ENSOR

Ensor is a python program with advanced partitioning algorithms for optimizing Quantum calculation for large molecules.

> If one can divide a species of interest into fragments, employ some level of ab-initio QM to calculate the wave function, energy, and properties of each fragment, and then combine the results from the fragment calculations to predict the same properties for the whole, the possibility exists that the accuracy of the outcome can approach that which would be obtained from a full non fragmented calculation.

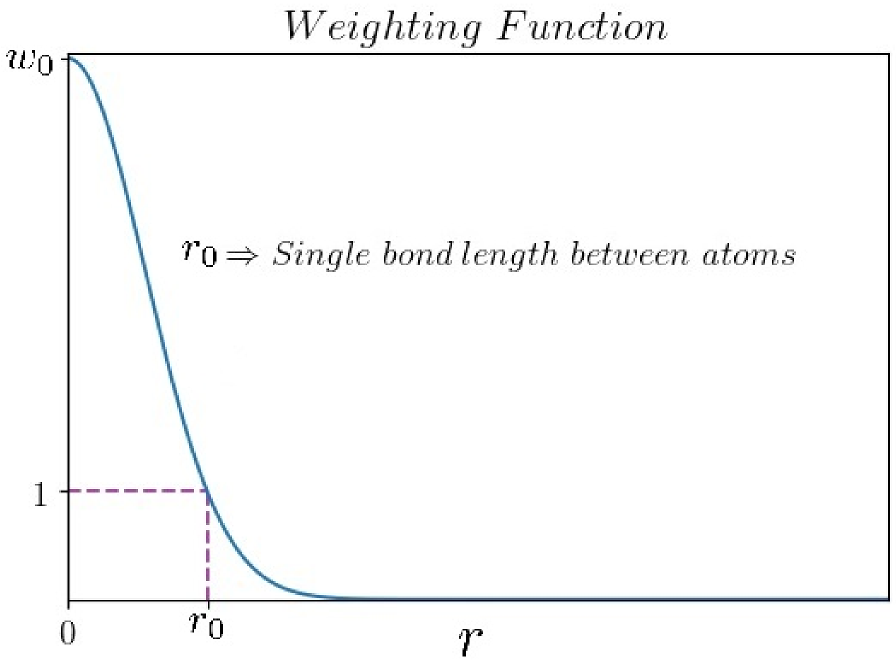
Table of Contents

Partitioning Algorithm

Building Connectivity of Molecule

We build Connectivity by assigning weight to the Matrix $C_{n,n}$ representing connectedness between atoms as.

- $w_{ij} = w_0^{\left(1-\frac{r_{ij}^2}{r_0^2}\right)}$ where r_0 is single-bond length between atoms.



So, weighted connectivity C is
$C_{i,j} = w_{ij}$

Spectral Graph Partitioning

We define Laplacian L as

- $L = D - C$ where $D_{n,n} = diag(d_1, d_2, .., d_n)$ with $d_i = \sum_{j=1}^{n} C_{i,j} = \sum_{j=1}^{n} w_{ij}$

> The Laplacian of any graph G have a balanced division knowledge, but the G must be a one connected component. Otherwise a balanced division does not occur.

