```
ATOM   3000  N   GLU A 476      22.858  22.742  68.556  1.00 18.28      N
ATOM   3001  CA  GLU A 476      23.317  24.142  68.341  1.00 19.70      C
ATOM   3002  C   GLU A 476      23.448  24.933  69.624  1.00 19.59      C
ATOM   3003  O   GLU A 476      23.266  24.315  70.692  1.00 22.22      O
ATOM   3004  CB  GLU A 476      22.365  24.792  67.378  1.00 19.67      C
ATOM   3005  CG  GLU A 476      22.258  24.022  66.051  1.00 24.41      C
ATOM   3006  CD  GLU A 476      21.290  24.685  65.070  1.00 29.08      C
ATOM   3007  OE1 GLU A 476      20.739  25.795  65.399  1.00 26.35      O
ATOM   3008  OE2 GLU A 476      21.098  24.083  63.986  1.00 25.46      O
TER    3009      GLU A 476
ATOM   3010  N   ARG B1088      18.074 -27.415  76.667  1.00 36.00      N
ATOM   3011  CA  ARG B1088      17.910 -28.872  76.893  1.00 35.83      C
ATOM   3012  C   ARG B1088      16.621 -29.427  76.231  1.00 34.86      C
ATOM   3013  O   ARG B1088      16.738 -30.301  75.374  1.00 34.88      O
ATOM   3014  CB  ARG B1088      18.027 -29.222  78.408  1.00 37.05      C
ATOM   3015  CG  ARG B1088      17.542 -30.704  78.866  1.00 41.16      C
ATOM   3016  CD  ARG B1088      18.655 -31.754  79.141  1.00 50.40      C
ATOM   3017  NE  ARG B1088      19.542 -31.812  77.975  1.00 55.88      N
ATOM   3018  CZ  ARG B1088      20.610 -31.016  77.814  1.00 61.76      C
ATOM   3019  NH1 ARG B1088      20.940 -30.126  78.771  1.00 61.49      N
ATOM   3020  NH2 ARG B1088      21.350 -31.097  76.703  1.00 63.30      N
ATOM   3021  N   THR B1089      15.419 -28.921  76.597  1.00 32.94      N
ATOM   3022  CA  THR B1089      14.115 -29.495  76.160  1.00 30.55      C
ATOM   3023  C   THR B1089      12.918 -28.505  75.897  1.00 28.62      C
ATOM   3024  O   THR B1089      12.890 -27.342  76.457  1.00 27.19      O
ATOM   3025  CB  THR B1089      13.703 -30.586  77.176  1.00 32.46      C
```

stripPDB Overview

STRIPPDB

stripPDB Overview

stripPDB is a PDB filter to strip specified atoms.

stripPDB Help Output (“stripPDB -h” output)

```
NAME

  stripPDB (version 1.0.2) -- a PDB filter to strip specified atoms

SYNOPSIS

  stripPDB [options]

  CHARACTER OPTION____KEYWORD OPTION________DESCRIPTION_____________________DEFAULT____
  -i <filename> ... --input=<filename> .... input PDB file ................... stdin
  -o <filename> ... --output=<filename> ... output PDB file .................. stdout
  -m <model#> ..... --model=<model#> ...... MODEL number of input PDB file ... first model
  -r <remove_str> . --remove=<remove_str> . remove atoms w/specific chain IDs. no action
     <remove_str> is either: <chain_ids>      (simple list of chain IDs w/no unquoted ';' or ',')
                         or: [[c],[r],[a];]+  (triplet list of chain ID, residue #, atom #)
  -t .............. --hetatms ............. remove all HETATMs ............... no action
  -w .............. --water ............... remove all HETATM water atoms .... no action
  -y .............. --hydrogen ............ remove all HETATM hydrogen atoms . no action
  -h .............. --help ................ prints help (Enter 'stripPDB -h' for more help!)
  <NO OPTIONS> ............................ shorter option synopsis (Enter 'stripPDB'.)
                   --license ............. prints license terms for stripPDB.

DESCRIPTION

  stripPDB removes specified atoms from an input PDB file and writes a new output PDB file.

  stripPDB will process one (possibly implied) MODEL from the input PDB file. The output
  PDB file will contain: (i) the header records from the input file, (ii) additional REMARK
  header records summarizing stripPDB execution, (iii) all the records from a single specific
  MODEL with the exception of any ATOM, HETATM, and TER records that match removal criteria,
  (iv) any CONECT records that reference output records, and (v) a final END record.
  Records of any other existing MODEL sections, and their associated CONECT records,
  will be removed. CONECT records will be updated to reflect removed records.

  The input file is expected to be in PDB format. Option '--input=' ('-i') specifies the
  input filename. If option '--input=' ('-i') is not present, input will be read from stdin.

  By default, the first MODEL (perhaps implicit if no MODEL records exist) from each
  input file is processed. Option '--model=' ('-m') may be used to specify the MODEL
  number of the input file to process.

  Option '--output=' ('-o') specifies the output filename. If option '--output=' ('-o')
  is not present, output will be written to stdout.

  Option '--hetatms' ('-t)' specifies that all HETATM records are to be removed.

  Option '--water' ('-w') specifies that all HETATM records that are also water atoms
  (hydrogen or oxygen) are to be removed. Water atoms are defined as HETATM records
  that have a Residue Name value of "HOH".

  Option '--hydrogen' ('-y') specifies that all HETATM records that are also hydrogen
  atoms (regardless of whether they are part of water) are to be removed.

  If the option '--remove=' ('-r') is present, then any ATOM, HETATM, and TER record
  having content that matches the target specification of the '--remove=' ('-r') option
  value will be removed. The '--remove=' ('-r') option value is either a simple list of
  chain identifiers ("Simple List Format") or it is a list of triplets that allow chain
  identifiers, residue sequence numbers, and atom serial numbers to be used to specify
  matching record targets ("Triplet List Format").

  Simple List Format

  The Simple List Format of the '--remove=' ('-r') option value is a character string of
  chain identifiers. This character string can be any size and can contain any non-NULL
  character. If this character string starts and ends with either a single quote (''')
  or a double quote ('"'), then these quotes will be treated as string delimiters and not
  as chain indentifiers. When quotes are intended to be used as chain indentifiers, then
  the quote characters must be preceded with a single backslash character ('\'). This
  character string may not contain semicolons (';') or commas (',') unless the semicolons
  and commas are preceeded with a backslash (e.g., '\;'); semicolons and commas not quoted
  by a backslash will cause the '--remove=' ('-r') option value to be interpreted as a
  Triplet List Format. Two consecutive backslashes ('\\') are used when the backslash
  character ('\') is intended to be used as a chain identifier (i.e., the backslash
  character that is the chain identifier gets quoted with a backslash).
```

