
NORMALISATION DE SPECTRE DE QUASARS À GRAND REDSHIFT

EVRARD ANTONIN

Résumé

Les quasars font partie des objets les plus lumineux de l'univers. Leur étude permet de sonder les propriétés de l'univers à des temps cosmologiques. Au cours de ce stage nous étudions le spectre des quasars de la banque de donnée Sloan Digital Sky Survey Data Release 16. Pour étudier les raies d'absorption produites par les objets situés sur la ligne de visée, nous devons séparer l'émission de l'absorption, c'est-à-dire normaliser le spectre. Nous approfondissons trois méthodes de normalisation de spectre. La première est l'ajustement du spectre avec une loi de puissance pour l'émission continue et une gaussienne par raie d'émission. La deuxième est une version améliorée de cet ajustement basée sur un code python développé par Q. Wu et Y. Shen, PyQSOFit. La troisième méthode est un lissage du spectre. Nous comparons ces méthodes pour dégager leurs avantages et inconvénients respectifs ainsi que des pistes d'optimisation.

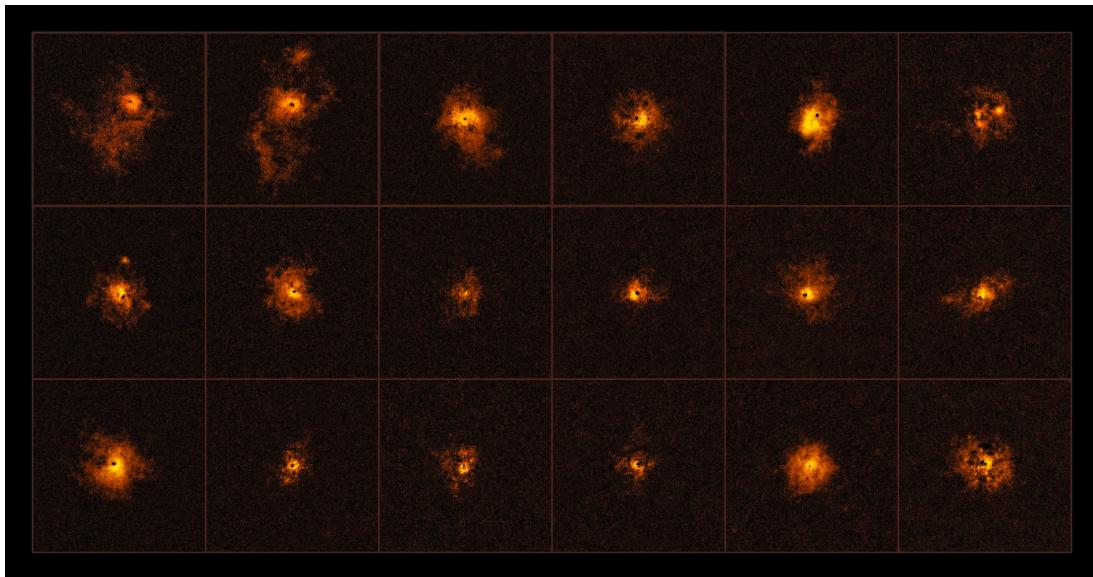


Figure 1: Mosaïque de 18 quasars entourés d'un brillant halo gazeux.

Crédit : [ESA/Hubble & Nasa](#)

Sommaire

1	Introduction	2
1.1	Présentation	2
1.2	Quasar et spectroscopie	2
1.2.1	Émission	3
1.2.2	Absorption	3
1.3	Sloan Digital Sky Survey	5
1.3.1	Télescope SDSS	5
1.3.2	Récupération des données	5
1.4	Problématique	6
2	Normalisation de spectre	6
2.1	Ajustement de spectre	6
2.2	PyQSOFit.	7
2.3	Smoothing	9
2.4	Discussion	10
3	Conclusion	11
4	Bibliographie	12
5	Annexe A : Code	13
5.1	Sélection dans le catalogue	13
5.2	Récupération du spectre SDSS	16
5.3	Récupération du fit.	16
5.4	Reconstruction du fit de Q. Wu et Y. Shen.	16
5.5	Smoothing.	17

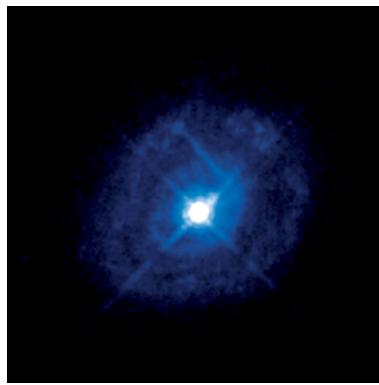
1 Introduction

1.1 Présentation

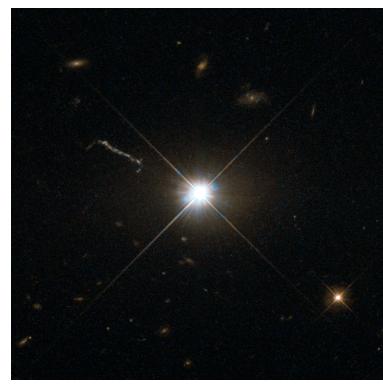
Notre univers observable, centré sur la Terre, abriterait quelques centaines de milliards de galaxies. Parmi celles-ci environ 2% sont considérées comme actives du fait de leurs extrêmes luminosités.

Ces galaxies actives émettent la grande majorité de leur luminosité depuis une petite région centrale que l'on nomme noyau actif de galaxie (AGN : *Active Galactic Nucleus*). Cette région centrale dont la taille varie entre un à une dizaine de parsec, émet sur l'ensemble du spectre électromagnétique une luminosité environ mille fois supérieure à celle de la galaxie hôte(Cf. Figure 2.a). Cette énergie électromagnétique n'est pas d'origine stellaire mais est due à l'accrétion de matière par un trou noir supermassif (SMBH : *Super Massive Black Hole*) dont la masse est de l'ordre de 100 millions de masses solaires.

Cette accrétion à la fois massive et violente, générée par le puits gravitationnel du SMBH, fait spiraler la matière jusqu'à des vitesses proches de 5000 km/s. La matière accrétiée étant soumise à d'énormes forces gravitationnelles et de friction réemet une partie de son énergie sous forme d'énergie électromagnétique. Les AGNs convertissent ainsi de 10% à 40% de l'énergie de masse en énergie lumineuse, ce qui est très nettement supérieure au 0.7% du processus de fusion stellaire. Les AGNs les plus lumineux, avec une luminosité typiquement supérieure à 1000 fois celle de la Voie Lactée, sont appelés quasars (QSO : *Quasi Stellar Object*).



