug.h, [14](#)
- hash
 - wmalloc.c, [33](#)
- HAVE__BOOL
 - config.h, [12](#)
- HAVE_FLOAT_H
 - config.h, [12](#)
- HAVE_INTTYPES_H
 - config.h, [12](#)
- HAVE_LIMITS_H
 - config.h, [12](#)
- HAVE_MALLOC
 - config.h, [12](#)
- HAVE_MEMORY_H
 - config.h, [12](#)
- HAVE_REALLOC
 - config.h, [12](#)
- HAVE_STDBOOL_H
 - config.h, [12](#)
- HAVE_STDINT_H
 - config.h, [12](#)
- HAVE_STDLIB_H
 - config.h, [12](#)
- HAVE_STRING_H
 - config.h, [12](#)
- HAVE_STRINGS_H
 - config.h, [12](#)
- HAVE_SYS_STAT_H
 - config.h, [12](#)
- HAVE_SYS_TYPES_H
 - config.h, [12](#)
- HAVE_UNISTD_H
 - config.h, [12](#)
- INFO
 - plsa-defn.h, [29](#)
- info, [6](#)
 - applyEStep_time, [9](#)
 - applyMStep_time, [9](#)
 - base_fn, [7](#)
 - block_end, [9](#)
 - block_size, [9](#)
 - block_start, [8](#)
 - calculateML_time, [9](#)
 - co_fn, [7](#)
 - column_ids, [8](#)
 - cos, [8](#)
 - debug, [7](#)
 - initEM_time, [9](#)
 - iter, [8](#)
 - m, [7](#)
 - maxiter, [7](#)
 - n, [7](#)
 - no_output, [7](#)
 - normalizeProbs_time, [9](#)
 - num_clusters, [7](#)
 - printCoProbs_time, [9](#)

- probw1_z, 8
 - probw2_z, 8
 - probz, 8
 - probz_w1w2, 8
 - program_end, 9
 - program_start, 9
 - readCO_time, 9
 - rounding, 7
 - row_ids, 8
 - run_time, 9
 - seed, 7
 - sigfpe_count, 9
 - textio, 7
 - verbose, 7
 - world_id, 8
 - world_size, 8
- initEM
 - em-estep.c, 15
 - em-estep.h, 16
- initEM_time
 - info, 9
- initialize
 - run.c, 30
 - run.h, 31
- initializePostInput
 - input.c, 19
 - input.h, 20
- initWMalloc
 - wmalloc.c, 33
 - wmalloc.h, 35
- input.c, 19
 - initializePostInput, 19
 - readCO, 19
- input.h, 20
 - initializePostInput, 20
 - readCO, 20
- inuse_malloc
 - wmalloc.c, 33
- iter
 - info, 8
- line
 - wmstruct, 10
- LN_LIMIT
 - plsa-defn.h, 28
- logSumsInline
 - plsa-defn.h, 28
- m
 - info, 7
- main
 - main.c, 21
- main.c, 21
 - main, 21
- MAINPROC
 - plsa-defn.h, 28
- max_malloc
 - wmalloc.c, 33
- maxiter
 - info, 7
- MIN_PROB
 - plsa-defn.h, 28
- ML_DELTA
 - plsa-defn.h, 28
- n
 - info, 7
- next
 - wmstruct, 10
- no_output
 - info, 7
- normalizeProbs
 - em-mstep.c, 17
 - em-mstep.h, 18
- normalizeProbs_time
 - info, 9
- num_clusters
 - info, 7
- output.c, 22
 - printCoProb, 22
- output.h, 23
 - printCoProb, 23
- PACKAGE
 - config.h, 12
- PACKAGE_BUGREPORT
 - config.h, 12
- PACKAGE_NAME
 - config.h, 12
- PACKAGE_STRING
 - config.h, 12
- PACKAGE_TARNAME
 - config.h, 12
- PACKAGE_VERSION
 - config.h, 12
- parameters.c, 24
 - _GNU_SOURCE, 24
 - checkSettings, 24
 - processOptions, 24
 - usage, 24
- parameters.h, 25
 - checkSettings, 25
 - processOptions, 25
 - usage, 25
- plsa-defn.h, 26
 - BLOCK_SIZE, 27
 - COOCCUR, 29

