Implementation of the LipIDens pipeline: assisted interpretation of lipid densities in membrane protein structures using simulations

T. Bertie Ansell  
Department of Biochemistry, University of Oxford, University of Oxford, South Parks Road, Oxford, OX1 3QU, UK  https://orcid.org/0000-0003-4870-5387

Wanling Song  
MSD R&D Innovation Centre, 120 Moorgate, London, EC2M 6UR, UK  https://orcid.org/0000-0003-4996-5973

Claire E. Coupland  
Molecular Medicine Program, The Hospital for Sick Children, Toronto, M5G 0A4, Canada.  
https://orcid.org/0000-0002-6041-7532

Loic Carrique  
Division of Structural Biology, Wellcome Centre for Human Genetics, University of Oxford, Roosevelt Drive, Oxford, OX3 7BN, UK.  https://orcid.org/0000-0001-5332-8593

Robin A. Corey  
School of Physiology, Pharmacology and Neuroscience, University of Bristol, Bristol, BS8 1TD, UK.  https://orcid.org/0000-0003-1820-7993

Anna L. Duncan  
Department of Chemistry, Aarhus University, Lagelsandsgade 140, 8000 Aarhus C, Denmark.  
https://orcid.org/0000-0001-9873-4552

C. Keith Cassidy  
Department of Physics and Astronomy, University of Missouri-Columbia, Columbia, MO 65211, USA.  
https://orcid.org/0000-0001-8106-1187

Maxwell M. G. Geurts  
Department of Biochemistry, University of Oxford, University of Oxford, South Parks Road, Oxford, OX1 3QU, UK.  https://orcid.org/0000-0001-7077-5336

Tim Rasmussen  
Biocenter and Rudolf-Virchow-Zentrum, Universität Würzburg, Haus D15, Josef-Schneider-Str. 2, 97080 Würzburg, Germany  https://orcid.org/0000-0003-1126-846X

Andrew B. Ward  
Department of Integrative Structural and Computational Biology, The Scripps Research Institute, La Jolla, California 92037, USA  https://orcid.org/0000-0001-7153-3769

Christian Siebold
Method Article

Keywords: Simulation, lipid, molecular dynamics, cryo-EM, density, structure, membrane protein

Posted Date: October 19th, 2023

DOI: https://doi.org/10.21203/rs.3.pex-2408/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Abstract

Interpretation of lipid and lipid-like densities surrounding membrane protein structures solved by cryo-electron microscopy (cryo-EM) is challenging. We developed the LipIDens pipeline for assisted structural interpretation of lipid densities using molecular dynamics (MD) simulations. This protocol details how to establish coarse-grained (CG) simulations of membrane proteins in biomimetic membranes, test lipid interaction cut-offs, calculate lipid interactions and binding sites, assess the quality of derived kinetics, compare lipid poses with structural densities, rank site lipids and refine lipid poses using atomistic simulations. These data provide a platform for identifying the most probable lipid accounting for a lipid-like density, assisting the modelling and interpretation of membrane protein structures.

Introduction

Reagents

Equipment

Expertise needed to implement the protocol:

To implement the protocol a user should be familiar with basic terminal commands and navigation. Prior experience with setting up and running CG or atomistic MD simulations using e.g. GROMACS (www.gromacs.org) is advisable, for which tutorials are available elsewhere (http://www.mdtutorials.com/gmx/, http://cgmartini.nl/index.php/tutorials).

Equipment:

Input data:

- Protein coordinate file in .pdb format as obtained either structurally or e.g. from the PDB (https://www.rcsb.org/). Protein models are also permitted and may be useful for switching to a different homologue (e.g. https://alphafold.ebi.ac.uk)\(^1\).

- Corresponding cryo-EM density map for the protein (locally determined or downloaded from the Electron Microscopy Data Bank (EMDB) (https://www.ebi.ac.uk/emdb/).
Optional input files:

- Topology files for the system. Coarse-grained MARTINI parameters (http://cgmartini.nl/index.php/downloads) are automatically downloaded in the protocol or alternative parameters can be provided by the user. For atomistic simulations, CG2AT provides a choice of forcefields automatically².