2.a AGN Markarian 509
Crédit : [NASA, ESA](#)



2.b Quasar 3C 273
Crédit : [ESA/Hubble & Nasa](#)

Figure 2: Astrophotographie d'un AGN et d'un quasar.

1.2 Quasar et spectroscopie

Les quasars sont présents à tous les redshifts. Toutefois, le pic de leur densité cosmologique est à $z \simeq 2$, ce qui correspond à une distance comobile d'environ 17.5 milliards d'années-lumière et à un temps de regard vers le passé d'environ 10 milliards d'années. Leurs zone d'émission étant relativement petite par rapport à la distance qui les séparent de la Terre les quasars nous apparaissent comme quasiment ponctuels (Cf. Figure 2.b). C'est cette apparente

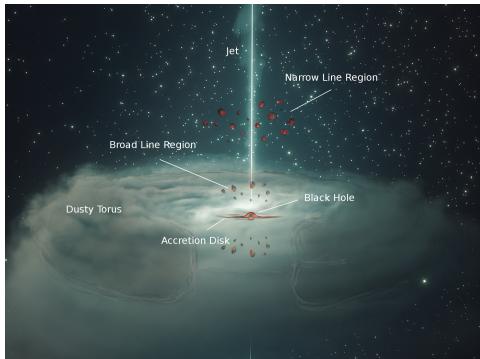
similitude avec les étoiles qui leur a valu le nom d'objet quasi-stellaire. La luminosité des quasars et leur répartition au cours du temps cosmologique en font des sujets d'étude particulièrement pertinents pour comprendre l'évolution de l'univers. Les quasars présentent une grande diversité de spectre. Cette diversité rend compte de l'émission aux alentours du quasar et de l'absorption d'une partie de cette émission par des objets qui peuvent se trouver sur la ligne de visée.

1.2.1 Émission

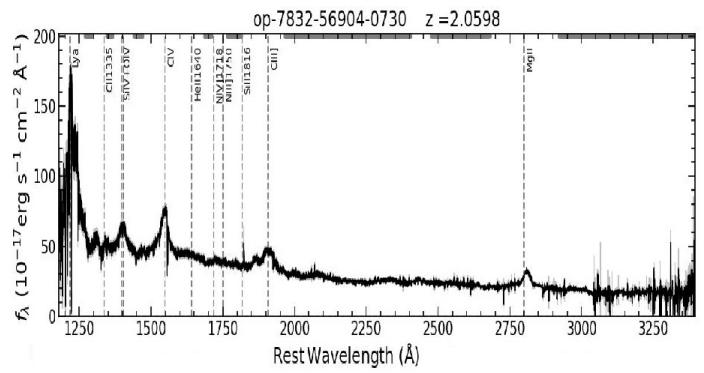
Les émissions se situent aux alentours du quasar et donc au même redshift z_{QSO} ¹. Nous distinguerons 3 zones principales (Cf. Figure 3.a) :

- Le disque d'accrétion (AD : *Accretion Disk*). Dont le rayon varie entre 0.003 à 0.3 année-lumière du SMBH, l'AD émet sur l'ensemble du spectre électromagnétique.
 - La région émettant des raies larges (BLR : *Broad Line Region*). Située de 0.3 à 3 années-lumières du SMBH.
 - La région émettant des raies étroites (NLR : *Narrow Line Region*). Située de 300 à 3000 années-lumière

L'émission de la BLR et NLR produit sur le spectre des raies dont la largeur est due à l'effet Doppler. Leur étude permet d'obtenir des indications sur les caractéristiques de la zone d'émission (Cf. Figure 3.b).



3.a Schéma d'un quasar
Crédit : H E S S



3.b Spectre de 7832-56904-730.

Figure 3: Structure et raies d'émission de quasar.

1.2.2 Absorption

Lors de sa propagation, la lumière émise par le quasar rencontre des nuages de gaz dans le milieu interstellaire de galaxies (ISM : *Inter Stellar Medium*) et dans le milieu intergalactique (IGM : *Inter Galactic Medium*). Cette matière absorbe une partie de la lumière du

¹Le redshift est obtenu avec la relation : $\lambda_{emit}(1 + z) = \lambda_{obs}$
Description de la procédure de détermination avec un exemple appliqué à une galaxie : <http://cas.sdss.org/dr8/en/proj/advanced/hubble/redshifts.asp>.

quasar. Comme pour les raies d'émission, les propriétés des raies d'absorption dépendent des propriétés physico-chimiques du milieu dont elles sont issues. L'absorption peut se situer n'importe où le long de la ligne de visée et donc à n'importe quel redshift tels que $z_{abs} \leq z_{QSO}$. Puisque les quasars sont très lointains, l'ensemble des objets situés entre eux et nous absorbent à des redshift différents. L'IGM est majoritairement composé de nuages d'hydrogène ionisé. Malgré une faible densité d'hydrogène neutre, la taille de ces nuages implique une densité de colonne suffisante pour générer des raies d'absorption. On observe alors une succession de raies d'absorption Lyman alpha (Ly_α) à des redshift inférieur à celui de la raie d'émission du quasar. On appelle cet ensemble de raies la forêt Lyman alpha (Cf. Figure 4). Les nuages de la forêt Ly_α ayant une colonne densité $10^{17} \text{ cm}^{-2} \leq n(\text{HI}) \leq 10^{20} \text{ cm}^{-2}$ sont appelés système Lyman limite (LLS : *Lyman Limit System*). Ceux avec une densité $n(\text{HI}) \geq 10^{20} \text{ cm}^{-2}$ sont appelés systèmes amortis de Lyman (DLA : *Damped Lyman Alpha*). Une telle colonne de densité est similaire à celle des que l'on observerait sur une ligne de visée traversant notre galaxie. Les métaux présents dans l'ISM produisent aussi des raies d'absorption². Ainsi on peut constater sur la Figure 4 que le DLA et la raie CIV sont au même redshift $z \sim 0.83$.

