Installation Instructions:

Pdb Tools can be installed from the github link:

https://github.com/haddocking/pdb-tools/

1.) Press the “clone or download” button to download the .zip file containing the toolset
2.) Drag the zip file somewhere useful (such as your desktop)
3.) Unzip the file using 7zip (can be installed on the windows store)
4.) If you already have python and pip installed use the following command to install pdb-tools:

pip install pdb-tools

5.) After this, you can start using these commands in any directory, but keep in mind, if you specify a new file to save by using “>” then it will save in whatever directory you are currently in

Most Useful Commands:

pdb_fetch

Downloads a structure in PDB format from the RCSB website. Allows downloading the (first) biological structure if selected.

Usage:
pdb_fetch [-biounit] <pdb code> > <newfilename.pdb>

Example:
pdb_fetch 1brs > 1brs.pdb # downloads unit cell, all 6 chains
pdb_fetch -biounit 1brs > 1brs_biounit.pdb # downloads biounit, 2 chains

pdb_gap

Detects gaps between consecutive residues in the sequence, both by a distance criterion or discontinuous residue numbering. Only applies for protein residues.

Usage:
pdb_gap <pdb file>

Example:
pdb_gap 1CTF.pdb

??pdb_mkensemble

Merges several PDB files into one multi-model (ensemble) file. Strips all

HEADER information and adds REMARK statements with the provenance of each conformer.

Usage:
pdb_mkensemble <pdb file> <pdb file>

Example:
pdb_mkensemble 1ABC.pdb 1XYZ.pdb

??pdb_chkensemble
Performs a basic check on a multi-model PDB file (ensemble) to ensure all models have exactly the same atoms (as they should).

Usage:
pdb_chkensemble <pdb file>

Example:
pdb_chkensemble 1CTF.pdb

pdb_delchain
Deletes all atoms matching specific chains in the PDB file.

Usage:
pdb_delchain -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_delchain -A 1CTF.pdb > 1CTF_noA.pdb # removes chain A from PDB file
pdb_delchain -A,B 1CTF.pdb > 1CTF_noAB.pdb # removes chains A and B from PDB file

pdb_delelem
Deletes all atoms matching the given element in the PDB file. Elements are read from the element column.

Usage:
pdb_delelem -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_delelem -H 1CTF.pdb > 1CTF_nohydro.pdb # deletes all protons
pdb_delatom -N 1CTF.pdb > 1CTF_nonitro.pdb # deletes all nitrogens
pdb_delatom -H,N 1CTF.pdb > 1CTF_noHorN.pdb # deletes all protons and nitrogens

pdb_delhetatm
Removes all HETATM records in the PDB file.

Usage:
pdb_delhetatm <pdb file> > <newfilename.pdb>

Example:
pdb_delhetatm 1CTF.pdb > 1CTF_noligand.pdb

pdb_delres
Deletes a range of residues from a PDB file. The range option has three components: start, end, and step. Start and end are optional and if ommitted the range will start at the first residue or end at the last, respectively. The step option can only be used if both start and end are provided. Note that the start and end values of the range are purely numerical, while the range actually refers to every N-th residue, regardless of their sequence number.

Usage:
pdb_delres.py -[resid]:[resid]:[step] <pdb file> > <newfilename.pdb

Example:
pdb_delres -1:10 1CTF.pdb > 1CTF_lessres.pdb # Deletes residues 1 to 10
pdb_delres -1: 1CTF.pdb > 1CTF_lessres.pdb # Deletes residues 1 to END
pdb_delres -:5 1CTF.pdb > 1CTF_lessres.pdb # Deletes residues from START to 5.
pdb_delres -::5 1CTF.pdb > 1CTF_lessres.pdb # Deletes every 5th residue
pdb_delres -1:10:5 1CTF.pdb > 1CTF_lessres.pdb # Deletes every 5th residue from 1 to 10

pdb_selatom
Selects all atoms matching the given name in the PDB file. Atom names are matched *without* taking into consideration spaces, so ' CA ' (alpha carbon) and 'CA ' (calcium) will both be kept if -CA is passed.

