```cpp
 1: #include <iostream>
 2: #include "analysis.h"
 3: #include "container.h"
 4: #include "potentials.h"
 5: #include "countdown.h"
 6: typedef pot_coulomb T_pairpot;        // Specify pair potential
 7: #include "markovmove.h"
 8:
 9: using namespace std;
10:
11: int main() {
12:   cell::cell con(100.);                // Use a spherical container
13:   canonical nvt;                       // Use the canonical ensemble
14:   pot_setup cfg;                       // Setup pair potential (default)
15:   interaction<T_pairpot> pot(cfg);     // Functions for interactions
16:   countdown<int> clock(10);            // Estimate simulation time
17:   macromolecule protein;               // Group for the protein
18:   ioaam aam(con);                      // Protein input file format is AAM
19:   protein.add( con, aam.load(
20:       "calbindin.aam" ) );             // Load protein from disk
21:   protein.move(con, -protein.cm);      // ..translate it to origo (0,0,0)
22:   protein.accept(con);                 // ..accept translation
23:   group salt;                          // Group for salt and counter ions
24:   salt.add( con, particle::NA, 34+19); //   Insert sodium ions
25:   salt.add( con, particle::CL, 34 );   //   Insert chloride ions
26:   saltmove sm(nvt, con, pot);          // Class for salt movements
27:   aam.load(con, "confout.aam");        // Load old config. (if present)
28:   chargereg tit(nvt,con,pot,salt,7.6); // Prepare titration. pH 7.6
29:   systemenergy sys(pot.energy(con.p)); // System energy analysis
30:   cout << con.info() << tit.info();    // Some information
31:
32:   for (int macro=1; macro<=10; macro++) {   // Markov chain
33:     for (int micro=1; micro<=1e3; micro++) {
34:       switch (rand() % 2) {                  // Randomly chose move
35:         case 0:
36:           sys+=sm.move(salt);                // Displace salt particles
37:           break;
38:         case 1:
39:           sys+=tit.titrateall();             // Titrate protein sites
40:           protein.charge(con.p);             // Re-calc. protein charge
41:           protein.dipole(con.p);             // Re-calc. dipole moment
42:           break;
43:       }
44:     }                                        // END of micro loop
45:     sys.update(pot.energy(con.p));           // Update system energy
46:     aam.save("confout.aam", con.p);          // Save config. to disk
47:     cout << "Macro step " << macro
48:          << " completed. ETA: " << clock.eta(macro);
49:   }                                          // END of macro loop
50:   cout << sys.info() << sm.info()            // Print results
51:        << tit.info() << salt.info() << protein.info();
52: }
```

53: