



PSE : Méthodes et protocoles

Détection de polluants grâce à la  
chimiotaxie bactérienne

*J. Lazenec, A. Dufour, C. Doucet*

# Sommaire

## Matériel général

### I - Choix des bactéries

### II - Préparation des milieux

- 1) Milieu LB
- 2) Milieu M9-Glc-Cm30
- 3) Chemotaxis buffer

### III - Préparation des bactéries

- 1) Préparation des cultures bactériennes
- 2) Préparation des bactéries pour les expériences

### IV – Préparation des inserts d'attractants

- 1) Préparation de l'agar
- 2) Préparation des attractants
  - a) Préparation de la serine
  - b) Préparation du toluène
  - c) Préparation du benzoate
- 3) Préparation des inserts

### V – Préparation des montages

- 1) Méthode capillaire au microscope (capillary assay)
- 2) Méthode de l'insert à la caméra (chemical-in-plug assay)

### VI – Analyse des données

### VII - Annexes

## Matériel général :

Produits	Matériel
<ul style="list-style-type: none"> <li>- bactéries</li> <li>- milieu LB</li> <li>- milieu <a href="#">M9</a> 5X Salts en poudre</li> <li>- glucose à 20%</li> <li>- <a href="#">acides casamino</a> à 20%</li> <li>- <a href="#">MgSO<sub>4</sub></a></li> <li>- mélange de minéraux</li> <li>- <a href="#">chloramphénicol</a></li> <li>- phosphate de potassium monobasique (K<sub>2</sub>HPO<sub>4</sub>) et dibasique (KH<sub>2</sub>PO<sub>4</sub>)</li> <li>- acide fort et base forte (NaOH et HCl)</li> <li>- glycérol</li> <li>- <a href="#">EDTA</a></li> <li>- <a href="#">serine</a> en poudre</li> <li>- <a href="#">toluène</a> anhydre à 99,8%</li> <li>- <a href="#">benzoate</a> en poudre</li> <li>- agar en poudre</li> <li>- eau distillée</li> </ul>	<ul style="list-style-type: none"> <li>- pots en verre de différentes tailles + couvercles</li> <li>- boîte de Pétri de 3 cm de diamètre</li> <li>- lamelles en verre</li> <li>- 3 <a href="#">cadres génétiques</a></li> <li>- balance de précision + spatule</li> <li>- éprouvette graduée de 100 mL et 1L</li> <li>- pipettes graduées de 5, 10, 20, et 50 mL + poire à pipeter</li> <li>- micropipette de 10, 100, 200, et 1000 µL</li> <li>- pipettes en plastique (coupées au bout)</li> <li>- seringue de 1 mL</li> <li>- capillaire de [ ??? ] mm de diamètre</li> <li>- tubes Falcons de 50 mL</li> <li>- incubateur</li> <li>- système d'autoclave</li> <li>- spectrophotomètre + cuves</li> <li>- centrifugeuse</li> <li>- micro-onde + gant anti-chaleur</li> <li>- microscope optique</li> <li>- boîte opaque avec un fond noir uni</li> <li>- 2 téléphones portables</li> <li>- 2 potences + noix de serrage</li> <li>- 2 supports fins en plexiglas</li> <li>- logiciel de traitement des images</li> </ul>

## I - Choix des bactéries

Notre projet se base sur un article publié par une équipe de chercheurs suisses<sup>1</sup>. Nous les avons contactés par mail, à l'adresse indiquée dans l'article. Ils ont accepté de nous envoyer 5 souches différentes de bactéries *Escherichia coli* (E. coli), qu'ils avaient modifiées génétiquement, afin de les rendre sensibles au benzoate et au toluène.

Souche	Détection		
	Serine	Toluène	Benzoate
5197	x	x	
5447	x		x
6085		x	
6068			x
5775	x		

<sup>1</sup> Roggo C, Clerc EE, Hadadi N, Carraro N, Stocker R, van der Meer JR. 2018. Heterologous expression of *Pseudomonas putida* methyl-accepting chemotaxis proteins yields *Escherichia coli* cells chemotactic to aromatic compounds. *Appl Environ Microbiol* 84:e01362-18. <https://doi.org/10.1128/AEM.01362-18>.

Les souches 5197, 5447, et 5775 sont dérivées de la souche MG1655, qui est une souche courante en laboratoire. La MG1655 permet naturellement de détecter la serine, un acide aminé très prisé par les bactéries. La souche 5197 comporte le plasmide pCR020, spécifique de la détection du toluène. La souche 5447 porte le plasmide pCR033, qui permet la détection du benzoate. La souche 5775 contient le plasmide pCR035, qui est un dérivé du plasmide pCR020, mais qui comporte une mutation empêchant la détection du toluène. Les souches 6085 et 6068 sont dérivées de la souche MG1655 modifiée afin de ne plus détecter la serine. La souche 6085 contient le plasmide pCR020, et la souche 6068 contient le plasmide pCR033.

Lorsque nous avons reçu les souches, nous avons agi dans les 24h afin de ne pas les perdre. Nous avons également travaillé en conditions stériles, c'est-à-dire sous hotte, avec des gants, et en veillant à ne pas circuler au-dessus de tubes ouverts. Chaque culture, saturée en bactéries, a été mélangée à du glycérol, puis placée à -80°C afin de bloquer leur métabolisme et pouvoir les conserver pendant plusieurs mois voire plusieurs années.

## II - Préparation des milieux

Une fois les stocks de bactéries réalisés, nous nous sommes intéressés à la préparation des milieux dans lesquels les bactéries allaient pouvoir grandir. Il existe différents types de milieux selon les applications souhaitées par la suite.

### 1) Milieu LB

Produits	Matériel
- milieu LB - eau distillée	- pot en verre + couvercle - système d'autoclave - balance de précision + spatule

Le milieu LB (Lysogeny Broth) est le milieu le plus classique pour la culture de bactéries E. coli, très riche. Il se présente sous la forme d'une poudre. On dilue une certaine masse de cette poudre dans de l'eau distillée afin d'obtenir une concentration massique de 25 g.L<sup>-1</sup> (soit 12,5 g dans 500 mL, ou 25 g dans 1L). Le mélange est ensuite autoclavé à 121°C pendant 1h.

