# Electronic Supplementary Information

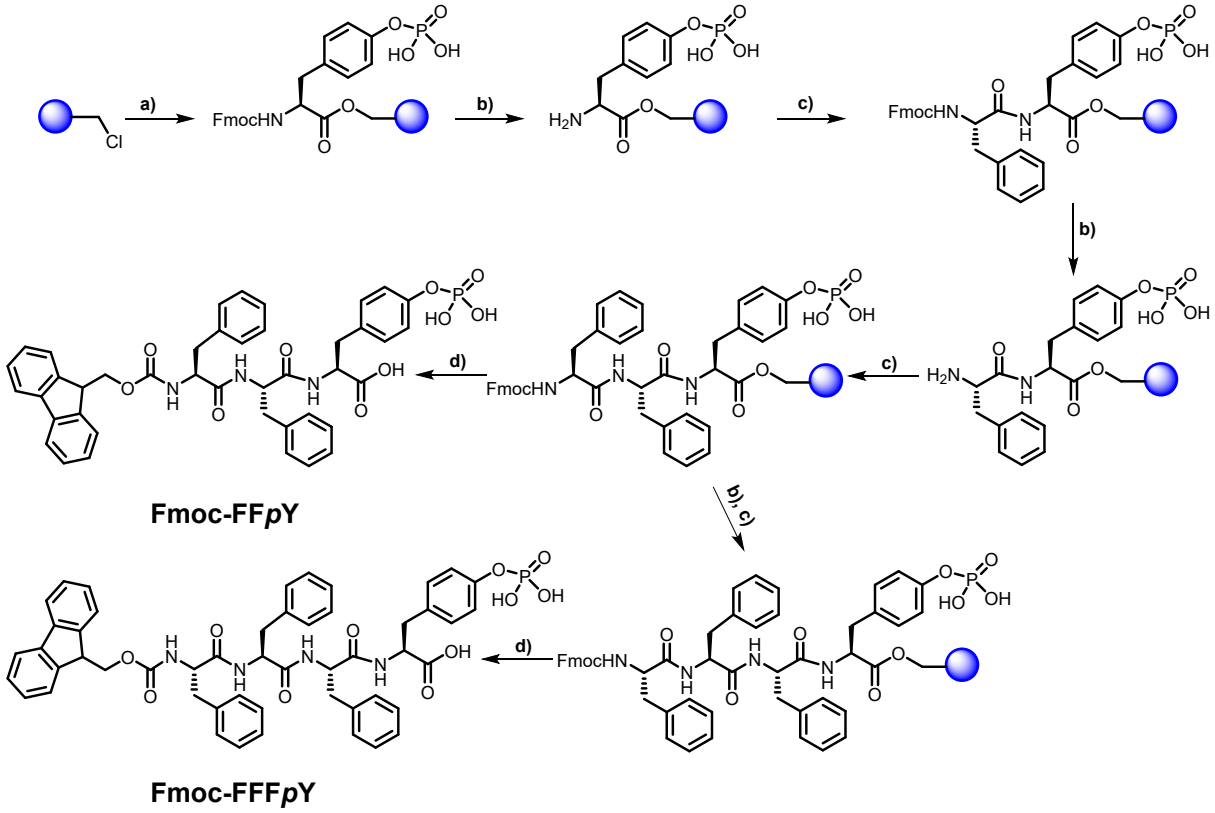
## Model to Rationalize and Predict the Formation of Organic Patterns originating from an Enzyme-Assisted Self-Assembly Liesegang-like Process of Peptides in a Host Hydrogel

Jean-Yves Runser, Shahaji H. More, Fatima Fneich, Timothée Boutfol, Pierre Weiss, Marc Schmutz, Bernard Senger, Loïc Jierry, Pierre Schaaf


| | | |
|---|---|---|
| **1.** | **Synthesis of Fmoc-FF*p*Y and Fmoc-FFF*p*Y……...** | **2** |
| | Brief description of the experimental procedure…… | 2 |
| | Kaiser test…….. | 3 |
| | Protocol for the preparation of the Kaiser test solutions…… | 3 |
| **2.** | **Characterization of Fmoc-FF*p*Y and Fmoc-FFF*p*Y……** | **4** |
| | $^{1}$H and $^{31}$P NMR analyses……. | 4 |
| | High Resolution Mass Spectrometry (HRMS)……. | 9 |
| | Analytical HPLC…….. | 11 |
| **3.** | **Numerical solutions of equations (2)-(6)…….....** | **13** |
| **4.** | **Parameters used in the simulations reported in the main text…….** | **14** |
| | **Table S1……** | **15** |
| **5.** | **Experiments performed in a crosslinked hyaluronic acid host gel..** | **15** |
| **6.** | **Script of the software……...** | **16** |
| | **Figures……...** | **31** |
| | Figure S1……... | 32 |
| | Figure S2……... | 35 |
| | Figure S3……... | 36 |
| | Figure S4……... | 37 |
| | Figure S5……... | 39 |
| | Figures S6, S6A, S6B, S6C, S6D, SE6, S6F and S6G……... | 40 |

# 1. Synthesis of Fmoc-FF*p*Y and Fmoc-FFF*p*Y

Fmoc-FF*p*Y and Fmoc-FFF*p*Y peptides were synthesized using earlier reported procedure (*Polymers*, 2021, **13**, 1793). In brief, standard "Fmoc strategy" of solid phase peptide synthesis (SPPS) was utilized on 2-CTC resin. After completion of Fmoc-FF*p*Y peptide synthesis as reported, an half of the resin was taken for Fmoc-FFF*p*Y synthesis. The route of synthesis is given in the scheme 1 below and experimental procedures and characterization with various techniques are shown below.




**Scheme 1.** Synthetic route for Fmoc-FF*p*Y and Fmoc-FFF*p*Y precursor peptide.

**Brief description of the experimental procedure:**

**a**) Loading of resin: 1.20 g of 2-CTC resin (1.6 mmol/g; 1.92 mmol) was soaked in a solid phase extraction (SPE) tube in dry DCM (16 mL) for 2h. DCM was removed and a solution of Fmoc-

*p*Y-OH (2.76 g, 5.76 mmol) and DIPEA (4 mL) in dry DCM (16 mL) was added and the tube was stirred overnight at RT. Solution was filtered off and beads were washed twice with methanol, and stirred for 1 hour at room temperature in 15 mL solution of DCM: MeOH: DIPEA (80:15:05) and then filtered off.

**b**) A solution of 20 % piperidine in DMF (16 mL) was added to the beads and the tube was stirred for 30 minutes. The solution was filtered off and the beads were washed thoroughly with DCM, DMF, DCM (5 Times with each solvent). A Kaiser test* was performed and it was positive (free amine present).

**c**) A solution of Fmoc-F-OH (2.23 g, 5.76mmol, 3eq), HBTU (2.19g, 3eq) and HOBt (790 mg, 3eq) and DIPEA (4 mL) in dry DMF (14 mL) was added. The tube was stirred overnight at RT. The solution was filtered off and the beads were washed thoroughly with DCM, DMF, DCM (5 Times with each solvent). A Kaiser test* was performed and test was negative (free amine absent).

**d**) A solution of TFA: $H_2O$: TIPS (95 : 2.5 : 2.5) (12 mL) was added to the beads and the tube was stirred for 2 hours. The filtrate was recovered and evaporated on rotatory evaporator until a thick oil is obtained into diethyl ether (25 mL) was added and peptide was precipitated by keeping solution at 4°C for overnight. The resulting precipitate was recovered as a white powder (580 mg 630 mmol, 33% vs resin).

* **Kaiser test:** (Ninhydrin Test): three drops of solution **A**, **B** and **C** is added in a test tube containing more than 10 beads of the resin. The test tube is heated at 100 °C. When the Kaiser test is positive, the beads and the solution turn into blue color. This indicates the presence of free amine that conclude Fmoc deprotection step succeeded or a failed coupling step. When the Kaiser test is negative, the beads and solution remain uncolored. This indicates that the coupling step is completed or a failed the Fmoc deprotection step. In case of a Fmoc deprotection or a coupling step failed, the step is repeated after a washing the beads until the Kaiser test leads to the required color.

**Protocol for preparation of the Kaiser test solution:**

**Solution A:**

- Dissolve 16.5 mg of KCN in 25 mL of distilled water.

- Dilute 1.0 mL of above solution with 49 mL of pyridine.
- Pour it into a small reagent bottle and label it “A”

**Solution B:**

- Dissolve 1.0 g of ninhydrin in 20 mL of BuOH.
- Pour into a small reagent bottle and label it as “B”.

**Solution C:**

- Dissolve 40 g of PhOH in 20 mL of BuOH.
- Pour it into a small reagent bottle and label it “C”.

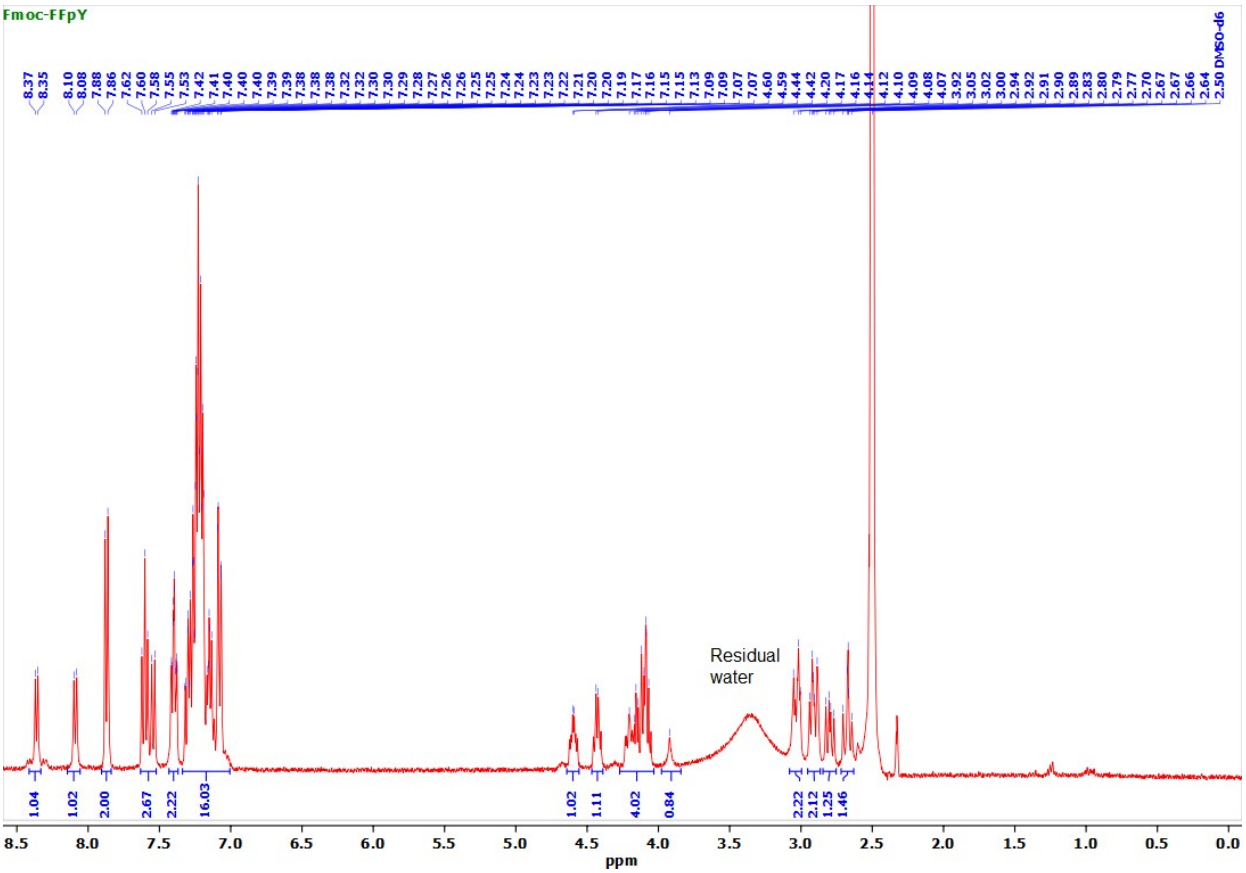
# 2. Characterization of Fmoc-FF*p*Y and Fmoc-FFF*p*Y

Note: the whole characterization of Fmoc-FF*p*Y was identical to previously already published description (*Polymers*, 2021, **13**, 1793).

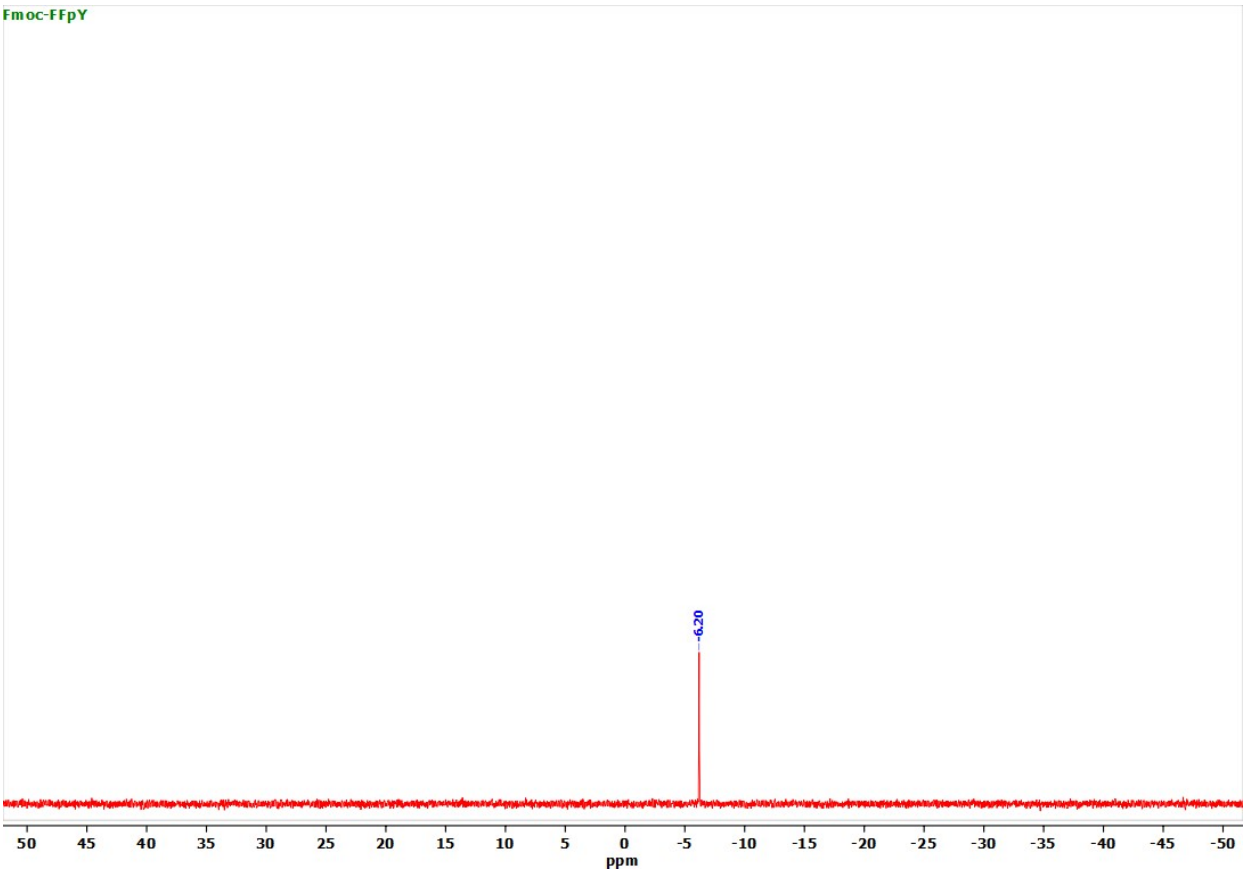
### $^{1}$H and $^{31}$P NMR analyses

NMR spectra were recorded on a BRUKER 400 spectrometer at 25°C.The chemical shifts were provided in ppm and coupling constants J were given in Hz. The following notation was used for the description of $^{1}$H-NMR spectra: singlet (s), doublet (d), triplet (t), multiplet (m).

