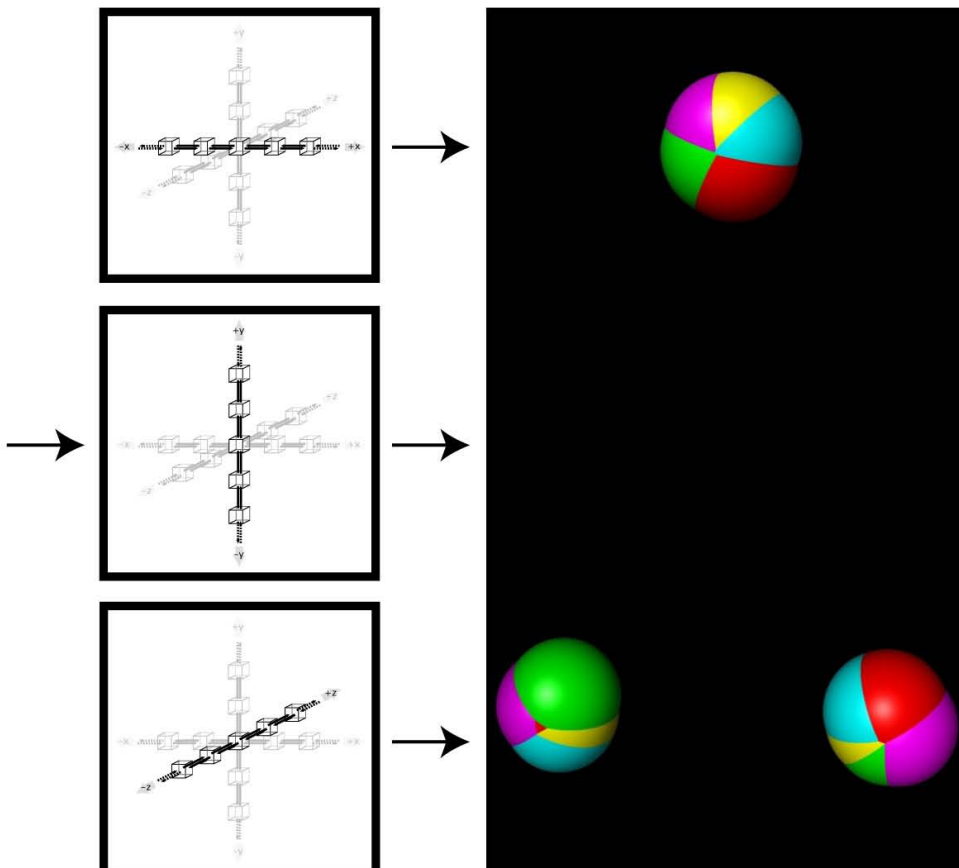


ATOM	9057	N	LEU	D3091	29.996	-26.876	100.443	1.00	23.81	N
ATOM	9058	CA	LEU	D3091	31.102	-26.433	101.296	1.00	23.13	C
ATOM	9059	C	LEU	D3091	30.932	-27.119	102.654	1.00	24.73	C
ATOM	9060	O	LEU	D3091	29.855	-27.037	103.240	1.00	25.85	N
ATOM	9061	CB	LEU	D3091	31.062	-24.879	101.483	1.00	22.59	C
ATOM	9062	CG	LEU	D3091	31.942	-24.258	102.571	1.00	19.79	C
ATOM	9063	CD1	LEU	D3091	33.446	-24.511	102.290	1.00	16.54	C
ATOM	9064	CD2	LEU	D3091	31.676	-22.670	102.596	1.00	16.08	C
ATOM	9065	N	ASN	D3092	31.981	-27.820	103.117	1.00	25.85	N
ATOM	9066	CA	ASN	D3092	32.093	-28.386	104.487	1.00	26.41	C
ATOM	9067	C	ASN	D3092	33.184	-27.597	105.278	1.00	26.65	C
ATOM	9068	O	ASN	D3092	34.331	-27.446	104.800	1.00	26.15	O
ATOM	9069	CB	ASN	D3092	32.394	-29.867	104.466	1.00	25.62	C
ATOM	9070	CG	ASN	D3092	31.384	-30.659	103.587	1.00	32.60	C
ATOM	9071	CD1	ASN	D3092	30.239	-30.257	103.502	1.00	36.96	O
ATOM	9072	ND2	ASN	D3092	31.812	-31.755	102.923	1.00	33.99	N
ATOM	9073	N	LEU	D3093	32.834	-27.103	106.474	1.00	25.72	N
ATOM	9074	CA	LEU	D3093	33.786	-26.294	107.224	1.00	27.23	C
ATOM	9075	C	LEU	D3093	34.854	-27.155	107.929	1.00	27.81	C
ATOM	9076	O	LEU	D3093	34.973	-27.077	109.121	1.00	28.68	O
ATOM	9077	CB	LEU	D3093	33.067	-25.347	108.208	1.00	25.64	C
ATOM	9078	CG	LEU	D3093	31.910	-24.598	107.554	1.00	23.71	C
ATOM	9079	CD1	LEU	D3093	31.102	-23.957	108.641	1.00	25.82	C
ATOM	9080	CD2	LEU	D3093	32.432	-23.491	106.614	1.00	23.20	C
ATOM	9081	N	THR	D3094	35.656	-27.908	107.177	1.00	28.59	N
ATOM	9082	CA	THR	D3094	36.618	-28.868	107.791	1.00	31.02	C
ATOM	9083	C	THR	D3094	37.972	-28.297	108.257	1.00	29.61	C
ATOM	9084	O	THR	D3094	38.695	-28.963	108.965	1.00	32.84	O
ATOM	9085	CB	THR	D3094	36.915	-30.044	106.832	1.00	31.61	C
ATOM	9086	CG1	THR	D3094	37.527	-29.508	105.627	1.00	38.49	O
ATOM	9087	CG2	THR	D3094	35.627	-30.714	106.330	1.00	29.32	C
ATOM	9088	N	LYS	D3095	38.330	-27.095	107.874	1.00	26.08	N
ATOM	9089	CA	LYS	D3095	39.670	-26.584	108.084	1.00	24.24	C
ATOM	9090	C	LYS	D3095	39.700	-25.545	109.236	1.00	23.84	C
ATOM	9091	O	LYS	D3095	38.648	-24.988	109.587	1.00	23.56	O
ATOM	9092	CB	LYS	D3095	40.144	-25.912	106.781	1.00	24.26	C
ATOM	9093	CG	LYS	D3095	39.947	-26.787	105.537	1.00	21.16	C
ATOM	9094	CD	LYS	D3095	40.674	-26.230	104.356	1.00	23.29	C
ATOM	9095	CE	LYS	D3095	40.385	-26.995	103.067	1.00	22.42	C
ATOM	9096	NZ	LYS	D3095	40.841	-26.164	101.922	1.00	23.95	N
ATOM	9097	N	PRO	D3096	40.892	-25.346	109.857	1.00	22.25	N
ATOM	9098	CA	PRO	D3096	41.024	-24.376	110.941	1.00	20.97	C
ATOM	9099	C	PRO	D3096	41.233	-22.988	110.370	1.00	19.58	C
ATOM	9100	O	PRO	D3096	41.409	-22.885	109.151	1.00	17.77	O
ATOM	9101	CB	PRO	D3096	42.295	-24.860	111.692	1.00	20.15	C
ATOM	9102	CG	PRO	D3096	43.134	-25.464	110.609	1.00	21.72	C
ATOM	9103	CD	PRO	D3096	42.136	-26.114	109.637	1.00	21.79	C
ATOM	9104	N	LEU	D3097	41.233	-21.967	111.236	1.00	19.31	N
ATOM	9105	CA	LEU	D3097	41.594	-20.624	110.793	1.00	20.71	C
ATOM	9106	C	LEU	D3097	43.085	-20.569	110.587	1.00	20.42	C
ATOM	9107	O	LEU	D3097	43.809	-21.255	111.321	1.00	21.70	O
ATOM	9108	CB	LEU	D3097	41.229	-19.561	111.827	1.00	20.69	C
ATOM	9109	CG	LEU	D3097	39.821	-18.973	111.943	1.00	24.31	C
ATOM	9110	CD1	LEU	D3097	39.843	-17.889	113.045	1.00	17.94	C
ATOM	9111	CD2	LEU	D3097	39.319	-18.421	110.589	1.00	18.76	C
ATOM	9112	N	CYS	D3098	43.555	-19.799	109.587	1.00	20.11	N
ATOM	9113	CA	CYS	D3098	44.977	-19.580	109.415	1.00	19.11	C
ATOM	9114	C	CYS	D3098	45.459	-18.685	110.560	1.00	19.61	C
ATOM	9115	O	CYS	D3098	44.682	-17.927	111.167	1.00	18.96	O
ATOM	9116	CB	CYS	D3098	45.283	-18.882	108.083	1.00	19.11	C
ATOM	9117	SG	CYS	D3098	44.702	-19.743	106.588	1.00	23.45	S
ATOM	9118	N	GLU	D3099	46.751	-18.761	110.846	1.00	18.90	N
ATOM	9119	CA	GLU	D3099	47.376	-17.916	111.834	1.00	18.32	C
ATOM	9120	C	GLU	D3099	47.408	-16.504	111.230	1.00	17.91	C
ATOM	9121	O	GLU	D3099	47.678	-16.382	110.018	1.00	18.55	O



