```
 1: #include <iostream>
 2: #include "../io.h"
 3: #include "../analysis.h"
 4: #include "../container.h"
 5: #include "../potentials.h"
 6: #include "../countdown.h"
 7: typedef pot_coulomb T_pairpot;
 8: #include "../markovmove.C"
 9:
10: using namespace std;
11:
12: int main() {
13:   cell cell(100.);                      // We want a spherical cell
14:   iopov povray(cell);                   // We want a POVRAY snapshot
15:   canonical nvt;                        // Use the canonical ensemble
16:   pot_setup cfg;                        // Setup pair potential (default values)
17:   interaction<T_pairpot> pot(cfg);      // Functions for interactions
18:   countdown<int> clock(10);             // Estimate simulation time
19:   macromolecule protein;                // Group for the protein
20:   ioaam aam(cell);                      // Protein input file format is AAM
21:   protein.add( cell, aam.load(
22:         "examples/calbindin.aam" ));    // Load protein from disk
23:   protein.move(cell, -protein.cm);      // ..and translate it to origo (0,0,0)
24:   protein.accept(cell);                 // ..accept translation
25:
26:   group salt;                           // Group for mobile ions
27:   salt.add( cell, particle::NA, 11+19); // Insert sodium ions
28:   salt.add( cell, particle::CL, 11 );   // Insert chloride ions
29:   saltmove sm(nvt, cell, pot);          // Class for salt movements
30:   chargereg tit(nvt,cell,pot,salt,7);   // Prepare titration. pH 7
31:   systemenergy sys(pot.energy(cell.p)); // System energy analysis
32:
33:   cout << cell.info() << tit.info();    // Some information
34:
35:   for (int macro=1; macro<=10; macro++) {       // Markov chain
36:     for (int micro=1; micro<=2e3; micro++) {
37:       sm.move(salt);                            // Displace salt particles
38:       sys+=sm.du;
39:       if (tit.titrateall()==true) {             // Titrate groups
40:         protein.charge(cell.p);                 // Re-calc. protein charge
41:         protein.dipole(cell.p);                 // Re-calc. dipole moment
42:         sys+=tit.du;
43:       }
44:       sys+=sm.du;                               // Keep system energy updated
45:     }
46:     cout << "Macro step " << macro << " completed. ETA: " << clock.eta(macro);
47:     sys.update(pot.energy(cell.p));             // Update system energy averages
48:     cell.check_vector();
49:   }
50:   cout << sys.info() << sm.info() << tit.info() // More information...
51:     << salt.info() << protein.info();
52:   povray.save("protein-example.pov", cell.p);   // Save POVRAY file
53: }
```