**Fmoc-FF*p*Y:** $^{1}$H NMR (400 MHz, DMSO-$d_6$) Chemical shift (δ) in ppm = 8.36 (d, $J$ = 7.6 Hz, 1H), 8.09 (d, $J$ = 8.2 Hz, 1H), 7.87 (d, $J$ = 7.5 Hz, 2H), 7.63 – 7.52 (m, 3H), 7.43 – 7.37 (m, 2H), 7.34 – 7.01 (m, 16H), 4.60 (d, $J$ = 4.6 Hz, 1H), 4.43 (d, $J$ = 6.0 Hz, 1H), 4.27 – 4.03 (m, 4H), 3.92 (s, 1H), 3.04 (d, $J$ = 13.8 Hz, 2H), 2.92 (dd, $J$ = 12.4, 9.2 Hz, 2H), 2.80 (dd, $J$ = 14.0, 9.4 Hz, 1H), 2.72 – 2.63 (m, 1H). $^{31}$P NMR Chemical shift (δ) in ppm = -6.20.

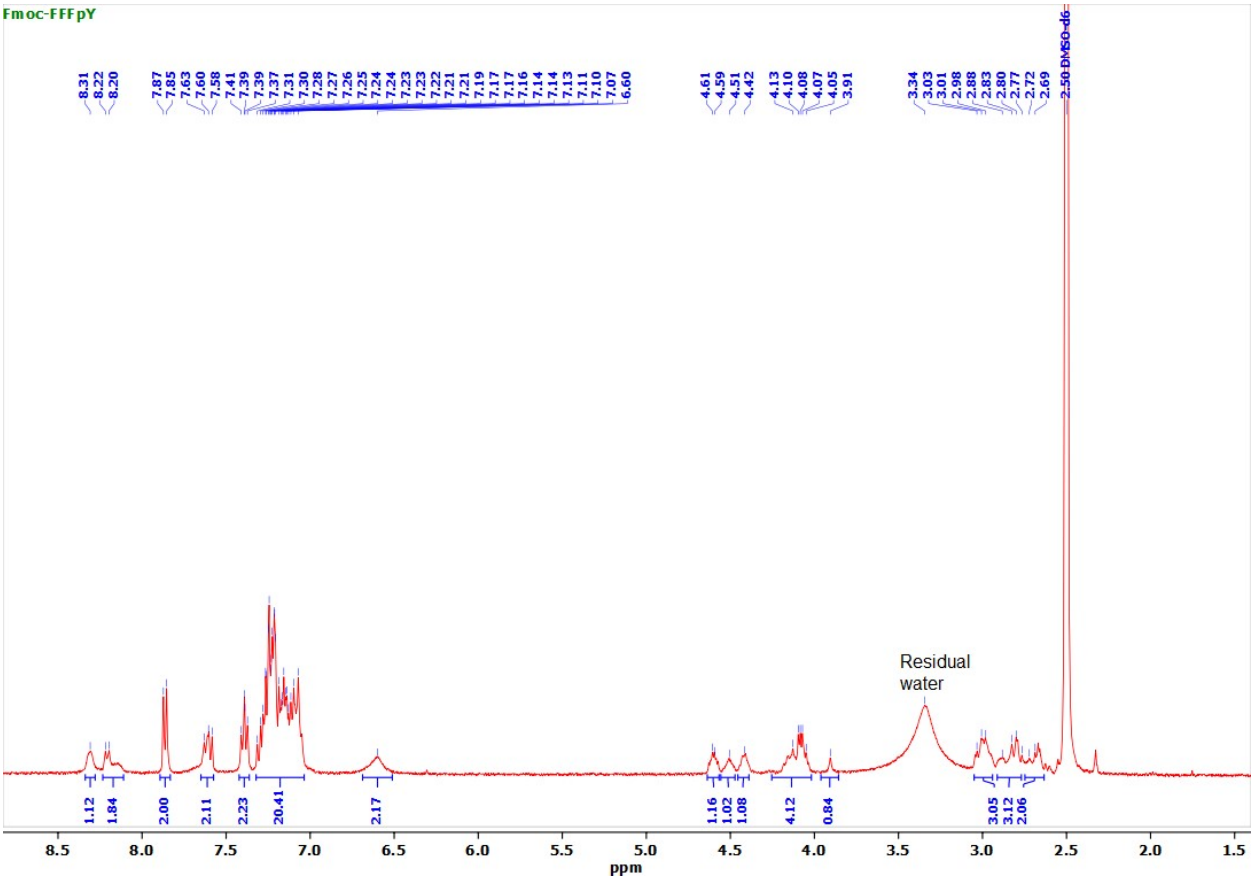



Fmoc-FF*p*Y: $^{1}$H NMR (400 MHz, DMSO-$d_6$)

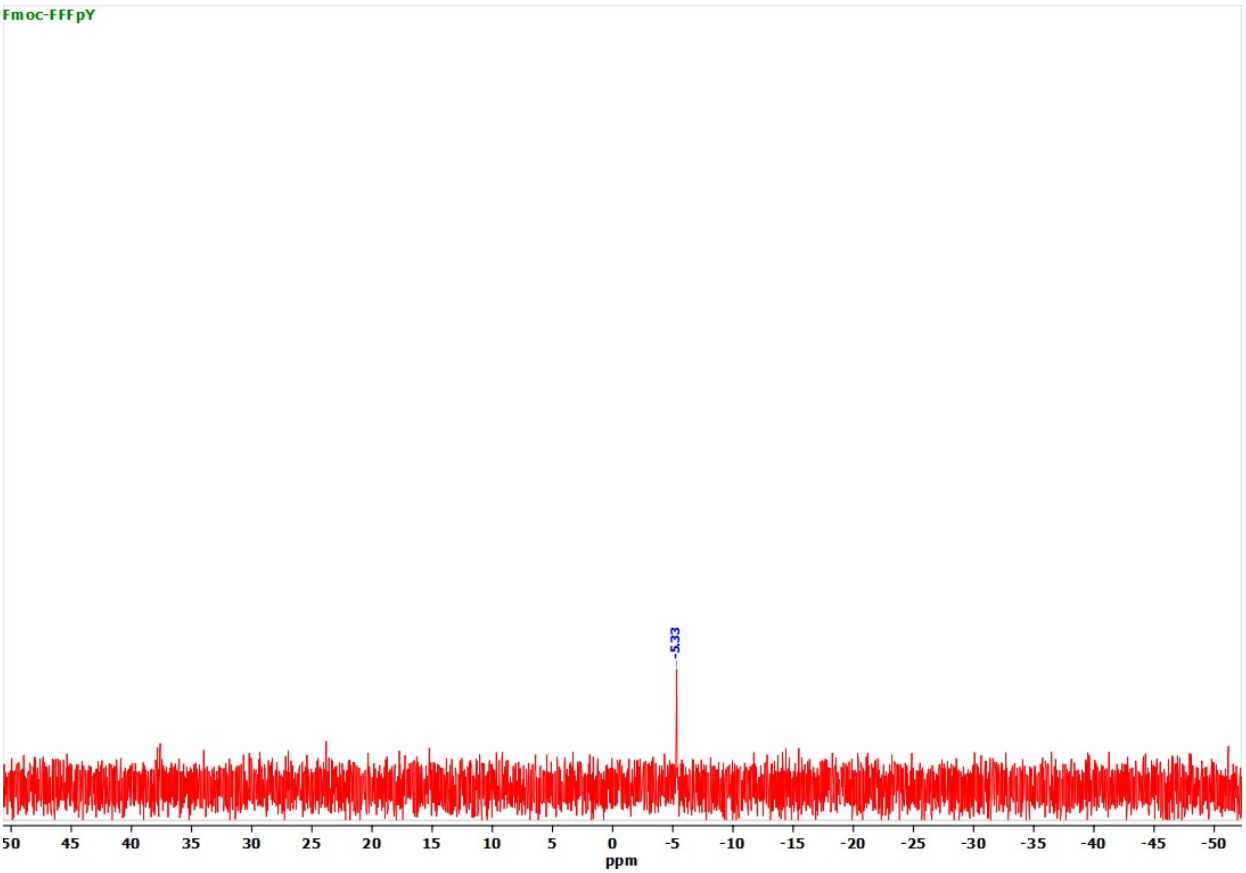



Fmoc-FF*p*Y: $^{31}$P NMR (202 MHz, DMSO-$d_6$)

**Fmoc-FFF*p*Y:** $^{1}$H NMR (400 MHz, DMSO-$d_6$) Chemical shift (δ) in ppm = 8.31 (s, 1H), 8.21 (d, $J$ = 8.4 Hz, 2H), 7.86 (d, $J$ = 7.5 Hz, 2H), 7.65 – 7.57 (m, 2H), 7.39 (t, $J$ = 7.3 Hz, 2H), 7.32 – 7.04 (m, 20H), 6.60 (s, 2H), 4.60 (d, $J$ = 5.9 Hz, 1H), 4.51 (s, 1H), 4.42 (s, 1H), 4.25 – 4.02 (m, 4H), 3.91 (s, 1H), 2.99 (d, $J$ = 10.2 Hz, 3H), 2.91 – 2.77 (m, 3H), 2.70 (d, $J$ = 13.8 Hz, 2H). $^{31}$P NMR Chemical shift (δ) in ppm = -5.33.



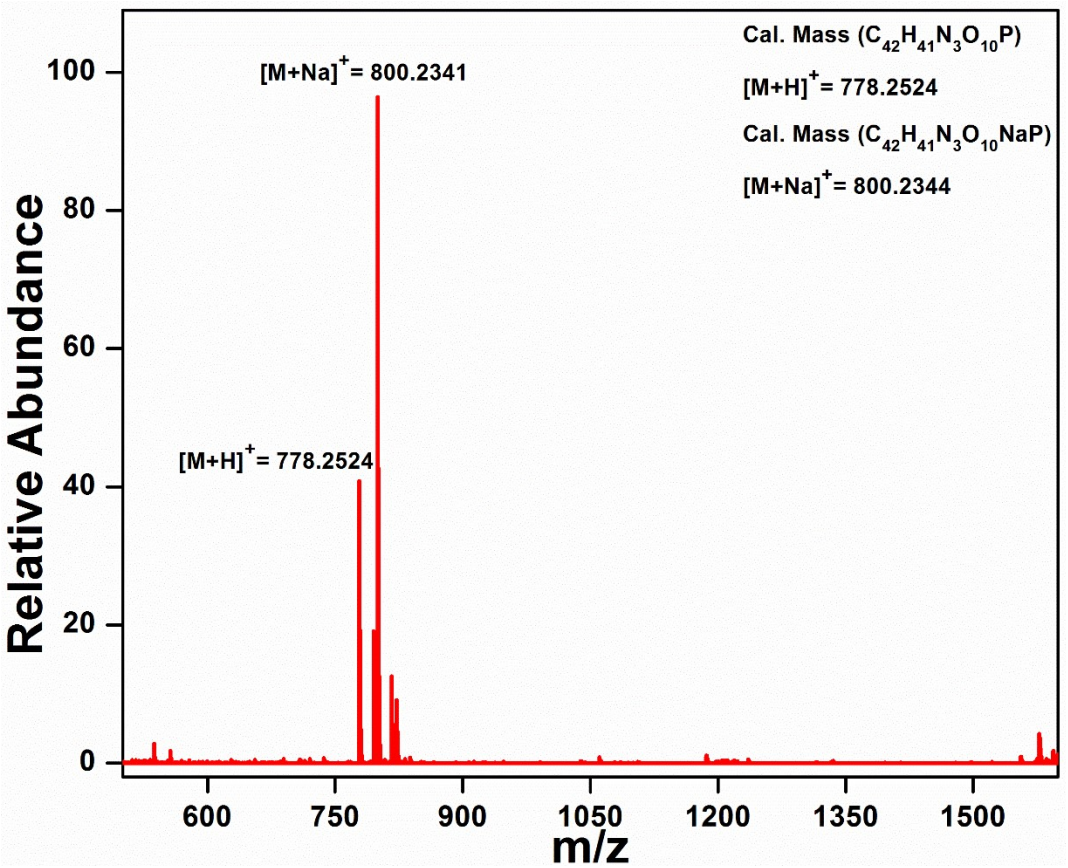

Fmoc-FFF*p*Y: $^{1}$H NMR (400 MHz, DMSO-$d_6$)



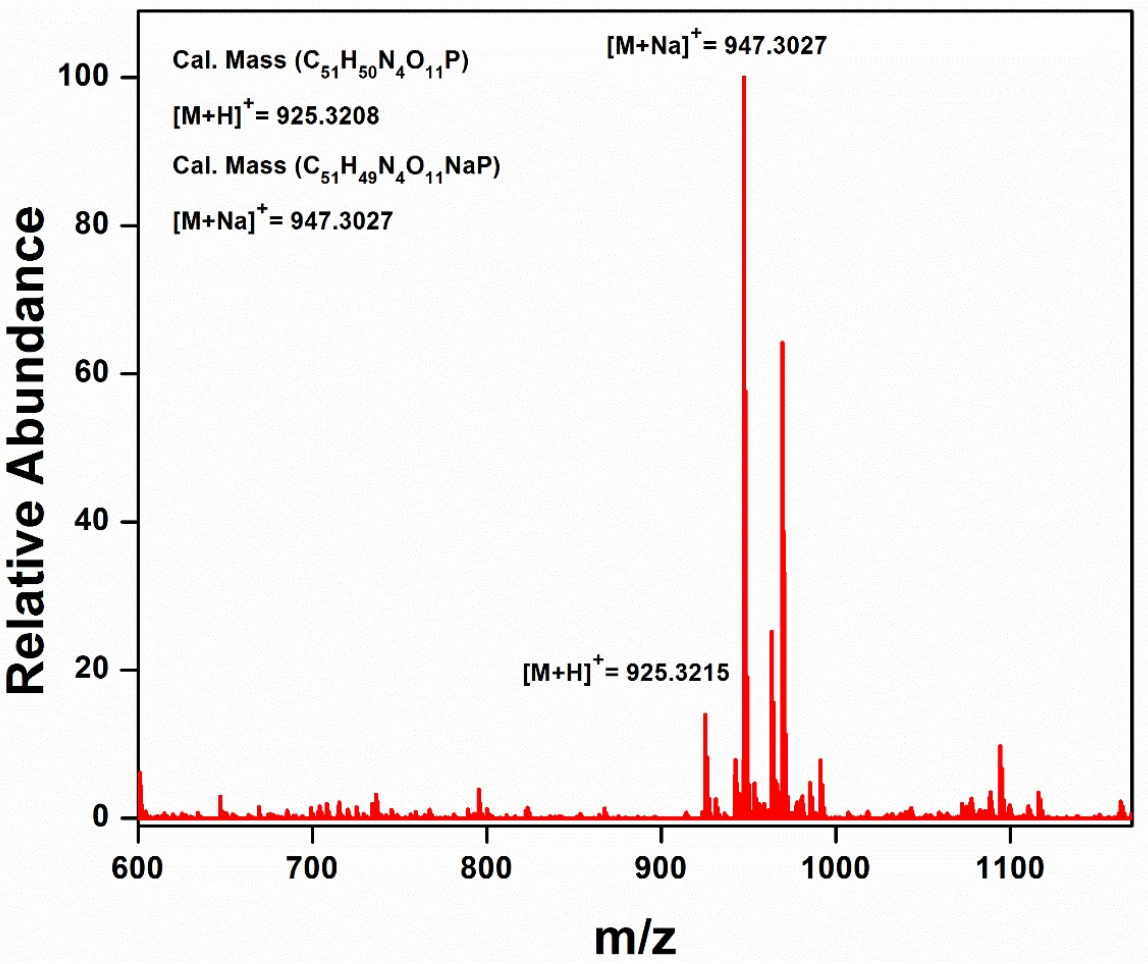

Fmoc-FFF*p*Y: $^{31}$P NMR (202 MHz, DMSO-$d_6$)

**High Resolution Mass Spectrometry (HRMS)**

**Fmoc-FF*p*Y:** Calculated mass for molecular formula $C_{42}H_{41}O_{10}N_3P$ $[M+H]^+$ = 778.2524, Observed mass = 778.2524. Calculated mass for molecular formula $C_{42}H_{40}O_{10}N_3NaP$ $[M+Na]^+$ = 800.2344, Observed mass = 800.2341.



