

ENSANE

PRESENTATION OVERVIEW

Improvements and downsides

Program workflow

How to use:

- Command syntax
- General commands
- Proteins
- Membranes/leaflets
 - Lipid definitions
- Solvations
 - Importing structures

Examples:

- Normal system
- Complex system
- Flooding system
- Abstract system

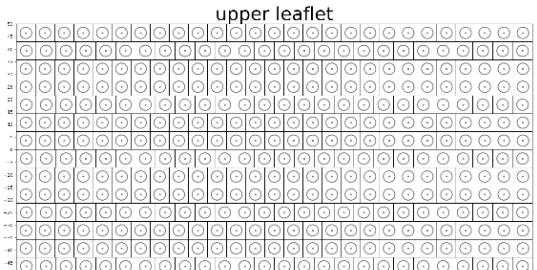
How to access

- GitHub

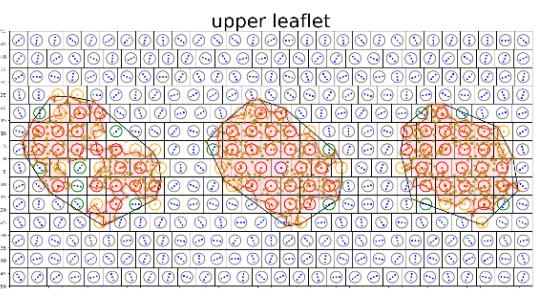
How it works

- Not presenting these slides

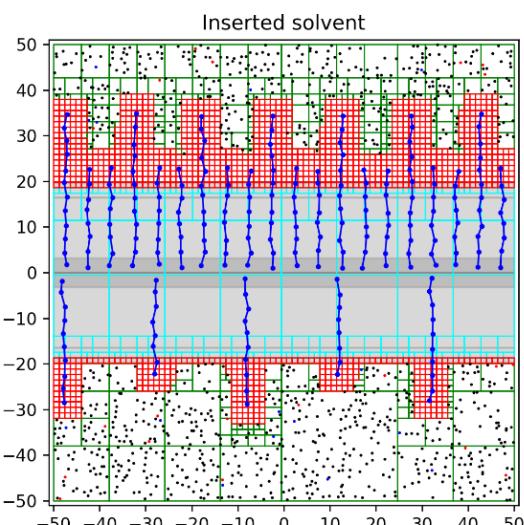
IMPROVEMENTS



Accurate number of lipids in each leaflet



More accurate lipid ratios in each leaflet



Better solvation algorithm

Ability to import structures to be used as "solvent"

'Unlimited' number of membranes, proteins and solvations

DOWNSIDES

Restricted to cubic (rectangular) systems

Slightly more complex to give commands

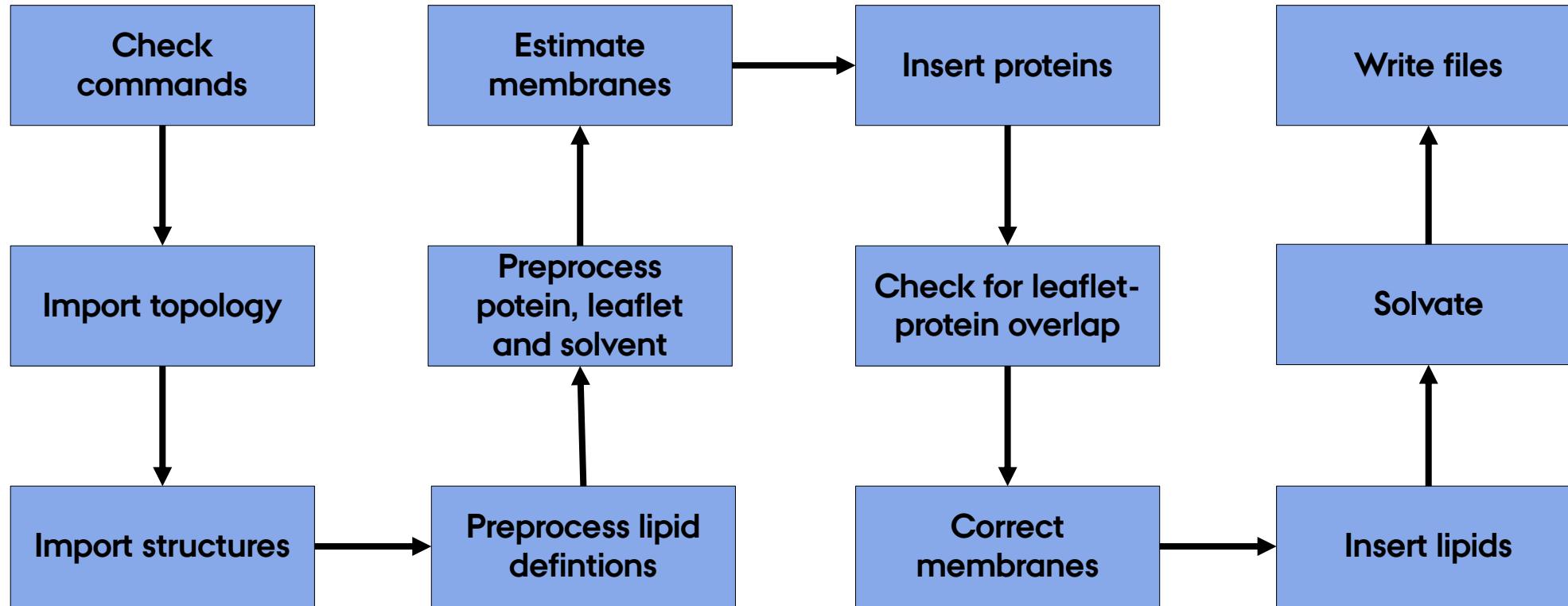
Slightly more complex lipid definitions for developers

Still a bit of spaghetti code in certain places

"Tech-debt" from early code that needs to be rewritten leading to odd restrictions

Documentation/help only partially written

PROGRAM WORKFLOW



HOW TO USE

COMMAND SYNTAX

`command_call` is simply a stand-in name

Script vs terminal

- Run from script : `ENSANE(command_call = "cmd")`
- Run from terminal: `python ENSANE.py -command_call cmd`

Multiple calls to the same command

Run from script : `command_call = ["cmd1", "cmd2"]`

Run from terminal: `-command_call cmd1 -command_call cmd2`

Alternative: (For some commands)

`command_call1 = "cmd1", command_call2 = "cmd2"`

Subcommands:

- `cmd = "SETTING:VAL1:VAL2"`
- `"mol_name:PROTEIN1:LIGAND1"`

Subcommands only affect the specific command call

GENERAL COMMANDS

Setting the pbc box size

- box = [10, 10, 10]
- x = 10, y = 10, z = 10

Setting the system default "force field"

- params = "default" (default)
- params = "dev18"
- params = "PhosV13"

System name

- sn = "Tutorial System"

Random seed

- rand = 5

Importing topology files:

- itp_input = "itps_for_ENSANE.itp"
- Takes charge-data directly from topology
- Understands "#include" statements and definitions

Writing structure file (.pdb/.gro)

- out = "output.pdb"

Writing topology file (.top)

- top_out = "topol.top"

Writing log file

- log_out = "log.log"

INSERTING PROTEINS

Used to insert structures

- protein = "Protein.pdb"
- prot = "Protein.pdb"

Protein names in topology files

- protein = "Protein.pdb **mol_names:PROTEIN1:LIGAND1:LIGAND1**"
- 1x charge for PROTEIN1
- 2x charge for LIGAND1

Can be moved and rotated

- protein = "Protein.pdb **tx:5 tz:3 ry:90**"

Center of structure

- "cen_method:**cog**" (default) (center of geometry)
- "cen_method:**axis**"
- "cen_method:**bead:beadnr**"
- "cen_method:**res:resnr**"
- "cen_method:**point:x:y:z**"

CREATING MEMBRANES

Used to create membranes

- membrane= "POPC:1"
- memb= "POPC:1"

Mebrane / leaflets

- "type:**bilayer**" (default) --> bilayer
- "type:**upper**" --> upper monolayer
- "type:**lower**" --> lower monolayer
- "type:**mono**" --> upper monolayer

Designating params

- membrane = "POPC:5 POPE:2.5 **CHOL:1:dev18** **params:PhosDev13**"

Size and centering

- box = [10, 10, 10], membrane = ["POPC:5 CHOL:1 **x:5 center:2.5:0:0**", "POPC:4 CHOL:2 **x:5 center:-2.5:0:0**"]



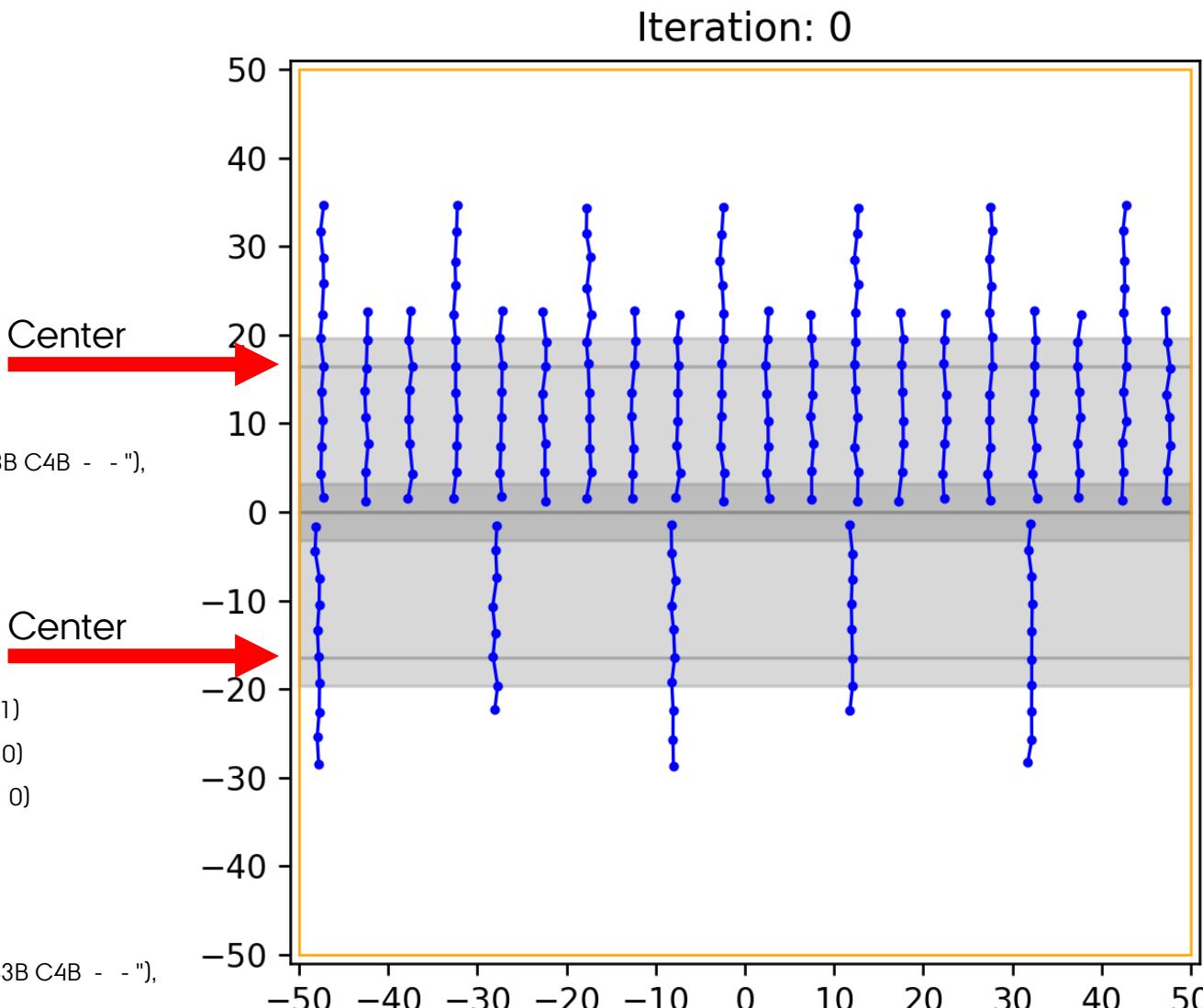
LIPID DEFINITIONS

Python2 insane.py

```
moltype = "lipid"
lipidsx[moltype] = ( 0, .5, 0, 0, .5, 0, 0, .5, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
lipidsy[moltype] = ( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
lipidsz[moltype] = ( 10, 9, 9, 8, 8, 7, 6, 6, 5, 4, 3, 2, 1, 0, 5, 4, 3, 2, 1, 0)
lipidsa.update({
    "POPC": (moltype, " - - - NC3 - PO4 GL1 GL2 C1A D2A C3A C4A - - C1B C2B C3B C4B - - "),
})
```

ENSANE.py

```
lipid_type, ff = "lipid", "M3"
lipid_defs[(lipid_type, ff)] = {}
lipid_defs[(lipid_type, ff)]["x"] = ( 0, .5, 0, 0, .5, 0, 0, .5, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
lipid_defs[(lipid_type, ff)]["y"] = ( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
lipid_defs[(lipid_type, ff)]["z"] = ( 10, 9, 9, 8, 8, 7, 6, 6, 5, 4, 3, 2, 1, 0, 5, 4, 3, 2, 1, 0)
lipid_defs[(lipid_type, ff)]["center"] = 7
lipid_defs[(lipid_type, ff)]["bd"] = (0.25, 0.25, 0.3)
lipid_defs[(lipid_type, ff)]["lipids"] = {
    ("POPC", "beads"): (" - - - NC3 - PO4 GL1 GL2 C1A D2A C3A C4A - - C1B C2B C3B C4B - - "),
}
```



SOLVATIONS

Used to place solvent in system

- solvation = "solv:W pos:NA neg:CL"
- solv = "solv:W pos:NA neg:CL"

Amount of solvent determined by molarity

- solvation = "solv:W pos:NA neg:CL **solv_molarity:55.56 salt_molarity:0.15**" (default values shown)

Ratios of different solvent

- solvation= "solv:**W:5** solv:**SW:2** pos:**NA:5** pos:**CA:1** neg:CL"

Place exact number of solvent molecules (flooding)

- solvation= ["solv:TRP **count:True** **solv_molarity:20**", "solv:W pos:NA neg:CL"]

IMPORT STRUCTURES

Import solutes

- `solute_input = ["ligands.pdb params:ligands", "lipids_in_solv.pdb params:lipids"]`

Use imported structure

- `solv = ["solv:LIG1 count:True solv_molarity:20 params:ligands", "solv:W pos:NA neg:CL"]`

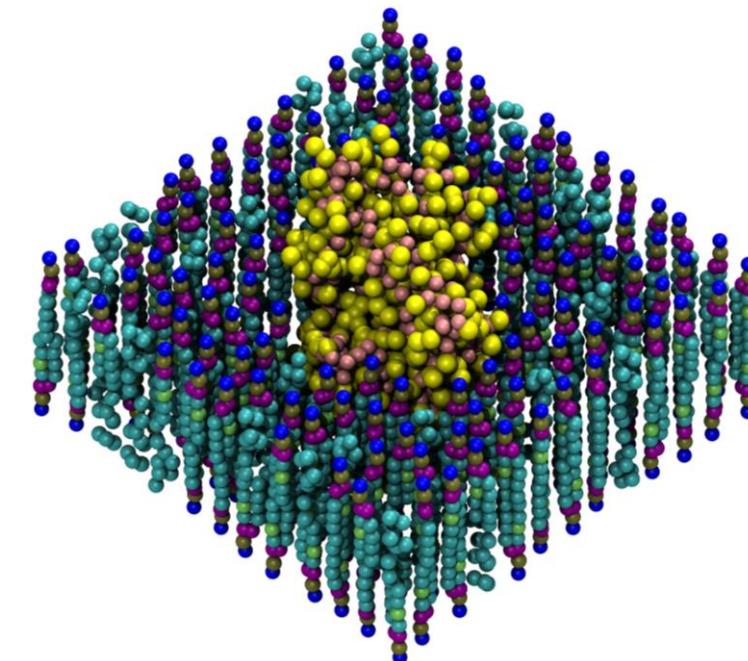
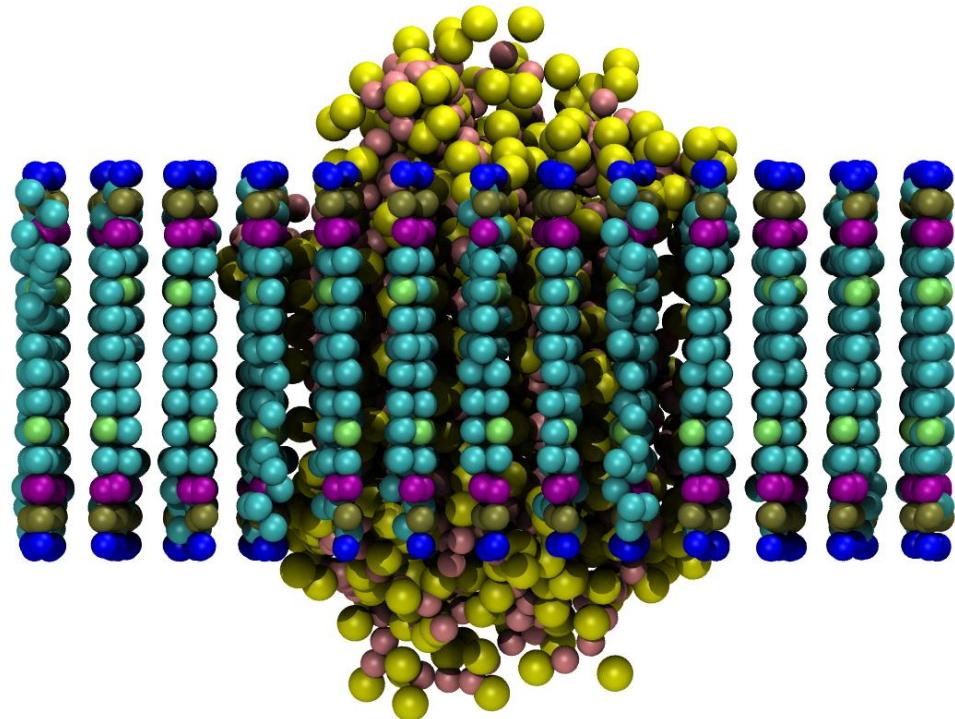
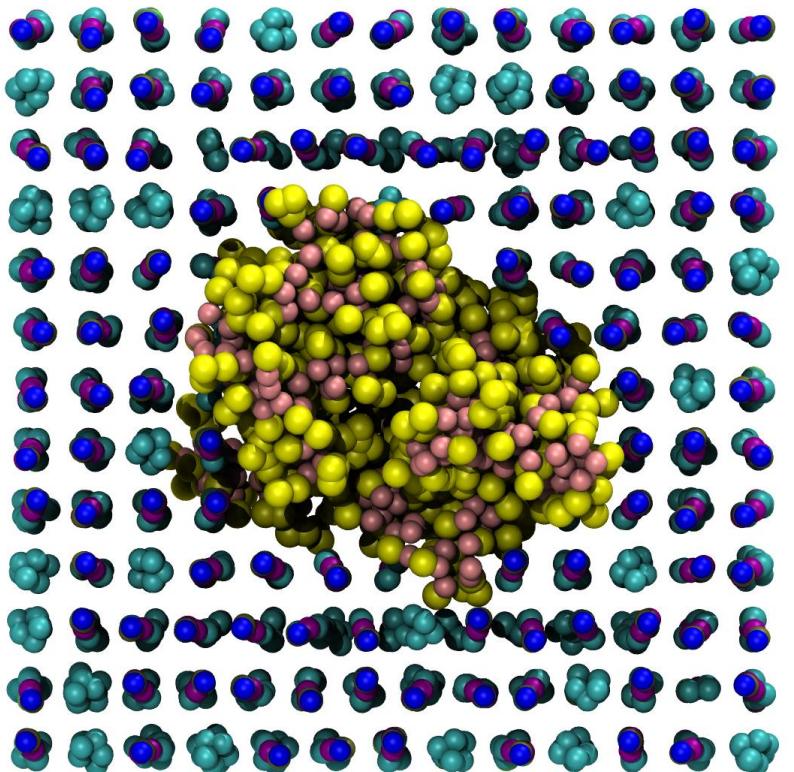
Currently restricted to 1-residue structures due to "tech-debt"

