

Spectroscopie de Fluorescence pour la Dermatologie

Talla DIAGNE et Cyprien LANNEAU G1

Nous attestons que ce travail est original, que nous citons en référence toutes les sources utilisées et qu'il ne comporte pas de plagiat.

Résumé

Les kératoses actiniques sont des excroissances précancéreuses de la peau provoquées par l'exposition solaire prolongée. C'est un des diagnostics les plus courants en dermatologie surtout dans la population âgée, on observe ainsi une prévalence de 14% dans la population mondial. Un des traitements possibles pour les kératoses actiniques est la photothérapie dynamique qui se base sur l'utilisation d'une molécule photosensible pour détruire la pathologie. Cette molécule utilisée se nomme la protoporphyrine IX et a la propriété d'être fluorescente. Cette propriété atypique nous permet d'étudier dans ce projet l'évolution de la concentration de cette molécule lors de traitements par photothérapie dynamique (PDT).

Nous présentons dans ce rapport les bases théoriques pour comprendre le sujet ainsi que notre approche détaillée du traitement des données expérimentales. Nos résultats obtenus sur différents patients traités par PDT suggèrent une efficacité du traitement cependant certaines observations tendent à montrer que les croûtes présentes sur les peaux traitées limitent l'efficacité du traitement.

Remerciements

Nous tenons à remercier Arthur GAUTHERON, le post-doctorant qui a permis la réalisation de ce projet ainsi que le Dr. Jean Luc PERROT qui nous a accueillis dans le service de dermatologie du CHU de Saint-Etienne.

Table des matières

1	Introduction	2
2	Base théoriques	3
3	Protocole expérimental	6
3.1	Acquisition des données	6
3.2	Traitement des données	7
4	Analyse critique des résultats	12
5	Conclusion	13
6	Références bibliographiques	14
7	Annexe	15

1 Introduction

La photothérapie dynamique est un traitement utilisé en dermatologie pour traiter les kératoses actiniques, des excroissances précancéreuses de la peau souvent diagnostiquées en dermatologie. Pour suivre l'évolution de ce traitement et ainsi quantifier son efficacité nous avons utiliser la spectroscopie de fluorescence qui nous donne des informations sur les conditions cutanées au niveau moléculaire de manière non invasive. Ce projet explore ainsi l'application de la spectroscopie de fluorescence dans une étude dermatologique, en mettant particulièrement l'accent sur l'acquisition, le traitement et l'analyse des données. En utilisant Python pour la manipulation et la visualisation des données, le projet vise à mieux comprendre les schémas de fluorescence cutanée et leur corrélation avec divers paramètres dermatologiques. Les parties suivantes fournissent la base théorique nécessaire, les protocoles expérimentaux, et une analyse critique des résultats, permettant ainsi d'explorer totalement le potentiel de la spectroscopie de fluorescence dans la recherche dermatologique.



FIGURE 1 – Application de la photothérapie dynamique

2 Base théoriques

Le traitement par photothérapie dynamique (PDT) se base sur l'utilisation de la protoporphyrine IX (PpIX), une molécule naturellement métabolisée par le corps. Cette molécule est photosensible et transforme l'énergie lumineuse pour produire des espèces chimiques oxydées, ce qui tue les cellules environnantes. Pour cibler uniquement les zones précancéreuses de la peau on applique une crème à base d'Acide 5-aminolévulinique (5-ALA), un précurseur biochimique qui va être transformé par le corps en protoporphyrine IX. Différents mécanismes vont permettre ensuite de transformer cette protoporphyrine IX dans les tissus sains et ainsi de baisser fortement sa concentration tandis qu'elle va se retrouver concentrée dans les tissus précancéreux (Figure[3]). Cela va permettre au traitement de cibler plus efficacement les kératoses actiniques.

La PpIX a deux pics d'absorption, un principal à 405 nm et un deuxième à 635 nm. Ce sont les logueurs d'onde utilisées dans les sources lumineuses pour activer la PpIX lors de PDT.

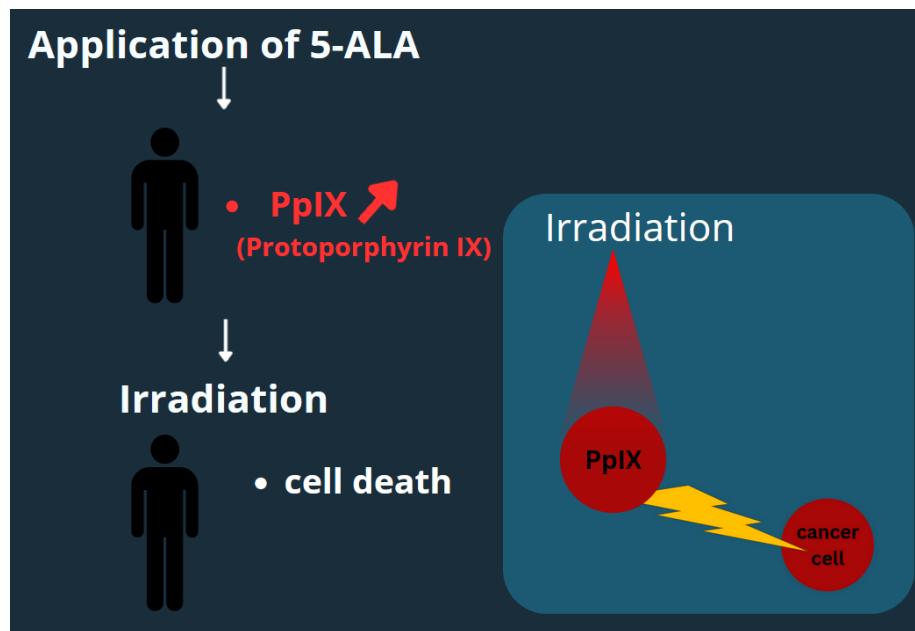


FIGURE 2 – Schéma de principe de la photothérapie

Le traitement par la source lumineuse se fait environ 3h après l'application de la crème 5-ALA. L'exposition à la source lumineuse a souvent pour effet secondaire une sensation de brûlure au niveau de la zone de peau traitée.