**Fmoc-FFF*p*Y:** Calculated mass for molecular formula $C_{51}H_{50}O_{11}N_4P$ $[M+H]^+ = 925.3208$, Observed mass = 925.3015. Calculated mass for molecular formula $C_{51}H_{49}O_{11}N_4NaP$ $[M+Na]^+ = 947.3027$, Observed mass = 947.3027.



**Analytical HPLC**

Analytical HPLC experiments were carried out with a 1100 Series from Agilent technologies. Chromatograms were recorded and analyzed by the software OpenLab Agilent 1100. Samples were prepared in ACN/$H_2O$ (50/50) and all solutions are filtrated through 0.2 µm syringe filter before to make injection.

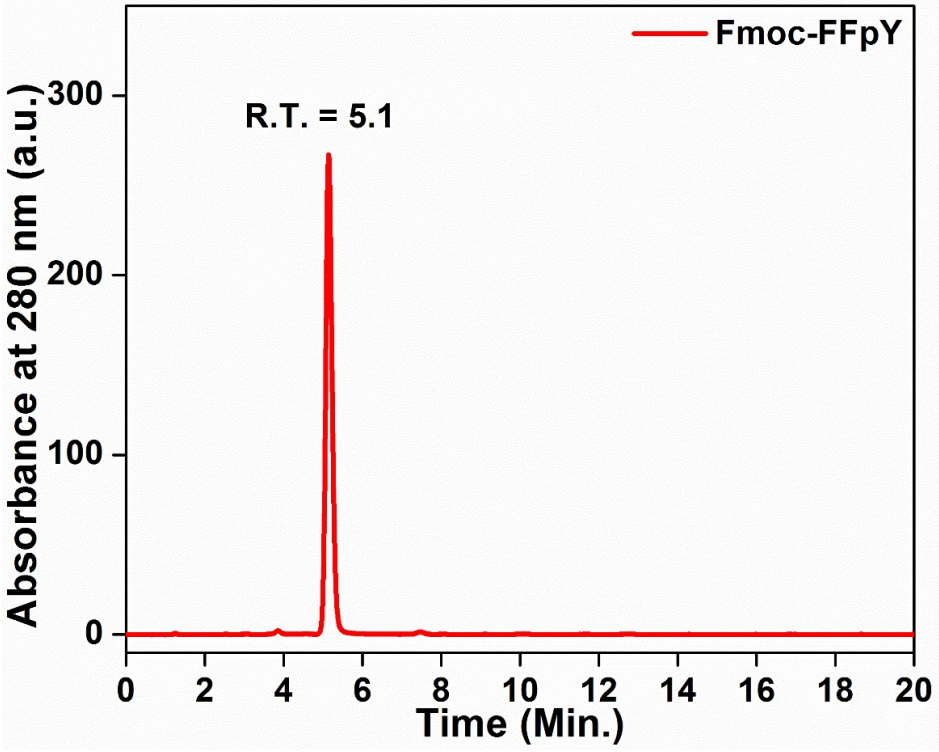
Stationary phase: Interchim Uptsisphere 5µm C18HP; size 100 × 4.6mm.

Elution conditions: isocratic ACN/$H_2O$+0.1%TFA (50/50); Flow rate: 1 mL/min.

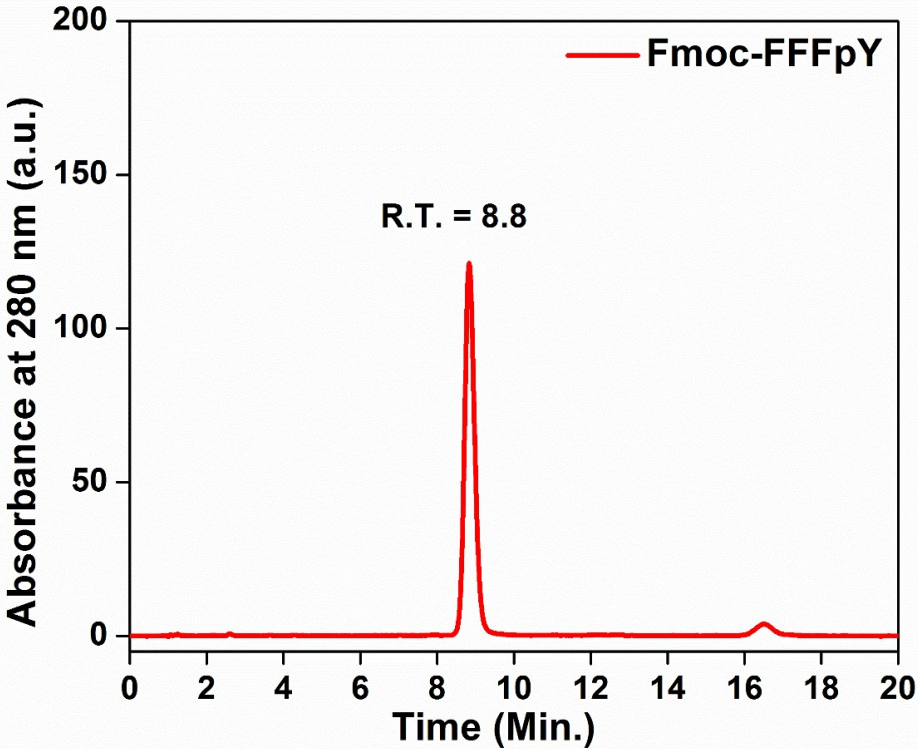
Temperature: 25 °C

Injection: 20 µL

Sample concentration: 0.25 mg/mL




HPLC Profile of Fmoc-FF*p*Y




HPLC Profile of Fmoc-FFF*p*Y

# 3. Numerical solutions of equations (2)-(6)

Let us concentrate on the numerical solutions of equations (2)-(6). In order to save computer time, we restrained our simulations to 2 dimensions. Compared to 3 dimensions, it does not change the physics of the problem in contrary to a 1 dimensional problem. We solved these equations by a finite element method both in space and time. Our space was divided in $n_x$ lines parallel to the $x$ axis ($n_x = 260$ in our simulations) and $n_y$ lines parallel to the $y$ axis ($n_y = 6$ in our simulations). The lines 0 up to $n_{x_g}$ parallel to the $x$ axis correspond to the solution and the lines $n_{x_g} + 1$ up to $n_x$ correspond to the gel ($n_{x_g} = 20$ in our simulations) At both ends parallel to the $x$ axis we use total reflection boundary conditions and at both ends parallel to the $y$ axis we use periodic boundary conditions. These lines delimit square cells, each being identified by the indices $i$ and $j$ . The time is subdivided into small intervals $\delta t$. The concentration within a cell is $c(i,j)$ and the diffusion coefficient $D(i,j)$. For the sake of simplicity, we omit here the index which would indicate which species is concerned.

Because the model consists in two parts (solution and gel), $D(i,j)$ may be different in these two compartments. Therefore, it is mandatory to introduce the diffusion coefficient through the wall separating two adjacent cells in place of the coefficients within the two cells. Assume that $D(i,j)$ varies with index $i$ (along the $x$ axis) whereas it is insensitive to index $j$ (along the $y$ axis). The flux of molecules from cell $(i,j)$ to cell $(i \pm 1,j)$ depends then on the harmonic mean of $D(i,j)$ and $D(i \pm 1,j)$, *i.e.*

$$\frac{1}{D^h(i,i \pm 1,j,j)} = \frac{1}{2}\left(\frac{1}{D(i,j)} + \frac{1}{D(i \pm 1,j)}\right)$$

where the superscript "h" stands for "harmonic". Similarly

$$\frac{1}{D^h(i,i,j,j \pm 1)} = \frac{1}{2}\left(\frac{1}{D(i,j)} + \frac{1}{D(i,j \pm 1)}\right)$$

useful in the case where the medium would be anisotropic along the $y$ axis too. Note that introducing the harmonic mean is required by the thermodynamics. Indeed, the diffusion of any species must lead to a uniform concentration of it at equilibrium whatever the local mobility of the molecules under consideration.

The flux $J$ from the cell $(i,j)$ to cell $(i \pm 1,j)$ or the cell $(i,j \pm 1)$ is the given by Fick's law, respectively written either

$$J = D^h(i,i \pm 1,j,j)[c(i \pm 1,j) - c(i,j)]$$

or

$$J = D^h(i,i,j,j \pm 1)[c(i,j \pm 1) - c(i,j)]$$

Accordingly, the variation $\delta c(i,j,t)$ of the concentration within cell $(i,j)$ during the time span $\delta t$ due to diffusion alone writes

$$\delta c(i,j,t) = \left\{ \sum_{n \in \{-1,1\}} D^h(i,i+n,j,j)[c(i+n,j,t) - c(i,j,t))] + \sum_{n \in \{-1,1\}} D^h(i,i,j,j+n)[c(i,j+n,t) - c(i,j,t))] \right\} \delta t$$

Equation (3) can then be written

$$\delta c_{PH}(i,j,t) = \left[ \Delta c_{PH}(i,j,t) + k_E(c_E(i,j,t) + c_{EF}(i,j,t)).c_{PH}(i,j,t) - k_a.c_{PH}(i,j,t).r(c_{PH}(i,j,t)) - k_{as}.c_{PH}(i,j,t).c_{P} \right] .\delta t$$

where

$$\Delta c_{PH}(i,j,t) = \left\{ \sum_{n \in \{-1,1\}} D_{PH}^h(i,i+n,j,j)\left[c_{PH}(i+n,j,t) - c_{PH}(i,j,t))\right] + \sum_{n \in \{-1,1\}} D_{PH}^h(i,i,j,j+n)\left[c_{PH}(i,j+n,t) - c_{PH}(i,j,t))\right] \right\}$$

Similarly for equations (4), (5) and (6).

The function $r(c_{PH}(i,j,t))$ is calculated as described in the main text.

# 4. Parameters used in the simulations reported in the main text

All simulations were performed on a two-dimensional lattice, the x axis being perpendicular to the gel-solution interface and the y axis being parallel to it. The simulated system extends from y=0 to y=6 and from x=0 to x=260. The solution extends from 0 to x=20 ( $n_{x_g}$). All simulations were performed with a diffusion coefficient of the precursor peptides $D_{PP}$ equal to 0.5, a diffusion coefficient of hydrogelator peptides $D_{PD}$ equal to 0.4 and a diffusion coefficient of the enzymes equal $D_E$ equal to 0.1. The diffusion coefficients of all species were always the same in the solution and in the gel even if the model could take different diffusion coefficients into account. In all simulations reported in the main text of the article the enzymatic activity was always 5 times larger in the solution than in the gel. The nucleation constant $k_a$ and the association constant between free peptides and self-assemblies $k_{as}$ were always equal to 0.1 both in the solution and in the gel. The desorption constant $k_{ad}$ was set equal to 0.0001. The constant of enzymatic activity in the gel $k_E$ was equal to 10. The adsorption constant of enzymes on self-assemblies $k_{ea}$ was

equal to 0.1 and the enzyme desorption constant $k_{ed}$ equal to 0.01. The parameter $x_0$ of the initial enzyme profile was always set equal to 5. The other parameters are given in table 1 below.

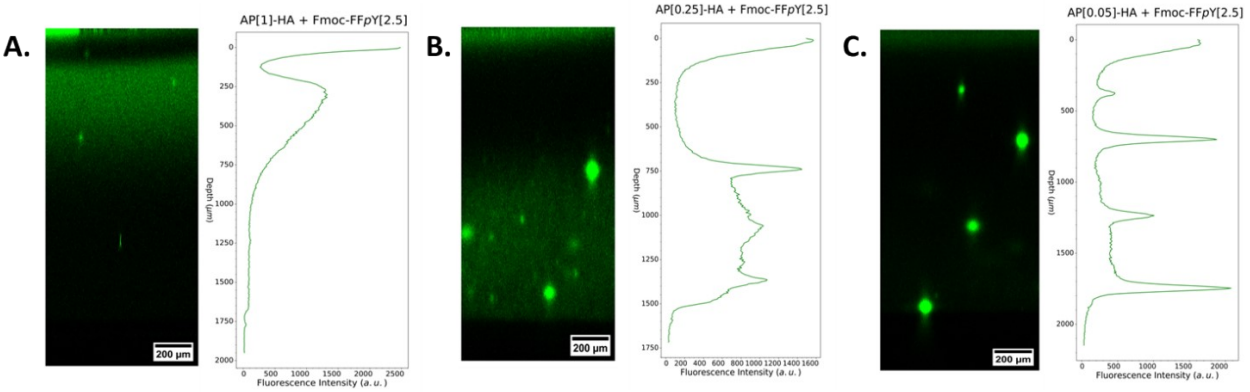
| | Figure 3 | | | | | Figure 4 | | | Figure 5 | | | | | | Figure 6 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | a. | b. | c. | d. | e. | Green | Orange | Blue | Blue | Orange | Green | Red | Purple | Braun | Green | Orange | Blue |
| $C_E$ (initial concentration of free enzymes) | 7 | 3 | 1 | 0.1 | 0.001 | 5 | 5 | 5 | 3 | 1 | 0.5 | 0.1 | 0.05 | 0.001 | 1 | 1 | 1 |
| $C_{PP}$ (initial concentration of hydrogelator peptides) | 5 | 5 | 5 | 5 | 5 | 10 | 5 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| $P_0$ (critical hydrogelator concentration for self-assembly) | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.05 | 0.05 | 0.05 |
| $P_1$ (range of concentration of hydrogelators on which nucleation occurs) | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| $x_1$ (Initial enzyme profile) | 15 | 15 | 15 | 15 | 15 | 35 | 35 | 35 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |

**Table S1.** List of parameters used and their value for each simulation graph provided in the article.

# 5. Experiments performed in a crosslinked hyaluronic acid host gel

We performed experiments by using a crosslinked hyaluronic acid host gel containing AP as enzymes (called AP-HA) onto which a solution of Fmoc-FF*p*Y was deposited similarly to what was done with AP-HPMC host gel. The AP-HA host gel was prepared as described in the Material and Methods section given in the manuscript. By keeping the precursor peptide Fmoc-FF*p*Y concentration at 2.5 mg/mL, we performed experiments at three different enzyme (*i.e.* AP) concentrations: (**A**) 1 mg/mL, (**B**) 0.25 mg/mL and (**C**) 0.05 mg/mL (see illustrations below). The AP solution was complemented with 0.005 mg/mL of ThT. On the left side of each illustration below **A**, **B** or **C**, is shown the CLSM image of a z-stack and on the right side is the corresponding

self-assembly profile. As in the case of the AP-HPMC host gel, at high AP concentration ([AP] = 1 mg/mL) one observes (**A**) a continuous self-assembly profile presenting a maximum at the gel/solution interface followed by a depletion zone and the secondary maximum in the gel (profile of type II). As the enzyme concentration decreases ([AP] = 0.25 mg/mL), (**B**) microglobules appear in the continuous profile (type III) and finally (**C**) when the enzyme concentration is still further decreased ([AP] = 0.05 mg/mL) only globules remain visible (type IV).




