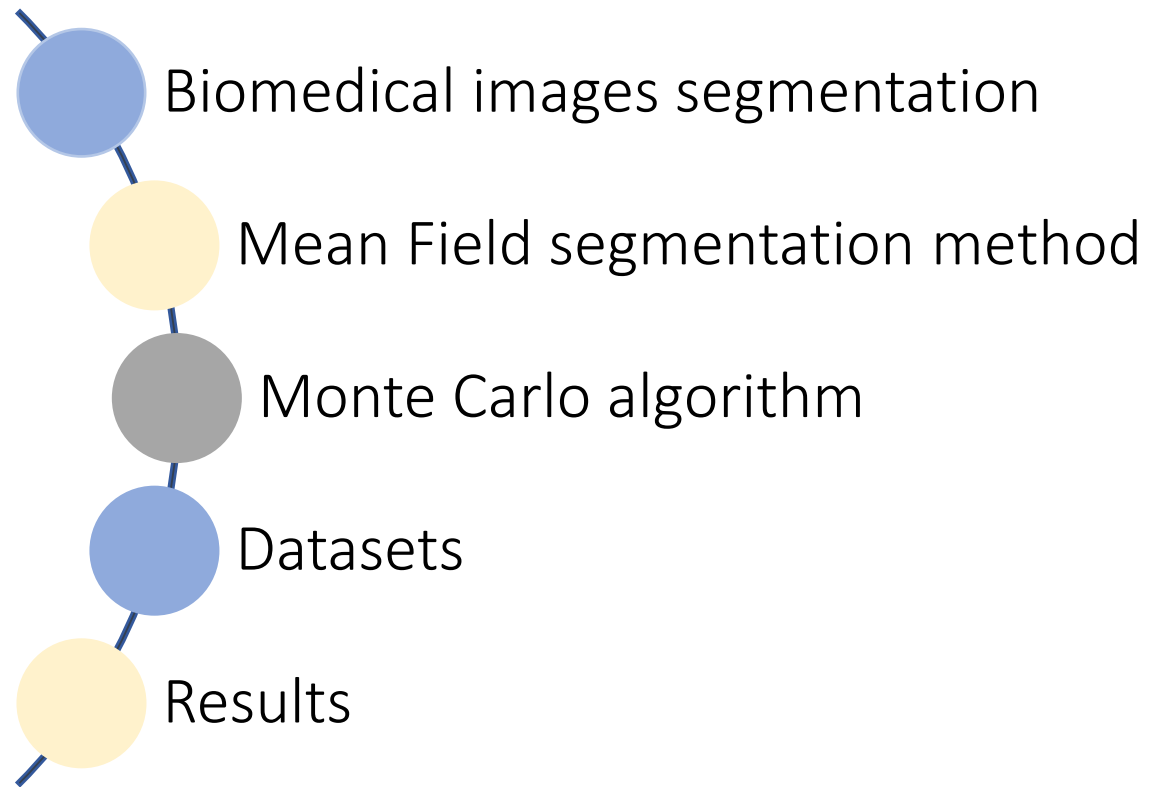# Application of mean field interaction algorithms to biomedical image segmentation