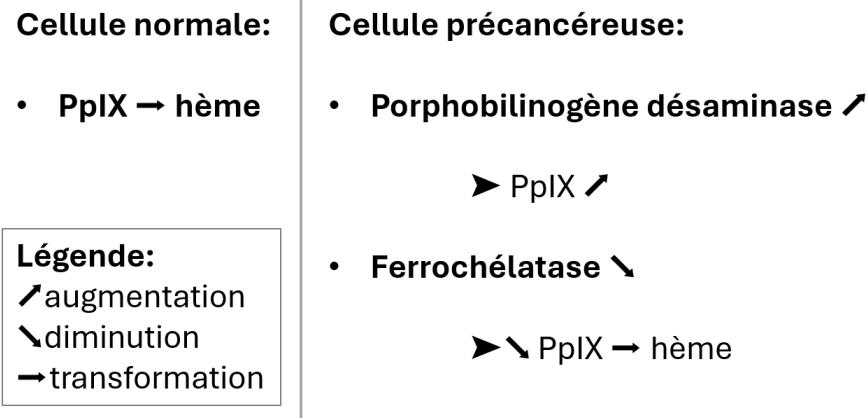


FIGURE 3 – Schéma des mécanismes moléculaires

La figure 3 décrit de manière sommaire les mécanismes moléculaires qui conduisent à une plus forte concentration de la PpIX dans les cellules précancéreuses. En effet la PpIX est une molécule qui permet la production de l'hème, molécule notamment présente dans le sang. Or ce processus de production se retrouve dérèglé dans les cellules précancéreuses ce qui conduit à une augmentation de la concentration de PpIX. De plus la production de PpIX est augmentée par l'application en crème d'acide 5-aminolévulinique qui est transformée par le corps en PpIX.

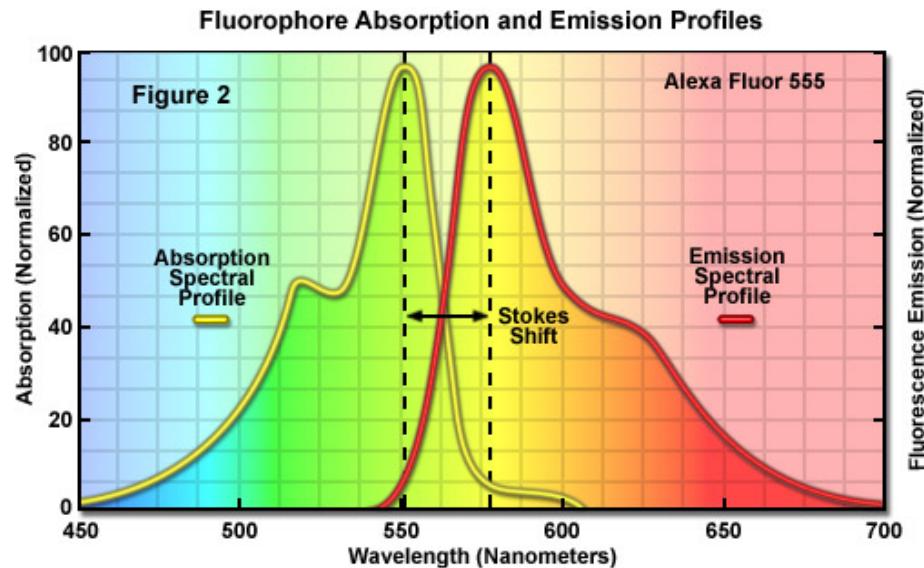


FIGURE 4 – représentation du principe de fluorescence

La protoporphyrine IX (PpIX) a aussi la propriété d'être fluorescente. La fluorescence est la propriété physique qu'ont certaines molécules/matériaux à émettre l'énergie lumineuse qu'ils ont reçue (comme décrit en Figure[4]). Cette émission se fait dans une longueur d'onde différente que celle reçue. Cette propriété de fluorescence nous permet d'observer la présence de PpIX puisque cette molécule émet à deux longueurs d'onde spécifiques : 622 nm et 684 nm (Figure[5]).

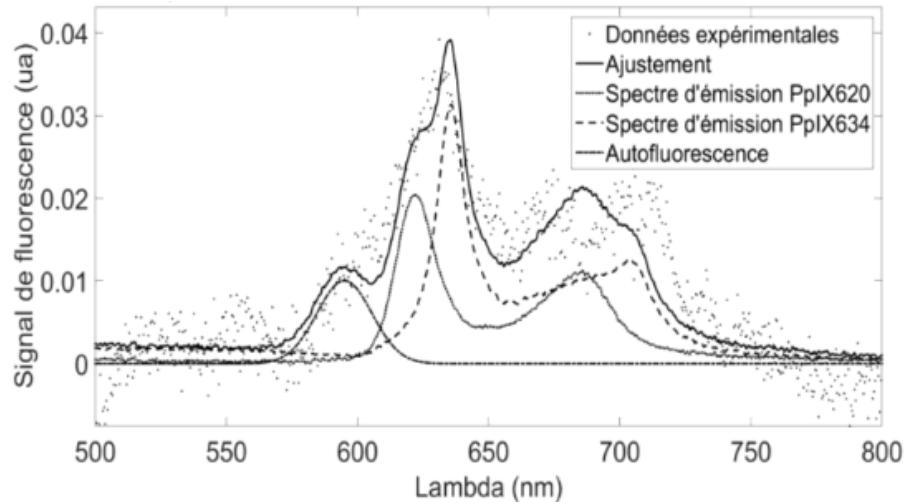


FIGURE 5 – Spectre d'émission de la PpIX

En plus du spectre d'émission de la PpIX il faut prendre en compte la capacité fluorescente naturelle du corps que l'on appelle l'autofluorescence. Dans la partie expérimentale nous allons soustraire ce spectre d'autofluorescence pour ne garder que le spectre d'émission par fluorescence de la PpIX.

3 Protocole expérimental

Le projet s'est déroulé en deux phases : la première consiste à acquérir des données sur les patients, et la seconde à les analyser pour statuer sur l'efficacité du traitement.

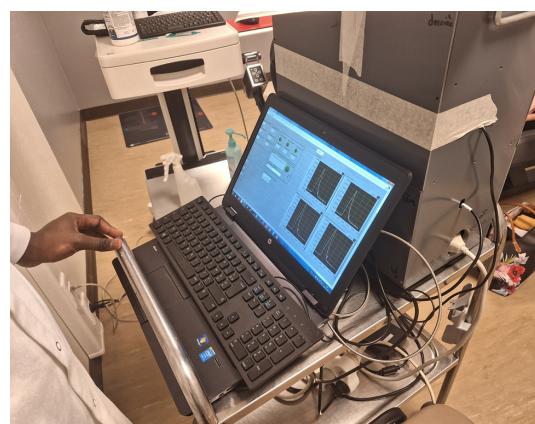
3.1 Acquisition des données

Dans le cadre du projet, nous avons eu l'occasion d'aller au CHU Nord Saint-Etienne pour effectuer quelques mesures. On a ainsi pu se renseigner sur la journée type d'un patient.

Ce dernier arrive dans la matinée et on effectue une première mesure (que l'on appellera *Condition 0* pour la suite) sur la zone à traiter à l'aide d'une sonde (Figure[6a]) reliée à une spectromètre puis à un ordinateur (Figure[6b]).



