Shapespyer: a versatile Python-driven toolchain to model soft matter

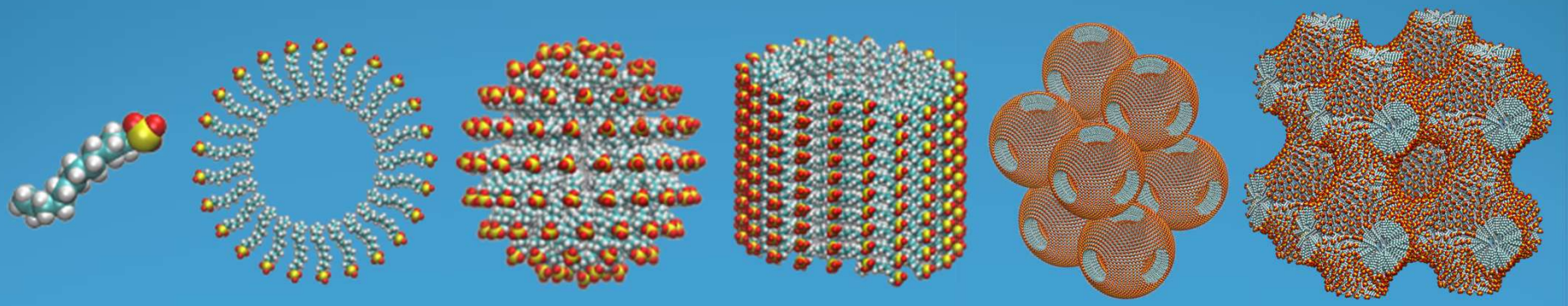
- ALC Project -

Andrey Brukhno,

Michael Seaton, John Purton, Valeria Losasso

– Scientific Computing, STFC/UKRI, Daresbury

Tim Snow – *Diamond,* **James Doutch** – *ISIS, RAL, Harwell*



https://www.scd.stfc.ac.uk/Pages/Shapespyer.aspx

https://gitlab.com/simnavi/shapespyer

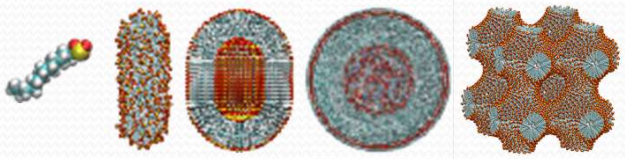
SHAPESPYER: a Python-driven framework for soft matter simulations

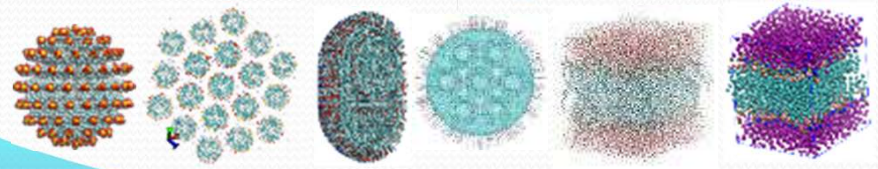
Andrey Brukhno, **Michael Seaton, John Purton, Valeria Losasso**, *SCD, STFC/UKRI,*

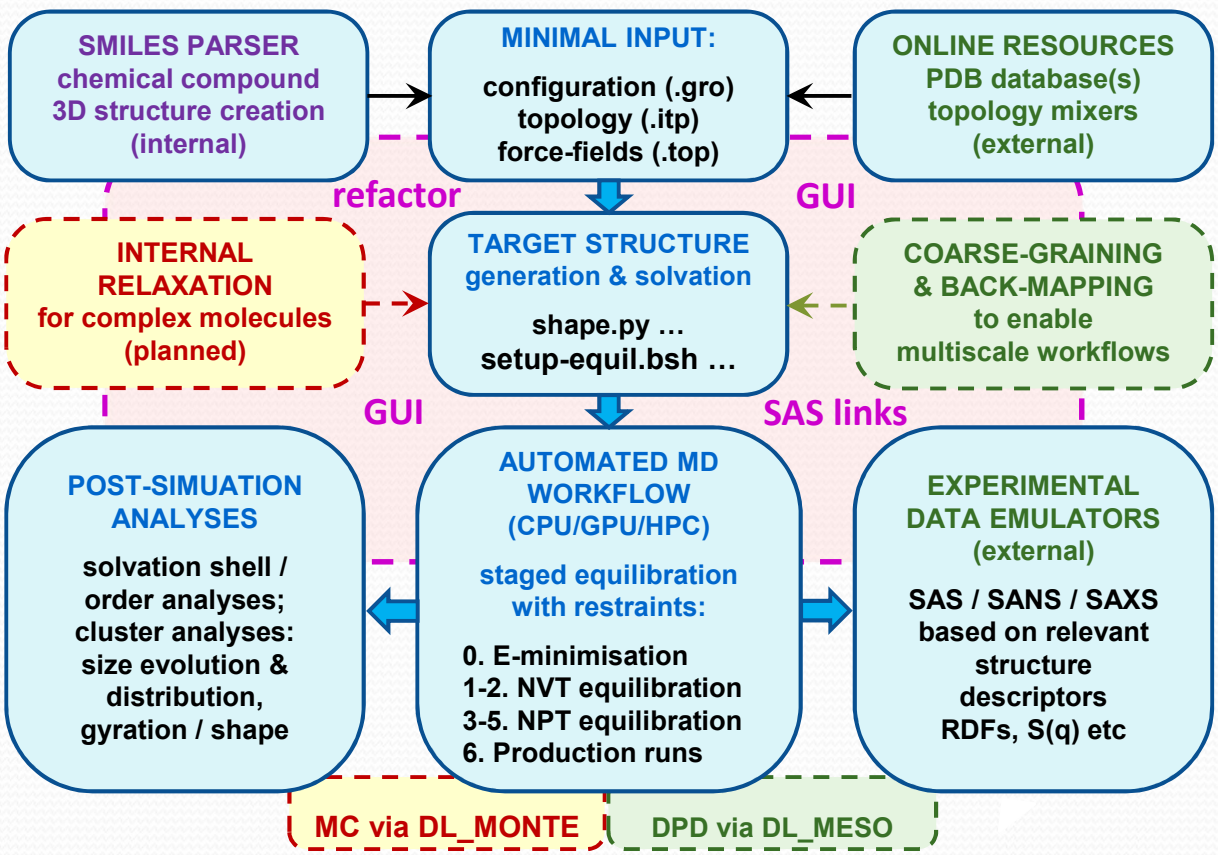
Tim Snow, *Diamond*, **and James Doutch,** *ISIS, STFC/UKRI*

Project overview

- **Generation, simulation & analyses**
  of pre-equilibrated molecular structures
  for functional soft matter research
- **Target structures / trajectories**
  for academic and industrial R&D;
  Linking with SANS, SAXS experiments
- **Provide automated workflows**
  Python library, APIs, Bash scripts
- **Partners / collaborators:**
  Ales Kutsepau, *ESS, Denmark*
  Maggie Holme, *Chalmers, Sweden*
  Hanna Barriga, *Karolinska Institutet, Sweden*
  Tom Headen, *ISIS, UK – MuSSiC software*
  SasView Team, *International*







gitlab.com/simnavi/shapespyer

ada lovelace centre · Scientific Computing · ISIS Neutron and Muon Source · diamond

andrey.brukhno@stfc.ac.uk

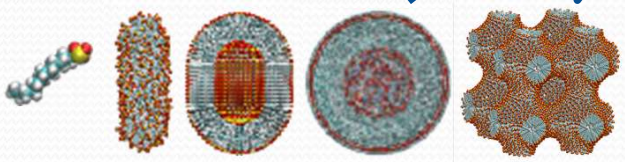
SHAPESPYER: a Python-driven framework for soft matter simulations

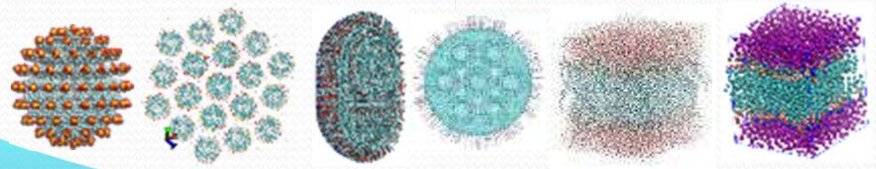
Andrey Brukhno, **Michael Seaton, John Purton, Valeria Losasso**, *SCD, STFC/UKRI,*

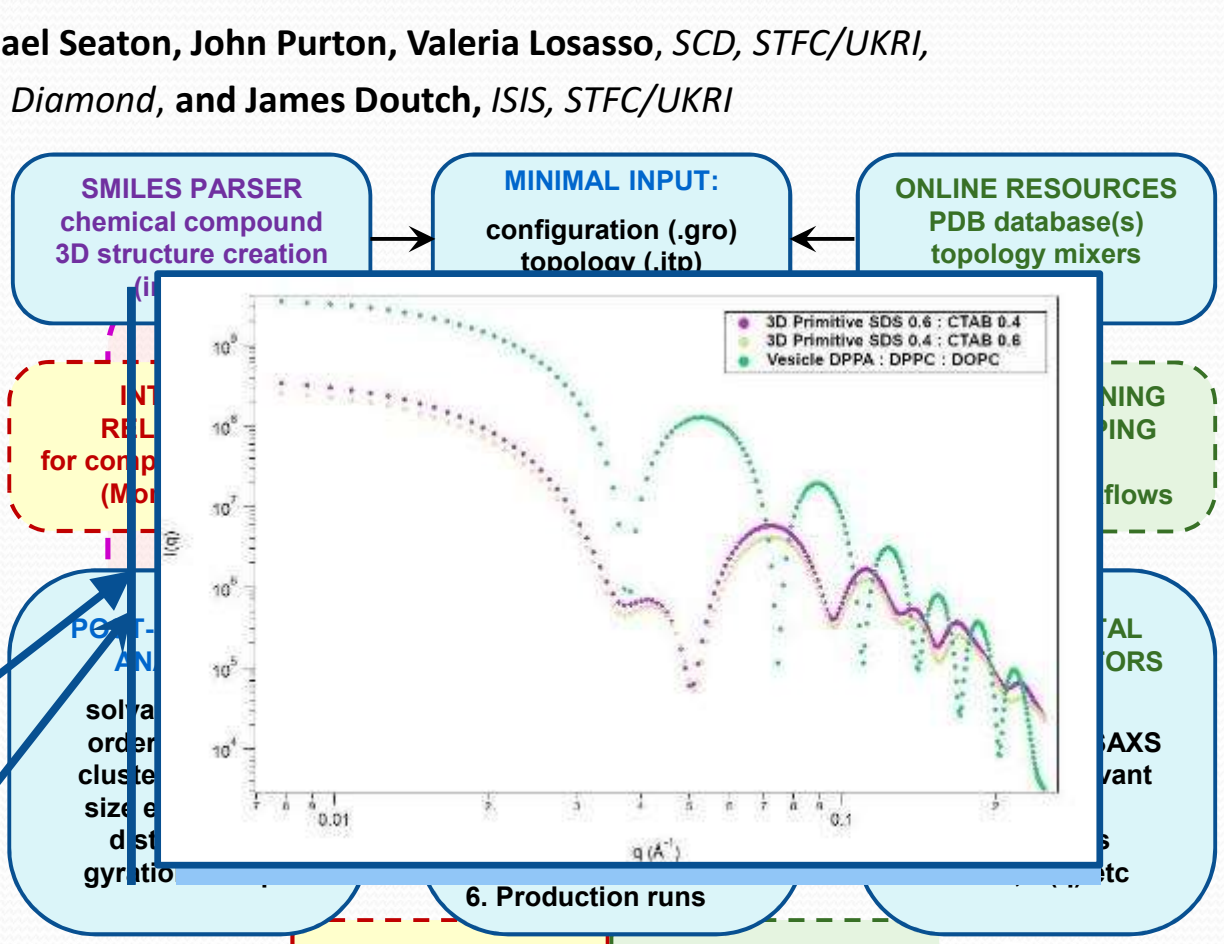
Tim Snow, *Diamond*, **and James Doutch,** *ISIS, STFC/UKRI*

