Implementing Molecular Hydrophobicity Potential Measurment for the Analysis of Dynamic Biomolecular Interactions

Peleg Bar Sapir[1]

Under supervision of
Prof. Maria Andrea Mroginski[2]

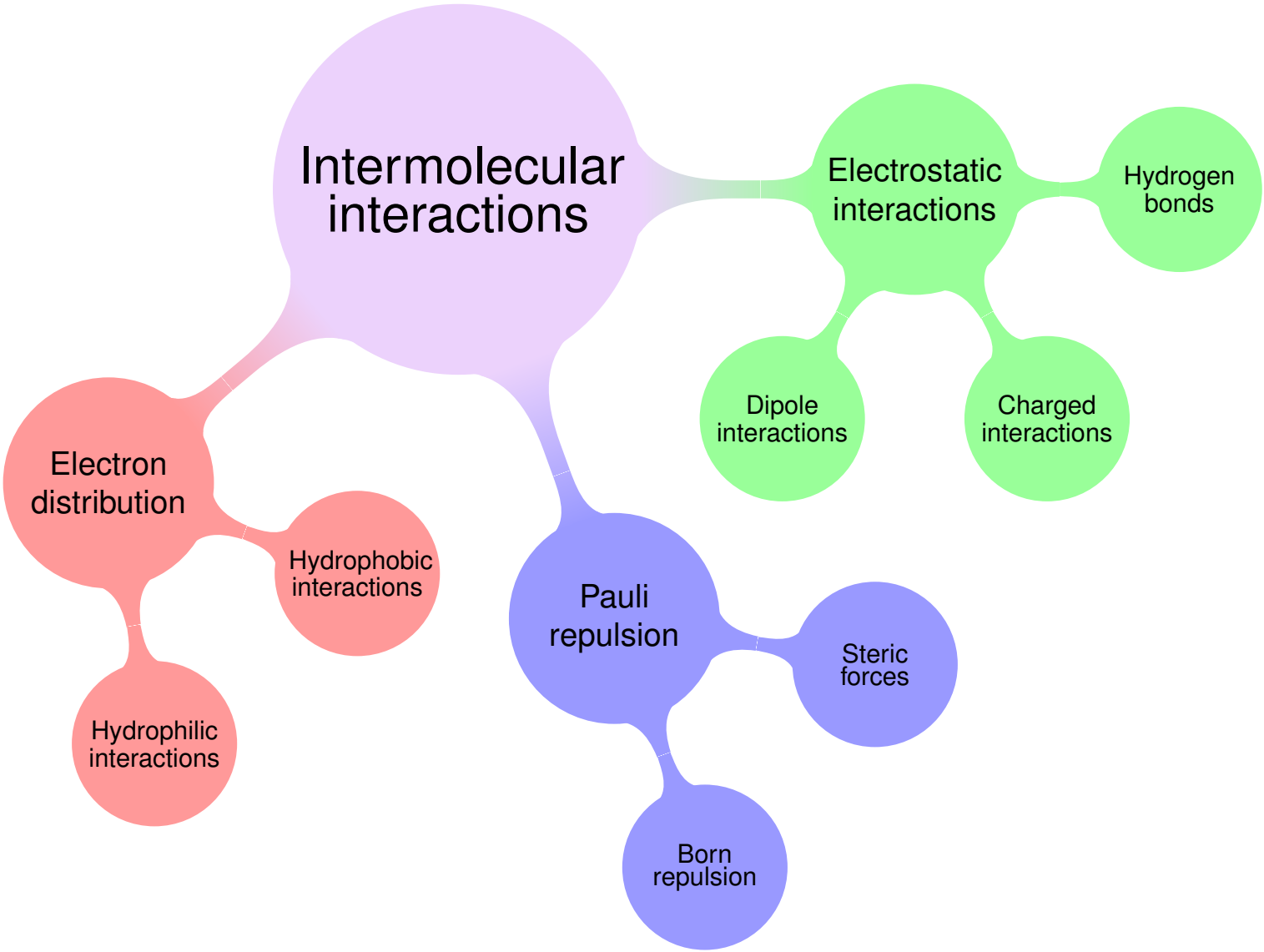
[1]Freie Universität Berlin
[2]Techniche Universität Berlin

February 18, 2018

Outline

Introduction
Hydrophobicity and log P
Partition Coefficient
Molecular Hydrophobicity Potential
What is it?
Potential
General form
Force Constants
Distance function
Surface
Solvent accesible surface
Evenly distributed points
Integration
Program
What are we interested in?
Program Specifications
Results
Validation via Known log p Values
An Example System

Hydrophobicity and log P



Partition Coefficient

Definition

The ratio of concentrations of a compound in a mixture of two immiscible phases at equilibrium

Partition Coefficient

Definition

The ratio of concentrations of a compound in a mixture of two immiscible phases at equilibrium

- Commonly used: water and octanol

Partition Coefficient

Definition

The ratio of concentrations of a compound in a mixture of two immiscible phases at equilibrium

- Commonly used: water and octanol
- Can be measured at an ionized or unionized state

Partition Coefficient

Definition

The ratio of concentrations of a compound in a mixture of two immiscible phases at equilibrium

- Commonly used: water and octanol
- Can be measured at an ionized or unionized state
- $\log P_{\text{octanol/water}} = \log \left(\frac{[\text{solute}]_{\text{water}}}{[\text{solute}]_{\text{octanol}}} \right)$

Partition Coefficient

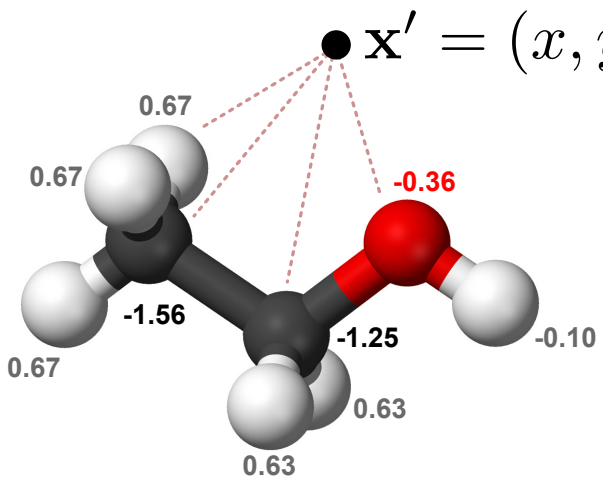
Definition

The ratio of concentrations of a compound in a mixture of two immiscible phases at equilibrium