- DBL_LESS, [27](#)
- DOEXP, [27](#)
- DOLOG, [27](#)
- DOLOG1PEXP, [27](#)
- DOLOGONE, [27](#)
- FCLOSE, [27](#)
- FOPEN, [27](#)
- GET_COS, [27](#)
- GET_COS_POSITION, [27](#)
- GET_PROBW1_Z, [27](#)
- GET_PROBW2_Z, [27](#)
- GET_PROBZ, [28](#)
- GET_PROBZ_W1W2, [28](#)
- INFO, [29](#)
- LN_LIMIT, [28](#)
- logSumsInline, [28](#)
- MAINPROC, [28](#)
- MIN_PROB, [28](#)
- ML_DELTA, [28](#)
- PROBNODE, [29](#)
- PROGRESS_MSG, [28](#)
- RANDOM_FLOAT, [29](#)
- ROUND_DIGITS, [29](#)
- SET_COS, [29](#)
- printCoProb
 - output.c, [22](#)
 - output.h, [23](#)
- printCoProbs_time
 - info, [9](#)
- printInUseWMalloc
 - wmalloc.c, [33](#)
 - wmalloc.h, [35](#)
- printWMalloc
 - wmalloc.c, [33](#)
 - wmalloc.h, [35](#)
- PROBNODE
 - plsa-defn.h, [29](#)
- probw1_z
 - info, [8](#)
- probw2_z
 - info, [8](#)
- probz
 - info, [8](#)
- probz_w1w2
 - info, [8](#)
- processOptions
 - parameters.c, [24](#)
 - parameters.h, [25](#)
- program_end
 - info, [9](#)
- program_start
 - info, [9](#)
- PROGRESS_MSG
 - plsa-defn.h, [28](#)
- ptr
 - wmstruct, [10](#)
- RANDOM_FLOAT
 - plsa-defn.h, [29](#)
- readCO
 - input.c, [19](#)
 - input.h, [20](#)
- readCO_time
 - info, [9](#)
- ROUND_DIGITS
 - plsa-defn.h, [29](#)
- rounding
 - info, [7](#)
- row_ids
 - info, [8](#)
- run
 - run.c, [30](#)
 - run.h, [31](#)
- run.c, [30](#)
 - initialize, [30](#)
 - run, [30](#)
 - uninitialize, [30](#)
- run.h, [31](#)
 - initialize, [31](#)
 - run, [31](#)
 - uninitialize, [31](#)
- run_time
 - info, [9](#)
- seed
 - info, [7](#)
- SET_COS
 - plsa-defn.h, [29](#)
- sigfpe_count
 - info, [9](#)
- size
 - wmstruct, [10](#)
- STDC_HEADERS
 - config.h, [12](#)
- tempstr
 - wmalloc.c, [33](#)
- TEMPSTRLEN
 - wmalloc.h, [35](#)
- textio
 - info, [7](#)
- uninitialize
 - run.c, [30](#)
 - run.h, [31](#)
- usage
 - parameters.c, [24](#)
 - parameters.h, [25](#)

verbose
 info, [7](#)
VERSION
 config.h, [12](#)

wfree
 wmalloc.c, [33](#)
 wmalloc.h, [35](#)
wm_array
 wmalloc.c, [33](#)
WM_SIZE
 wmalloc.h, [35](#)
wmalloc
 wmalloc.c, [33](#)
 wmalloc.h, [35](#)
wmalloc.c, [32](#)
 countFree, [33](#)
 countMalloc, [33](#)
 hash, [33](#)
 initWMalloc, [33](#)
 inuse_malloc, [33](#)
 max_malloc, [33](#)
 printInUseWMalloc, [33](#)
 printWMalloc, [33](#)
 tempstr, [33](#)
 wfree, [33](#)
 wm_array, [33](#)
 wmalloc, [33](#)
 wrealloc, [33](#)
wmalloc.h, [34](#)
 countFree, [35](#)
 countMalloc, [35](#)
 initWMalloc, [35](#)
 printInUseWMalloc, [35](#)
 printWMalloc, [35](#)
 TEMPSTRLEN, [35](#)
 wfree, [35](#)
 WM_SIZE, [35](#)
 wmalloc, [35](#)
 WMSTRUCT, [35](#)
 wrealloc, [35](#)
WMSTRUCT
 wmalloc.h, [35](#)
wmstruct, [10](#)
 file, [10](#)
 line, [10](#)
 next, [10](#)
 ptr, [10](#)
 size, [10](#)
world_id
 info, [8](#)
world_size
 info, [8](#)
wrealloc
 wmalloc.c, [33](#)
 wmalloc.h, [35](#)

x
 cooccur, [5](#)