- Molecular dynamics parameter files. Simulation parameter files (e.g. mdp files for use with GROMACS) are automatically provided in the pipeline but users may wish to supply alternatives. The default linear constraint solver (LINCS) parameters (lincs_order=4, lincs_iter=1) are used in mdp files unless MARTINI-2.2 cholesterol with virtual sites³ is included in the bilayer, whereby lincs_order=12 and lincs_iter=2 are used instead, in line with recent findings⁴.

Computational hardware:

- A computer or laptop with an operating system capable of running a terminal (Linux or MacOS recommended). For running simulations, a desktop computer, computational cluster or local high-performance computing facility is recommended for improved running times. Alternatively, national/cloud compute resources can also be used to run simulations.

Computational software:


- Software for density map visualisation for example Coot⁵ (https://www2.mrc-lmb.cam.ac.uk/personal/pemsley/coot/), UCSF Chimera (https://www.cgl.ucsf.edu/chimera/), ChimeraX⁶(https://www.rbvi.ucsf.edu/chimerax/) or PyMOL (https://PyMOL.org/2/).

- Software for trajectory visualisation (optional) e.g VMD⁷ (https://www.ks.uiuc.edu/Research/vmd/) or PyMOL (https://PyMOL.org/2/).


- The LipIDens pipeline (https://github.com/TBGAnsell/LipIDens). LipIDens can be installed by following the instructions of the GitHub repository. LipIDens uses the following additional packages which are automatically installed (see https://github.com/TBGAnsell/LipIDens/blob/main/requirements.txt for updated list of package dependencies):
PyLipID (version >=1.5). PyLipID is also available from https://github.com/wlsong/PyLipID and can be installed using pip install pylipid or git clone git://github.com/wlsong/PyLipID.git. PyLipID also implements the following additional packages (installed automatically): kneebow, logomaker, matplotlib>=3.3.4, mdtraj, networkx, numpy, p-tqdm, pandas, python-louvain, scikit-learn, scipy, seaborn, statsmodels, tqdm.

Martinize2 (version >=0.7) (https://github.com/marrink-lab/vermouth-martinize). Martinize2 can also be installed using pip install vermouth or pip install git+https://github.com/marrink-lab/vermouth-martinize.git#vermouth.

- dssp (https://swift.cmbi.umcn.nl/gv/dssp/).

Procedure

Procedure:

LipIDens may be run either using an interactive standalone master python file (`lipidens_master_run.py`) or by pre-defining variables within the pipeline jupyter (https://jupyter.org) notebook (`LipIDens.ipynb`). We recommend running primarily through the master python file as this accounts for stage termination and re-initiation or later pipeline stages, while the jupyter notebook is useful for tutorial purposes. The pipeline is composed of multiple stages, described under the subheadings below. When running LipIDens using the master python script, users are prompted to enter a number of variables. In the notebook, these are defined under the 'USER DEFINED VARIABLES' section, followed by the corresponding 'CODE' for each step.

Analysis stages of the protocol can be run independently if the user has pre-existing CG simulations they wish to examine, however we recommend some familiarity with all protocol steps.

The GROMACS 2019 MD simulation software (https://www.gromacs.org/) was used to run simulations. The MARTINI-2.2 forcefield was used for CG simulations. The protocol could be adapted for use with other MD simulation software (e.g Amber, NAMD) or CG forcefields (e.g. MARTINI-3, ESPResSo, Fat SIRAH).
Structure processing:

TIMING: Steps 1-2, 5 mins, Step 3, 10-20 mins.

1. Load the input .pdb file in PyMOL and remove any superfluous components by selecting only the protein of interest and saving as a new .pdb file.

2. The input .pdb file should not have any residues with missing atoms but larger missing segments (e.g. loops, TM helices etc) are permitted. Simulation competent .pdb files of published membrane protein structures can be downloaded from MemProtMD [http://memprotmd.bioch.ox.ac.uk](http://memprotmd.bioch.ox.ac.uk) or adjusted using [https://github.com/pstansfeld/MemProtMD/blob/main/PDB-fix.ipynb](https://github.com/pstansfeld/MemProtMD/blob/main/PDB-fix.ipynb). Decide whether any additional changes to the .pdb are required such as a) addition of missing atoms (required) or b) modelling of missing protein segments (optional) (step 3). If none, proceed to step 4.