Usage:
pdb_selatom -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_selatom -CA 1CTF.pdb > 1CTF_CAonly.pdb # keeps only alpha-carbon atoms
pdb_selatom -CA,C,N,O 1CTF.pdb > 1CTF_backbone.pdb # keeps only backbone atoms

pdb_selchain
Extracts one or more chains from a PDB file.

Usage:

pdb_selchain -<chain id> <pdb file> > <newfilename.pdb>

Example:
pdb_selchain -C 1CTF.pdb > 1CTF_C.pdb # selects chain C
pdb_selchain -A,C 1CTF.pdb > 1CTF_AC.pdb # selects chains A and C

pdb_selelem
Selects all atoms that match the given element(s) in the PDB file. Elements are read from the element column.

Usage:
pdb_selelem -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_selelem -H 1CTF.pdb > 1CTF_protons.pdb # selects all protons
pdb_selelem -N 1CTF.pdb > 1CTF_N.pdb # selects all nitrogens
pdb_selelem -H,N 1CTF.pdb > 1CTF_HN.pdb # selects all protons and nitrogens

pdb_selres
Extracts a range of residues from a PDB file. The range option has three components: start, end, and step. Start and end are optional and if ommitted the range will start at the first residue or end at the last, respectively. The step option can only be used if both start and end are provided. Note that the start and end values of the range are purely numerical, while the range actually refers to every N-th residue, regardless of their sequence number.

Usage:
pdb_selres -[resid]:[resid]:[step] <pdb file> > <newfilename.pdb>

Example:
pdb_selres -1:10 1CTF.pdb > 1CTF_edit.pdb # Extracts residues 1 to 10
pdb_selres -1: 1CTF.pdb > 1CTF_edit.pdb # Extracts residues 1 to END
pdb_selres -:5 1CTF.pdb > 1CTF_edit.pdb # Extracts residues from START to 5.
pdb_selres -::5 1CTF.pdb > 1CTF_edit.pdb # Extracts every 5th residue
pdb_selres -1:10:5 1CTF.pdb > 1CTF_edit.pdb # Extracts every 5th residue from 1 to 10

pdb_sort
Sorts the ATOM/HETATM/ANISOU/CONECT records in a PDB file.

Atoms are always sorted by their serial number, meaning the original ordering of the atoms within each residue are not changed. Alternate locations are sorted

by default.

Residues are sorted according to their residue sequence number and then by their insertion code (if any).

Chains are sorted by their chain identifier.

Finally, the file is sorted by all keys, and the records are placed in the following order:

- ATOM/ANISOU, intercalated if the latter exist
- HETATM
- CONECT, sorted by the serial number of the central (first) atom

MASTER, TER, END statements are removed. Headers (HEADER, REMARK, etc) are kept and placed first. Does NOT support multi-model files. Use pdb_splitmodel, then pdb_sort on each model, and then pdb_mkensemble.

Usage:
pdb_sort -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_sort 1CTF.pdb > 1CTF_sort # sorts by chain and residues
pdb_sort -C 1CTF.pdb > 1CTF_sort # sorts by chain (A, B, C ...) only
pdb_sort -R 1CTF.pdb > 1CTF_sort # sorts by residue number/icode only

??pdb_splitchain
Splits a PDB file into several, each containing one chain.

Usage:
pdb_splitchain <pdb file>

Example:
pdb_splitchain 1CTF.pdb

??pdb_splitmodel
Splits a PDB file into several, each containing one MODEL.

Usage:
pdb_splitmodel <pdb file>

Example:
pdb_splitmodel 1CTF.pdb

??pdb_splitseg
Splits a PDB file into several, each containing one segment.