## wwavePDB Overview

### WWAVEPDB

#### Overview

[wwavePDB Help Output](#)

[wwavePDB Tutorial](#)

### wwavePDB Overview

**wwavePDB** identifies atoms in a PDB file within a distance range of other atoms. **wwavePDB** is useful for identifying the common spatial occupancy of atoms. Atom output may be further restricted by set properties (e.g., heterogeneity of chain identifiers, chain identifier content, heterogeneity of atom type) and individual atom properties (e.g., atom type, atom chain, structural connectivity). Output is atom-based and is written as PDB output to new PDB files.

**wwavePDB** identifies:

- features without initial target feature input
- contact interface atoms
- binding site features, including between highly divergent structures
- distributed features within a structure or between structures that are not related to contact or binding but represent a structural feature of a group of molecules

**wwavePDB** can be used to:

- rapidly discover features in a molecular structure or molecular interaction without pre-specifying a search feature
- deliver atom sets for creating common reference surfaces that can be used to compactly display conformational differences between structurally related molecules
- find structural relationships in highly divergent structures
- identify sets of atoms associated with particular molecular functions test
- test whether particular common spatial occupancy of atoms in molecules are unique (i.e., not present in other data) and are consistent with their binding without pre-selecting atoms to be considered

---

## wwavePDB Help Output (“wwavePDB -h” output)

### NAME

wwavePDB -- identifies atoms in a PDB file within a distance range of other atoms

### SYNOPSIS

wwavePDB [options] (version 1.0.4)

CHARACTER OPTION	KEYWORD OPTION	DESCRIPTION	DEFAULT
------------------	----------------	-------------	---------

#### Options for input:

-i <filename> ....	--inputfn=<filename> ...	input pdb filename .....	required
-n # .....	--model=# .....	MODEL # within PDB to process .....	first model
-x .....	--read_water .....	include water atoms .....	no waters
-y .....	--read_hydrogen .....	include hydrogen atoms .....	no hydrogen

#### Options for output:

-o <prefix> .....	--output_prefix=<str> ..	output file prefix .....	none
-w <filename> ....	--wwfn=<filename> .....	complete filename for "_ww" file .....	stdout
-s [<filename>] ..	--sets .....	output "_sets" file, opt. w/name .....	no output
	--setsfn=<filename> ....	output "_sets" file, opt. w/name .....	no output
-l .....	--log .....	output execution summary to stderr ..	no log

#### Options for distance range:

-a # .....	--min=# .....	minimum distance between atoms .....	0.0
-z # .....	--max=# .....	maximum distance between atoms .....	infinity