- Commonly used: water and octanol
- Can be measured at an ionized or unionized state
- $\log P_{\text{octanol/water}} = \log \left(\frac{[\text{solute}]_{\text{water}}}{[\text{solute}]_{\text{octanol}}} \right)$
- Hydrophobicity increases with the (common) $\log P$

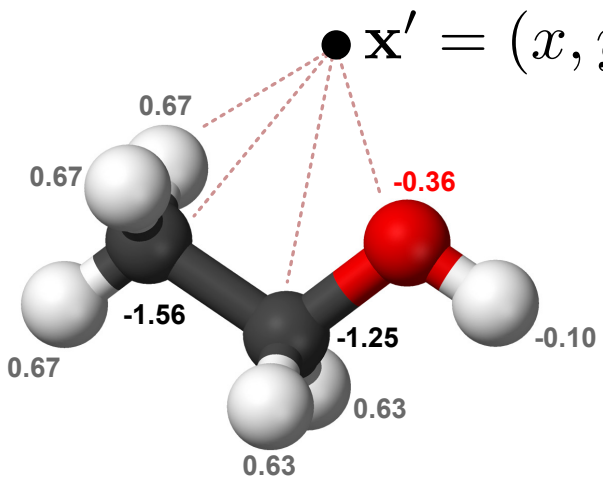
What is MHP

The MHP Formula



$$\text{MHP}\left(\mathbf{x}'\right) = \sum_{i=1}^{k} \left[f_i \cdot D\left(\mathbf{x} - \mathbf{x}'_i\right) \right]$$

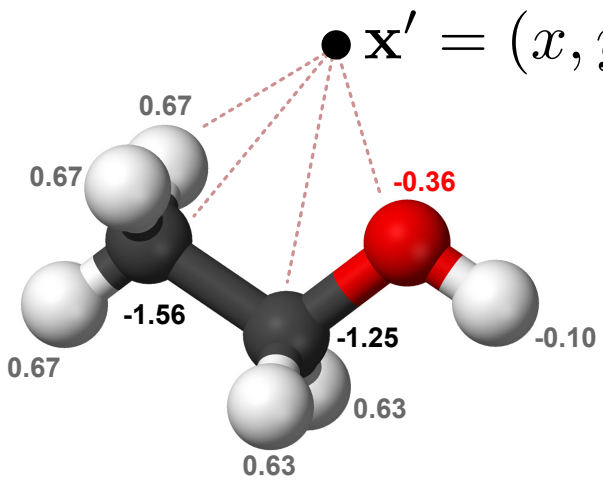
The MHP Formula



$$\text{MHP}\left(\mathbf{x}'\right) = \sum_{i=1}^{k} \left[f_i \cdot D\left(\mathbf{x} - \mathbf{x}'_i\right) \right]$$

Summing over all atoms

The MHP Formula

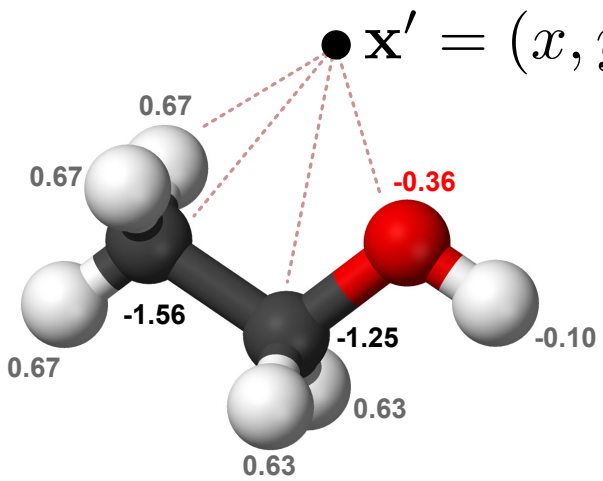


$$\text{MHP}\left(\mathbf{x}'\right) = \sum_{i=1}^{k} \left[f_i \cdot D\left(\mathbf{x} - \mathbf{x}'_i\right) \right]$$

Summing over all atoms

Force constants

The MHP Formula



$$\text{MHP}\left(\mathbf{x}'\right) = \sum_{i=1}^{k} \left[f_i \cdot D\left(\mathbf{x} - \mathbf{x}'_i\right) \right]$$

Summing over all atoms

Force constants

Distance function

Force Constants - Carbon

Carbon atom contribution to hydrophobicity[1]

Type	Description	f_i value
	Carbon in:	
1	CH_3R	-1.5603
3	CHR_3	-0.6681
7	CH_2X_2	-1.0305
13	RCX_3	0.7894
17	$=CR_2$	0.0383
24	$R--CH--R$	-0.3251
25	$R--CR--R$	0.1492
26	$R--CX--R$	0.1539

[1] Source: Ghose et al, J. Phys. Chem. A 1998, 102, 3762-3772

Force Constants - Hydrogen

Hydrogen atom contribution to hydrophobicity[2]

Type	Description	f_i value
	Hydrogen attached to:	
46	C_{sp^3}, no X in α	0.7341
47	C_{sp^2}	0.6301
50	X	-0.1036
52	C_{sp^3}, 1 X in α	0.6666
54	C_{sp^3}, 3 X in α	0.6338

[2]Source: Ghose et al, J. Phys. Chem. A 1998, 102, 3762-3772

Force Constants - Oxygen

Oxygen atom contribution to hydrophobicity[3]

Type	Description	f_i value
	Oxygen in:	
56	Alcohol	-0.3567
57	Phenol, enol, carboxyl OH	-0.0127
58	Ketone	-0.0233
61	Nitro, N-oxides	1.0520
62	O^-	-0.7941

[3] Source: Ghose et al, J. Phys. Chem. A 1998, 102, 3762-3772

Force Constants - Various

Various atom contribution to hydrophobicity[4]

Type	Description	f_i value
66	Primary amine	-0.5427
67	Secondary amine	-0.3168
81	F attached to $\mathrm{C_{sp^3}}$	0.4797
106	S in $\mathrm{R{-}SH}$ (thiol)	1.0520
119	P in $\mathrm{PR_3}$ (phosphine)	-0.7941