### 2) Milieu M9-Glc-Cm30

Produits	Matériel
- milieu M9 5X Salts en poudre - glucose à 20% - acides casamino à 20% - MgSO <sub>4</sub> - mélange de minéraux - eau distillée	- pot en verre + couvercle - système d'autoclave - balance de précision + spatule - pipette graduée de 5 mL + poire à pipeter - micropipette de 200 µL + cônes

Nous avons utilisé le milieu LB pour nos premières expériences, puis nous avons continué avec un milieu plus adapté, le M9-Glc-Cm30. Il s'agit d'un milieu moins riche que le LB, qui permet d'optimiser la motilité des bactéries. Nous souhaitons un volume final de 200 mL. Pour utiliser ce milieu, on prélève 40 mL de cette poudre pour atteindre une concentration finale de 1X. On ajoute 4

mL de glucose à 20%, afin d'obtenir une concentration de  $4 \text{ g.L}^{-1}$ , soit 0,4% dans le milieu final. On ajoute également 2 mL d'acides casamino à 20%, pour atteindre une concentration de  $1 \text{ g.L}^{-1}$ , soit 0,1% dans le milieu final. On ajoute de plus 200  $\mu\text{L}$  de  $\text{MgSO}_4$ , et 200  $\mu\text{L}$  d'un mélange de minéraux. Enfin, on complète le volume par 154 mL d'eau distillée. On veille à travailler en conditions stériles, ainsi qu'à ajouter les différentes quantités de la plus grande à la plus faible.

### 3) Chemotaxis buffer

Produits	Matériel
<ul style="list-style-type: none"> <li>- phosphate de potassium monobasique (<math>\text{K}_2\text{HPO}_4</math>) et dibasique (<math>\text{KH}_2\text{PO}_4</math>)</li> <li>- acide fort et base forte (NaOH et HCl)</li> <li>- glycérol</li> <li>- EDTA</li> </ul>	<ul style="list-style-type: none"> <li>- pot en verre + couvercle</li> <li>- micropipette de 1 mL + cônes</li> <li>- balance de précision + spatule</li> <li>- éprouvette graduée de 1 L</li> <li>- système d'autoclave</li> </ul>

Les milieux de culture permettent la prolifération des bactéries. Au cours des expériences, nous avons besoin que la quantité de bactéries reste environ constante, afin d'observer un déplacement plutôt qu'une prolifération accélérée. Nous avons donc placé les bactéries dans un milieu favorisant la chimiotaxie, tout en contrôlant d'autres paramètres comme le pH. Afin de réaliser 1 L de ce tampon de chimiotaxie (chemotaxis buffer), on pèse 4,67 g de phosphate de potassium dibasique ( $\text{KH}_2\text{PO}_4$ ), et 3,14 g de phosphate de potassium monobasique ( $\text{K}_2\text{HPO}_4$ ), que l'on mélange dans 1L d'eau distillée. On ajuste le pH à 7 en ajoutant du NaOH ou du HCl. Cette préparation correspond à la partie tampon, d'une molarité de 50 mmol/L. On ajoute ensuite 476  $\mu\text{L}$  de glycérol, pour une teneur de 0,06%, ainsi que 2,9 mg d'EDTA, pour une concentration de 10  $\mu\text{mol/L}$ . On autoclave ensuite cette préparation.

## III - Préparation des bactéries

### 1) Préparation des cultures bactériennes

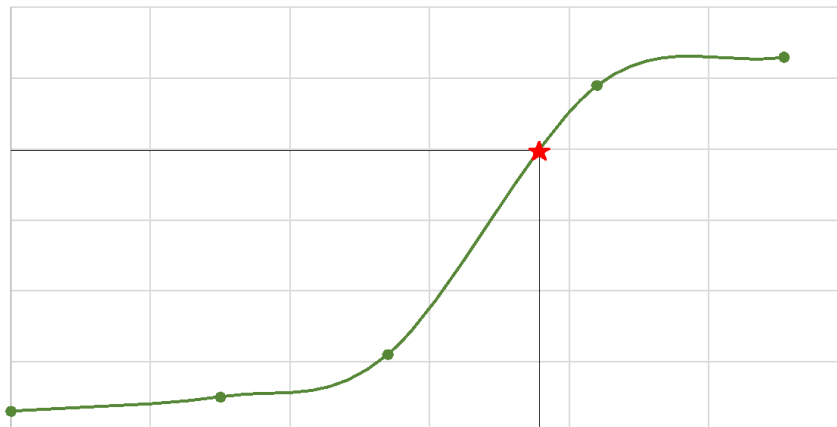
Produits	Matériel
<ul style="list-style-type: none"> <li>- stock de bactéries génétiquement modifiées, et conservées à <math>-80^\circ\text{C}</math></li> <li>- milieu M9-Glc-Cm30 préparé précédemment</li> <li>- chloramphénicol (Sigma Aldrich)</li> </ul>	<ul style="list-style-type: none"> <li>- incubateur</li> <li>- tubes Falcons de 50 mL</li> <li>- pipette graduée de 10 mL + poire à pipeter</li> <li>- micropipette de 100 <math>\mu\text{L}</math> + cônes</li> <li>- scotch</li> <li>- spectrophotomètre + cuves</li> <li>- pipettes en plastique (coupées au bout)</li> </ul>

La méthode de préparation des bactéries est la même quelle que soit la souche utilisée. On prépare des cultures overnight, puis on dilue ces cultures dans du milieu neuf, et on laisse les bactéries se développer jusqu'à atteindre leur phase de croissance exponentielle.

On prélève 10  $\mu\text{L}$  de la souche souhaitée, que l'on place dans 5 mL de milieu de culture. Les bactéries sont ensuite placées à  $37^\circ\text{C}$ , avec une agitation de 180 rpm, pendant 12h. Cette étape permet d'accumuler un grand nombre de cellules.

Après une nuit, on dilue la culture overnight par 100. Pour cela, on prélève 9,9 mL de milieu M9-Glc-Cm30, et on l'introduit dans un tube Falcon de 50 mL. On introduit également 100  $\mu$ L de culture overnight. Enfin, on ajoute 10  $\mu$ L de chloramphénicol. Le chloramphénicol est un antibiotique, qui permet de sélectionner les bactéries porteuses du gène de détection de l'attractant souhaité (toluène ou benzoate). En effet, les plasmides introduits dans les bactéries contiennent un gène de résistance à cet antibiotique. On ne conserve ainsi dans le milieu que les bactéries qui ont intégré ce plasmide. On visse ensuite le couvercle du Falcon, avant de le rouvrir d'un quart de tour pour permettre le passage de l'oxygène, et de coller un scotch pour empêcher un saut de bouchon. Le tube est placé à 37°C avec une agitation de 180 rpm, pendant environ 3h15.