Examples of the Simple List Format follow:

Terse option	Keyword option	Description
-r ABC	--remove=ABC	chain IDs 'A','B','C'
-r 'ABC XYZ'	--remove='ABC XYZ'	chain IDs 'A','B','C',' ','X','Y','Z'
-r "ABC AAA"	--remove="ABC AAA"	chain IDs 'A','B','C',' ','A','A','A'

Note that a single quote (') or a double quote (") may be used to delimit an option value. The '--remove=' ('-r') option values shown above are quoted in order to specify that a space (' ')be used as a character identifier.

Note that use of the hyphen character ('-') in the Simple List Format will be interpreted as the value of a chain identifier. See “Triplet List Format” below for range specifications.

Triplet List Format

The Triplet List Format of the '--remove=' ('-r') option value allows records to be specified for removal by presence of chain identifier, residue sequence number, and atom serial number. This triplet list is defined as a semicolon-delimited list of triplets, where each triplet consists of three comma-delimited fields:

chain identifier, residue sequence number, atom serial number

Any of the '--remove=' ('-r') triplet fields may optionally have a range specification in one of the following formats:

Format	Examples	Description
<lower_limit> - <upper_limit>	A-Z 107-200	Matches any value in between and including lower limit and upper limit.
- <upper_limit>	-Z -200	Matches any value lower than and including upper limit.
<lower_limit> -	A- 107-	Matches any value higher than and including lower limit.

Any of the '--remove=' ('-r') triplet fields may have a null value (i.e., an empty field). Null-valued triplet fields will match any record value of that field type. A PDB record has to match all non-null fields in a triplet in order for record removal.

Comma delimiters need only be supplied as needed to specify a field index. Semicolon delimiters are required with the exception of the last semicolon, which is optional.

Examples of the triplet list format follow:

Terse option	Keyword option	Description
-r 'A;B;C;'	--remove='A;B;C;'	chain IDs (any residue # and any atom #)
-r 'A;B;C'	--remove='A;B;C'	chain IDs (any residue # and any atom #)
-r 'A;B;C; ;X;Y;Z'	--remove='A;B;C; ;X;Y;Z'	chain IDs (any residue # and any atom #)
-r "A;B;C; ;A;A;A"	--remove="A;B;C; ;A;A;A"	chain IDs (any residue # and any atom #)
-r "A,,;"	--remove="A,,;"	chain ID 'A' (any residue # and any atom #)
-r "A,35,;"	--remove="A,35,;"	residue 35 of chain ID 'A' (with any atom #)
-r "A,35,293;"	--remove="A,35,293;"	atom 293 of residue 35 of chain ID 'A'
-r ",35,;"	--remove=",35,;"	residue 35 (any chain ID and any atom #)
-r ",,293;"	--remove=",,293;"	atom 293 (any chain ID and any residue #)
-r "A,,;,35,;,,293;"	--remove="A,,;,35,;,,293;"	three triplets with complete delimiters
-r "A;,35;,,293"	--remove="A;,35;,,293"	three triplets with necessary delimiters
-r ",-78;,110-;"	--remove=",-78;,110-;"	all residue #s except 79 through 109

Notes for both Simple List Format and Triplet List Format

The '--remove=' ('-r') target specification consists of single characters (representing chain identifiers) for the Simple List Format and triplets (providing more flexible specifications) for the Triplet List Format.