[4] Source: Ghose et al, J. Phys. Chem. A 1998, 102, 3762-3772

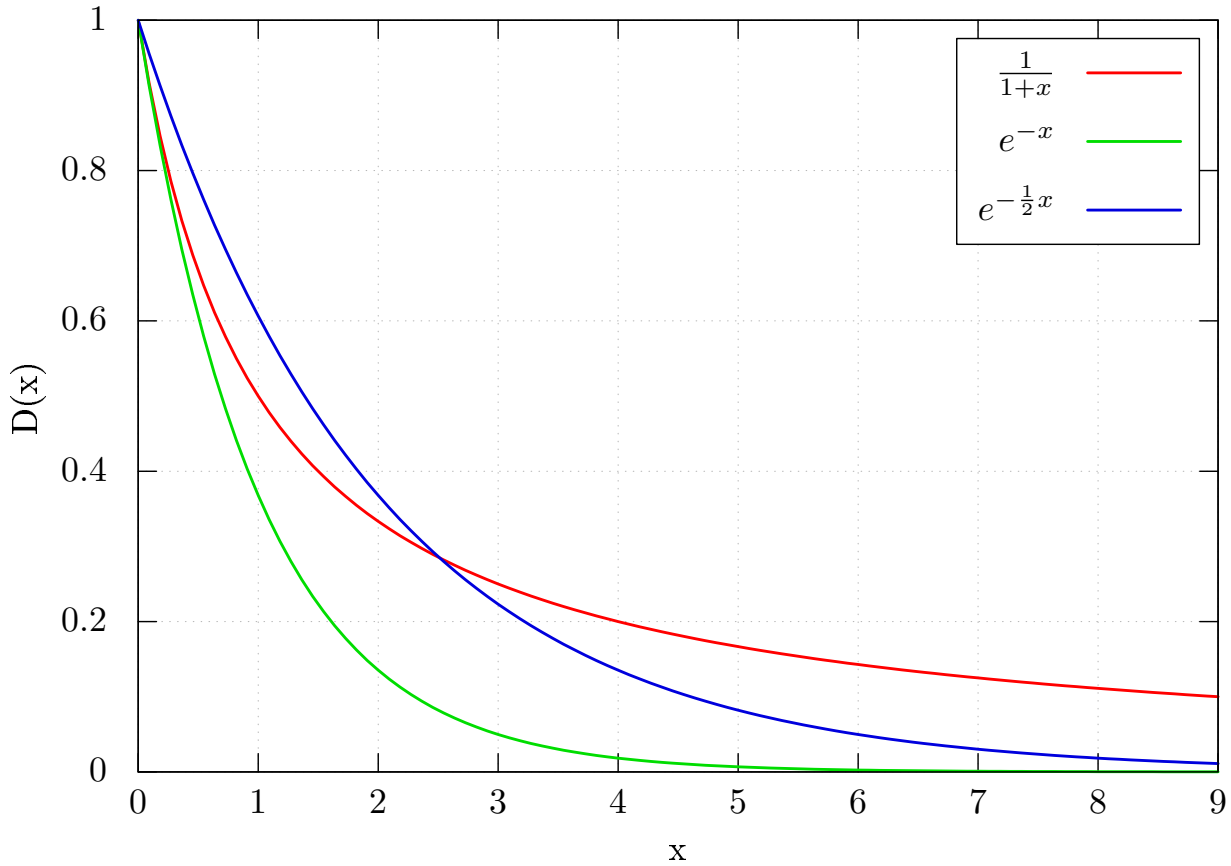
Distance function

Audry form

$$D(x) = \frac{1}{1+x}$$

Exponential decay form

$$D(x) = e^{-\alpha x}$$

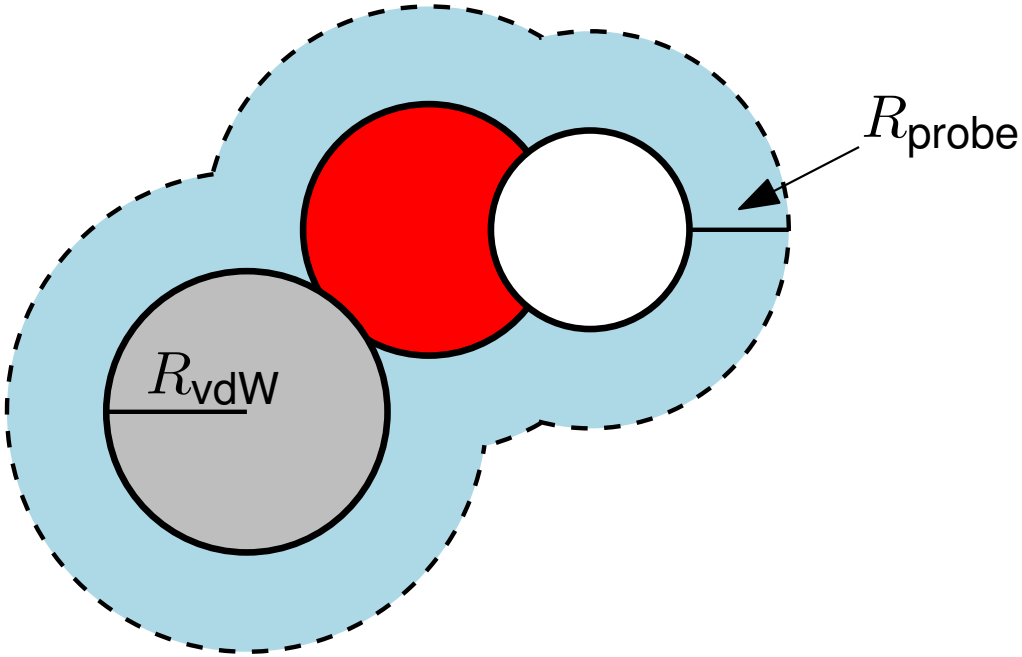


Solvent accesible surface

- The surface around a molecule accesible to solvent molecules

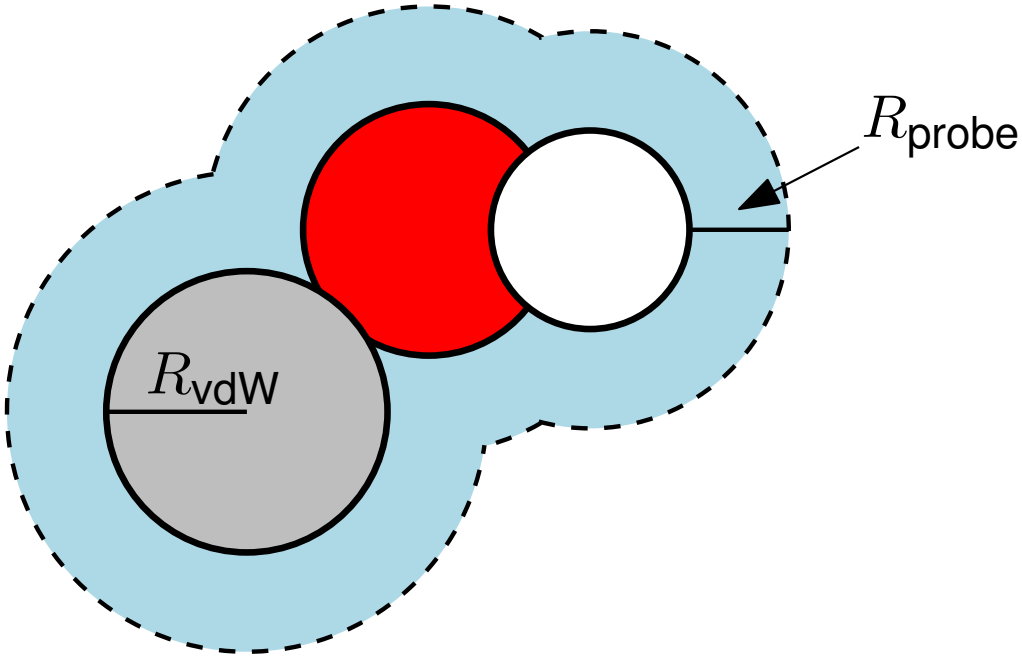
Solvent accesible surface

- The surface around a molecule accesible to solvent molecules



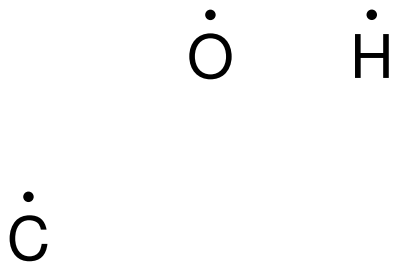
Solvent accesible surface

- The surface around a molecule accesible to solvent molecules

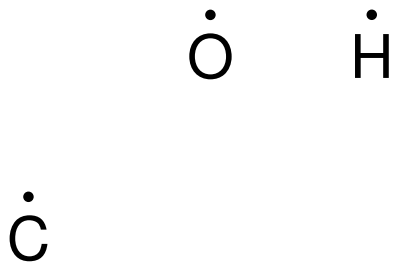


