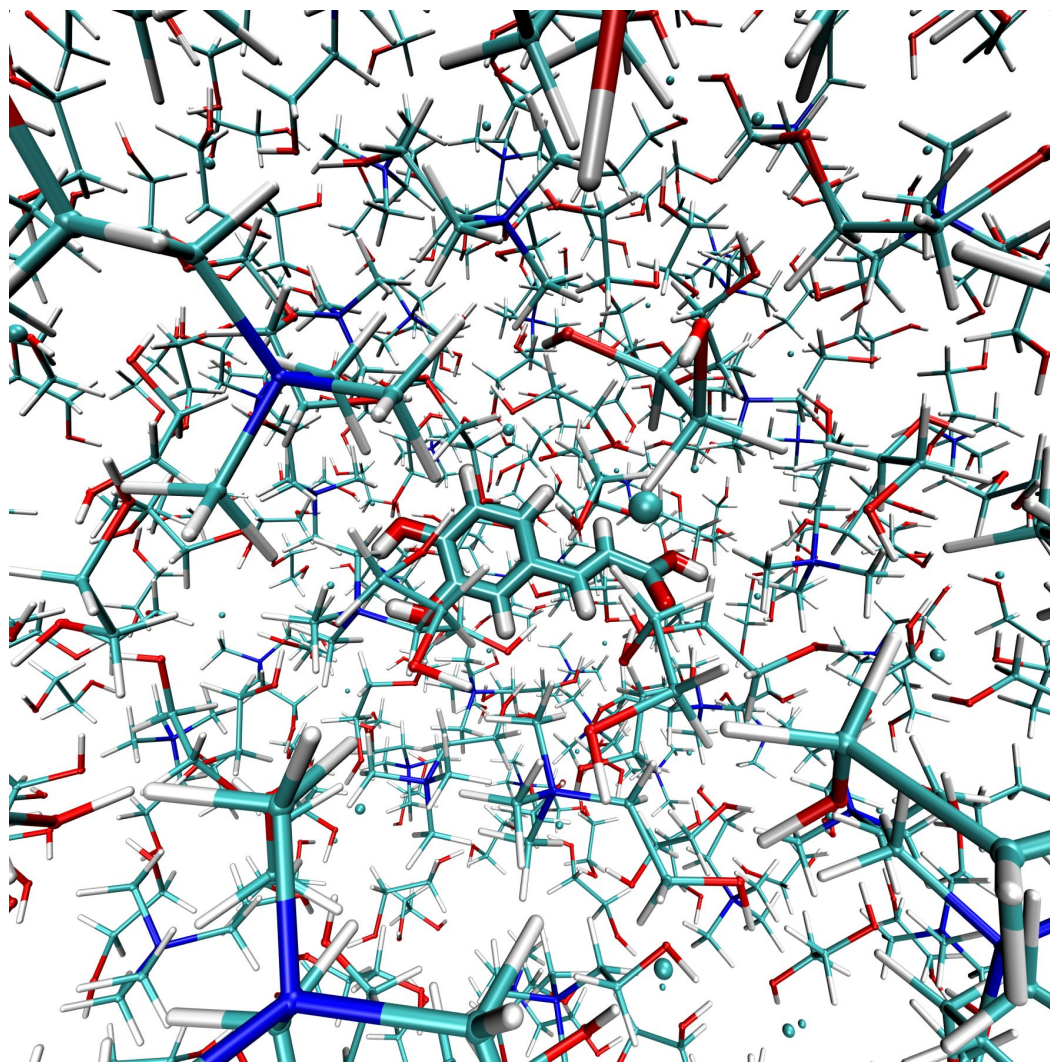


SCUOLA
NORMALE
SUPERIORE

Lab Session 1

Goal: Parameterizing a Fluctuating Charges forcefield for DES



$$\mathcal{E} = E_{QM} + E_{MM} + E_{QM/MM}^{int}$$

$$E_{QM/MM}^{int} = E_{QM/MM}^{ele} + E_{QM/MM}^{pol} + E_{QM/MM}^{ex-rep} + E_{QM/MM}^{disp} + E_{QM/MM}^{CT}$$

Classical Nature

- **ele** Electrostatics
- **pol** Polarization

QM Nature

- **ex-rep** Exchange-repulsion
- **disp** Dispersion
- **CT** Charge-Transfer

QM/MM: Electrostatic Embedding

Fixed charges are placed on each atom of the MM portion and the Coulomb interaction is added to the QM Hamiltonian

$$E_{QM/MM}^{\text{int}} = E_{QM/MM}^{\text{ele}} = \sum_{j=1}^{N_{\text{MM}}} \int_{\mathbf{R}^3} \frac{\rho_{\text{QM}}(\mathbf{r}) q_j}{|\mathbf{r} - \mathbf{r}_j|} d\mathbf{r}$$