Project overview

- **Generation, simulation & analyses**
  of pre-equilibrated molecular structures
  for functional soft matter research
- **Target structures / trajectories**
  for academic and industrial R&D;
  Linking with SANS, SAXS experiments
- **Provide automated workflows**
  Python library, APIs, Bash scripts
- **Partners / collaborators:**
  Ales Kutsepau, *ESS, Denmark*
  Maggie Holme, *Chalmers, Sweden*
  Hanna Barriga, *Karolinska Institutet, Sweden*
  Tom Headen, *ISIS, UK – MuSSiC software*
  SasView Team, *International*







gitlab.com/simnavi/shapespyer

ada lovelace centre · Scientific Computing · ISIS Neutron and Muon Source · diamond

andrey.brukhno@stfc.ac.uk

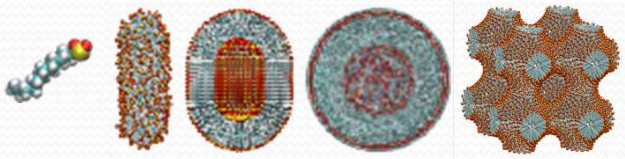
SHAPESPYER: a Python-driven framework for soft matter simulations

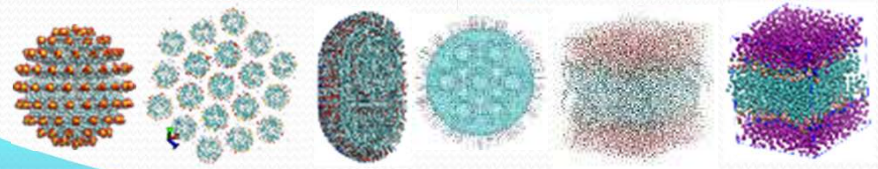
Andrey Brukhno, **Michael Seaton, John Purton, Valeria Losasso**, *SCD, STFC/UKRI,*

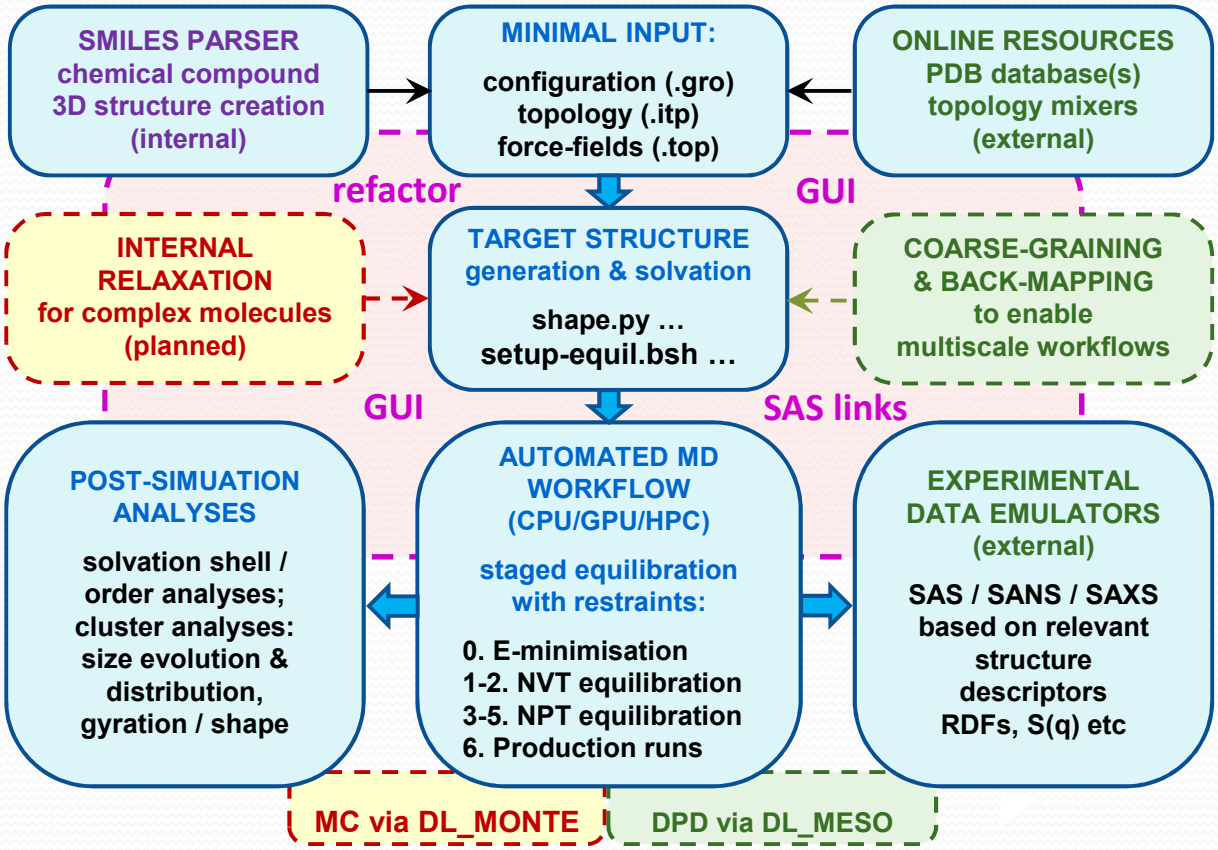
Tim Snow, *Diamond*, **and James Doutch,** *ISIS, STFC/UKRI*

Project roadmap (2024-25)

- **Scientific Computing Dept / ISIS (UK)**
  MC / DPD workflows - ***A Brukhno / M Seaton***
  Topol. / FFs – ***M Demir / S Beck / C Yong***
  Mono- & Bi-layer / NAMD – ***V Losasso***
  Links to SasView – ***A Brukhno / T Snow***
- **Chalmers (Sweden) / ESS (Denmark)**
  GUI via easyScience – ***A Kutsepau***
  SMILES module refactor => sub-package
  Links to SANS, SAXS experiments
- **Partners / collaborators:**
  Maggie Holme, *Chalmers, Sweden*
  Hanna Barriga, *Karolinska Institutet, Sweden*
  Tom Headen, *ISIS, UK – MuSSiC software*
  SasView Team, *International*







www.scd.stfc.ac.uk/Pages/Shapespyer.aspx
gitlab.com/simnavi/shapespyer

ada lovelace centre · Scientific Computing · ISIS Neutron and Muon Source · diamond

andrey.brukhno@stfc.ac.uk

Project overview

- **What is it about?**

  Generation and analyses of pre-equilibrated molecular nano-structures (soft matter systems)

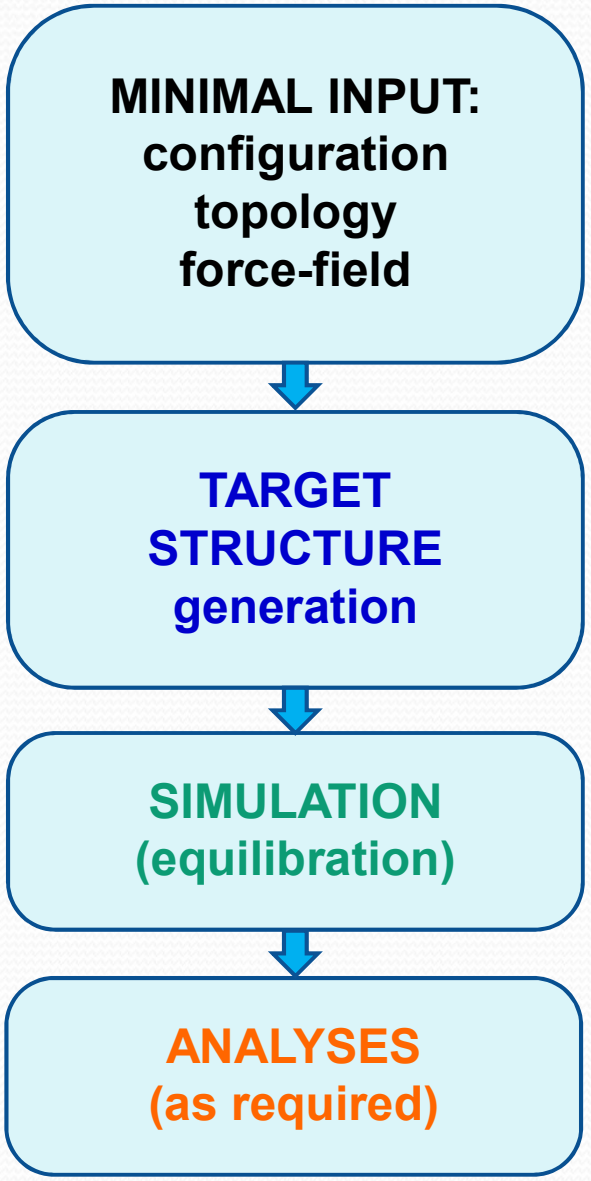
- **Objectives:**

  Create and deploy a semi-automated workflow: Python package / APIs & Bash scripts

- **Scope:**

  Providing simulation trajectories => reference structures for SAS (SANS, SAXS) & other scattering techniques and associated analyses

- **Partners:**
  - **Hanna Barriga**, *Karolinska Institutet, Sweden*
  - **Margaret Holme**, *Chalmers, Sweden*
  - **Lorna Dugan**, *University of Leeds, UK*
  - **Jian Lu**, *University of Manchester, UK*
  - **Tom Headen, H. Bindu Kolli**, *Disordered Materials, ISIS, UK*