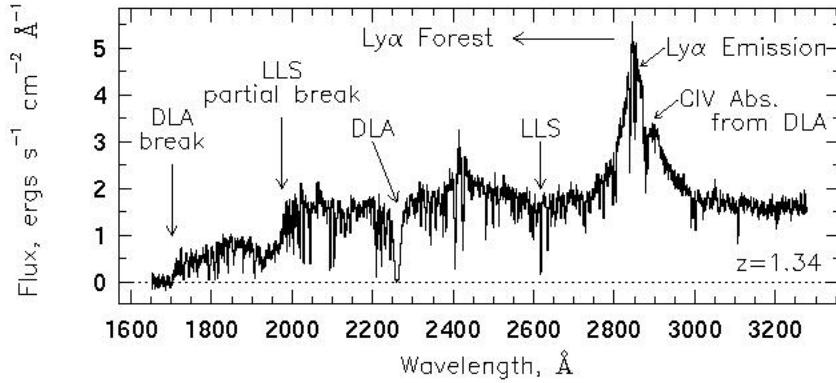


Figure 4: Spectre typique de quasar montrant l'émission continue et les raies d'émission du quasar ainsi que les raies d'absorption produites par le ISM et IGM.

Credit : NASA/IPAC Galactic Database

Ces raies représentent donc un outil puissant permettant de sonder la composition, l'ionisation, la température, la densité ou encore la masse d'objets astronomiques. La précision sur les propriétés physico-chimiques du quasar et du milieu de propagation dépendra fortement de la qualité des données spectrales. Environ un million de quasars ont été identifiés par spectroscopie astronomique. Nous nous restreindrons ici à l'étude spectroscopique des quasars du catalogue SDSS (*Sloan Digital Sky Survey*) qui en a répertorié plus de 750 000.[1]

²Données de références pour les raies d'émission et d'absorption : <https://classic.sdss.org/dr6/algorithms/linestable.php>.

1.3 Sloan Digital Sky Survey

1.3.1 Télescope SDSS

Le Sloan Digital Sky Survey est un programme de relevé d'objets célestes utilisant un télescope situé au Nouveau-Mexique aux États-Unis.



5.a Télescope SDSS
Crédit : [SDSS](#)



5.b Arrière du télescope avec la caméra (noire) et les spectrographes (verts).
Crédit : [SDSS](#)

Figure 5: Télescope SDSS

Issu de la collaboration d'universités internationales et en grande partie financé par la *Sloan Fundation*, SDSS a pour but de cartographier un quart du ciel et de répertorier plus de 100 millions d'objets célestes. Plus particulièrement, le but premier est de déterminer le redshift de galaxies et de quasars pour en déduire la structure à grande échelle de l'univers[2]. Muni d'un miroir principal de 2.5 mètres et d'un miroir secondaire de 1.08 mètres, le télescope est capable de focaliser efficacement des images dans un champ de vue de 3° soit l'équivalent de 20 pleines lunes. Le design, les caractéristiques ainsi que les performances du télescope SDSS sont décrites en détails dans [cet article](#). On notera que les longueurs d'ondes accessibles aux spectrographes se situent dans le domaine optique entre 3 650 et 10 400 Å.

1.3.2 Récupération des données

L'ensemble des quasars observés par SDSS est disponible dans un unique fichier FITS³ (*Flexible Image Transport System*) que l'on appellera le catalogue⁴. Le catalogue contient l'ensemble des caractéristiques clés de chaque quasar[3]. Chaque quasar observé est indexé par sa date d'observation (MJD : *Modified Julian Date*), la plaque (*plate*) et la fibre optique (*fiberID*) qui lui correspond. Ainsi chaque triplet MJD-plate-fiberID correspond à un unique quasar. Les données d'observation sont stockées sur des fichiers FITS individuels. On peut sélectionner les quasars sur le catalogue en fonction de critères tels que le redshift, le rapport signal sur bruit (S/N , *signal to noise*), ou encore la magnitude. Une fois sélectionné on utilise son triplet MJD-plate-fiberID pour pouvoir rechercher son spectre. Les spectres des quasars

³Description du format de fichier : https://fits.gsfc.nasa.gov/fits_primer.html

⁴Description du catalogue : https://data.sdss.org/datamodel/files/BOSS_QSO/DR16Q/DR16Q_v4.html

sont accessibles sur la base de donnée de SDSS DR16 (*Data Release 16*) sous la forme d'un fichier FITS⁵.

1.4 Problématique

L'étude des raies d'absorption implique leur ajustement par un profil gaussien ou de Voigt[4]. Pour ce faire nous avons besoin de séparer la partie émission du spectre de la partie absorption. Cette étape est appelée normalisation⁶. Nous chercherons donc à obtenir la meilleure normalisation possible pour pouvoir ajuster au mieux ces raies d'absorption. La méthode de normalisation devra aussi être automatisable pour pouvoir effectuer une étude statistique sur l'ensemble du catalogue SDSS.

Le sujet de ce stage est : comment normaliser au mieux et de manière automatique un spectre de quasar ?

2 Normalisation de spectre

Pour nous adapter à la convention d'écriture nous appellerons le "continu" du quasar, l'ensemble de son spectre d'émission (Cf. courbe noire [Figure 7.a](#)). Cette courbe continue ne doit pas être confondue avec l'émission continue du AD définie à [1.2.1](#).

Nous allons voir 3 méthodes de normalisation de spectre de quasar. La première permettra d'expliciter le processus d'ajustement (*fit*) de spectre. La deuxième est une version améliorée du simple fit et est basée sur un code python public : PyQSOFit. La troisième est un lissage (*smoothing*) du spectre.

2.1 Ajustement de spectre

Cette technique consiste à faire un fit du spectre du quasar en utilisant un modèle régressif. On teste la faisabilité de cette méthode grâce au logiciel Sculptor⁷. On ajuste d'abord l'émission continue avec une loi de puissance. L'ajustement est fait dans le domaine optique dépourvu de raies d'émission (Cf. [Figure 6.a](#)). Cette courbe est soustraite au spectre pour obtenir un spectre de raies d'émission. On ajuste ensuite sur ce spectre chaque raie avec une seule gaussienne (Cf. [Figure 6.b](#)). Le fit final est obtenu en faisant la somme de ces deux ajustements. (Cf. [Figure 6.c](#)). On peut remarquer que le fit correspond mal au spectre autant dans l'émission continue autour de $\lambda = 1650 \text{ \AA}$ que sur les raies d'émission $\text{CII}_{\lambda=1335 \text{ \AA}}$, $\text{SiIV}_{\lambda=1400 \text{ \AA}}$, $\text{CIV}_{\lambda=1549 \text{ \AA}}$ et $\text{CIII}]_{\lambda=1908 \text{ \AA}}$. Faire un fit des raies d'absorption avec une seule gaussienne semble donc insuffisant. On peut voir que si on fit la raie Ly_α avec 3 gaussiennes

⁵Description du FITS contenant le spectre : https://data.sdss.org/datamodel/files/BOSS_SPECTRO_REDUX/RUN2D/spectra/PLATE4/spec.html

⁶L'absorption d'un nuage "plan-parallèle" homogène, d'épaisseur l et sans émissivité est donnée par : $I(\lambda) = I_0(\lambda) \exp(-nl\sigma(\lambda))$, où n et σ sont respectivement la densité et la section efficace du gaz constituant le nuage. Pour pouvoir remonter aux paramètres l , n , σ du nuage, on doit diviser le spectre observé $I(\lambda)$ par le spectre émit $I_0(\lambda)$.