3. Model missing atoms/segments using a preferred modelling software (MODELLER/Chimera-MODELLER\textsuperscript{15}, SWISS-MODEL\textsuperscript{16}, trRosetta\textsuperscript{17}, AlphaFold2\textsuperscript{1}, RoseTTAFold\textsuperscript{18}).

Setting up and performing coarse-grained simulations:

TIMING: Step 4, 2 mins, Steps 5-6, ~2-10 days (variable depending on available compute resources, simulation system size and length)

4. Type python lipidens_master_run.py to initiate the LipIDens pipeline. The user will be prompted to select the directory to store simulations/data (save_dir) and the protocol stage (‘1a’). Default settings can be selected by pressing the ENTER/RETURN key. Follow the prompts to set a number of simulation variables described below. If using the jupyter notebook, define the variables for the first section of the protocol by modifying the file:

   i. Change save_dir to a directory name in which to build the system and save analysis.
   ii. Change protein_AT_full to the location of the input protein .pdb file.
   iii. Set nprot to the number of homomeric protein chains or use nprot = 1 for heteromers.
   iv. The variables protein_shift and protein_rotate can be used to alter the alignment of the protein within the bilayer. To alter the z axial position for protein insertion into the bilayer change the protein_shift value (decimal, negative and positive numbers accepted). Set the protein_rotate angle (in
x y z) with respect to alignment along the first principal axis (default ‘0 90 0’ i.e. rotate in y by 90°). The position of the bilayer in MemProtMD (https://github.com/pstansfeld/MemProtMD/blob/main/MemProtMD_Self_Assembly.ipynb) can be used to guide value selections\textsuperscript{14}. TROUBLESHOOTING

v. Set the simulation boxsize (nanometres in x y z).

vi. Set the CG forcefield to use in simulations (currently compatibility with martini_v2.0, martini_v2.1, martini_v2.2 and martini_v3.0.0)\textsuperscript{11,19}.

vii. The membrane_composition can be defined using either a) a predefined bilayer type or b) a custom bilayer composition. Current predefined bilayer names (e.g. ‘Plasma membrane’) and corresponding compositions are provided on the GitHub page. To build a custom bilayer change the membrane_composition variable to the chosen composition in the upper (-u) and lower (-l) leaflets using insane.py syntax\textsuperscript{20}. For example, for a bilayer composed of 100% POPC in the upper leaflet and 90% POPC plus 10% POPS in the lower leaflet then membrane_composition=’-u POPC:100 -l POPC:90 -l POPS:10’. Note, currently only a subset of lipids are available for the MARTINI-3 forcefield. Users should check whether the required lipids are available for a specific forcefield.

viii. Set martini_maxwarn to the maximum number of warmings permitted when running the martinize command (default 0).

ix. Change the ring_lipids variable to ‘True’ to place lipids within the protein when assembling the bilayer (default ‘False’).

x. Alter CG_simulation_time to the duration of each CG simulation in $\mu$s and set the number of replicates. It is recommended to run simulations for at least 10 $\mu$s and to run multiple repeats (at least 8 recommended) however the time taken to reach convergence will be system dependant and users should adjust accordingly. The ‘Screening PyLipID data’ section details how the quality of binding site data can be assessed using PyLipID outputs and, if necessary, users should consider extending the simulations.

xi. Finally alter n_cores to the number of CPUs to be used by GROMACS when setting up the CG simulations.

5. If using the notebook, run the code corresponding to CG simulation setup (up to the stage marked ‘PAUSE POINT’ after the ‘run_CG() function). This will happen automatically if using the master python script. The output of this step is a GROMACS md.tpr file for each CG simulation replicates. An outline of the commands automatically implemented within this section is given below:

i. Convert the protein to CG resolution using martinize$^{21}$ (https://github.com/marrink-lab/vermouth-martinize) with an ELNeDyn elastic network$^{22}$ (for MARTINI-2) or Martini3001 forcefield (for
MARTINI-3). A spring force constant of 1000 kJ mol\(^{-1}\) nm\(^{-2}\), a lower cut-off of 0.5 nm and an upper cut-off of 0.9 nm are applied. Users may alter the default elastic network, force constant and cut-off values within the lipidens/simulation/CG_simulation_setup.shscript using the -ff, -ef, -el, -eu, -ea and -ep flags. For more information see [http://cgmartini.nl/index.php/tools2/proteins-and-bilayers/204-martinize](http://cgmartini.nl/index.php/tools2/proteins-and-bilayers/204-martinize).