L'objectif est d'atteindre une densité optique, mesurée à une longueur d'onde de 660 nm, comprise entre 0,4 et 0,6. Pour mesurer cet OD, on utilise un spectrophotomètre. On réalise un blanc avec du milieu M9-Glc-Cm30. On remplit ensuite une cuve de culture bactérienne avec une pipette dont on a coupé le bout, pour ne pas casser les flagelles des bactéries. On place ensuite la cuve devant le détecteur, et on relève la valeur. Nous avons réalisé un étalonnage afin de déterminer la durée d'apparition de la phase exponentielle.



On en déduit que la durée optimale pour atteindre la phase exponentielle, et une densité optique compatible avec les expériences, est comprise entre 3h et 3h30. On récupère les bactéries sur cette plage horaire pour la suite des manipulations.

## 2) Préparation des bactéries pour les expériences

Produits	Matériel
- culture bactérienne précédente - chemotaxis buffer	- centrifugeuse - pipette plastique (coupée au bout) - pipette graduée de 5mL + poire à pipeter

On centrifuge les bactéries en les laissant dans les Falcons, préalablement revissés et équilibrés. On règle la centrifugeuse sur 4000 tours/min, avec une montée en vitesse de 2 min, une stabilisation de 5 min, et une descente en 1 min. On observe un culot plus ou moins bien fixé au fond du tube.

On retire le surnageant, soit en le versant rapidement, soit en le pipettant. On garde le plus de bactéries possible, tout en enlevant le plus de surnageant possible. En fonction de la densité optique, on ajuste le volume à ajouter pour resuspendre les bactéries. En général, on ajoute 5 mL de chemotaxis buffer. On agite doucement le tube à la main.

#### IV – Préparation des inserts d'attractants

##### 1) Préparation de l'agar

Produits	Matériel
- agar en poudre - eau distillée	- pot en verre + couvercle - balance de précision + spatule - éprouvette graduée de 100 mL - micro-onde + gant anti-chaueur

On a besoin d'un insert solide pour qu'il ne diffuse pas trop pendant l'expérience. On utilise de l'agar à 2%. On pèse 1 g d'agar, puis on le dissout dans 50 mL d'eau distillée. On chauffe ce mélange au micro-onde, en agitant régulièrement, jusqu'à obtenir un mélange homogène. Attention, l'agar a tendance à déborder assez rapidement (quelques secondes une fois chaud) ! On attend de pouvoir toucher le contenant sans se brûler (environ 50°C) avant d'utiliser cet agar.

Une fois l'agar complètement solidifié, il est possible de le refaire chauffer un nombre infini de fois pour l'utiliser.

##### 2) Préparation des attractants

Nous avons souhaité étudier le lien entre la concentration d'attractant et la vitesse/la quantité d'accumulation des bactéries. Nous avons donc préparé des solutions de différentes concentrations, pour chacun des trois attractants.

###### a) Préparation de la serine

Avant d'utiliser le toluène et le benzoate, nous avons effectué des tests avec la serine, puisque toutes les souches de bactéries E. coli sont naturellement attirées par cette substance, et de manière plus prononcée que pour le toluène et le benzoate, même après modification génétique.

Produits	Matériel
- serine en poudre - eau distillée	- balance de précision + spatule - pipettes graduées de 10, 20 mL + poire à pipeter - éprouvette graduée de 100 mL - 5 petits contenants en verre

On prépare d'abord une solution de 250 mg/L. Pour cela, on pèse 25 mg de serine, que l'on dissout dans 100 mL d'eau distillée. Pour réaliser une solution à 200 mg/L, on prélève 16 mL de cette solution à 250 mg/L, et on complète avec 4 mL d'eau distillée. Pour une solution à 100 mg/L, on prélève 10 mL de la solution à 200 mg/L, et on complète avec 10 mL d'eau distillée. On dilue à nouveau par deux pour obtenir la solution à 50 mg/L. On prélève 10 mL de la solution à 100 mg/L, et

on complète avec 10 mL d'eau distillée. Enfin, pour réaliser une solution à 150 mg/L, on prélève 12 mL de la solution à 250 mg/L, et on complète avec 8 mL d'eau distillée. On obtient finalement des solutions à 50, 100, 150, 200 et 250 mg/L.

#### b) Préparation du toluène

Une fois les techniques de préparation maîtrisées, nous avons utilisé le toluène et le benzoate, afin de tester à la fois l'efficacité de la modification génétique, et la spécificité des bactéries. Nous avons réalisé, comme pour la serine, des solutions d'étalonnage en concentration.

Produits	Matériel
- toluène anhydre à 99,8% - eau distillée	- pipettes graduées de 2, 5, et 10 mL + poire à pipeter - micropipette de 10 µL + cônes - 6 petits contenants en verre

On commence par préparer une solution initiale de toluène anhydre à 99,8%, afin de connaître précisément sa concentration, et se rapprocher de la plage de concentrations souhaitée. On prélève donc 10 µL de toluène, que l'on place dans 1,6 mL d'eau distillée. On obtient ainsi une solution d'une concentration de 60 mmol/L. On souhaite préparer des solutions à 30 µmol/L, 60 µmol/L, 100 µmol/L, et 200 µmol/L, car les bactéries détectent le plus efficacement ces polluants autour de ces valeurs. On réalise également une solution à 6 mmol/L afin d'étudier le comportement des bactéries pour des concentrations plus importantes. Pour réaliser une solution à 30 µmol/L, on place 5 µL de la solution initiale de toluène dans 10 mL d'eau distillée. Pour les solutions à 60, 100 et 200 µmol/L, on prélève 10 µL de la solution initiale, que l'on dilue respectivement dans 10, 6, et 3 mL. Enfin, on prépare la solution à 6 mmol/L en plaçant 100 µL de la solution initiale dans 1 mL d'eau distillée.

#### c) Préparation du benzoate

Nous n'avons pas eu le temps d'étudier la réponse des bactéries aux variations de concentration en benzoate, seulement de vérifier la spécificité de ces bactéries. Nous avons donc préparé une seule solution de benzoate, à une concentration à laquelle la réponse est la plus forte.