EXAMPLES

Using only syntax

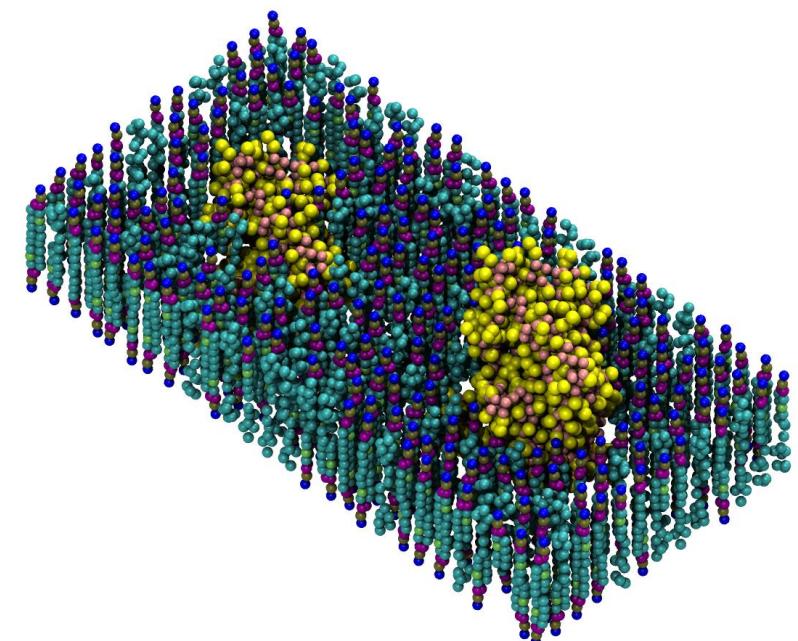
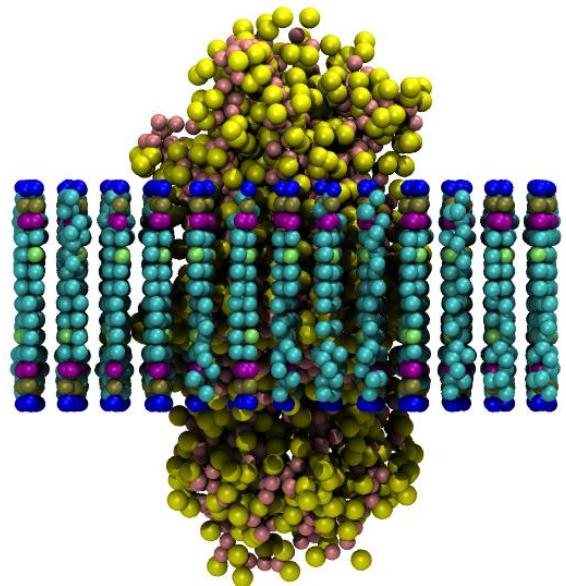
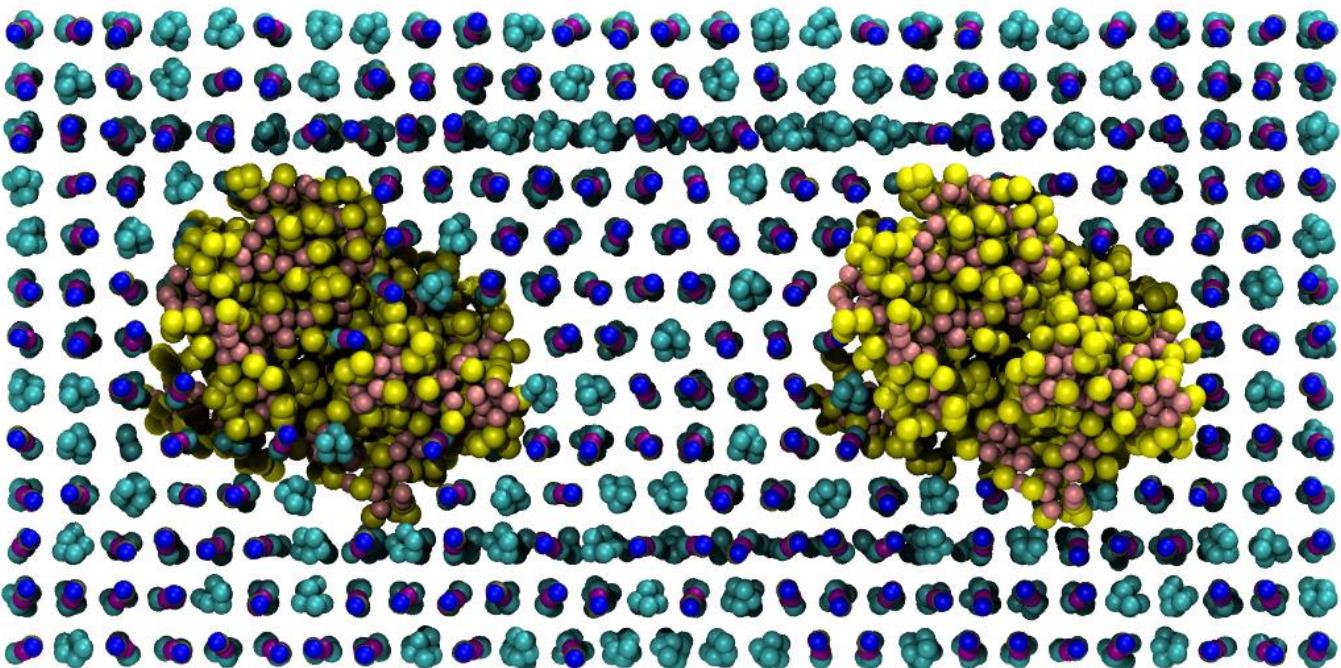
NORMAL

```
ENSANE(  
    pbc = [10, 10, 10],  
    prot = "04_output_martinize.pdb prot_name:AtSUC1_152ASP",  
    leaf = "type:bilayer POPC:10 POPE:4 CHOL:3",  
    imp_top = "Tutorial/top_for_ENSANE.top",  
    backup = False,  
    out = "Tutorial/example_systems/normal_system.pdb",  
    top = "Tutorial/example_systems/normal_topol.pdb",  
)
```



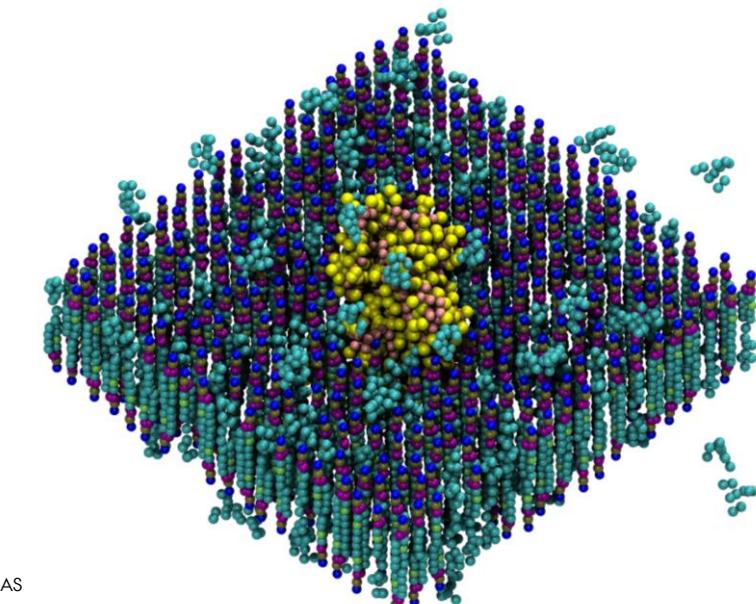
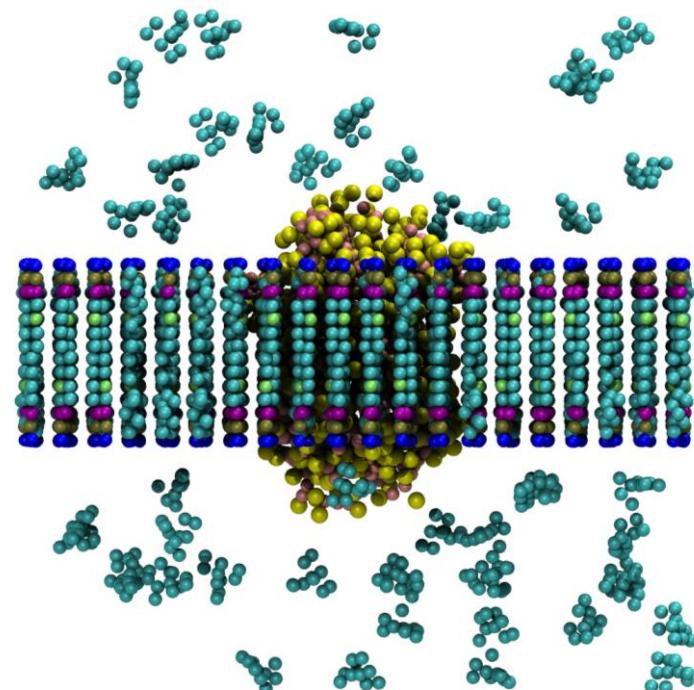
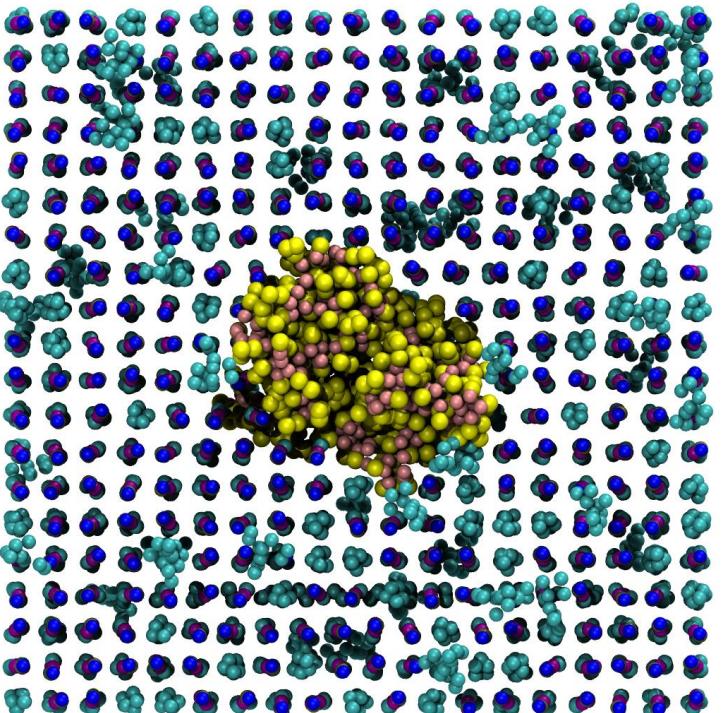
COMPLEX

```
ENSANE(  
    pbc = [20, 10, 12],  
    prot = ["04_output_martinize.pdb prot_name:AtSUC1_152ASP tx:5 tz:1.5",  
            "04_output_martinize.pdb prot_name:AtSUC1_152ASP tx:-5 tz:-1.5"],  
    leaf = ["type:upper POPC:5 POPE:3 CHOL:3",  
            "type:lower POPC:9 POPE:2 CHOL:4"],  
    imp_top = "Tutorial/top_for_ENSANE.top",  
    backup = False,  
    out = "Tutorial/example_systems/complex_system.pdb",  
    top = "Tutorial/example_systems/complex_topol.pdb",  
)
```



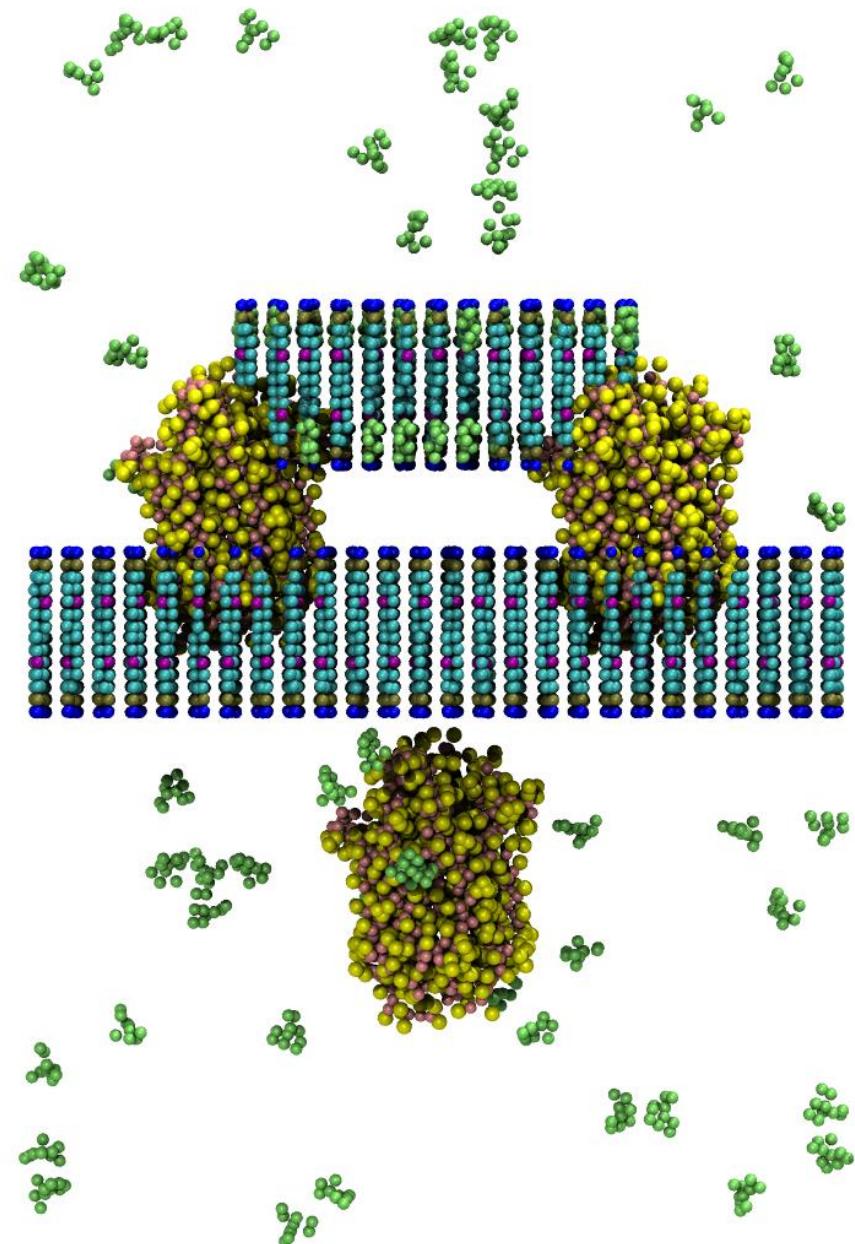
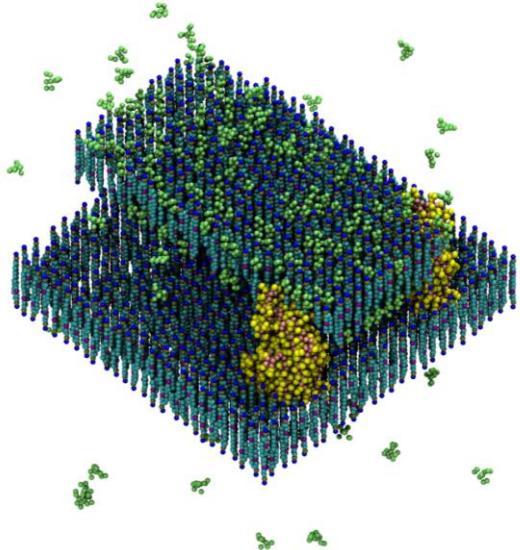
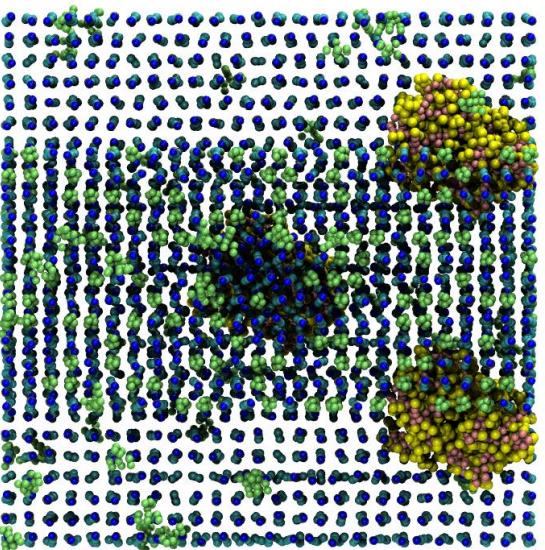
FLOODING

```
tensane = ENSANE(  
    pbc = [15, 15, 15],  
    prot = "04_output_martinize.pdb prot_name:AtSUC1_152ASP",  
    leaf = "type:bilayer POPC:10 POPE:4 CHOL:3",  
    solv = "solv:RH00 count:True solv_molarity:50 ff:imp",  
    imp_top = "Tutorial/top_for_ENSANE.top",  
    imp_struc = "rhodamine.pdb ff:imp",  
    backup = False,  
    out = "Tutorial/example_systems/flooding_system.pdb",  
    top = "Tutorial/example_systems/flooding_topol.pdb",  
)
```