QM/MM: Polarizable Embedding

Mutual polarization between QM and MM portions is included in the calculation. Several models to include such a term (Fluctuating Charges, Induced Dipoles, Drude Oscillators, multipole expansions, AMOEBA).

$$E_{QM/MM}^{\text{int}} = E_{QM/MM}^{\text{ele}} + E_{QM/MM}^{\text{pol}}$$

C. Cappelli *Int. J. Quantum Chem.*, **2016**, 116, 1532

S. Caprasecca, C. Curutchet, B. Mennucci *J. Chem. Theory Comput.*, **2012**, 8, 4462

E. Boulanger, W. Thiel *J. Chem. Theory Comput.*, **2012**, 8, 4527

D. Loco, *et al.* *J. Chem. Theory Comput.*, **2016**, 12, 3654

Fluctuating Charges forcefield – Conceptual DFT

Taylor Expansion of the energy in terms of charges:

$$E = E_0 + \frac{\partial E}{\partial q} q + \frac{1}{2} \frac{\partial^2 E}{\partial q^2} q^2$$

Electronegativity and Chemical Hardness definition:

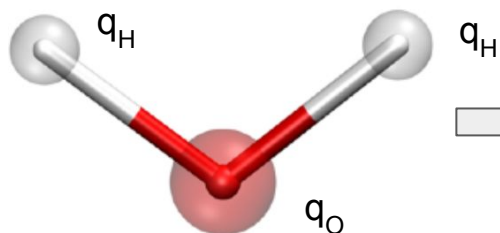
$$\chi = \left. \frac{\partial E}{\partial q} \right|_{q=0} \quad 2\eta = \left. \frac{\partial^2 E}{\partial q^2} \right|_{q=0}$$

For a molecular system:

$$E = E_0 + \sum_i \left[\chi_i q_i + \eta_i q_i^2 + \sum_{j>i} J_{ij} q_i q_j \right]$$

Electronegativity Equalization Principle (EEP)

At equilibrium, the instantaneous electronegativity $\tilde{\chi}$ of each atom has the same value.



Depend on atomic electronegativity (χ) and chemical hardness (η)

$$E^{tot} = E_{QM} + E_{MM} + E_{QM/MM}^{int}$$

$$E^{tot} = E_{QM}[\rho] + \frac{1}{2} \mathbf{q}^\dagger \mathbb{J} \mathbf{q} + \mathbf{q}^\dagger \boldsymbol{\chi} + \mathbf{q}^\dagger \boldsymbol{\lambda} + \mathbf{q}^\dagger \boldsymbol{\Phi}[\rho]$$

$E_{FQ}(\mathbf{q}, \boldsymbol{\lambda})$

$=$

$\sum_{\alpha,i}$

$q_{\alpha i}$

$\chi_{\alpha i}$

$+$

$\frac{1}{2}$

$\sum_{\alpha,i}$

$\sum_{\beta,j}$

$q_{\alpha i}$

$J_{\alpha i, \beta j}$

$q_{\beta j}$

$+$

\sum_{α}

λ_{α}

\sum_i

$(q_{\alpha i} - Q_{\alpha})$

molecules

atoms

FQ Charges

Atomic Electronegativity

Interaction kernel between the charges.

Lagrangian multiplier to ensure charge conservation

$J_{ij}^{qq} = \frac{\eta_{ij}}{[1 + \eta_{ij}^2 r_{ij}^2]^{\frac{1}{2}}}$

Ohno form

$\eta_{ij} = \frac{\eta_i + \eta_j}{2}$

$J_{ii}^{qq} = 2\eta_i$

$$E^{tot} = E_{QM} + E_{MM} + E_{QM/MM}^{int}$$

$$E^{tot} = E_{QM}[\rho] + \frac{1}{2} \mathbf{q}^\dagger \mathbb{J} \mathbf{q} + \mathbf{q}^\dagger \chi + \mathbf{q}^\dagger \lambda + \mathbf{q}^\dagger \Phi[\rho]$$