R.F. Cabini, A. Pichiecchio, A. Lascialfari, S. Figini and M. Zanella

Outline

Biomedical images segmentation

Mean Field segmentation method

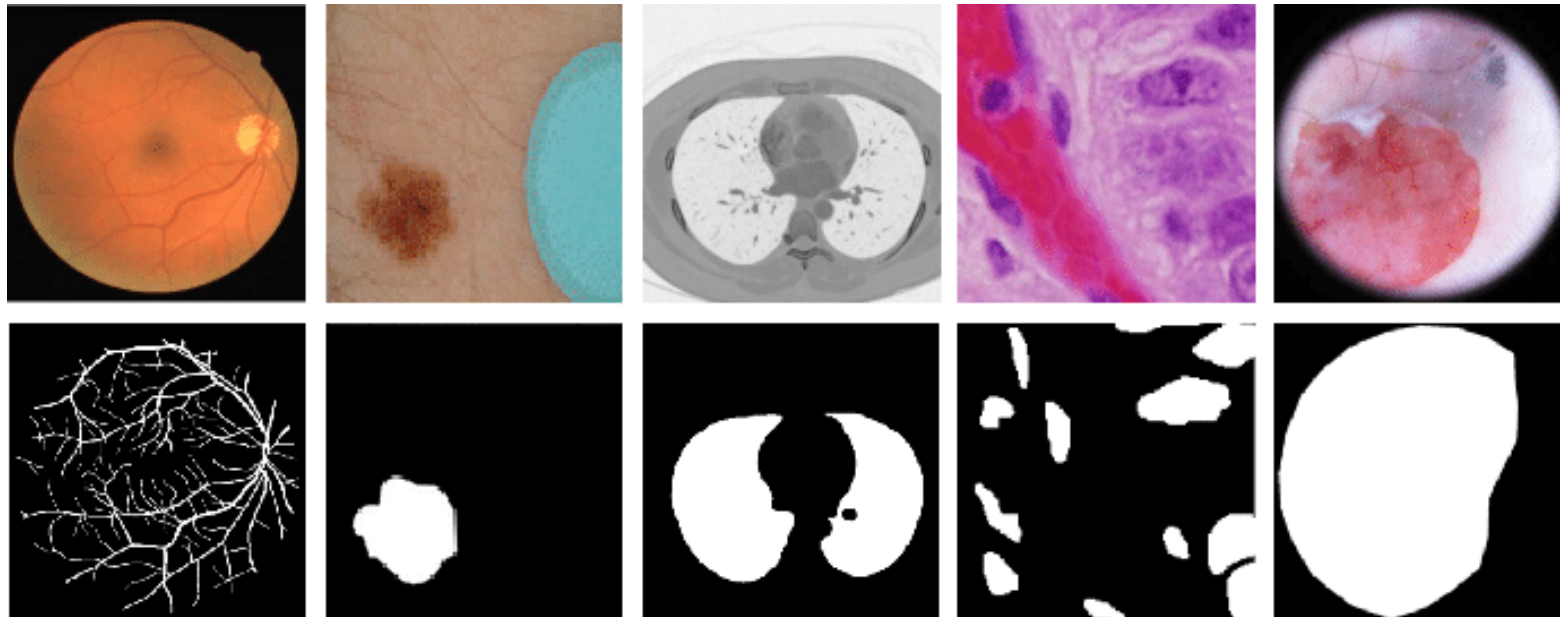Monte Carlo algorithm

Datasets

Results

Medical Image Segmentation

Segmentation is the process of partitioning an image into different meaningful segments.
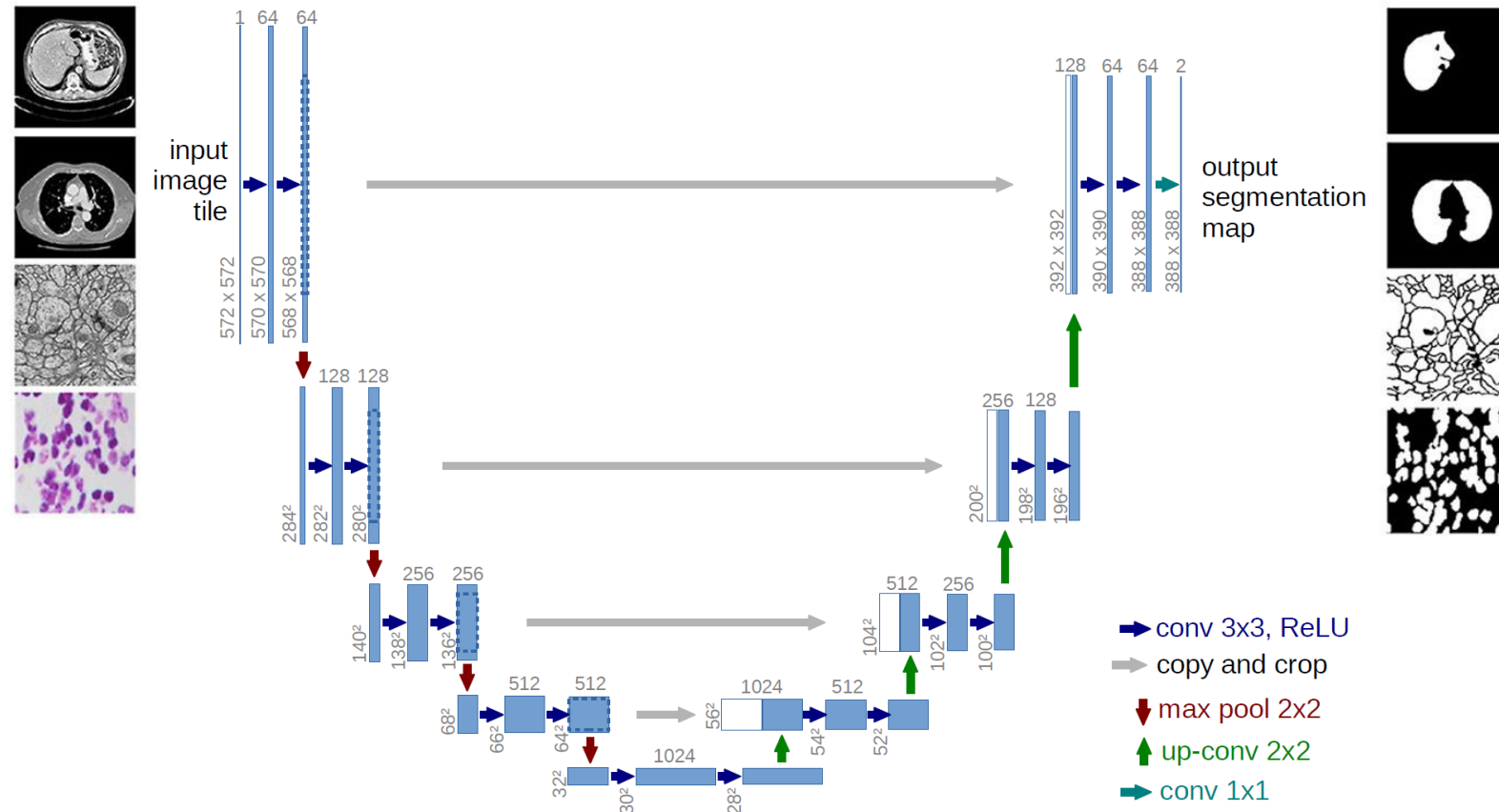
In medical imaging, these segments correspond to:

- **tissue classes**,
- **pathologies**,
- **organs**,
- or other biologically relevant structures.