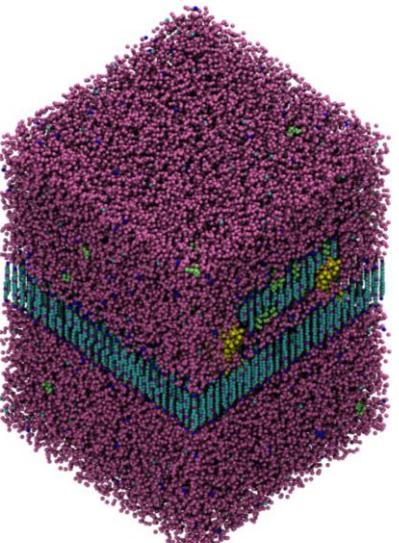
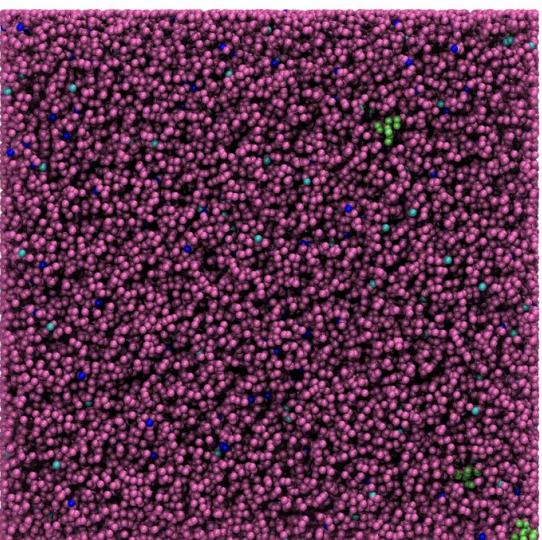
ABSTRACT

```
tensane = ENSANE(  
    pbc = [20, 20, 30],  
    prot = ["04_output_martinize.pdb prot_name:AtSUC1_152ASP tz:-6",  
            "04_output_martinize.pdb prot_name:AtSUC1_152ASP tx:7 ty:5 tz:3",  
            "04_output_martinize.pdb prot_name:AtSUC1_152ASP tx:7 ty:-5 tz:3"],  
  
    leaf = ["type:bilayer POPC:1 center:0:5:0 y:10",  
            "type:bilayer POPE:1 center:0:-5:0 y:10",  
            "type:bilayer POPC:1 POPE:1 CHOL:1 center:0:0:6 y:10"],  
  
    solv = "solv:RH00 count:True solv_molarity:50 ff:imp",  
  
    imp_top = "Tutorial/top_for_ENSANE.top",  
    imp_struc = "rhodamine.pdb ff:imp",  
    backup = False,  
    out = "Tutorial/example_systems/abstract_system.pdb",  
    top = "Tutorial/example_systems/abstract_topol.pdb",  
)
```

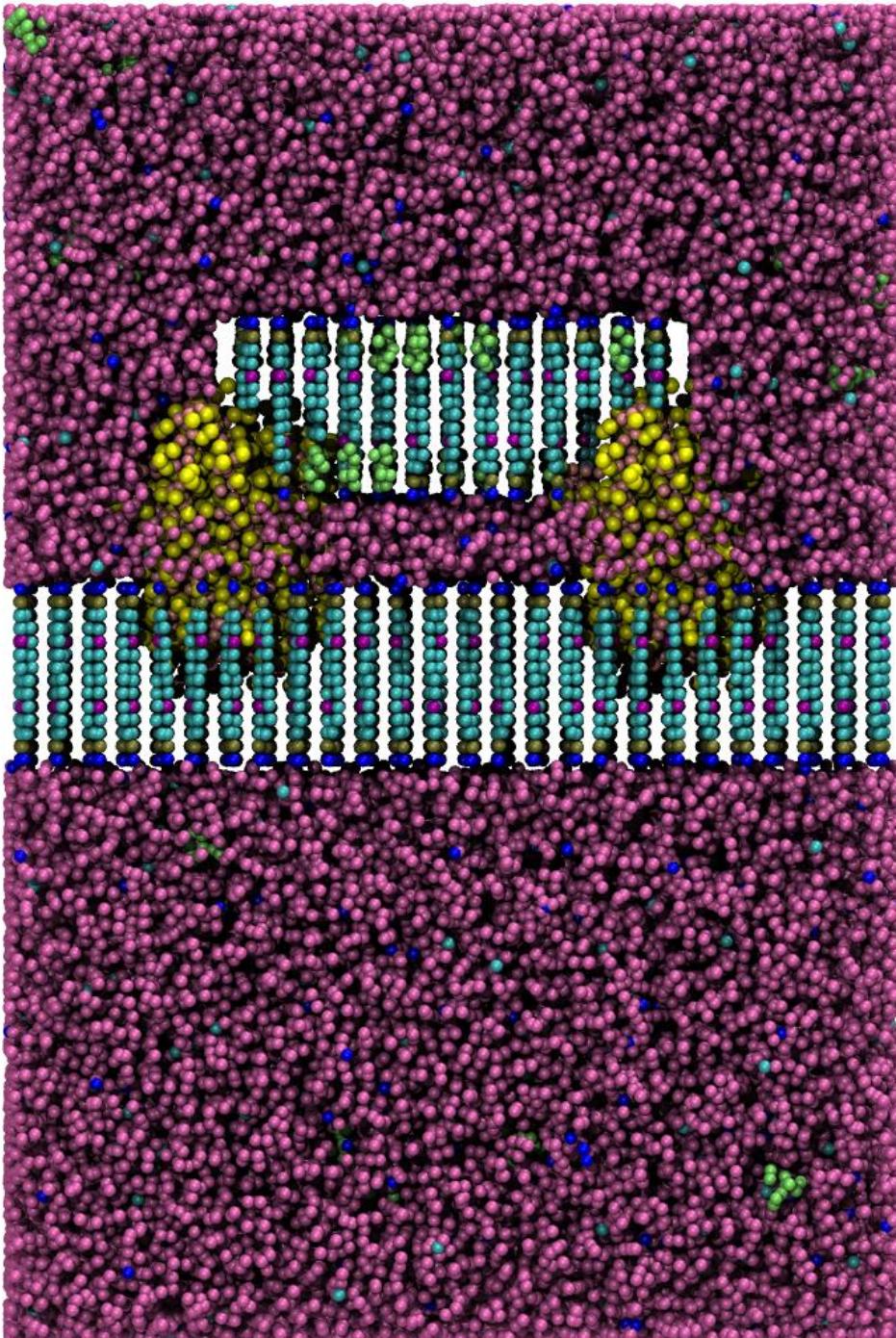


ABSTRACT WITH SOLVENT

```
tensane = ENSANE(  
    pbc = [20, 20, 30],  
    prot = ["04_output_martinize.pdb prot_name:AtSUC1_152ASP tz:-6",  
            "04_output_martinize.pdb prot_name:AtSUC1_152ASP tx:7 ty:5 tz:3",  
            "04_output_martinize.pdb prot_name:AtSUC1_152ASP tx:7 ty:-5 tz:3"],  
  
    leaf = ["type:bilayer POPC:1 center:0:5:0 y:10",  
            "type:bilayer POPE:1 center:0:-5:0 y:10",  
            "type:bilayer POPC:1 POPE:1 CHOL:1 center:0:0:6 y:10"],  
  
    solv = ["solv:RH00 count:True solv_molarity:50 ff:imp",  
            "solv:W pos:NA neg:CL"],  
  
    imp_top = "Tutorial/top_for_ENSANE.top",  
    imp_struc = "rhodamine.pdb ff:imp",  
    backup = False,  
    out = "Tutorial/example_systems/abstract_solv_system.pdb",  
    top = "Tutorial/example_systems/abstract_solv_topol.pdb",  
)
```



ANDREASEN
JDENT



HOW TO ACCESS

GITHUB

<https://github.com/MikkelDA/ENSANE>

Contains ENSANE.py, and a rough tutorial explaining what i presented today

Use with caution and please report any bugs

- Though i probably don't have time to make any major changes due to having to write my thesis

QUESTIONS?



AARHUS
UNIVERSITET

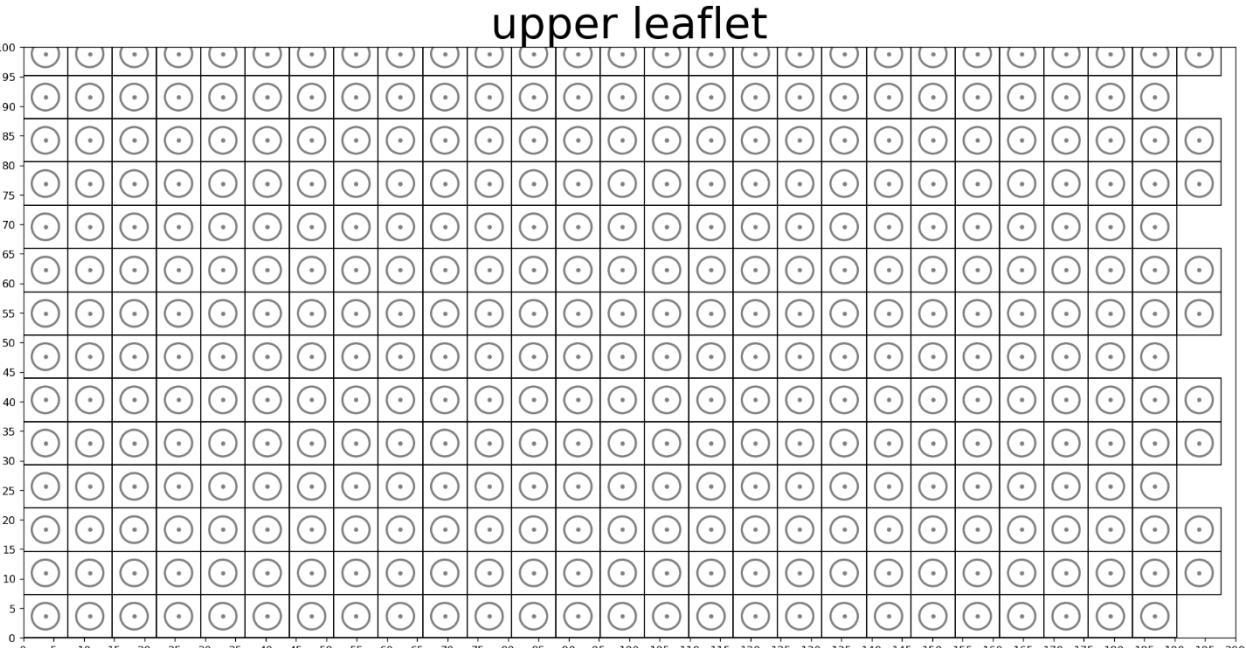
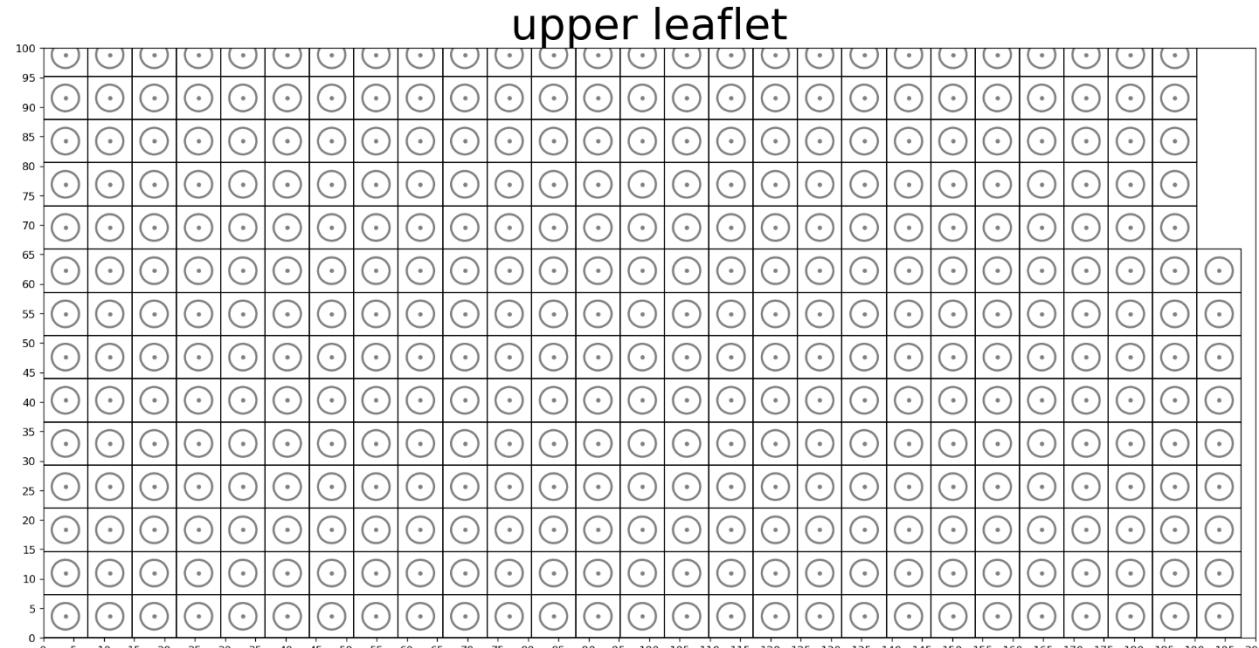
PROGRAM ALGORITHMS

The rest of the slides contain illustrations of how
the program's various algorithms work

LEAFLET GRID-CREATION

Leaflet details:
 $x = 200 \text{ \AA}$
 $y = 100 \text{ \AA}$
 $\text{apl} = 0.53632 \text{ nm}^2$

- Calculates max possible lipids from area of leaflet and area per lipid
- Inserts extra "grid points" until a "column" is full, then moves on to the next "column" until all grid points are placed
- It then divides the last "column's" grid points evenly amongst the "rows"