QM potential on the FQ positions

$$\Phi[\rho_{QM}](\mathbf{r}_i) \stackrel{def}{=} \underbrace{V_i^N(\mathbf{P})}_{\text{Nuclear}} + \underbrace{V_i^e(\mathbf{P})}_{\text{Electronic}} = \sum_{\zeta=1}^{N_n} \frac{Z_\zeta}{|\mathbf{r}_i - \mathbf{R}_\zeta|} - \int_{\mathbb{R}^3} d\mathbf{r} \frac{\overbrace{\rho^{el}(\mathbf{r})}^{\text{Electron density}}}{|\mathbf{r}_i - \mathbf{r}|}$$

$$E^{tot} = E_{QM} + E_{MM} + E_{QM/MM}^{int}$$

$$E^{tot} = E_{QM}[\rho] + \frac{1}{2} \mathbf{q}^\dagger \mathbb{J} \mathbf{q} + \mathbf{q}^\dagger \chi + \mathbf{q}^\dagger \lambda + \mathbf{q}^\dagger \Phi[\rho]$$

$$\frac{\delta E^{tot}}{\delta \rho(\mathbf{r})} = h_{KS}^0[\rho] + \hat{v}^{emb} = \tilde{F}$$

Fock Matrix

$$\frac{\delta E^{tot}}{\delta \mathbf{q}_\lambda} = \mathbb{D} \mathbf{q}_\lambda + \Phi[\rho] + \chi = 0$$

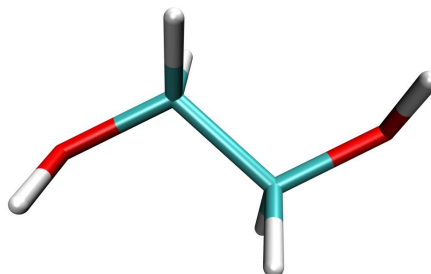
Polarization Equations

Solve with SCF procedure.

$$\mathbb{D} = \begin{bmatrix} \mathbb{J} & \mathbf{1}_\lambda \\ \mathbf{1}_\lambda^\dagger & 0 \end{bmatrix}$$

Fluctuating Charges Force Field: Parameterization

We need to find χ and η for each atomtype of our system



REMEMBER: FQ is a polarizable approach.
Thus how to parameterize it?

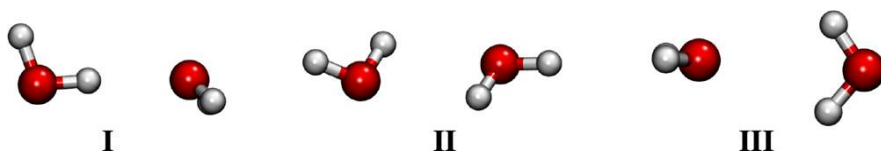
Reproducing
Electrostatic + Polarization
contributions to the interaction energy

Different strategies...

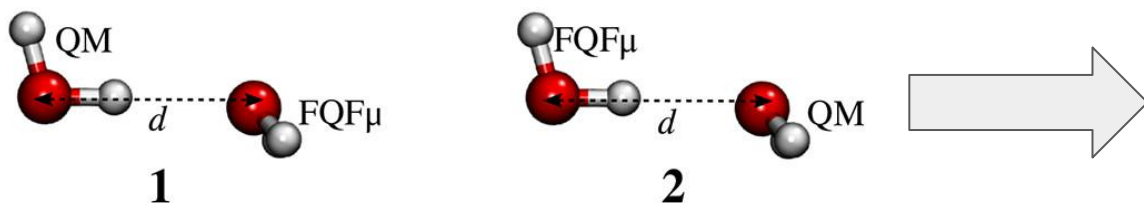
Fluctuating Charges Force Field: Parameterization

We need to find χ and η for each atomtype of our system

Reproduce electrostatic energies obtained by energy Decomposition Analysis



Treat one molecule at the QM level (HF) and the other at the FQ level.