(a) Sonde utilisée



(b) PC avec le spectromètre

FIGURE 6 – Protocole expérimental d'acquisition des données

L'acquisition des spectres est déclenchée par un utilisateur via l'interface Labview. Trois LEDs différentes sont ici utilisées, respectivement à 385 nm, 405nm et 420 nm. La lumière des LEDs est acheminée vers la sonde à travers un faisceau de fibre optique. La sonde, appliquée sur la peau, excite le tissu, et la fluorescence émise est captée par cette même sonde. Le signal de fluorescence passe à travers un filtre passe-haut de longueur d'onde de coupure 485 nm avant d'être collecté par un spectromètre, qui transmet les données à l'ordinateur. L'ordinateur sert d'interface homme/machine, contrôlant l'ensemble du système via le logiciel Labview. L'interface utilisateur permet de définir plusieurs paramètres d'acquisition dont le nom du fichier à enregistrer. En lançant l'acquisition, l'interface affiche le déroulement du programme, avec notamment le numéro de l'acquisition en cours, l'état des LEDs, les spectres récemment acquis, et la vérification des paramètres non modifiables. Toutes les données de spectroscopie sont ensuite enregistrées dans un fichier *.txt*. Les données, anonymisées et référencées par des numéros, sont ensuite chargées dans la base de données de l'unité de recherche en imagerie médicale Creatis.

On applique au patient la crème 5-ALA, il prend ensuite un repos de quelques heures pour permettre une absorption optimale par la peau et la métabolisation de protoporphyrine IX (PpIX).

À son retour, il est placé dans une salle où une nouvelle mesure (*Condition 1*) est faite. Comme illustré sur la Figure [1], une source lumineuse avec une longueur d'onde de 635 nm est utilisée pour éclairer la zone traitée. Tout au long de l'exposition du patient à cette source, de nouvelles mesures successives sont réalisées pour suivre attentivement l'évolution de la quantité de PpIX.

3.2 Traitement des données

Cette partie est consacrée à l'analyse approfondie des données de spectroscopie obtenues. Cette analyse se déploie à travers plusieurs étapes, débutant par le post-traitement essentiel pour extraire le spectre émis par la PpIX à partir des données mesurées en séances de photothérapie dynamique (PDT).

Pour chaque mesure effectuée, trois fichiers sont enregistrés dans le dossier :

- Paramètres acquisition (*_param_acq*) : contient les informations suivantes : compteur_on_off, compteur_acquisitions, Numero_LED, Etat_LED_385, Etat_LED_405, Etat_LED_420, Valeur_envoyee, et Temps_dexecution
- Paramètres initialisation (*_param_init*) : renferme quant à lui les informations suivantes : T_Maya, Tdelai, nb_acq_85, nb_acq_405, nb_acq_420, intensite385, intensite405, intensite420, et durée totale enr(s)
- Un fichier Matlab

Pour le traitement, nous avons réalisé un code Python que vous trouverez en annexe. Il est constitué de plusieurs fonctions :

- **readFluoFile(chemin_signal)** : Contrairement aux codes suivants, cette fonction ne traite les données que d'une mesure à la fois. Elle charge dans un premier temps les données spectrales depuis un fichier Matlab ('lambda.mat'). Elle récupère les chemins des fichiers de paramètre d'acquisition et d'initialisation à partir du chemin du signal fourni. Ensuite, elle lit les données brutes d'acquisition et d'initialisation depuis ses fichiers texte présents dans le répertoire, et extrait le signal brut de leur fichier binaire. Le signal est traité en transposant et en extrayant les informations pertinentes. Pendant cette étape on soustrait au spectre étudié le spectre du fond lumineux. Enfin, le nombre d'acquisitions est déterminé, et le signal final est assigné à la variable 'S'. On identifie ensuite les indices des acquisitions pour chaque longueur d'onde (385 nm, 405 nm, 420 nm) où la LED correspondante est allumée. La fonction vérifie ensuite si le nombre d'acquisitions expérimentales correspond au nombre d'acquisitions commandées, et affiche un avertissement en cas de divergence. Enfin, elle renvoie les longueurs d'onde ainsi que les différences successives des signaux correspondants, prêtes à être utilisées pour l'analyse.
- **plot_curves(chemin_CHUSE_Dermato)** : Cette fonction Python effectue le traitement de toutes les données pour un patient spécifique. Ces données sont dans le dossier ayant pour adresse dans le répertoire : chemin_CHUSE_Dermato. Elle guide l'utilisateur dans le choix d'un patient et d'un échantillon, puis explore les différentes conditions associées à cet échantillon. Pour chaque condition, elle charge les données de fluorescence à trois longueurs d'onde spécifiques (S385, S405, S420). On obtient par exemple la courbe ci-dessous (Figure 7) pour le cas Patient 1/Echantillon 1 de notre base de données, avec l'affichage des trois longueurs d'ondes pour toutes les conditions mesurées :

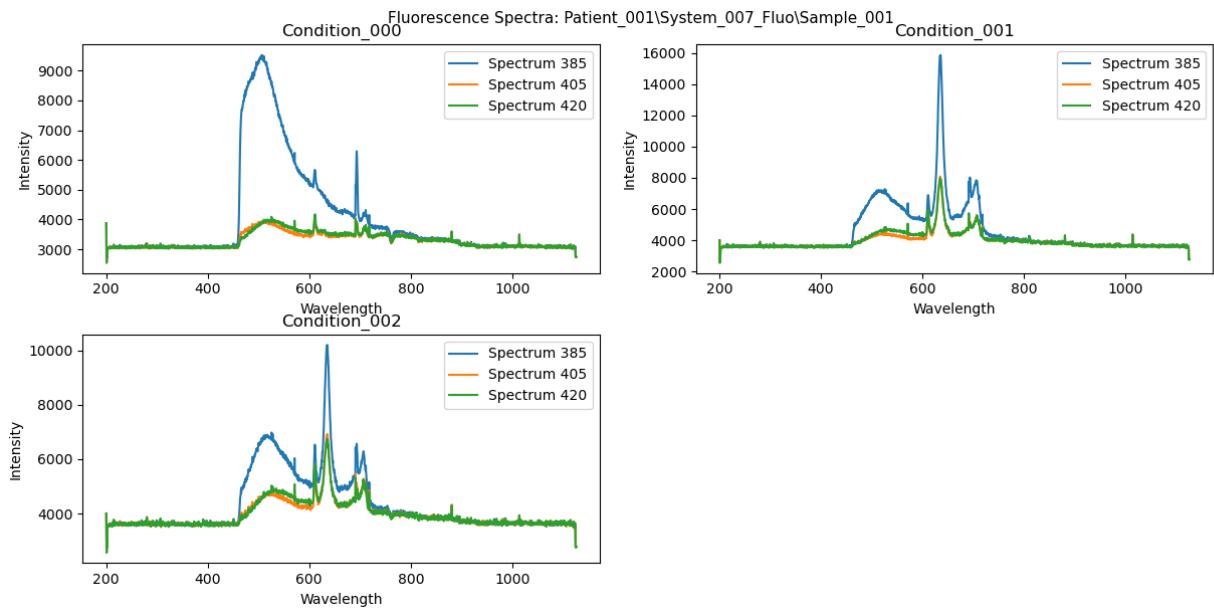


FIGURE 7 – Courbes des données brutes pour les 3 longueurs d’ondes et pour chaque condition
(Patient 1/Echantillon 1)

En isolant une de ces courbes on peut observer plusieurs phénomènes intéressants. Sur la Figure[8], on peut noter qu’on a dans un premier temps une exponentielle décroissante, puis un pic central ici autour de 600 nm. L’exponentielle en question est dûe au spectre d’autofluorescence du corps, que l’on va ici chercher à supprimer afin de quantifier uniquement le pic central qui correspond au vrai spectre de fluorescence causé par le PpIX.