Options for "\_ww" output:

```
-r <suboptions> .. --restrict_set=<opt> ... set restrictions ..... 3 or "any"
[1|2|3]{1}      pick one:                restrict set chain id content ..... 3 or "any"
 1 .....      "one_chain" ..... only homogeneous sets of chain ids
 2 .....      "two_or_more_chains" .. only heterogeneous sets of chain ids
 3 .....      "any_chains" ..... heterogeneous and/or homogeneous sets

[a|b|c]?        pick none or one:          restrict set atom content ..... none
 a .....      "only_atoms" ..... only ATOMS
 b .....      "only_hetatms" ..... only HETATMs
 c .....      "atom_and_hetatm" ..... 1+ ATOMS and also 1+ HETATMs

[g|h|i]?        pick one:                restrict set reference atom type .... i or "ref_any_type"
 g .....      "only_ref_atom" ..... only ATOMS
 h .....      "only_ref_hetatm" ..... only HETATMs
 i .....      "ref_any_type" ..... ATOMS and HETATMs

[w]?           pick none or one:          restrict set chain id content ..... none
 w .....      "includes_all_cids" ... subset of atoms w/all '-c' chain ids

[x|y|z]?        pick none or one:          restrict set chain id content ..... none
 x .....      "no_cids" ..... no atoms with a '-c' chain id
 y .....      "one_or_more_cids" .... 1+ atoms with a '-c' chain id
 z .....      "only_cids" ..... only atoms with a '-c' chain id

-f <suboptions> .. --filter=<option> ..... select output filters ..... 6 or "any"
[4|5|6]{1}      pick one:                filter set output by atom type ..... 6 or "any"
 4 .....      "output_only_atoms" ... output only ATOMS
 5 .....      "output_only_hetatms" . output only HETATMs
 6 .....      "output_any_atom_type" output any ATOMS and/or HETATMs

[r|t]?         pick none or one:          filter set output by chain id ..... none
 r .....      "output_only_cid" ..... output only atoms w/a '-c' chain id
 t .....      "output_no_cid" ..... output only atoms w/no '-c' chain ids

-c <chainids> ... --chainids=<chainids> .. for '-r', '-f', '--restrict_set=' ... none

-m <suboptions> .. --make_ww=<opt> ..... outputs additional "_ww" files ..... none
[1-5r]+ .....  pick one or more:
 1 .....      "charged" ..... make "_ww_charged" file
 2 .....      "no_mainchain" ..... make "_ww_no_mainchain" file
 3 .....      "charged_non_mainchain" make "_ww_charged_non_mainchain" file
 4 .....      "helices_and_beta" .... make "_ww_helices_and_beta" file
 5 .....      "non_mainchain_carbon" make "_ww_non_mainchain_carbon" file
12345 .....    "all" ..... make all the above additional files
 r .....      "residue" ..... make residue copies of "_ww" files
```

Options for "\_sets" output:

```
-q <suboptions> .. --column_order=<opt> ... column sort list by col.# or name ... 0 or "dist"

[[0-9]+|n] ..... pick the character string "none" or one or more:
 0 .....      "dist" ..... distance
 1 .....      "atype" ..... atom type
 2 .....      "aname" ..... atom name
 3 .....      "anum" ..... atom number
 4 .....      "chain" ..... chain id
 5 .....      "rname" ..... residue name
 6 .....      "rnum" ..... residue number
 7 .....      "x" ..... x coordinate
 8 .....      "y" ..... y coordinate
 9 .....      "z" ..... z coordinate
 n .....      "none" ..... NO SORTING
```

Other options:

```
-h .....      --help ..... print more help (Enter 'wwavePDB -h' for help.)
<NO OPTIONS> ..... shorter option synopsis (Just enter 'wwavePDB'.)
                --license ..... prints license terms for wwavePDB.
```

## DESCRIPTION

wwavePDB identifies atoms in a PDB file within a distance range of other atoms. Atom output may be further restricted by set properties (heterogeneity of chain identifiers, chain identifier content, heterogeneity of atom type) and individual atom properties (atom type, atom chain, structural connectivity). Output is atom-based and is written as PDB output to new PDB files.

wwavePDB is a program with a unix command line interface. Input is a file in pdb format. The structural output files, containing the atoms that satisfy the specified conditions, are also in the PDB format, and by default are named with the suffix "\_ww". "\_ww" files are subsets of the original input pdb file, and may be viewed with existing PDB display programs. It may be useful to display the original pdb input file with overlaid output "\_ww" files that are created with different distance ranges. By default, only the "\_ww" file is output.

An informational file showing the atoms sets may be output; this file is suffixed "\_sets" by default. The "\_sets" file lists each atom separately with its related atoms that satisfy the range and other specified set restriction and output requirements. The "\_sets" file is not in the PDB format. The "\_sets" file is not output by default.

The 'OPTIONS' section below describes how to control an individual execution of wwavePDB; this control is further broken down into subsections that reflect the basic functioning of wwavePDB:

- INPUT SPECIFICATION  
(specification of input atoms: PDB file, MODEL, whether water and hydrogens are read)
- DISTANCE RANGE AND REFERENCE ATOM TYPE SPECIFICATION  
(specification of the range allowed between atoms to create the initial sets; specification of the required atom type of each set's 'reference atom'.)
- SET RESTRICTION SPECIFICATION  
(further selection of sets by the set properties: heterogeneity of chain identifiers, chain identifier content, and heterogeneity of atom type)
- ATOM FILTERING SPECIFICATION  
(selection of atoms to be output from the restricted sets: either all atoms or by the atom properties atom type and atom chain)
- ALTERNATIVE "\_WW" FILES SPECIFICATION: CHARGE AND STRUCTURAL RESTRICTIONS  
(selection of additional output files with further structural restrictions: charged, no main chain, charged non main chain, helices and beta, non main chain carbon; and a separate option to make full residue copies of all structural files)
- INFORMATIONAL "\_SETS" FILE SPECIFICATION  
(optional informational output file with set information; column sort options)
- OUTPUT SPECIFICATION  
(specification of output: location, file names, directory paths)

## OPTIONS

### SINGLE CHARACTER OPTIONS VS. KEYWORD OPTIONS

Single character options and keyword options are used on the command line to specify the execution of that program: input, output, and the control of the data. Single character options, and associated option values, are terse. Keyword options, and associated option values, are descriptive.

Single character options start with a single hyphen (e.g., '-h'). If a single character option has an associated value, a space must separate the option from the value (e.g., '-o run1').