⁷Tutoriel et documentation : https://sculptor.readthedocs.io/en/latest/first_fit.html

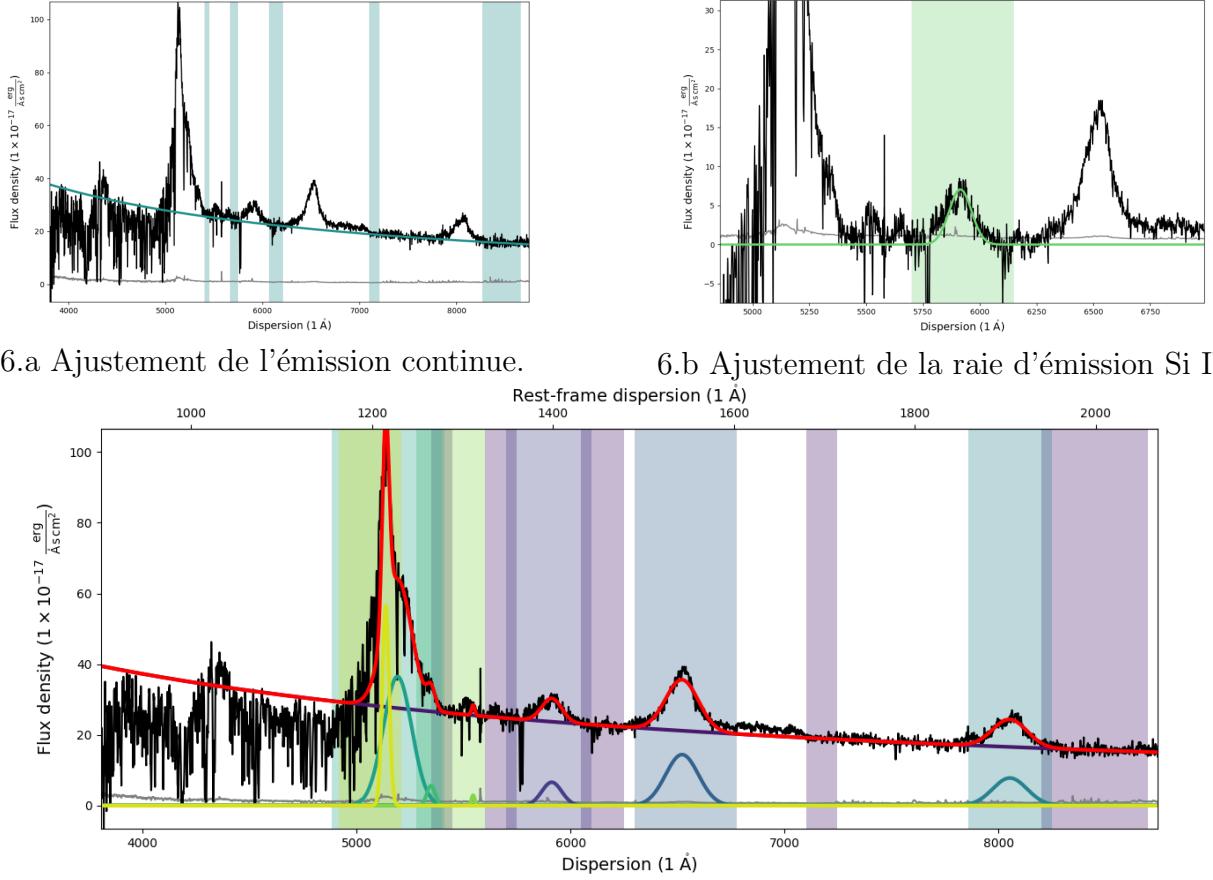


Figure 6: Procédure d'ajustement du spectre sur les zones colorées.

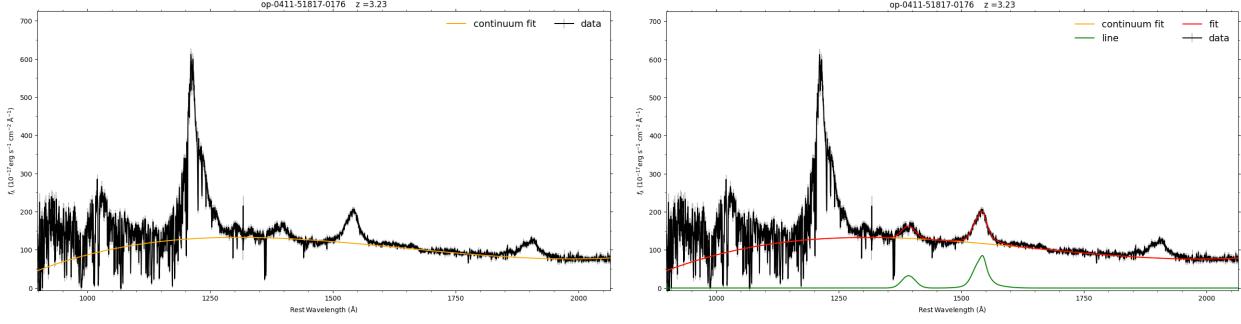
on obtient un résultat plus satisfaisant. Nous allons donc chercher une méthode d'ajustement qui utilise plusieurs gaussiennes par raies d'émission.

2.2 PyQSOFit.

Pour étudier les SMBH, Q. Wu et Y. Shen s'intéressent aux propriétés des raies d'émission[5]. Leur but est de reconstruire l'émission continue et les raies d'émission des quasars, d'identifier les raies fines et larges, d'estimer le redshift des SMBH et leur masse ainsi que l'impact du S/N sur la mesure des propriétés spectral[6]. Ces mesures nécessitent d'ajuster les raies d'émission très précisément. Pour ce faire Guo et al. (2018)[7] ont développé un code python : PyQSOFit⁸. Le code suit une procédure d'ajustement similaire à celle décrite en 2.1. L'émission continue du quasar est ajustée avec une loi de puissance et un polynôme de degré 3. Certaines raies d'émission sont ajustées avec plusieurs gaussiennes (Cf. Figure 7.a, 7.b et 7.c). Le fit de l'ensemble des quasars de SDSS a déjà été effectué par Q. Wu et Y. Shen⁹. On récupère donc le fichier FITS contenant le spectre et les paramètres de fit. Ensuite on reconstruit le fit à partir des paramètres d'ajustement.