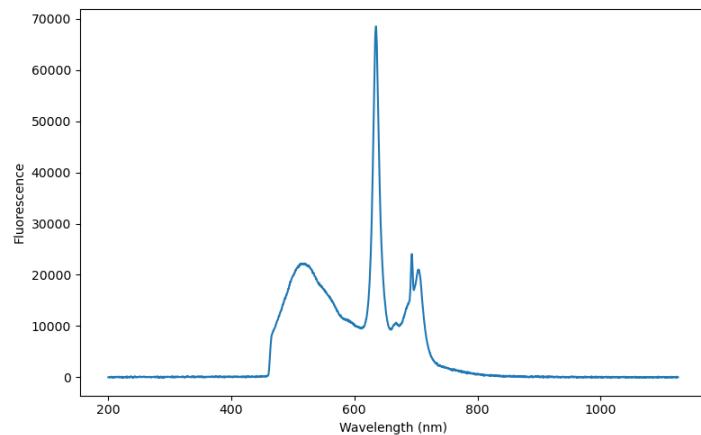


FIGURE 8 – Exemple isolé de courbes des données brutes
(Patient 1/Echantillon 1/Condition 1)

Par la suite la fonction **plot_curves** effectue un ajustement exponentiel sur ces données, utilisant une fonction exponentielle définie, avec les paramètres obtenus par la méthode des moindres carrés (Figure[9]). Cette méthode nous permet d'approcher la courbe d'autofluorescence de la peau par l'équation :

$$f(x) = a \cdot \exp(-b \cdot (x - x_0)) \quad (1)$$

avec a , b et x_0 des constantes déterminées par la fonction, x la longueur d'onde et $f(x)$ le point associé à la modélisation du spectre d'autofluorescence.

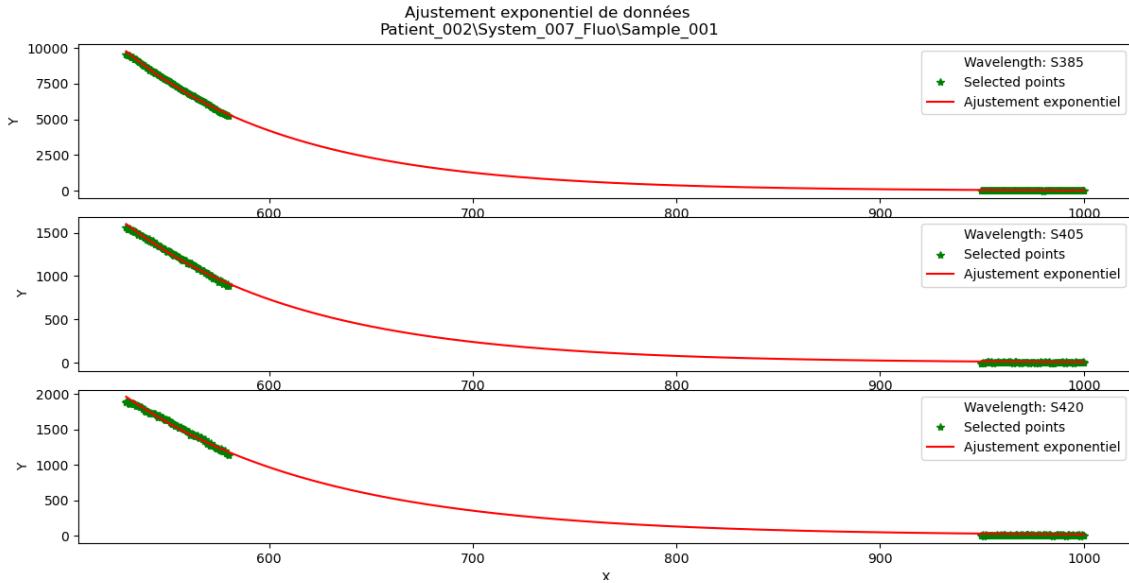


FIGURE 9 – Exemple d'ajustement de la fonction exponentielle du spectre
(Patient 1/Echantillon 1)

À partir des données originales et de l'ajustement exponentiel, la fonction génère un signal corrigé en soustrayant le signal ajusté du signal total. Les résultats sont affichés graphiquement pour chaque condition, mettant en évidence l'ajustement exponentiel et le signal corrigé (Figure[9]). Le processus est répété pour toutes les conditions de l'échantillon. En fin de compte, la fonction retourne une liste de paramètres liés à l'ajustement exponentiel. Ces données peuvent être utiles pour améliorer la modélisation puisque la fonction **plot_curves** utilise des valeurs estimées de a , b et x_0 que nous lui avons fournies au préalable pour réaliser sa modélisation.

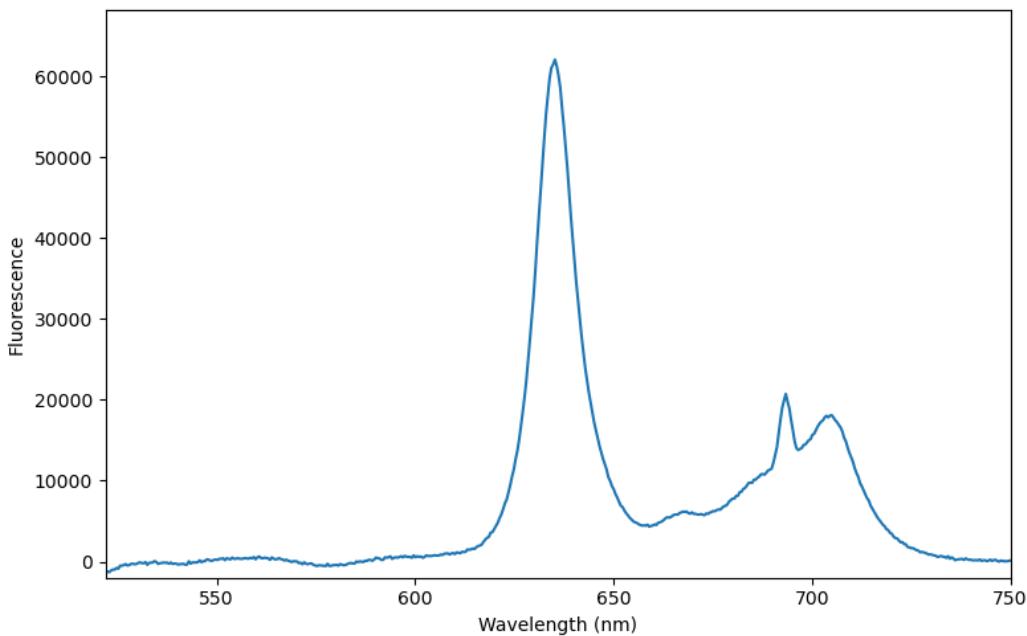


FIGURE 10 – Graphiques des données traitées avec la fonction `plot_curves` pour différentes conditions
(Patient 1/Echantillon 1/Condition 1)

