Calculation

+calculator: ase.calculators.interface.Calculator
+selector: csmmcalc.mixer.selector.AtomSelector

+__init__(name:str,selector:AtomSelector)
+calculation_required(atoms:ase.Atoms,quantities)
+get_subset(atoms:ase.Atoms)

EnergyCalculation

+__init__(name:str,selector:AtomSelector)
+get_energy(atoms:ase.Atoms)

ForceCalculation

+__init__(name:str,selector:AtomSelector,
          debug:int=0,debug_file:File=None)
+get_forces(atoms:Atoms)

AtomSelector

+*select_atoms(atoms:ase.Atoms): tuple*

AtomListSelector

+__init__(atom_ids:list,weights:list)
+set_atom_ids(atom_ids:list)
+set_weights(weights:list)
+select_atoms(atoms:ase.Atoms)

CalcRegion

+__init__(name:str,pos:tuple(float, float,
          float),cutoff:float,pbc:tuple,
          debug:int=0)
+*atom_inside(pos:tuple(float, float, float))*
+*get_weight(pos:tuple(float, float, float))*
+*get_bounding_box(): tuple(pos, dim)*
+length2(a:tuple(float, float, float),b:tuple(float,
          float, float))
+select_atoms(atoms:ase.Atoms)

OctreeNode

+__init__(pos:tuple(float, float, float),
          dim:tuple(float, float, float),
          res:float)
+add_object(obj,pos:tuple(float, float, float))
+get_subnode_addr(pos:tuple(float, float,
          float)): (int, int, int)
+get_objects(): set()
+find_objects(pos:tuple(float, float, float),
          reach:float): set()

CalcBox

+__init__(name:str,pos:tuple(float, float,
          float),dim:tuple(float, float,
          float),inner_dim:tuple(float, float,
          float)=None,cutoff:float,pbc:tuple=None,
          debug:int=0)
+get_bounding_box()
+get_weight(pos)
+atom_inside(pos)

<<ase.calculators.interface.Calculator>>
Mixer

+__init__(name:str,forces:[ForceCalculation],
          energies:[EnergyCalculation],debug:int)
+calculation_required(atoms,quantities)
+get_forces(atoms:ase.Atoms)
+get_potential_energy(atoms:ase.Atoms,force_consistent:boolean=False)
+set_atoms(atoms:ase.Atoms)
+set_atom_ids(atoms:ase.Atoms)
+get_atom_ids(atoms:ase.Atoms): np.array