Workflow via Python (+lib) & bash scripts

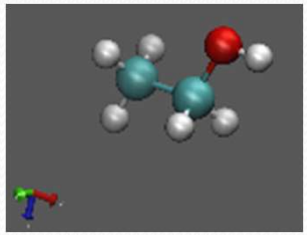
- **Core scripts for semi-automated preparation and running of simulations**
  - **shape.py** – generates a variety of structures, starting with a single molecule input (SMILES, configuration / topology)
  - **gmx-setup-equil.bsh** – adds ionic species, solvates the structure and creates the topology / force-field input for the entire system
  - **gmx-equilibrate.bsh** – runs equilibration in a few stages: energy minimisation (0), NVT (1, 2) / NPT (3 - 5) with *gradually relieved position (shape) restrains*, it also allows to proceed with subsequent production runs
- **Core utilities for automated analysis in parallel workflows via task-farming**
  - **ana-gyration.bsh** – gyration analysis & principal moments of inertia (PCA)
  - **ana-clusters.bsh** – cluster size, number and distribution analysis (a few scripts)
  - **gmx-ana-solvation.py** – solvent, atom groups content in a spherical cavity
  - **shapespyer/scripts/** – Gromacs, NAMD and DL_MESO (DPD) workflows

Molecule chemical structure from SMILES

SMILES string parser => no need to search through databases

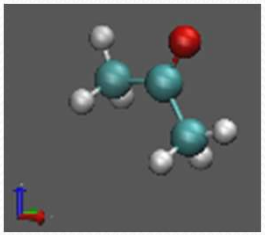
Shapespyer generates 3D coordinates for relatively simple molecules (rings**)

shape.py … -i 'smiles.sml' --shape='smiles' -r 'ETHANOL'



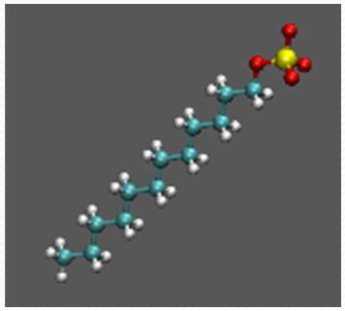
CCO

Ethanol



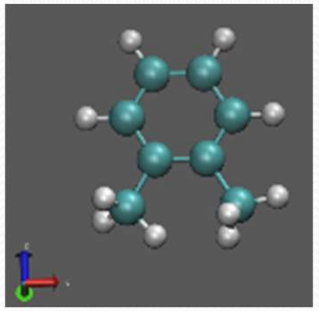
CC(=O)C

Acetone



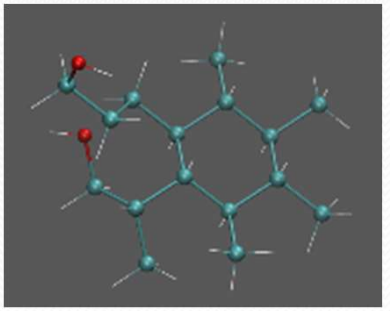
CC{11}OS(=O)(=O)[O-]

SDS



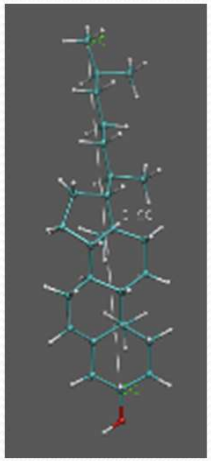
c1(C)c(C)cccc1

Orthoxylene



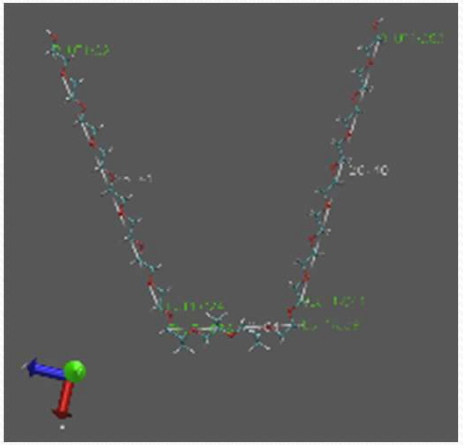
CC1C(R1)C(R2)C(R3)C(R4)C1C

Cyclohexane**



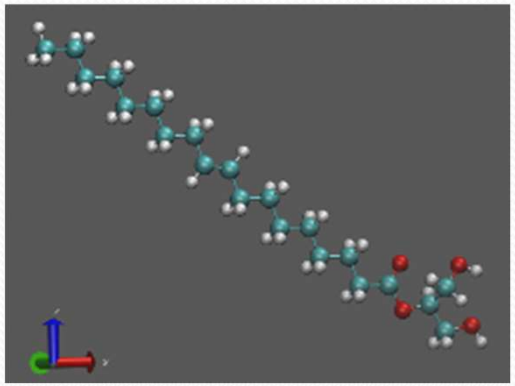
Cholesterol**:

[C27H46O]



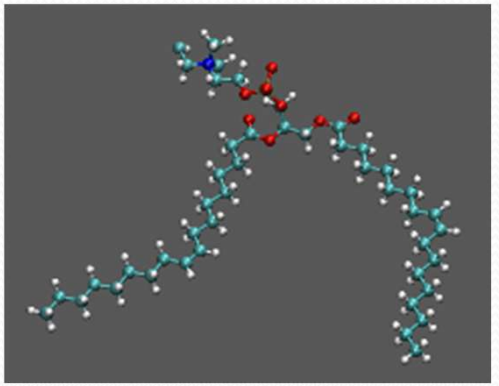
Pluronic P123*:

PEO{n}-PPO[m]-PEO[n]



Monoolein:

[C21H40O4]



DOPC lipid:

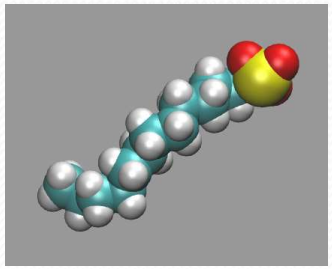
[C44H84NO8P]

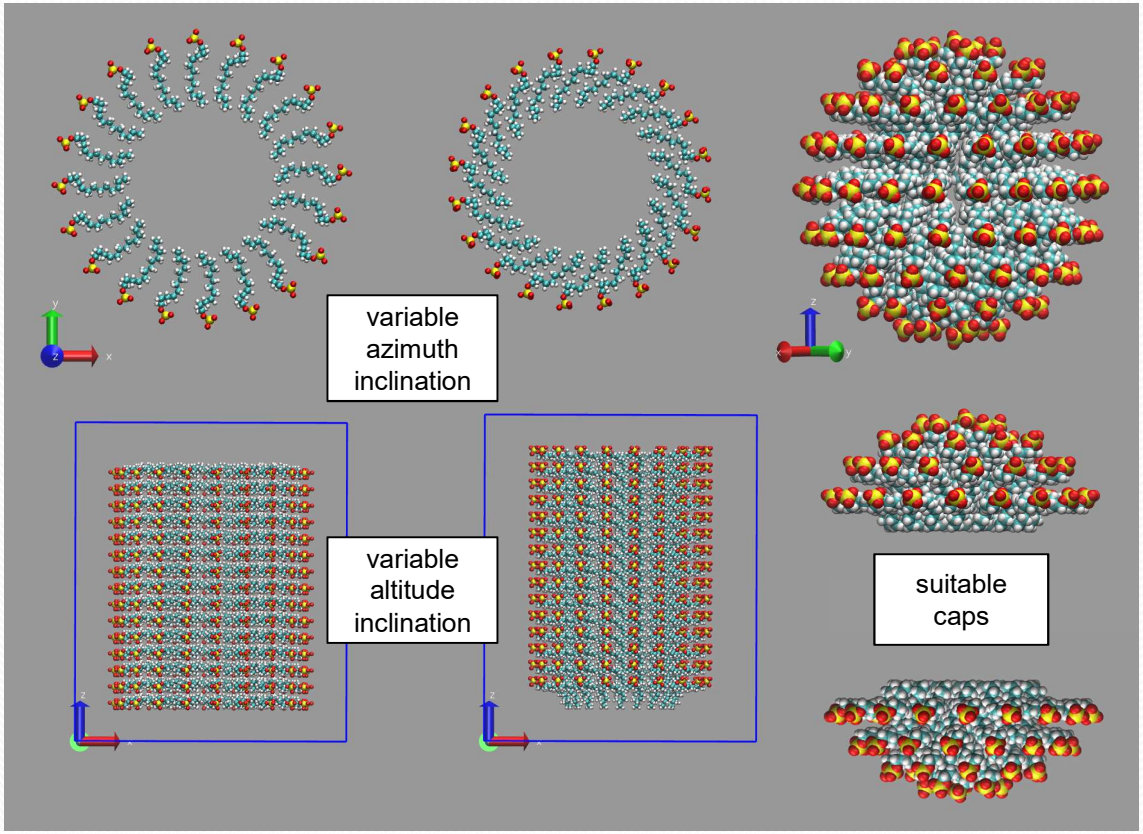
Workflow

Example: SDS

sodium-dodecyl-sulphate

- the most ubiquitous surfactant / detergent





INPUTS:
template molecule(s)
config.pdb / .gro / .xyz /
smiles.sml

force-fields
CHARMM / OPLS / Gromos
topology.top, molecule.itp

shape.py
ring (disc)
ball / ves. (sphere / micelle)
rod (cylinder / stack)
lattice (cubic, FCC, hex.)