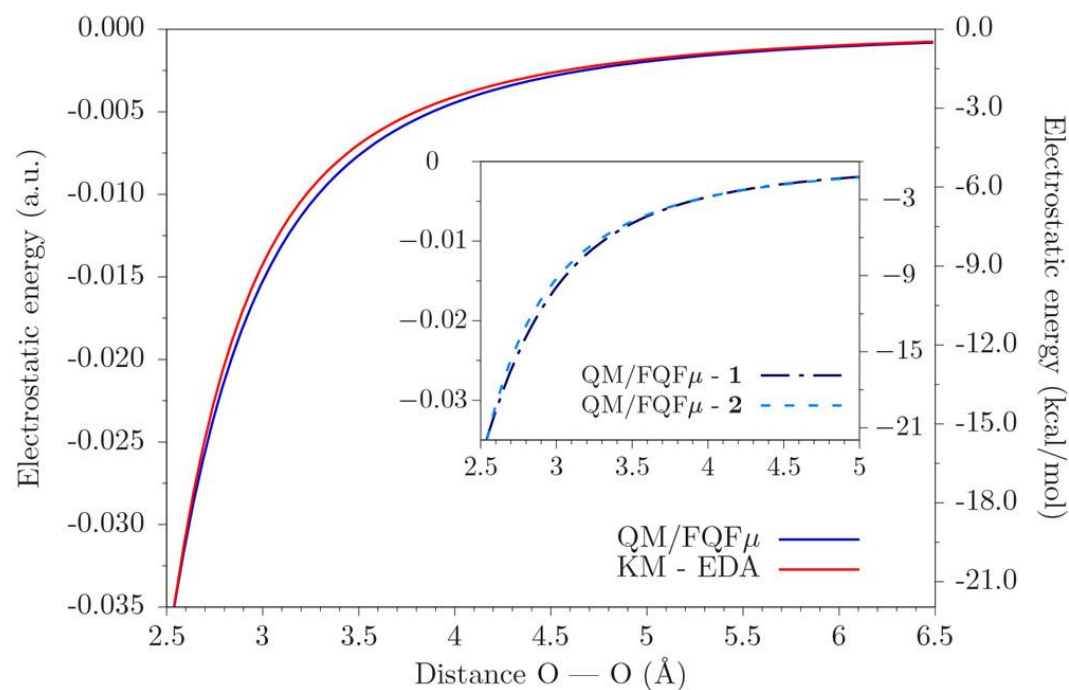
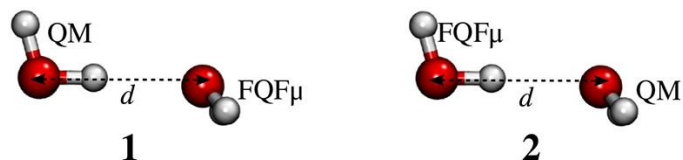


Fit your parameters in order to reproduce the Electrostatic (+ Polarization) energy obtained by Energy decomposition analysis (EDA, in this case KM) for specific geometries.

Fluctuating Charges Force Field: Parameterization

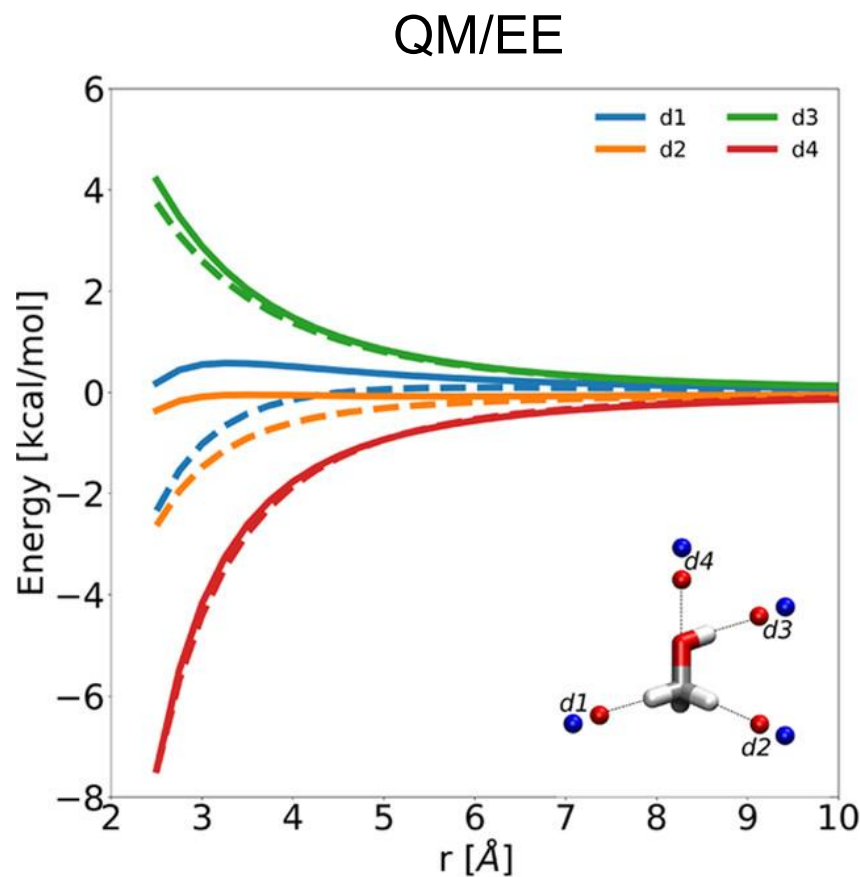
We need to find χ and η for each atomtype of our system

Reproduce electrostatic energies obtained by energy Decomposition Analysis



Fluctuating Charges Force Field: Parameterization

Obtain χ and η by reproducing: 1) QM/EE interaction energies



Treat the molecule at the FQ level of theory and solve the system of equations with an external source of charge (FQ/EE).

