NMRlipids project

Open Collaboration to understand lipid systems in atomistic resolution

- We have mainly focused to find models that correctly capture lipid headgroup conformational ensemble and ion binding
- CHARMM36 has best conformational ensembles, but not correct[1,4]. However, it captures many essential differences between PC, PE, PG and PS headgroups[4b].
- Cation binding affinity to PC, PG and PS headgroups can be improved by electronic continuum correction (ECC)[5,6].
- How about other lipids and ions? Membrane proteins? Other properties than headgroup? etc.

[1] http://dx.doi.org/10.1021/acs.jpcb.5b04878 (NMRlipids I, 2015)
[2] http://dx.doi.org/10.1039/C6CP04883H (NMRlipids II, 2016)
[3] https://github.com/NMRLipids/NmrLipidsCholXray/blob/master/MANUSCRIPT/manuscript.pdf (NMRlipids III, in progress)
[4] https://pubs.acs.org/doi/10.1021/acs.jpcb.9b06091 (NMRlipids IV, 2019)
[4b] https://github.com/NMRLipids/NMRlipidsIVPEandPG/blob/master/Manuscript/manuscriptPGPE.pdf (NMRlipids IVb, in progress)
[5] http://dx.doi.org/10.1021/acs.jpcb.7b12510
[6] http://dx.doi.org/10.1021/acs.jctc.9b00824

NMRlipids project

Open Collaboration to understand lipid systems in atomistic resolution

Some relevant lipid headgroups and their mixtures

	PC	PE	PG	PS	chol	CL	GM1	PIP	PA	DAG
PC	X[1,2]	X[4b]	X[4b]	X[4]	X[3]					
PE		X[4b]								
PG			X[4b]							
PS				X[4]						
chol										
CL										
GM1										
PIP										
PA										
DAG										

[1] http://dx.doi.org/10.1021/acs.jpcb.5b04878 (NMRlipids I, 2015)
[2] http://dx.doi.org/10.1039/C6CP04883H (NMRlipids II, 2016)
[3] https://github.com/NMRLipids/NmrLipidsCholXray/blob/master/MANUSCRIPT/manuscript.pdf (NMRlipids III, in progress)
[4] https://pubs.acs.org/doi/10.1021/acs.jpcb.9b06091 (NMRlipids IV, 2019)
[4b] https://github.com/NMRLipids/NMRlipidsIVPEandPG/blob/master/Manuscript/manuscriptPGPE.pdf (NMRlipids IVb, in progress)

NMRlipids databank

Quality evaluated atomistic resolution MD simulations of biologically relevant lipid mixtures

- User/contributor feeds information of the data (location etc.)
- Databank **automatically** indexes the data and makes quality evaluation
- Databank content can be openly accessed and analysed

	PC	PE	PG	PS	chol	CL	GM1	PIP	PA	DAG
PC	X	X	X	X	X					
PE		X								
PG			X							
PS				X						
chol										
CL										
GM1										
PIP										
PA										
DAG										

NMRlipids databank
expected applications

- Force field evaluation: *What is the best force field for my application?*
- Reference simulations: *For example, reference pure bilayer simulations for membrane-protein interaction studies.*
- Analysis of bilayer properties from large datasets: *For example, calculate P-N vector angle from all available PC and PG simulations.*
- Exercise and example for sharing simulation data: *“PDB” for simulations?*

NMRlipids databank
general properties

- Overlay databank: *NMRlipids databank contains indexed links to the data. The actual MD simulation data is currently in Zenodo, but could be in any stable location.*

- Quality evaluation: *NMRlipids databank contains a quality evaluation protocol that is applied to all contributed datasets. Also the quality evaluation results are also stored in the databank.*

- Analysis of the data: *NMRlipids databank enables flexible analysis of the content.*

NMRlipids databank
Current status

- Prototype:

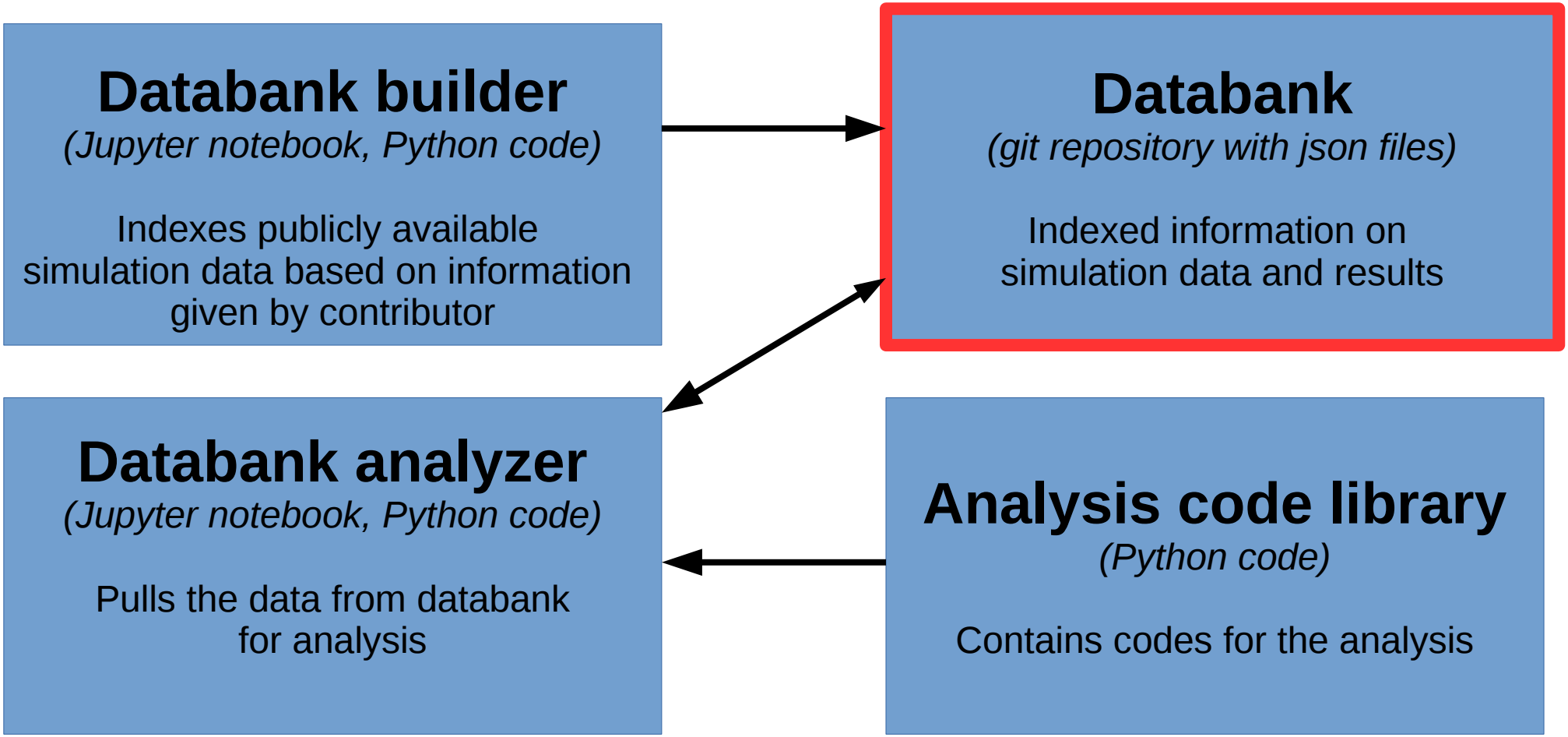
  www.nmrlipids.fi
  https://github.com/NMRLipids/MATCH/tree/master/Data/Lipid_Bilayers

- Indexing with folder names (not very flexible), adding data and analysis not very well documented

- New indexing system designed in the NMRlipids19 workshop in Berlin is now being implemented (currently mainly by Anne Kiirikki)

NMRlipids databank

New structure



NMRlipids databank

Databank builder

Current version (jupyter notebook):
https://github.com/NMRLipids/NMRlipidsIVPEandPG/blob/master/scripts/AddData.ipynb

- User gives information in the first shell:
  - DOI
  - def file for order parameter calculation
  - System name, software, force field name, source, and date
  - File names (xtc, tpr, gro or corresponding information)
  - Names of the molecules
- Program reads from tpr/gro file:
  - Amount of each molecule
  - Temperature
- This information is written into a json file that is stored into the databank

NMRlipids databank

New structure

Demo version with few datasets (json files, git):
https://github.com/NMRLipids/NMRlipidsIVPEandPG/tree/master/Data/Simulations

- Information from Databank Builder stored in dictionaries in json files. For example see:

  https://github.com/NMRLipids/NMRlipidsIVPEandPG/blob/master/Data/Simulations/0d5/d1d/0d5d1dcb43e775faf4e53c4f9ff255a67481bd38/9b487701b24d3fad83991e311188b08d3d5ea768/README.json

- Folder structure (see https://github.com/NMRLipids/NMRlipidsVIpolarizableFFs/issues/3):

  '/asd/fgh/topology_checksum/trajectory_checksum/README.json',

  where 'asd' and 'fgh' are the 1-3rd and 4-6th letters of the topology checksum.

NMRlipids databank

Databank analyzer

Current version (jupyter notebook):
https://github.com/NMRLipids/NMRlipidsIVPEandPG/blob/master/scripts/readDATA.ipynb

- Goes through the *Databank* and reads the dictionary files
- The data with selected lipids and atoms can be then analyzed calling functions in the *Analysis code library*

Urgent issues

- What should we store in the dictionary (readme.json)?

  - File names and locations, Force field and software, number of molecules, temperature

  - Something else?

- We need unique names for lipid molecules and the corresponding names in the simulation given by the user

  - CURRENT IDEA: We have a list of default names for molecules. If these do not correspond the simulation, contributor has to correct.

- We need unique naming convention for atoms within the molecules

  - For now, we are planning to use the mapping file (http://nmrlipids.blogspot.com/2015/03/mapping-scheme-for-lipid-atom-names-for.html)

  - Should we go more towards Sundaralingam notation?

  - Or should go for SMIRKS or similar (dx.doi.org/10.26434/chemrxiv.8304578.v1)?

Other issues

- Temperature, pressure, thermostat, etc. parameters cannot be read using Mdanalysis or Mdtraj. For Gromacs, the solution is to use gmx dump, but other programs I do not know.

- In which extend we should save simulation parameters (thermostats, timestep, PME settings, etc.) in the dictionary?

- What properties will be automatically analyzed? Order parameters, Form factors, all dihedrals angles, something else?

- Which results will be stored in the databank? Only automatically analyzed or all?