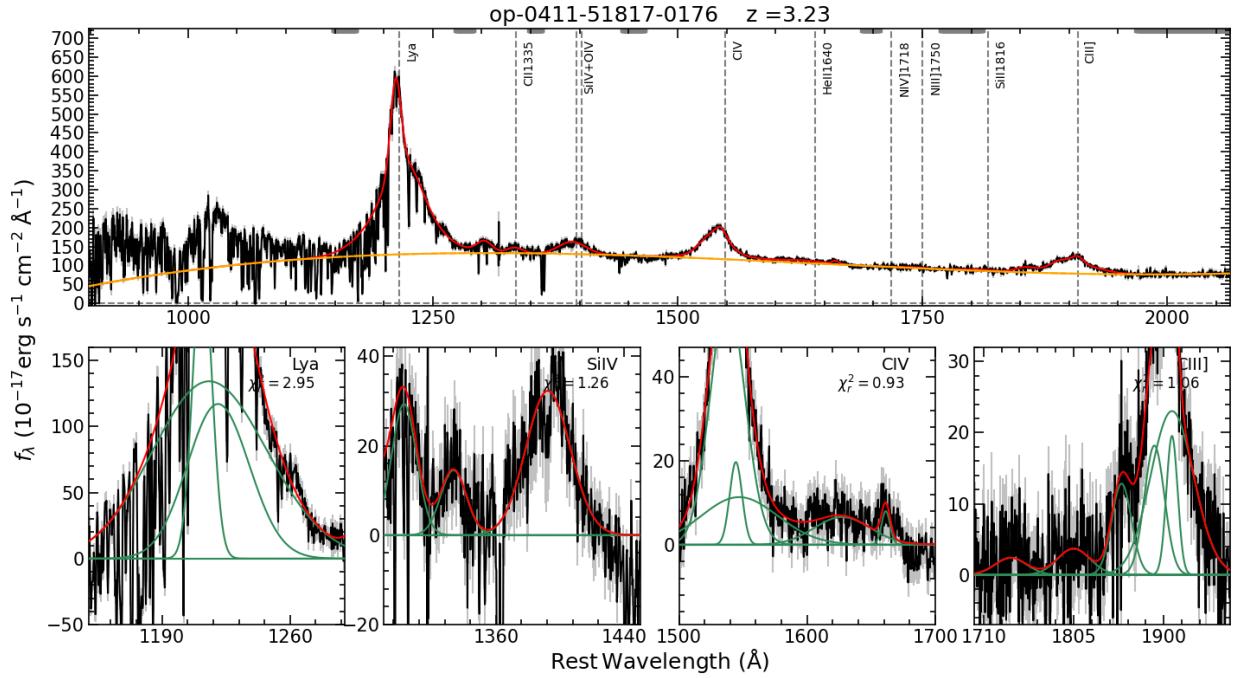
⁸Descriptif et codes : <https://github.com/legolason/PyQSOFit>

⁹Fit et paramètres de fit : http://quasar.astro.illinois.edu/paper_data/DR16Q/



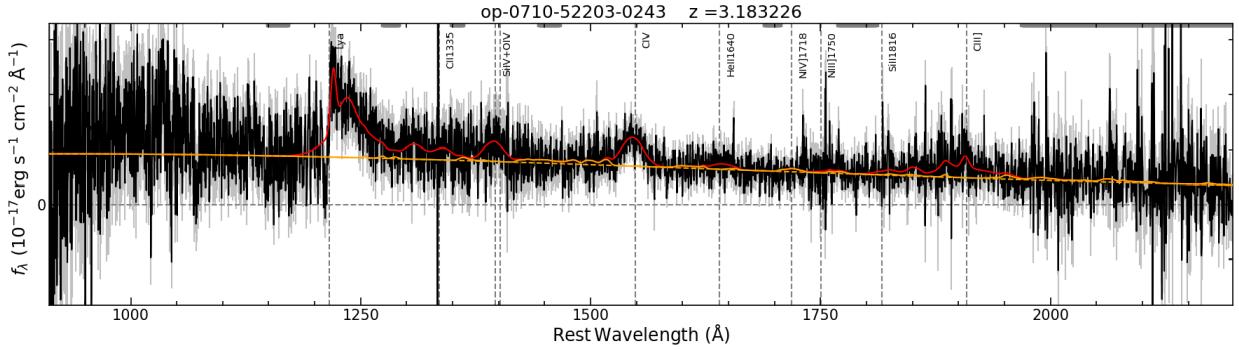
7.a Ajustement de l'émission continue.

7.b Ajustement des raies SiIV et CIV.



7.c Fit de 411-51817-176 avec $S/N = 22.46$.

On peut voir les différentes composantes gaussiennes utilisées pour chaque raie.



7.d Fit de 710-52203-243 avec $S/N = 2.96$.

Figure 7: Fit PyQSOFit.

On peut remarquer que le fit¹⁰ fonctionne particulièrement bien pour les longueurs d'onde supérieures au Ly α . L'émission continue et les raies d'émission sont correctement reconstruites. Utiliser plusieurs gaussiennes par raie d'émission est donc une méthode adaptée. De plus, le continu est correctement reconstruit même à faible S/N (Cf. Figure 7.d). On peut cependant voir qu'il est incapable de reproduire la forêt Ly α . Comme indiqué par les auteurs du code il arrive que le fit fonctionne mal dans certains cas et pour certaines raies d'émissions (Cf. Figure 8). On peut constater que les raies situées à $\lambda \sim 1250$ Å et $\lambda \sim 1300$ Å ainsi que l'émission continue vers $\lambda \sim 1725$ Å, $\lambda \sim 1980$ Å et $\lambda \sim 2080$ Å sont mal reconstruites. On pourra voir dans la section 2.4 que ces écarts, en apparence faibles, auront un impact significatif sur la normalisation du spectre. Il faudrait donc faire une vérification visuelle ou bien appliquer un autre algorithme qui vérifierait que l'écart entre la courbe et le fit.