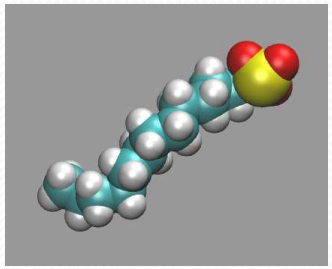
gmx-setup-equil.bsh
add ionic species & water
overall topology for system

gmx-equilibrate.bsh
runs equilibration
5 stages (1ns) → production

Workflow

Example: SDS

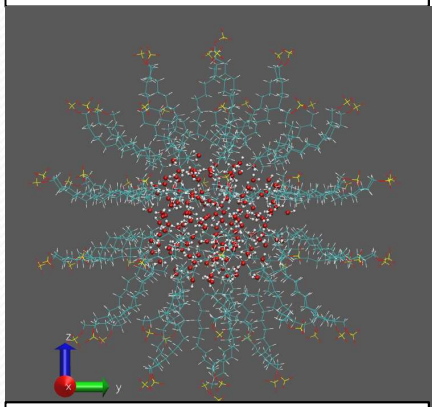
sodium-dodecyl-sulphate



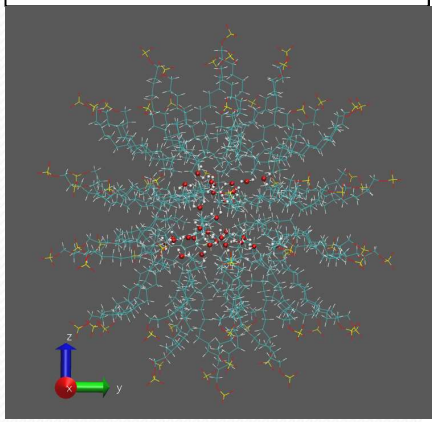
ana-solvation.py

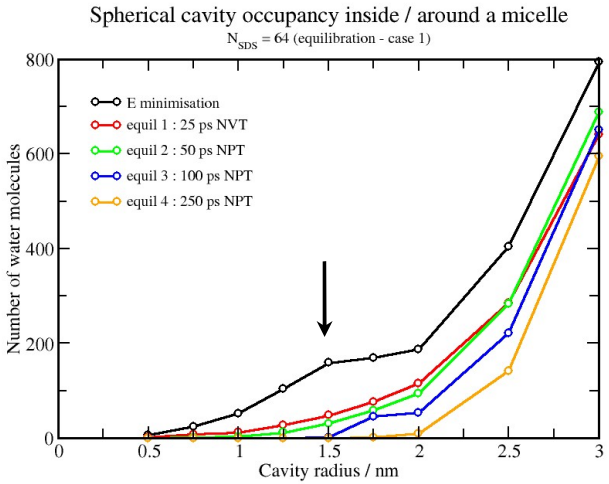
solvent content analysis
within a given radius or shell

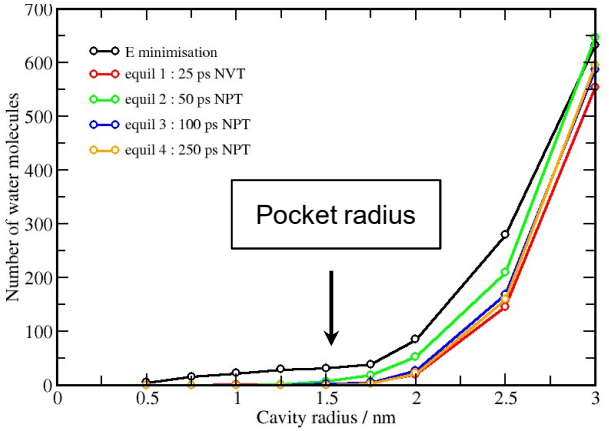
Initial structure 1 (64 mols)
water content



Initial structure 2 (64 mols)
water content



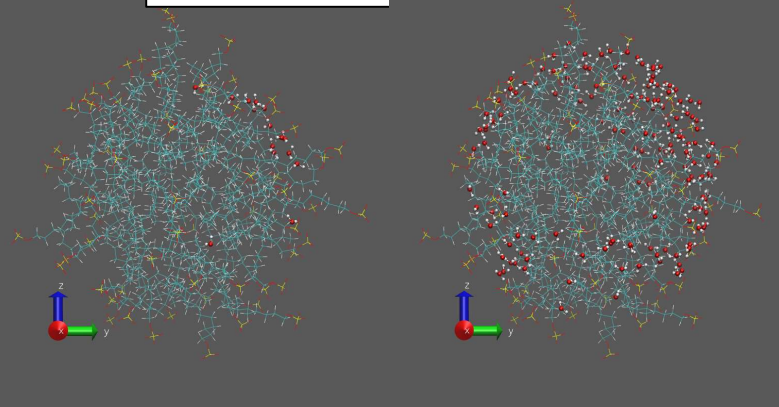




R = 2 nm

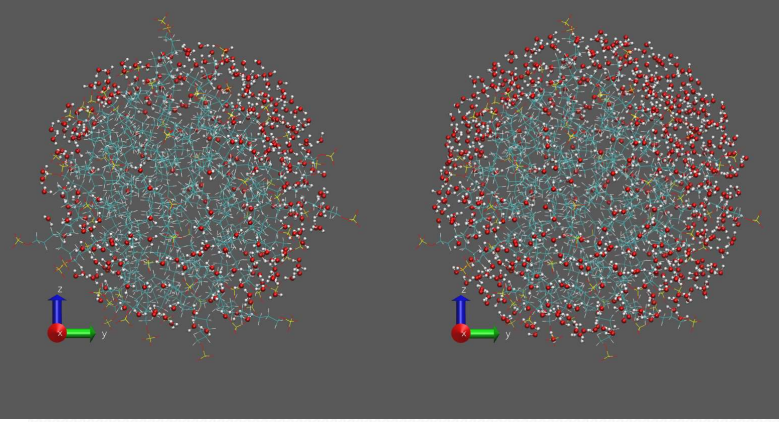
Eq: 5-th stage

R = 2.6 nm



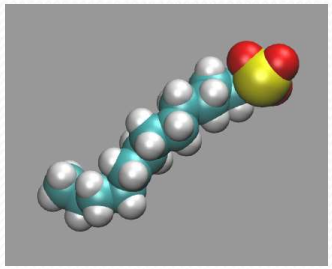
R = 2.8 nm

R = 3.0 nm



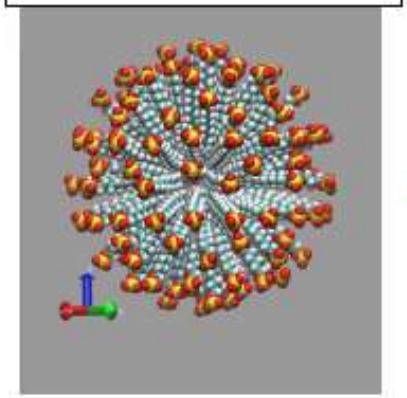
Workflow

Example: SDS
sodium-dodecyl-sulphate

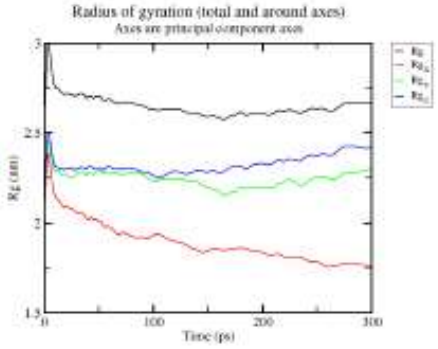


ana-gyration.bsh
radius of gyration analysis
around the 3 principal axes

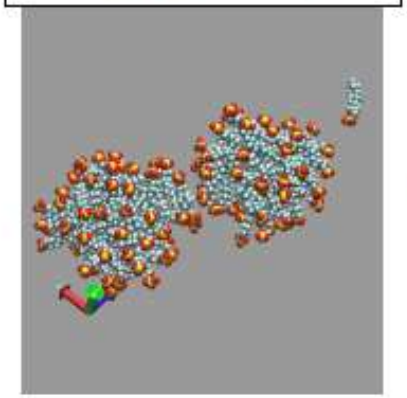
Initial structure with 126 mols



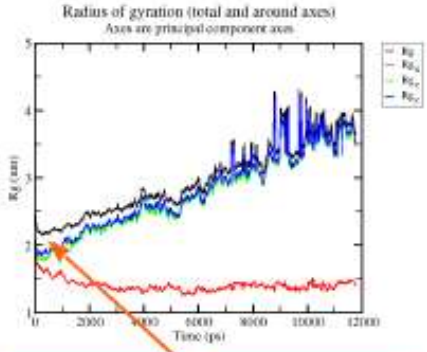
Equilibration: 300 ps



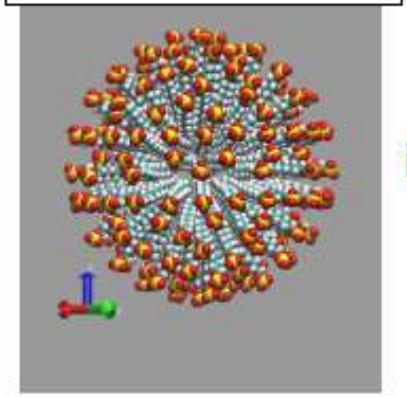
Equilibration: 735 ps



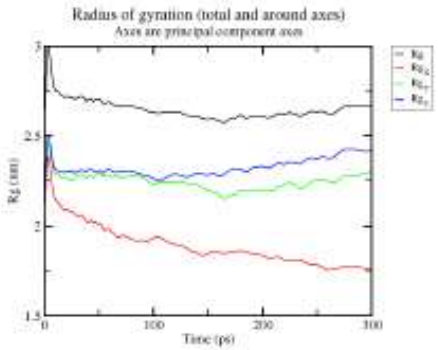
Production: 1 ns (2)



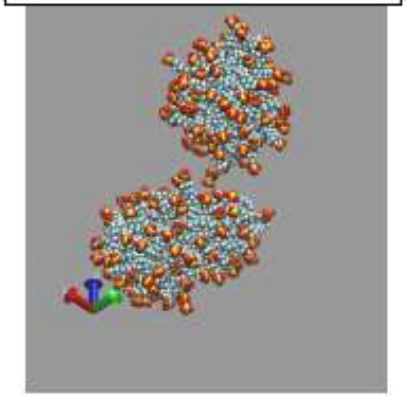
Initial structure with 180 mols



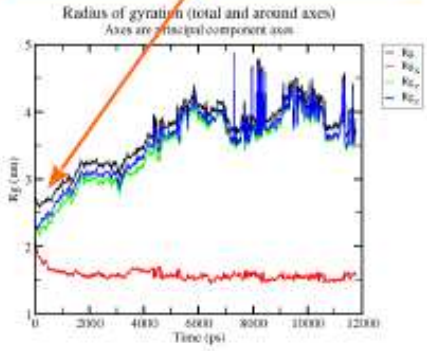
Equilibration: 300 ps



Equilibration: 435 ps

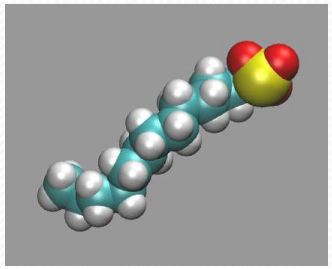


The SDS micelle splits!



Workflow

Example: SDS
sodium-dodecyl-sulphate



ana-gyration.bsh
radius of gyration analysis
around the 3 principal axes

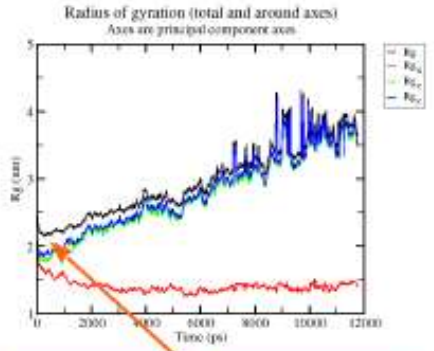
Initial stru

Initial stru

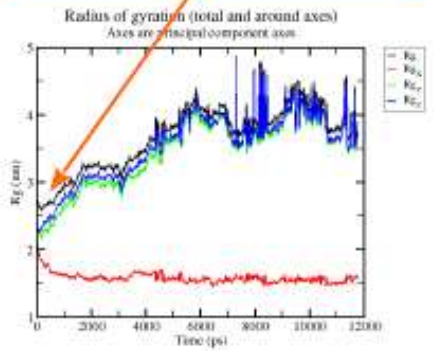
Equilibration: 735 ps

Equilibration: 435 ps

Production: 1 ns (2)

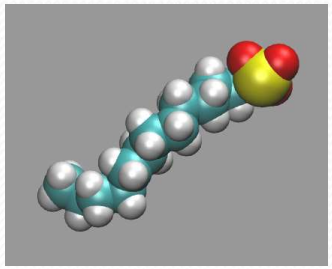


The SDS micelle splits!