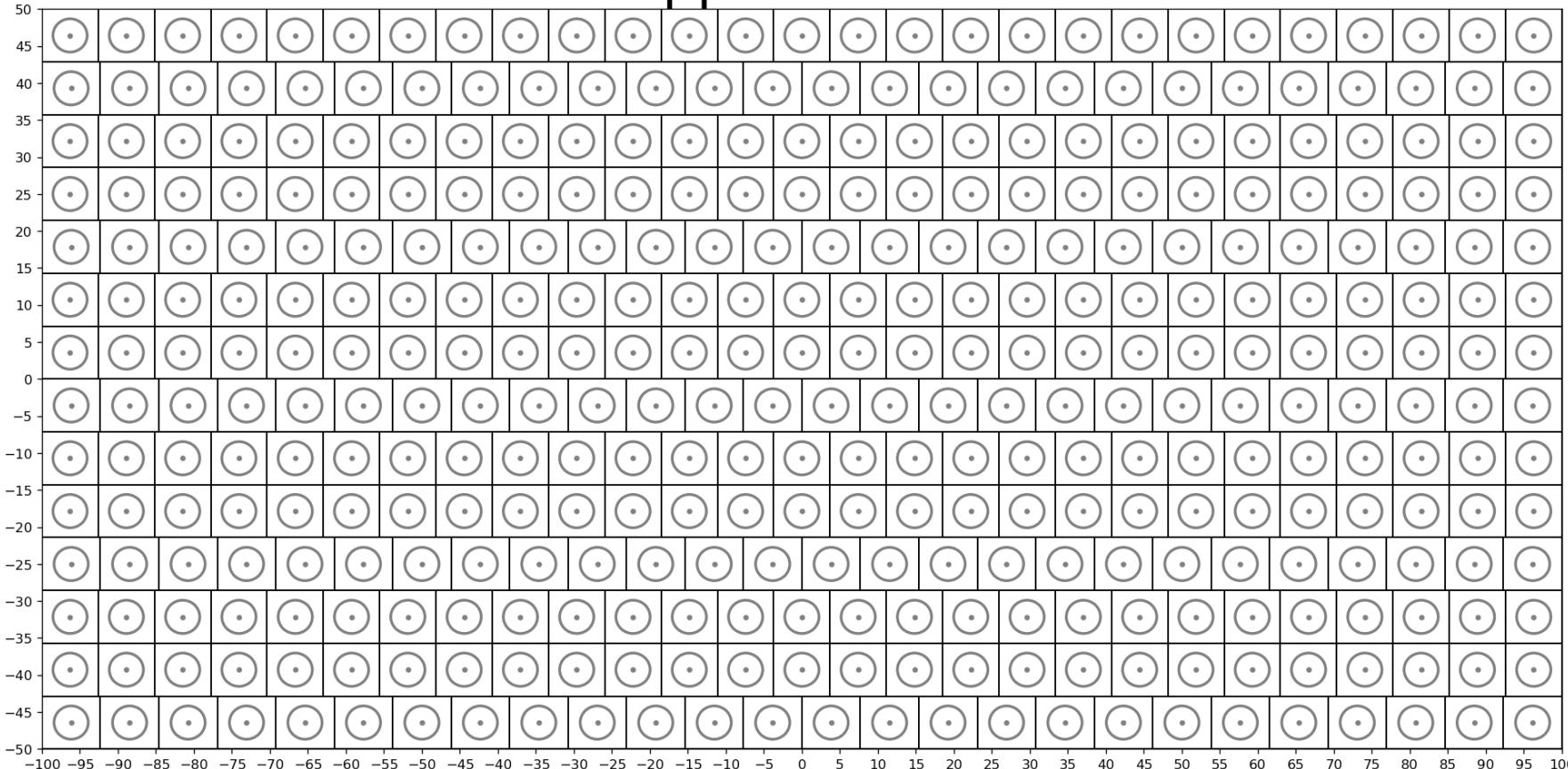


Circles radius is the largest radius any given lipid in the leaflet

LEAFLET GRID-CREATION

- Finally it "squeezes" all "columns" and "rows" such that the total height and width for each "column"/"row" matches the leaflets values

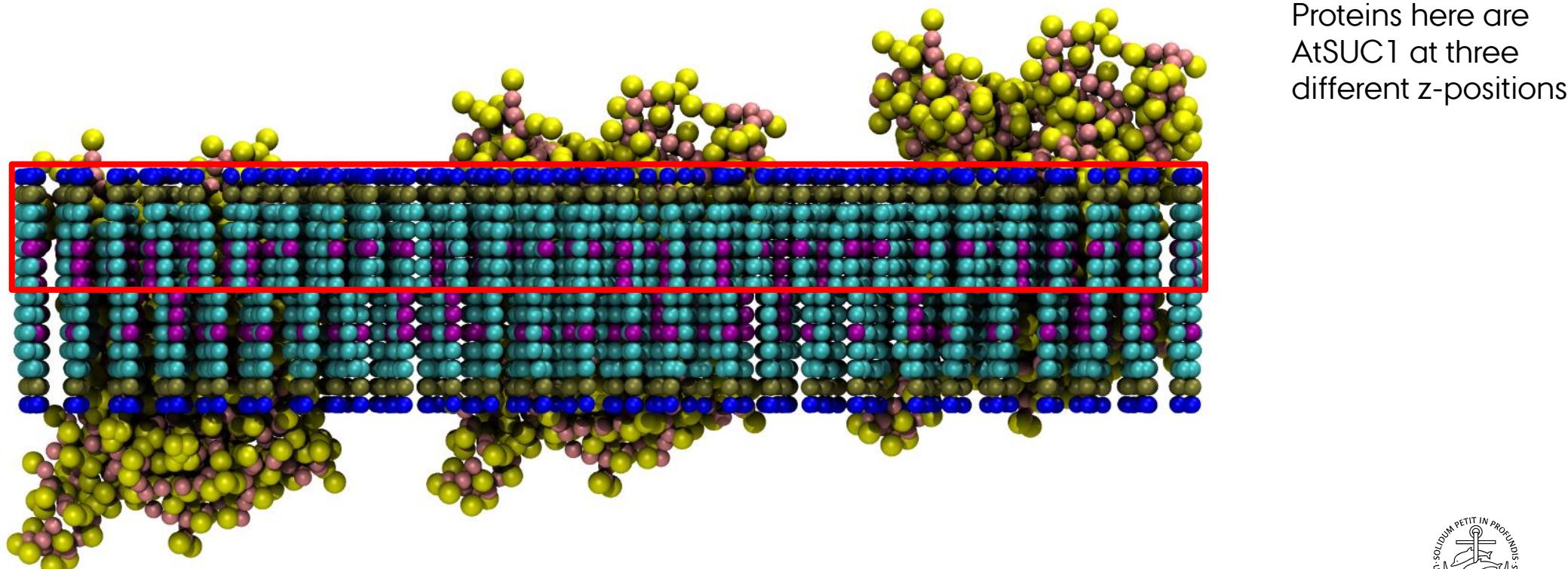
upper leaflet



PROTEIN INSERTION

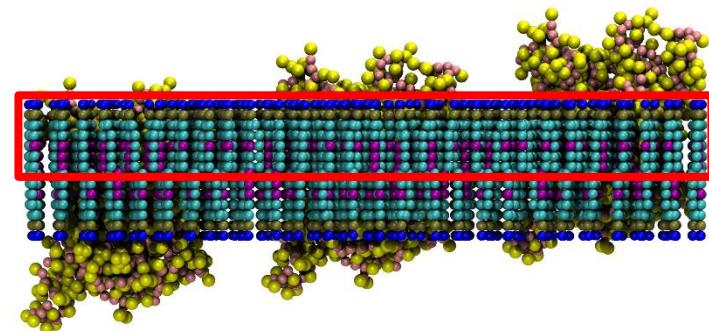
Can insert any number of proteins into the system

Protein beads within the leaflets minimum and maximum z-values are "flattened" onto the grid



Proteins here are AtSUC1 at three different z-positions

PROTEIN INSERTION

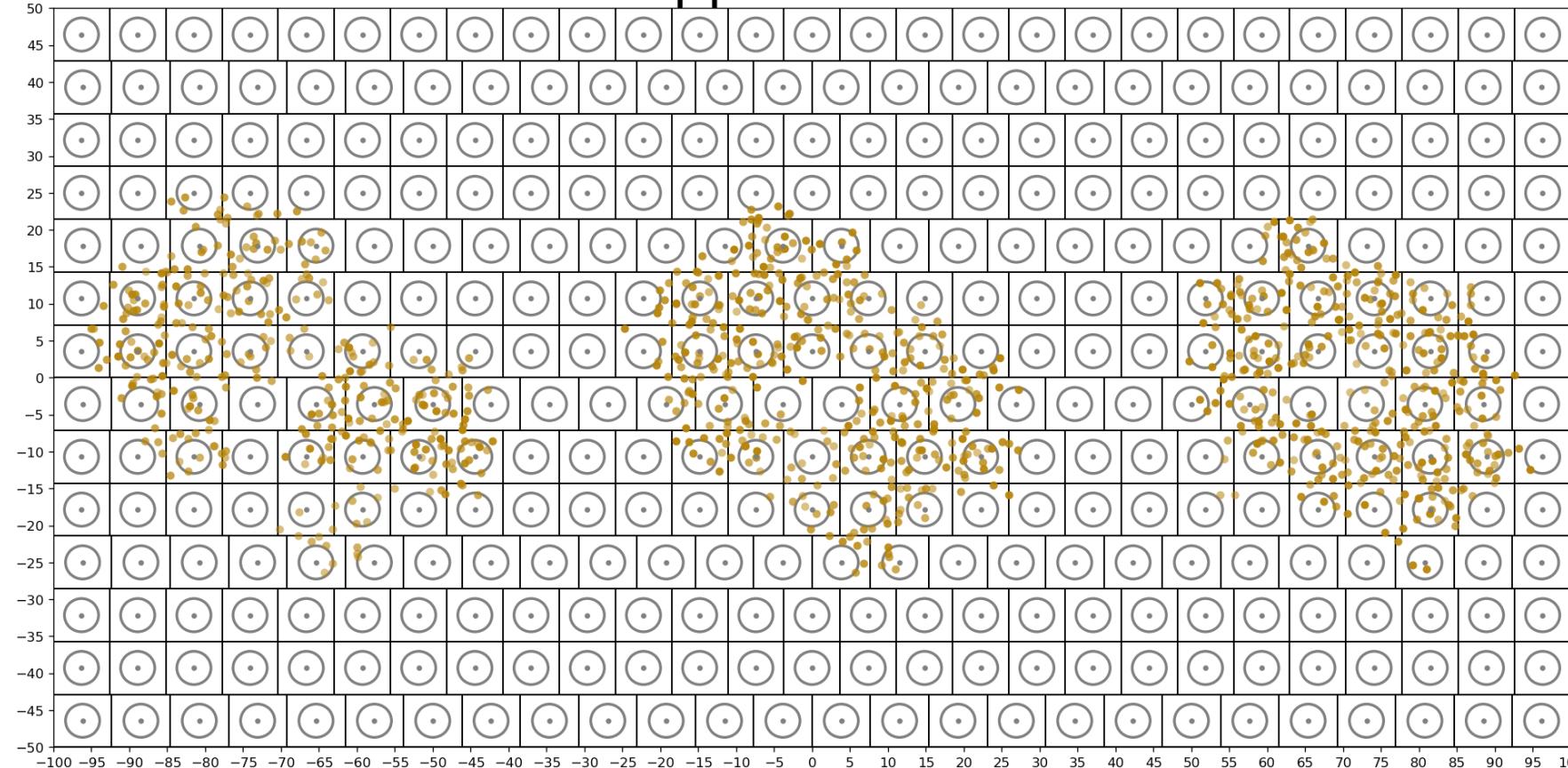


Can insert any number of proteins into the system

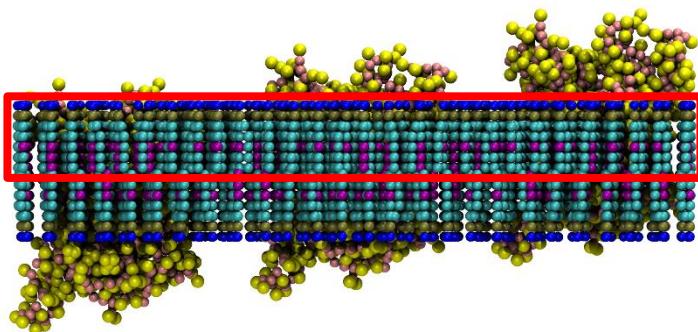
Protein beads within the leaflets minimum and maximum z-values are "flattened" onto the grid

Point:
- Protein bead

upper leaflet



OVERLAP



Mark grid points with protein bead overlap

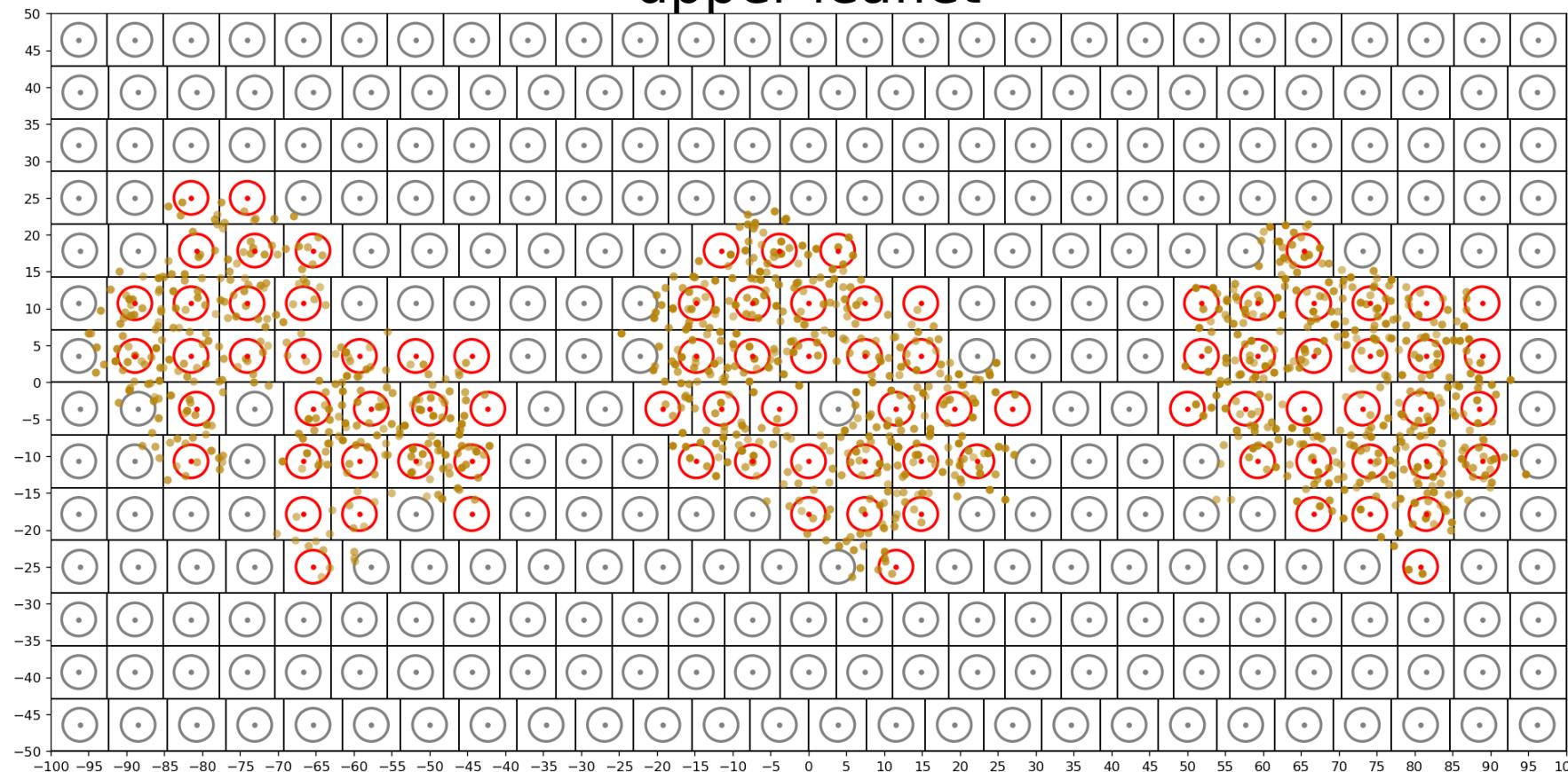
Not perfect definition of grid points that cannot be used for lipids

Must eliminate internal grid points

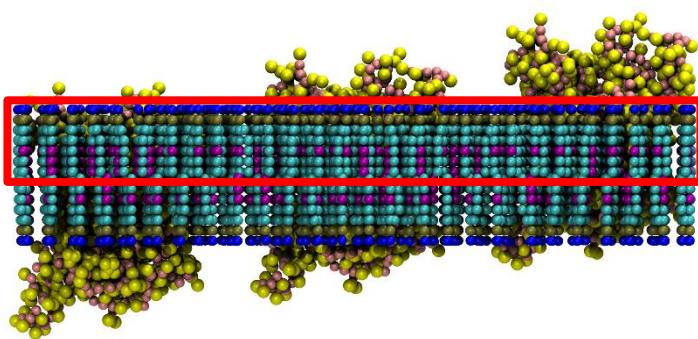
Point:
- Protein bead

Circles:
- Protein bead overlap

upper leaflet



CONVEX HULL



Convex hull?