- **IHM(chemin_CHUSE_Dermato)** : L'utilisateur est encore une fois invité à saisir le numéro du patient et de l'échantillon à étudier. Les chemins des fichiers correspondants sont construits en utilisant ces numéros, puis la fonction lit les données de fluorescence à partir des fichiers DAT. Pour chaque condition de l'échantillon, un ajustement exponentiel est appliqué aux données de fluorescence pour chaque longueur d'onde (S_{385} , S_{405} , S_{420}). L'objectif est d'utiliser la fonction `plot_curves` afin de corriger les signaux en éliminant une composante exponentielle indésirable et d'obtenir un signal corrigé. Ensuite, le signal corrigé est obtenu en soustrayant l'ajustement exponentiel du signal total pour chaque longueur d'onde. Les intégrales des signaux corrigés à une longueur d'onde spécifique (S_{405}) sont calculées et stockées dans une liste (`list_integ`). Ces intégrales servent de mesure relative de l'activité fluorescente. Si la durée totale des mesures entre deux conditions dépasse 30 minutes, les intégrales sont tracées en fonction du temps en laissant un écart vide afin de marquer la grande intervalle de temps. Chaque longueur d'onde est affichée séparément sur deux sous-graphiques. Si la durée totale est inférieure à 30 minutes, un seul graphique est généré avec toutes les intégrales en fonction du temps. Les graphiques obtenus (voir Figure[11]) présentent les résultats de la spectroscopie par fluorescence pour chaque condition de l'échantillon, fournissant une visualisation détaillée de l'évolution temporelle des intégrales. Cette représentation visuelle peut aider à identifier des tendances ou des variations significatives dans l'activité fluorescente au fil du temps.

Spectroscopie par fluorescence: Patient_001\System_007_Fluo\Sample_001

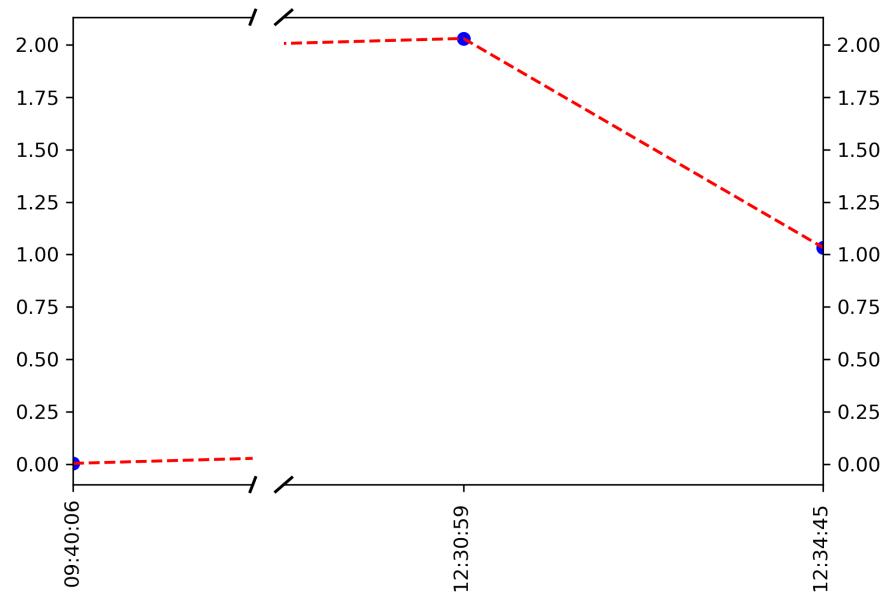


FIGURE 11 – Courbe d'évolution du taux de fluorescence en fonction du temps
(Patient 1/Echantillon 1)

Pour améliorer et faciliter le traitement des données nous conseillons une compartimentation du programme informatique en plusieurs fonctions. Cette méthode permet de structurer le code informatique et facilite sa modification.

4 Analyse critique des résultats

Le succès global de la PDT combinée à la PpIX dans le traitement de diverses pathologies cutanées est indéniable. Pour la Figure[12a] par exemple, on voit nettement que le taux de fluorescence, qui vaut au début autour de 0, est maximale au début de l'éclairement par la lumière. Au fur et à mesure des acquisitions, on observe de moins en moins de PpIX par fluorescence, son émission allant jusqu'à quasiment disparaître. Nous pouvons donc supposer que la quasi-totalité de la PpIX métabolisée a oxydé les cellules où elle était présente ce qui a mené à sa disparition.

Cependant, certaines observations suggèrent d'approfondir certains aspects du traitement. En effet, on remarque une inefficacité potentielle pendant certains traitements. Parmi les facteurs possibles pour expliquer l'inefficacité observer chez certains patients, on peut dans un premier temps penser à une application incorrecte de la crème photosensibilisante sur la peau. Des erreurs dans la procédure d'application pourraient expliquer la concentration inattendue de PpIX dans la zone ciblée.

Un autre point crucial à considérer est l'application de la crème sur une croûte de plaie. Cette pratique pourrait entraver la pénétration appropriée de la PpIX dans la peau ou la diffusion de la lumière sur la PpIX, limitant ainsi son interaction avec les tissus cibles. Une de nos recommandations est donc d'enlever la croûte afin de favoriser l'absorption maximale et donc l'efficacité du traitement. Il ne faut pas non plus écarter une erreur sur la prise de la mesure puisque la sonde peut être un peu écarté de la zone traitée. Il est à noter que certains de nos résultats ne sont que partiellement exploitable par manque de données pour obtenir une évolution complète du taux de PpIX.