ii. Embed the CG protein into a bilayer (of composition provided with the membrane_composition variable) and solvate with water using insane.py\(^{20}\). The position of the TMD within the bilayer is set using the protein_shift and protein_rotate variables. ?TROUBLESHOOTING.

iii. Neutralise the system by addition of ions using gmx grompp and gmx genion. By default, the system is neutralised using ~0.15 M NaCl however the cation and anion names and concentration can be altered using the -pname, -nname and -conc flags within CG_simulation_setup.sh.

iv. Make an index file (gmx make_ndx) which contains groups corresponding to the Protein, Lipids and Solvent (water and ions).

v. Generate a .tpr file for energy minimisation using gmx grompp and run the energy minimisation using gmx mdrun. ?TROUBLESHOOTING.

vi. Equilibrate the system using gmx grompp and gmx mdrun. In the first equilibration step restraints are applied to the protein backbone (BB) beads. In the second equilibration step restraint are not applied. ?TROUBLESHOOTING

vii. Generate the .tpr file for the production run using gmx grompp.

6. Run the CG simulation using gmx mdrun. ?TROUBLESHOOTING. It is recommended to offload this calculation to a cluster or high-performance computing facility. ■PAUSE POINT: wait until CG simulations have finished running before proceeding.

**Testing PyLipID cut-offs:**

TIMING: Step 7, 5-15 mins, Step 8, 2 mins, Step 9, ~1 h per lipid species, Step 10, 5 mins

7. Once all the CG simulations have reached completion, trajectories can be processed. Type python lipidens_master_run.py and select the protocol stage ("1b") when prompted, or run the corresponding notebook code. The trjconv_CG() function makes molecules whole across the periodic boundary and skips the number of frames provided with stride. Although not technically required for subsequent analysis, trajectory processing can improve the usability of outputted lipid binding poses from PyLipID and reduce PyLipID running times.
i. Set stride to the number of frames to skip during trajectory processing and downstream analysis of protein-lipid interactions (recommended to speed up processing).

8. Once the CG replicates are processed, PyLipID analysis can be performed. ▲ CRITICAL STEP: In this first step, PyLipID is used to test a range of lower and upper cut-off values for lipid interactions with the protein. In general, users should test cut-offs for several chemically diverse lipids such as e.g. sterols or phospholipids. Run LipIDens and select the protocol stage (‘2’) when prompted. Within the notebook/master script set the user defined variables for the second section of the pipeline:

i. Set the lipid_atoms variable to the CG bead names PyLipID will use for cut-off testing. The default (lipid_atoms=None) will use all CG beads for each lipid.

ii. In the first stage of cut-off testing, the minimum distance of each lipid to a residue is plotted, provided that the lipid comes within distance_threshold of the residue for longer than the number of contact_frames. Set the distance_threshold value to a reasonably generous interaction distance (i.e. 0.65 nm for CG simulations or 0.4 nm for atomistic simulations). Select a value for contact_frames to screen interacting lipids.

iii. In the second stage of cut-off testing, a list of upper and lower cut-offs are exhaustively screened in a pairwise fashion. Change the lower_cutoff and upper_cutoff variables to lists of cut-off values to test (in nm).

iv. Change timeunit to the preferred axis unit on analysis plots.

9. Run the code corresponding to PyLipID cut-off screening (up until the next segment of user defined variables) within the notebook. This will happen automatically if using the python script. An outline of the steps implemented in this section are described below:

i. Calculate the minimum distances of each interacting lipid to a residue over the length of the trajectory. Plots are provided in the ‘PyLipID_cutoff_test_Lipid/Figures’ subdirectory.

ii. Plot the probability distribution of minimum distances between the lipid and the protein.

iii. Exhaustively test a range of upper and lower cut-off value pairs. The output is a plot of interaction duration times, number of calculated binding sites and number of contacting residues for each dual-cut-off combination.