Produits	Matériel
- benzoate en poudre - eau distillée	- balance de précision + spatule - pipette graduée de 50 mL + poire à pipeter - petit contenant en verre

On pèse 122 mg de benzoate, que l'on dissout dans 50 mL d'eau distillée. On obtient ainsi une solution d'une concentration de 16,8 mmol/L.

### 3) Préparation des inserts

Produits	Matériel
- agar à 2% préparé précédemment	- pipette graduée de 2 mL



- attractant préparé précédemment	- micropipette de 200 µL + cônes - petit contenant en verre - seringue de 1 mL - capillaire de [ ??? ] mm de diamètre
-----------------------------------	--

On prélève 1,4 mL d'agar à 2% et 0,15 mL d'attractant (serine, toluène, ou benzoate, quelle que soit leur concentration), que l'on place dans un petit contenant en verre. On mélange manuellement pour lier les deux, car l'agar refroidit très vite. En fonction de la méthode utilisée, on remplit de cette préparation soit une seringue de 1 mL, soit un capillaire de [ ??? ] mm de diamètre par capillarité, et on laisse solidifier. Il n'est pas nécessaire de remplir entièrement la seringue et le capillaire, 0,3 mL pour l'une et 2 cm pour l'autre suffisent amplement.

## V – Préparation des montages

Nous avons étudié la mobilité et la chimiotaxie de nos bactéries grâce à deux méthodes principales : la méthode par capillaire, et la méthode de l'insert chimique. Pour vérifier la mobilité des bactéries au préalable, nous avons simplement placé une goutte de culture bactérienne sur une lame de verre, recouverte d'une lamelle, puis nous l'avons observée au microscope.

### 1) Méthode capillaire au microscope (capillary assay)

Produits	Matériel
- bactéries préparées pour la chimiotaxie - capillaire préparé précédemment	- microscope optique - boîte de Pétri de 3 cm de diamètre

On remplit une boîte de Pétri en verre de 3 cm de diamètre, avec les bactéries préparées pour la chimiotaxie, grâce à une pipette plastique coupée au bout. On pose le capillaire rempli du mélange agar à 2% + attractant sur le bord de la boîte, en veillant à placer l'extrémité en contact avec les bactéries. On place ce montage sous un microscope, et on enregistre des images régulières, une toutes les 30 à 60 secondes, pendant 15 min.

### 2) Méthode de l'insert à la caméra (chemical-in-plug assay)

Produits	Matériel
- bactéries préparées pour la chimiotaxie - seringue préparée précédemment (agar à 2% + attractant)	- boîte opaque avec un fond noir uni - 2 téléphones portables - 2 potences + noix de serrage - 2 supports fins en plexiglas - 3 cadres génétiques - lamelles en verre

On construit un contenant pour les bactéries, en superposant 3 cadres génétiques, que l'on colle sur une lamelle de verre. On conserve le tour en plastique sur le dessus pour éviter la colle.

On pousse doucement sur la seringue pour faire sortir quelques millimètres d'agar. A l'aide d'un scalpel, on coupe un cylindre d'une hauteur équivalente à celle des bords du contenant. On veille à produire le cylindre le plus homogène possible (bords droits et parallèles), afin de ne pas introduire d'irrégularités pour les mesures. On place ce cylindre au centre du contenant. On vérifie la hauteur du cylindre en plaçant une lamelle en verre au-dessus : si la lamelle touche le cylindre et chaque bord du contenant, c'est bon ! Sinon, on reprend le protocole au début du paragraphe.

On remplit ensuite le contenant du mélange bactéries + chemotaxis buffer, grâce à une pipette plastique coupée au bout, afin de ne pas abîmer les flagelles de nos bactéries. On recouvre le contenant avec une lamelle en verre, en veillant à la fois à ne pas introduire de bulles d'air, et à ne pas faire déborder les bactéries.

Attention, pour plus de répétabilité, la durée entre le remplissage du contenant et le début de l'expérience doit être la plus courte possible ! Dans la boîte opaque, on place les supports fins en plexiglas écartés d'une distance équivalente à la longueur d'une lamelle, puis on pose le contenant en équilibre dessus, de manière à le surélever. On fixe un premier téléphone portable sur une potence, avec sa lumière allumée et dirigée vers le bas, afin que le contenant soit éclairé de façon homogène. On programme la caméra d'un deuxième téléphone portable pour une durée de 30 min, avec une photo prise toutes les 30 secondes (60 images). On fixe ce téléphone sur une potence au-dessus du contenant, et on ajuste l'image de manière à observer la totalité du cadre génétique, mais pas plus. On referme ensuite la boîte opaque et on laisse l'expérience se dérouler.