- For water molecules usually $r = 1.4\,[\text{Å}]$

How to Create the Solvent Accesible Surface?

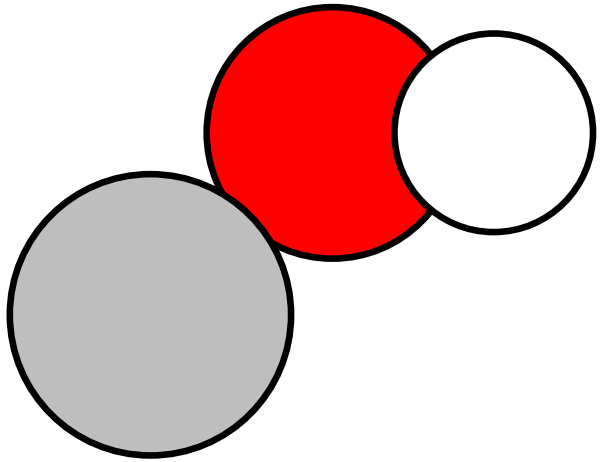


How to Create the Solvent Accesible Surface?



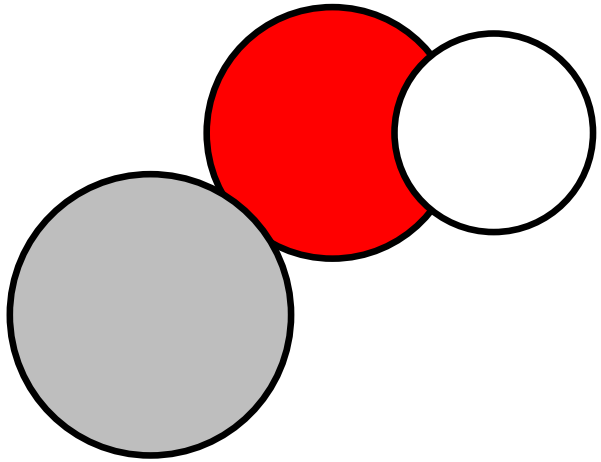
1. Take all atoms with their vdW-radii

How to Create the Solvent Accesible Surface?



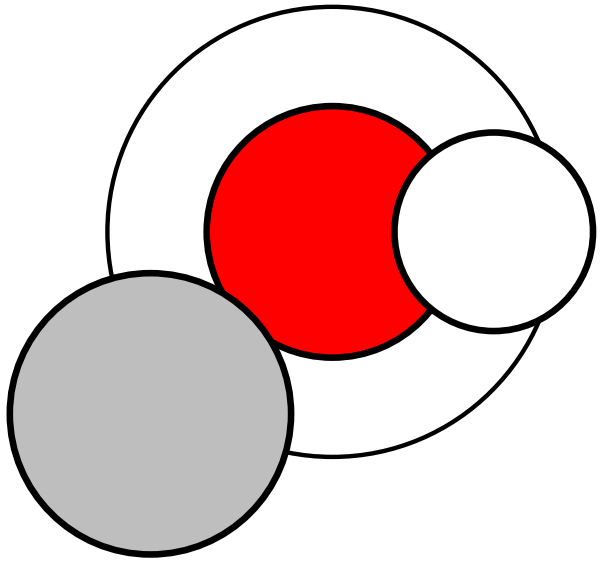
1. Take all atoms with their vdW-radii

How to Create the Solvent Accesible Surface?



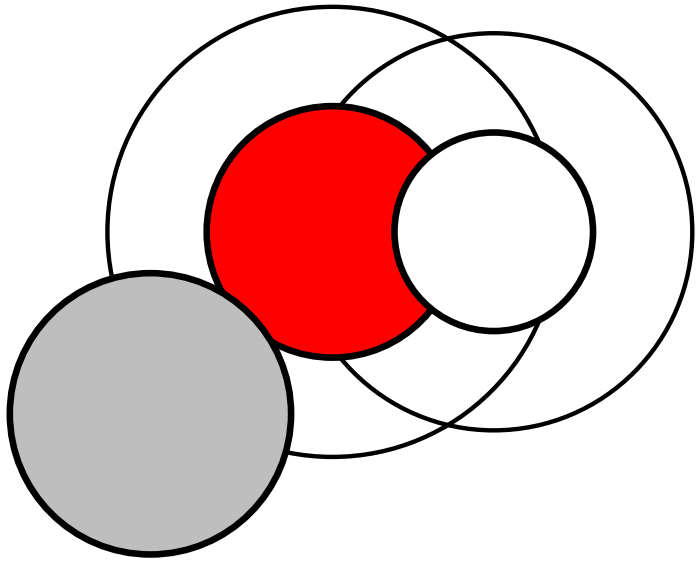
1. Take all atoms with their vdW-radii
2. Create spheres around all atoms with $R^i = R^i_{\text{vdw}} + R_{\text{probe}}$

How to Create the Solvent Accesible Surface?