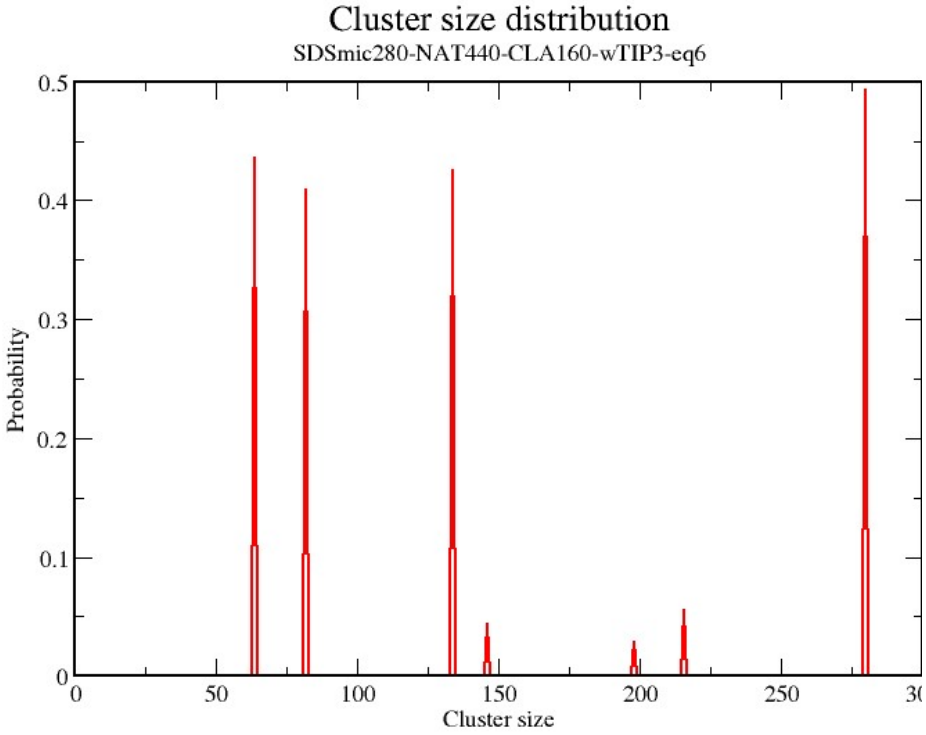


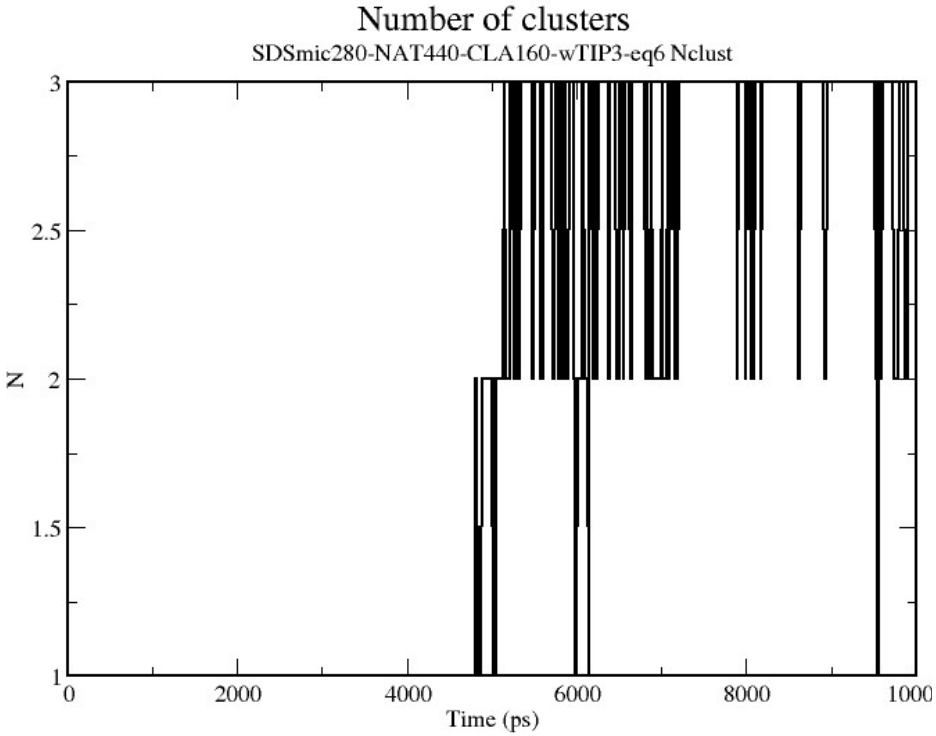
Workflow

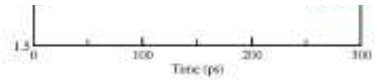
Example: SDS
sodium-dodecyl-sulphate

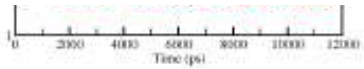


ana-clusters.bsh
cluster size and distribution analysis along a trajectory





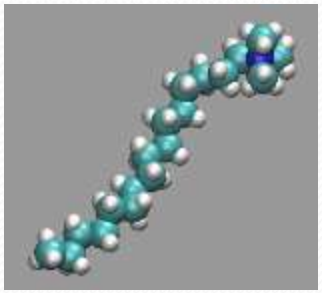




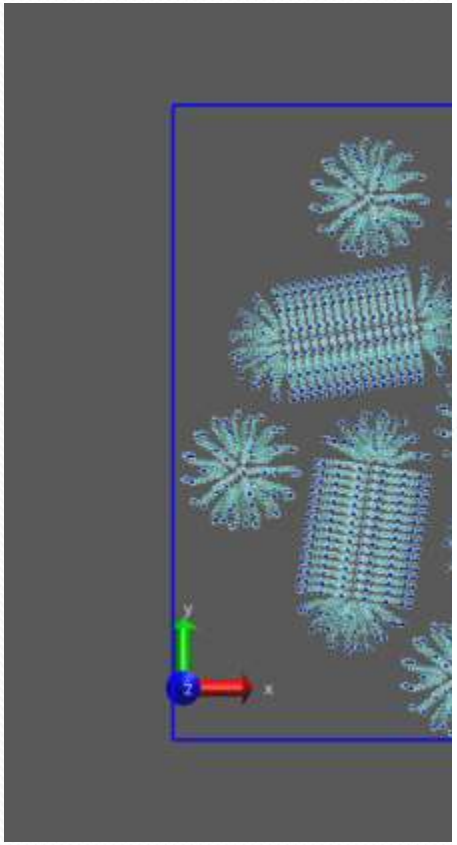
Workflow

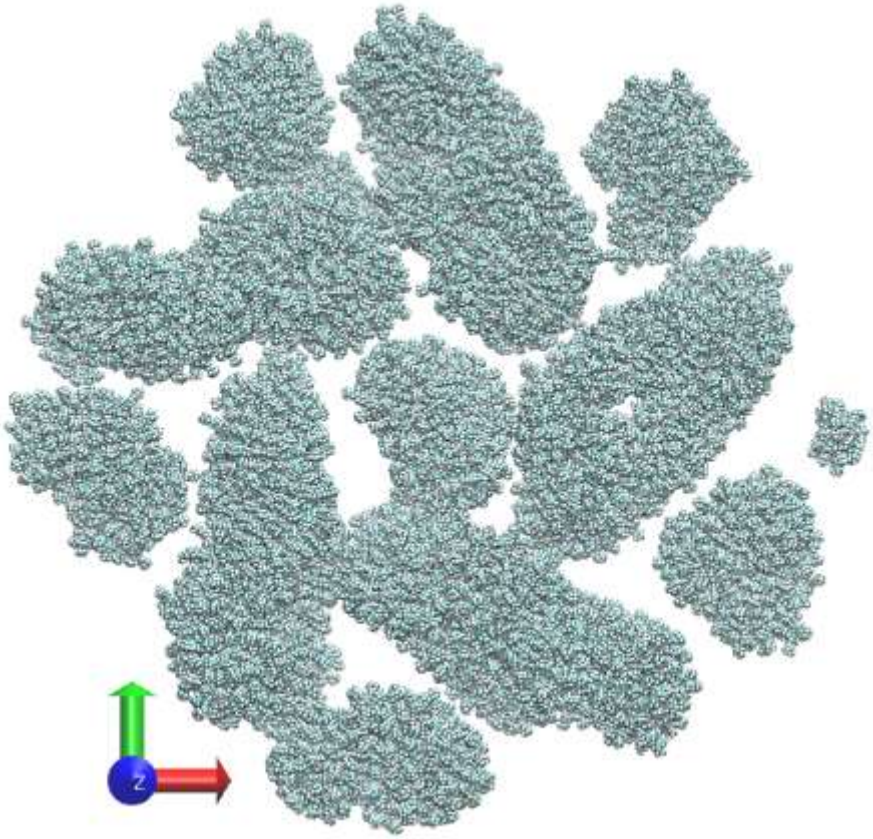
<u>Example</u>: CTAB 4528 mols

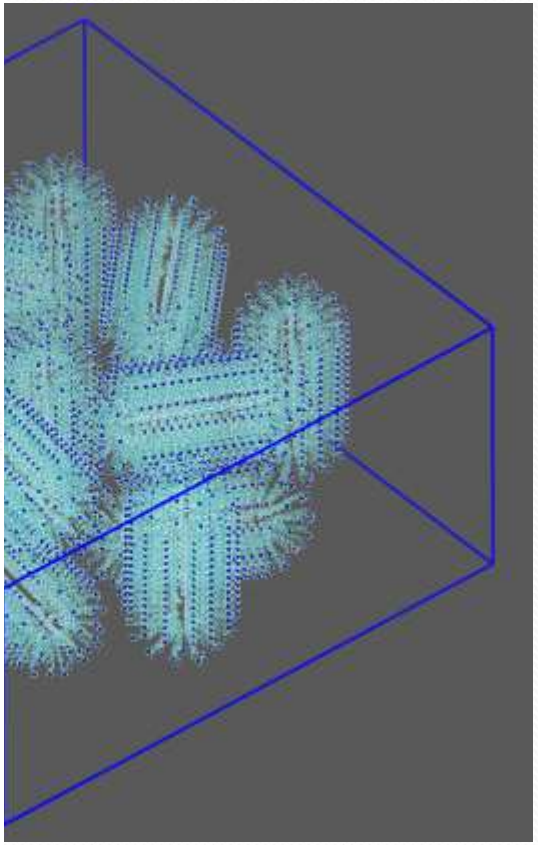