10. Using the distribution plot from step 9ii and the exhaustive cut-off testing in step 9iii, select the lower and upper interaction cut-offs to use when running PyLipID.

i. The lower cut-off localises to the first solvation peak in the probability distribution plot. Additionally, the lower cut-off corresponds to an increase in interaction durations, computed binding sites and residues comprising each site compared with smaller lower cut-off values.
ii. The upper cut-off localises to the first trough between the first and second interaction shells in the probability density distribution. The upper cut-off is appropriate when interaction metrics plateau. If interaction metrics increase further as upper cut-off is increased this is an indication that the second solvation shell is being captured which should be avoided.

**Selecting PyLipID input parameters and running PyLipID analysis:**

TIMING: Step 11, 5 mins, Step 12, ~15 mins per lipid species

11. Next, lipid interactions, kinetics and binding sites are calculated using PyLipID. Run the LipIDens python script and select the appropriate stage (‘3’) when prompted. In the next user defined variables section of the notebook/master script set the following variables to run site analysis using PyLipID:

i. Set the cutoffs variable to the selected lower and upper cut-off (step 10). Additional information regarding cut-off selection is provided at [https://pylipid.readthedocs.io/en/master/tutorials](https://pylipid.readthedocs.io/en/master/tutorials).

ii. Tune the lipid_atoms variable based on the putative lipid densities present in the cryo-EM map. If only headgroup-like density is present the lipid_atoms variable can be restricted to the CG headgroup beads to speed up calculation times. If tail density is present it is recommended to perform calculations on all lipid atoms however the search could be restricted to beads comprising the tail if required. This can be useful for assessing the relative contribution of different lipid segments to binding site residence times.

iii. If multiple, identical proteins and/or protein complexes are present, such as in homo-oligomeric ion channels, set the nprot flag to the copy number in the system. This will average calculated kinetic parameters over repeat domains and improve protein-lipid contact sampling.

iv. Set the binding_site_size valuable to the minimum number of residues that can comprise an identified binding site (default 4). This is recommended to avoid artefactual identification of very small ‘binding sites’ due to non-specific interactions.

v. Select the number of top lipid binding poses to be outputted for each binding site using the n_top_poses variable (default 3). At each site the specified number of representative lipid binding poses will be calculated using an empirical scoring function to rank lipid binding sites against the simulation derived lipid density at the site.

vi. Alter the n_clusters variable to calculate the number of distinct lipid pose clusters to be calculated for each site. This can be useful for assessing the conformational diversity of lipid binding poses at a site. If n_clusters is set to auto (default) PyLipID will use a density-based clustering algorithm to identify all possible clusters.
vii. Set save_pose_format to the coordinate file format for outputted lipid poses.

viii. Set save_pose_traj to True to output lipid binding poses to a trajectory format provided with save_pose_traj_format.

ix. Set the timeunit to use in outputted data.

x. Alter the resi_offset to offset the residue index number in outputs.

xi. The radii variable should be used to set the Van der Waals radius of non-standard atoms in a trajectory. The Van der Waals radius of common atoms are already accounted for, including CG beads in MARTINI 2.0-2.2.

xii. Set the pdb_file_to_map variable to an atomistic protein coordinate file (such as from step 4ii) onto which binding site information will be mapped by PyLipID.

xiii. Set the fig_format variable to the preferred image output file extension.

xiv. Change num_cpus to the number of CPUs to use during multiprocessing steps of PyLipID.

12. Perform PyLipID analysis on lipids in the CG simulations by running the corresponding section of code in the notebook. PyLipID runs automatically if using the master script. It is worth noting that PyLipID is highly modular and contains a number of functions that can be run independently to study other biological phenomena. Please refer to [https://pylipid.readthedocs.io/en/master/](https://pylipid.readthedocs.io/en/master/) for details on how to write custom analysis scripts and/or select only those outputs of interest. PyLipID creates an 'Interaction_Lipid' directory containing the outputs for each lipid. This includes a 'Dataset_Lipid' directory containing data stored in pickle format and a summary of the kinetics associated with each residue and binding site (Dataset.csv). This subdirectory also includes a PyMOL script for automatically mapping binding site kinetics onto the atomistic structure provided with pdb_file_to_map. Other outputs include top ranked and/or clustered binding poses for each binding site (within the 'Bound_Poses_Lipid' subdirectory) and .pdb files with kinetics mapped to the B-factor column (within the 'Coordinate_Lipid' subdirectory).