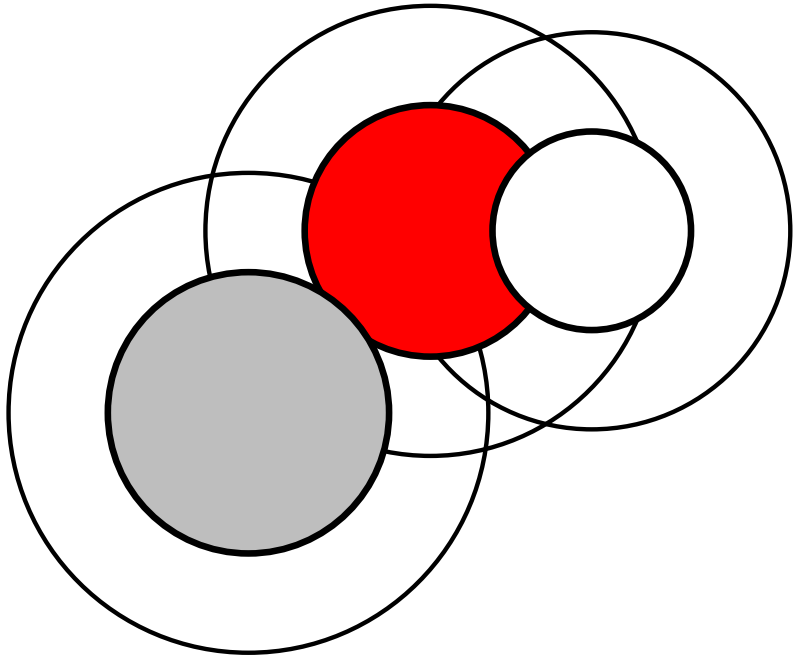
1. Take all atoms with their vdW-radii
2. Create spheres around all atoms with $R^i = R^i_{\text{vdw}} + R_{\text{probe}}$

How to Create the Solvent Accesible Surface?



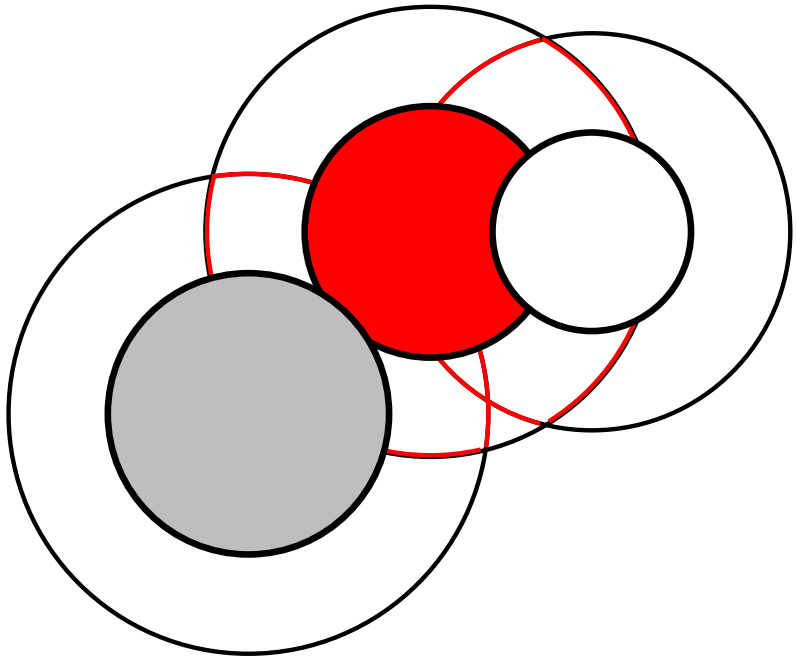
1. Take all atoms with their vdW-radii
2. Create spheres around all atoms with $R^i = R^i_{\text{vdw}} + R_{\text{probe}}$

How to Create the Solvent Accesible Surface?



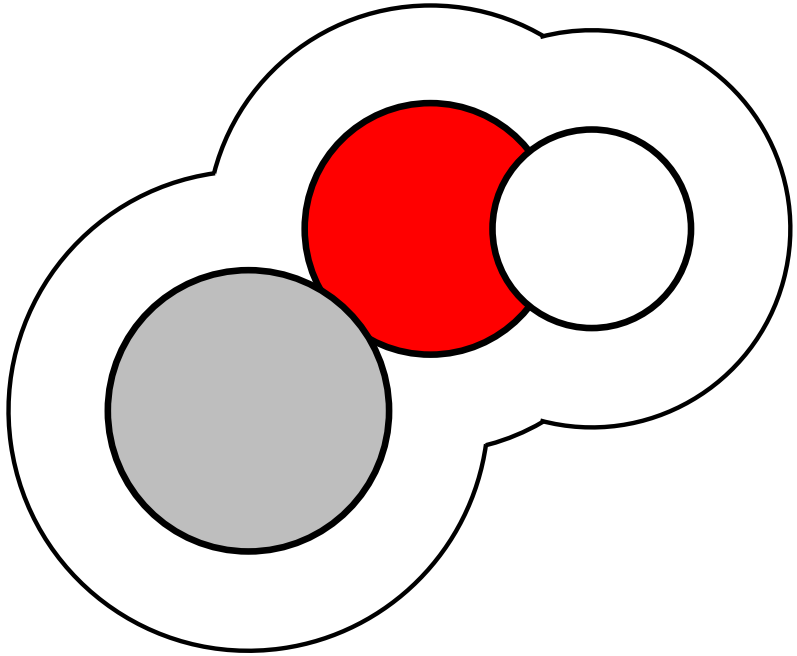
1. Take all atoms with their vdW-radii
2. Create spheres around all atoms with $R^i = R^i_{\text{vdw}} + R_{\text{probe}}$

How to Create the Solvent Accesible Surface?



1. Take all atoms with their vdW-radii
2. Create spheres around all atoms with $R^i = R^i_{\text{vdw}} + R_{\text{probe}}$
3. Delete all points that are "burried" in other extended spheres (i.e. $\Delta\left(p^i, c^j\right) \leq R^j + R_{\text{probe}}$)

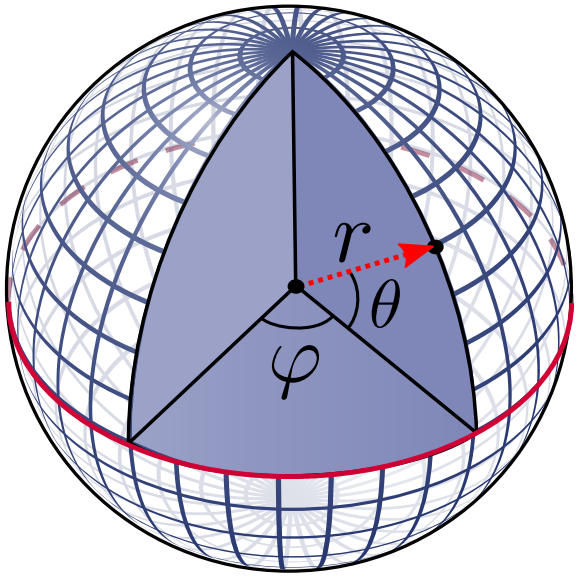
How to Create the Solvent Accesible Surface?