Cetyl-trimethyl-ammonium-bromide



ana-clusters.bsh
ana-gyration.bsh



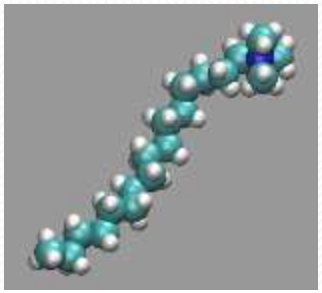




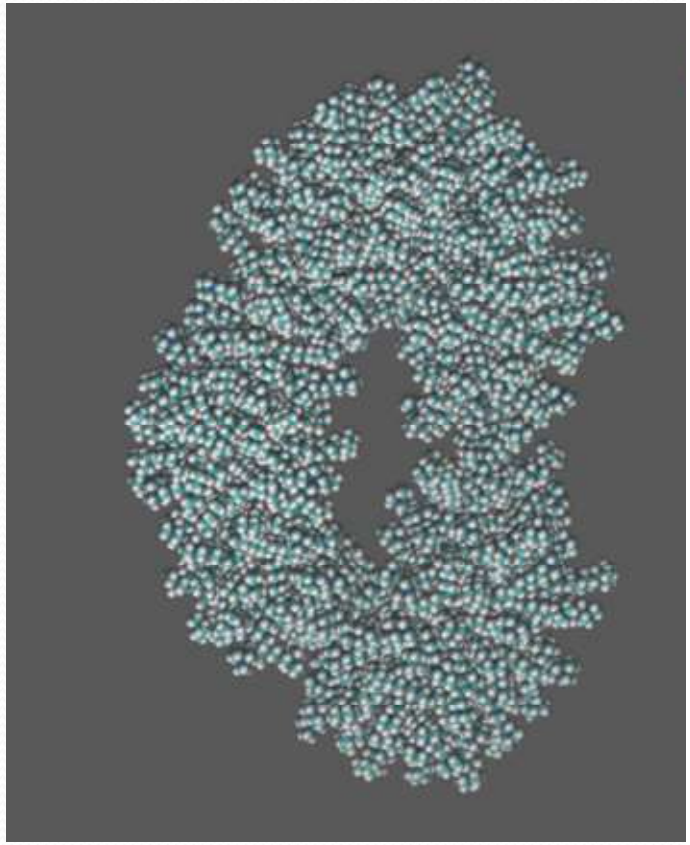
Workflow

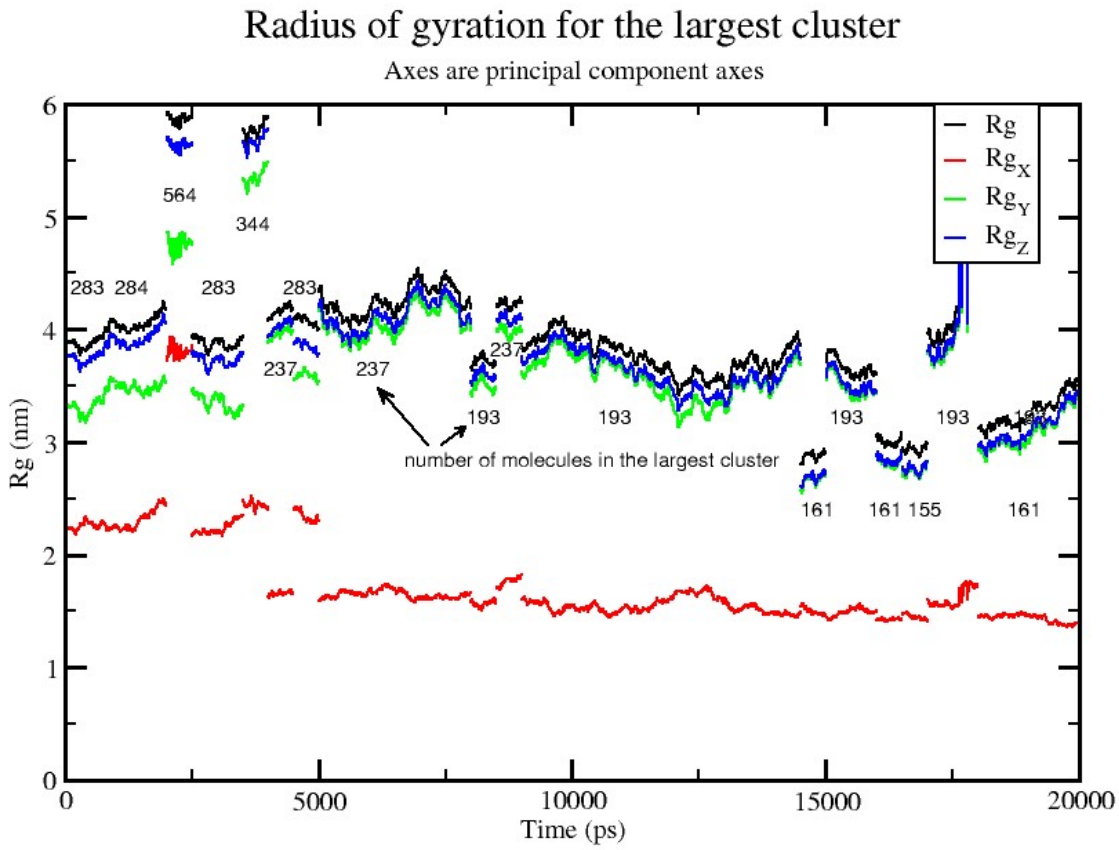
Example: CTAB 4528 mols

Cetyl-trimethyl-ammonium-bromide



ana-clusters.bsh
&
ana-gyration.bsh

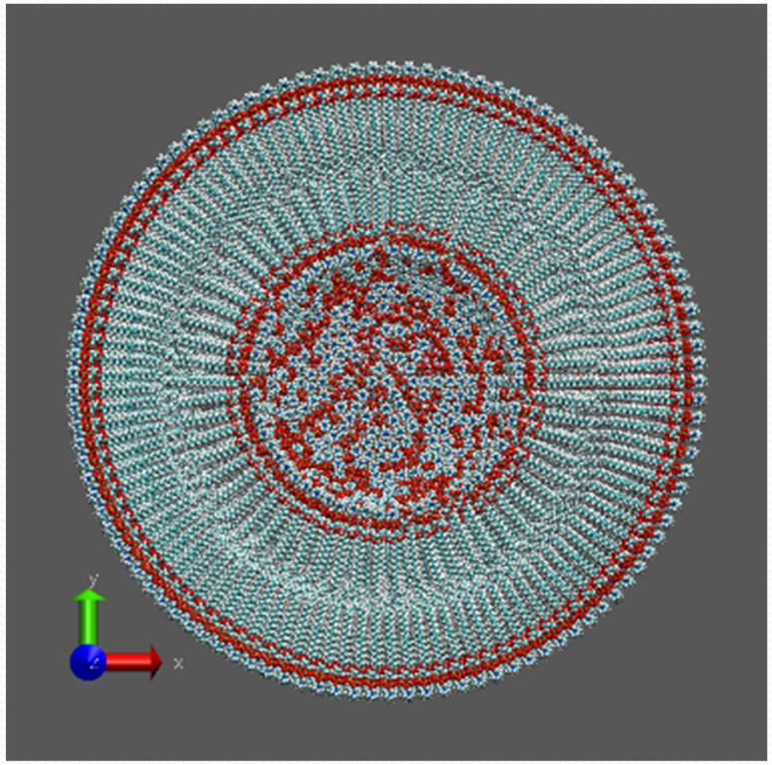


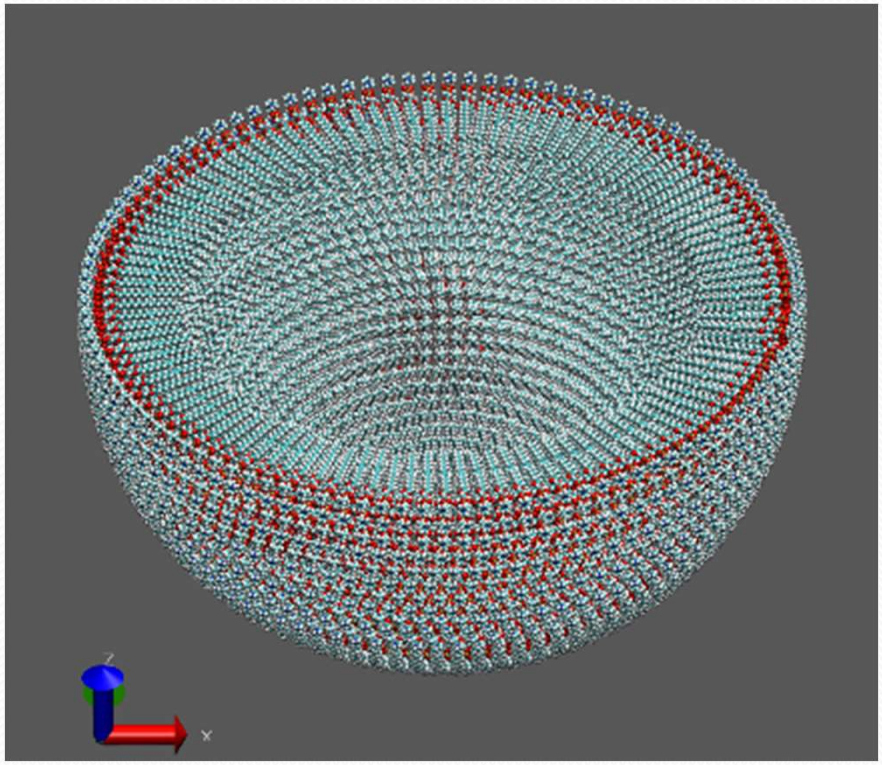


Spherical vesicles

$[DPPA\ 0.4\ /\ DPPC\ 0.6]_{int}$: $[DPPC\ 0.3\ /\ DOPC\ 0.7]_{ext}$

shape.py … --shape='ves' –frc='40,60,0:0,30,70'