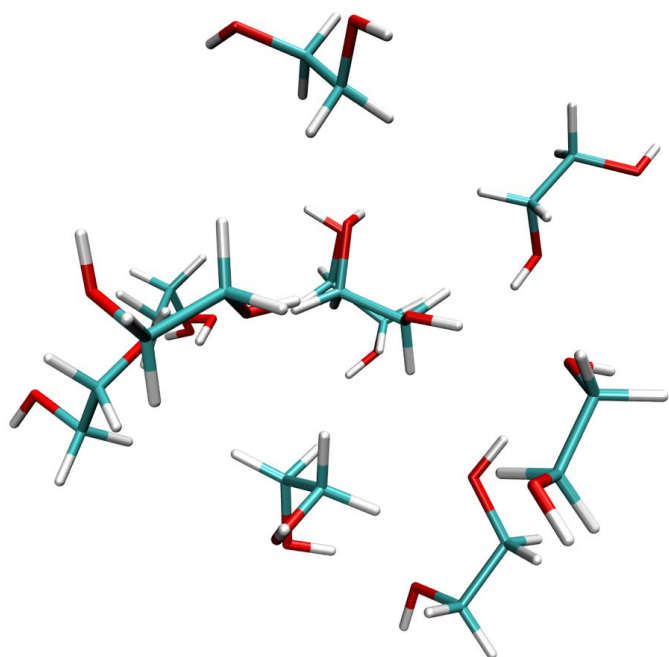
Task: Write the FQ/EE system of equations

Try to reproduce QM/EE results.

Fluctuating Charges Force Field: Parameterization

Obtain χ and η by reproducing: 2) QM static polarizability

QM static polarizability tensor
of clusters of molecules



Treat the cluster of molecules at the FQ level of theory and obtain the polarizability tensor.

GOAL: Try to set the diagonal elements equal to QM results.

$$\begin{pmatrix} \mu_x \\ \mu_y \\ \mu_z \end{pmatrix} = \begin{pmatrix} \alpha_{xx} & \alpha_{xy} & \alpha_{xz} \\ \alpha_{yx} & \alpha_{yy} & \alpha_{yz} \\ \alpha_{zx} & \alpha_{zy} & \alpha_{zz} \end{pmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix}$$

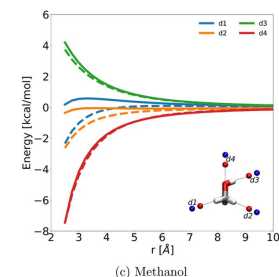
Fluctuating Charges Force Field: Parameterization

Obtain χ and η by reproducing:

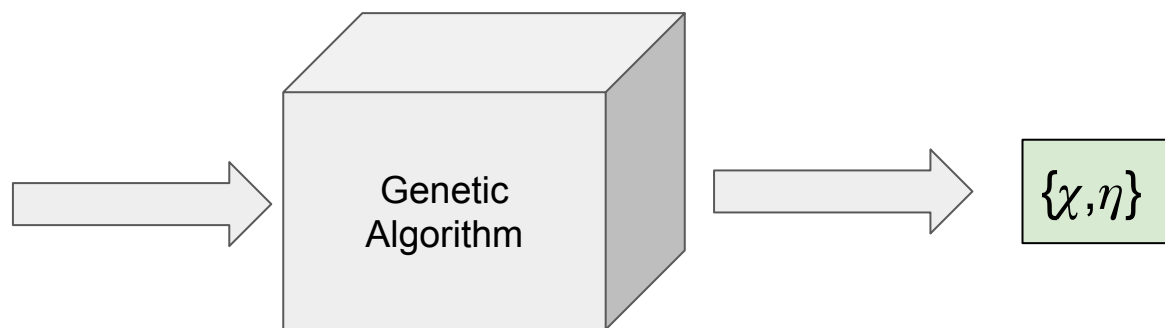
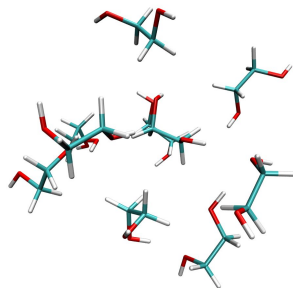
- 1) QM/EE interaction energies
- 2) QM static polarizability

The goal of this lab will be that of parameterizing the FQ parameters χ and η for a Deep Eutectic Solvent by using a Genetic Algorithm.