**Screening PyLipID data:**

**TIMING: Steps 13-16, 5 mins**

13. The next stage of the pipeline involves inspecting and screening the PyLipID outputs. Within the notebook/master python file run the next section of code (‘4’) to rank binding site kinetics.

14. Inspect the output plot (Site_stats_rank_compare.pdf) located within the ‘Interaction_Lipid’ directory. The script ranks lipid binding sites from lowest to highest Occupancy, Residence time or Surface area.
or \( \Delta k_{\text{off}} \) closest to 0 (defined as the difference between the \( k_{\text{off}} \) calculated form the curve fit of the survival function and the bootstrapped \( k_{\text{off}} \) of the same data). This plot can be used to inspect the quality of calculated binding sites by e.g. comparing sites which rank highly (in their Residence times/Occupancies) and are well fitted/sampled. Typically, a good site has a \( \Delta k_{\text{off}} \) between \( \pm 1 \) \( \mu \)s. ▲ CRITICAL STEP: Always inspect the quality of the identified binding sites and remove any poor sites from future analysis.

15. Review the binding site \( k_{\text{off}} \) plots (BS_idX.pdf) located within the ‘Interaction_Lipid/Binding_Sites_koffs_Lipid’ directory. Well sampled binding sites which rank highly should show good agreement between the bootstrapped and bi-exponential curve fits to the survival function and sufficient sampling of interaction durations. Poorly fitted sites are indicated by disagreement between bootstrapped and bi-exponential curve fits and/or sites which an infrequently observed as indicated by a sparse interaction duration plot. This serves as a second method for assessing binding site quality in addition to the Site_stats_rank_compare.pdf plot. Finally, \( R^2 \) values for the residence times of each binding site are found within the Dataset.csv and BindingSites_Info_Lipid.txt files. These can also be used to assess whether CG simulations have been run for long enough to sufficiently sample protein-lipid interactions and to yield reliable outputs from PyLipID.

16. Exclude any poor binding sites from future analysis (e.g. pose/density comparisons).

**Comparing lipid poses with cryo-EM densities and ranking site lipids:**

TIMING: Step 17, 1 min, Step 18, ~10 mins, Step 19, 1 min, Steps 20-21, ~10 mins

17. In this stage of the protocol the residence times of different lipids binding to the same site are compared. Site poses are subsequently backmapped to atomistic resolution and loaded into an interactive PyMOL (https://PyMOL.org/2/) session for direct comparison with partitioned site densities. Run the LipIDens python script and select the appropriate stage (‘5’). Binding site residues for different lipids are iteratively compared to those of the reference lipid (first lipid inputted). Sites which match most closely are selected across lipids species i.e. to compare different lipids binding to similar site locations. Corresponding binding sites IDs are written out in order and stored as a python dictionary BindingSite_ID_dict. Where corresponding sites could not be identified these are marked by ‘X’. ▲ CRITICAL STEP: Check predicted binding site matches by comparing the lipid poses/binding sites from PyLipID. Users may visualise the CG bound lipid poses outputted by PyLipID within the ‘Interaction_Lipid/Bound_poses_Lipid’ directory using VMD. Top ranked lipid binding poses (‘BSidX_rank’) and clustered poses (‘BSidX_clusters’) are located within subsidiary directories for each binding site. The BindingSite_ID_dict may need to be adjusted if a site was previously identified as poor (see
‘Screening PyLipID data’) or is assigned multiple times. If you are happy with predicted sites, accept the BindingSite_ID_dict, run the corresponding code and proceed to step 19. If not, follow step 18.

18. Within the notebook/master python script change the BindingSite_ID_dict dictionary keys to the lipids to compare. Set the corresponding values for each lipid to a list of binding site IDs for each corresponding site. For example, to compare the residence time of POPC binding site 2 and POPE binding site 4 (assuming these correspond to similar locations on the protein) then BindingSite_ID_dict={‘POPC’: [2], ‘POPE’: [4]}. If the lipid does not bind to a site then set the site ID to “X”.