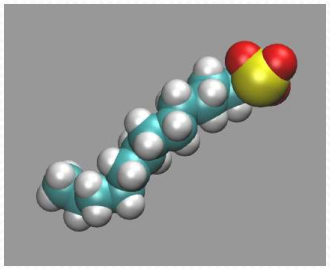


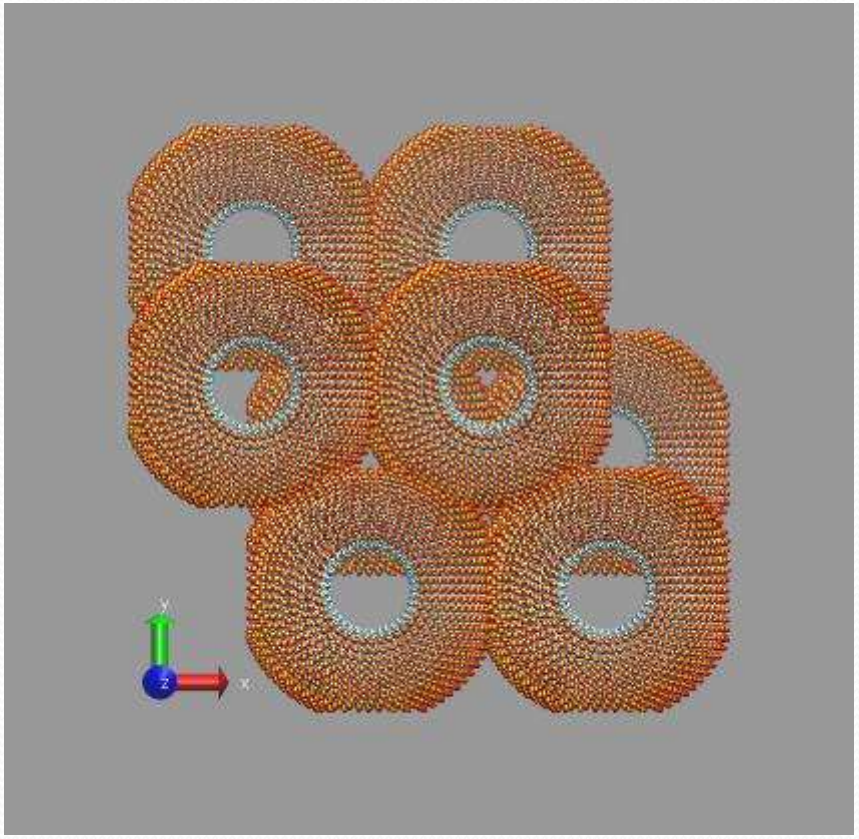


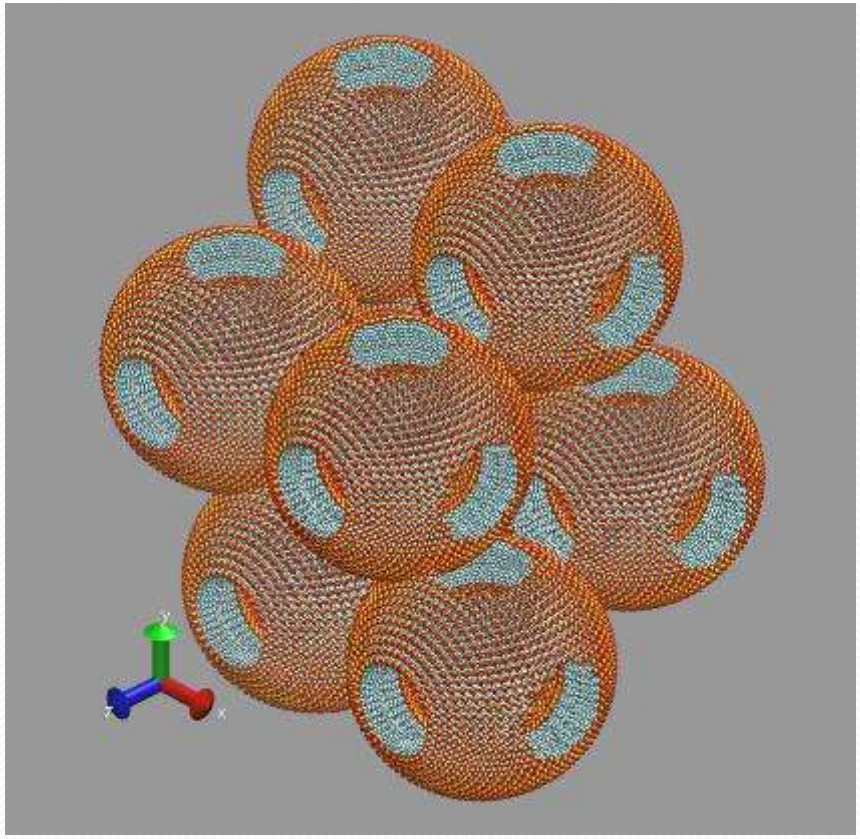
Nano-aggregates on a lattice

Simple Cubic, FCC, Hexagonal

shape.py … --nx # --ny # --nz #





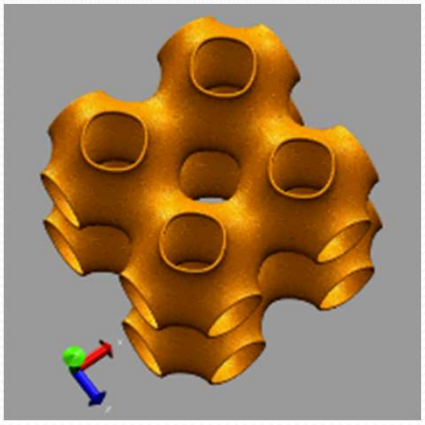


Nanostructures on Schwarz minimal surfaces

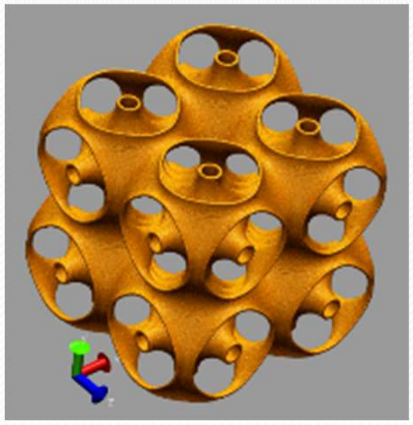
3D surfaces with zero average curvature => minimum surface tension