Keyword options start with double hyphens (e.g., '--help'). If a keyword option has an associated value, an '=' must immediately follow the keyword and be immediately followed by the value; there are no included spaces (e.g., '--output\_prefix=run1'). Most wwavePDB keyword options can be shortened:

Keyword_Option	Shortened_Keyword_Options
--chainids= .....	--chains ..... --cids=
--column_order= .....	--order=
--inputfn= .....	--input= ..... --in=
--make_ww= .....	--makeww= ..... --make=
--output_prefix= .....	--output= ..... --out=
--read_hydrogen= .....	--hydrogen= ..... --readh
--read_water= .....	--water= ..... --readw
--restrict_set= .....	--restrict=
--wwfn= .....	--ww=

Single character options and keyword options may be used together on the same command line as long as they do not conflict with each other; wwavePDB will object with an error if they conflict. All options accept terse or verbose option values (e.g., 'c' or "atom\_and\_hetatm"). Options that have multiple option values (e.g., 'column\_order=' ('-q') or '--make\_ww=' ('-m')) may have these option values be given as a list, with either no intervening spaces (e.g., '-q 240') or separated by a non-alphanumeric character (e.g., '--column\_order=aname,chain,dist').

#### INPUT SPECIFICATION

Character_Option	Keyword_Option	Description	Default
-i <filename> ....	--inputfn=<filename> ...	input pdb filename .....	required
-n # .....	--model=# .....	MODEL # within PDB to process .....	first model
-x .....	--read_water .....	include water atoms .....	no waters
-y .....	--read_hydrogen .....	include hydrogen atoms .....	no hydrogen

The input PDB file must be specified with the '--inputfn=' ('-i') option. If the input PDB file has no MODEL records, then all atoms will be processed; if MODEL records exist, the model specified with the '--model=' ('-n') option will be processed. If no model is specified, then only the atoms in the first model will be processed. The option '--read\_water' ('-x') specifies that water atoms are to be included in pdb atom input. The option '--read\_hydrogen' ('-y') specifies that hydrogen atoms are to be included in pdb atom input. By default, no water or hydrogen atoms are read in as pdb atom input.

#### OUTPUT SPECIFICATION

Character_Option	Keyword_Option	Description	Default
-o <prefix> .....	--output_prefix=<str> ..	output file prefix .....	none
-w <filename> ....	--wwfn=<filename> .....	complete filename for "_ww" file ....	stdout
-s [<filename>] ..	--sets .....	output "_sets" file, opt. w/name ....	no output
	--setsfn=<filename> ....	output "_sets" file, opt. w/name ....	no output
-l .....	--log .....	output execution summary to stderr ..	no log

The option '--output\_prefix=' ('-o') specifies an output filename prefix to be used for the output files. If the '--output\_prefix=' ('-o') option is used, output filenames will start with "user specified prefix" and end with: "\_sets" or "\_ww". Different suffixes will automatically be added reflecting "alternative structure" file content (e.g., "\_non\_mainchain\_carbon").

The '--setsfn=' ('-s') and '--wwfn=' ('-w') options override the '--output\_prefix=' ('-o') option and specify complete names for the "\_sets" and "\_ww" output files, respectively. If neither option '--output\_prefix=' ('-o') nor option '--wwfn=' ('-w') is used, the "\_ww" file will be written to stdout. ('stdout' and 'stderr' are specifications for unix file pointers that are normally output to your screen, but may be redirected.) If the option '--sets' ('-s') is specified without a filename, and the option '--output\_prefix=' ('-o') is not used, then the "\_sets" file will be output to stdout. wwavePDB will handle directions to simultaneously output both "\_sets" and "\_ww" files to stdout as an error.

Errors and warnings go to stderr. stderr may be redirected to a new file by appending ' 2>filename' to your wwavePDB command line; this will overwrite the previously existing file. stderr may be redirected to be appended to a possibly existing file, or to write a new file if the file does not exist, by appending ' 2>>filename' to your wwavePDB command. The option '--log' ('-l') specifies that an execution summary should be output to stderr. A log file, containing all wwavePDB execution summaries and any existing errors and warnings, may be made by using the '--log' ('-l') option and by appending ' 2>>filename' to your wwavePDB command. (But change 'filename', above, to the filename of your log file. Make sure to have a space separating your wwavePDB command arguments from the stderr redirection.)

Output file locations specified using options '--output\_prefix=' ('-o'), '--setsfn=' ('-s'), or '--wwfn=' ('-w') may include directory paths; files do not have to be in the immediate directory. Specified directories will be created if they do not already exist, assuming appropriate user ownership and permissions. (So a specific directory may be specified for program wwavePDB output by using the option '-output\_prefix=<path>'; e.g., '-o N6/lw1x' directs program wwavePDB output to be prefaced 'lw1x' and placed in a directory named 'N6').

NOTE: All output files will overwrite identically named existing files.

## DISTANCE RANGE SPECIFICATION

Character_Option	Keyword_Option	Description	Default
-a #	--min=#	minimum distance between atoms	0.0
-z #	--max=#	maximum distance between atoms	infinity

The keyword options '--min=' ('-a') and '--max=' ('-z') specify the required distance range as restricted minimum and maximum distances between atoms. An associated numeric value (either integer or real) is required for these keywords. If '--min=#' ('-a') is not specified then no minimum is required; the default value of '0.0' will be used. If '--max=#' ('-z') is not specified then no maximum is required; the default value of infinity will be used. While all atom distances are allowed by default, useful output requires specifying this distance range.

## SET RESTRICTION SPECIFICATION

Character_Option	Keyword_Option	Description	Default
-r <suboptions>	--restrict_set=<opt>	set restrictions	3 or "any"
[1 2 3]{1}	pick one:	restrict set chain id content	3 or "any"
1	"one_chain"	only homogeneous sets of chain ids	
2	"two_or_more_chains"	only heterogeneous sets of chain ids	
3	"any_chains"	heterogeneous and/or homogeneous sets	
[a b c]?	pick none or one:	restrict set atom content	none
a	"only_atoms"	only ATOMs	
b	"only_hetatms"	only HETATMs	
c	"atom_and_hetatm"	1+ ATOMs and also 1+ HETATMs	
[g h i]?	pick one:	restrict set reference atom type	i or "ref_any_type"
g	"only_ref_atom"	only ATOMs	
h	"only_ref_hetatm"	only HETATMs	
i	"ref_any_type"	ATOMs and HETATMs	
[w]?	pick none or one:	restrict set chain id content	none
w	"includes_all_cids"	subset of atoms w/all '-c' chain ids	
[x y z]?	pick none or one:	restrict set chain id content	none
x	"no_cids"	no atoms with a '-c' chain id	
y	"one_or_more_cids"	1+ atoms with a '-c' chain id	
z	"only_cids"	only atoms with a '-c' chain id	
-c <chainids>	--chainids=<chainids>	for '-r', '-f', '--restrict_set='	none

Each 'reference' atom and its associated atoms within the specified distance range, define an initial set. These initial sets are solely based on distance. By default, i.e. if no restrictive options other than '--min=' ('-a') and '--max=' ('-z') are used, the atoms in these sets will be output to the "\_ww" file. The keyword option '--restrict\_set=' ('-r') specifies further restriction of these initial "distance specified" sets; these initial sets may be culled by a restriction of the reference atom type or by the set properties: heterogeneity of chain identifiers, chain identifier content, and heterogeneity of atom type.

