Protein Structure is one of the bases of the biomedicine and nanotechnology bases. Different methods are used to determine the structures.

The particle-particle distance problem (pp-distance) is used in several applications from Astrophysics to Molecular Dynamics. We solve the pp-distance problem to calculate the theoretical X-Ray spectra (see right #7) and validate it with the real one (see right #3). Our C program reduce the consuming time using HPC over FPGA and GPU devices (see below).

We have to compute all the distances between a set of $N$ particles. This is a $O(N^2)$ complexity problem, where $N$ is the particles number. FPGA OpenCL Kernel is called by OmpSs.

The benchmarks have been done computing a model of 2 million particles.

The dynamic unroll, the kernel code:

```c
int my_threshold = ((N-j)/UF) * UF; // particles for the unrolls
for (l=j; l<j+my_threshold; l+=UF) {
    #pragma ivdep
    #pragma unroll UF
    for (m=0; m < UF; m++) {
        int k = l + m;
        int distances = (sqrt((x-h_B1[k]+factor_coor)*(x-h_B1[k]+factor_coor) + (y-h_B2[k]+factor_coor)*(y-h_B2[k]+factor_coor) + (z-h_B3[k]+factor_coor)*(z-h_B3[k]+factor_coor)));
        float densitat = p * (h_B4[k]-factor_pes) / 1000.0;
        j_B5[distances][m]+= densitat; // sum em els pesos de les densitats electroniques a la posicio de la seva distancia en el seu vector de paral.lelitzacio
    }
}
// REMAINDER CASES for each "j"
for (li=my_threshold; li<N; li++) {
    int k = li + m;
    int distancesR = (sqrt((x-h_B1[k]+factor_coor)*(x-h_B1[k]+factor_coor) + (y-h_B2[k]+factor_coor)*(y-h_B2[k]+factor_coor) + (z-h_B3[k]+factor_coor)*(z-h_B3[k]+factor_coor)));
    float densitatR = p * (h_B4[k]-factor_pes) / 1000.0;
    j_B5[distancesR][0]+= densitatR; // IDEM a l vector "0" els remainder cases
}
```