Shapespyer can generate any analytically defined surface: F(x,y,z) = const

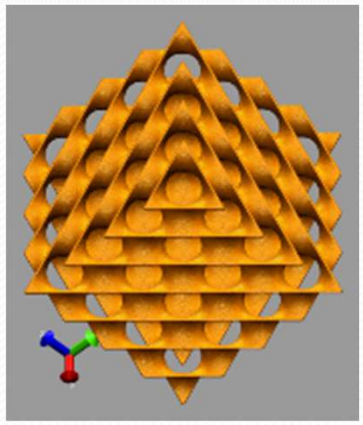
surface3d.py …



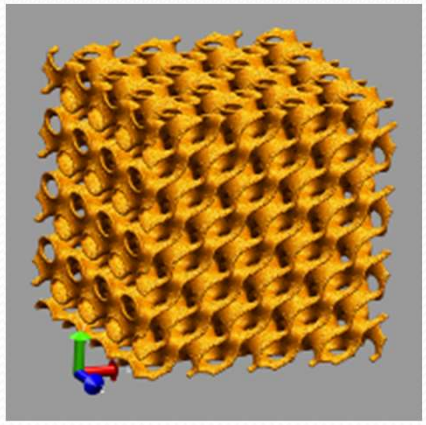
Primitive



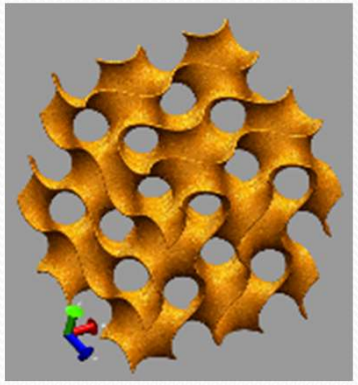
Double Primitive



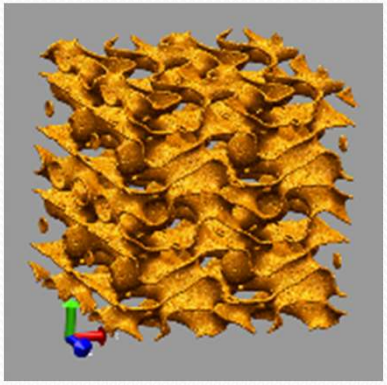
Diamond



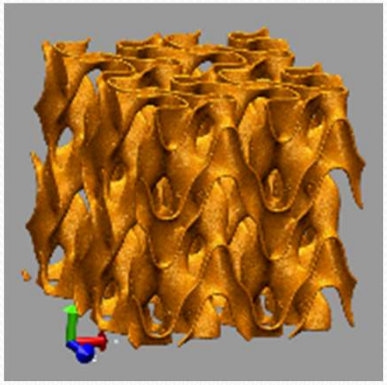
Double Diamond



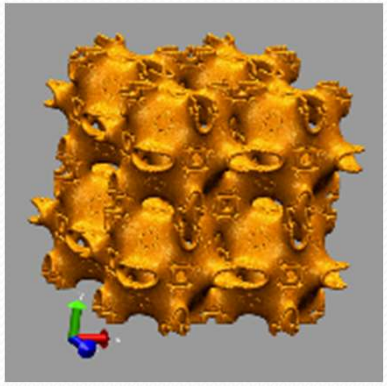
Gyroid



Double Gyroid

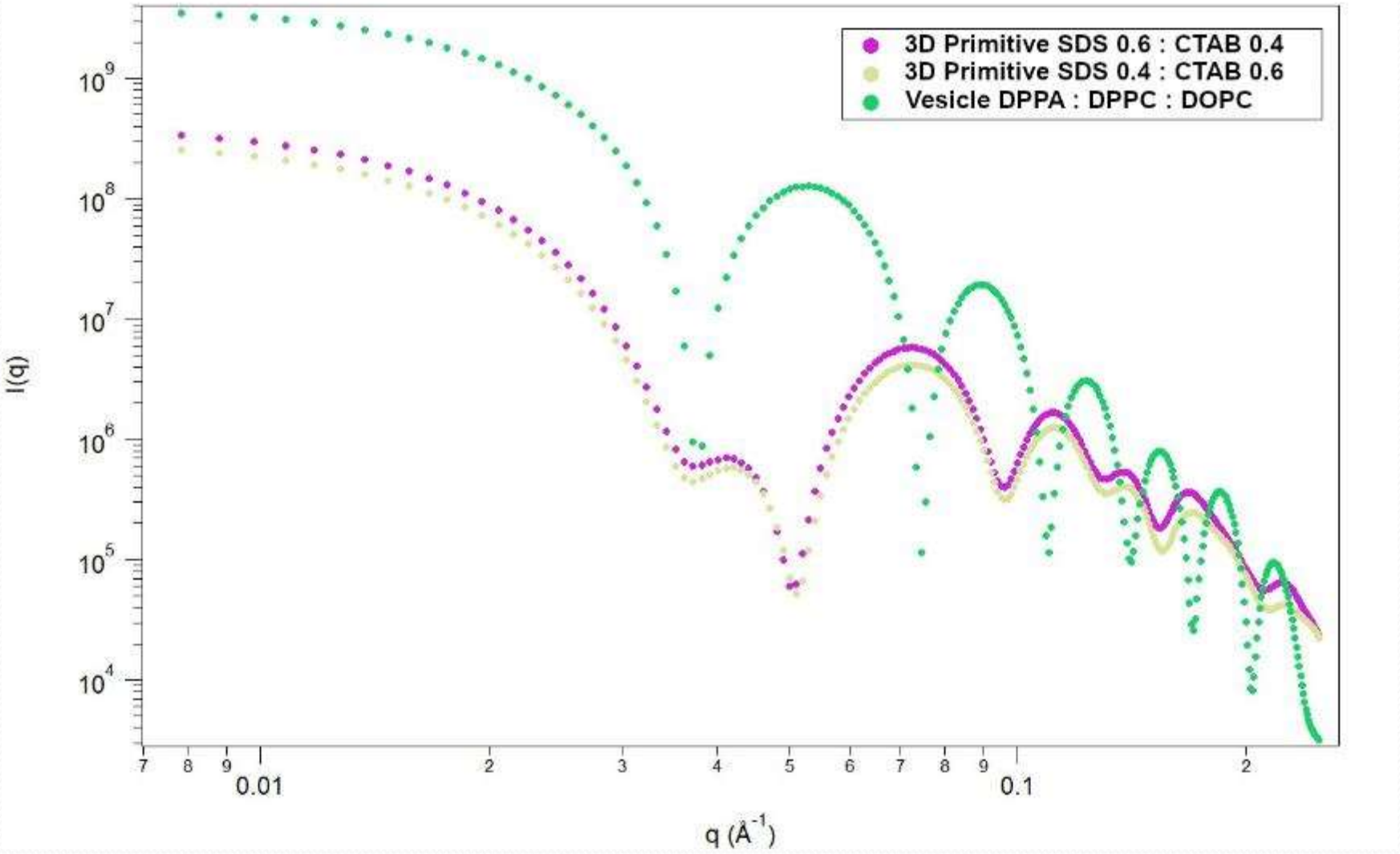


Fischer-Koch



Neovius

Nanostructures on Schwarz minimal surfaces

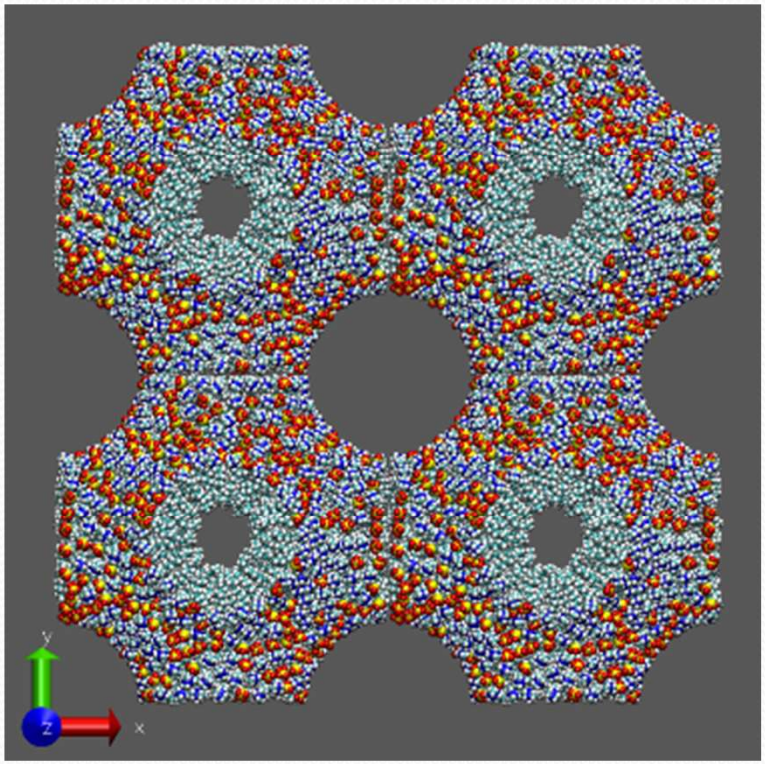


Nanostructures on Schwarz minimal surfaces

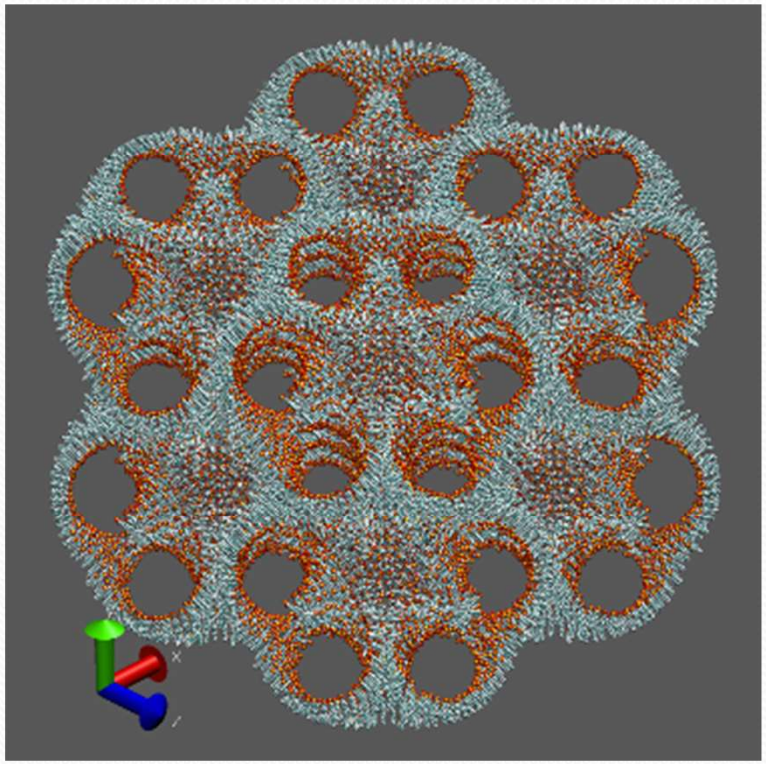
3D surfaces with zero average curvature => minimum surface tension

Shapespyer can generate any analytically defined surfaces: F(x,y,z) = const

surface3d.py …



Primitive: unit cell 10 nm



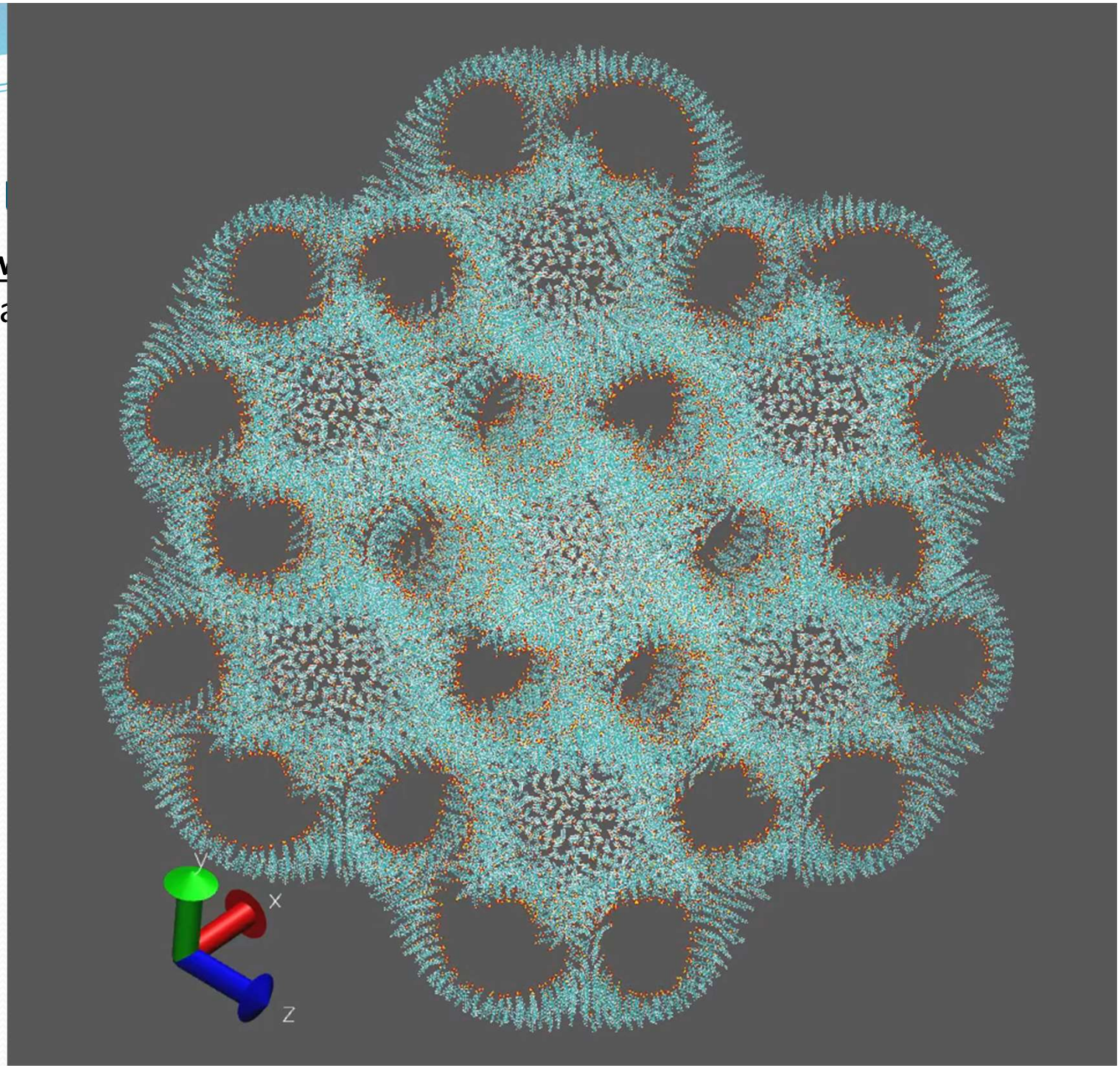
Double Primitive: unit cell 20 nm

Nanostr

3D surfaces w

Shapespyer ca

surface3d.py



Progress as of May 2025

- Generation and simulation of multicomponent nanostructures **√**
- Parsing SMILES strings for branched molecules **√**
- Assemblies on 3D lattices and minimal surfaces **√**
- Overarching Python framework: library of objects with APIs **√**
- Inputs as dictionaries via popular file format (Yaml, Json) **√**
- Jupyter notebooks with interactive visualization (NGL Viewer) for tutorials **√** …
- Coarse-graining functionality: Martini and DPD force-fields **√** …
- Molecule topology (bonds/angles) recognition **√** / *conversion* via *DL_FIELD* …
- *DL_MESO (DPD) / DL_POLY (MD) & NAMD workflows have been added* **√** …
- *Bespoke structures & simulations in response to users' requests …*
- *Relaxation engine for atomic groups & GUI being developed …*

https://www.scd.stfc.ac.uk/Pages/Shapespyer.aspx **https://gitlab.com/simnavi/shapespyer**