Usage:
pdb_splitseg <pdb file>

Example:
pdb_splitseg 1CTF.pdb

pdb_tidy
Modifies the file to adhere (as much as possible) to the format specifications.
Expects a sorted file - REMARK/ATOM/HETATM/END - so use pdb_sort in case you are not sure.

This includes:
- Adding TER statements after chain breaks/changes
- Truncating/Padding all lines to 80 characters
- Adds END statement at the end of the file

Will remove all original TER/END statements from the file.

Usage:
pdb_tidy <pdb file> > <newfilename.pdb>

Example:
pdb_tidy 1CTF.pdb > 1CTF_tidy.pdb

pdb_validate
Validates the PDB file ATOM/HETATM lines according to the format specifications.
Does not catch all the errors though... people are creative!

Usage:
pdb_format <pdb file>

Example:
pdb_format 1CTF.pdb

pdb_wc
Summarizes the contents of a PDB file, like the wc command in UNIX.

Several options are available to produce only partial summaries:
- [m] - no. of models.
- [c] - no. of chains (plus per-model if multi-model file).
- [r] - no. of residues (plus per-model if multi-model file).
- [a] - no. of atoms (plus per-model if multi-model file).
- [h] - no. of HETATM (plus per-model if multi-model file).
- [o] - no. of disordered atoms (altloc) (plus per-model if multi-model file).
- [i] - no. of insertion codes (plus per-model if multi-model file).
- [g] - presence/absence of gaps (discontinuous residue numbering).

Usage:
pdb_wc [-<option>] <pdb file>

Example:
pdb_wc 1CTF.pdb

Master List:

pdb_b
Modifies the temperature factor column of a PDB file (default 10.0).

Usage:
pdb_b -<bfactor> <pdb file> > <newfilename.pdb>

Example:
pdb_b -10.0 1CTF.pdb > 1CTF_difftemp.pdb

pdb_chain
Modifies the chain identifier column of a PDB file (default is an empty chain).

Usage:
pdb_chain -<chain id> <pdb file> > <newfilename.pdb>

Example:
pdb_chain -C 1CTF.pdb > 1CTF_mod.pdb

pdb_chainxseg
Replaces the segment identifier column by the value in the chain identifier column of the PDB file.

Usage:
pdb_chainxseg <pdb file> > <newfilename.pdb>

Example:
pdb_chainxseg 1CTF.pdb > 1CTF_mod.pdb

pdb_chkensemble
Performs a basic check on a multi-model PDB file (ensemble) to ensure all models have exactly the same atoms (as they should).

Usage:
pdb_chkensemble <pdb file>

Example:
pdb_chkensemble 1CTF.pdb

pdb_delchain
Deletes all atoms matching specific chains in the PDB file.

Usage:
pdb_delchain -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_delchain -A 1CTF.pdb > 1CTF_noA.pdb # removes chain A from PDB file
pdb_delchain -A,B 1CTF.pdb > 1CTF_noAB.pdb # removes chains A and B from PDB file

pdb_delelem

Deletes all atoms matching the given element in the PDB file. Elements are read from the element column.

Usage:
pdb_delelem -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_delelem -H 1CTF.pdb > 1CTF_nohydro.pdb # deletes all protons
pdb_delatom -N 1CTF.pdb > 1CTF_noN.pdb # deletes all nitrogens
pdb_delatom -H,N 1CTF.pdb > 1CTF_noNH.pdb # deletes all protons and nitrogens

pdb_delhetatm
Removes all HETATM records in the PDB file.

Usage:
pdb_delhetatm <pdb file> > <newname.pdb>

Example:
pdb_delhetatm 1CTF.pdb > 1CTF_nolig.pdb

pdb_delres
Deletes a range of residues from a PDB file. The range option has three components: start, end, and step. Start and end are optional and if ommitted the range will start at the first residue or end at the last, respectively. The step option can only be used if both start and end are provided. Note that the start and end values of the range are purely numerical, while the range actually refers to every N-th residue, regardless of their sequence number.