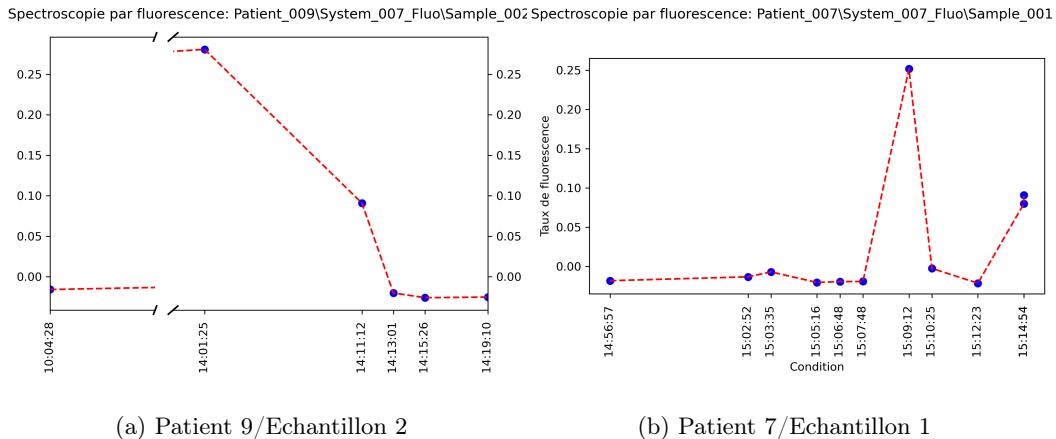


FIGURE 12 – Exemples supplémentaires d'évolution du taux de fluorescence en fonction du temps

5 Conclusion

Pour conclure les propriétés uniques de la PpIX nous ont permis de réaliser un suivi par spectroscopie de fluorescence du processus de photothérapie dynamique. La plupart de nos résultats ont corroboré l'efficacité de la photothérapie dynamique cependant certains tendent à montrer des difficultés dont on peut présumer quelques causes comme la présence de croûte sur la peau traitée, une mauvaise application de la crème 5-ALA ou encore la prise d'un mauvais point de mesure sur la peau. Toutefois la spectroscopie de fluorescence s'est révélée être un outil efficace pour réaliser un suivi de la PDT. Une piste d'amélioration serait la réalisation de ce traitement de données en temps réel pour pouvoir informer le médecin sur le déroulement du traitement. Un autre axe amélioration pour le suivi de la photothérapie dynamique serait une étude sur son efficacité en profondeur dans la peau en mesurant la présence de PpIX en profondeur.

6 Références bibliographiques

- Laure Alston. *Spectroscopie de fluorescence et imagerie optique pour l'assistance à la résection de gliomes : conception et caractérisation des systèmes de mesure et modèles de traitement des données associées, sur fantômes et au bloc opératoire* - Imagerie. Université Claude Bernard Lyon 1, 2017. Français. NNT : . tel-01797625
- *Spectroscopie de fluorescence de la protoporphyrine IX induite par 5-ALA pour l'assistance préopératoire en neurochirurgie*, L.Alston, L. Mahieu-Willame, M. Hébert, J. Guyotat, D. Rousseau, B. Montcel, <https://hal.science/hal-01451704>
- Shi L, Liu P, Liu J, et al. *Application of 5-aminolevulinic acid photodynamic therapy in common skin diseases* Translational Biophotonics. 2020;2:e201900028. <https://doi.org/10.1002/tbio.201900028>
- *Fluorescence Spectroscopy Study of Protoporphyrin IX in Optical Tissue Simulating Liquid Phantoms*, National Center for Biotechnology Information, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7254220/>
- *Photothérapie dynamique* Mitchel P. Goldman, Coordination scientifique de l'édition française, Nicole Basset-Seguin, Christophe Bédane
- *Actinic Keratosis : Current Therapies and Insights Into New Treatments*, <https://jddonline.com/articles/actinic-keratosis-current-therapies-and-insights-into-new-treatments-S1545961619S>
- *The Global Epidemiology of Actinic Keratosis in the General Population : A Systematic Review and Meta-Analysis* <https://pubmed.ncbi.nlm.nih.gov/37890083/>
- *Prevalence of actinic keratosis among dermatology outpatients in Spain* <https://www.actasdermo.org/en-prevalence-actinic-keratosis-among-dermatology-articulo-S1578219016302037>
- Kératoses actiniques, <https://www.msdmanuals.com/fr/accueil/troubles-cutan%C3%A9s-1%C3%A9sions-cutan%C3%A9es-dues-au-soleil/k%C3%A9ratoses-actiniques>

7 Annexe

Cette annexe présente la version finale de notre code utilisé pour traiter les données brutes des patients.

```
1 # -*- coding: utf-8 -*-
2
3 from datetime import *
4
5 import matplotlib
6 matplotlib.use('TKAgg')
7 from scipy.signal import argrelextrema
8
9 import scipy.io
10 import pandas as pd
11 import numpy as np
12 import matplotlib.pyplot as plt
13 import matplotlib
14 import os
15 import sys
16
17 from scipy.optimize import curve_fit
18 from lmfit import Model
19
20
21 # matplotlib.use('TKAgg')
22
23 def readFluoFile(chemin_signal):
24     lambda_data = scipy.io.loadmat('./lambda.mat')
25     wl = lambda_data['lambda']
26     chemin_param_acq = chemin_signal.replace('.dat', '_param_acq.txt')
27     chemin_param_init = chemin_signal.replace('.dat', '_param_init.txt')
28     donnees_acq_brut = np.genfromtxt(chemin_param_acq, dtype=None, encoding='None', delimiter='\t', skip_header=1)
29     donnees_init_brut = np.genfromtxt(chemin_param_init, dtype=None, encoding='None', delimiter='\t', skip_header=1)
30     with open(chemin_signal, 'rb') as file:
31         signal_init = np.fromfile(file, dtype=np.uint16).astype('float')
32         signal_init = signal_init.reshape(wl.shape[1], -1, order='F')
33         signal_brut = signal_init.T
34         signal_final = signal_brut[2:, :]
35         LEDallumee = donnees_acq_brut[2:, 3:6]
36         nb_acquisition = signal_brut.shape[0]
37
38     S = signal_final
39
40     idx385 = np.where(LEDallumee[:, 0] == 1)[0]
41     idx405 = np.where(LEDallumee[:, 1] == 1)[0]
42     idx420 = np.where(LEDallumee[:, 2] == 1)[0]
43
44     nb_acq_commande = donnees_init_brut[1, 2:5]
```

```

45     nb_acq_commande_tot = donnees_acq_brut.shape[0]
46     nb_acq_exp = [S[idx385, :].shape[0], S[idx405, :].shape[0], S[idx420,
47     :].shape[0]]
48     if np.any(np.array(nb_acq_exp, dtype=float) != np.array(nb_acq_commande,
49     , dtype=float)):
50         print('Warning: Number of experimental acquisitions does not
51 correspond to what is required!')
52     else:
53         print('Experimental Acquisition Number OK')
54
55     if nb_acquisition != nb_acq_commande_tot:
56         print('Number of acquisitions in the .dat file does not match the
57 number in the .txt file')
58
59     return wl.T, np.subtract(S[idx385, :], S[idx385 + 1, :]).T, np.subtract(
60     S[idx405, :],
61     S[idx405 + 1, :]).T, np.subtract(
62     S[idx420, :], S[idx420 + 1, :]).T
63
64
65
66
67
68 def numNormalize(taille, num):
69     num = str(num)
70     length = len(num)
71     c = taille - length
72     for i in range(c):
73         num = "0" + num
74     return num
75
76
77 def slice_indices(L, a, b):
78     x=0
79     y=len(L)-1
80     for i in range(len(L)):
81         if L[i]<a and L[i]>L[x]:
82             x=i
83         if L[i]>b and L[i]<L[y]:
84             y=i
85     return x,y
86
87
88 def acqui_time(chemin_dossier):
89     chemin_fichier = chemin_dossier + '\\\\AcqTimes.txt'
90     if os.path.exists(chemin_fichier):
91         print('oui')
92         t = []
93         min = []
94         hour = []
95         echan = [x.split('\\t')[0] for x in open(chemin_fichier).readlines()
96     ]
97         time = [x.split('\\t')[1] for x in open(chemin_fichier).readlines()]
98         for elt in time:
99