19. Run the corresponding code to plot a comparison of residence times and R² values for each site across lipid species. A ‘Lipid_compare’ directory containing plots (Lipid_compare_BSstats_PyLipID_Site_idxY_ref_Lipid_BSX.pdf) for each site are generated where Y is an ordered index for each site and X corresponds to the reference lipid binding site ID number in BindingSite_ID_dict e.g. BindingSite_ID_dict={‘POPC’: [2, 3], ‘POPE’: [4, 5]} would produce two plots numbered Lipid_compare_BSstats_PyLipID_Site_idx0_ref_POPC_BS2.pdf (comparing POPC site 2 with POPE site 4) and Lipid_compare_BSstats_PyLipID_Site_idx1_ref_POPC_BS3.pdf (comparing POPC site 3 with POPE site 5) respectively.

   i. The generated plots can be used to infer the most likely identity of a lipid species accounting for a density within the cryo-EM map.

20. Next the top-ranked CG lipid binding poses for all poses within the BindingSite_ID_dict are automatically converted to atomistic resolution using CG2AT². ?TROUBLESHOOTING. The density map is loaded and segmented around each site (at the specified sigma_value based on the binding site residues identified by PyLipID. Backmapped lipid poses are aligned to the atomistic structure based on the protein coordinates such that lipid poses (for all lipids bound to a site) can be directly compared to site densities within an interactive PyMOL session (https://PyMOL.org/2/). Set the following variables to run this section:

   i. Set protein_AT_full_align to the protein coordinate .pdb file which aligns with the density map.

   ii. Set density_map to the corresponding cryo-EM density map. Note, protein_AT_full_align and density_map must be aligned.

   iii. Set sigma_factor to the level at which to display the density map.

21. Navigate into the ‘Density_Pose_Compare’ directory. A series of subdirectories are generated (‘Site_idxY_BS_ID_X’) where Y is an ordered site index and X is the binding site of the reference lipid. Within each subdirectory are the backmapped atomistic poses for all lipids which bind to that site (i.e. matching lipids sites within BindingSite_ID_dict). To load the interactive PyMOL session type pymol Lipid_poses_density_compare.py. The script will load and align the coordinates of the lipid poses to the protein_AT_full_align structure. The binding site residues of the reference lipid are shown as spheres
scaled by residence time. The cryo-EM density map is partitioned around each binding site at the specified sigma_factor for direct comparison with all lipids which bind to the sites. Lipid poses, binding sites and map segments are coloured identically for each site, facilitating pose/density comparisons.

**Lipid pose refinement using atomistic simulations:**

**TIMING:** Steps 22-23, 30 mins, Step 24, ~4-10 days (variable depending on available compute resources, simulation system size and length), Steps 25-26, 20 mins

22. The final section of the protocol is optional and relates to the refinement of CG lipid poses using atomistic simulations. Run the master python script and select the appropriate step (‘6’). Set the following user defined variables:

   i. Set input_CG_frame to the CG simulation frame to use for back-mapping to atomistic resolution. Specified CG frames can be written to coordinate files using gmx trjconv with the -dump flag. The replicate and frame from which a lipid binding pose was obtained is noted within the pose_info.txt file (within with ‘BSidX_rank/cluster’ subdirectories).

   ii. Set protein_AT_full to the atomistic structure used to establish CG simulations during the first stage of the protocol. To use an alternative input structure (e.g. including alternative protonation states or conformations), redefine protein_AT_full within this section as the modified input protein pdb file.

   iii. Set model_type to either ‘de_novo’ or ‘aligned’ to select the output model from CG2AT to use for atomistic simulations. In the *de novo* model the protein coordinates are mapped to their positions within the CG frame. In the aligned model the protein coordinates are mapped to those of the input atomistic pdb (protein_AT_full).

   iv. Change replicates_AT to the number of atomistic simulation replicates.

   v. Change AT_simulation_time to the simulation time in nanoseconds.

23. Run the code corresponding to atomistic simulation setup (up to the stage marked ‘PAUSE POINT’). This is done automatically in the master python script. The output of this step is a GROMACS md.tprfile for each atomistic simulation replicate. An outline of the commands implemented within this section is given below:

   i. Convert the protein-lipid system from CG to atomistic resolution using CG2AT, details of which are provided in \(^2\). The protein conformation is backmapped either based on the coordinates in the CG frame (model_type=’de_novo’) or those of the atomistic structure (model_type=’aligned’). Lipid coordinates are backmapped to their positions in the CG frame. Users may select an atomistic forcefield
and water model to build the system. The system is energy minimised and equilibrated. 