Our training set will be *QM/EE interaction energies* and *QM polarizabilities*



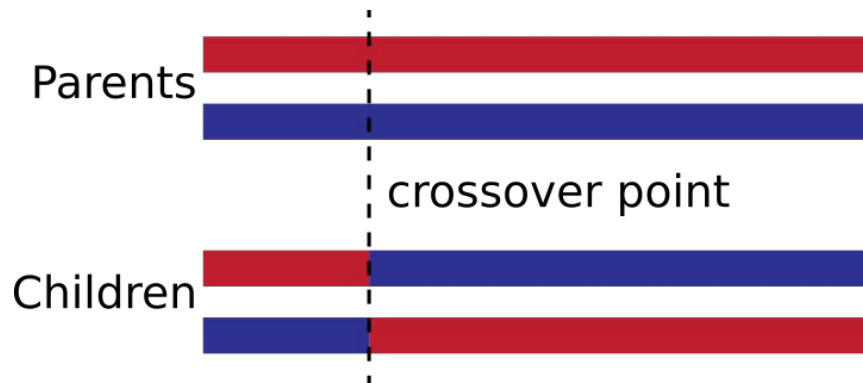
(c) Methanol



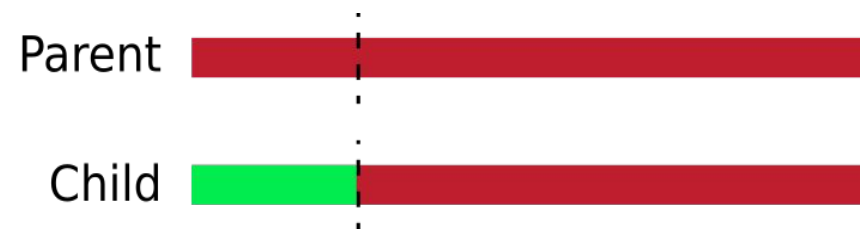
How does a Genetic Algorithm work

1. **Initialization:** Create an initial population of individuals (chromosomes).
2. **Evaluation:** Assign a fitness value to each individual based on how well it solves the problem.
3. **Selection:** Select individuals from the population for reproduction based on their fitness.
4. **Crossover:** Exchange genetic material (genes) between parents to produce new chromosomes.
5. **Mutation:** Introduce random changes to the offspring's chromosomes.
6. **Replacement:** Replace some individuals in the current population with the newly created offspring.
7. **Iteration:** Repeat the process until the algorithm converges or a termination condition is satisfied.