## VI – Analyse des données

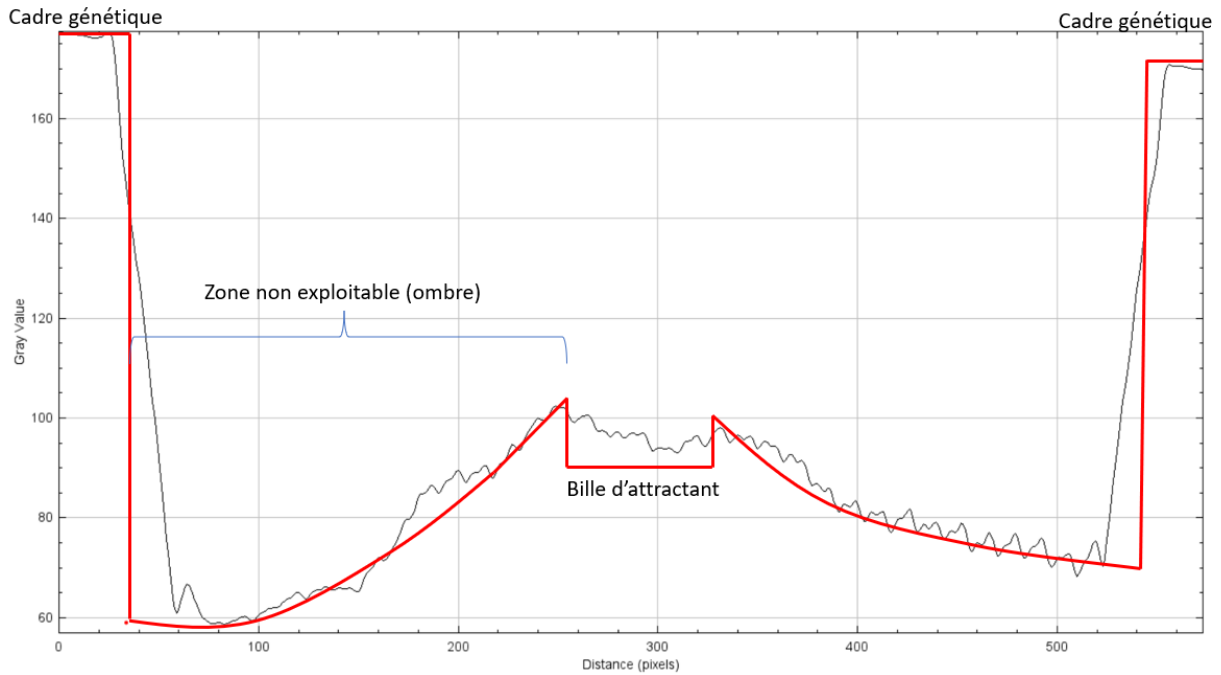
### 1) Analyse sur Image J

Une fois les photos récupérées pour nos manipulations, on importe les photos en un stack sur ImageJ. On rogne les photos de sorte à centrer l'image sur le cadre génétique.

Les images étant parfois décalées les unes par rapport aux autres (si la potence n'est pas assez serrée et que le téléphone qui filme tombe légèrement notamment), on fait en sorte de décaler les images du stack en fixant la position de l'insert : Plugins/Registration/Linear with SIFT.

On moyenne ensuite les images avec la commande Mean. En effet, le tissu en fond de chaque image implique un fond non homogène sous forme d'un quadrillage, et on souhaite diminuer l'influence de cette non-homogénéité.

On trace une ligne sur l'image qui traverse dans la largeur le cadre génétique, d'une largeur line width=30. On peut alors utiliser la commande Plot Profile pour tracer le profil d'intensité le long de cette ligne (moyenné sur une largeur de 30 pixels). On observe généralement un profil de la forme suivante :



L'objectif étant de récupérer les données de ce profil dans un fichier exploitable sous Matlab, cela pour chaque image du stack, sur la même ligne pour chaque image, on utilise un Macro ImageJ donné en annexe.

## 2) Analyse sur Python

Nous injectons ensuite ces données dans notre code Python écrit (en annexe) qui nous permet d'effectuer plusieurs choses :

- Afficher les graphiques en 3D et en 2D avec échelle de couleur (bleu/rouge) de l'évolution de l'intensité de nos pixels au cours du temps : plus l'intensité est élevée, plus nous sommes en présence d'une zone à concentration forte en bactéries.

Pour cela, nous importons le fichier .csv et nous en extrayons les données sous forme de matrice de taille Nombre de photos × Nombre de pixels. Les données sont ensuite insérées dans un code traçant les profils expliqués ci-dessus.

Pour l'ensemble de nos expériences, l'évolution du témoin conjointement aux différents tests a été soustraits à nos matrices de données, nous permettant ainsi bien de nous affranchir de changements d'intensité parasites dans le système.

- Calculer la vitesse de déplacement en pixel/photo puis mètres/secondes des bactéries du milieu et reporter cette évolution en fonction de la concentration en attractant ou polluant de l'insert.

Il a été nécessaire de réaliser une recherche de maximum d'intensité au niveau de la fin de la zone d'accumulation des bactéries, qui s'agrandit lorsque le temps défile. Le front de bactéries à l'insert quant à lui se concentre mais reste immobile car les bactéries ne sont pas en capacité de manger l'insert et donc se stockent à sa frontière. La fin de zone d'accumulation correspond à la zone de déplétion observée à l'œil nu et détermine donc la

distance à partir de laquelle les bactéries ne sont plus sensibles à la présence du polluant ou attractant dans le milieu.

La recherche de maximum s'est opérée entre les premiers instants de l'expérience (deux premières photos) et les derniers instants (deux dernières photos) afin de s'affranchir d'éventuelles erreurs de recherche de maxima et donc de fausser le calcul de vitesse.

- Réaliser un fit sur nos courbes afin d'obtenir une courbe d'étalonnage utilisable pour déduire la concentration d'un échantillon de concentration en toluène inconnu à partir de la mesure de sa vitesse de déplacement. Nous avons effectué un fit inverse qui concorde bien avec nos données expérimentales et nous avons affiché nos données finales ainsi que ce fit sur le même graphe.

## VII - Annexe

### 1) Macro Image J

```
macro "Stack profile Data" {
  if (!(selectionType()==0 || selectionType==5 || selectionType==6))
    exit("Line or Rectangle Selection Required");
  setBatchMode(true);

  run("Plot Profile");
  Plot.getValues(x, y);
  run("Clear Results");
  for (i=0; i<x.length; i++)
    setResult("x", i, x[i]);
  close();

  n = nSlices;
  for (slice=1; slice<=n; slice++) {
    showProgress(slice, n);
    setSlice(slice);
    profile = getProfile();
    sliceLabel = toString(slice);
    sliceData = split(getMetadata("Label"), "\n");
    if (sliceData.length>0) {
      line0 = sliceData[0];
      if (lengthOf(sliceLabel) > 0)
        sliceLabel = sliceLabel+ " (" + line0 + ")";
    }
    for (i=0; i<profile.length; i++)
      setResult(sliceLabel, i, profile[i]);
  }
  setBatchMode(false);
  updateResults;
}
```