Traditional Automatic Segmentation methods

Traditional biomedical image segmentation techniques include:

- Neural Network-based algorithms

- Clustering methods (e.g., k-means, c-means and hierarchical clustering)

Traditional Automatic Segmentation methods

Traditional biomedical image segmentation techniques include:

- Neural Network-based algorithms

- Clustering methods (e.g., k-means, c-means and hierarchical clustering):
  - grouping data points with similar feature vectors;
  - for image segmentation, the feature space is represented by **gray levels of pixels.**

Original Image

Segmented Image when K = 6

Objectives

- Define a new clustering method that identifies clusters based on the Euclidean distance and the distance of the gray intensity between pixels;

- evaluate which formulation of the model guarantees the best predictive and time performances;

- integrate the developed technique into the biomedical imaging segmentation protocol and evaluate its feasibility.
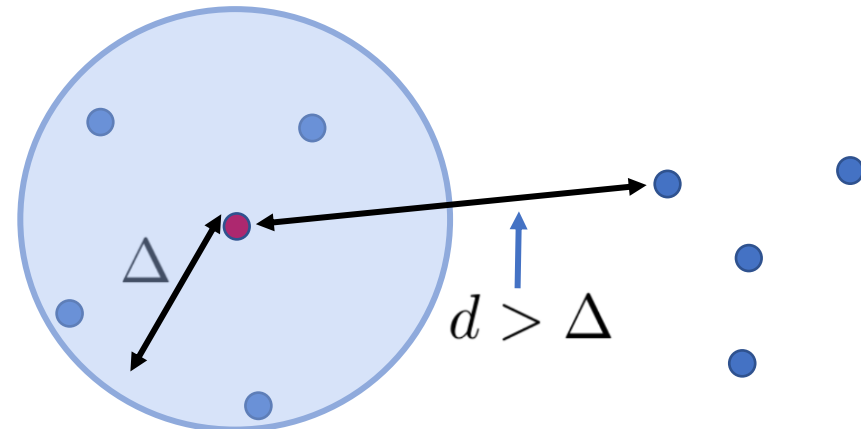
Consider a population of N agents (or particles) with an initial opinion $x_i \in \mathbb{R}^2$ and a **bounded confidence interval** defined by $\Delta \geq 0$.

The dynamic of the i-th particle is given by:

$$\frac{d}{dt}\mathbf{x}_i(t) = \frac{1}{N} \sum_{j=1}^{N} \chi(||\mathbf{x}_j(t) - \mathbf{x}_i(t)|| \leq \Delta)(\mathbf{x}_j(t) - \mathbf{x}_i(t)) \qquad i = 1, \ldots, N$$

In a finite time, the HK model converges to a steady configuration where the initial states are grouped in a finite number of clusters.

Δ

$d > \Delta$

Hegselmann, Rainer, and Ulrich Krause. "Opinion dynamics and bounded confidence models, analysis, and simulation." *Journal of artificial societies and social simulation* 5.3 (2002).

To each particle $i=1,\ldots,N$ associate:

- a time-dependent state vector $\mathbf{x}_i = \mathbf{x}_i(t) \in \mathbb{R}^2 \rightarrow$ **position** of the pixel

- features $c_i \in \mathbb{R}$ representing static characteristics of the system $\rightarrow$ **gray level** of the pixel

The opinion dynamics equation becomes:

$$c_i(t) = c_i(0) \quad i = 1, \ldots, N$$

$$\frac{d}{dt}\mathbf{x}_i(t) = \frac{1}{N}\sum_{j=1}^{N} P_{\Delta_1,\Delta_2}(\mathbf{x}_i(t), \mathbf{x}_j(t), c_i, c_j)(\mathbf{x}_j(t) - \mathbf{x}_i(t)) \quad i = 1, ..., N$$

Where P is the interaction function:

$$P_{\Delta_1,\Delta_2}(\mathbf{x}_i(t), \mathbf{x}_j(t), c_i, c_j) = \chi(\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\| \leq \Delta_1)\chi(|c_j(t) - c_i(t)| \leq \Delta_2)$$

Herty, Michael, Lorenzo Pareschi, and Giuseppe Visconti. "Mean field models for large data-clustering problems." *arXiv preprint arXiv:1907.03585* (2019).

To quantify the segmentation error, we introduce a non-constant diffusion coefficient D(·) which depends on the value of the features c_i.

The opinion dynamics equation becomes:

$$c_i(t) = c_i(0) \quad i = 1, \ldots, N$$

$$\frac{d}{dt}\mathbf{x}_i(t) = \frac{1}{N}\sum_{j=1}^{N} P_{\Delta_1,\Delta_2}(\mathbf{x}_i(t), \mathbf{x}_j(t), c_i, c_j)(\mathbf{x}_j(t) - \mathbf{x}_i(t)) + \sqrt{D(c_i(t))}d\mathbf{x}_i(t) \quad i = 1, \ldots, N$$

In this formulation the equation can be rewritten as a Boltzmann equation.