Chain identifiers, specified in '--remove=' ('-r') option values, can be any non-NULL character. If semicolons (';'), commas (','), backslashes ('\'), or option value quotes (''' or '"') are used as chain identifier values in any '--remove=' ('-r') option value, then these characters need to be preceded with two consecutive backslashes ('\\').

As the unix shell interprets semicolons (';') as command line separators, command line option values containing semicolons (';') are best handled by quoting the entire option value with either single quotes (') or double quotes (").

New "REMARK 250" lines will be added to the output PDB specifying execution details.

By default, no PDB records are altered or removed. Options '--hetatms' ('-t)', '--water' ('-w'), '--hydrogen' ('-y'), or '--remove=' ('-r') all independently specify PDB records to be removed (i.e., any record specified to be removed by any of these options will be removed regardless of the use or value of any other option.) If no removal options are provided, then stripPDB will make a copy of the entire PDB (including all MODELS).

Any errors and warnings will be written to stderr.

EXAMPLE

The following will remove all HETATM records (and any associated CONECT records) from 'in.pdb'. The file 'out.pdb' will be created.

With keyword options:

```
stripPDB --input=in.pdb --output=out.pdb --hetatms
```

With character options:

```
stripPDB -i in.pdb -o out.pdb -t
```

The following will remove all records (and any associated CONECT records) from 'in.pdb' that contain either 'P' or 'Q' as a chain identfier. The file 'out.pdb' will be created.

With keyword options:

```
stripPDB --input=in.pdb --output=out.pdb --remove=PQ
```

With character options:

```
stripPDB -i in.pdb -o out.pdb -r PQ
```

The following will remove all records (and any associated CONECT records) from 'in.pdb' that contain either 'P' or 'Q' or space (' ') as a chain identfier. The file 'out.pdb' will be created.

With keyword options:

```
stripPDB --input=in.pdb --output=out.pdb --remove="PQ "
```

With character options:

```
stripPDB -i in.pdb -o out.pdb -r "PQ "
```

With keyword options in a Triplet List Format:

```
stripPDB --input=in.pdb --output=out.pdb --remove="P;Q; ;"
```

With character options in a Triplet List Format:

```
stripPDB -i in.pdb -o out.pdb -r "P;Q; ;"
```

The following will remove all records (and any associated CONECT records) from 'in.pdb' that contain both the chain identifier 'A' and the residue number 36.

With keyword options:

```
stripPDB --input=in.pdb --output=out.pdb --remove="A,36"
```

With character options:

```
stripPDB -i in.pdb -o out.pdb -r "A,36"
```

The following will remove all records (and any associated CONECT records) from 'in.pdb' that contain the residue sequence number 35 (and any chain identifier).

With keyword options:

```
stripPDB --input=in.pdb --output=out.pdb --remove=",35"
```

With character options:

```
stripPDB -i in.pdb -o out.pdb -r ",35"
```

The following will remove all records (and any associated CONECT records) from 'in.pdb' having the atom serial number 293 (regardless of chain identifier or residue number).

With keyword options:

```
stripPDB --input=in.pdb --output=out.pdb --remove=",,293"
```

With character options:

```
stripPDB -i in.pdb -o out.pdb -r ",,293"
```

```
  The following will remove only the record(s) (and any associated CONECT records)
  from 'in.pdb' that contains all of: chain identifier 'A', residue sequence number 35,
  and atom serial number 293.

     With keyword options:

        stripPDB --input=in.pdb --output=out.pdb --remove="A,35,293"

     With character options:

        stripPDB -i in.pdb -o out.pdb -r "A,35,293"

  The following will remove all the ATOM, HETATM, or TER records having:
    chain identifier A, or
    chain identifier B, or
    both chain identifier C and a residue sequence number >= 1 and <= 49, or
    both chain identifier C and a residue sequence number >= 63, or
    chain identifier D.

  With keyword options:

    stripPDB --input=1BBT.pdb --output=temp1.pdb --from="A;B;C,1-49;C,63-;D;"

  With character options:

    stripPDB -i 1BBT.pdb -o temp1.pdb -f "A;B;C,1-49;C,63-;D;"

LICENSE INFORMATION

  stripPDB is a software program from Arthur Weininger (www.weiningerworks.com).
  stripPDB is subject to a license; use the keyword option '--license' in order to view
  the license terms. Your use of this software contitutes an agreement to the license
  terms. Do not use this software if you do not agree to the license terms.
```

---

stripPDB Tutorial

The **Picornavirus Monograph Superposition Shell Script** gives examples of using **stripPDB**.