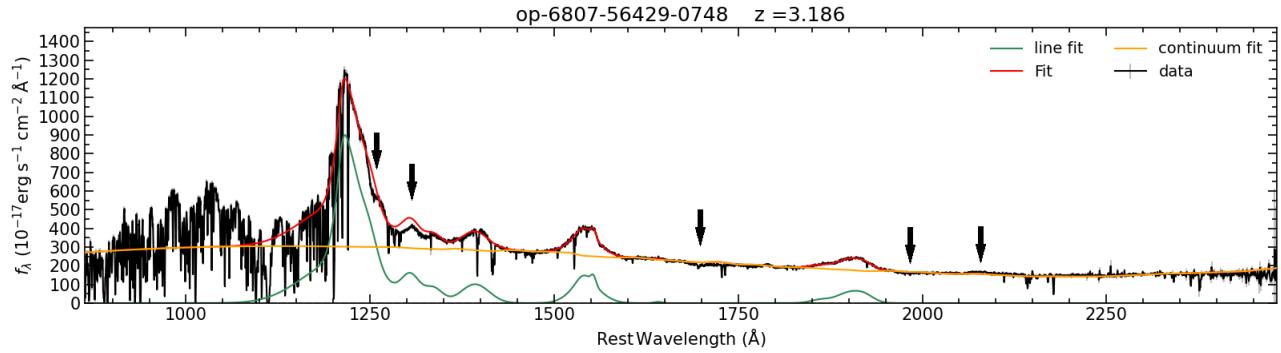


Figure 8: Fit de 6807-56429-748 avec $S/N = 46.5$.

Pour palier aux problèmes de fit et du temps de calcul on peut se demander si une méthode plus simple ne serait pas plus adaptée. Nous choisissons donc une des méthodes d'ajustement les plus simples : le smoothing.

2.3 Smoothing

Le smoothing est effectué avec la fonction *convolve* de la bibliothèque python numpy¹¹. On fait un premier lissage global de l'ensemble du spectre appelé *baseline* (Cf Figure 9.a). Pour chacun des points du spectre on effectue une convolution de l'ensemble des points compris dans une certaine fenêtre autour du point considéré. Cette baseline sert de référence de comparaison pour retirer les éléments du spectre qui en sont trop éloignés (Cf. Figure 9.b). On compare l'écart entre la baseline et le spectre puis on retire ce point si cet écart est supérieur à une valeur de coupure. On interpole ensuite les éléments manquants puis on lisse la courbe ainsi obtenue¹² (Cf. Figure 9.c). Comme attendu avec un smoothing, certaines raies trop piquées peuvent être “rabottées” (Cf. la raie Ly α Figure 9.c). On peut aussi noter que la forêt Ly α est beaucoup mieux reconstruite que pour la méthode de PyQSOFit. Les deux paramètres qui contrôlent la procédure de smoothing sont la largeur de la fenêtre de convolution et la valeur de coupure pour les spikes.

¹⁰Le fit est reconstruit à partir des paramètres d'ajustement fournis par PyQSOFit, voir le code en Annexe 5.4.

¹¹Description : <https://numpy.org/doc/stable/reference/generated/numpy.convolve.html>

¹²Code du smoothing en Annexe 5.5.

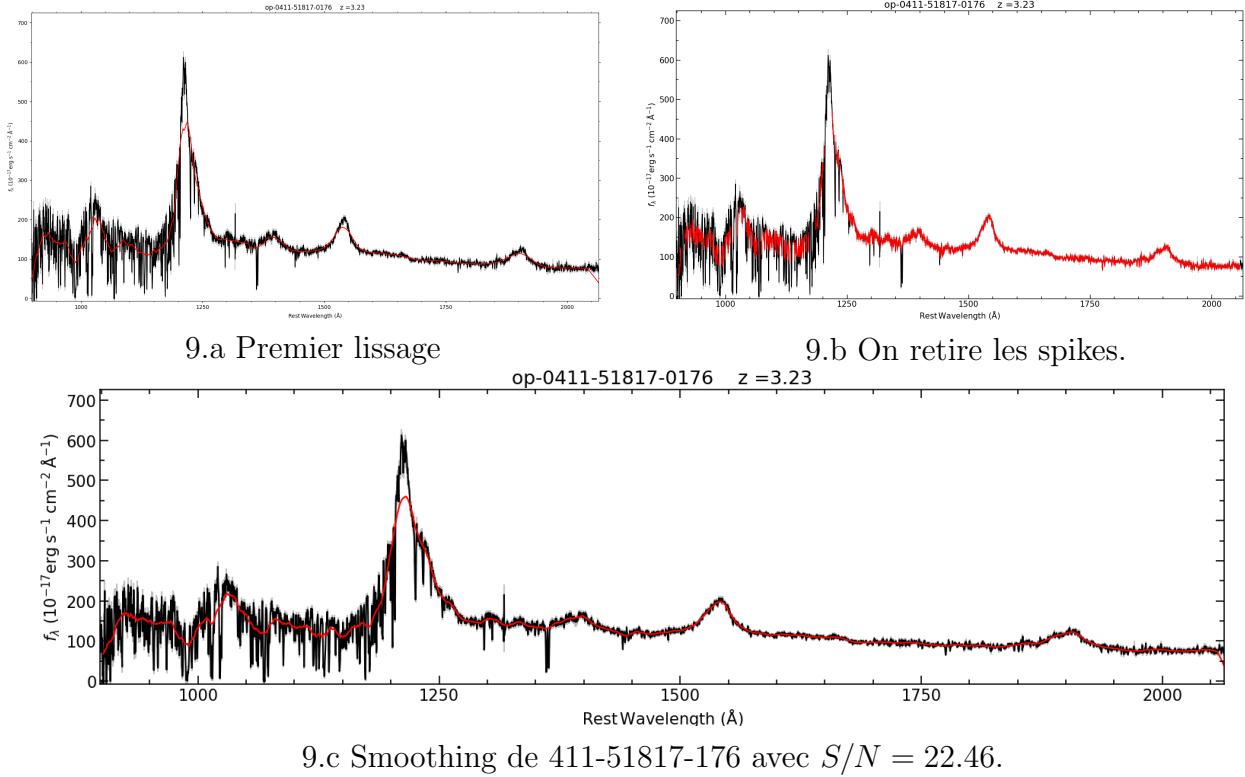


Figure 9: Procédure de smoothing.

Le premier contrôle le lissage de la courbe et le deuxième quels points seront sélectionnés. La valeur de coupure est donc fortement dépendante du S/N.

2.4 Discussion

Le spectre normalisé est obtenu en faisant le rapport de la courbe du spectre par la courbe d'ajustement. On obtient une courbe normalisée à 1 avec des raies d'absorption particulièrement visibles. On s'attend à ce que la forêt Ly α soit particulièrement mal normalisée pour la méthode PyQSOFit et la raie d'émission Ly α pour le smooth (Cf Figure 10).