After normalization, we get,

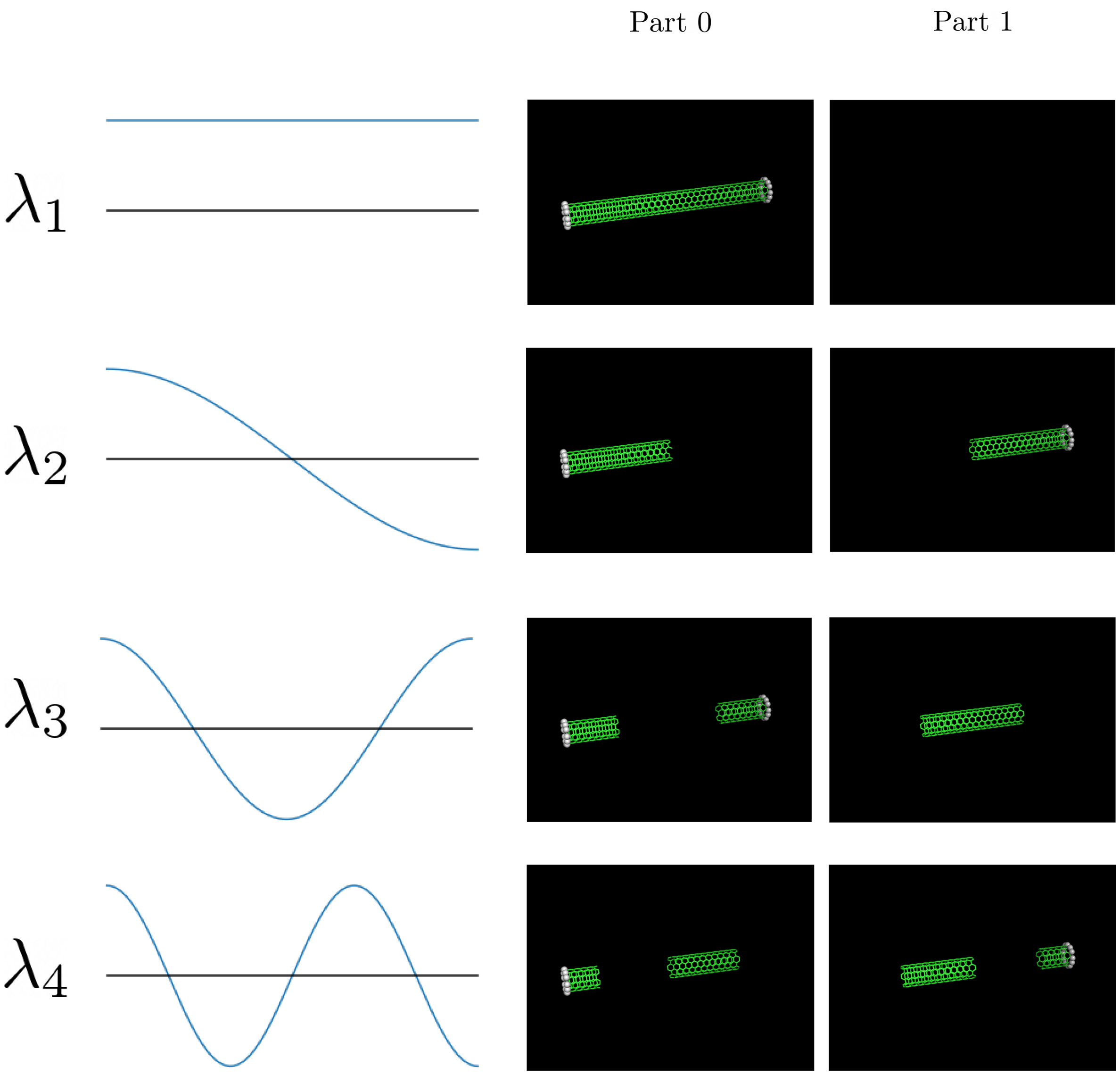
$$L_{i,j} = \begin{cases} 1, & \text{if i=j} \\ -\dfrac{w_{ij}}{\sqrt{d_i d_j}}, & \text{if i}\neq\text{j} \end{cases}$$

Because L is a symmetric matrix, its eigenvalues are real, and its eigenvectors are orthogonal to each other. Calculating eigenvalues and eigenvector using $Lv = \lambda v$, the eigenvector corresponding to the second smallest eigenvalue(λ_2) of L(G) is known as the Fiedler vector(f_2).