### Restricting the Atom Type of the Reference Atom

By default, the reference atom (i.e., the atom in each set from which other atoms are measured) may have an atom type of either 'ATOM' or 'HETATM'; this is the property defined by the '--restrict\_set=' ('-r') option value "ref\_any\_type" ("i"). The '--restrict\_set=' ('-r') option value "only\_ref\_atom" ("g") restricts reference atoms to those with the atom type 'ATOM'. The '--restrict\_set=' ('-r') option value "only\_ref\_hetatm" ("h") restricts reference atoms to those with the atom type 'HETATM'. One, and only one, of these options must be selected (if only by default).

### Heterogeneity of Chain Identifiers

By default, there are no restrictions on sets based upon chain identifiers (i.e., either all atoms in a set may have the same chain identifier or a set may have atoms with different chain identifiers); this is the property defined by the '--restrict\_set=' ('-r') option value "any\_chains" ("3"). The '--restrict\_set=' ('-r') option value "one\_chain" ("1") restricts sets to those sets that only have atoms with the same chain identifier. The '--restrict\_set=' ('-r') option value "two\_or\_more\_chains" ("2") restricts sets to those sets that have at least two atoms with different chain identifiers. One, and only one, of these options must be selected (if only by default).

## Heterogeneity of Atom Type

By default, sets may have any number of atoms with either an ATOM or HETATM atom type. Set restriction by atom type may be specified with the '--restrict\_set=' ('-r') option values: "only\_atoms" ("a") restricts sets to those sets with only ATOMs (and no HETATMs), "only\_hetatms" ("b") restricts sets to those sets with only HETATMs (and no ATOMs), "atom\_and\_hetatm" ("c") restricts sets to those sets that have at least one ATOM and also at least one HETATM. One or none of these options may be selected.

## Specifying a List of Chain Identifiers

The option '--chainids=' ('-c') requires as an option value a list of chain identifiers. This list of chain identifier characters should be consecutively listed (e.g., '-c ABC'). If characters other than alphanumeric are used as chain identifiers, the chain id characters may be enclosed in single quotes (e.g., ' ABC' for chain identifiers ' ', 'A', 'B', and 'C'.) A backslash '\' may be used to quote a single quote (e.g., '\') or a backslash (e.g., '\\') should such a character be used as a chain identifier. This list of chain identifiers is used with the option '--restrict\_set=' ('-r') and the option '--filter=' ('-f').

## Chain Identifier Content

By default, sets may have atoms with any chain identifier. Set restriction by chain identifier content may be specified with specific chain identifiers listed as the '--chainids=' ('-c') option value and with one of the following '--restrict\_set=' ('-r') option values: "no\_cids" ("x") restricts sets to those sets that contain no atoms having chain identifiers in the chain identifier list, "one\_or\_more\_cids" ("y") restricts sets to those sets that contain one or more atoms having chain identifiers in the chain identifier list, "only\_cids" ("z") restricts sets to those sets that contain only atoms having chain identifiers from the chain identifier list. One or none of these options may be selected.

The '--restrict\_set=' ('-r') option value "includes\_all\_cids" ("w") further restricts sets to those sets that contain a subset of atoms with all of the chain identifiers listed with option '--chainids=' ('-c').

## ATOM FILTERING SPECIFICATION

Character_Option	Keyword_Option	Description	Default
-f <suboptions>	--filter=<option>	select output filters	6 or "any"
[4 5 6]{1}	pick one:	filter set output by atom type	6 or "any"
4	"output_only_atoms"	output only ATOMs	
5	"output_only_hetatms"	output only HETATMs	
6	"output_any_atom_type"	output any ATOMs and/or HETATMs	
[r t]?	pick none or one:	filter set output by chain id	none
r	"output_only_cid"	output only atoms w/a '-c' chain id	
t	"output_no_cid"	output only atoms w/no '-c' chain ids	

Options have been described above to define sets based upon distance range and to restrict these sets by set properties. If no further options are used, the atoms in these sets will be output to the "\_ww" file. The keyword option '--filter=' ('-f') specifies which atoms of these sets will be output to the "\_ww" file by using filters based upon the atom properties atom type and atom chain.

## Filter by Atom Type

By default, all ATOMs or HETATMs of the sets satisfying the distance and the set restriction options will be output to the "\_ww" file; this is the property defined by the '--filter=' ('-f') option value "output\_any\_atom\_type" ("6"). The '--filter=' ('-f') option value "output\_only\_atom" ("4") restricts atom output to only atoms with the atom type 'ATOM'. The '--filter=' ('-f') option value "output\_only\_hetatms" ("5") restricts atom output to only atoms with the atom type 'HETATM'. One, and only one, of these options must be selected (if only by default).

## Filter by Chain Identifier

By default, all atoms, with any chain identifier, of the sets satisfying the distance and the set restriction options will be output to the "\_ww" file. Atom output restriction by chain identifier content may be specified with specific chain identifiers listed as the '--chainids=' ('-c') option value and with either of the following '--filter=' ('-f') option values: "output\_only\_cid" ("r") restricts atom output to only those atoms having chain identifiers specified in the chain identifier list, "output\_no\_cid" ("t") restricts atom output to only those atoms NOT having chain identifiers specified in the chain identifier list. One or none of these options may be selected.