1. Take all atoms with their vdW-radii
2. Create spheres around all atoms with $R^i = R^i_{\text{vdw}} + R_{\text{probe}}$
3. Delete all points that are "burried" in other extended spheres (i.e. $\Delta\left(p^i, c^j\right) \leq R^j + R_{\text{probe}}$)
4. The remaining surface is the solvent-accesible surface of the molecule

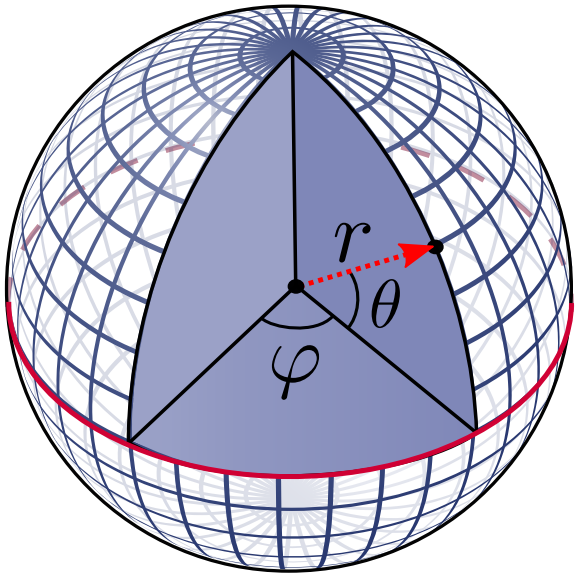
Evenly distributed points

How to distribute N points on a surface of a sphere?



Evenly distributed points

How to distribute N points on a surface of a sphere?

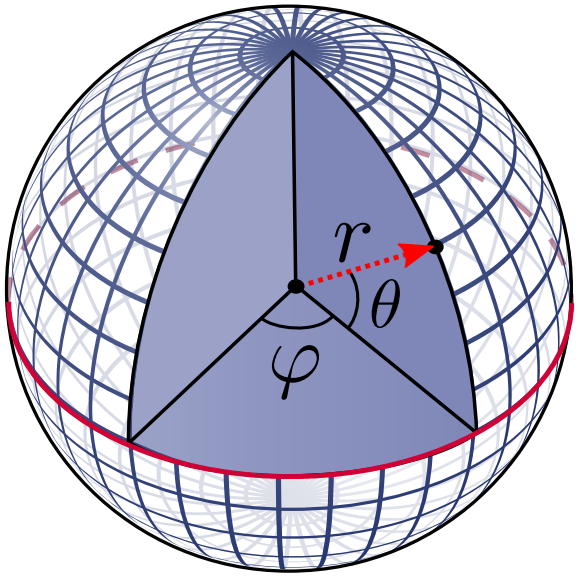


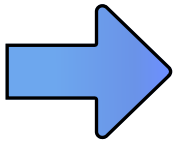
$$\varphi_i = i \cdot \frac{2\pi}{N}$$

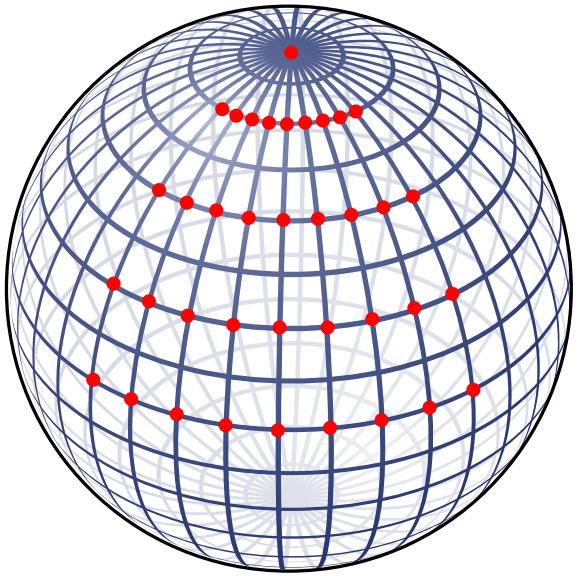
$$\theta_j = j \cdot \frac{\pi}{N}$$

Evenly distributed points

How to distribute N points on a surface of a sphere?





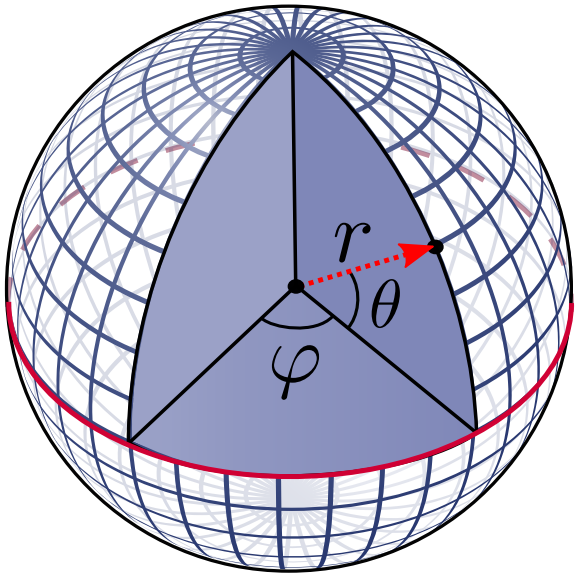


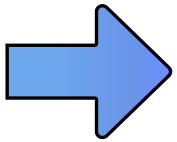
$$\varphi_i = i \cdot \frac{2\pi}{N}$$

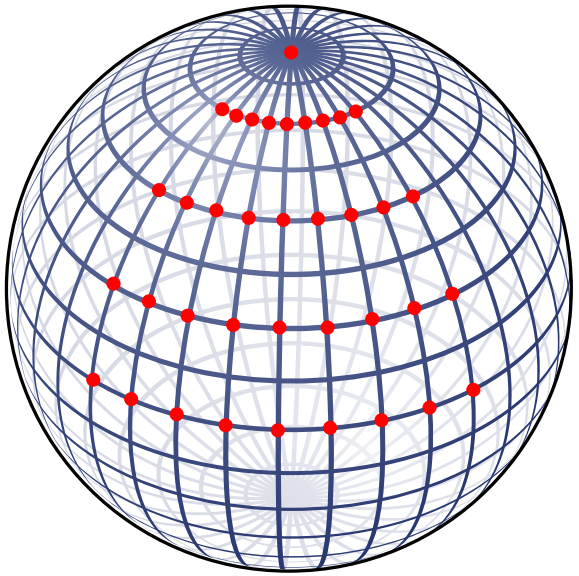
$$\theta_j = j \cdot \frac{\pi}{N}$$

Evenly distributed points

How to distribute N points on a surface of a sphere?