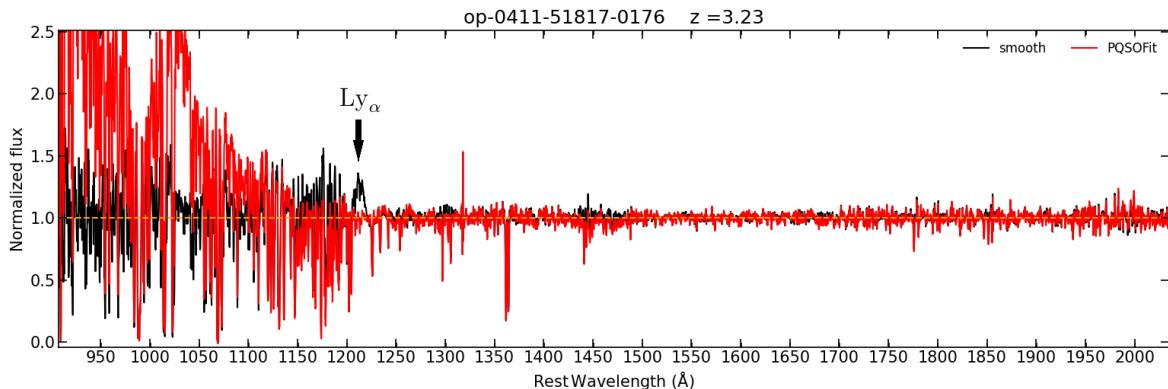


Figure 10: Comparaison PyQSOFit et smooth pour 411-51817-176.

Pour le reste du spectre les deux méthodes produisent des résultats similaires pour la majorité des quasars. On peut cependant constater que le smooth est plus stable que PyQSOFit. On peut voir que les écarts entre le fit et le continu du spectre de la [Figure 8](#) étaient de l'ordre de $10 \sim 20\%$ (Cf. [Figure 11](#)). De tels écarts pourraient s'avérer problématiques pour la détection et l'étude des raies d'absorption.

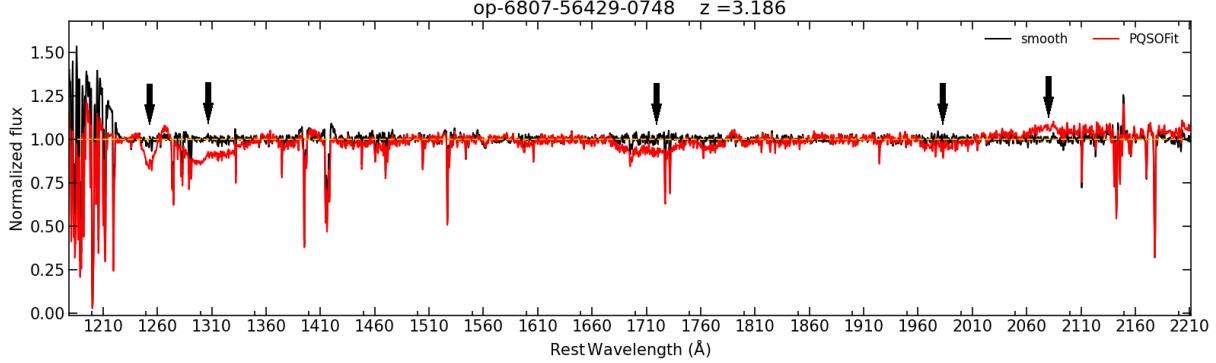


Figure 11: Comparaison PyQSOFit et smooth pour 6807-56429-748.

3 Conclusion

Au cours de ce stage nous avons développé un programme permettant de sélectionner des quasars sur le catalogue SDSS DR16 en fonction de certains critères ; de récupérer les données spectrales de quasars depuis les banques de donnée SDSS et des fits de Q. Wu et Y. Shen ; de reconstruire les fit à partir des paramètres fournis par Q. Wu et Y. Shen ; de construire un smooth à partir des données de SDSS.

Ce travail nous a permis de comparer les trois méthodes, l'ajustement avec une seule gaussienne par raie, PyQSOFit et le smooth pour différents quasars. Nous avons conclu que l'ajustement avec une seule gaussienne était insuffisant pour pouvoir reconstruire le continu. Il semble que la méthode PyQSOFit soit plus adaptée pour l'étude des propriétés des raies d'émission à travers les paramètres contenus dans le fit. Alors que la méthode du smooth semble plus adaptée à la normalisation de spectre de quasar pour l'étude des raies d'absorption.

Pour améliorer la méthode PyQSOFit nous pourrions refaire le fit nous même en changeant les options de la fonction d'ajustement ou bien les paramètres de fit pour les spectres qui posent problème. On pourrait par exemple ajouter ou enlever des gaussiennes pour le fit de certaines raies d'émission. Pour améliorer la méthode smooth nous pourrions affiner la sélection des points à supprimer en optimisant les deux paramètres : la valeur de coupure et la fenêtre de convolution. Nous pourrions aussi directement utiliser la méthode de smooth développée par Ďurovčíková et al (2020)[8].

4 Bibliographie

- [1] E. W. Flesch, “The million quasars (milliquas) catalogue, v8,” The Open Journal of Astrophysics, vol. 6, Dec. 2023.
- [2] “What is the sdss?.” <https://classic.sdss.org/documents/goals.php>, 1993.
- [3] B. W. Lyke, A. N. Higley, J. N. McLane, D. P. Schurhammer, A. D. Myers, A. J. Ross, K. Dawson, S. Chabanier, P. Martini, N. G. Busca, H. d. Mas des Bourboux, M. Salvato, A. Streblyanska, P. Zarrouk, E. Burtin, S. F. Anderson, J. Bautista, D. Bizyaev, W. N. Brandt, J. Brinkmann, J. R. Brownstein, J. Comparat, P. Green, A. d. l. Macorra, A. M. Gutiérrez, J. Hou, J. A. Newman, N. Palanque-Delabrouille, I. Pâris, W. J. Percival, P. Petitjean, J. Rich, G. Rossi, D. P. Schneider, A. Smith, M. Vivek, and B. A. Weaver, “The sloan digital sky survey quasar catalog: Sixteenth data release,” The Astrophysical Journal Supplement Series, vol. 250, p. 8, Aug. 2020.
- [4] T. Tepper García, “Voigt profile fitting to quasar absorption lines: an analytic approximation to the voigt-hjerting function: A new method to compute voigt profiles,” Monthly Notices of the Royal Astronomical Society, vol. 369, p. 2025–2035, June 2006.
- [5] Q. Wu and Y. Shen, “A catalog of quasar properties from sloan digital sky survey data release 16,” The Astrophysical Journal Supplement Series, vol. 263, p. 42, Dec. 2022.
- [6] Y. Shen, P. B. Hall, K. Horne, G. Zhu, I. McGreer, T. Simm, J. R. Trump, K. Kinemuchi, W. N. Brandt, P. J. Green, C. J. Grier, H. Guo, L. C. Ho, Y. Homayouni, L. Jiang, J. I.-H. Li, E. Morganson, P. Petitjean, G. T. Richards, D. P. Schneider, D. A. Starkey, S. Wang, K. Chambers, N. Kaiser, R.-P. Kudritzki, E. Magnier, and C. Waters, “The sloan digital sky survey reverberation mapping project: Sample characterization,” The Astrophysical Journal Supplement Series, vol. 241, p. 34, Apr. 2019.
- [7] H. Guo, Y. Shen, and S. Wang, “PyQSOFit: Python code to fit the spectrum of quasars.” Astrophysics Source Code Library, Sept. 2018.
- [8] D. Ďurovčíková, H. Katz, S. E. I. Bosman, F. B. Davies, J. Devriendt, and A. Slyz, “Reionization history constraints from neural network based predictions of high-redshift quasar continua,” Monthly Notices of the Royal Astronomical Society, vol. 493, p. 4256–4275, Feb. 2020.