# 6. Script of the software

```
# Random Matrix

# Attention: in the software the x axes is parallel to the gel/solution interface

# The y axis is perpendicular to the gel/solution interface. The solution extends from y=0 up to y=20

# The gel extends from y=21 up to y=260


from cmath import *

import numpy as np

import matplotlib.pyplot as plt

import random as rd

import pandas as pd



#creation of matrices
```

```
X=6

Y=260

# the Y is perpendicular to the gel/solution interface

# the X axis is parallel to the gel/solution interface


MPP=np.zeros((X,Y)) #precursor peptides

tMPP=np.zeros((X,Y))

MPD=np.zeros((X,Y)) #dephosphorylated peptides

tMPD=np.zeros((X,Y))

MPA=np.zeros((X,Y)) #self-assembled peptides

MPT=np.zeros((X,Y)) #total number of peptides

tMPA=np.zeros((X,Y))

tME=np.zeros((X,Y))

ME=np.zeros((X,Y)) #free enzymes

MEF=np.zeros((X,Y)) #density of fix enzymes

tMEF=np.zeros((X,Y))

MET=np.zeros((X,Y)) # total concentratioin of enzymes

Dpp=np.zeros((X,Y))

Dpd=np.zeros((X,Y))

DPP=np.zeros((X,Y,X,Y)) #diffusion coefficient of phosphorylated peptides

DPD=np.zeros((X,Y,X,Y)) # diffusion coefficient of dephosphorylated peptides

De=np.zeros((X,Y))

DE=np.zeros((X,Y,X,Y)) # diffusion coefficient of free enzymes

KE=np.zeros((X,Y)) #Enzyme activity coefficient

P0N=np.zeros((X,Y)) # critical nucleation concentration in the gel




MI1=np.zeros((X,Y)) # intermediate matrix 1
```

```
# for the plot

ix=np.zeros(Y)

iy1=np.zeros(X)

iy2=np.zeros(Y)

iy3=np.zeros(Y)

iy4=np.zeros(Y)

iy5=np.zeros(Y)

iy6=np.zeros(Y)

ix=np.zeros(Y)


for i in range(0,Y):

    ix[i]=i



tfin=100000000.0



dt=0.01 # time increment



#paramètres


YL=20  # begining of the gel. The gel extends from YL to Y-1

CPP=5.0 #initial concentration of phosphorylated peptides

xDPP=0.5 #diffusion coefficient of phosphorylated peptides dans le gel

xDPD=0.4 #diffusion coefficient of dephosphorylated peptides dans le gel

xDE=0.1 #diffusion coefficient of free enzymes dans le gel

rke=1.0 # ratio of enzyme activity coefficient in solution and in the matrix

rDif=1.0 # coefficients de diffusion solution/gel
```

```
xDPP=xDPP*dt

xDPD=xDPD*dt

xDE=xDE*dt


P0=0.05 # critical concentration for self-assembly denoted as CC in the text

P1=0.01

ka=0.1 #self-assembly constant during nucleation

kas=0.1 # association constant between free dephosphorylated peptides and self-assembly

kad1=0.0 # desorption constant from an aggregate, proportional to MPA

el=0.1/7000 # initial concentration of free enzymes

kea=0.0 # adsorption constant of enzymes from peptide aggregates

ked=0.0 # desorption constant of enzymes from peptide aggregates


ke=10.0  # enzymatic constant



kad=ka*dt

kas=kas*dt

ke=ke*dt

kea=kea*dt

ked=ked*dt

kad1=kad1*dt


for j in range(0,Y):
    if (j<YL):
        xq=rDif
        xke=rke
        P0I=0
    else:
        xq=1.0
```

```
            xke=1.0
            P0I=P0
        for i in range (0,X):
            Dpp[i][j]=xDPP*xq
            Dpd[i][j]=xDPD*xq
            De[i][j]=xDE*xq
            KE[i][j]=ke*xke
            P0N[i][j]=P0I

#diffusion coefficients between cells

#inside of the system
for i in range (1,X-1):
        for j in range(1,Y-1):
            DPP[i-1][j][i][j]=2*(Dpp[i-1][j]*Dpp[i][j]/(Dpp[i-1][j]+Dpp[i][j]))
            DPD[i-1][j][i][j]=2*(Dpd[i-1][j]*Dpd[i][j]/(Dpd[i-1][j]+Dpd[i][j]))
            DE[i-1][j][i][j]=2*(De[i-1][j]*De[i][j]/(De[i-1][j]+De[i][j]))
            DPP[i][j][i-1][j]=DPP[i-1][j][i][j]
            DPD[i][j][i-1][j]=DPD[i-1][j][i][j]
            DE[i][j][i-1][j]=DE[i-1][j][i][j]
            DPP[i+1][j][i][j]=2*(Dpp[i+1][j]*Dpp[i][j]/(Dpp[i+1][j]+Dpp[i][j]))
            DPD[i+1][j][i][j]=2*(Dpd[i+1][j]*Dpd[i][j]/(Dpd[i+1][j]+Dpd[i][j]))
            DE[i+1][j][i][j]=2*(De[i+1][j]*De[i][j]/(De[i+1][j]+De[i][j]))
            DPP[i][j][i+1][j]=DPP[i+1][j][i][j]
            DPD[i][j][i+1][j]=DPD[i+1][j][i][j]
            DE[i][j][i+1][j]=DE[i+1][j][i][j]

            DPP[i][j-1][i][j]=2*(Dpp[i][j-1]*Dpp[i][j]/(Dpp[i][j-1]+Dpp[i][j]))
            DPD[i][j-1][i][j]=2*(Dpd[i][j-1]*Dpd[i][j]/(Dpd[i][j-1]+Dpd[i][j]))
            DE[i][j-1][i][j]=2*(De[i][j-1]*De[i][j]/(De[i][j-1]+De[i][j]))
            DPP[i][j][i][j-1]=DPP[i][j-1][i][j]
```

```
            DPD[i][j][i][j-1]=DPD[i][j-1][i][j]
            DE[i][j][i][j-1]=DE[i][j-1][i][j]


            DPP[i][j+1][i][j]=2*(Dpp[i][j+1]*Dpp[i][j]/(Dpp[i][j+1]+Dpp[i][j]))
            DPD[i][j+1][i][j]=2*(Dpd[i][j+1]*Dpd[i][j]/(Dpd[i][j+1]+Dpd[i][j]))
            DE[i][j+1][i][j]=2*(De[i][j+1]*De[i][j]/(De[i][j+1]+De[i][j]))
            DPP[i][j][i][j+1]=DPP[i][j+1][i][j]
            DPD[i][j][i][j+1]=DPD[i][j+1][i][j]
            DE[i][j][i][j+1]=DE[i][j+1][i][j]



# at the edges of the system with periodic boundary conditions


for j in range(1,Y):
        DPP[X-1][j][0][j]=2*(Dpp[X-1][j]*Dpp[0][j]/(Dpp[X-1][j]+Dpp[0][j]))
        DPD[X-1][j][0][j]=2*(Dpd[X-1][j]*Dpd[0][j]/(Dpd[X-1][j]+Dpd[0][j]))
        DE[X-1][j][0][j]=2*(De[X-1][j]*De[0][j]/(De[X-1][j]+De[0][j]))
        DPP[0][j][X-1][j]=DPP[X-1][j][0][j]
        DPD[0][j][X-1][j]=DPD[X-1][j][0][j]
        DE[0][j][X-1][j]=DE[X-1][j][0][j]
        DPP[X-1][j-1][X-1][j]=2*(Dpp[X-1][j]*Dpp[X-1][j-1]/(Dpp[X-1][j]+Dpp[X-1][j-1]))
        DPD[X-1][j-1][X-1][j]=2*(Dpd[X-1][j]*Dpd[X-1][j-1]/(Dpd[X-1][j]+Dpd[X-1][j-1]))
        DE[X-1][j-1][X-1][j]=2*(De[X-1][j]*De[X-1][j-1]/(De[X-1][j]+De[X-1][j-1]))
        DPP[X-1][j][X-1][j-1]=DPP[X-1][j-1][X-1][j]
        DPD[X-1][j][X-1][j-1]=DPD[X-1][j-1][X-1][j]
        DE[X-1][j][X-1][j-1]=DE[X-1][j-1][X-1][j]
        DPP[0][j-1][0][j]=2*(Dpp[0][j]*Dpp[0][j-1]/(Dpp[0][j]+Dpp[0][j-1]))
        DPD[0][j-1][0][j]=2*(Dpd[0][j]*Dpd[0][j-1]/(Dpd[0][j]+Dpd[0][j-1]))
        DE[0][j-1][0][j]=2*(De[0][j]*De[0][j-1]/(De[0][j]+De[0][j-1]))
        DPP[0][j][0][j-1]=DPP[0][j-1][0][j]
        DPD[0][j][0][j-1]=DPD[0][j-1][0][j]
```

```
        DE[0][j][0][j-1]=DE[0][j-1][0][j]


# initial profile of dephosphorylated peptides
for j in range (0,Y):
        if (j<YL):
                xCPP=CPP
        else:
                xCPP=0.0

        for i in range(0,X):
                MPP[i][j]=xCPP



# initial profile of free enzymes
for j in range (0,Y):
        if (j>YL-5):
                xj=(j-(YL-5))*1.0
                xel=el*(1-np.exp(-xj/15))
        else:
                xel=0.0
        for i in range(0,X):
                ME[i][j]=xel




#-----------------


t=0
ti=int(t/100)
if (t>0):
```

```
        MPP=np.loadtxt('MPP_{}.txt'.format(ti))
        MPD=np.loadtxt('MPD_{}.txt'.format(ti))
        MPA=np.loadtxt('MPA_{}.txt'.format(ti))
        ME=np.loadtxt('ME_{}.txt'.format(ti))
        MEF=np.loadtxt('MEF_{}.txt'.format(ti))



mtini=0
for j in range(0,Y):
        for i in range(0,X):
                mtini=mtini+MPP[i][j]+MPD[i][j]+MPA[i][j]



xj=0
xk=0
mt=0
md=0
for j in range(0,Y):
        for i in range(0,X):
                md=md+MPD[i][j]+MPA[i][j]
q=(md/mtini)*100
pourcd=int(q)



while (q>-1):
        xj=xj+1
```

```
        t=t+dt


        for i in range(0,X):
            for j in range(0,Y):


                MET[i][j]=ME[i][j]+MEF[i][j]
                tMPP[i][j]=0.0
                tMPD[i][j]=0.0
                tMPA[i][j]=0.0
                tME[i][j]=0.0
                tMEF[i][j]=0.0




#nucleation
        for i in range(0,X):
            for j in range(0,Y):
                x=MPD[i][j]
                xx=rd.random()
                z=np.exp((x-P0N[i][j])/P1)*dt
                if (xx<z):
                    z=dt
                else:
                    z=0.0
                tMPA[i][j]=ka*MPD[i][j]*z
                tMPD[i][j]=-tMPA[i][j]
# end of nucleation


        for i in range(0,X):
            for j in range(0,Y):
                MI1[i][j]=MET[i][j]*MPP[i][j]*KE[i][j]
```

```
                tMPP[i][j]=-MI1[i][j]
                tMPD[i][j]=tMPD[i][j]+MI1[i][j]-kas*MPD[i][j]*MPA[i][j]+kad1*MPA[i][j]
                tMPA[i][j]=tMPA[i][j]+(kas*MPD[i][j]-kad1)*MPA[i][j]
                tME[i][j]=-kea*ME[i][j]*MPA[i][j]+ked*MEF[i][j]
                tMEF[i][j]=-tME[i][j]

# Diffusion of the different species
# In the center of the system

    for i in range (1,X-1):
        for j in range (1,Y-1):
            # diffusion of phosphorylated peptides
            u=DPP[i-1][j][i][j]*(MPP[i-1][j]-MPP[i][j])+DPP[i+1][j][i][j]*(MPP[i+1][j]-MPP[i][j])
            u=u+DPP[i][j-1][i][j]*(MPP[i][j-1]-MPP[i][j])+DPP[i][j+1][i][j]*(MPP[i][j+1]-MPP[i][j])
            tMPP[i][j]=tMPP[i][j]+u
            # diffusion of dephosphorylated peptides
            v=DPD[i-1][j][i][j]*(MPD[i-1][j]-MPD[i][j])+DPD[i+1][j][i][j]*(MPD[i+1][j]-MPD[i][j])
            v=v+DPD[i][j-1][i][j]*(MPD[i][j-1]-MPD[i][j])+DPD[i][j+1][i][j]*(MPD[i][j+1]-MPD[i][j])
            tMPD[i][j]=tMPD[i][j]+v
            # diffusion of free enzymes
            w=DE[i-1][j][i][j]*(ME[i-1][j]-ME[i][j])+DE[i+1][j][i][j]*(ME[i+1][j]-ME[i][j])
            w=w+DE[i][j-1][i][j]*(ME[i][j-1]-ME[i][j])+DE[i][j+1][i][j]*(ME[i][j+1]-ME[i][j])
            tME[i][j]=tME[i][j]+w

# On the 4 sides
    for j in range (1,Y-1):
        u=DPP[X-1][j][0][j]*(MPP[X-1][j]-MPP[0][j])+DPP[1][j][0][j]*(MPP[1][j]-MPP[0][j])
        u=u+DPP[0][j-1][0][j]*(MPP[0][j-1]-MPP[0][j])+DPP[0][j+1][0][j]*(MPP[0][j+1]-MPP[0][j])
```

```
        tMPP[0][j]=tMPP[0][j]+u
        v=DPD[X-1][j][0][j]*(MPD[X-1][j]-MPD[0][j])+DPD[1][j][0][j]*(MPD[1][j]-MPD[0][j])
        v=v+DPD[0][j-1][0][j]*(MPD[0][j-1]-MPD[0][j])+DPD[0][j+1][0][j]*(MPD[0][j+1]-MPD[0][j])
        tMPD[0][j]=tMPD[0][j]+v
        w=DE[X-1][j][0][j]*(ME[X-1][j]-ME[0][j])+DE[1][j][0][j]*(ME[1][j]-ME[0][j])
        w=w+DE[0][j-1][0][j]*(ME[0][j-1]-ME[0][j])+DE[0][j+1][0][j]*(ME[0][j+1]-ME[0][j])
        tME[0][j]=tME[0][j]+w

for j in range (1,Y-1):
        u=DPP[X-1][j][0][j]*(MPP[0][j]-MPP[X-1][j])+DPP[X-1][j][X-2][j]*(MPP[X-2][j]-MPP[X-1][j])
        u=u+DPP[X-1][j-1][X-1][j]*(MPP[X-1][j-1]-MPP[X-1][j])+DPP[X-1][j+1][X-1][j]*(MPP[X-1][j+1]-MPP[X-1][j])
        tMPP[X-1][j]=tMPP[X-1][j]+u
        v=DPD[X-1][j][0][j]*(MPD[0][j]-MPD[X-1][j])+DPD[X-1][j][X-2][j]*(MPD[X-2][j]-MPD[X-1][j])
        v=v+DPD[X-1][j-1][X-1][j]*(MPD[X-1][j-1]-MPD[X-1][j])+DPD[X-1][j+1][X-1][j]*(MPD[X-1][j+1]-MPD[X-1][j])
        tMPD[X-1][j]=tMPD[X-1][j]+v
        w=DE[X-1][j][0][j]*(ME[0][j]-ME[X-1][j])+DE[X-1][j][X-2][j]*(ME[X-2][j]-ME[X-1][j])
        w=w+DE[X-1][j-1][X-1][j]*(ME[X-1][j-1]-ME[X-1][j])+DE[X-1][j+1][X-1][j]*(ME[X-1][j+1]-ME[X-1][j])
        tME[X-1][j]=tME[X-1][j]+w

for i in range (1,X-1):
        u=DPP[i-1][0][i][0]*(MPP[i-1][0]-MPP[i][0])+DPP[i+1][0][i][0]*(MPP[i+1][0]-MPP[i][0])
        u=u+DPP[i][1][i][0]*(MPP[i][1]-MPP[i][0])
        tMPP[i][0]=tMPP[i][0]+u
        v=DPD[i-1][0][i][0]*(MPD[i-1][0]-MPD[i][0])+DPD[i+1][0][i][0]*(MPD[i+1][0]-MPD[i][0])
        v=v+DPD[i][1][i][0]*(MPD[i][1]-MPD[i][0])
        tMPD[i][0]=tMPD[i][0]+v
        w=DE[i-1][0][i][0]*(ME[i-1][0]-ME[i][0])+DE[i+1][0][i][0]*(ME[i+1][0]-ME[i][0])
        w=w+DE[i][1][i][0]*(ME[i][1]-ME[i][0])
        tME[i][0]=tME[i][0]+w

for i in range (1,X-1):
```

```
        u=DPP[i-1][Y-1][i][Y-1]*(MPP[i-1][Y-1]-MPP[i][Y-1])+DPP[i+1][Y-1][i][Y-1]*(MPP[i+1][Y-1]-MPP[i][Y-1])
        u=u+DPP[i][Y-1][i][Y-2]*(MPP[i][Y-2]-MPP[i][Y-1])
        tMPP[i][Y-1]=tMPP[i][Y-1]+u
        v=DPD[i-1][Y-1][i][Y-1]*(MPD[i-1][Y-1]-MPD[i][Y-1])+DPD[i+1][Y-1][i][Y-1]*(MPD[i+1][Y-1]-MPD[i][Y-1])
        v=v+DPD[i][Y-1][i][Y-2]*(MPD[i][Y-2]-MPD[i][Y-1])
        tMPD[i][Y-1]=tMPD[i][Y-1]+v
        w=DE[i-1][Y-1][i][Y-1]*(ME[i-1][Y-1]-ME[i][Y-1])+DE[i+1][Y-1][i][Y-1]*(ME[i+1][Y-1]-ME[i][Y-1])
        w=w+DE[i][Y-1][i][Y-2]*(ME[i][Y-2]-ME[i][Y-1])
        tME[i][Y-1]=tME[i][Y-1]+w


# On the 4 corners
    u=DPP[X-1][0][0][0]*(MPP[X-1][0]-MPP[0][0])+DPP[1][0][0][0]*(MPP[1][0]-MPP[0][0])
    u=u+DPP[0][1][0][0]*(MPP[0][1]-MPP[0][0])
    tMPP[0][0]=tMPP[0][0]+u
    v=DPD[X-1][0][0][0]*(MPD[X-1][0]-MPD[0][0])+DPD[1][0][0][0]*(MPD[1][0]-MPD[0][0])
    v=v+DPD[0][1][0][0]*(MPD[0][1]-MPD[0][0])
    tMPD[0][0]=tMPD[0][0]+v
    w=DE[X-1][0][0][0]*(ME[X-1][0]-ME[0][0])+DE[1][0][0][0]*(ME[1][0]-ME[0][0])
    w=w+DE[0][1][0][0]*(ME[0][1]-ME[0][0])
    tME[0][0]=tME[0][0]+w


    u=DPP[X-1][0][0][0]*(MPP[0][0]-MPP[X-1][0])+DPP[X-2][0][X-1][0]*(MPP[X-2][0]-MPP[X-1][0])
    u=u+DPP[X-1][1][X-1][0]*(MPP[X-1][1]-MPP[X-1][0])
    tMPP[X-1][0]=tMPP[X-1][0]+u
    v=DPD[X-1][0][0][0]*(MPD[0][0]-MPD[X-1][0])+DPD[X-2][0][X-1][0]*(MPD[X-2][0]-MPD[X-1][0])
    v=v+DPD[X-1][1][X-1][0]*(MPD[X-1][1]-MPD[X-1][0])
    tMPD[X-1][0]=tMPD[X-1][0]+v
    w=DE[X-1][0][0][0]*(ME[0][0]-ME[X-1][0])+DE[X-2][0][X-1][0]*(ME[X-2][0]-ME[X-1][0])
    w=w+DE[X-1][1][X-1][0]*(ME[X-1][1]-ME[X-1][0])
```

```
tME[X-1][0]=tME[X-1][0]+w


u=DPP[X-1][Y-1][0][Y-1]*(MPP[X-1][Y-1]-MPP[0][Y-1])+DPP[1][Y-1][0][Y-1]*(MPP[1][Y-1]-MPP[0][Y-1])
u=u+DPP[0][Y-2][0][Y-1]*(MPP[0][Y-2]-MPP[0][Y-1])
tMPP[0][Y-1]=tMPP[0][Y-1]+u
v=DPD[X-1][Y-1][0][Y-1]*(MPD[X-1][Y-1]-MPD[0][Y-1])+DPD[1][Y-1][0][Y-1]*(MPD[1][Y-1]-MPD[0][Y-1])
v=v+DPD[0][Y-2][0][Y-1]*(MPD[0][Y-2]-MPD[0][Y-1])
tMPD[0][Y-1]=tMPD[0][Y-1]+v
w=DE[X-1][Y-1][0][Y-1]*(ME[X-1][Y-1]-ME[0][Y-1])+DE[1][Y-1][0][Y-1]*(ME[1][Y-1]-ME[0][Y-1])
w=w+DE[0][Y-2][0][Y-1]*(ME[0][Y-2]-ME[0][Y-1])
tME[0][Y-1]=tME[0][Y-1]+w


u=DPP[X-1][Y-1][0][Y-1]*(MPP[0][Y-1]-MPP[X-1][Y-1])+DPP[X-1][Y-1][X-2][Y-1]*(MPP[X-2][Y-1]-MPP[X-1][Y-1])
u=u+DPP[X-1][Y-2][X-1][Y-1]*(MPP[X-1][Y-2]-MPP[X-1][Y-1])
tMPP[X-1][Y-1]=tMPP[X-1][Y-1]+u
v=DPD[X-1][Y-1][0][Y-1]*(MPD[0][Y-1]-MPD[X-1][Y-1])+DPD[X-1][Y-1][X-2][Y-1]*(MPD[X-2][Y-1]-MPD[X-1][Y-1])
v=v+DPD[X-1][Y-2][X-1][Y-1]*(MPD[X-1][Y-2]-MPD[X-1][Y-1])
tMPD[X-1][Y-1]=tMPD[X-1][Y-1]+v
w=DE[X-1][Y-1][0][Y-1]*(ME[0][Y-1]-ME[X-1][Y-1])+DE[X-1][Y-1][X-2][Y-1]*(ME[X-2][Y-1]-ME[X-1][Y-1])
w=w+DE[X-1][Y-2][X-1][Y-1]*(ME[X-1][Y-2]-ME[X-1][Y-1])
tME[X-1][Y-1]=tME[X-1][Y-1]+w



for i in range(0,X):
    for j in range(0,Y):
        MPP[i][j]=MPP[i][j]+tMPP[i][j]
        MPD[i][j]=MPD[i][j]+tMPD[i][j]
        MPA[i][j]=MPA[i][j]+tMPA[i][j]
        ME[i][j]=ME[i][j]+tME[i][j]
        MEF[i][j]=MEF[i][j]+tMEF[i][j]
```

```
# plot every 100 iterations
        if (xj==100):

                md=0.0
                mt=0.0
                for j in range(0,Y):
                        for i in range(0,X):

                                md=md+MPD[i][j]+MPA[i][j]
                                mt=mt+MPD[i][j]+MPA[i][j]+MPP[i][j]

                q=(md/mtini)*100


                te=t/100
                if (te>ti):
                        tq=int(te)
                        np.savetxt('MPP_{}.txt'.format(tq),MPP)
                        np.savetxt('MPD_{}.txt'.format(tq),MPD)
                        np.savetxt('MPA_{}.txt'.format(tq),MPA)
                        np.savetxt('ME_{}.txt'.format(tq),ME)
                        np.savetxt('MEF_{}.txt'.format(tq),MEF)


                        ti=ti+1
```