Algorithm 1 Monte Carlo algorithm for Boltzmann equation

1: Given N particles $(\mathbf{x}_n^0, c_n^0)$, with $n = 1, \dots, N$ computed from the initial distribution $f_0(\mathbf{x}, c)$;
2: **for** $t = 1$ **to** T **do**
3:   set $n_p = \lfloor N/2 \rfloor$;
4:   sample n_p pairs (i, j) uniformly without repetition among all possible pairs of particles at time step t;
5:   for each pair (i, j), sample η_i and η_j from the uniform distribution $\mathcal{U}([-\sqrt{3\epsilon\sigma^2/2}, \sqrt{3\epsilon\sigma^2/2}])$;
6:   for each pair (i, j), compute the data change:

$$\Delta\mathbf{x}_i^t = \epsilon P_{\Delta_1, \Delta_2}(\mathbf{x}_i^t, \mathbf{x}_j^t, c_i^0, c_j^0)(\mathbf{x}_j^t - \mathbf{x}_i^t) + \sqrt{D(c_i^0)}\eta_i$$

$$\Delta\mathbf{x}_j^t = \epsilon P_{\Delta_1, \Delta_2}(\mathbf{x}_j^t, \mathbf{x}_i^t, c_j^0, c_i^0)(\mathbf{x}_i^t - \mathbf{x}_j^t) + \sqrt{D(c_j^0)}\eta_j$$

7:   **if** $\mathbf{x}_{i,j}^t + \Delta\mathbf{x}_{i,j}^t \in \Omega_1$ **then**

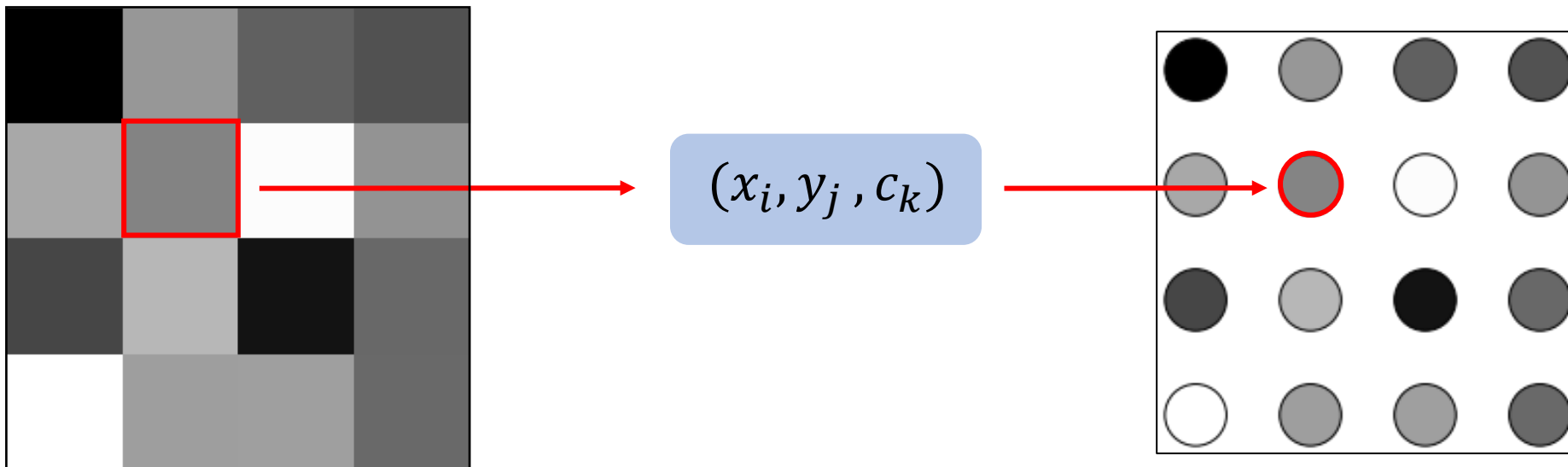$$\mathbf{x}_{i,j}^{t+1} = \mathbf{x}_{i,j}^t + \Delta\mathbf{x}_{i,j}^t$$

