# MCCE Tools

List of miscellaneous tools MCCE offers for your research and convenience. Most of these tools support "-h" flags for additional information and use cases.

Some of these tools are intended for *pre-run* analysis, and some are intended for **post-run** analysis. Pre-run tools will be *italicized*, and are found in the MCCE_bin of a MCCE4-Alpha directory. Post-run tools will be **bolded**, and are found in MCCE4-Tools, a separate git directory.

## cif2pdb (MCCE4-Tools)

usage: cif_to_pdb file.cif [file.pdb]

Converts a .cif file to .pdb format.

## clear_mcce_folder (MCCE4-Tools)

Deletes all MCCE outputs from the present working directory, except: run.prm, the original PDB file, prot.pdb, and any non-MCCE files.

## detect_hbonds.py

Detect H-bonds in a PDB file, with the option to include BK (backbone) atoms.
hbonds_pdb_collection uses this function on a collection of PDB files.

usage: detect_hbonds.py [-h] [--include_bk] [--no_empty_files] [--out_dir OUT_DIR] [inpdb]

## extract_md_frames

Extracts the trajectory's frames with the given indices into PDB files. Requires the MDAnalysis package.

## filesdiff

Obtain the column difference between two MCCE files or the differences of all files in two MCCE output folders. Use the "-threshold" flag to output absolute differences beyond a given value (0 is

default).

Applicable to the following MCCE files: 'all_pK.out', 'all_sum_crg.out', 'entropy.out', 'fort.38', 'head3.lst', 'pK.out', 'residues_stats.txt', 'sum_crg.out', 'vdw0.lst'.

# fix_psf_mdanalysis

Provides a reformatted PSF file if "MDAnalysis" fails to parse the given PSF. Requires the "MDAnalysis" and "parmed" packages.

# getpdb

Downloads one or more (bioassembly) PDB files from the RCSB Protein Databank. For example, to download triclinic hew lysozyme (4LZT), one could type in

usage: getpdb [RCSB protein code]

# glossary

Gives detailed information regarding the various parameters of run.prm, where MCCE looks to handle more granular customization.

You can search for specific parameters by with a given (case-sensitive) prefix string. For example, "glossary T" will return all parameters starting with T, like "TITR_TYPE". The command "glossary --print" also prints the entire glossary.

# hbonds_pdb_collection

Detects Hydrogen bonds, using detect_hbonds.py, over a collection of PDB files, in the step2_out.pdb format. ASK HOW TO USE THIS

usage: hbonds_pdb_collection [-h] [-input_dir INPUT_DIR] [-output_dir OUTPUT_DIR] [--include_bk] [--no_empty_files]

# mcce_stat

Prints a table to keep track of progressing MCCE runs. Four "sentinel" files are looked for, to signify completion of each of the four basic steps of MCCE: step1_out.pdb, step2_out.pdb, head3.lst, and pK.out.

pK.out signifies completion of step 4, so if a book.txt exists for a protein when mcce_stat is run, that protein will receive a "c" in book.txt to signify completion.

We recommend using mcce_stat with p_batch.

# ms_hbond_percentages.py

Creates a table displaying all Hydrogen bond connections across microstate PDBs, and their percentages. Defaults to the local directory named pdb_output_mc_hbonds.

usage: ms_hbond_percentages.py [-h] [dir]

# ms_top2pdbs

Stands for Tautomeric Charge MicroStates. Outputs: the top N tautomeric charge microstates, along with related properties energy (E), net charge (sum_crg), count, and occupancy (occ); a summary file identifying ionizble residues with non-canonical charge, and which residues that do not change charge over the topN set; and the top N files of each charge state in PDB and PQR format.

By default, charge microstates are retrieved at pH 7, and the number of most favorable charge microstates (N_TOP) returned is five.

usage: ms_top2pdbs inputpdb_filepath [-ph PH] [-n_top N_TOP]

# *p_batch (MCCE_bin)*

Starts multiple protein runs at once, using the same set of instructions, and creates a book.txt file to manage their completion status. p_batch accepts a directory containing protein files, and (optionally) a shell script given custom instructions. If a shell script is not provided, a default one will be created, and may be edited to the user's preference. If a file named "run.prm.custom" is in the present working directory at runtime, the file will be read to override the default run.prm instructions.

p_batch creates a run directory for each protein file, and begins running MCCE for each one. Files will be created for their respective directories as each step is completed. Use mcce_stat to check how each run is progressing.

To stop a run in progress, delete the files or directory associated with the run.

# *p_info (MCCE_bin)*

Gives a high-level summary of characteristics of a PDB file, including residue, chain, and ligand counts, as well as other aspects of a PDB changed during step 1 of MCCE, including how residues are named. If step 1 has not been run on the PDB file at runtime, p_info will automatically run step 1 before continuing as normal.

# pdbs2pse (MCCE4-Tools)

usage: pdbs2pse file1.pdb file2.pdb ... [--pse_name <output_name>]

Converts one or more PDB files into a single PyMOL session file (.pse). The session file contains all the loaded PDB structures as separate objects. The user can specify an optional output name for the .pse file, or it will default to the name of the last input PDB file.

# postrun (MCCE4-Tools)

usage (in a directory with sum_crg.out, pK.out files): postrun [-h] [-run_dir RUN_DIR] [--is_benchmark]

postrun provides basic diagnostics on sum_crg.out and pk.out files, after a run is completed. postrun looks for non-canonically charged residues, residues without curve fit or a chi-squared above 3, and residues that are out-of-bounds. The problem residues are outputted to the terminal and saved to a "postrun.bad" file. If there are no problem residues, a "postrun.ok" file is created instead.

postrun can be run on a directory of completed protein runs, with the flag "-run_dir".

# txt_to_csv

A quick script that copies a given file into a .csv format. The source file does not need to be a .txt file. Recommended to use with spreadsheets.

---

Revision #5
Created 16 April 2025 18:05:41 by Jared
Updated 16 June 2025 17:47:26 by Jared