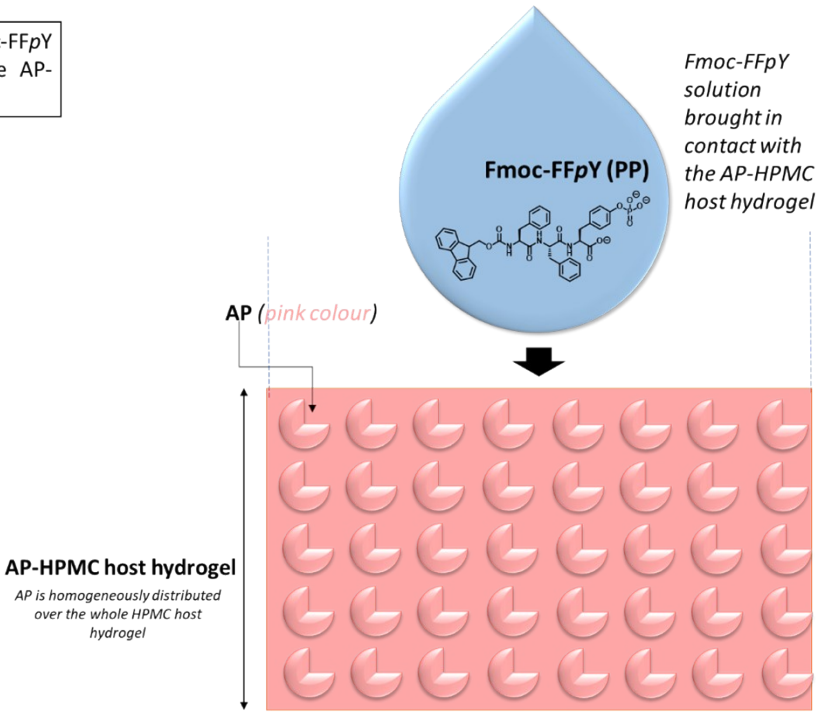
```
for j in range(0,Y):
        iy2[j]=MPA[2][j]
        iy3[j]=MPD[2][j]
        iy4[j]=MPP[2][j]
        iy5[j]=ME[2][j]*7000
        ix[j]=j




plt.ylim(0,5.0)
plt.xlim(-1,Y)
plt.plot(ix,iy2)
plt.plot(ix,iy3)
plt.plot(ix,iy4)
plt.plot(ix,iy5)
plt.pause(0.01)
plt.clf()
xj=0
```

# FIGURES

**FIGURE S1.** Schematics of the Fmoc-FFY self-assembled patterns mechanism formation when Fmoc-FF*p*Y is diffusing through the AP-HPMC host hydrogel.

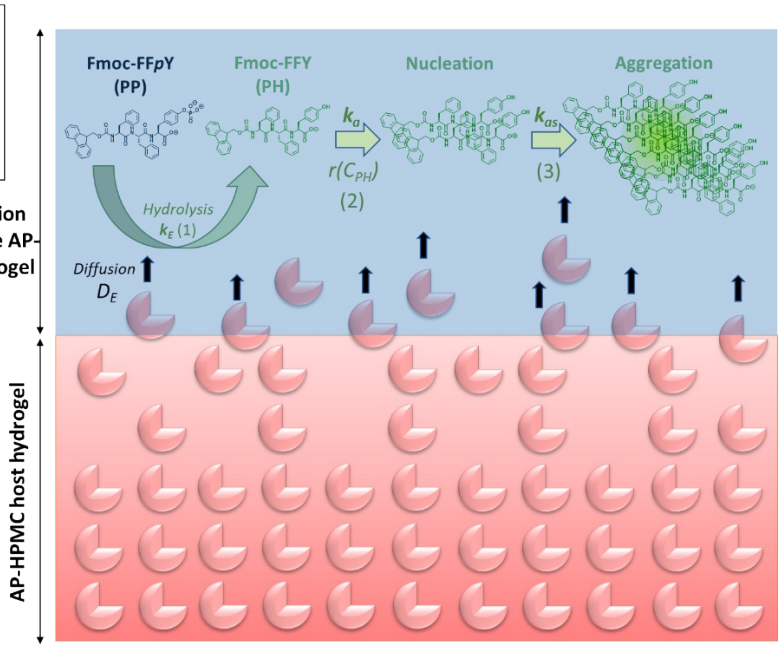
**Steps a-b.** Deposition of the Fmoc-FF*p*Y precursor (**black color**) solution on the AP-HPMC host hydrogel (*pink*).

PP : Precursor peptide (*i.e.* Fmoc-FF*p*Y)




**Steps a-b** (*next*). Immediately after the deposition of the Fmoc-FF*p*Y solution on the AP-HPMC host hydrogel, some AP are diffusing from the gel to the solution, transforming Fmoc-FF*p*Y in the self-assembling bluiding block Fmoc-FFY (*in green*).

$D_E$ : enzyme diffusion constant
$k_E$: enzymatic activity constant
$k_a$: nucleation constant
$r(C_{PH})$ : stochastic function of nucleation
PP : Precursor peptide (*i.e.* Fmoc-FF*p*Y)
PH : Hydrogelator peptide (*i.e.* Fmoc-FFY)
$k_{as}$ : Aggregation constant




(1) $$-\frac{dc_{PP}}{dt} = \frac{dc_{PP}}{dt} = k_E(c_E + c_{EF}).c_{PP}$$

(2) $$\frac{dc_{PA}}{dt} = -\frac{dc_{PH}}{dt} = k_a \,.\, c_{PH}.r(c_{PH})$$

(3) $$\frac{dc_{PA}}{dt} = k_{as}.c_{PH}.c_{PA}$$

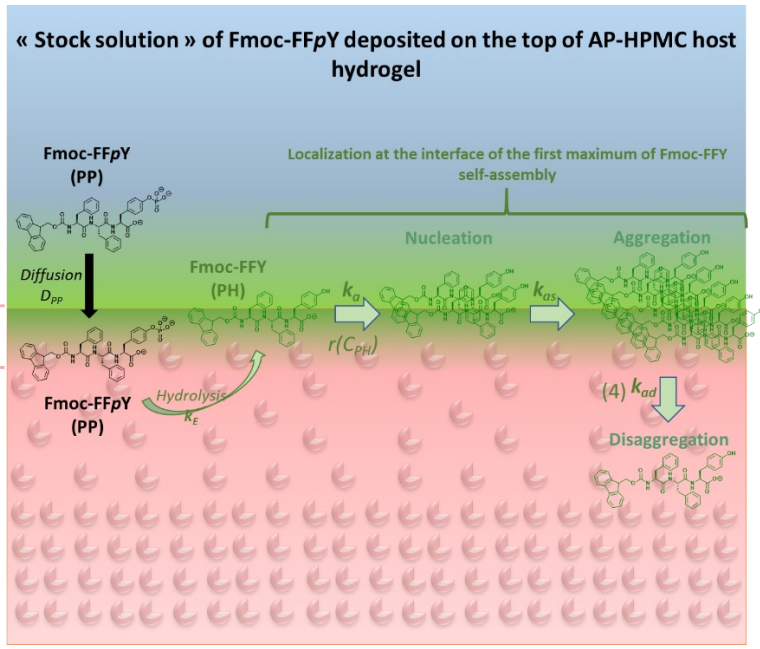
**Steps c and c'.** Concomitantly to the previous step 2, the Fmoc-FF*p*Y solution is diffusing in the AP-HPMC host hydrogel, and near the interface, the precursor is transformed in Fmoc-FFY leading to the first maximum of self-assembly. For sake of clarity, all enzyme representations are not shown.

$D_{PP}$ : Diffusion coefficient of the precursor peptide (*i.e.* Fmoc-FF*p*Y)
$k_{ad}$: Desaggregation constant

A high density of AP is colocalized in this Fmoc-FFY self-assembly area since AP adsorbed irreversibly on it.

(4) $-\frac{dc_{PA}}{dt} = \frac{dc_{PH}}{dt} = k_{ad} . c_{PA}$

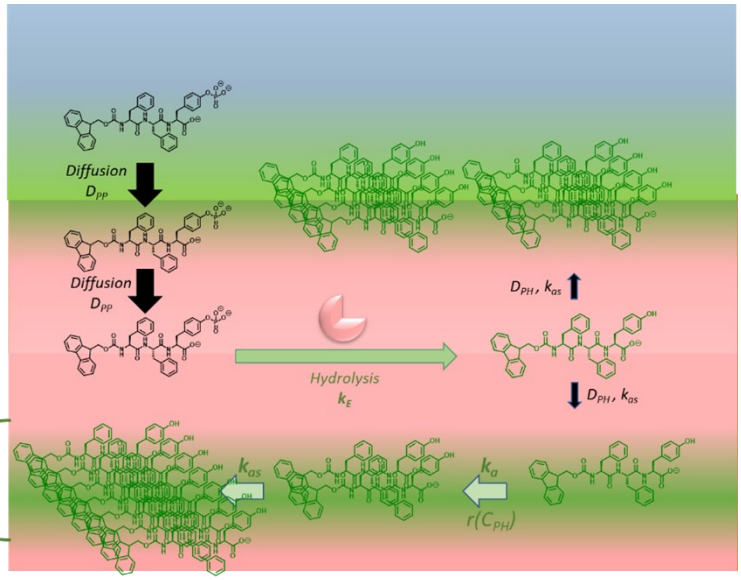
*Note: for sake of clarity, the disaggregation process ($k_{ad}$) is only shown on this schematic.*




**Steps d - f.** The Fmoc-FF*p*Y solution is diffusing deeper in the AP-HPMC host hydrogel, and Fmoc-FF*p*Y is continuously hydrolyzed by AP resulting in Fmoc-FFY which diffuses in all directions. When diffusing to the interface, Fmoc-FFY self-assembles on the first self-assembly area located there. When diffusing to the opposite direction (to the less concentrated region), Fmoc-FFY is accumulating further away from the interface. Thanks to high AP concentration present within AP-HPMC host hydrogel, Fmoc-FFY is largely produced reaching a suitable concentration to generate a 2nd continuous maximum of peptide self-assembly within the host hydrogel.