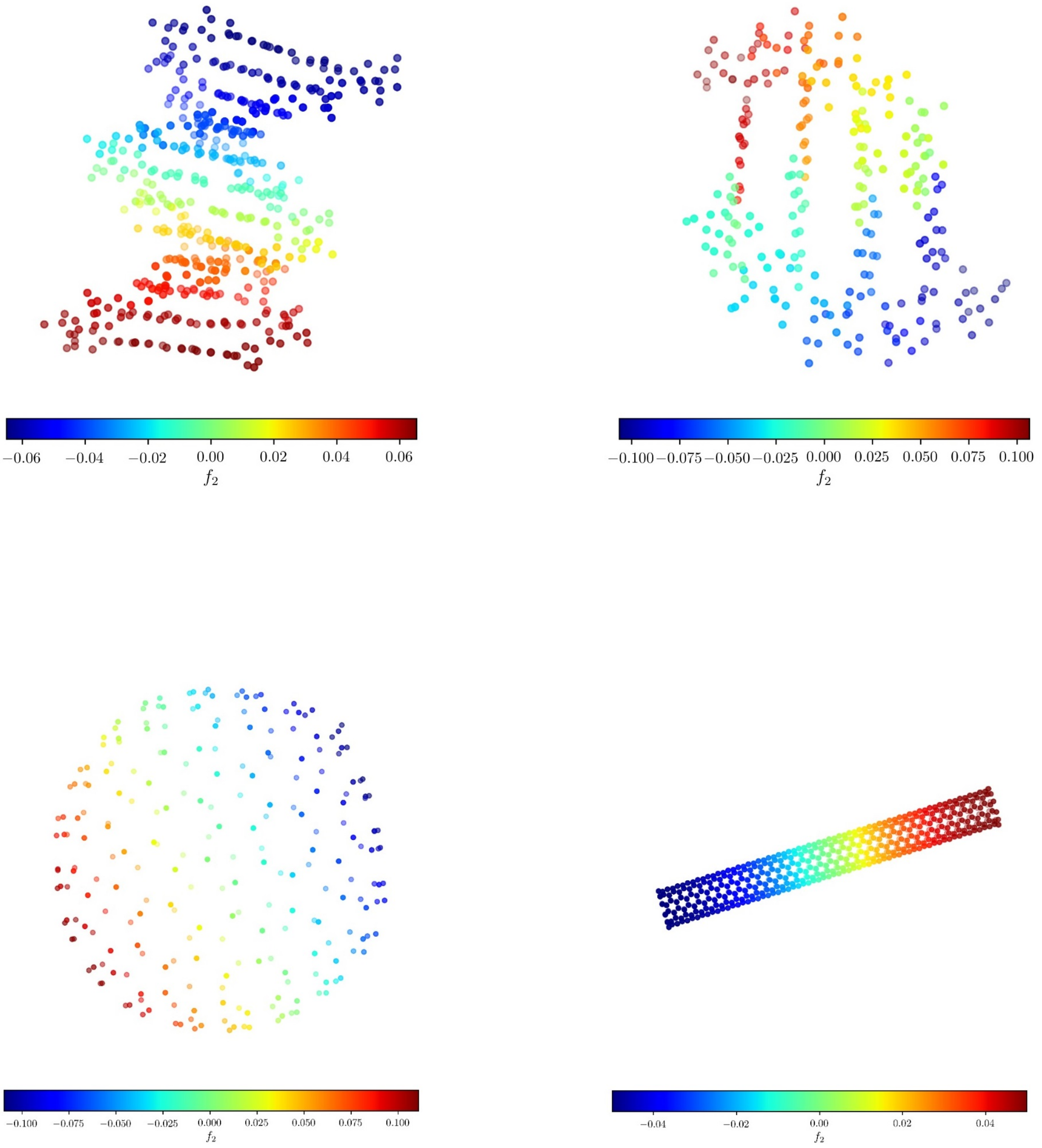
we can partition the molecule in two parts using these eigenvectors.

> if $f_2^n > median(f_2)$ than node n is in $part_A$ else in $part_B$

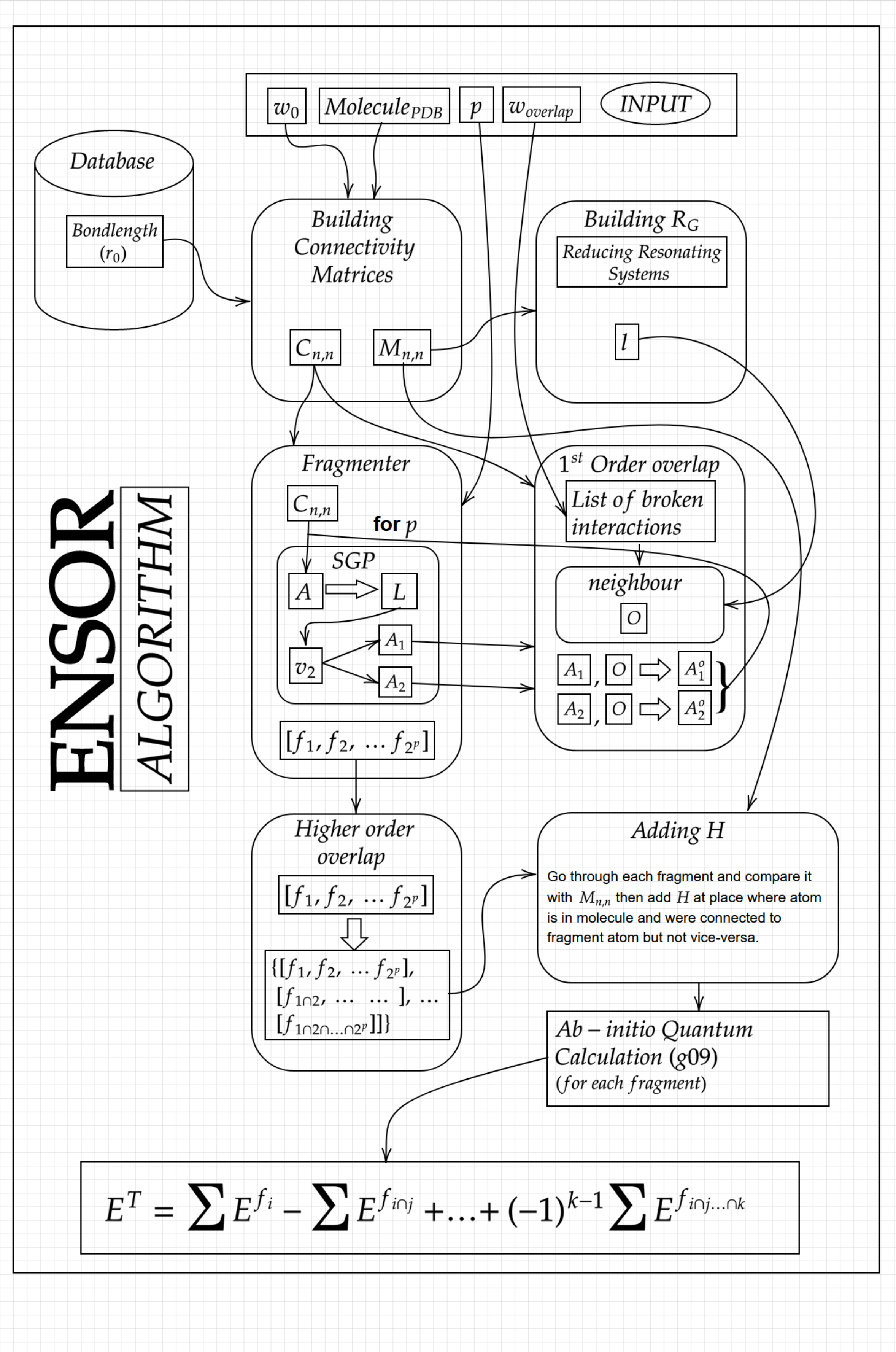
Partition of 804 atom nanotube using eigenvector corresponding to first, second, third, fourth eigen values.