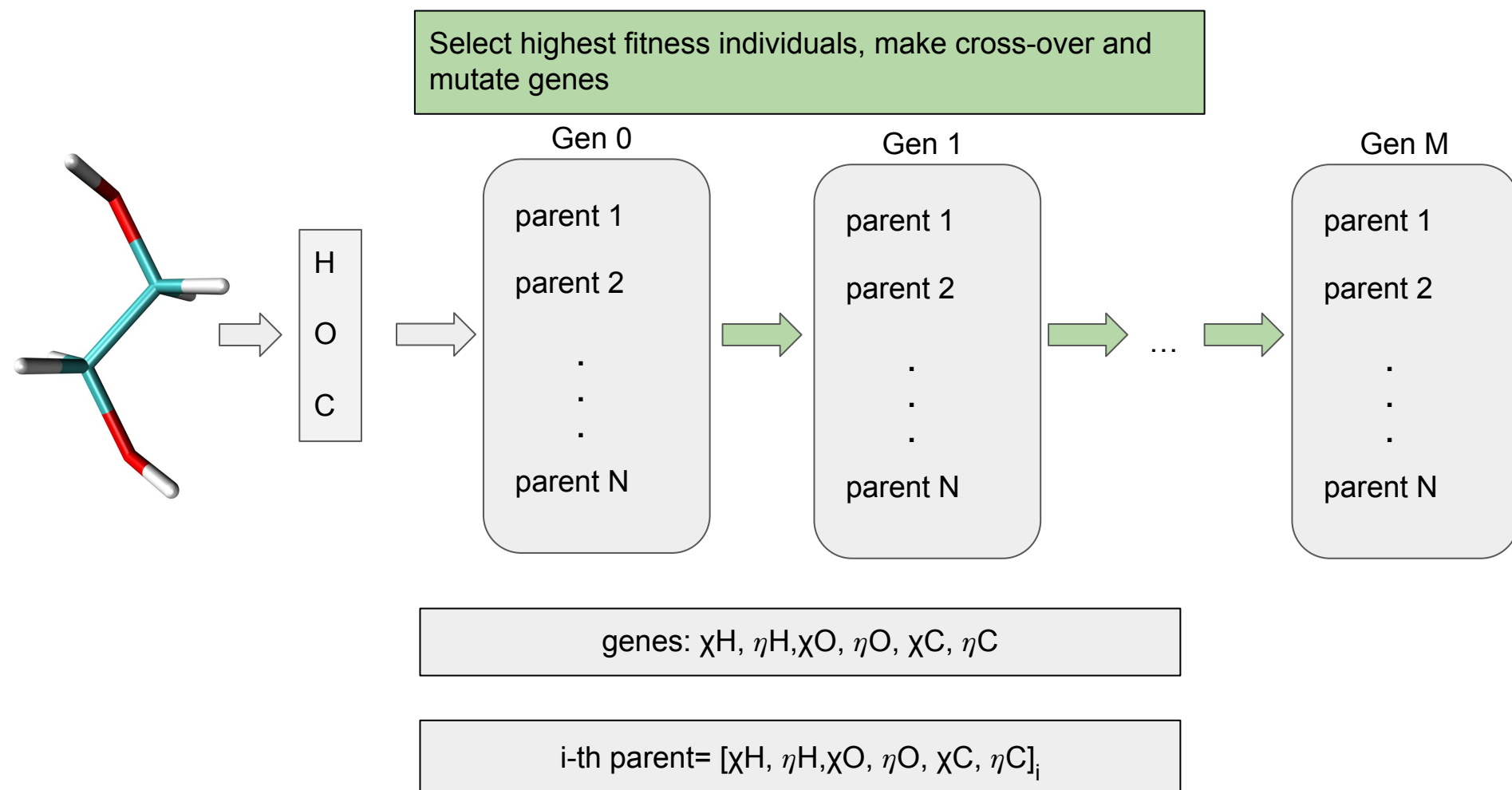
Crossover



Mutation



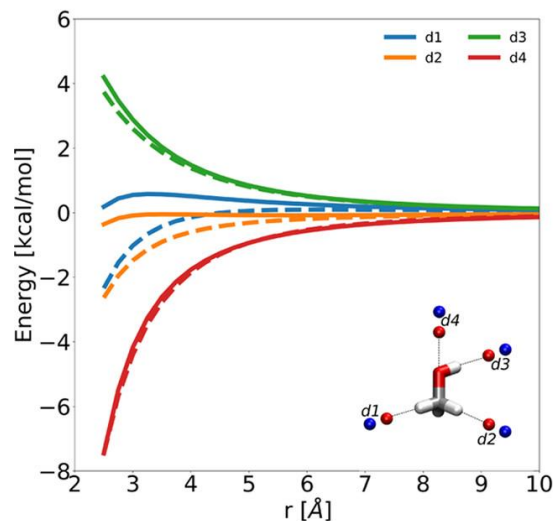
In our specific case..



The fitness of each individual depends on how well it reproduces:

- 1) QM/EE interaction energies;
- 2) QM polarizabilities

What you'll do



(c) Methanol

1) Position the CM of the fixed dipoles around the selected molecules (Cl/Choline and Ethylenglycol).

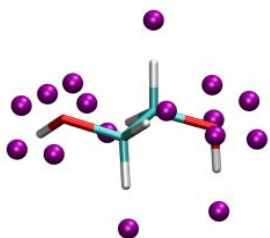
To this end you'll use the `make_dipoles.py` code.

2) Translate the CM of each dipole along the correct direction.

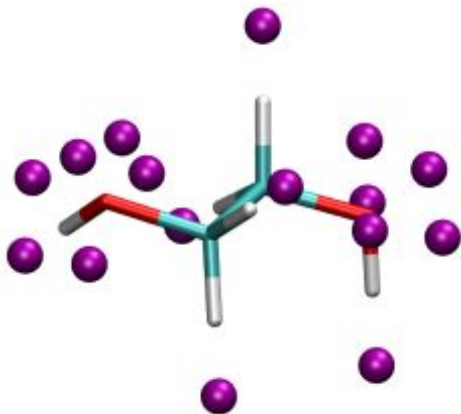
To this end you'll use the `translate_dipoles.py` code.

This will allow us to set up the training set for the QM/EE calculations.

In the next session we will compute these energies and thus we will have prepared our reference data.



How?



Download session_1 folder from the shared drive.

Upload it on tranter (scp -r session_1 your-user-name@tranter01.sns.it/)

Go to *dipoles_generation/*, here you have the *make_dipoles.py* and *translate_dipoles.py*.

Modify in these codes the following line:

```
sys.path.insert(1, 'PATH-TO-YOUR-CODE/des_param_code')
```

with your right path to *des_param_code*.