$D_{PH}$ : Diffusion coefficient of the hydrogelator peptide (*i.e.* Fmoc-FFY)

2nd maximum of Fmoc-FFY self-assembly localized within the host AP-HPMC hydrogel



**Steps d' – f'.** The Fmoc-FF*p*Y solution is diffusing deep in the AP-HPMC host hydrogel. Because of the low density (concentration) of AP distributed within HPMC, few Fmoc-FFY peptides are generated *in situ*. When diffusing to the interface, Fmoc-FFY self-assembles on the first self-assembly area located there. When these peptides are diffusing to the opposite direction (toward the less concentrated region), Fmoc-FFY is accumulating but without reaching the suitable concentration to self-assemble. However, spatial overconcentrations of Fmoc-FFY appear in a stochastic way, and the stochasticity of the nucleation process results in localized microglobular self-assembled Fmoc-FFY distributed in the host hydrogel. Then, these microglobules grow, creating thus a depletion zone around them.

**Formation of Fmoc-FFY self-assembled microglobules localized within the host AP-HPMC hydrogel**

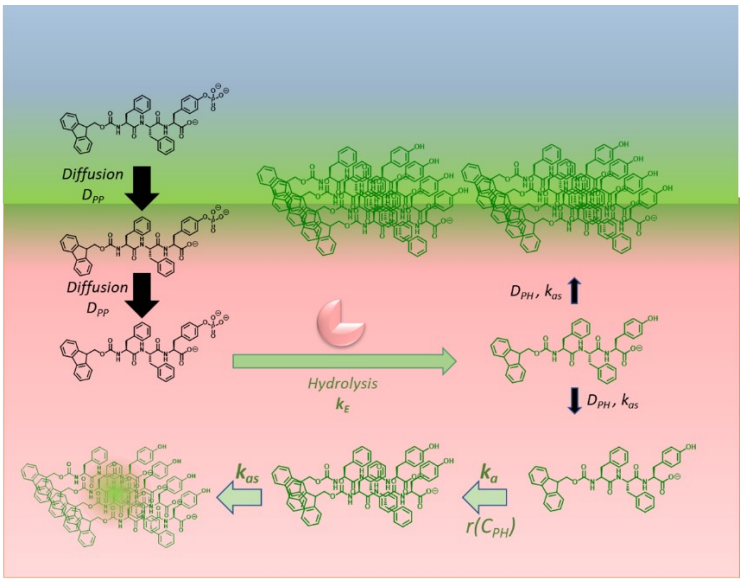


**FIGURE S2.** Determination of the *CC* of Fmoc-FFY and Fmoc-FFFY self-assembly through the fluorescence emission intensity (a.u.) measured at various concentration of Fmoc-FF*p*Y or Fmoc-FFF*p*Y in presence of AP and ThT. Details are given in the Material and Methods section. The mean value of fluorescence emission intensity for each precursor peptide (Fmoc-FF*p*Y or Fmoc-FFF*p*Y) concentration was calculated from at least three independent measurements, allowing us to include error bars in the graph below.

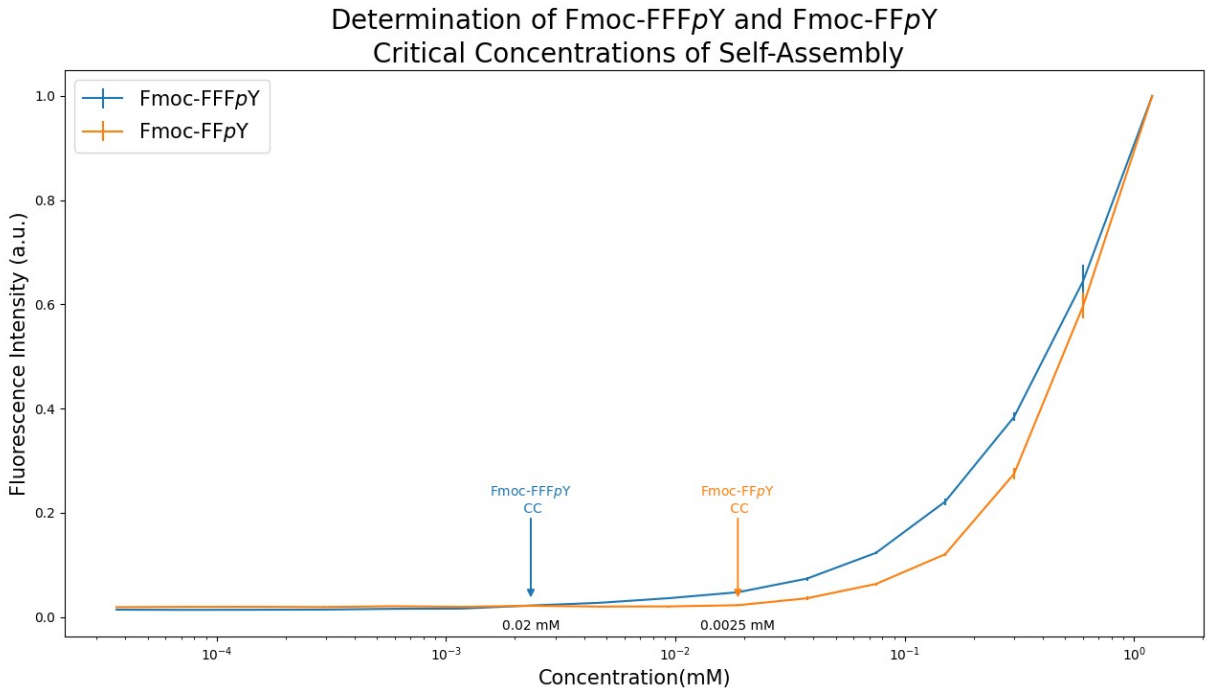


**FIGURE S3.** Visualization of the AP diffusion profile determined from AP-HPMC host hydrogel containing alkaline phosphatase labeled with rhodamine ($AP^{RHO}$) brought in contact with a borax buffer solution. The fluorescence emission intensity of the rhodamine dye was measured by CLSM. From these data we extracted the evolution of the fluorescence profile as a function of time

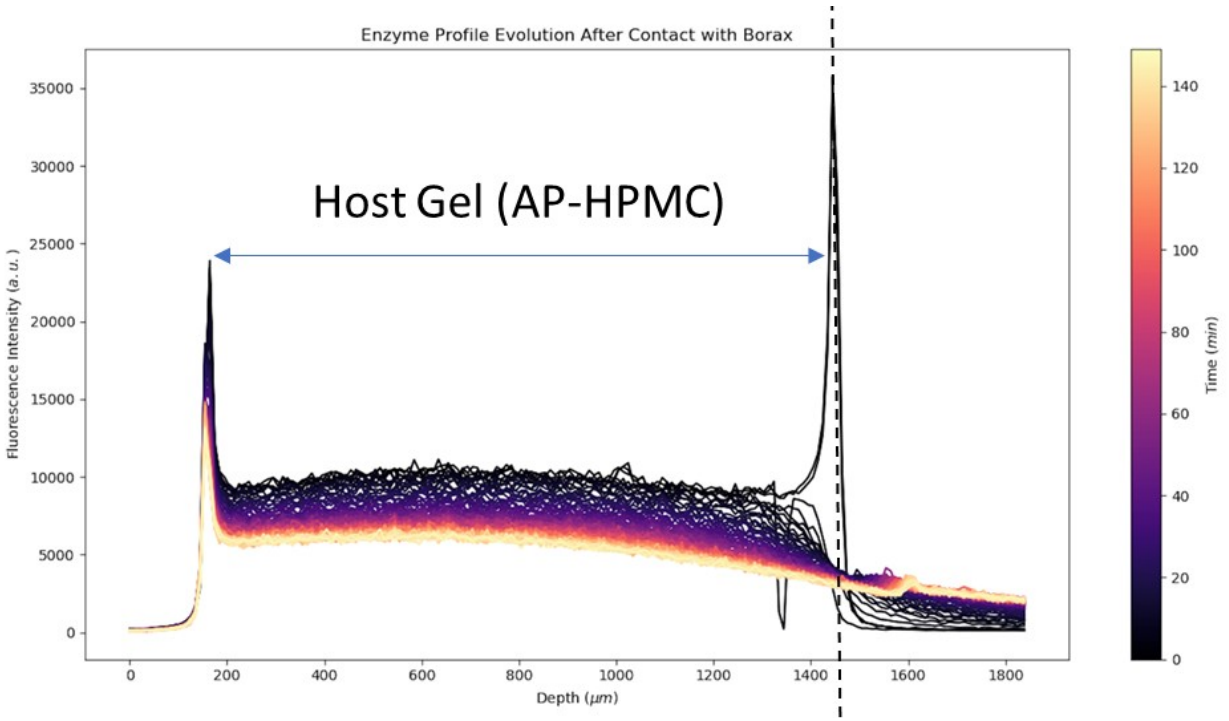
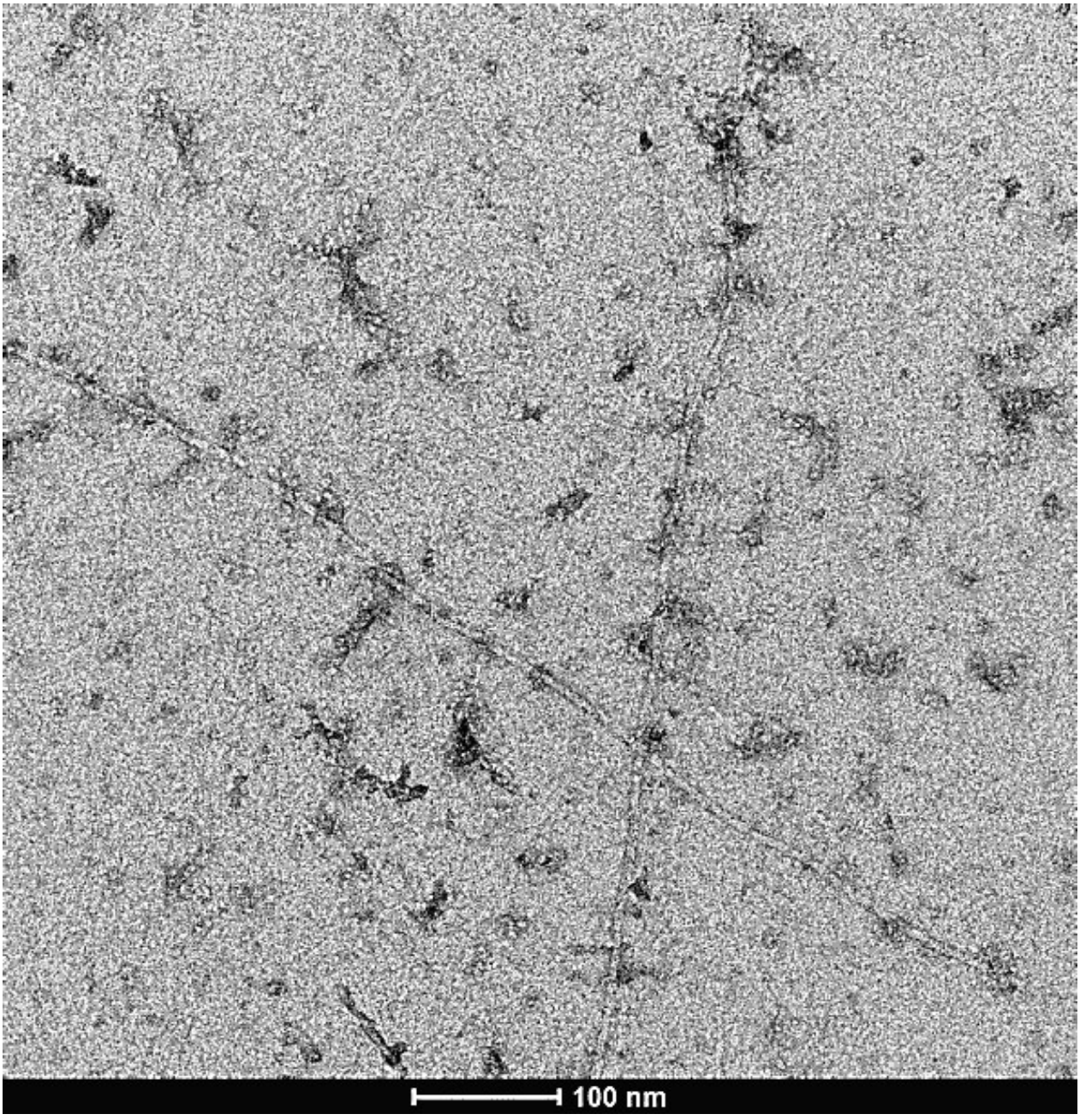


**FIGURE S4.** Magnification of the typical TEM images given in the manuscript (Fig. 2a) and corresponding to the (**A**) self-assembled Fmoc-FFY nanofibers and to (**B**) the self-assembled Fmoc-FFFY nanofibers.

**A.**



**B.**

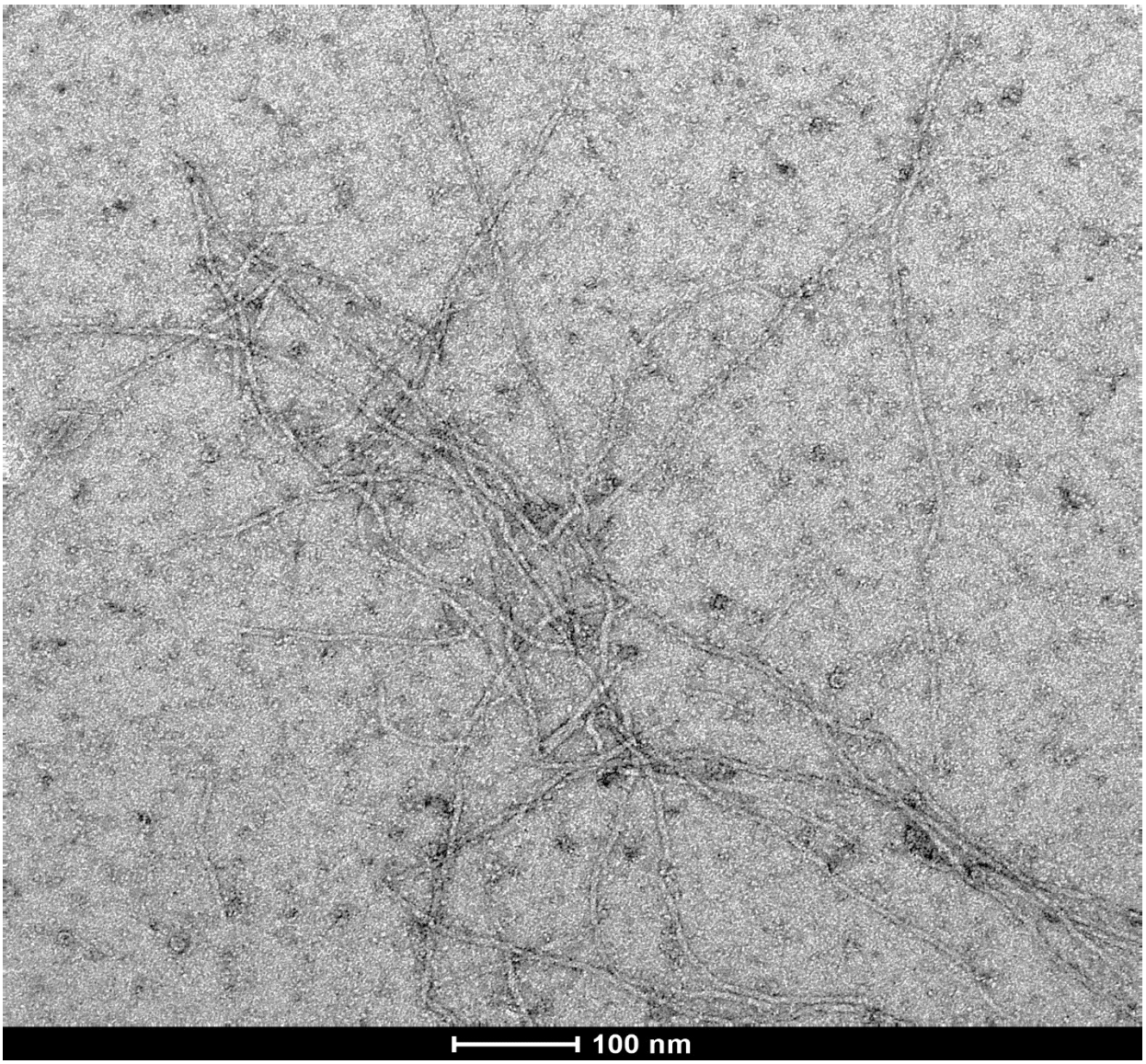


**Figure S5**. (*left*) Experimental CLSM monitoring of the Fmoc-FFY self-assembled pattern revealed by ThT (green emission) within the AP-HPMC gel and its corresponding measured cross-section profiles of the fluorescence emission intensity of ThT (*middle*). The precursor peptide Fmoc-FF*p*Y concentration is kept at 1.2 mg/mL and the AP concentration in the AP-HPMC gel is 0.001 mg/mL. The modeled self-assembled pattern (called "Model") based on our described model is given on the *right* side.