Partition for different molecules(each atom's color represent value of that atom corresponding to second eigen vector)



Algorithm



Go through each fragment and compare it with $M_{n,n}$ then add H at place where atom is in molecule and were connected to fragment atom but not vice-versa.

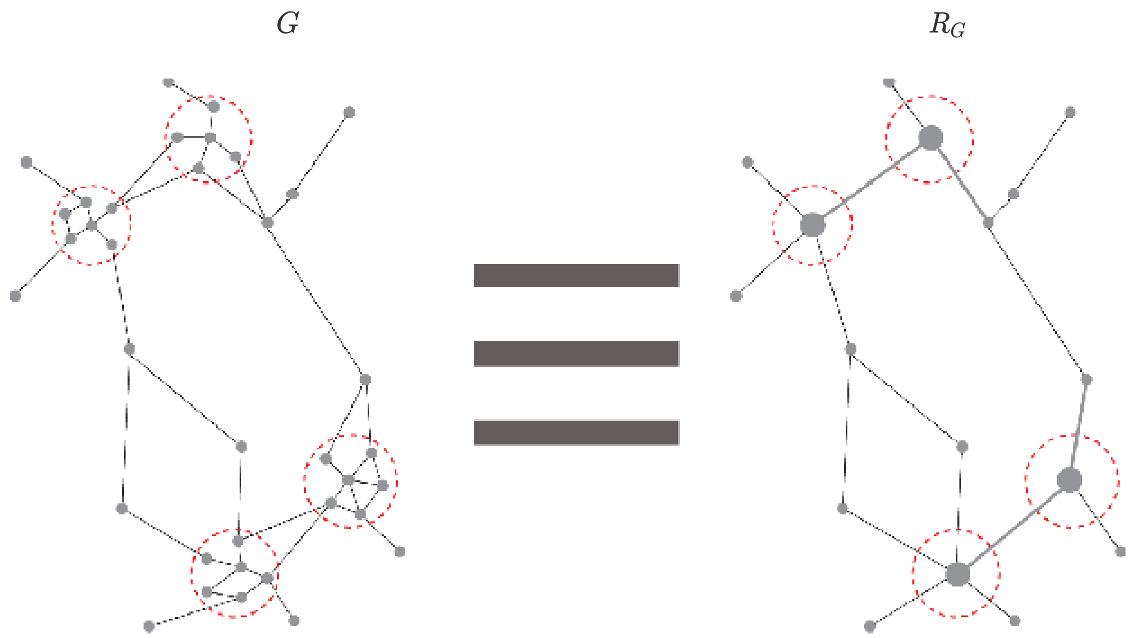
$$E^T = \sum E^{f_i} - \sum E^{f_{i\cap j}} + \ldots + (-1)^{k-1} \sum E^{f_{i\cap j \ldots \cap k}}$$

Overlap Selection

After splitting the molecule in two parts, we calculate overlap extend using those atom with some cut-off weightage, which were connected in main molecule but not in fragments.

we define $w_{overlap}$ as the minimum cut-off weight between atom ,to start adding those atom in overlap.

We have main molecule as G but for selecting neighbor and overlap, we use reduced R_G graph to crawl, as we have to cover all of that resonating system even if we count one atom of that system in overlap.



R_G is G but with {aromatic rings, resonating system, double and triple bonds} Grouped as one node.

we also defined $Neighbor(atom, c, R_G)$ where c is the cut-off weight over which we assume atoms are connected and are neighbor.

which returns all neighbor of the atom in R_G. So, now we can go multiple level of neighbors to extend the overlap.

Adding Hydrogens

After getting all fragments, we go through each one and compare them with main Molecule then add H atoms at that place where atom is in molecule and were connected to fragment atom but not vice-versa.

Under Construction...