## ALTERNATIVE "\_ww" FILES SPECIFICATION: CHARGE AND STRUCTURAL RESTRICTIONS

Character_Option	Keyword_Option	Description	Default
-m <suboptions>	.. --make_ww=<opt>	..... outputs additional "_ww" files	..... none
[1-5r]+ ..... pick one or more:			
1	.....	"charged" ..... make "_ww_charged" file	
2	.....	"no_mainchain" ..... make "_ww_no_mainchain" file	
3	.....	"charged_non_mainchain" make "_ww_charged_non_mainchain" file	
4	.....	"helices_and_beta" .... make "_ww_helices_and_beta" file	
5	.....	"non_mainchain_carbon" make "_ww_non_mainchain_carbon" file	
12345	.....	"all" ..... make all the above additional files	
r	.....	"residue" ..... make residue copies of "_ww" files	

Options have been described above to define sets based upon distance range, to optionally restrict these sets by set properties, and to optionally further restrict the atoms to be output by atom properties. A file with the suffix "\_ww" will be written with these atoms in PDB format.

Additional "\_ww" files can be created that have further structural or electrostatic restrictions to the atom output. The option '--make\_ww=' ('-m') creates these additional files with the following option values, named by atom output restriction: "charged" ("1"), "no\_mainchain" ("2"), "charged\_non\_mainchain" ("3"), "helices\_and\_beta" ("4"), and "non\_mainchain\_carbon" ("5"). These additional "\_ww" files will be named with suffixes of "\_ww\_" appended by the option value (e.g., "\_ww\_charged"). Note that '--make\_ww=all' may be used to make all of these alternative files without requiring individual specification.

Using the keyword option/value pair '--make\_ww=residue' or the terse option '-m r' will create "full residue" copies of all output "\_ww" files. These files will have "\_allres" further added to the filename suffix. The entire residue of every atom output to the original version of the specific "\_ww" file will be output to the "full residue" version of these "\_ww" files.

## INFORMATIONAL "\_SETS" FILE SPECIFICATION

Character_Option	Keyword_Option	Description	Default
-q <suboptions>	.. --column_order=<opt>	... column sort list by col.# or name	... 0 or "dist"
[[0-9]+[n] ..... pick the character string "none" or one or more:			
0	.....	"dist" ..... distance	
1	.....	"atype" ..... atom type	
2	.....	"aname" ..... atom name	
3	.....	"anum" ..... atom number	
4	.....	"chain" ..... chain id	
5	.....	"rname" ..... residue name	
6	.....	"rnum" ..... residue number	
7	.....	"x" ..... x coordinate	
8	.....	"y" ..... y coordinate	
9	.....	"z" ..... z coordinate	
n	.....	"none" ..... NO SORTING	

The option '--sets' ('-s') creates an optional informational file, suffixed "\_sets", that lists sets of atoms that match the execution restrictions. These sets consist of one line for each atom in that set, each atom line having 10 columns. These lines may be sorted by the column order specified as a list with the option '--column\_order=<column\_list>' ('-q'). Columns in the column list may be either specified as zero-ordered column numbers (0 - 9) or as short descriptors ("dist", "atype", "aname", "anum", "chain", "rname", "rnum", "x", "y", or "z"). The first listed column specification in the '--column\_order=' ('-q') list will be the primary column that will be sorted. If there are atom lines in a set that have identical values in that primary column, and a second column specification is listed in the '--column\_order=' ('-q') list, then the second column will be used as a secondary sort, and so on for further column specifications. The column specification list may have no delimiters (e.g., '-q 463') or it may be delimited by a non-alphanumeric character (e.g., '--column\_order=chain,rnum,anum'). The first column, distance, is sorted upon by default; this is equivalent to '--column\_order=0'. To specify that no sorting is to be done, use the sort descriptor "none" or the letter 'N'.

## HELP

Character_Option	Keyword_Option	Description	Default
-h	.....	--help	..... print more help (Enter 'wwavePDB -h' for help.)
<NO OPTIONS>	.....	.....	..... shorter option synopsis (Just enter 'wwavePDB'.)



## USE

While an individual execution of wwavePDB is simple, the use of wwavePDB may include running wwavePDB multiple times on an individual PDB file, running wwavePDB on several PDB files, and also combining wwavePDB output as input for additional wwavePDB executions.

Creating multiple "\_ww" files with different distance range restrictions may be helpful in identifying sets of atoms associated with particular molecular functions. Creating multiple "\_ww" files with different required distance ranges may be helpful in identifying common spatial occupancy. Options '--min=' ('-a') and '--max=' ('-z') set the required distance range between atoms. For example, run wwavePDB with the '--max=' ('-z') values: '7.0', '6.5', '6.0', '5.0', '4.0', '3.5', '3.0', and '2.5'. A maximum around 3.0 to 4.0 may be useful for identifying contact interface atoms. Distributed features within a structure are not necessarily restricted to a local cluster of atoms. For identifying distributed features whose common spatial occupancy is found in two different structures, the upper limit for a distance range would be the maximum distance between any of the atoms of the smaller of the structures being examined. Compare the results at different ranges!

When multiple structures are found that share a common spatial occupancy (i.e., there exists three or more specific atoms in each structure where the distances between each pair of specific atoms within each structure matches the distances between each corresponding pair of specific atoms in other structures), then there is the possibility of a shared structural feature.

Existing molecular display programs can be used to visualize the shared common spatial occupancy. Molecular display programs may have a "pair fitting" command to superposition structures. Alternatively, the free Weininger Works program 'twistPDB' will map the atom coordinates of one PDB file to the atom coordinates of a different PDB file given three specific atoms from each PDB file.

The option '--restrict\_set=atom\_and\_hetatm' ('-r c') specifies that, in each set of atoms within a range of a selected atom, there must be at least one ATOM and at least one HETATM; this may be useful if the initial input PDB file contains a substrate, and the resultant "\_ww" file is intended to contain identifying contact interface atoms. Conversely, lack of the options '--restrict\_set=only\_atoms' ('-r a'), '--restrict\_set=only\_hetatms' ('-r b'), and '--restrict\_set=atom\_and\_hetatm' ('-r c') specifies that selected atom sets, in the absence of other constraints, may include any number of ATOM or HETATM atoms; this may be useful when you have no HETATMs present in the original PDB file, and the resultant "\_ww" file is intended to contain conserved structure. Distributed features across multiple structures may be found by comparing the "\_ww" files containing conserved structure.