**TROUBLESHOOTING**

ii. Generate the .tpr file for the production run using gmx grompp.

24. Run the atomistic simulation using gmx mdrun. **TROUBLESHOOTING.** It is recommended to offload this calculation to a cluster or high-performance computing facility. ■PAUSE POINT: wait until atomistic simulations have finished running before proceeding.

25. Once the atomistic simulations have finished running, compare the refined lipid binding pose to the cryo-EM density by loading in a preferred visualisation software.

26. Evaluate the match between the simulation derived lipid pose and the cryo-EM density using Q scores\(^23\). Details on implementation of this step in UCSF Chimera are given below:

i. Align the simulation frame with the atomistic input structure to which the density map corresponds. In PyMOL this can be done using the align or cealign commands, selecting the protein C\(\alpha\) or backbone beads of each structure (e.g. cealign *structure_name* and name CA, *simulation_frame_name* and name CA). It is often easier to remove superfluous components (i.e. everything except the protein and lipid of interest) from the system and save as a new .pdb file.

ii. Open UCSF Chimera and ensure the MapQ plugin\(^23\) is installed (for details see [https://github.com/gregdp/mapq](https://github.com/gregdp/mapq)).

iii. Load the aligned simulation frame and cryo-EM map.

iv. Open the MapQ plugin using Tools > Volume Data > MapQ.

v. Enter the resolution of the cryo-EM map in the box marker ‘Res:’ and click ‘Calc’ to calculate Q scores.

vi. After the Q score calculation has finished. Select a protein sequence in proximity to the lipid using Ctrl-D and click to show the per atom Q scores on the structure. For further details see [https://github.com/gregdp/mapq/tree/master/tutorials](https://github.com/gregdp/mapq/tree/master/tutorials). Q scores are also mapped to the B-factor column of an output .pdb file from MapQ. It is expected that low/ negative Q scores may be observed for lipid regions outside of observed densities due to increased fluctuation of non-bound lipid regions.

**Troubleshooting**

Troubleshooting advice can be found in Supplementary Table 1.

**Time Taken**

Steps 1-3, Structure processing: ~20 mins
Steps 4-6, Setting up and performing coarse-grained simulations: ~1 week

Steps 7-10, Testing PyLipID cut-offs: ~4 h

Steps 11-12, Selecting PyLipID input parameters and running PyLipID analysis: ~3 h

Steps 13-16, Screening PyLipID data: 5 mins

Steps 17-21, Comparing lipid poses with cryo-EM densities and ranking site lipids: ~25 mins

Steps 22-26, Lipid pose refinement using atomistic simulations: ~1 week

**Anticipated Results**

**References**


Acknowledgements

The LipIDens pipeline was developed by T.B.A, W.S., R.A.C. and A.L.D. who contributed methodological development of PyLipID. Code testing was performed by R.A.C., C.K.C. and M.M.G.G. Development of the LipIDens pipeline was assisted by the expertise of C.E.C., L.C., T.R., C.S., A.B.W., P.J.S. and M.S.P.S.

T.B.A, C.E.C and M.M.G.G. are supported by Wellcome (102164/Z/13/Z). W.S. was supported by a Newton International Fellowship. R.A.C., A.L.D. M.S.P.S. and P.J.S. are funded by Wellcome (208361/Z/17/Z). A.L.D. has been additionally supported by BBSRC (BB/R00126X/1) and the Department of Biochemistry. C.K.C. is funded by the BBSRC (BB/S003339/1). P.J.S. is also supported by the BBSRC (BB/P01948X/1, BB/R002517/1, and BB/S00339/1), and the MRC (MR/S009213/1) and M.S.P.S.'s research is supported by the BBSRC (BB/R00126X/1) and PRACE (Partnership for Advanced Computing in Europe, 2016163984). C.S. is supported by Cancer Research UK (C20724/A26752), the BBSRC (BB/T01508X/1) and the European Research Council (647278). L.C. is supported by a Wellcome administrative support grant (203141/Z/16/Z). A.B.W. acknowledges support from the Ray Thomas Edwards Foundation.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ProtocolsTable1.pdf