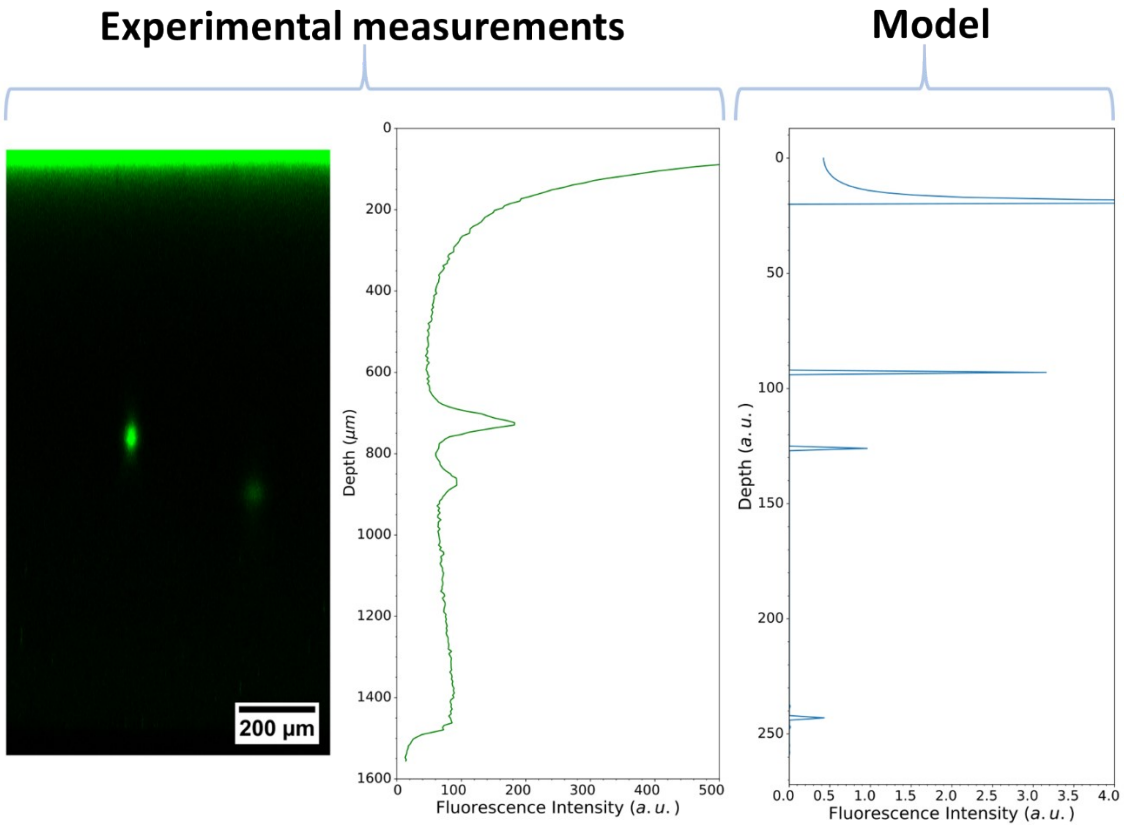


**FIGURE S6.** Search of the minimum of parameters needed to account qualitatively for the experimental trends.

**FIGURE S6A. Effect of enzyme adsorption constant $k_{ea}$ and the "desorption" constant $k_{ad}$ equal to zero.**

The values of the parameters were:

$x_0$=5, $x_1$=15, $c_{PP}$=5, $D_{PP}$=0.5, $D_{PH}$=0.4, $D_E$=0.1, CC=0.05 (in the gel), CC=0 (in the solution), $P_1=10^{-3}$, $k_a$=0.1, $k_{as}$=0.1, $k_E$=10 (in the gel), $k_E$=50 (in the solution), $\delta$t=0.01 (30-5-2024)
**blue line:** $k_{ad}=10^{-4}$, $k_{ea}$=0.1, $k_{ed}$=0.01
**orange line:** $k_{ad}$=0, $k_{ea}$=0, $k_{ed}$=0

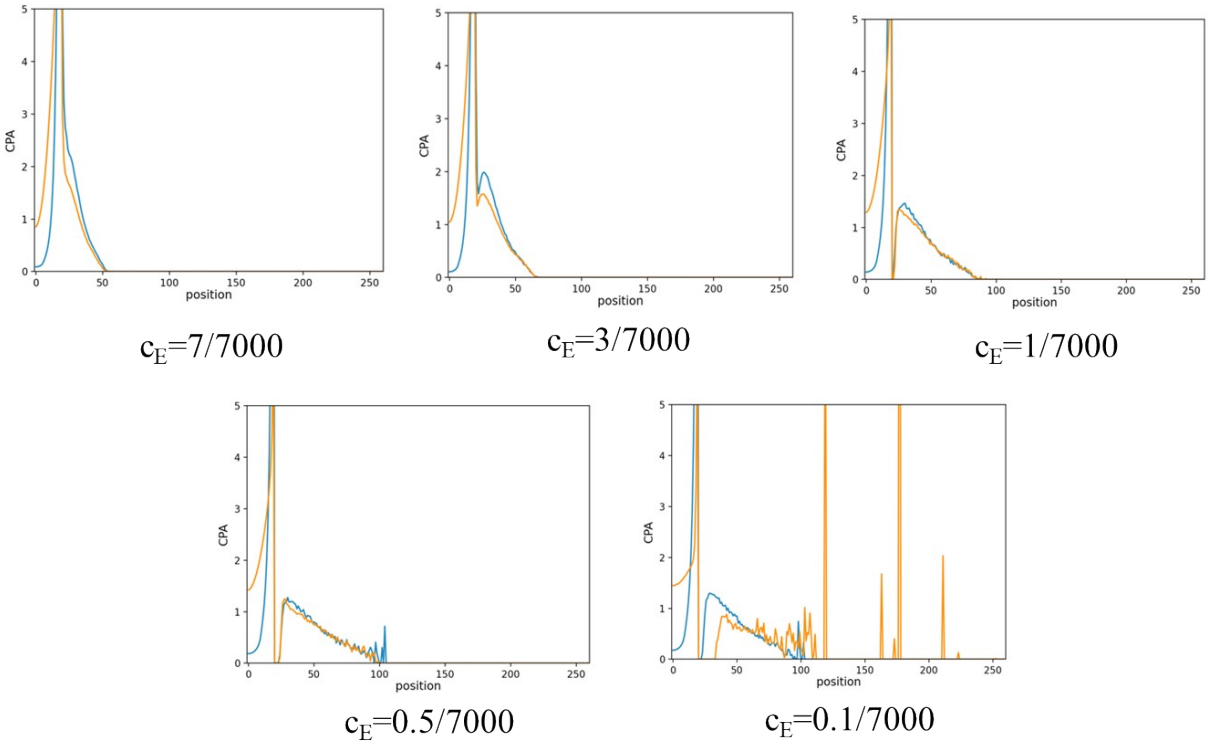
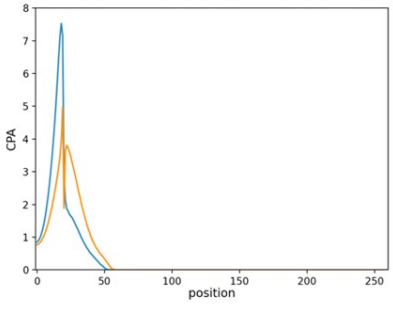
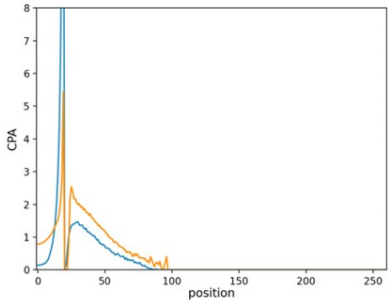


**FIGURE 6B. Effect of the higher enzymatic constant $k_E$ in the solution than in the gel.** Simulations were thus performed with the same value $k_E$ in the solution and in the gel by varying the enzyme concentration.

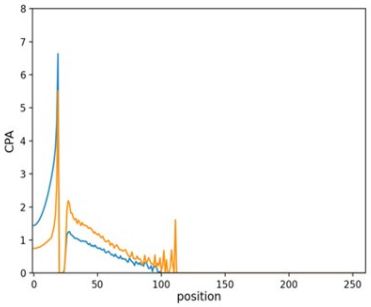
The values of the parameters were: $x_0$=5, $x_1$=15, $c_{PP}$=5, $D_{PP}$=0.5, $D_{PH}$=0.4, $D_E$=0.1, CC=0.05 (in the gel), CC=0 (in the solution), $P_1=10^{-3}$, $k_a$=0.1, $k_{as}$=0.1, $\delta$t=0.01, $k_{ad}$=0, $k_{ea}$=0, $k_{ed}$=0
**Blue line** : $k_E$=10 (in the gel), $k_E$=50 (in the solution)
**Orange line** : $k_E$=10 (in the gel and in the solution)



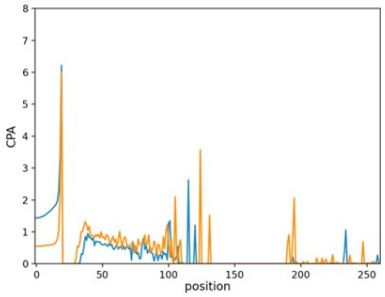

$c_E$=7/7000



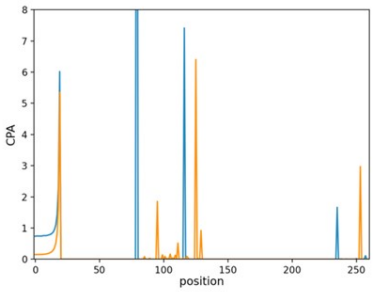

$c_E$=1/7000




$c_E$=0.5/7000




$c_E$=0.1/7000




$c_E$=0.01/7000

**FIGURE S6C. Effect of the critical aggregation concentration *CC* in the solution by keeping *CC* fixed in the gel.** The simulations were run with the same enzymatic activity in the solution and in the gel.

The values of the parameters were:

$x_0$=5, $x_1$=15, $c_{PP}$=5, $D_{PP}$=0.5, $D_{PH}$=0.4, $D_E$=0.1, CC=0.05 (in the gel, $CC_g$), $P_1=10^{-3}$, $k_a$=0.1, $k_{as}$=0.1, $\delta t$=0.01, $k_{ad}$=0, $k_{ea}$=0, $k_{ed}$=0, $k_E$=10 (in the solution and in the gel), $c_E$=0.5/7000

Values of CC in the solution: **blue**: 0; orange: (4/5).$CC_g$ ; **green**: (3/5). $CC_g$; **red**: (2/5). $CC_g$;

**violet**: (1/5).$CC_g$; **brown**: (4.5/5).$CC_g$; **pink**: (5/5).$CC_g$

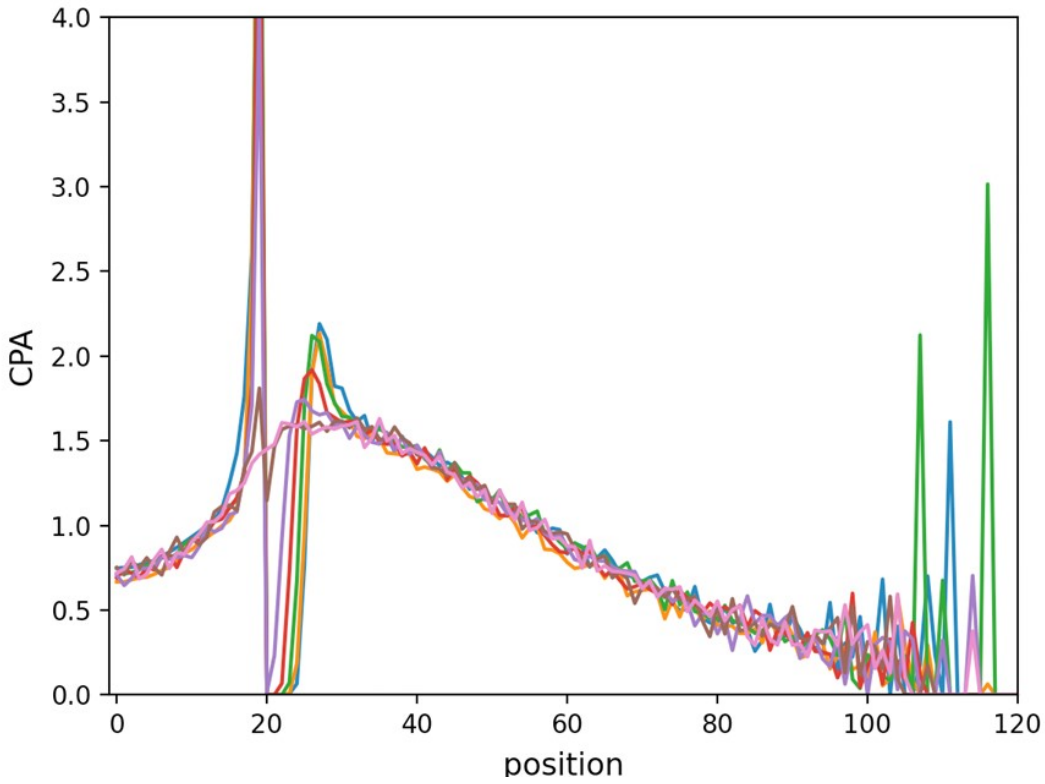


**FIGURE S6D. Effect of a higher enzymatic activity in the solution than in the gel, the critical aggregation concentration *CC* being the same in the solution and in the gel.**

We have investigated the effect of a higher enzymatic activity in the solution than in the gel, the critical aggregation concentration *CC* being the same in the solution and in the gel. The goal was to investigate if a higher enzymatic activity in the solution than in the gel has the same effect as a smaller value of *CC* in the solution compared to its value in the gel. The conclusion is that this is not the case.