In my case it will be

```
sys.path.insert(1, '/home/lnicoli/session_1/des_param_code')
```

Now you are ready to work on the code. Indeed, both *make_dipoles.py* and *translate_dipoles.py* won't work if you don't fix *des_param_code*.

Task 1)

Work on the *get_connectivity* procedure of the *molecule_class*, which will allow you to define for each atom of the considered molecule, the number of atoms to which this atom is connected.

In particular, you have to initialize the values of *molecule.number_connections* (the list defining for each atom the number of atoms to which it is bonded), and *molecule.connected_to* (a list of dimensions *molecule.atoms*, where the i-th element is a list containing the indices of the atoms connected to the i-th atom).

This will allow you to get the connectivity of each atom.

Depending on the connectivity, we will generate the dipoles along different directions:

- 0 bonds: if the atom under consideration has 0 bonds, either we made a mistake or we are considering the Cl⁻ ion. In the latter case, you will create dipoles along the directions of the four lone pairs.
- 1 bond: the dipole center of mass will be generated along the direction defined by the single bond of the target atom.
- 2 bonds: generate dipoles along the directions defined by the bonds; then generate dipoles along the bisector of the angle formed by the two bonds; eventually generate dipoles along the direction orthogonal to the plane identified by these three atoms.

Task 2)

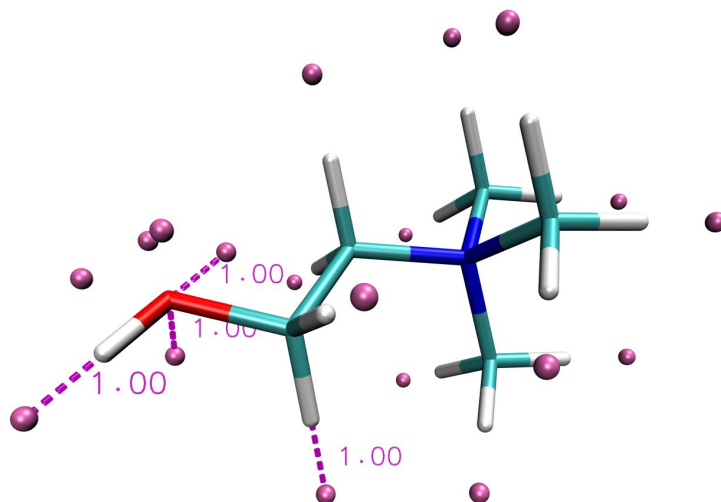
Work on the *position_the_dipoles_around* procedure of the *dipoles_class*, which will allow you to place the center of mass of the dipoles in the correct starting positions (before translating them).

In particular, you have to initialize:

positions (a list of the positions of the CM of the dipoles $[[x1,y1,z1],[x2,y2,z2],\dots]$),

versors (a list of the versor of each dipole $[[x1,y1,z1],[x2,y2,z2],\dots]$),

n_dipoles (the number of dipoles added).



Once you'll have done this you'll be able to run *make_dipoles.py*

You'll end up with the folder *seed_dipoles*.

Visualize the dipoles and check whether they are placed where you want them to be. Shall we do something else?

molecule class -> **.xyz files** (you can visualize them with Avogadro or VMD)

```

10
C      -0.568917  0.510994  -0.030643
H      -0.511051  1.129547  -0.935109
H      -0.475895  1.182285  0.843266
C       0.568886  -0.511002  -0.030857
H       0.476058  -1.182584  0.842841
H       0.510947  -1.129317  -0.935495
O       1.851348  0.115017  -0.061085
H       1.988491  0.593873  0.765428
O      -1.851447  -0.114764  -0.060967
H      -1.987568  -0.595781  0.764479

```

dipoles class -> **.xyz files** (you can visualize them with Avogadro or VMD)

.dip files (you cannot visualize them, because they have information also about directions and sign)

total number of dipoles: 4

Dipole	Pos X	Pos Y	Pos Z	Dir X	Dir Y	Dir Z	Sign
0	11.69565	14.34065	29.22765	-0.57735	-0.57735	-0.57735	+-
1	12.85035	15.49535	29.22765	0.57735	0.57735	-0.57735	+-
2	12.85035	14.34065	30.38235	0.57735	-0.57735	0.57735	+-
3	11.69565	15.49535	30.38235	-0.57735	0.57735	0.57735	+-

Tasks for the next session

- 1) Write the system of equations of a FQ system in presence of fixed external charges (FQ/EE)
- 2) Write an analytical expression for the static polarizability for the FQ forcefield.