## 2) Code Python

@author: lazen

```
import numpy as np
```

```
#####  
#Premiere partie de code permettant de passer du fichier .csv à une matrice numpy
```

```
# Lire le fichier csv et stocker les données dans une liste  
with open('27-01 100µM.csv', 'r') as f:  
    data = [line.strip().split(',') for line in f]
```

```
# Supprimer les apostrophes autour des valeurs  
data = np.char.strip(data, "'")
```

```
#Enlever la valeur '' et 'x' correspondant à y et x des coordonnées du graphes du début, remplacées  
temporairement par -1  
data[data==''] = -1  
data[data=='x'] = -1
```

```
# Supprimer les apostrophes autour des valeurs  
data = np.char.strip(data, "'")
```

```
# Convertir les données en float  
data = data.astype(np.float)
```

```
# Convertir la liste en matrice numpy  
data = np.array(data, dtype=np.float)
```

```
# Supprimer la première ligne et la première colonne  
data = data[1:, 2:]
```

```
#print(data)
```

```
import matplotlib.pyplot as plt  
from mpl_toolkits.mplot3d import Axes3D
```

```
## Créer une matrice de données pour chaque axe
```

```
shape = data.shape#on récupère la taille de data pour avoir en x le nombre de photos et en y le numéro du  
pixel  
x = np.arange(0, shape[1],1)  
y = np.arange(0,shape[0],1)  
z = data  
x, y = np.meshgrid(x, y)  
#on a donc "au sol" du graphe 3D le numéro de photo en x, le numéro du pixel en y  
#en ordonnée on a l'intensité du pixel, analysé en noir et blanc sur image J.
```

```
# Créer une figure et un axe 3D  
fig = plt.figure()  
ax = fig.add_subplot(111, projection='3d')
```

```
# Tracer la surface  
ax.plot_surface(x, y, z)
```

```
# Ajouter des étiquettes d'axe  
ax.set_xlabel('Numéro de photo')  
ax.set_ylabel('Numéro de Pixel')  
ax.set_zlabel('Intensité du pixel')
```

```
# Afficher le graphique  
plt.show()
```

```
# Afficher le diagramme couleur  
plt.imshow(z, cmap='coolwarm')
```

```
# Ajouter une barre de couleur  
plt.colorbar()
```

```
# Afficher le graphique couleur avec la barre  
plt.show()  
#####
```

```
#Deuxieme partie de code : on cherche la valeur de concentration la plus faible/la plage de valeur  
correspondant à l'anneau de déplétion  
#puis on souhaite suivre son évolution d'une photo à l'autre au niveau des pixels afin de connaître sa vitesse de  
déplacement  
#on réalise enfin la moyenne de ces vitesses instantanées d'une photo à l'autre pour avoir la vitesse moyenne
```

```

#pour avoir la plage de valeur minimale il faut penser afin de pouvoir comparer les résultats entre chaque
expérience
#à retirer les valeurs de pixels du témoin

def create_matrix_from_stacks():
    # Lire le fichier csv et stocker les données dans une liste
    with open('27-01 200µM.csv', 'r') as f:
        data = [line.strip().split(',') for line in f]

    # Supprimer les apostrophes autour des valeurs
    data = np.char.strip(data, "'")

    #Enlever la valeur "" et 'x' correspondant à y et x des coordonnées du graphes du début, remplacées
temporairement par -1
    data[data==""] = -1
    data[data=='x'] = -1

    # Supprimer les apostrophes autour des valeurs
    data = np.char.strip(data, "'")

    # Convertir les données en float
    data = data.astype(np.float)

    # Convertir la liste en matrice numpy
    data = np.array(data, dtype=np.float)

    # Supprimer la première ligne et la première colonne
    data = data[1:, 2:]

    return data, data.shape

def create_matrix_from_stacks_test():
    # Lire le fichier csv et stocker les données dans une liste
    with open('27-01 témoin.csv', 'r') as f:
        data = [line.strip().split(',') for line in f]

    # Supprimer les apostrophes autour des valeurs
    data = np.char.strip(data, "'")

    #Enlever la valeur "" et 'x' correspondant à y et x des coordonnées du graphes du début, remplacées
temporairement par -1
    data[data==""] = -1
    data[data=='x'] = -1

    # Supprimer les apostrophes autour des valeurs
    data = np.char.strip(data, "'")

    # Convertir les données en float
    data = data.astype(np.float)

    # Convertir la liste en matrice numpy
    data = np.array(data, dtype=np.float)

    # Supprimer la première ligne et les deux premières colonnes
    data = data[1:, 2:]

    return data, data.shape

#fonction pour recup la matrice en enlevant le témoin soit en s'affranchissant des conditions d'éclairage
(reproductibilité de l'expérience)

def matrix_without_test0() :
    matrix_stacks = create_matrix_from_stacks()[0]
    #matrice test pour les expe sans témoin
    # matrix_test = np.zeros(create_matrix_from_stacks()[0].shape)

    #matrice test pour les expe avec témoin
    matrix_test = create_matrix_from_stacks_test()[0]

    #Ajuste la taille de deux matrices en supprimant les dernières lignes ou colonnes pour les rendre de la même
taille.
    #Retourne les deux matrices ajustées.
    lignes_matrice1, colonnes_matrice1 = matrix_stacks.shape
    lignes_matrice2, colonnes_matrice2 = matrix_test.shape

    # Si les matrices ont déjà la même taille, retourner les deux matrices telles quelles
    if lignes_matrice1 == lignes_matrice2 and colonnes_matrice1 == colonnes_matrice2:
        return matrix_stacks-matrix_test

```

```

# Si le nombre de lignes de la matrice 1 est supérieur à celui de la matrice 2, supprimer les dernières lignes de
la matrice 1
elif lignes_matrice1 > lignes_matrice2:
    difference_lignes = lignes_matrice1 - lignes_matrice2
    matrix_stacks = matrix_stacks[:-difference_lignes,:]

# Si le nombre de lignes de la matrice 2 est supérieur à celui de la matrice 1, supprimer les dernières lignes de
la matrice 2
elif lignes_matrice2 > lignes_matrice1:
    difference_lignes = lignes_matrice2 - lignes_matrice1
    matrix_test = matrix_test[:-difference_lignes,:]

# Si le nombre de colonnes de la matrice 1 est supérieur à celui de la matrice 2, supprimer les dernières
colonnes de la matrice 1
if colonnes_matrice1 > colonnes_matrice2:
    difference_colonnes = colonnes_matrice1 - colonnes_matrice2
    matrix_stacks = matrix_stacks[:,:-difference_colonnes]

# Si le nombre de colonnes de la matrice 2 est supérieur à celui de la matrice 1, supprimer les dernières
colonnes de la matrice 2
elif colonnes_matrice2 > colonnes_matrice1:
    difference_colonnes = colonnes_matrice2 - colonnes_matrice1
    matrix_test = matrix_test[:,:-difference_colonnes]

return matrix_stacks-matrix_test

#print(matrix_without_test0().shape)

#fonction passant des données matricielles récupérées aux graphes 2D et 3D
def from_matrix_to_graphic() :

    # Créer une matrice de données pour chaque axe
    shape = matrix_without_test0().shape#on récupère la taille de data pour avoir en x le nombre de photos et
en y le numero du pixel
    x = np.arange(0, shape[1],1)
    y = np.arange(0,shape[0],1)
    z = matrix_without_test0()
    x, y = np.meshgrid(x, y)
    #on a donc "au sol" du graphe 3D le numéro de photo en x, le numéro du pixel en y
    #en ordonnée on a l'intensité du pixel, analysé en noir et blanc sur image J.

    # Créer une figure et un axe 3D
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')

    # Tracer la surface
    ax.plot_surface(x, y, z)

    # Ajouter des étiquettes d'axe
    ax.set_xlabel('Numéro de photo')
    ax.set_ylabel('Numéro de Pixel')
    ax.set_zlabel('Intensité du pixel')

    # Afficher le graphique
    plt.show()

    # Afficher le diagramme couleur
    plt.imshow(z, cmap='coolwarm')

    # Ajouter une barre de couleur
    plt.colorbar()

    # Afficher le graphique couleur avec la barre
    plt.show()

from_matrix_to_graphic()
#pas possible d'automatiser au sein de la fonction avec le nom du fichier .csv donc il faudra éditer la démarche
pour chaque concentration différente

#calcul de la vitesse à partir du relevé des graphes 3D et en couleur avec prise en compte du témoin
#pour faire ce calcul il faut chercher le maximum d'intensité (correspondant à un amas maximal de bactéries et
donc plus ou moins au front de déplacement)
#puis regarder où celui-ci se situe à la fin de l'expérience par rapport au début au niveau des pixels de la photo
#il faut séparer le relevé en 2 car il y a la tache d'insert au milieu et donc deux fronts de déplacement (coupe en
ligne sur toute la largeur de la boîte avec insert centré au milieu)
#on peut envisager ensuite de faire une moyenne sur les deux fronts (si cela est cohérent)

#fonction pour chercher la valeur max dans notre matrice
#avant cela il faut "découper à la main les zones pertinentes de chaque relevé pour ensuite rechercher le max
dans les bonnes parties