```

```

88         elt2 = elt.replace('\n', '')
89         t.append(elt2)
90         hours, minutes, seconds = map(int, elt2.split(':'))
91         hour.append(hours)
92         min.append(minutes)
93     return echan, t, hour, min
94 else:
95     print('non')
96
97 def fonction_exponentielle(x, a, b, x0):
98     arr = a * np.exp(-b*(x-x0))
99     return arr
100
101 def IHM(chemin_CHUSE_Dermato):
102     list_integ = []
103     t = []
104     nombre_patients = len(os.listdir(chemin_CHUSE_Dermato))
105     num_patient = input("Entrez le numéro du patient: ")
106     if int(num_patient) > int(nombre_patients):
107         print("Ce patient n'est pas répertorié")
108         sys.exit()
109     num_patient = numNormalize(3, num_patient)
110     print("Traitement du patient ", num_patient, " en cours...")
111     chemin_echantillons = chemin_CHUSE_Dermato + '\Patient_' + num_patient
112     + '\System_007_Fluo'
113     nombre_echantillons = len(os.listdir(chemin_echantillons))
114     print("Le nombre d'échantillons pour le patient_" + num_patient + " est "
115     + f"{nombre_echantillons}")
116     num_echantillon = input("Choisissez le numéro de l'échantillon à étudier: ")
117     if int(num_echantillon) > int(nombre_echantillons):
118         print("Cet échantillon n'est pas répertorié")
119         sys.exit()
120     num_echantillon = numNormalize(3, num_echantillon)
121     chemin_dossier = chemin_CHUSE_Dermato + '\Patient_' + num_patient + \
122     '\System_007_Fluo\Sample_' + num_echantillon
123     nombre_conditions = len([f for f in os.listdir(chemin_dossier) if not f
124     .endswith('.txt')])
125     print("Le nombre de conditions pour le patient_" + num_patient + " est "
126     + f"{nombre_conditions}")
127     i = 0
128     while i < nombre_conditions:
129         if i < 10:
130             chemin_signal = chemin_CHUSE_Dermato + '\Patient_' +
131             num_patient + '\System_007_Fluo\Sample_' + num_echantillon + \
132             'Condition_00' + str(i) + '\eCHAN_' + num_echantillon[1:] + '_00' + str(i)
133             + '.dat'
134         else:
135             chemin_signal = chemin_CHUSE_Dermato + '\Patient_' +
136             num_patient + '\System_007_Fluo\Sample_' + num_echantillon + \

```

```

Condition_0' + str(i) + '\echan_' + num_echantillon[1:] + '_0' + str(i)
+ '.dat'
    if os.path.exists(chemin_signal):
        t.append(i)
        wavelength, S385, S405, S420 = readFluoFile(chemin_signal)
        wave_data = {'S385': S385, 'S405': S405, 'S420': S420}
        figure, axs = plt.subplots(nrows=3, ncols=1, figsize=(8, 18))
        for var_name, ax in zip(wave_data.keys(), axs):
            wave = wave_data[var_name]
            a1, b1 = slice_indices(wavelength, 530, 580)
            xs1, ys1 = wavelength[a1:b1 + 1], wave[a1:b1 + 1]
            a2, b2 = slice_indices(wavelength, 950, 1000)
            xs2, ys2 = wavelength[a2:b2 + 1], wave[a2:b2 + 1]
            xdata = np.concatenate((xs1, xs2))
            xdata = xdata.reshape((len(xdata), ))
            ydata = np.mean(np.concatenate((ys1, ys2)), axis=1)
            ydata = ydata.reshape((len(ydata), ))
            a_guess, b_guess, x0_guess = 10000, 0.001, 510
            parametres, covariance = curve_fit(fonction_exponentielle,
xdata, ydata, p0=[a_guess, b_guess, x0_guess], bounds=[[0, -np.inf,
400], [np.inf, np.inf, 600]])
            a, b, c = parametres
            y_fit = fonction_exponentielle(wavelength.T, a, b, c)
            signal = (np.subtract(np.mean(wave, axis=1), y_fit).T)/np.
mean(wave, axis=1)[a1]
            if var_name == 'S405':
                index_wave = 635
                a, b = slice_indices(wavelength, index_wave, index_wave
)
                arr1, arr2 = (signal[a:b+1]).reshape((len(signal[a:b
+1]),)), (wavelength[a:b+1]).reshape((len(wavelength[a:b+1]),))
                integ_val = np.trapz(arr1, arr2)
                list_integ.append(integ_val)
            plt.close()
        else:
            nombre_conditions += 1
        i = int(i)
        i += 1
    try:
        echan, t, hour, min = acqui_time(chemin_dossier)
        time = [hour[i]*60+min[i]-(hour[0]*60+min[0]) for i in range(len(
echan))]
        title = ''
        eIndex = len(chemin_dossier)
        sIndex = chemin_dossier.index("Patient")
        length = eIndex - sIndex
        if 0 < length:
            title = chemin_dossier[sIndex:sIndex + length]
        if time[1]-time[0] > 30:

```

```

169         f, (ax, ax2) = plt.subplots(1, 2, sharey=True, facecolor='w',
170         gridspec_kw={'width_ratios': [1, 3]})  

171         ax.plot(time, list_integ, 'bo', time, list_integ, 'r--')  

172         ax2.plot(time, list_integ, 'bo', time, list_integ, 'r--')  

173         ax.set_xticks(time)  

174         ax.set_xticklabels(t, rotation='vertical')  

175         ax2.set_xticklabels(t, rotation='vertical')  

176         ax.set_xlim(0, time[0]+2)  

177         ax2.set_xlim(time[1]-2, time[-1]+2)  

178         # hide the spines between ax and ax2  

179         ax.spines['right'].set_visible(False)  

180         ax2.spines['left'].set_visible(False)  

181         ax.yaxis.tick_left()  

182         ax.tick_params(right=False) # Remove right y-axis ticks and  

183         labels for the left axis  

184         ax2.yaxis.tick_right()  

185         # ax2.tick_params(left=False) # Remove left y-axis ticks and  

186         labels for the right axis  

187         ax2.tick_params(labelright=True, labelleft=False, right=True,  

188         left=False)  

189         # ax.yaxis.tick_left()  

190         # ax.tick_params(labelright='off')  

191         # ax2.yaxis.tick_right()  

192         d = .015 # how big to make the diagonal lines in axes  

193         coordinates  

194         kwargs = dict(transform=ax.transAxes, color='k', clip_on=False)  

195         ax.plot((1-d, 1+d), (-d, +d), **kwargs)  

196         ax.plot((1-d, 1+d), (1-d, 1+d), **kwargs)  

197         kwargs.update(transform=ax2.transAxes) # switch to the bottom  

198         axes  

199         ax2.plot((-d, +d), (1-d, 1+d), **kwargs)  

200         ax2.plot((-d, +d), (-d, +d), **kwargs)  

201         f.tight_layout(rect=[0, 0.03, 1, 0.95])  

202         plt.suptitle('Spectroscopie par fluorescence: '+str(title))  

203         # save_path = r'C:\Users\TD273339\Desktop\SupOp\TP Projet\  

204         Images\Patient_\' + num_patient + ' Echantillon_\' + num_echantillon + '.  

205         png'  

206         # plt.savefig(save_path, dpi=300, format='png')  

207         plt.show()  

208     else:  

209         plt.figure()  

210         plt.plot(time, list_integ, 'bo', time, list_integ, 'r--')  

211         plt.suptitle('Spectroscopie par fluorescence: '+str(title))  

212         plt.xticks(time,t, rotation='vertical')  

213         plt.ylabel('Taux de fluorescence')  

214         plt.xlabel('Condition')  

215         plt.tight_layout(rect=[0, 0.03, 1, 0.95])  

216         # save_path = r'C:\Users\TD273339\Desktop\SupOp\TP Projet\  

217         Images\Patient_\' + num_patient + ' Echantillon_\' + num_echantillon + '.  


```

```

210     png'
211         # plt.savefig(save_path, dpi=300, format='png')
212         plt.show()
213
214     except Exception as e:
215         print(e)
216
217 def plot_curves(chemin_CHUSE_Dermato):
218     list_param = []
219     list_integ = []
220     max_wave = []
221
222     nombre_patients = len(os.listdir(chemin_CHUSE_Dermato))
223     num_patient = input("Entrez le numéro du patient: ")
224     # num_patient = 8
225     if int(num_patient) > int(nombre_patients):
226         print("Ce patient n'est pas répertorié")
227         sys.exit()
228     num_patient = numNormalize(3, num_patient)
229     print("Traitement du patient ", num_patient, " en cours...")
230     chemin_echantillons = chemin_CHUSE_Dermato + '\Patient_' + num_patient
231     + '\System_007_Fluo'
232     nombre_echantillons = len(os.listdir(chemin_echantillons))
233     print("Le nombre d'échantillons pour le patient_" + num_patient + " est "
234     : " + f"{nombre_echantillons}")
235     num_echantillon = input("Choisissez le numéro de l'échantillon à étudier: ")
236     # num_echantillon = 1
237     if int(num_echantillon) > int(nombre_echantillons):
238         print("Cet échantillon n'est pas répertorié")
239         sys.exit()
240     num_echantillon = numNormalize(3, num_echantillon)
241     chemin_dossier = chemin_CHUSE_Dermato + '\Patient_' + num_patient + \
242     '\System_007_Fluo\Sample_' + num_echantillon
243     nombre_conditions = len(os.listdir(chemin_dossier))
244     i = 0
245     while i < nombre_conditions:
246         chemin_signal = chemin_CHUSE_Dermato + '\Patient_' + num_patient +
247         '\System_007_Fluo\Sample_' + num_echantillon + '\Condition_00' + str(i)
248         + '\echan_' + num_echantillon[1:] + '_00' + str(i) + '.dat'
249         if os.path.exists(chemin_signal):
250             temp = []
251             wavelength, S385, S405, S420 = readFluoFile(chemin_signal)
252             wave_data = {'S385': S385, 'S405': S405, 'S420': S420}
253             figure, axs = plt.subplots(nrows=3, ncols=1, figsize=(8, 18))
254             title = ''
255             eIndex = chemin_signal.index("echan") - 1
256             sIndex = chemin_signal.index("Patient")
257             length = eIndex - sIndex
258             if 0 < length:
259                 title = chemin_signal[sIndex:sIndex + length]

```

```

253     for var_name, ax in zip(wave_data.keys(), axs):
254         wave = wave_data[var_name]
255         a1, b1 = slice_indices(wavelength, 530, 580)
256         xs1, ys1 = wavelength[a1:b1 + 1], wave[a1:b1 + 1]
257         a2, b2 = slice_indices(wavelength, 950, 1000)
258         xs2, ys2 = wavelength[a2:b2 + 1], wave[a2:b2 + 1]
259         xdata = np.concatenate((xs1, xs2))
260         xdata = xdata.reshape((len(xdata), ))
261         ydata = np.concatenate((ys1, ys2)), axis=1
262         ydata = ydata.reshape((len(ydata), ))
263         a_guess, b_guess, x0_guess = 10000, 0.001, 510
264         parametres, covariance = curve_fit(fonction_exponentielle,
265         xdata, ydata, p0=[a_guess, b_guess, x0_guess], bounds=( [0, -np.inf,
266             400], [np.inf, np.inf, 600]))
267         a, b, c = parametres
268         y_fit = fonction_exponentielle(wavelength.T, a, b, c)
269         signal = np.subtract(np.sum(wave, axis=1), np.multiply(
270             y_fit, wave.shape[1])).T
271         ax.plot(wavelength, signal)
272         ax.set_xlabel('X')
273         ax.set_ylabel('Y')
274         ax.set_xlim(left=522, right=750)
275         ax.set_ylim(bottom=-2000, top= 1.1*np.max(signal))
276         figure.suptitle('Ajustement exponentiel de données \n'+str(
277             title))
278         figure.tight_layout(rect=[0, 0.03, 1, 0.95])
279         plt.show()
280     return list_param
281
282
283
284 if __name__ == '__main__':
285     chemin_CHUSE_Dermato = r'C:\Users\TD273339\Desktop\SupOp\TP Projet\
286         Acquisition\CHUSE_Dermato'
287 #liste = np.array(plot_curves(chemin_CHUSE_Dermato))
288 liste = np.array(IHM(chemin_CHUSE_Dermato))

```