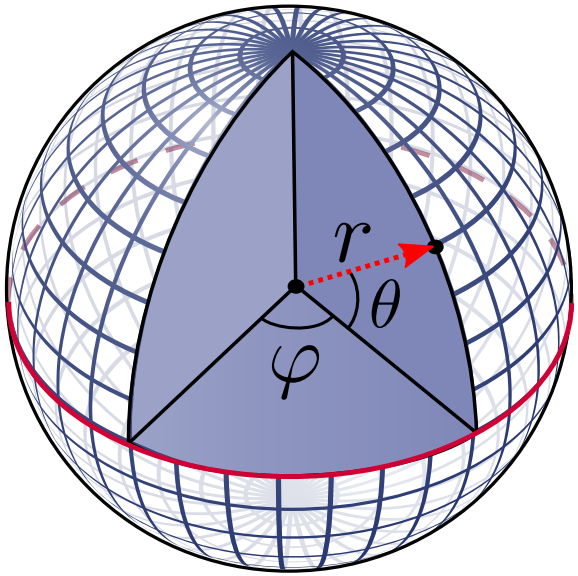
$$\varphi_i = i \cdot \frac{2\pi}{N}$$

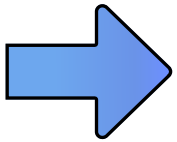
$$\theta_j = j \cdot \frac{\pi}{N}$$

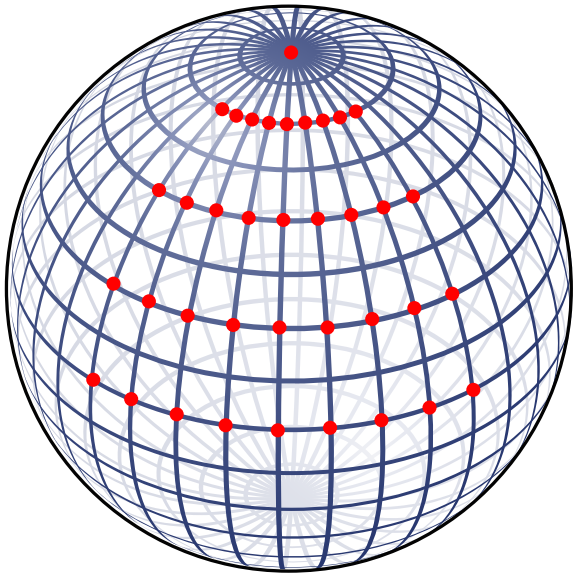
- Points are not evenly distributed

Evenly distributed points

How to distribute N points on a surface of a sphere?







$$\varphi_i = i \cdot \frac{2\pi}{N}$$

$$\theta_j = j \cdot \frac{\pi}{N}$$

- Points are not evenly distributed
- Several points overlap at poles

Evenly distributed points

Solution: Vogel's method

In 2 dimensions:

Evenly distributed points

Solution: Vogel's method

In 2 dimensions:

- Distances: $r_i = \sqrt{\frac{i}{N}}$
- Angle: $\theta_i = \varphi i$

  (φ is the golden ratio!)

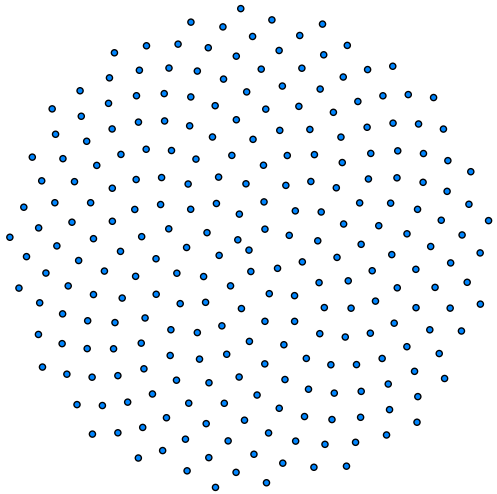
Evenly distributed points

Solution: Vogel's method

In 2 dimensions:

- Distances: $r_i = \sqrt{\frac{i}{N}}$
- Angle: $\theta_i = \varphi i$

  (φ is the golden ratio!)



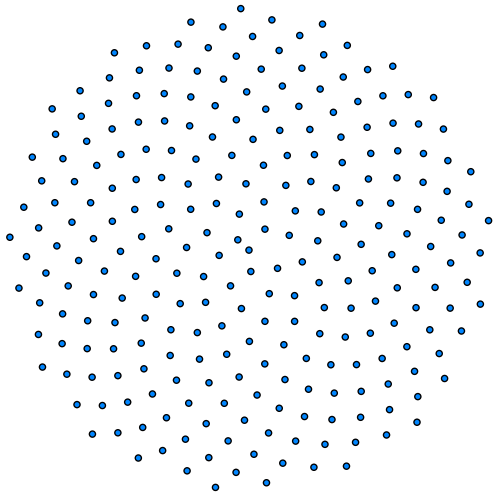
Evenly distributed points

Solution: Vogel's method

In 2 dimensions:

- Distances: $r_i = \sqrt{\frac{i}{N}}$
- Angle: $\theta_i = \varphi i$

  (φ is the golden ratio!)



In 3 dimensions (cylindrical coordinates):

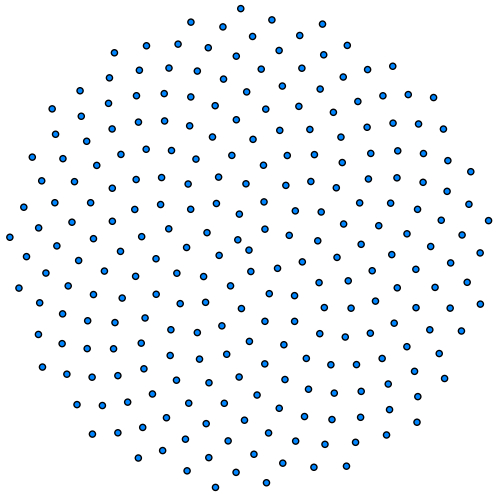
Evenly distributed points

Solution: Vogel's method

In 2 dimensions:

- Distances: $r_i = \sqrt{\frac{i}{N}}$
- Angle: $\theta_i = \varphi i$

  (φ is the golden ratio!)