```

#Paramètres à déterminer à chaque fois, a et b pour les photos, on peut se restreindre à quelques colonnes au début et quelques colonnes à la fin pour avoir les deux positions de pixels pour calculer la vitesse

```
def decoupe_matrix(a,b,c,d,matrice):
```

```
    #a,b,c,d = bornes du découpage de données conservées dans la matrice
    matrice = matrice[c:d, a:b]
    return matrice
```

```
def trouver_valeur_maximale_et_indices(matrice):
```

```
    # Recherche et retourne la valeur maximale dans une matrice ainsi que ses indices.
    valeur_maximale = np.max(matrice)
    indices_max = np.unravel_index(np.argmax(matrice), matrice.shape)
    return valeur_maximale, indices_max
```

#fonction pour calculer les deux valeurs extrêmes de la matrice pour une expérience

```
def extremal_value(a_1,b_1,a_2,b_2,c,d,matrice):
```

```
    #les ai et bi sont pour choisir les photos utiles à la recherche
    matrice_debut = decoupe_matrix(a_1, b_1, c, d, matrice)
    matrice_fin = decoupe_matrix(a_2, b_2, c, d, matrice)
    max_debut, indice_debut = trouver_valeur_maximale_et_indices(matrice_debut)
    max_fin, indice_fin = trouver_valeur_maximale_et_indices(matrice_fin)
    return indice_debut, indice_fin
```

#fonction pour calculer la vitesse de déplacement finale de pixel à pixel

#attention on a coupé les matrices dont les indicages des photos ont changé

#il faut rééquilibrer en ajoutant le nombre de départ a\_i du cut

```
def speed(a_1,b_1,a_2,b_2,c,d,matrice):
```

```
    indice_debut = extremal_value(a_1, b_1, a_2, b_2, c, d, matrice)[0]
    indice_fin = extremal_value(a_1, b_1, a_2, b_2, c, d, matrice)[1]
    vitesse = (indice_fin[0]-indice_debut[0])/(indice_fin[1]+a_1-indice_debut[1]+a_2)
    return indice_debut, indice_fin, vitesse
```

#pour les expériences sans témoin, on se rend compte que les intensités de pixels sont inversées

#il faut donc chercher ici les valeurs minimales dans les matrices afin d'avoir les fronts de bactéries

```
def trouver_valeur_minimale_et_indices(matrice):
```

```
    # Recherche et retourne la valeur maximale dans une matrice ainsi que ses indices.
    valeur_minimale = np.min(matrice)
    indices_min = np.unravel_index(np.argmin(matrice), matrice.shape)
    return valeur_minimale, indices_min
```

#fonction pour calculer les deux valeurs extrêmes de la matrice pour une expérience sans témoin

```
def extremal_value_2(a_1,b_1,a_2,b_2,c,d,matrice):
```

```
    #les ai et bi sont pour choisir les photos utiles à la recherche
    matrice_debut = decoupe_matrix(a_1, b_1, c, d, matrice)
    matrice_fin = decoupe_matrix(a_2, b_2, c, d, matrice)
    min_debut, indice_debut = trouver_valeur_minimale_et_indices(matrice_debut)
    min_fin, indice_fin = trouver_valeur_minimale_et_indices(matrice_fin)
    return indice_debut, indice_fin
```

#fonction pour calculer la vitesse de déplacement finale de pixel à pixel

#attention on a coupé les matrices dont les indicages des photos ont changé

#il faut rééquilibrer en ajoutant le nombre de départ a\_i du cut

```
def speed_2(a_1,b_1,a_2,b_2,c,d,matrice):
```

```
    indice_debut = extremal_value_2(a_1, b_1, a_2, b_2, c, d, matrice)[0]
    indice_fin = extremal_value_2(a_1, b_1, a_2, b_2, c, d, matrice)[1]
    vitesse = (indice_fin[0]-indice_debut[0])/(indice_fin[1]+a_1-indice_debut[1]+a_2)
    return indice_debut, indice_fin, vitesse
```

#quand on a obtenu la vitesse, on réitère l'opération pour chaque photo avec soustraction de son témoin, choix de la plage de pixels

#et de la plage de photos et ensuite en fonction de la concentration initiale on trace un graphique vitesse = f(concentration)

#attention à bien tracer un graphe différent pour chaque espèce étudiée

#attention le slicing des matrices cest valeur finale exclue

#commande utilisée pour calculer pour les expe avec temoins

#attention bien choisir la matrice normale pour le "temoin" dans le code create\_matrix\_from\_stacks\_test()

#print(speed(8,11,35,41,390,465, matrix\_without\_test0()))