Marks grid points that could contain a lipid

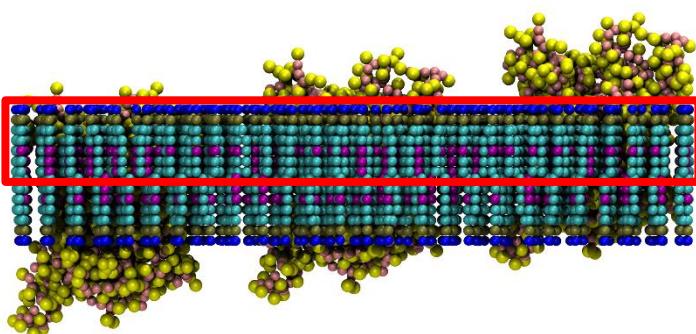
Point:
- Protein bead

Circles:
- Protein bead overlap
- Inside convex hull

upper leaflet



CONCAVE HULL - ALPHASHAPE

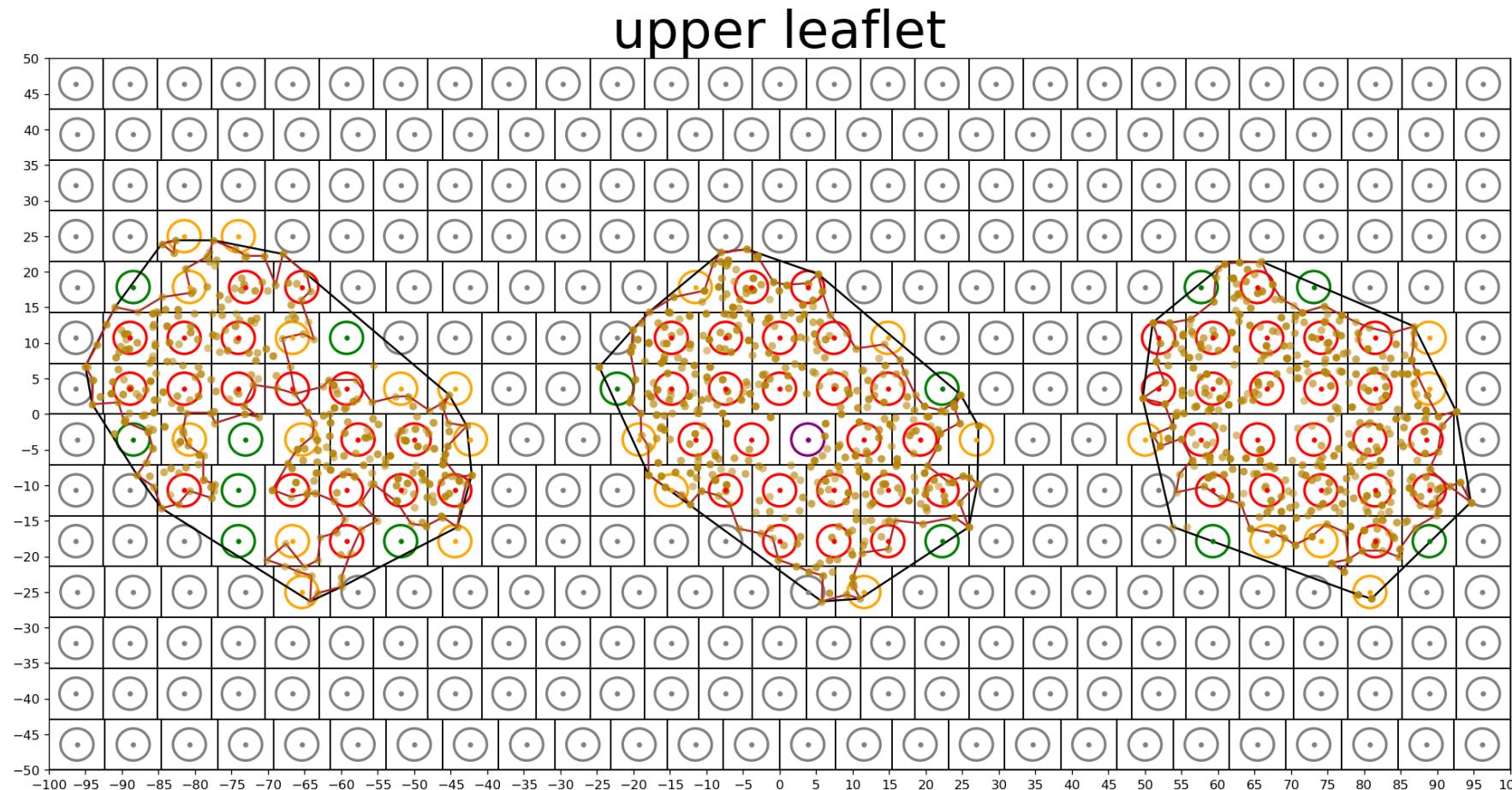


Concave hull / Alphashape

Based on a minimum distance between outermost points on hull

Point:
- Protein bead

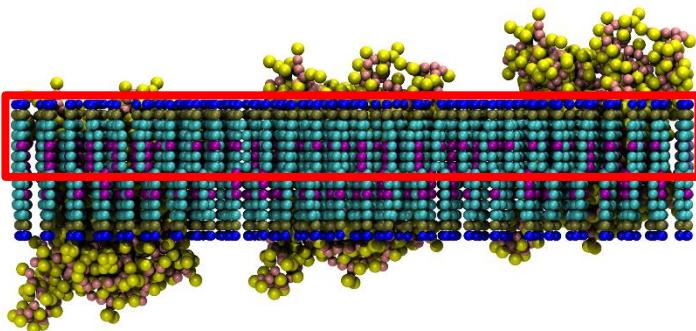
Circles:
- Protein bead overlap
- Inside convex hull
- Outside concave hull
- Overlap but outside hull



AARHUS
UNIVERSITET
NATURAL SCIENCES



NEW AREA



New area due to proteins

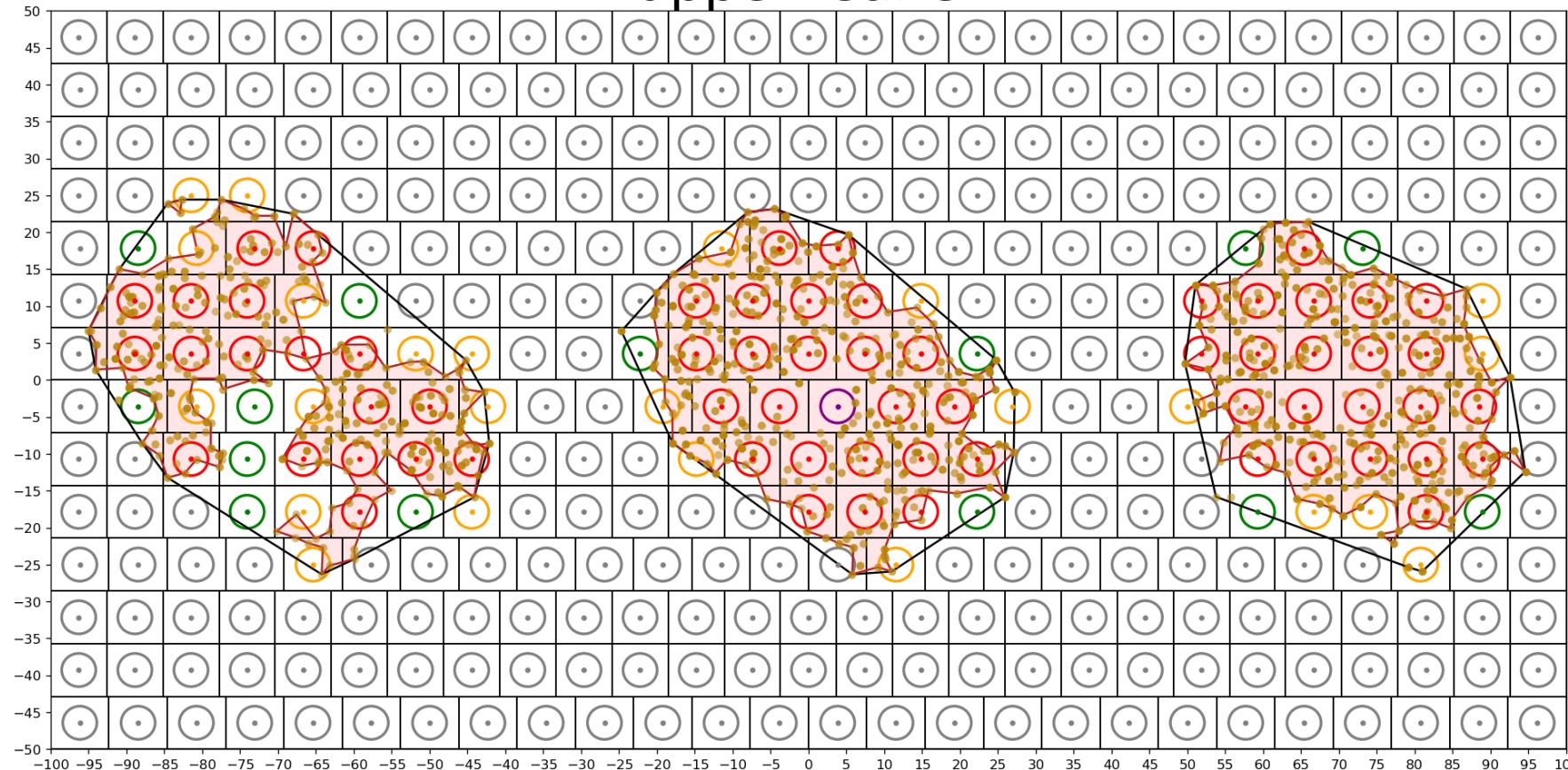
Must calculate new maximum number of lipids

What is the cross-sectional area of a protein when beads are simply points?

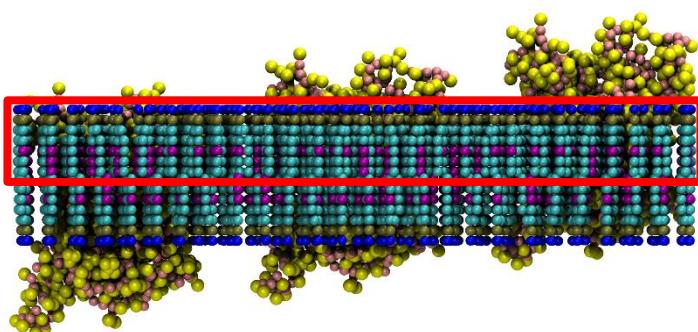
upper leaflet

Point:
- Protein bead

Circles:
- Protein bead overlap
- Inside convex hull
- Outside concave hull
- Overlap but outside hull



PROTEIN AREA - BUFFER



Adding a buffer to simulate extra space reserved for protein

Point:

- Protein bead

Circles:

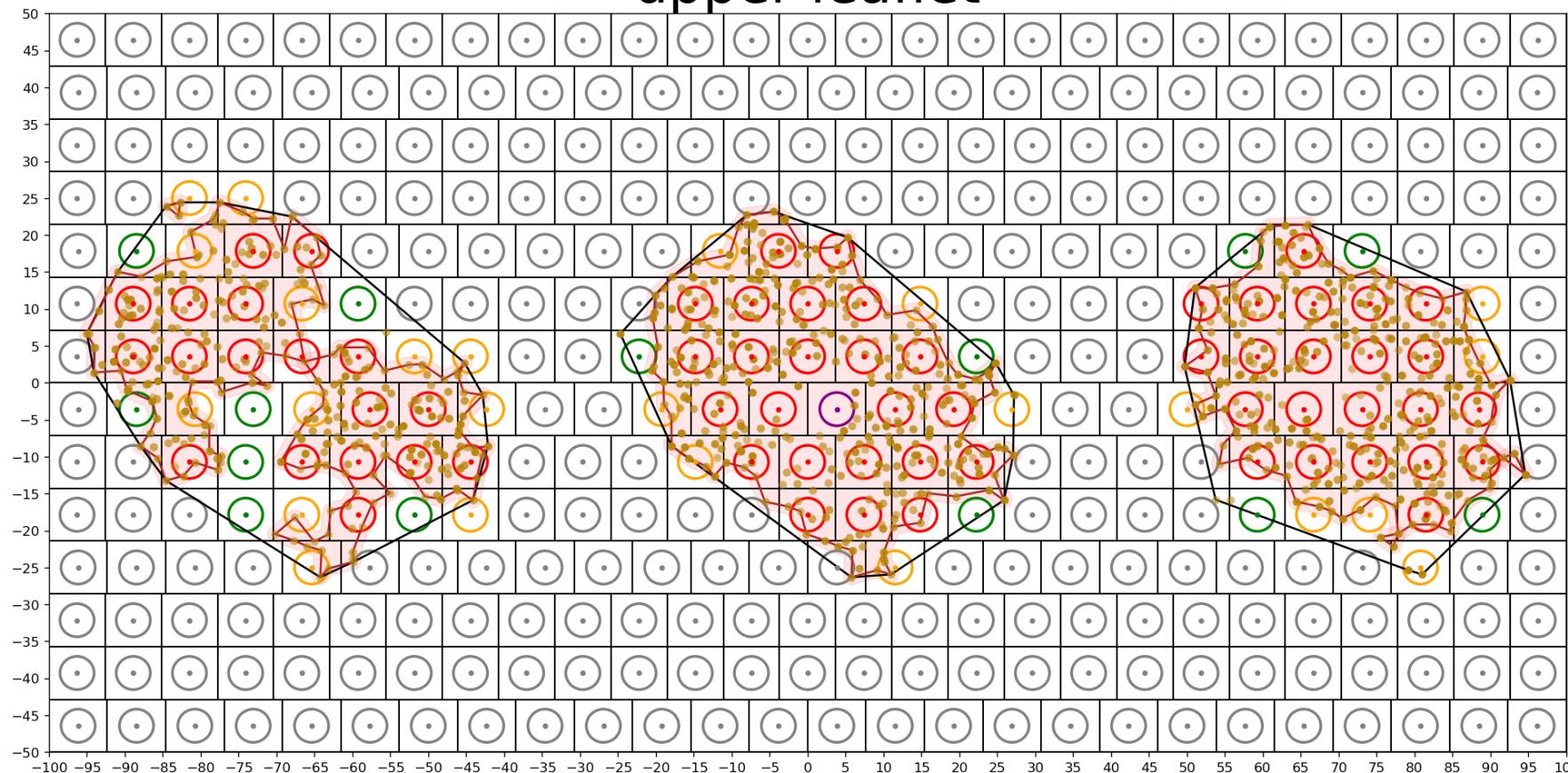
- Protein bead overlap

- Inside convex hull

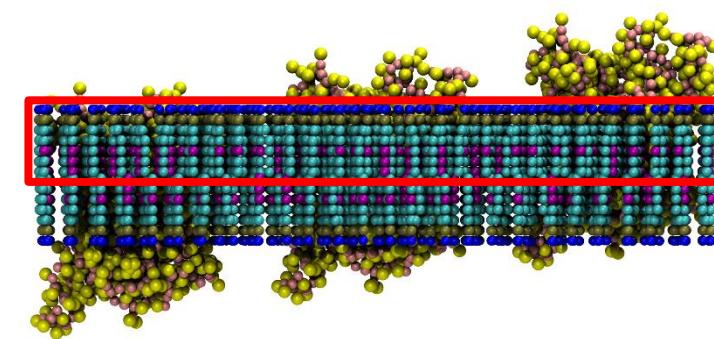
- Outside concave hull

- Overlap but outside hull

upper leaflet



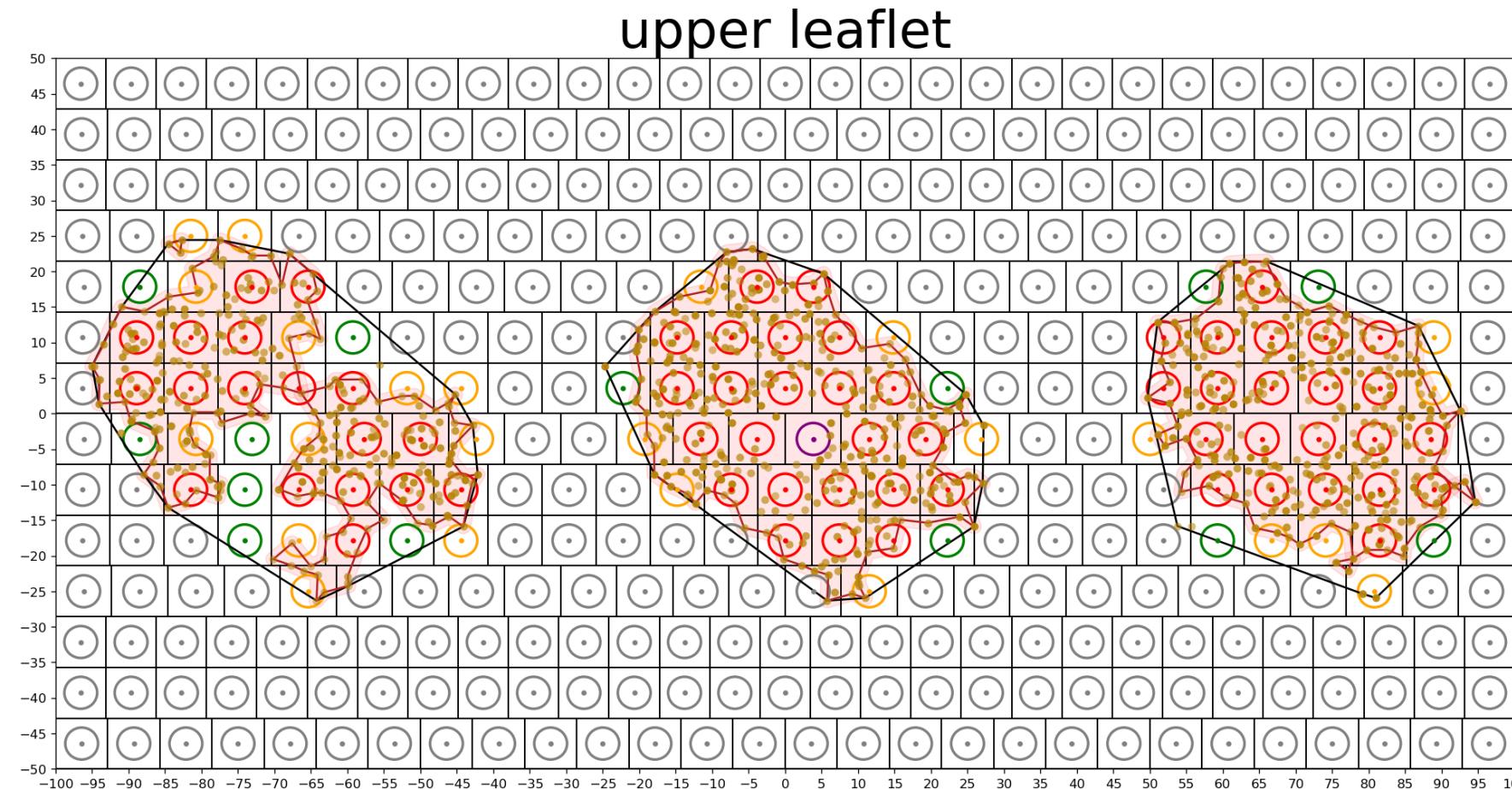
ADJUSTING GRIDS



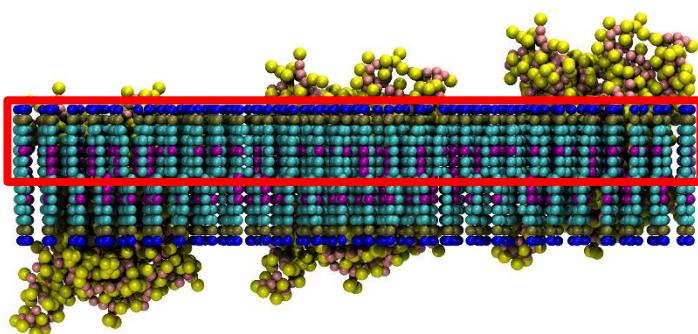
Adjusting number of grid points in rows without proteins to accommodate max potential number of lipids due to proteins taking up space