In 3 dimensions (cylindrical coordinates):

- Distances: $z_i = \left(1 - \frac{1}{N}\right)\left(1 - \frac{2i}{N-1}\right)$
- Angles:

  $\theta_i = \varphi i,\ \rho_i = \sqrt{1 - z_i^2}$

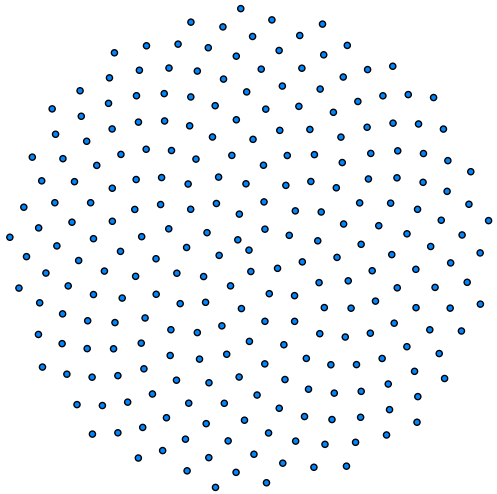
Evenly distributed points

Solution: Vogel's method

In 2 dimensions:

- Distances: $r_i = \sqrt{\frac{i}{N}}$
- Angle: $\theta_i = \varphi i$

  (φ is the golden ratio!)



In 3 dimensions (cylindrical coordinates):

- Distances: $z_i = \left(1 - \frac{1}{N}\right)\left(1 - \frac{2i}{N-1}\right)$
- Angles:

  $\theta_i = \varphi i,\ \rho_i = \sqrt{1 - z_i^2}$

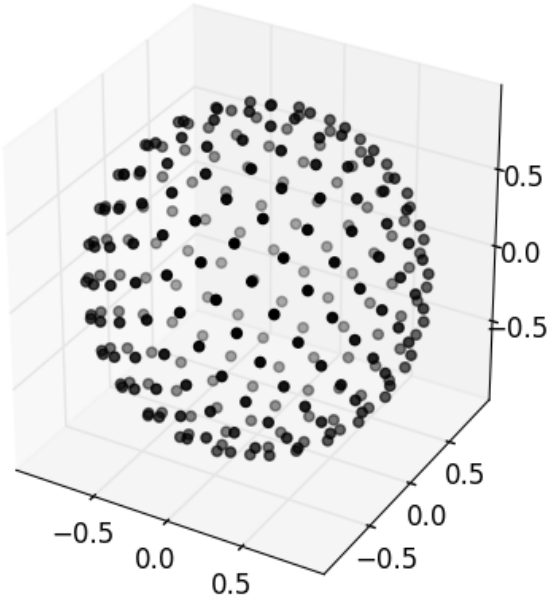


Image source: Marmakoide's Blog

Integration

- Each atom's total surface area:
  $V^a = 4\pi \left(R^a_{\text{vdW}} + R_{\text{probe}}\right)^2$

Integration

- Each atom's total surface area:
  $V^a = 4\pi \left(R^a_{\text{vdW}} + R_{\text{probe}}\right)^2$
- The surface is represented by N points

Integration

- Each atom's total surface area:
  $V^a = 4\pi \left(R^a_{\text{vdW}} + R_{\text{probe}}\right)^2$
- The surface is represented by N points
- Meaning: each point has $V^a_j = \frac{4\pi}{N} \left(R^i_{\text{vdW}} + R_{\text{probe}}\right)^2$

Integration

- Each atom's total surface area:
  $V^a = 4\pi \left(R^a_{\text{vdW}} + R_{\text{probe}}\right)^2$
- The surface is represented by N points
- Meaning: each point has $V^a_j = \frac{4\pi}{N} \left(R^i_{\text{vdW}} + R_{\text{probe}}\right)^2$
- In addition, each point has: MHP^a_j

Integration

- Each atom's total surface area:
  $V^a = 4\pi \left(R^a_{\text{vdW}} + R_{\text{probe}}\right)^2$
- The surface is represented by N points
- Meaning: each point has $V^a_j = \frac{4\pi}{N} \left(R^i_{\text{vdW}} + R_{\text{probe}}\right)^2$
- In addition, each point has: MHP^a_j

Therefore, each atom has a total MHP of:

$$\text{MHP}^a = \frac{4\pi}{N} \sum_{j=0}^{M} \text{MHP}^a_j$$

Program Specifications

- Written in **Python3**, utylizing **ProDy**

Program Specifications

- Written in **Python3**, utylizing **ProDy**
- Heavy calculation written in **Cython**

Program Specifications

- Written in **Python3**, utylizing **ProDy**
- Heavy calculation written in **Cython**
- Uses neighbor cells implementation for faster calculation

Program Specifications

- Written in **Python3**, utylizing **ProDy**
- Heavy calculation written in **Cython**
- Uses neighbor cells implementation for faster calculation
- Uses PSF, PDB and DCD files

Program Specifications

- Written in **Python3**, utylizing **ProDy**
- Heavy calculation written in **Cython**
- Uses neighbor cells implementation for faster calculation
- Uses PSF, PDB and DCD files
- Generates a PDB output, MHP values in **beta** column

Program Options

- Input: PSF + PDB or DCD

Program Options

- Input: PSF + PDB or DCD
- Subselection (optional): Atomic selection (like vmd)

Program Options

- Input: PSF + PDB or DCD
- Subselection (optional): Atomic selection (like vmd)
- Number of points per atom (default: 64)

Program Options

- Input: PSF + PDB or DCD
- Subselection (optional): Atomic selection (like vmd)
- Number of points per atom (default: 64)
- Solvent probe radius (defalt: $1.4\mathring{A}$)

Program Options

- Input: PSF + PDB or DCD
- Subselection (optional): Atomic selection (like vmd)
- Number of points per atom (default: 64)
- Solvent probe radius (defalt: $1.4\mathring{A}$)
- Cutoff distance for distance function (default: $4\mathring{A}$)

Program Options

- Input: PSF + PDB or DCD
- Subselection (optional): Atomic selection (like vmd)
- Number of points per atom (default: 64)
- Solvent probe radius (defalt: $1.4\mathring{A}$)
- Cutoff distance for distance function (default: $4\mathring{A}$)
- Frame range (if DCD)

Validation via Known $\log P$ Values

An Example System