8:   **else**

$$\mathbf{x}_{i,j}^{t+1} = \mathbf{x}_{i,j}^t$$

9:   **end if**
10: **end for**

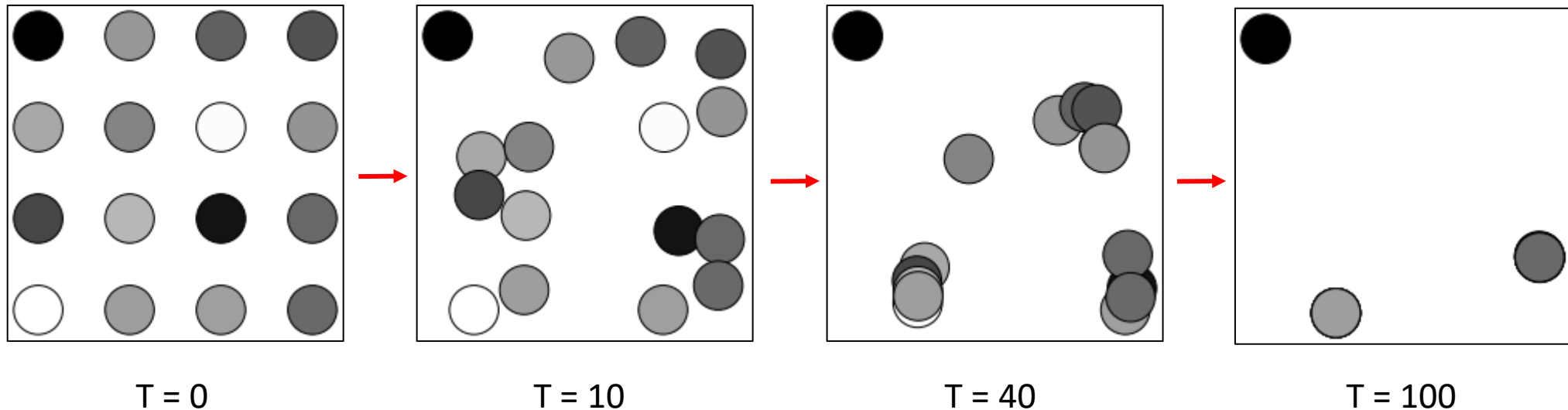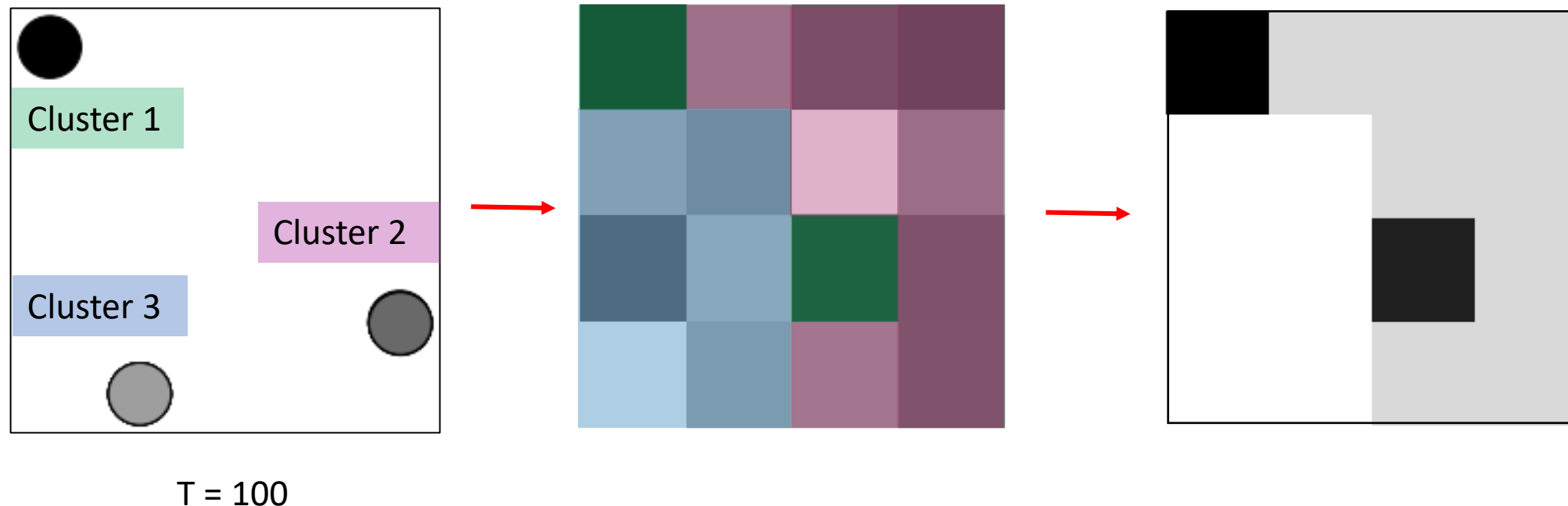Procedure to obtain the segmentation mask:

1. The pixels of the 2D image must be interpreted as evenly spaced particles, each characterized by the spatial position (x_i, y_j) and the static feature c_k.

(x_i, y_j, c_k)

