

```

                                partition_array_3D_bigly
function [N_0, s_0] = partition_array_3D_bigly(input_array)
vec_size = size(input_array);
num_cols = vec_size(2);
alpha = .5;
N_0 = ceil(log2((num_cols)/2));
N_1 = ceil((1 - alpha)*N_0);
s_0 = test_entropy_array_3D(input_array, N_0);
s_1 = test_entropy_array_3D(input_array, N_1);
if((1/N_1^2)*log2(N_1^2)*s_1 > (1/N_0^2)*log2(N_0^2)*s_0)
    while((1/N_1^2)*log2(N_1^2)*s_1 > (1/N_0^2)*log2(N_0^2)*s_0 && (ceil((1 -
alpha)*N_0) > 1))
        N_0 = N_1;
        N_1 = ceil((1 - alpha)*N_0);
        s_0 = test_entropy_array_3D(input_array, N_0);
        s_1 = test_entropy_array_3D(input_array, N_1);
    endwhile
else
    N_1 = ceil((1 + alpha)*N_0);
    s_0 = test_entropy_array_3D(input_array, N_0);
    s_1 = test_entropy_array_3D(input_array, N_1);
    while((1/N_1^2)*log2(N_1^2)*s_1 > (1/N_0^2)*log2(N_0^2)*s_0 && ceil((1 +
alpha)*N_0) <= num_cols)
        N_0 = N_1;
        N_1 = ceil((1 + alpha)*N_0);
        s_0 = test_entropy_array_3D(input_array, N_0);

```

```
                                partition_array_3D_bigly
s_1 = test_entropy_array_3D(input_array, N_1);
endwhile
endif
endfunction
```