Usage:
pdb_delres -[resid]:[resid]:[step] <pdb file> > <newfilename.pdb>

Example:

pdb_delres -1:10 1CTF.pdb > 1CTF_lessres.pdb # Deletes residues 1 to 10
pdb_delres -1: 1CTF.pdb > 1CTF_lessres.pdb # Deletes residues 1 to END
pdb_delres -:5 1CTF.pdb > 1CTF_lessres.pdb # Deletes residues from START to 5.
pdb_delres -::5 1CTF.pdb > 1CTF_lessres.pdb # Deletes every 5th residue
pdb_delres -1:10:5 1CTF.pdb > 1CTF_lessres.pdb # Deletes every 5th residue from 1 to 10

pdb_element
Assigns the element column to the PDB file, guessing the element from the atom names.

Usage:
pdb_element <pdb file>

Example:
pdb_element 1CTF.pdb

pdb_fetch
Downloads a structure in PDB format from the RCSB website. Allows downloading the (first) biological structure if selected.

Usage:
pdb_fetch [-biounit] <pdb code> > <pdbfilename.pdb>

Example:
pdb_fetch 1brs > 1brs.pdb # downloads unit cell, all 6 chains
pdb_fetch -biounit 1brs > 1brs_biounit.pdb # downloads biounit, 2 chains

pdb_fromcif
Converts a mmCIF file to the PDB format. Will not convert if the file does not 'fit' in PDB format, e.g. too many chains, residues, or atoms. Will convert only the coordinate section.

Usage:
pdb_fromcif <pdb file>

Example:
pdb_fromcif 1CTF.pdb

pdb_gap
Detects gaps between consecutive residues in the sequence, both by a distance

criterion or discontinuous residue numbering. Only applies for protein residues.

Usage:
pdb_gap <pdb file>

Example:
pdb_gap 1CTF.pdb

??pdb_intersect
Atoms are judged equal is their name, altloc, res. name, res. num, insertion code and chain fields are the same. Coordinates are taken from the first input file. Keeps matching TER/ANISOU records.

Usage:
pdb_intersect <pdb file> <pdb file>

Example:
pdb_intersect 1XYZ.pdb 1ABC.pdb

pdb_keepcoord
Removes all non-coordinate records from the file.
Keeps only MODEL, ENDMDL, END, ATOM, HETATM, and CONECT.

Usage:
pdb_keepcoord <pdb file>

Example:
pdb_keepcoord 1CTF.pdb

??pdb_merge
Merges several PDB files into one. The contents are not sorted and no lines are deleted (e.g. END, TER statements) so we recommend piping the results through `pdb_tidy.py`.

Usage:
pdb_merge <pdb file> <pdb file>

Example:
pdb_merge 1ABC.pdb 1XYZ.pdb

??pdb_mkensemble
Merges several PDB files into one multi-model (ensemble) file. Strips all

HEADER information and adds REMARK statements with the provenance of each conformer.

Usage:
pdb_mkensemble <pdb file> <pdb file>

Example:
pdb_mkensemble 1ABC.pdb 1XYZ.pdb

pdb_occ
Modifies the occupancy column of a PDB file (default 1.0).

Usage:
pdb_occ -<occupancy> <pdb file>

Example:
pdb_occ -1.0 1CTF.pdb

pdb_reatom
Renumbers atom serial numbers of the PDB file starting from a given value (default 1).

Usage:
pdb_reatom -<number> <pdb file>

Example:
pdb_reatom -10 1CTF.pdb # renumbers from 10
pdb_reatom --1 1CTF.pdb # renumbers from -1

pdb_reres
Renumbers the residues of the PDB file starting from a given number (default 1).

Usage:
pdb_reres -<number> <pdb file>

Example:
pdb_reres -10 1CTF.pdb # renumbers from 10
pdb_reres --1 1CTF.pdb # renumbers from -1

pdb_rplchain
Performs in-place replacement of a chain identifier by another.

Usage:
pdb_rplchain -<from>:<to> <pdb file>

Example:
pdb_rplchain -A:B 1CTF.pdb # Replaces chain A for chain B

pdb_rplresname
Performs in-place replacement of a residue name by another. Affects all residues with that name.

Usage:
pdb_rplres -<from>:<to> <pdb file>

Example:
pdb_rplres -HIP:HIS 1CTF.pdb # changes all HIP residues to HIS

pdb_seg
Modifies the segment identifier column of a PDB file (default is an empty segment).

Usage:
pdb_seg -<segment id> <pdb file>

Example:
pdb_seg -C 1CTF.pdb

pdb_segxchain
Replaces the chain identifier column by the value in the segment identifier column of the PDB file. Truncates the segment identifier if longer than one character.

Usage:
pdb_segxchain <pdb file>

Example:
pdb_segxchain 1CTF.pdb

pdb_selaltloc
Picks one location for each atom with fractional occupancy values. By default, picks the atom with the highest occupancy value. User can define one specific location using an option.

Usage:
pdb_selaltloc [-<option>] <pdb file>

Example:
pdb_selaltloc 1CTF.pdb # picks locations with highest occupancy
pdb_selaltloc -A 1CTF.pdb # picks alternate locations labelled 'A'

pdb_selatom
Selects all atoms matching the given name in the PDB file. Atom names are matched *without* taking into consideration spaces, so ' CA ' (alpha carbon) and 'CA ' (calcium) will both be kept if -CA is passed.

Usage:
pdb_selatom -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_selatom -CA 1CTF.pdb > 1CTF_CAonly.pdb # keeps only alpha-carbon atoms
pdb_selatom -CA,C,N,O 1CTF.pdb > 1CTF_backbone.pdb # keeps only backbone atoms

pdb_selchain
Extracts one or more chains from a PDB file.

Usage:
pdb_selchain -<chain id> <pdb file> > <newfilename.pdb>

Example:
pdb_selchain -C 1CTF.pdb > 1CTF_C.pdb # selects chain C
pdb_selchain -A,C 1CTF.pdb > 1CTF_AC.pdb # selects chains A and C

pdb_selelem
Selects all atoms that match the given element(s) in the PDB file. Elements are read from the element column.

Usage:
pdb_selelem -<option> <pdb file> > <newfilename.pdb>

Example:
pdb_selelem -H 1CTF.pdb > 1CTF_protons.pdb # selects all protons
pdb_selelem -N 1CTF.pdb > 1CTF_N.pdb # selects all nitrogens
pdb_selelem -H,N 1CTF.pdb > 1CTF_HN.pdb # selects all protons and nitrogens

pdb_selres
Extracts a range of residues from a PDB file. The range option has three components: start, end, and step. Start and end are optional and if ommitted the range will start at the first residue or end at the last, respectively. The step option can only be used if both start and end are provided. Note that the start and end values of the range are purely numerical, while the range actually refers to every N-th residue, regardless of their sequence number.

Usage:
pdb_selres -[resid]:[resid]:[step] <pdb file> > <newfilename.pdb>

Example:
pdb_selres -1:10 1CTF.pdb > 1CTF_edit.pdb # Extracts residues 1 to 10
pdb_selres -1: 1CTF.pdb > 1CTF_edit.pdb # Extracts residues 1 to END
pdb_selres -:5 1CTF.pdb > 1CTF_edit.pdb # Extracts residues from START to 5.
pdb_selres -::5 1CTF.pdb > 1CTF_edit.pdb # Extracts every 5th residue
pdb_selres -1:10:5 1CTF.pdb > 1CTF_edit.pdb # Extracts every 5th residue from 1 to 10

pdb_selseg
Extracts one or more segments from a PDB file based on their segment identifiers.

Usage:
pdb_selseg -<segment id> <pdb file>

Example:
pdb_selseg -C 1CTF.pdb # selects segment C
pdb_selseg -C,D 1CTF.pdb # selects segments C and D

pdb_shiftres
Renumbers the residues of the PDB file by adding/subtracting a given number from the original numbering.