Monte Carlo Algorithm for Boltzmann equation

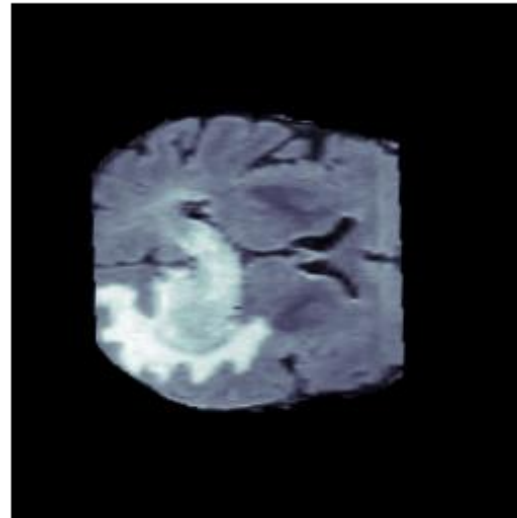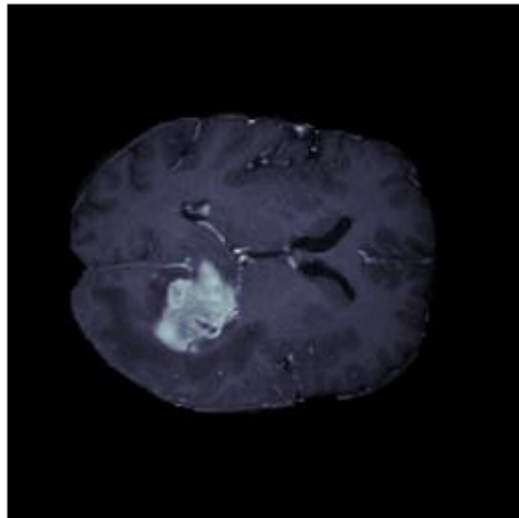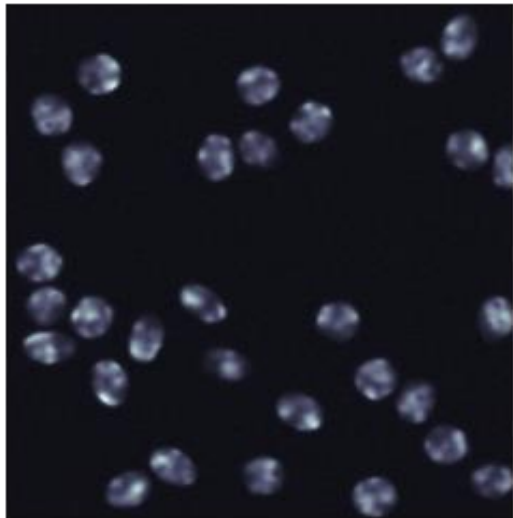Procedure to obtain the segmentation mask:

1. The pixels of the 2D image must be interpreted as evenly spaced particles, each characterized by the spatial position (x_i, y_j) and the static feature c_k.

2. MC Algorithm is applied to simulate the time evolution of the particles' position.

| T = 0 | T = 10 | T = 40 | T = 100 |

Monte Carlo Algorithm for Boltzmann equation

Procedure to obtain the segmentation mask:

1. The pixels of the 2D image must be interpreted as evenly spaced particles, each characterized by the spatial position (x_i, y_j) and the static feature c_k.

2. MC Algorithm is applied to simulate the time evolution of the particles' position.

3. The mean values of the pixels grouped in the same cluster are computed and these values are assigned to the initial position of the pixels in the original image.

Cluster 1

Cluster 2

Cluster 3

T = 100

Procedure to obtain the segmentation mask:

1. The pixels of the 2D image must be interpreted as evenly spaced particles, each characterized by the spatial position (x_i, y_j) and the static feature c_k .

2. MC Algorithm is applied to simulate the time evolution of the particles' position.

3. The mean values of the pixels grouped in the same cluster are computed and these values are assigned to the initial position of the pixels in the original image.

4. Transform the multi-level mask into a binary mask.
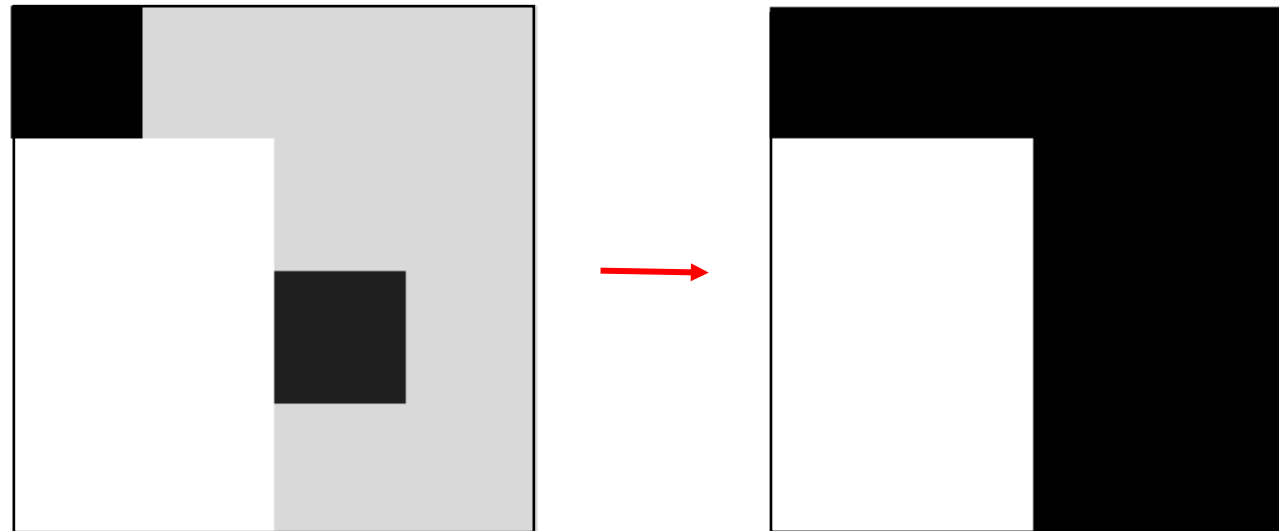
Datasets

We consider three different datasets:

- The HL60 cell nuclei dataset consists of 2D time-lapse video of cell nuclei realized with the Fluorescence Microscopy technique;

- The brain tumor dataset consists 3D in multi-parametric MRI of patients affected by glioblastoma;

- The thigh muscles dataset consists of 3D MRI scans of left and right thighs of healthy subjects.
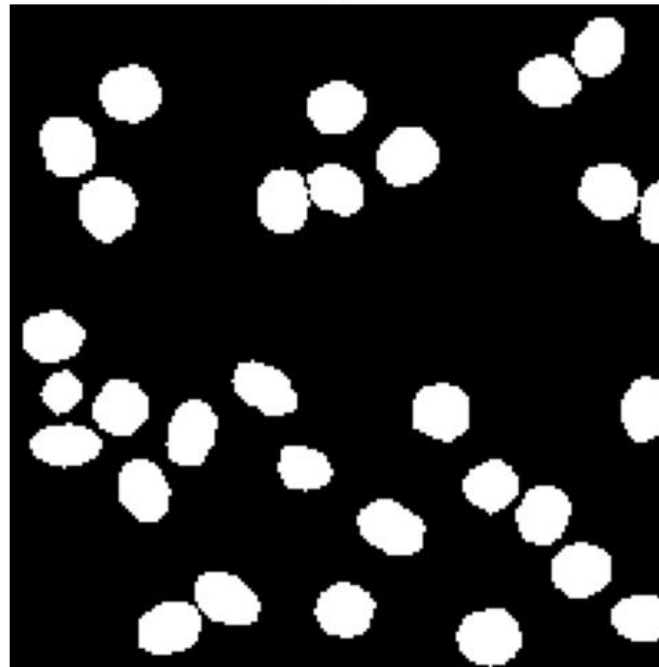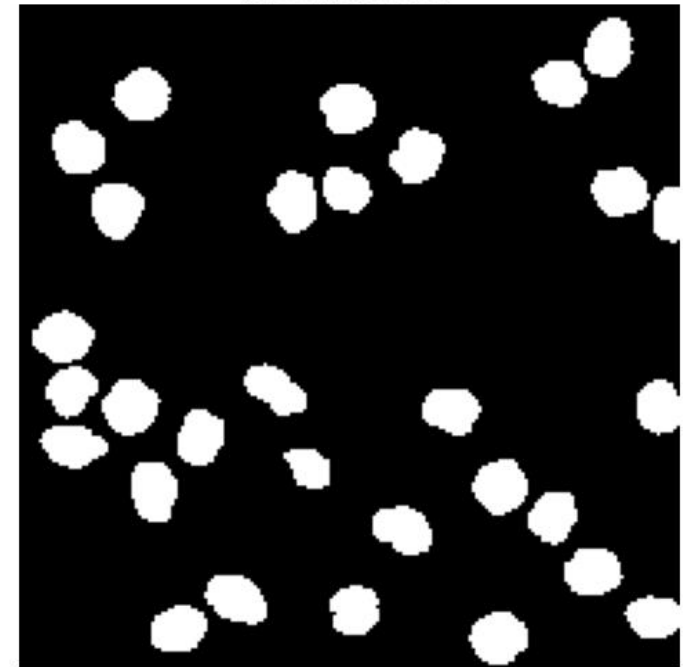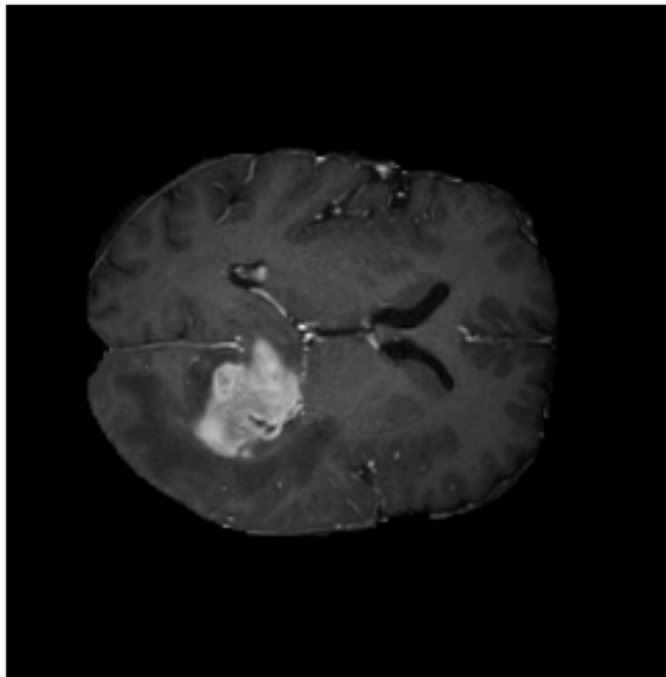
Results on the HL60 cell nuclei dataset:

True

Reconstructed
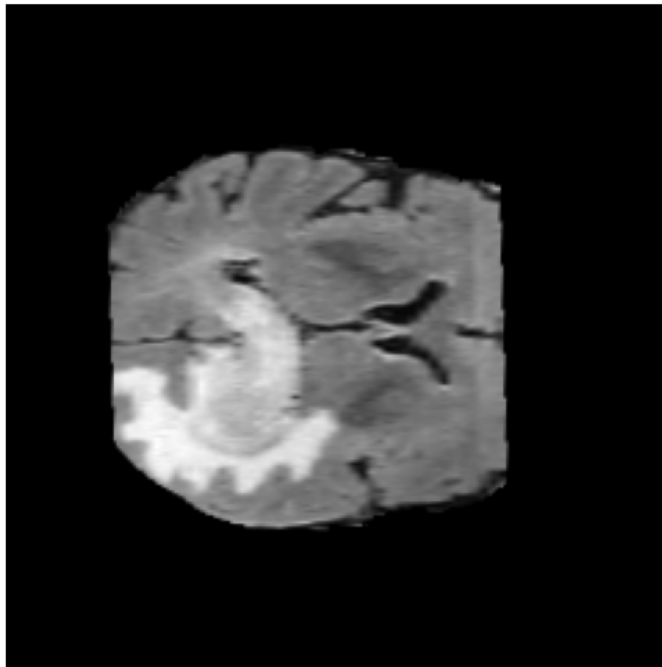
Results on the brain tumor dataset for the "non-enhancing core" segmentation task:

Results – brain tumor segmentation

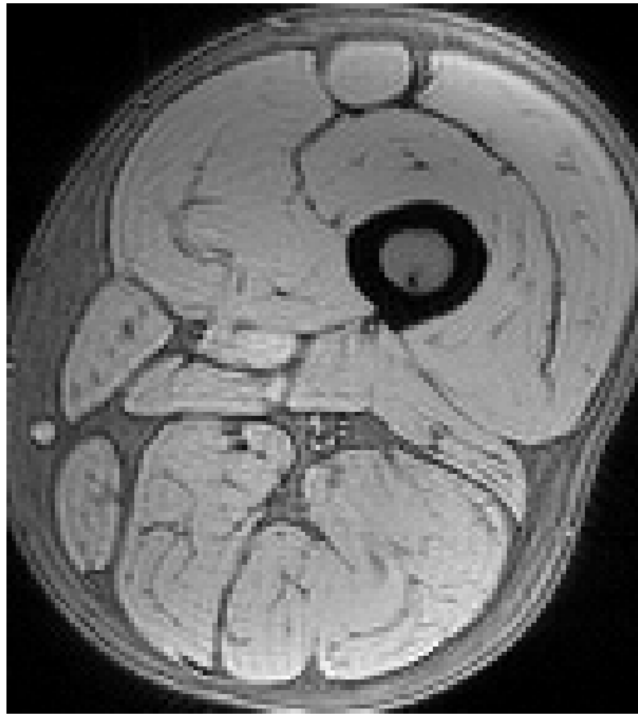Results on the brain tumor dataset for the "edema" segmentation task:
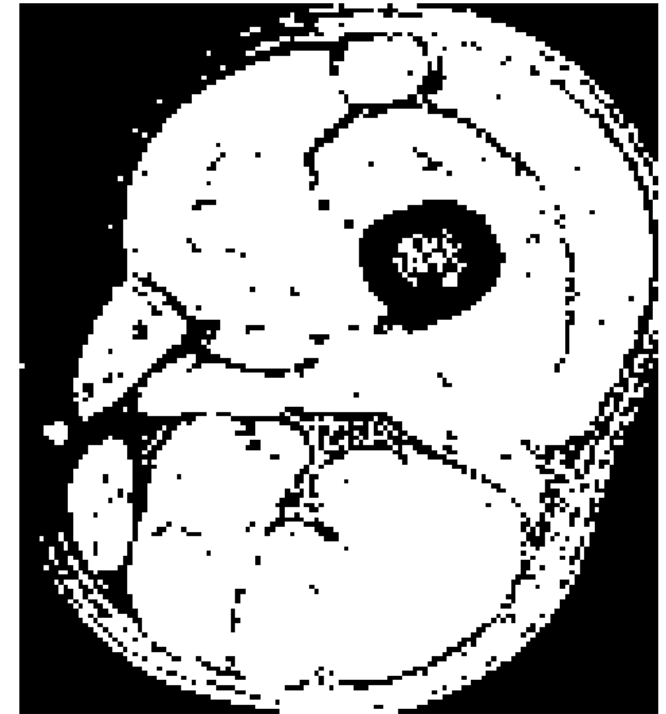
Results on the thigh muscles dataset:

True   Reconstructed

Conclusions

- We developed a mean field-based algorithm for biomedical images segmentations;

- through the Boltzmann formulation we can compute the final position of the clusters with a time-efficient algorithm;

- we presented some applications to biomedical image segmentation problems, for three research contexts;

- the technique reached quite good performances on the cell nuclei dataset and on the brain tumor dataset;

- to improve the results even on more complex datasets, such as the thigh muscles dataset, we plan to add a morphological refinement step.

Thank you!