The option '--chainids=' ('-c') can be used with the '--restrict\_set=' ('-r') option and '--filter=' ('-f') option to further restrict output atom sets based on chain identifiers. This may be used to identify features related to one, or more than one chain. This may also be used to compare different molecules. The chain identifiers of a PDB file (or of wwavePDB output) can be easily changed with the free Weininger Works program 'chainidPDB'.

## EXAMPLES

The following example generates 2 output files, '1AIY\_sets.txt' and '1AIY\_ww.pdb', where all output atoms must be within 5.0 Angstroms of another atom.

```
wwavePDB --inputfn=1AIY.pdb --out_prefix=1AIY --max=5.0 --sets
```

or

```
wwavePDB -i 1AIY.pdb -o 1AIY -z 5.0 -s
```

```
-i 1AIY.pdb .. --inputfn=1AIY.pdb ..... input is from file '1AIY.pdb'  
-o 1AIY ..... --output_prefix=1AIY ..... output will be prepended with '1AIY'  
-z 5.0 ..... --max=5.0 ..... atoms must be within 5.0 A of each other  
-s ..... --sets ..... output '_sets' file
```

The following example is similar to the above, except that the files will only include sets of atoms that are a distance of 3.0 to 5.0 Angstroms apart and where each set also has at least one ATOM and one HETATM.

```
wwavePDB --inputfn=1AIY.pdb --out_prefix=1AIY --min=3.0 --max=5.0 --sets \
--restrict_set=any_chains --restrict_set=atom_and_hetatm \
--filter=output_any_atom_type
```

or

```
wwavePDB -i 1AIY.pdb -o 1AIY -a 3.0 -z 5.0 -s -r 3c -f 6
```

as above, and:

```
-a 3.0 ..... --min=3.0 ..... atoms must be further than 3.0 A apart
-r 3c ..... output "_ww" file, where sets have:
 3 ..... --restrict_set=any_chains ..... any # of chain identifiers
 c ..... --restrict_set=atom_and_hetatm ... 1+ ATOMS and 1+ HETATMs
-f 6 ..... output to "_ww" file:
 6 ..... --filter=output_any_atom_type .... all ATOMS and HETATMs from matching sets
```

The following example is similar to the above, except that one "\_sets" file and 12 "\_ww" files will be output into a new, if not already existing, directory 'laiy':

```
1AIY_sets.txt,
1AIY_ww.pdb, 1AIY_ww_allres.pdb,
1AIY_ww_charged.pdb, 1AIY_ww_charged_allres.pdb,
1AIY_ww_no_mainchain.pdb, 1AIY_ww_no_mainchain_allres.pdb,
1AIY_ww_charged_non_mainchain.pdb, 1AIY_ww_charged_non_mainchain_allres.pdb,
1AIY_ww_helices_and_beta.pdb, 1AIY_ww_helices_and_beta_allres.pdb,
1AIY_ww_non_mainchain_carbon.pdb, 1AIY_ww_non_mainchain_carbon.pdb.
```

```
wwavePDB --inputfn=1AIY.pdb --out_prefix=laiy/1AIY --min=3.0 --max=5.0 --sets \
--restrict_set=any_chains --restrict_set=atom_and_hetatm \
--filter=output_any_atom_type \
--make_ww=all --make_ww=residue
```

or

```
wwavePDB -i 1AIY.pdb -o laiy/1AIY -a 3.0 -z 5.0 -s -r 3c -f 6 -m 12345r
```

as above, and:

```
-o laiy/1AIY . --out_prefix=laiy/1AIY ..... output to dir. 'laiy' with '1AIY' prefix
-m 12345 ..... --make_ww=all ..... output additional "_ww" files:
 1 ..... "_ww_charged"
 2 ..... "_ww_no_mainchain"
 3 ..... "_ww_charged_non_mainchain"
 4 ..... "_ww_helices_and_beta", and
 5 ..... "_ww_non_mainchain_carbon"
 r..... --make_ww=residue ..... make "full residue" copies of "_ww" files
```

The following example generates the file '1AIY\_KL\_ww.pdb' containing atoms of the chains 'K' and 'L' in 1AIY.pdb that are within a distance of 3.3 Angstroms, and also generates a file '1AIY\_KL\_sets.txt' containing the sets that fulfill the specified restrictions as lists ordered primarily by chain identifier, secondarily by atom name, and lastly by distance.

```
wwavePDB --inputfn=1AIY.pdb --output_prefix=1AIY_KL \
--restrict_set=any_chains --restrict_set=includes_all_cids \
--filter=output_any_atom_type --filter=output_only_cid --chainids=KL \
--max=3.3 --sets --column_order=chain,aname,dist
```

or, more concisely (using keyword abbreviations and option value lists)

```
wwavePDB --in=1AIY.pdb --out=1AIY_KL --restrict=any_chains,includes_all_cids \
--filter=output_any_atom_type,output_only_cid --chains=KL \
--max=3.3 --sets --order=chain,aname,dist
```

or, more succinctly (as above and leaving out defaults)

```
wwavePDB --in=1AIY.pdb --out=1AIY_KL --max=3.3 --order=chain,aname,dist \
--restrict=includes_all_cids --filter=output_only_cid --chains=KL
```

or, more tersely (using single character options)

```
wwavePDB -i 1AIY.pdb -o 1AIY_KL -r 3w -f 6r -c 'KL' -z 3.3 -s -q 420
```

```
-i 1AIY.pdb .. --inputfn=1AIY.pdb ..... input is from file '1AIY.pdb'
-o 1AIY ..... --output_prefix=1AIY ..... output will be prepended with '1AIY_KL'
-r 3w ..... output "_ww" file, where sets have:
  3 ..... --restrict_set=any_chains ..... any # of chain identifiers
  w ..... --restrict_set=includes_all_cids . at least one atom with each '-c' chain id
-f 6 ..... output to "_ww" file:
  6 ..... --filter=output_any_atom_type .... all ATOMs and HETATMs from matching sets
  r ..... --filter=output_only_cid ..... atoms restricted to any '-c' chain ids
-c 'KL' ..... --chainids=KL ..... chains 'K', 'L' for use with '-r' and '-f'
-z 3.3 ..... --max=3.3 ..... atoms must be within 3.3 A from each other
-s ..... --sets ..... output '_sets' file
-q 420 ..... --column_order=chain,aname,dist .. order '_sets' file:
  4 ..... chain ..... primarily by the 4th column (chain id)
  2 ..... anum ..... secondarily by the 2nd column (atom name)
  0 ..... dist ..... and lastly by the 0th column (distance)
```

## IMPLEMENTATION

A doubly linked list is searched to return all atoms with an exact distance range of another atom. This linked list is ordered in the Cartesian axis with the most extreme points (within an axis). One node is allocated for each atom. The doubly linked list is created with a single bucket sort.

Consider the following example data structure:

```
typedef struct point {          // data structure for range searching
  double x, y, z;              // coordinate values
  struct point *orig;          // singly linked list of points in original order
  struct point *sort;          // singly linked list for bucket sort collisions
  struct point *less, *more;    // doubly linked list for range searching
} POINT;

POINT **point_array;           // array of pointers for coordinate bucket sort
```

A separate singly linked list (orig) is set on reading the points.

To efficiently create the doubly linked list:

- (i) Translate the coordinates of all points to be positive.
- (ii) Allocate memory for a one dimensional array (point\_array) of node pointers, sized to the largest truncated dimension of the newly translated set of points.
- (iii) Set the pointers of the array to NULL.
- (iv) Fill the array with pointers to the POINT structs by using the truncated coordinates of the axis (with the largest truncated dimension) as an index. Create singly linked lists on collisions (using the pointer 'sort' in the point struct.)
- (iv) Order the sort array for each point array index as needed.
- (v) Read the point array to fill an ordered doubly linked list (e.g., less\_x and more\_x).

While some sorting (e.g., quicksort moving linked list pointers) will be required in step (iv) on the bucket sort collisions (i.e., when multiple nodes are assigned to the identical array index), no sorting is required in step (v) when moving between the array head pointers.

Creating a list of points having the property of being a specific distance range apart from each other consists of running two linear searches starting at each node in an axis-ordered doubly linked list. Each of these linear searches can stop once the maximum specified distance between points (in all dimensions) is exceeded in the dimension of the search of the current linked list.

Searching a single doubly-linked list (ordered in any axis) will provide a complete search. It is assumed that the axis with the furthest extreme points will produce a search with the fewest required node distance calculations for a specific range. (While this processing time is not necessarily shorter, it is more likely the case.)

The construction of the data structure for this algorithm has a worse case scenario of having all nodes indexing into a single point\_array index; the subsequent collision reduces the processing order of the data structure build time from  $O(N)$  to one dependent on the sorting algorithm used for collisions (e.g., for the quicksort used here, ' $O(2N \log(N))$ ' for a random permutation or ' $O(N^2)$ ' for a worse case ordered permutation). This worse case scenario would never be seen with input from a single normal PDB file as atoms in a single structure can not all share the same location. This might be possible if the input PDB file represents atoms from multiple PDB files; but more likely only a few atoms would share the same location.

This algorithm requires that most of the traditional calculation of a distance, the square root of the sums of the squares of differences of the points, be performed for every node examined in these linear searches; note that the final square root can be ignored and the sum of the squares can be compared.

This algorithm has a worse case search of having to examine all points to find one single point satisfying the range criteria; this will happen when all the points are spread out from the reference point from which the distance range is calculated in the doubly linked list, along one of the dimensional axes that are not the sorted axis of the doubly linked list being searched. This is minimized by having the axis of the doubly linked list be the axis with the most extreme points (in a single axis).

#### NOTES

Memory issues should not be seen with normal use of wwavePDB on modern computers for standard pdb files. wwavePDB will fail with an error if a memory problem is encountered.

This version of wwavePDB does not handle multiple "SPLIT" PDB files.

This version of wwavePDB does not perform parallel processing.

This version of wwavePDB handles (or rather mishandles) PDB files with alternate location indicators ("AltLoc" records) by ignoring, with a warning, any alternate atom or residue other than the first alternate atom or residue of each alternate set.

This version of wwavePDB does not handle non-standard "Chimera" PDB files with 6 byte atom serial numbers and 4 byte residue names.

As the defaults for the minimum and maximum ranges between atoms are zero and infinity, respectively, not specifying both the minimum and maximum range constraints results in a search that will return all the original atoms of the input file --not that useful unless other constraints are made. Further requesting the "\_sets" file for these range defaults will create a file for  $N$  distance sets with a total of  $N^2$  atoms (i.e., the distance ranges between every atom and every other atom): a potentially large file.

'wwave', a predecessor program of wwavePDB, identifies atoms in a PDB file within a "rough maximum distance" of other atoms. 'wwave' uses a different algorithm (the "Collected Grid Algorithm") for finding an imperfect range of atom distances; it uses the indexing of truncated coordinates to find both all atoms within a specific distance and possibly also some atoms of a larger distance (the square root of 3 larger). 'wwave' has a  $O(N)$  processing time --regardless of how distributed the search range. 'wwave' scales linearly with the number of atoms. However, unlike wwavePDB, 'wwave' does NOT identify individual sets of atoms, but instead identifies the superset of all atoms that match the distance requirements. While 'wwave' and wwavePDB have similar interfaces, 'wwave' does NOT have the option '--restrict\_set=' ('-r'). The set handling of wwavePDB is necessary for solving certain problems. If your problem can be handled by your hardware and wwavePDB, wwavePDB has exact range searching and set restrictions and filters that act upon sets based on individual atoms.

## LICENSE INFORMATION

wwavePDB is a software program from Arthur Weininger ([www.weiningerworks.com](http://www.weiningerworks.com)).  
wwavePDB is subject to a license; use the keyword option '--license' in order to view the license terms. Your use of this software constitutes an agreement to the license terms. Do not use this software if you do not agree to the license terms.

---

## wwavePDB Tutorial

**wwavePDB Tutorial Page** gives examples of using **wwavePDB** .

---

Arthur Weininger and Susan Weininger do not accept confidential information under any circumstance. Weininger Works™ is a trademark of Arthur Weininger and Susan Weininger. All images constitute content of this website and are copyrighted material of this website.

Copyright © 2013 – 2023 by Arthur Weininger and Susan Weininger. All Rights Reserved.