Usage:
pdb_shiftres -<number> <pdb file>

Example:
pdb_shiftres -10 1CTF.pdb # adds 10 to the original numbering
pdb_shiftres --5 1CTF.pdb # subtracts 5 from the original numbering

pdb_sort
Sorts the ATOM/HETATM/ANISOU/CONECT records in a PDB file.

Atoms are always sorted by their serial number, meaning the original ordering of the atoms within each residue are not changed. Alternate locations are sorted by default.

Residues are sorted according to their residue sequence number and then by their insertion code (if any).

Chains are sorted by their chain identifier.

Finally, the file is sorted by all keys, and the records are placed in the following order:
- ATOM/ANISOU, intercalated if the latter exist
- HETATM
- CONECT, sorted by the serial number of the central (first) atom

MASTER, TER, END statements are removed. Headers (HEADER, REMARK, etc) are kept

and placed first. Does NOT support multi-model files. Use pdb_splitmodel, then pdb_sort on each model, and then pdb_mkensemble.

Usage:
pdb_sort -<option> <pdb file>

Example:
pdb_sort 1CTF.pdb # sorts by chain and residues
pdb_sort -C 1CTF.pdb # sorts by chain (A, B, C ...) only
pdb_sort -R 1CTF.pdb # sorts by residue number/icode only

pdb_splitchain
Splits a PDB file into several, each containing one chain.

Usage:
pdb_splitchain <pdb file>

Example:
pdb_splitchain 1CTF.pdb

pdb_splitmodel
Splits a PDB file into several, each containing one MODEL.

Usage:
pdb_splitmodel <pdb file>

Example:
pdb_splitmodel 1CTF.pdb

pdb_splitseg
Splits a PDB file into several, each containing one segment.

Usage:
pdb_splitseg <pdb file>

Example:
pdb_splitseg 1CTF.pdb

pdb_tidy
Modifies the file to adhere (as much as possible) to the format specifications.

Expects a sorted file - REMARK/ATOM/HETATM/END - so use pdb_sort in case you are not sure.

This includes:
- Adding TER statements after chain breaks/changes
- Truncating/Padding all lines to 80 characters
- Adds END statement at the end of the file

Will remove all original TER/END statements from the file.

Usage:
pdb_tidy <pdb file>

Example:
pdb_tidy 1CTF.pdb

pdb_tocif
Converts a PDB file to the mmCIF format. Will convert only the coordinate section.

Usage:
pdb_tocif <pdb file>

Example:
pdb_tocif 1CTF.pdb

pdb_tofasta

Extracts the residue sequence in a PDB file to FASTA format. Canonical amino acids and nucleotides are represented by their one-letter code while all others are represented by 'X'.

The -multi option splits the different chains into different records in the FASTA file.

Usage:
pdb_toseq [-multi] <pdb file>

Example:
pdb_toseq 1CTF.pdb

pdb_validate
Validates the PDB file ATOM/HETATM lines according to the format specifications.
Does not catch all the errors though... people are creative!

Usage:
pdb_format <pdb file>

Example:
pdb_format 1CTF.pdb

pdb_wc
Summarizes the contents of a PDB file, like the wc command in UNIX.

Several options are available to produce only partial summaries:
- [m] - no. of models.
- [c] - no. of chains (plus per-model if multi-model file).
- [r] - no. of residues (plus per-model if multi-model file).
- [a] - no. of atoms (plus per-model if multi-model file).
- [h] - no. of HETATM (plus per-model if multi-model file).
- [o] - no. of disordered atoms (altloc) (plus per-model if multi-model file).
- [i] - no. of insertion codes (plus per-model if multi-model file).
- [g] - presence/absence of gaps (discontinuous residue numbering).

Usage:
pdb_wc [-<option>] <pdb file>

Example:
pdb_wc 1CTF.pdb