Point:
- Protein bead

Circles:
- Protein bead overlap
- Inside convex hull
- Outside concave hull
- Overlap but outside hull



DISTRIBUTING LIPIDS



Randomly distribute lipids in membrane based on input ratios

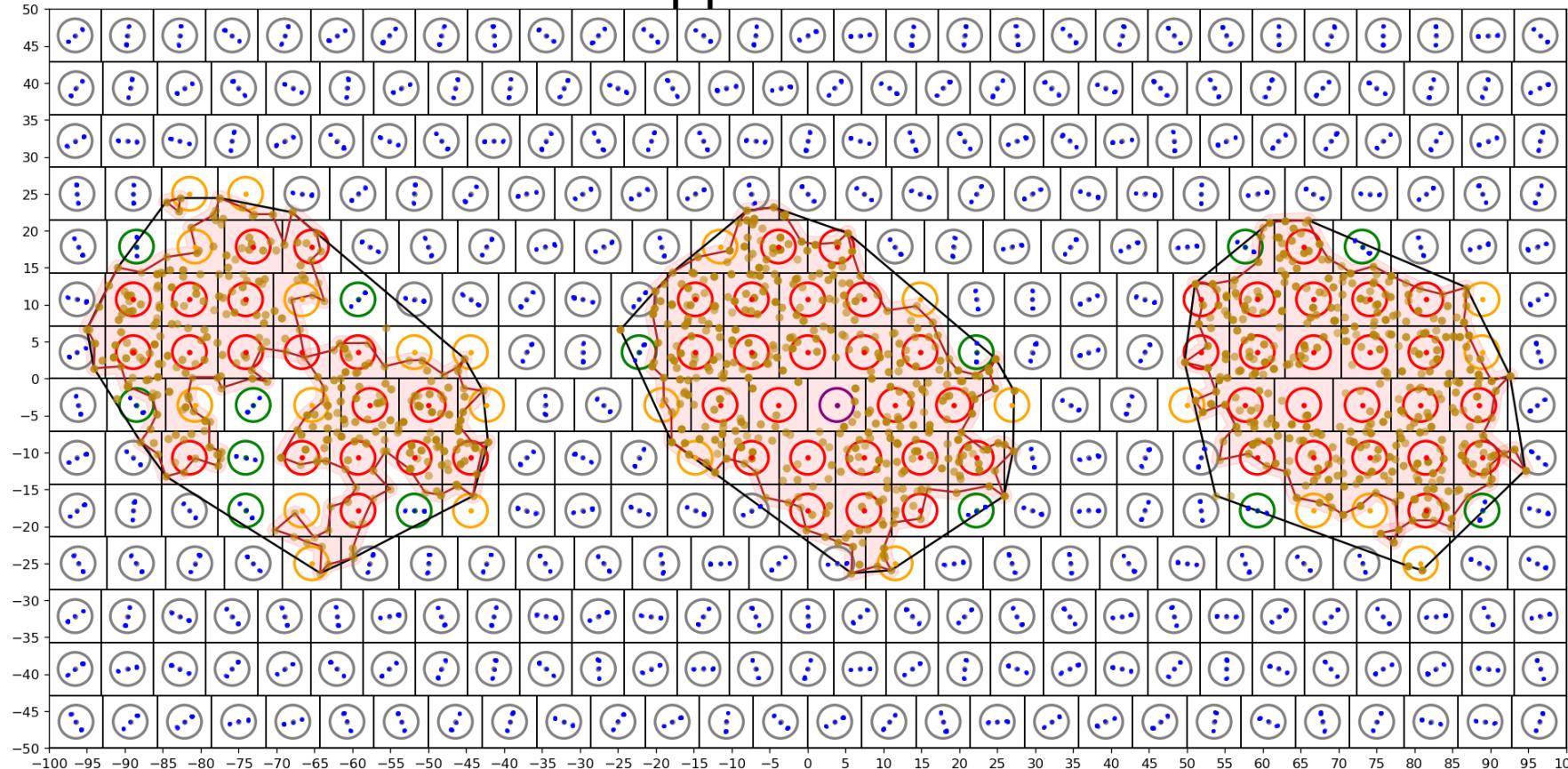
Point:

- Protein bead
- Lipid beads

Circles:

- Protein bead overlap
- Inside convex hull
- Outside concave hull
- Overlap but outside hull

upper leaflet

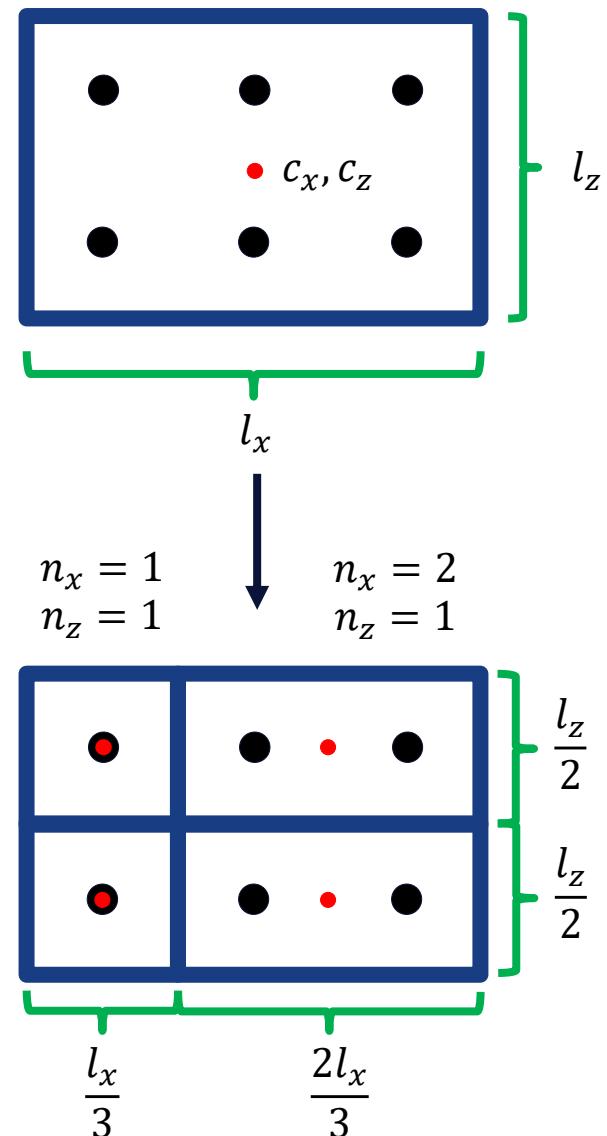


AARHUS
UNIVERSITET
NATURAL SCIENCES

"IMPLICIT" SOLVENT MODEL

"Implicit" solvent representation during "3D-grid/cell" calculations:

- A solvent cell contains:
 - x/y/z center coordinates
 - Cell side lengths in x/y/z axes
 - Number of potential solvent particles in x/y/z directions
 - "Implicit 3D-grid"
- Example: (example in 2D on x/z-plane)
 - x/y/z center = center of simulation box
 - Widths = length of simulation box along x/y/z axes
 - Potential solvent particles:
 - $n = \frac{\text{width}}{\text{grid resolution}}$ Grid resolution = smallest allowed cell size
 - Rounded down
 - Must be an integer
 - Cells subdivisible (each axis cut in half)
 - Potential solvent must remain integers

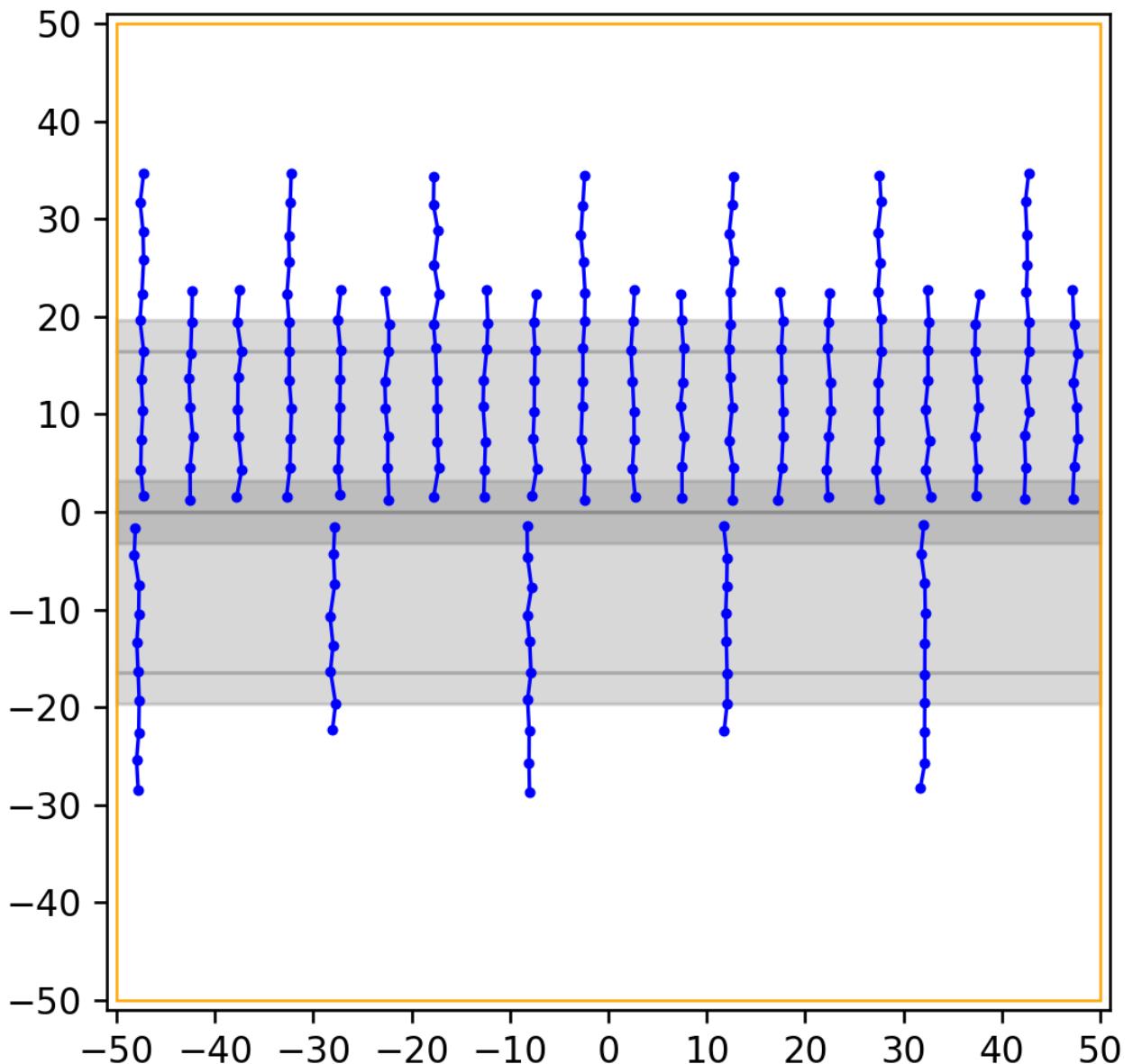


2D EXAMPLE

Starting parameters:

- *grid resolution* = 1.2
- *Buffer* = *grid resolution* + 2
- $c_x = 0, l_x = 100, n_x = 83$
- $c_z = 0, l_z = 100, n_z = 83$
- *Max potential solvent* = $n_x * n_z = 6889$

Iteration: 0



2D EXAMPLE

Lines:

- **Lipids**

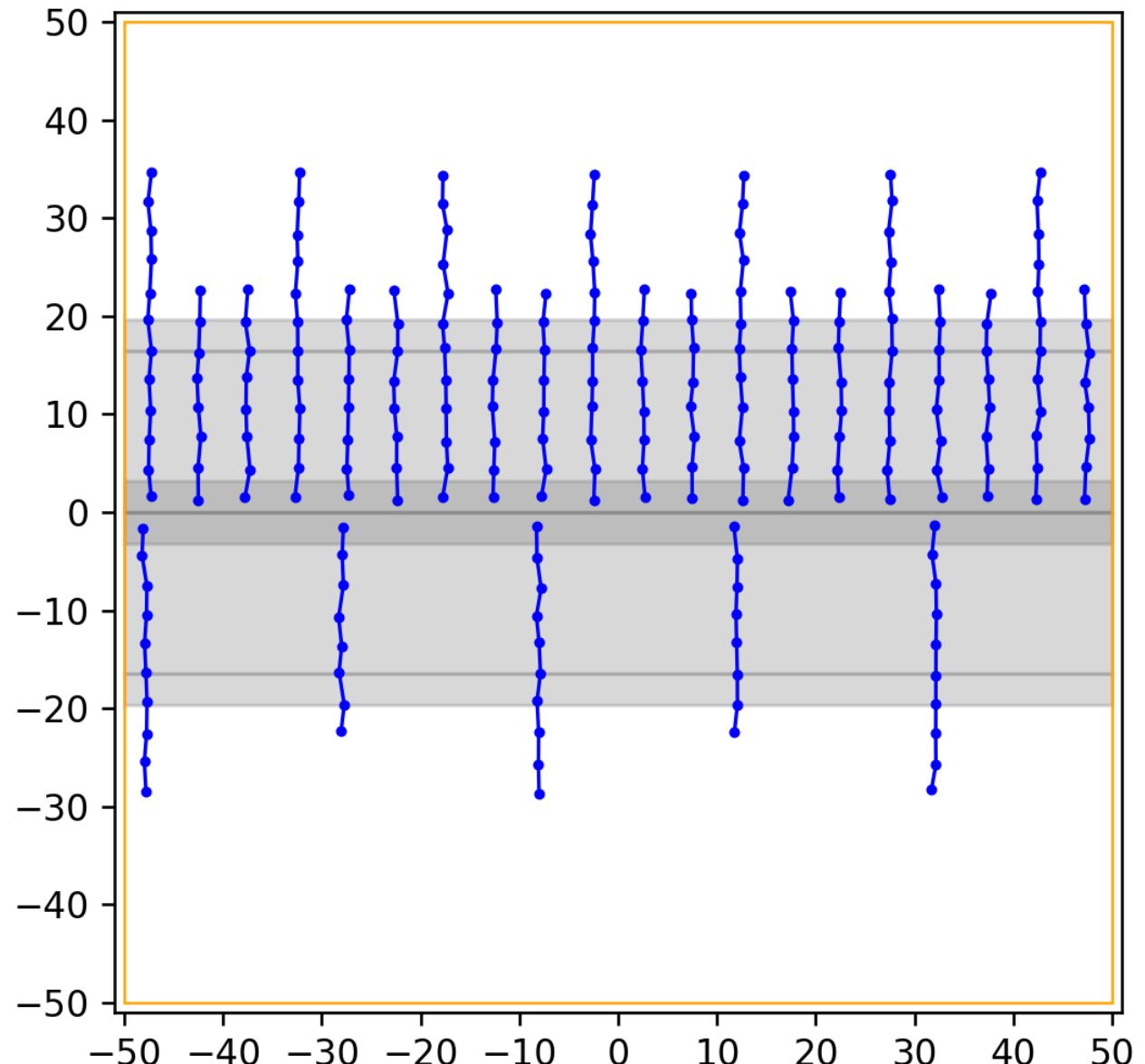
Dots:

- **Lipid beads**

Squares

- **Cell with overlapping beads**

Iteration: 0



2D EXAMPLE

Lines:

- **Lipids**

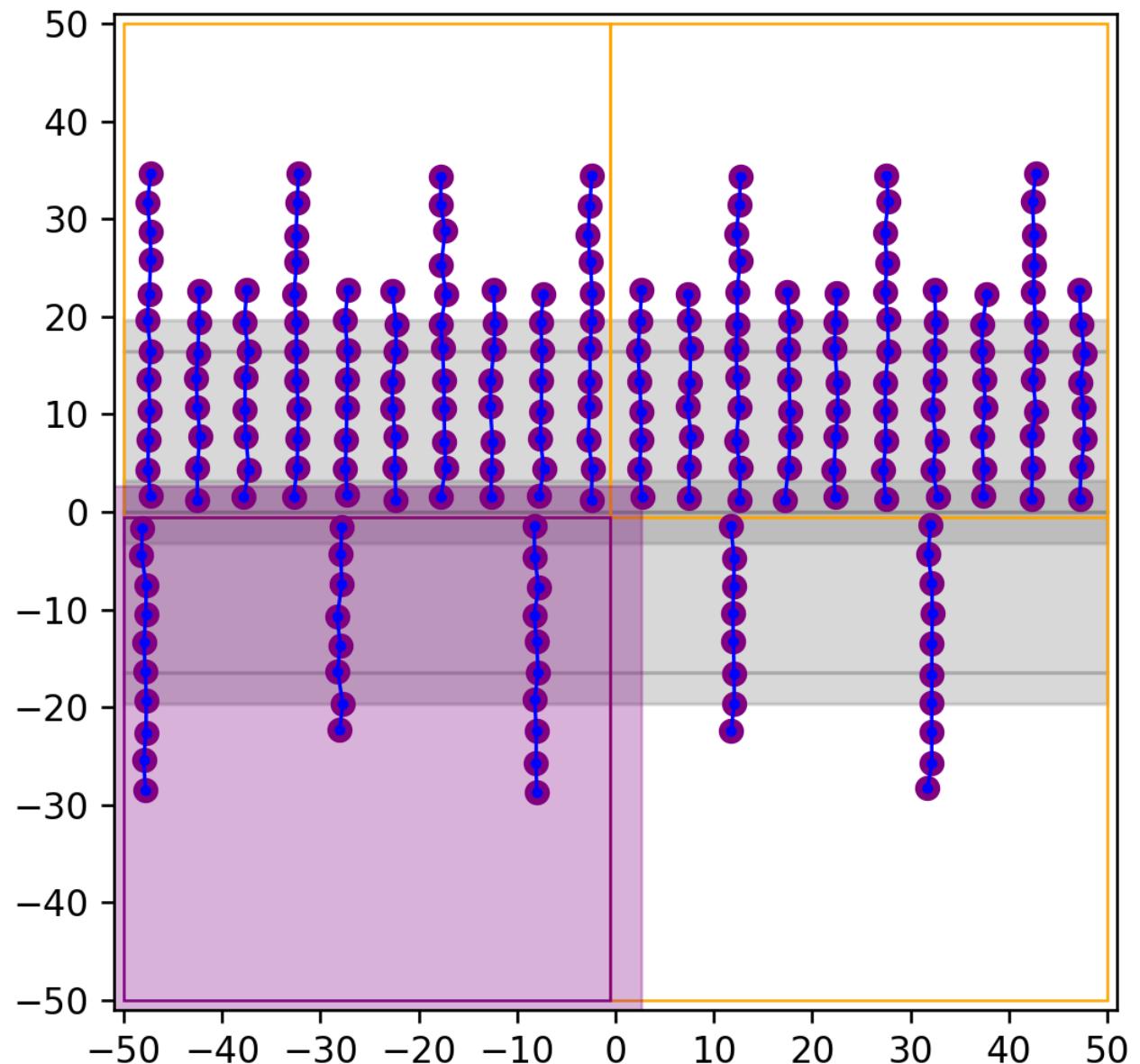
Dots:

- **Lipid beads**
- **Beads used for overlap checks**

Squares

- **Cell with overlapping beads**
- **Cell used for overlap checks**

Iteration: 1



2D EXAMPLE

Lines:

- **Lipids**

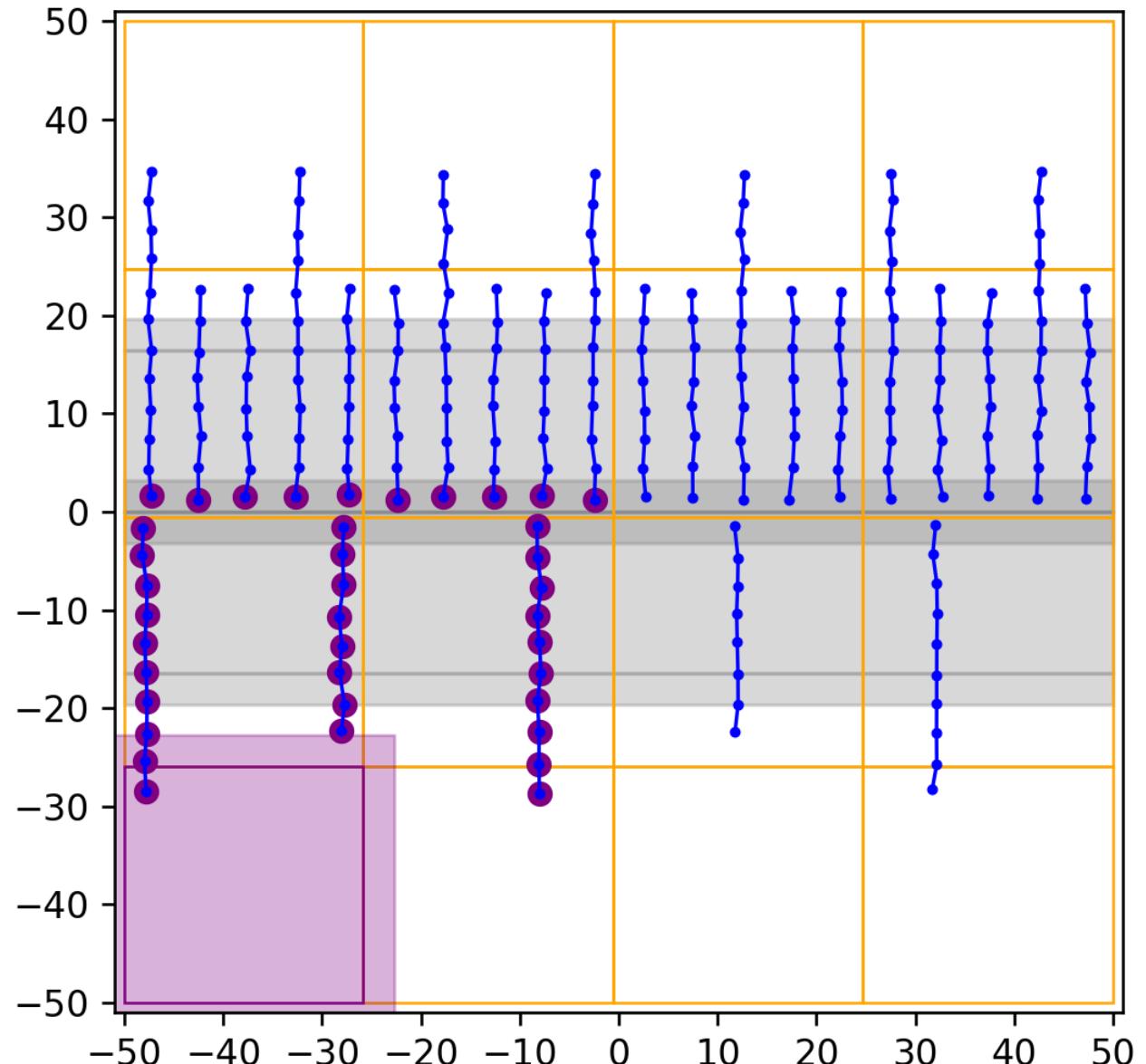
Dots:

- **Lipid beads**
- **Beads used for overlap checks**

Squares

- **Cell with overlapping beads**
- **Cell used for overlap checks**

Iteration: 2



2D EXAMPLE

Lines:

- **Lipids**

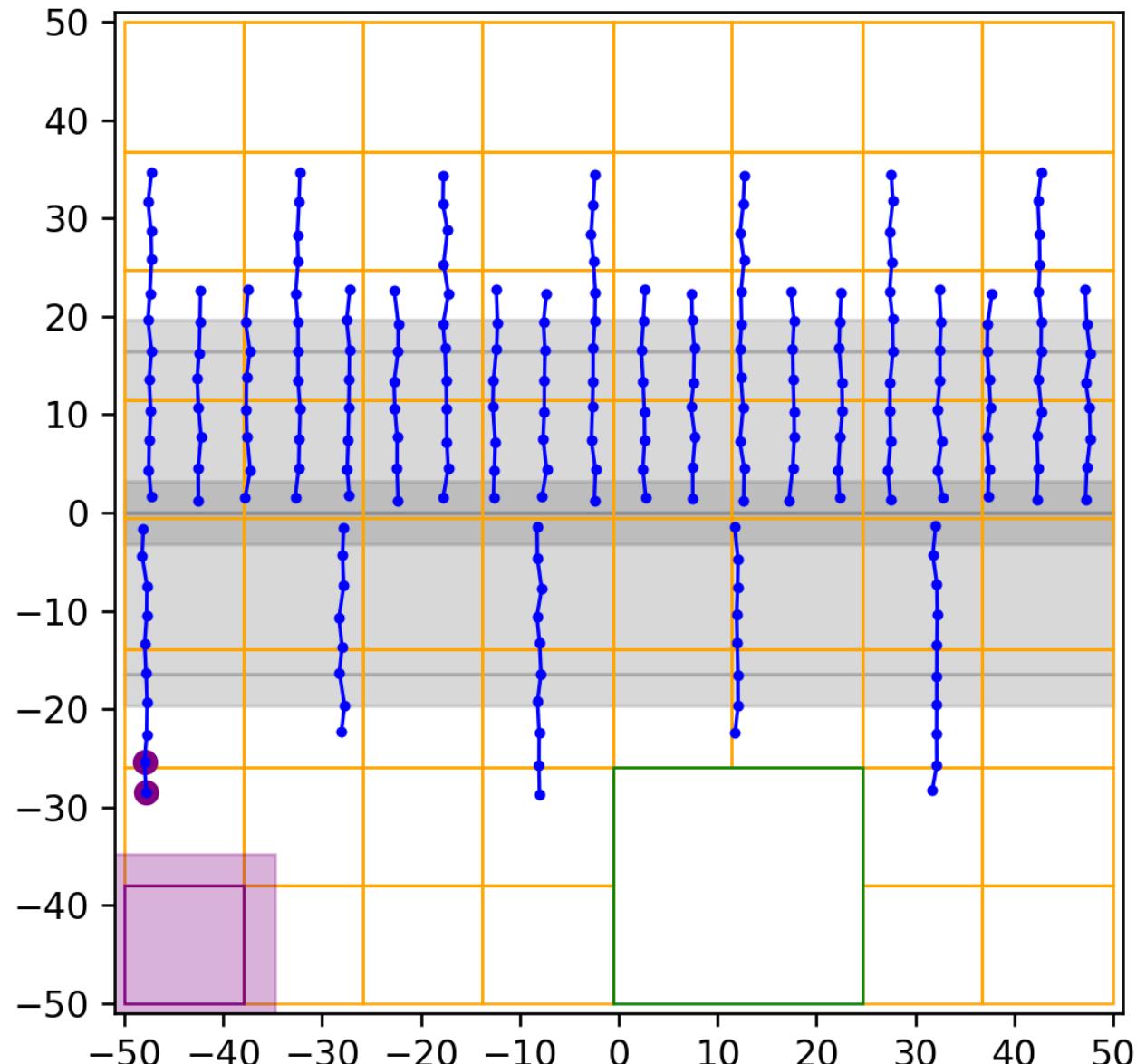
Dots:

- **Lipid beads**
- **Beads used for overlap checks**

Squares

- **Cell with overlapping beads**
- **Cell used for overlap checks**
- **"Free" cell**

Iteration: 3



2D EXAMPLE

Lines:

- **Lipids**

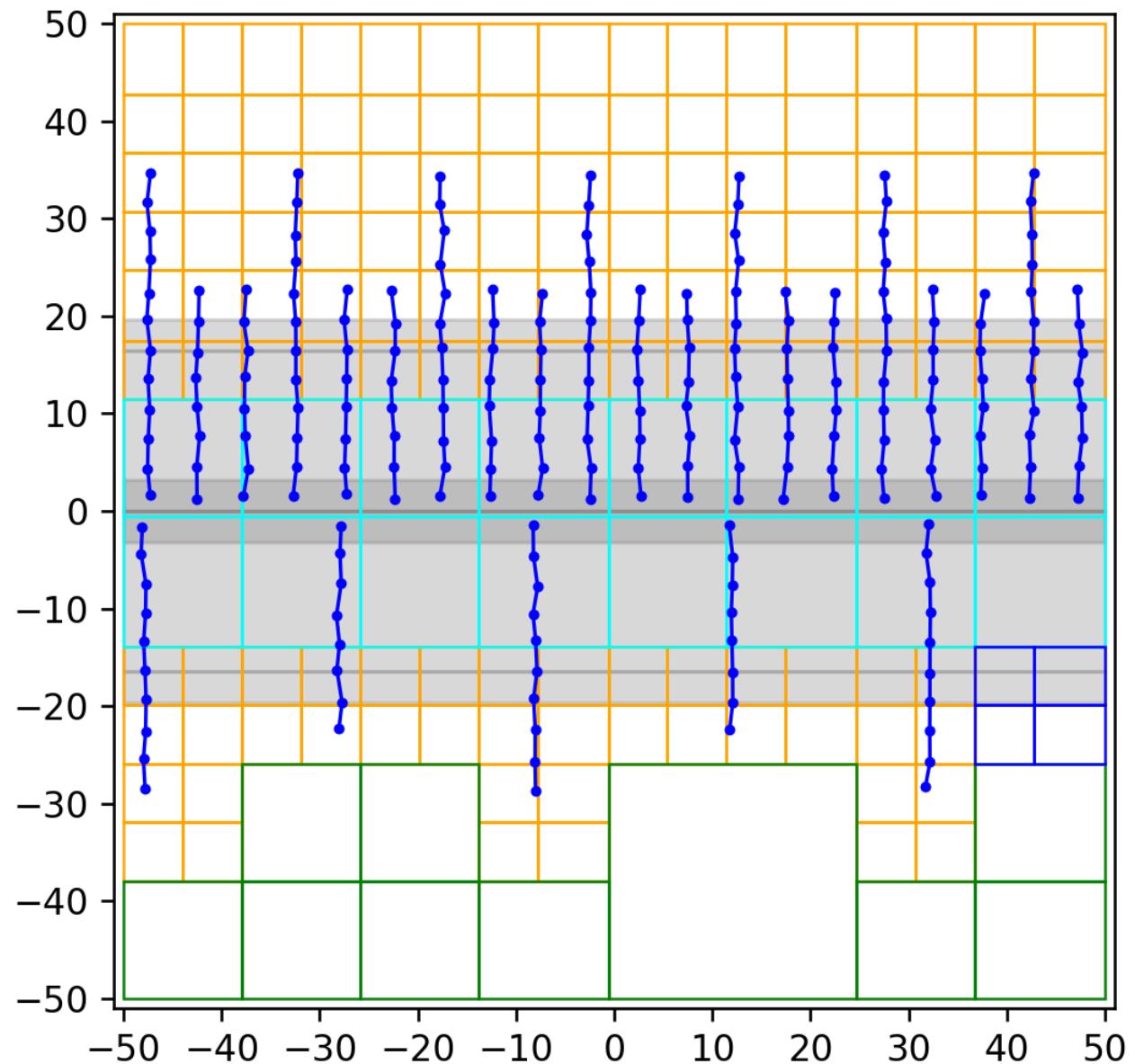
Dots:

- **Lipid beads**
- **Beads used for overlap checks**

Squares

- **Cell with overlapping beads**
- **Cell used for overlap checks**
- **"Free" cell**
- **Cell completely contained in hydrophobic area**
- **Cell partially contained in hydrophobic area while not overlapping with beads**

Iteration: 4



2D EXAMPLE

Lines:

- **Lipids**

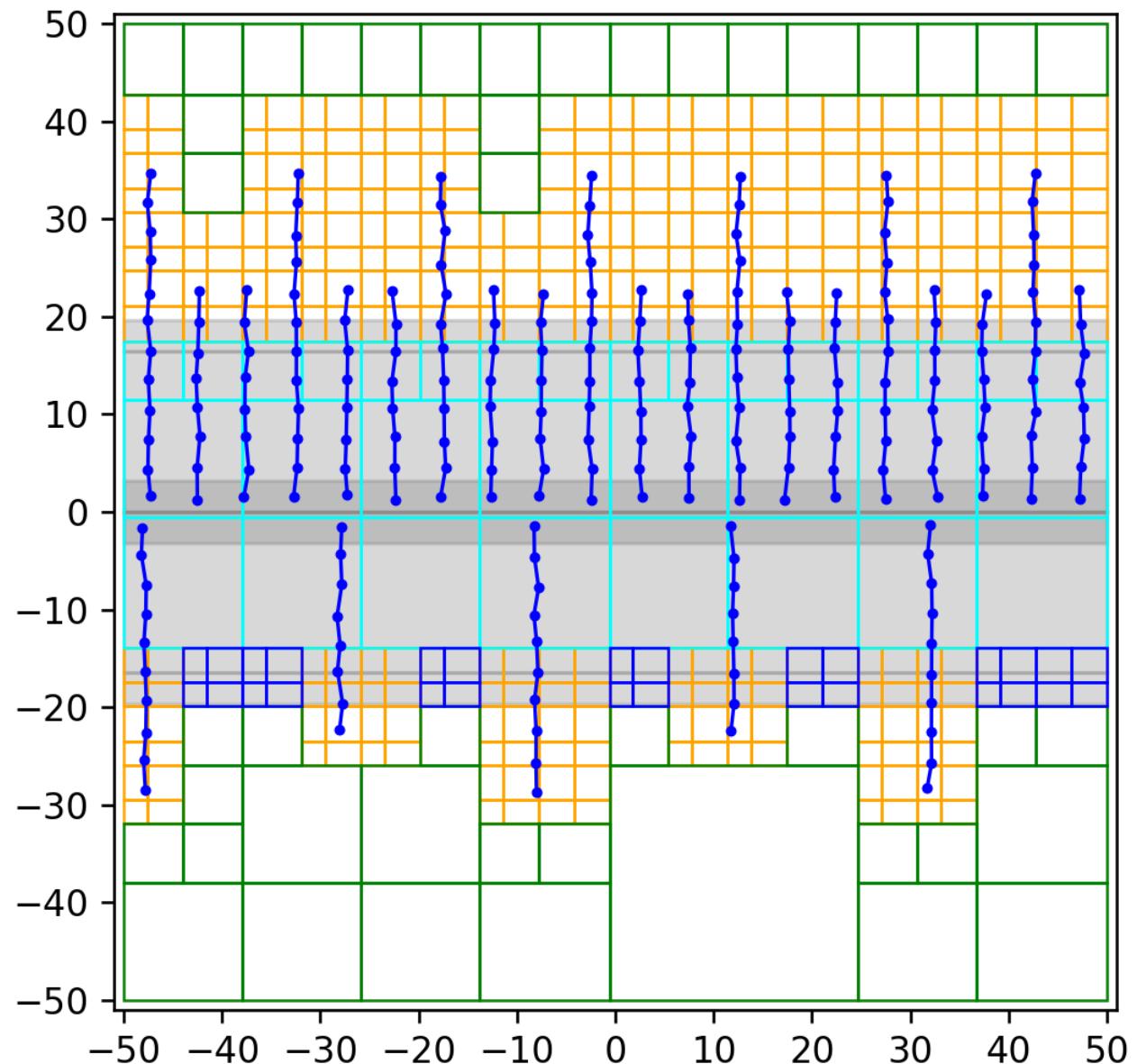
Dots:

- **Lipid beads**
- **Beads used for overlap checks**

Squares

- **Cell with overlapping beads**
- **Cell used for overlap checks**
- **"Free" cell**
- **Cell completely contained in hydrophobic area**
- **Cell partially contained in hydrophobic area while not overlapping with beads**

Iteration: 5



2D EXAMPLE

Lines:

- Lipids

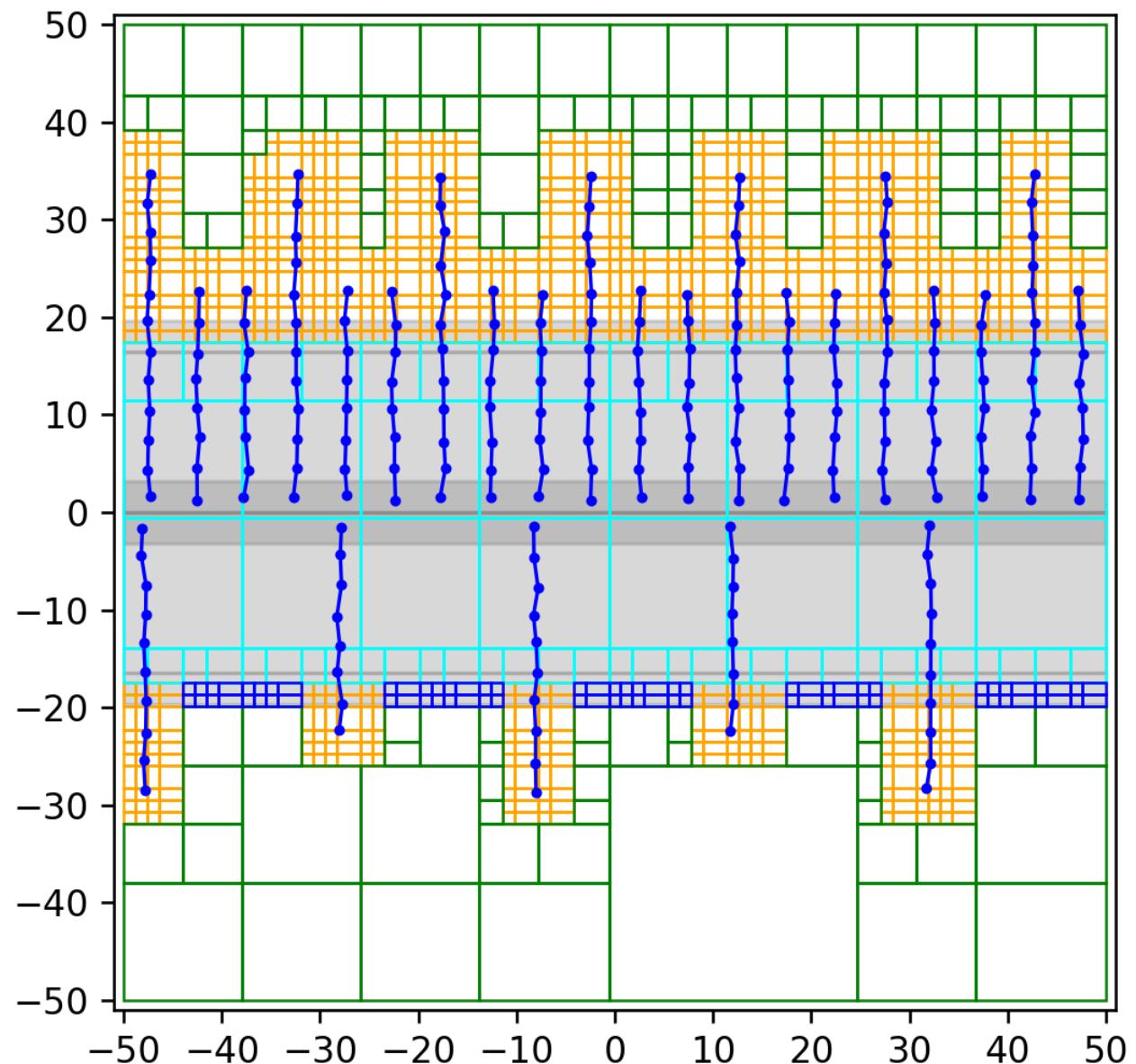
Dots:

- Lipid beads
- Beads used for overlap checks

Squares

- Cell with overlapping beads
- Cell used for overlap checks
- "Free" cell
- Cell completely contained in hydrophobic area
- Cell partially contained in hydrophobic area while not overlapping with beads

Iteration: 6



2D EXAMPLE

Lines:

- Lipids

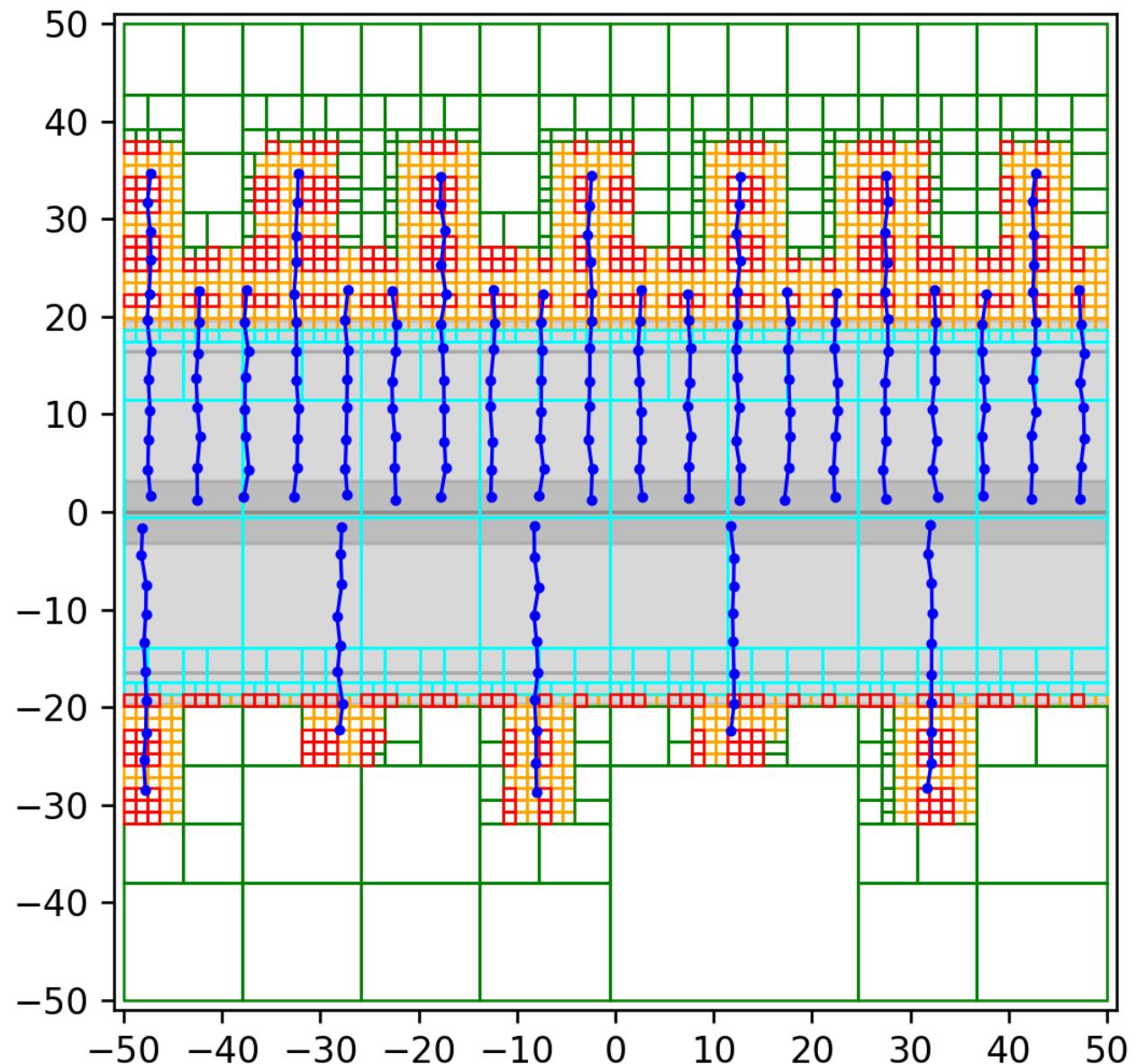
Dots:

- Lipid beads
- Beads used for overlap checks

Squares

- Cell with overlapping beads
- Cell used for overlap checks
- "Free" cell
- Cell completely contained in hydrophobic area
- Cell partially contained in hydrophobic area while not overlapping with beads
- Cells removed due to size restrictions

Iteration: 7



2D EXAMPLE

Lines:

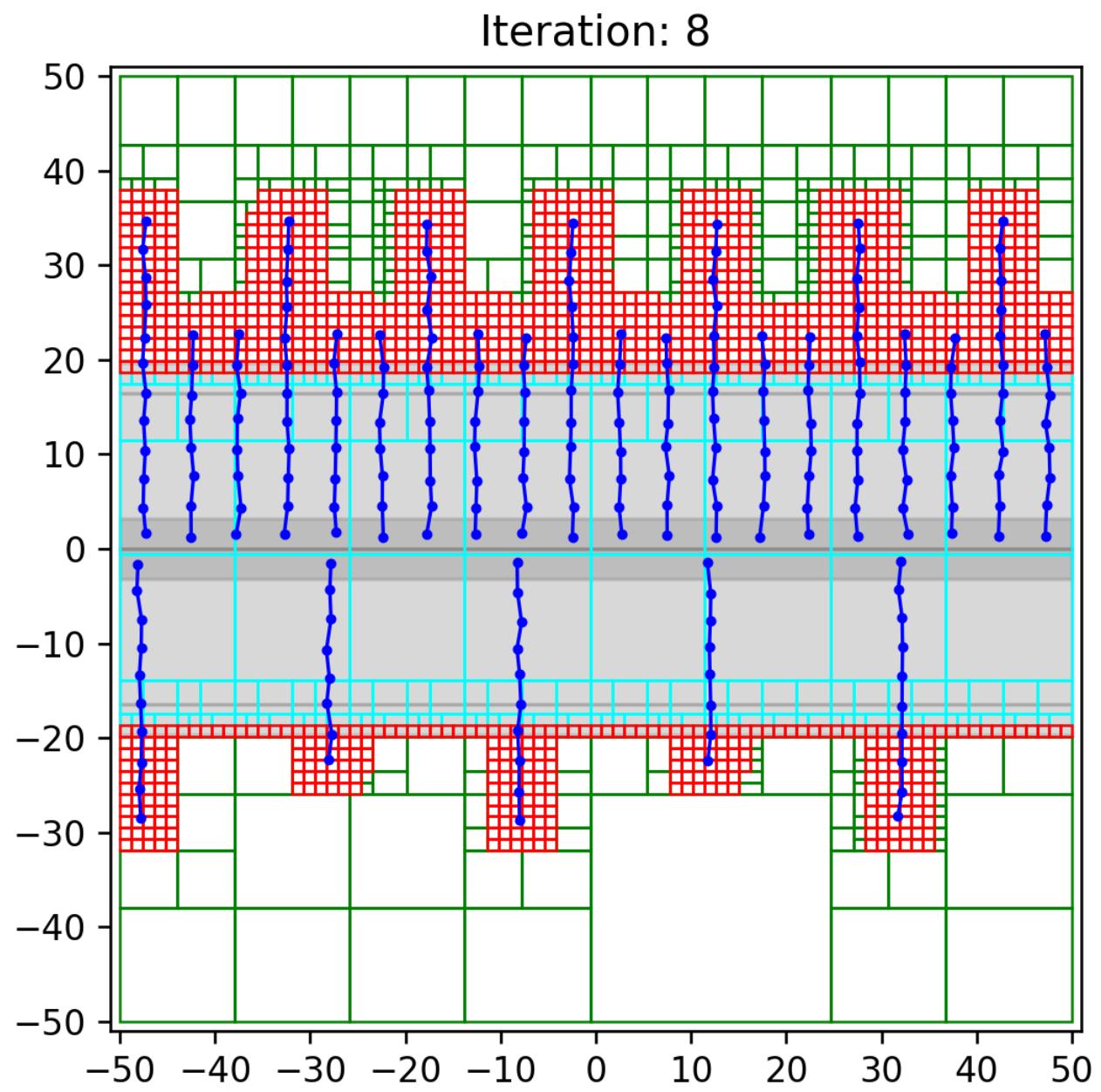
- **Lipids**

Dots:

- Lipid beads
 - Beads used for overlap checks

Squares

- Cell with overlapping beads
 - Cell used for overlap checks
 - "Free" cell
 - Cell completely contained in hydrophobic area
 - Cell partially contained in hydrophobic area while not overlapping with beads
 - Cells removed due to size restrictions



2D EXAMPLE

Lines:

- **Lipids**

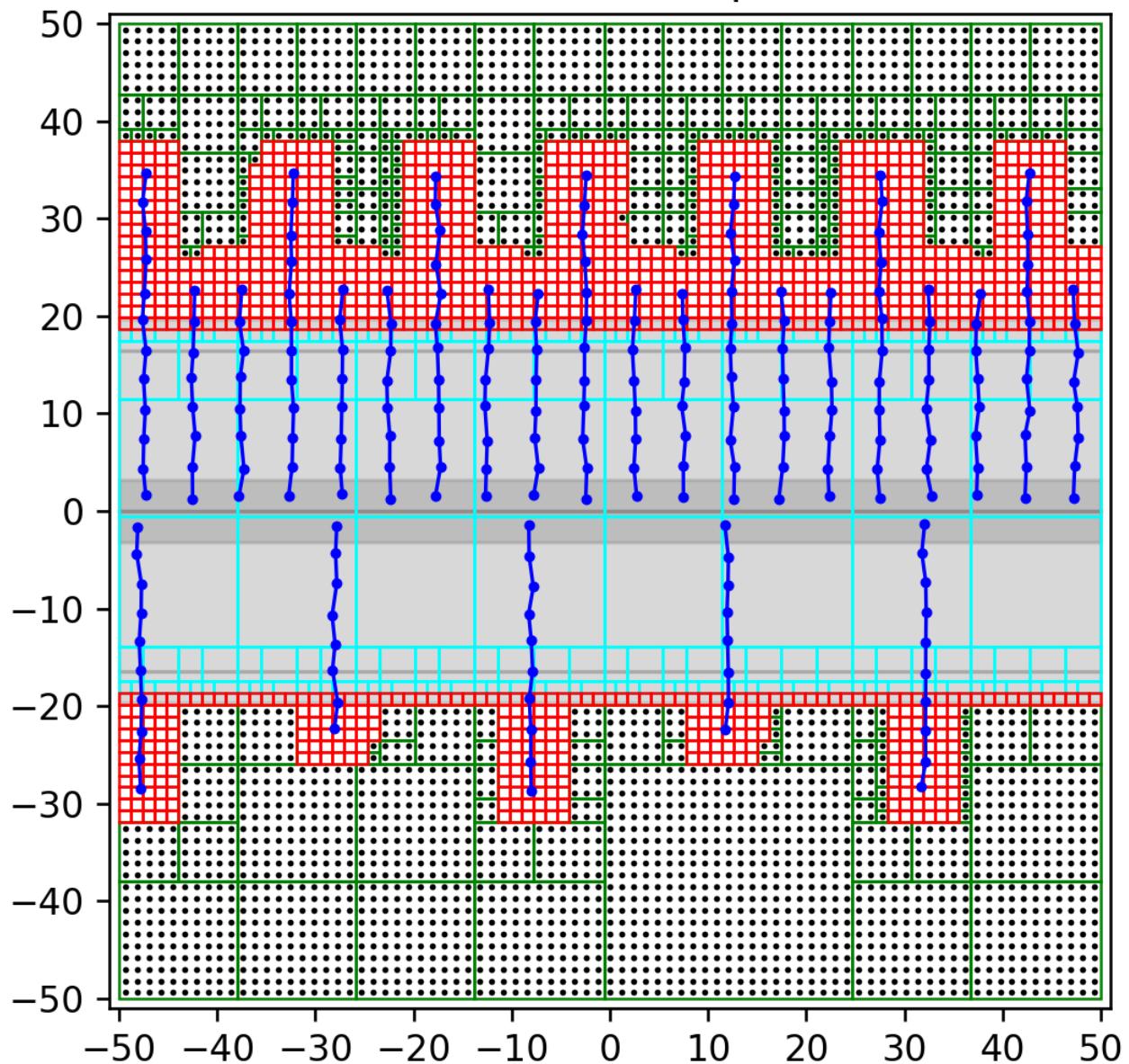
Dots:

- **Lipid beads**
- **Beads used for overlap checks**
- **Grid points available for solvent insertion**
 - 3039 grid points available down from 6889

Squares

- **Cell with overlapping beads**
- **Cell used for overlap checks**
- **"Free" cell**
- **Cell completely contained in hydrophobic area**
- **Cell partially contained in hydrophobic area while not overlapping with beads**
- **Cells removed due to size restrictions**

Potential solvent points



2D EXAMPLE

—

Lines:

- Lipids

Dots:

- Lipid beads
- Beads used for overlap checks
- Water beads - NA beads - CL beads
 - 1000 water, 15 NA, 20 CL inserted

Squares

- Cell with overlapping beads
- Cell used for overlap checks
- "Free" cell
- Cell completely contained in hydrophobic area
- Cell partially contained in hydrophobic area while not overlapping with beads
- Cells removed due to size restrictions

Inserted solvent