We used the following parameters to perform the simulations:

$x_0$=5, $x_1$=25, $c_{PP}$=5, $D_{PP}$=0.5, $D_{PH}$=0.4, $D_E$=0.1, $k_a$=0.1, $k_{as}$=0.1, $\delta$t=0.01, $k_{ad}$=0, $k_{ea}$=0, $k_{ed}$=0, *CC*=0.05 (in the solution and in the gel), $P_1 = 0.01$, el=0.1/7000, $k_E = 10$ **(in the gel)**

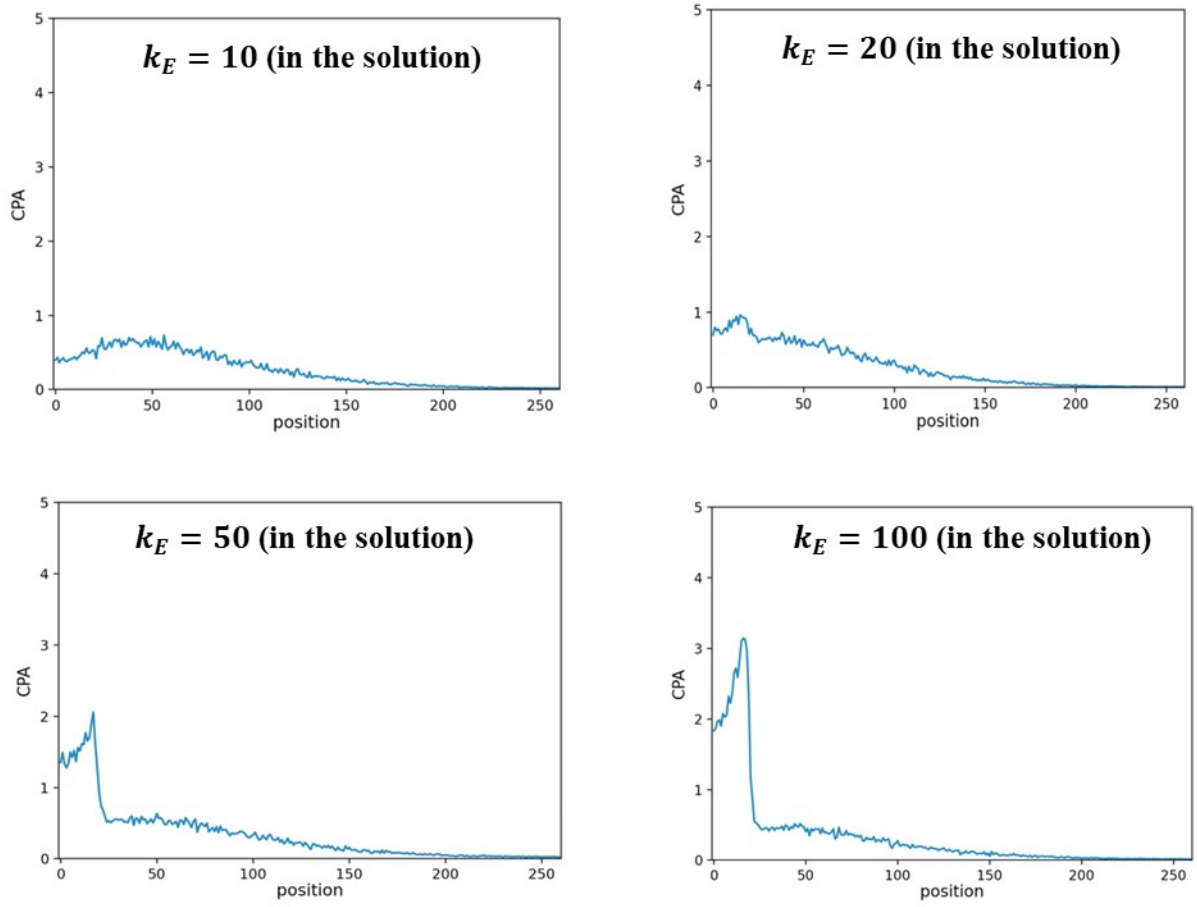
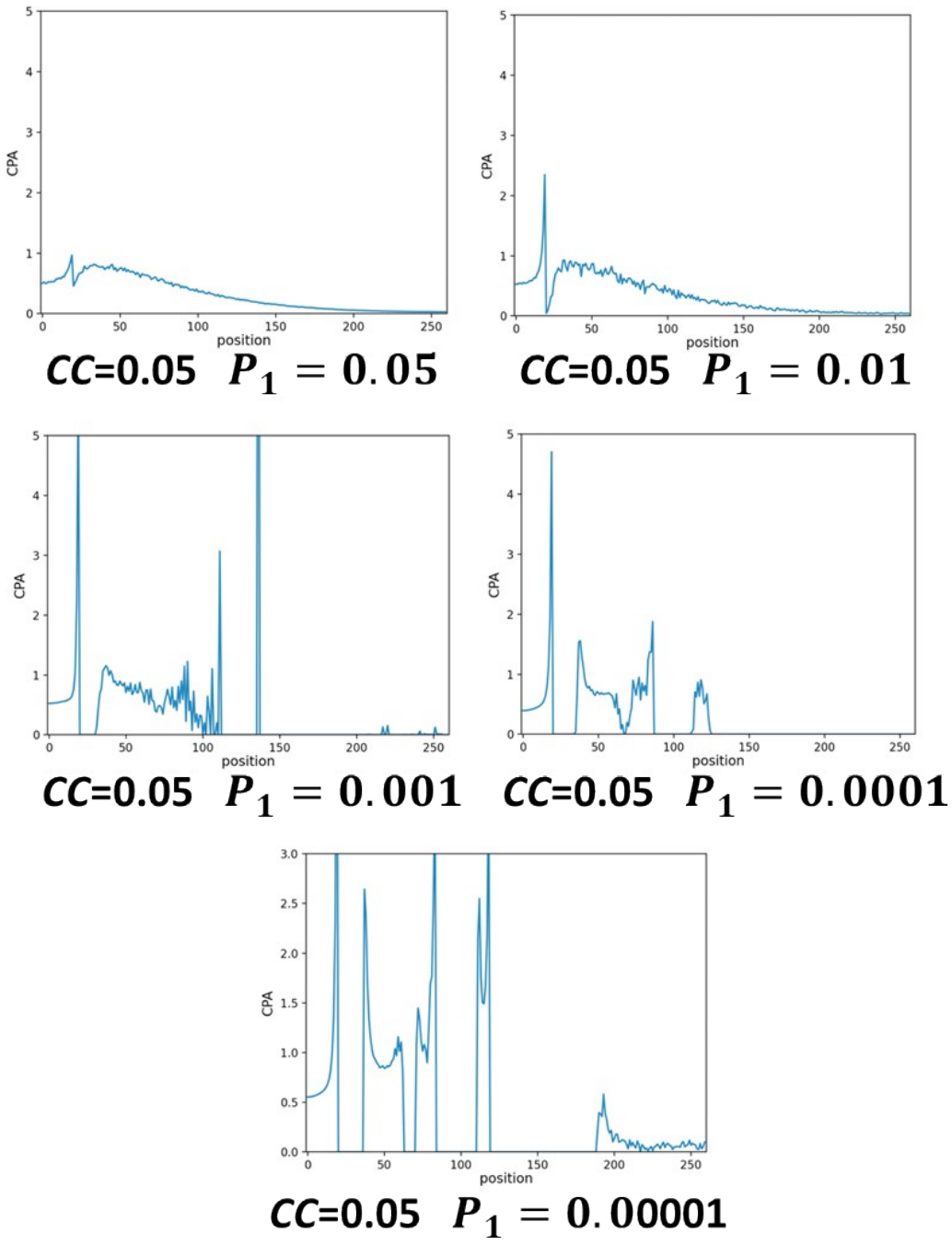


**FIGURE S6E. Effect of $P_1$ on the enzyme profile at the end of the self-assembly process**. The simulations were performed by using the same enzymatic activity in the solution and in the gel, a critical aggregation concentration $CC = 0.05$ in the gel but 0 in the solution. The values of the other parameters were:
$x_0$=5, $x_1$=15, $c_{PP}$=5, $D_{PP}$=0.5, $D_{PH}$=0.4, $D_E$=0.1, $k_a$=0.1, $k_{as}$=0.1, $\delta$t=0.01, $k_{ad}$=0, $k_{ea}$=0, $k_{ed}$=0, $k_E$=10 (in the solution and in the gel), $c_E$=0.1/7000.

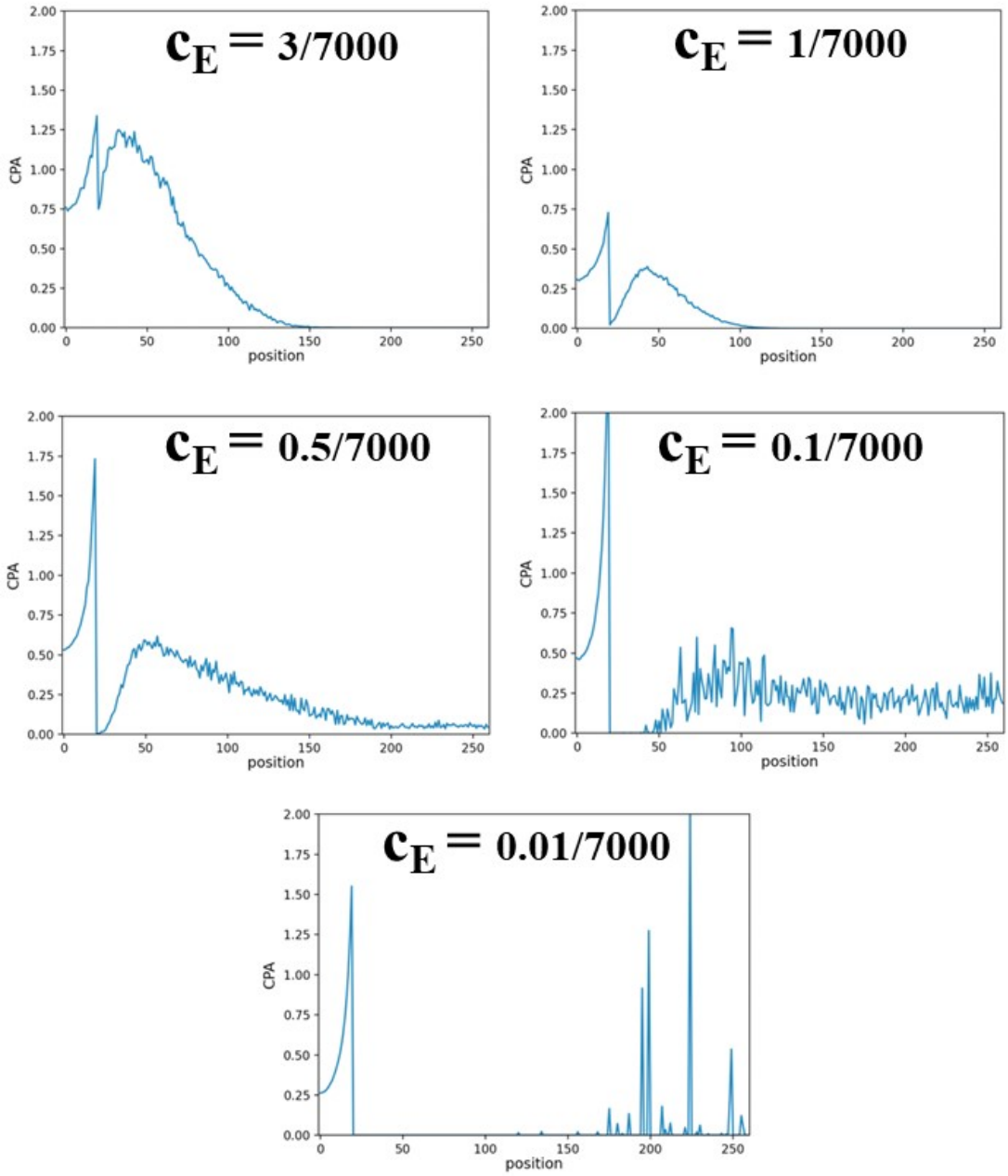


## FIGURE S6F. Effect of the random function $r(c_{PH})$, n = 8

We have investigated the effect of expression of the random function $r(c_{PH})$ on the evolution of the self-assembly profile when varying the enzyme concentration. We have chosen the function $r(c_{PH}) = \frac{\delta t}{\tau}\left(\frac{c_{PH}}{CC}\right)^n$ with $\tau = {}^1\!/\!_{k_a}$. The critical aggregation concentration $CC$ was set equal to 0.1* $CC$ in the solution where $CC$ represents the critical aggregation concentration in the gel. Simulations were first performed with n=8. Two series of simulations were run, one with the same enzyme concentration in the gel and in the solution and one where the enzyme activity was 5 times larger in the solution than in the gel. All other parameters are as follows:

$x_0$=5, $x_1$=15, $c_{PP}$=5, $D_{PP}$=0.5, $D_{PH}$=0.4, $D_E$=0.1, $k_a$=0.1, $k_{as}$=0.01, $\delta$t=0.01, $k_{ad}$=0, $k_{ea}$=0, $k_{ed}$=0, $CC = 0.1$, n=8

Simulations with the same enzymatic activity in the solution and in the gel : $k_E = 1$, n=8



Simulations with an enzymatic activity 5 times higher in the solution and in the gel : $k_E = 1$ in the gel, n=8

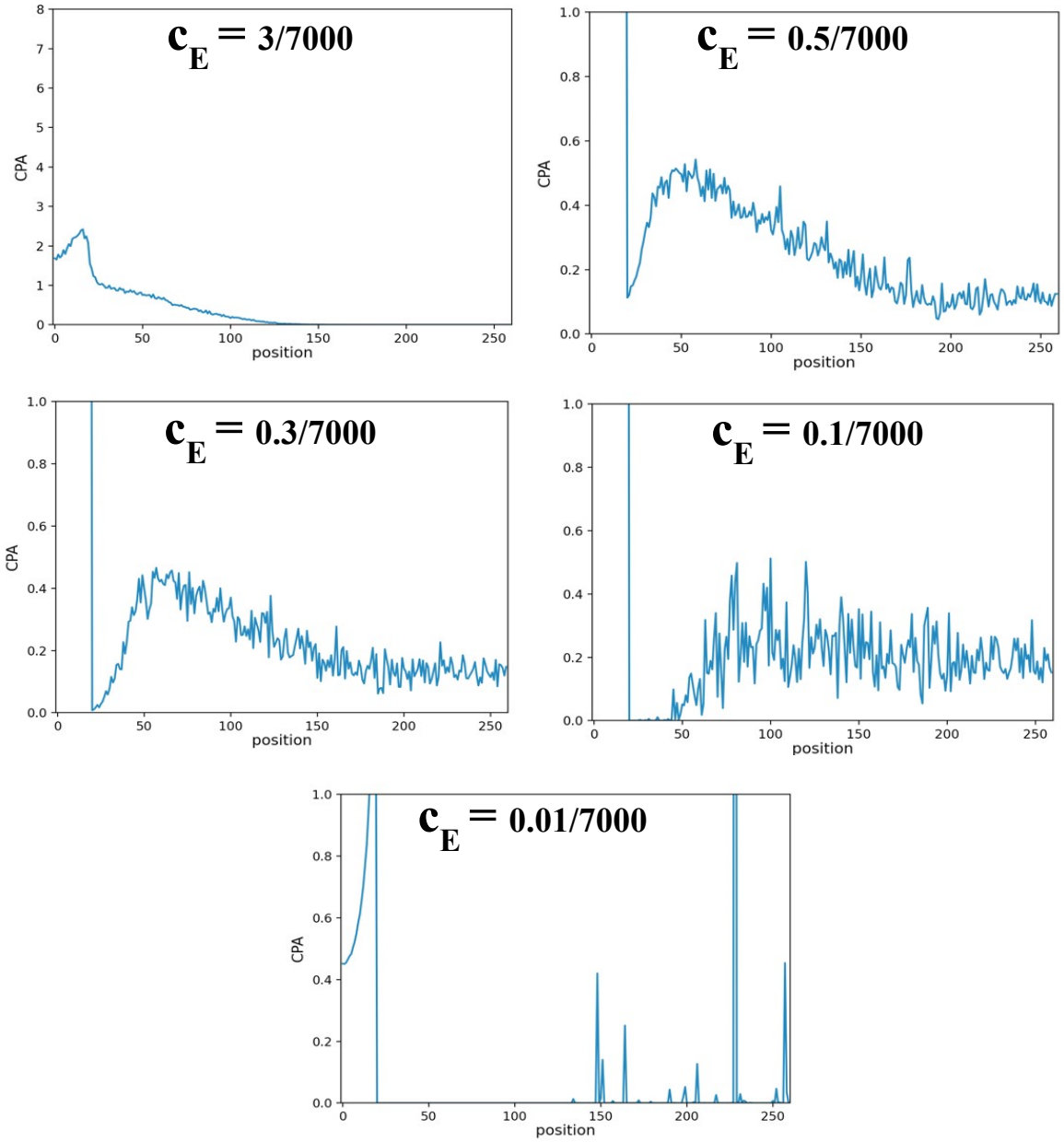


**FIGURE S6G. Effect of the random function $r(c_{PH})$, n = 4**

**Simulations performed with n=4**. The enzyme activity was 5 times larger in the solution than in the gel. The other parameters were : $x_0$=5, $x_1$=15, $c_{PP}$=5, $D_{PP}$=0.5, $D_{PH}$=0.4, $D_E$=0.1, $k_a$=0.1, $k_{as}$=0.05, $\delta$t=0.01, $k_{ad}$=0, $k_{ea}$=0, $k_{ed}$=0, $CC = 0.1$, $k_E = 1$ (in the gel)