#commande utilisée pour les expe sans temoin pour toluene et benzoate

#attention bien choisir la matrice np.zero pour le "temoin" dans le code create\_matrix\_from\_stacks\_test()

#print(create\_matrix\_from\_stacks()[1])

print(speed\_2(0,2,39,41,260,320, matrix\_without\_test0()))

#troisième partie : tracer sur un graphe 2D l'évolution de la vitesse moyenne en fonction de la concentration

#pour cela penser à mettre les première et deuxième partie sous forme de fonction python



```

#remise au propre des données récupérées après calcul manuel via le code ci dessus
#donnees finales = [{"experience", "photo debut in", "photo debut fin", "photo fin in haut", "photo fin fin haut", "photo fin in bas", "photo fin fin bas", "vitesse haut", "vitesse bas"}, [{"03-02 6mM", 0,3,60,61,110,201,250,350,0.01886792,0.01886792}, {"03-02 100µM", 0,3,50,61,110,201,250,350,0.054545454,0.29629629}, {"27-01 6mM", 0,3,35,41,100,111,310,381,0.0285714265,0.324324324}, {"27-01 30µM", 0,3,35,41,120,162,230,271,0.4,0.39393939}, {"27-01 200µM", 0,3,35,41,120,151,220,290,0,0.34}, {"tol 5197", 0,2,38,41,120,180,0.3589743589,0.0263157}, {"tol 5447", 0,2,38,41,120,200,280,350,0.307692307,0.3684210526}, {"benzo 5197", 0,2,18,21,160,211,300,350,0.05555555,0.05555555}, {"benzo 5447", 0,2,18,21,150,190,266,310,0.8235294,0.77777}]]
#print(donnees_finales)
#on garde les vitesses qui semblent les plus coherentes pour les tracés
#il reste donc à réitérer le code pour chaque expériences différentes afin de remplir
#un tableau numpy contenant toutes ls vitesses puis d'afficher le graphe
#vitesses_finales_25 = np.array([[6000,100],[0.1886792,0.29629]])
#vitesses_finales_27 = np.array([[6000,30,200],[0.32432432,0.39393939,0.3421052631]])

#on rassemble tout dans un seul tableau de données en faisant des moyennes sur les valeurs obtenues si
#besoin
vitesses_nouv_finales_tot = np.array([[6000,30,100,200],[0.02631578947368421,0.6086956521739131,0.2831215970961888,0.1684210526315789]])
vitesses_nouv_finales_tot_2 = np.array([[6000,30,100,200],[0.02631578947368421/30,0.6086956521739131/30,0.2831215970961888/30,0.1684210526315789/30]])

def tracer_graphe(x_1,y_1,x_2,y_2):
    plt.subplots()
    plt.plot(x_1,y_1)
    plt.plot(x_2,y_2)
    plt.xlabel('concentration (µMol)')
    plt.ylabel('vitesse')
    plt.title('Vitesse de déplacement des bactéries en fonction de la concentration')
    plt.show()

def tracer_graphe_2(x,y):
    plt.plot(x,y, '.')
    plt.xlabel('concentration (µMol)')
    plt.ylabel('vitesse')
    plt.title('Vitesse de déplacement des bactéries en fonction de la concentration')
    plt.show()

tracer_graphe_2(vitesses_nouv_finales_tot[0],vitesses_nouv_finales_tot[1])

# on cherche enfin à effectuer un fit sur cette courbe pour pouvoir connaitre ou placer nos points pour
#le benzo et le toluene et etudier lequel serait le plus réactifs à nos bactéries

from scipy.optimize import curve_fit

def inverse_regression(x, y, n,x_t,y_t):
    # Définition de la fonction de régression
    def inverse_function(x, a, b):
        return a / (x**n) + b

    # Régression
    popt, pcov = curve_fit(inverse_function, x, y)
    plt.subplots()
    # Tracé des points de données
    plt.scatter(x, y, label='Données')
    plt.scatter(x_t, y_t, label = 'Théorie')
    # Tracé de la régression
    x_fit = np.linspace(min(x), max(x), 100)
    y_fit = inverse_function(x_fit, *popt)
    #plt.plot(x_fit, y_fit, 'r', label='Régression')
    plt.plot(x_t, y_t, 'b', label = 'Théorie')
    # Utiliser une échelle logarithmique sur l'axe des x et y
    plt.xscale('log')
    plt.yscale('log')
    # Ajout d'une légende et affichage du graphe
    plt.legend()
    plt.xlabel('Concentration insert µmol/L')
    plt.ylabel('Vitesse bactéries pix/sec')
    plt.title('Régression 1/(x^{n}).format(n))
    plt.grid(True)
    plt.show()

```

```
# regression en fonction inverse
n = 1/2
```

```
inverse_regression(vitesses_nouv_finales_tot_2[0],vitesses_nouv_finales_tot_2[1],
n,courbe_théo_2[0],courbe_théo_2[1])
print(vitesses_nouv_finales_tot_2, courbe_théo_2)
```

```
# On trace la régression pour en déduire la concentration de nos bactéries pouvant correspondre
#aux vitesses relevées pour le toluene et le benzoate
#on affiche les points du benzoate et du toluene sur notre régression finale
```

```
#analyse effectué résumé : front de concentration des bactéries (en rouge), mais sur le bas, soit au niveau de
l'accumulation
```

```
#car au niveau de l'insert on a juste une intensité qui augmente mais cela ne veut pas dire grand chose
#car les bactéries peuvent s'accumuler de maniere assez anarchique (on a pas plus d'infos sur cette dynamique
là)
```

```
#ici on recommence la mesure des vitesses avec les codes déjà écrit mais sur le bas des figure en couleur
```

```
#on analyse les donnes du O3-02 et du 27-01 sur le toluene pour avoir l'étalonnage
```

```
#on obtient une vitese en pixels/seconde (pente suivant sur les y)
```

```
#1photo = intervalle de 30secondes donc axe des x à multiplier par 30 pour convertir en secondes
```

```
def intensity_global_insert(a_1,b_1,a_2,b_2,c,d,matrice):
    matrice_debut = decoupe_matrix(a_1, b_1, c, d, matrice)
    matrice_fin = decoupe_matrix(a_2, b_2, c, d, matrice)
    max_debut, indice_debut = trouver_valeur_maximale_et_indices(matrice_debut)
    max_fin, indice_fin = trouver_valeur_maximale_et_indices(matrice_fin)
    return max_fin - max_debut
```