5 Annexe A : Code

5.1 Sélection dans le catalogue

```
import os
from astropy.io import fits

class Fetch:
    def __init__(self,
                 name=None,
                 data_name=None,
                 number=None,
                 z_min=None,
                 z_max=None,
                 BAL=None,
                 SN=None,
                 ):

#name : name of the .fits catalog file, default is
#'DR16Q_V4.fits', description :
#      https://data.sdss.org/datamodel/files/BOSS_QSO/DR16Q/\
#      DR16Q_v4.html
#data_name : name of the output data file, default is\
#'qso-data.txt'
#number : number of quasar wanted, default is 10, 'all' fetches\
#      all quasars
#z_min : minimum value the redshift z can take, default is 0
#z_max : maximum value the redshift z can take, default is 20
#BAL : minimum Broad Absorption Line probability, default is 0
#SN : minimum Signal to Noise ratio, default is 0

        if name is None:
            self.name='DR16Q_V4.fits'
        else:
            self.name = name
        if data_name is None:
            self.data_name = 'qso-data.txt'
        else:
            self.data_name = data_name
        self.folder_path = os.path.join(os.getcwd(), \
'catalog')
        self.file_path = os.path.join(self.folder_path, \
self.name)
        self.data_path = os.path.join(os.getcwd(), 'data', \
self.data_name)
```

```

        self.data = fits.getdata(self.file_path,1)
        if number == 'all':
            self.number = len(self.data)
        elif number is None:
            self.number = 10
        else:
            self.number = number-1
        if BAL is None:
            self.BAL = 0
        else :
            self.BAL = BAL
        if z_max is None:
            self.z_max = 20
        else:
            self.z_max = z_max
        if z_min is None:
            self.z_min = 0
        else:
            self.z_min = z_min
        if SN is None:
            self.SN = 0
        else:
            self.SN = SN

    def get_data(self):

#Fetch the required data from the .fits catalog and write it\
#in a separate file

        if self.number is None:
            self.number = len(self.data)
        open(self.data_path, 'w').close()
        j, i = 0, 0
        while i in range(len(self.data)) and j <= self.number:
            #TO DO: make a logical function condition
            if (self.data[i].field('Z')) >= self.z_min and \
                (self.data[i].field('Z') <= self.z_max) and \
                (self.data[i].field('BAL_PROB') >= self.BAL)\ and \
                (self.data[i].field('SN_MEDIAN_ALL'))\ >= self.SN):
                f = open(self.data_path, "a")
                line = ''
                for elem in self.data[i][0:6]:
                    line = line + str(elem) + ','
                j = j + 1
                line = line + str(self.data[i].field('Z')) + ','\
```

```

                + str(self.data[i].field('BAL_PROB')) \\
                + ' '
                + str(self.data[i].field('SN_MEDIAN_ALL'))
        f.write(line + "\n")
        f.close()
        i = i + 1
    self.file = open(self.data_path, 'r')
    print('Checked', i, 'quasars out of', len(self.data))
    print('Found', j, 'quasars corresponding to criteria')

def dir_check(self):
    if not os.path.exists(self.folder_path):
        print('Catalog file must be in folder:', \
              self.folder_path)
        return
    folder_path = os.path.join(self.folder_path, self.name)
    if not os.path.exists(folder_path):
        print('Catalog file', self.name, \
              'not found in folder:', self.folder_path)
        return
    elif os.path.exists(folder_path):
        print('Catalog file located:', self.name)

def fetch_data(self):

#Fetch the data from the .txt data file created in the
#get_data() function

    data=[]
    for i in range(self.number+1):
        line = self.file.readline()
        line = list(line.removesuffix('\n').split(' '))
        data.append(line)

    return data

```

On utilise cette class Fetch avec par exemple :

```

from quasar_fetch import Fetch

p = Fetch(name=None, data_name=None, number='all', \
           z_min=2.5, z_max=3.5, BAL=0.5, SN=30)
p.dir_check()
p.get_data()

```

5.2 Récupération du spectre SDSS

```
from astroquery.sdss import SDSS

# Fetching SDSS spectra of 5808-56325-560
specs = SDSS.get_spectra(plate=5808, mjd=56325, fiberID=560)

# Assigning spectra data
spectra_data = specs[0][1].data
wavelength = 10**spectra_data.field('loglam')
flux = spectra_data.field('flux')
```

5.3 Récupération du fit.

```
from astropy.io import fits
# Definition of spectra ID
plate,mjd,fiberID,z = 6807,56429,748,3.186

# Download the fit file
spec_url = f'http://quasar.astro.illinois.edu\
/paper_data/DR16Q/fits\
/{plate:04d}/op-{plate:04d}-{mjd}-{fiberID:04d}.fits.gz'

# Assigning spectra data
op_data2 = fits.getdata(spec_url,1)
op_data2_MC = fits.getdata(spec_url,2)
op_data2_spec = fits.getdata(spec_url,3)
```

5.4 Reconstruction du fit de Q. Wu et Y. Shen.

Le fit est obtenu en faisant la somme de *line_flux*, *conti_model* et *pl_model*.

```
from qsofunc import *
# read continuum fitting parameters
conti_keyword = ['PL_norm', 'PL_slope', 'POLY_a', 'POLY_b', \
    'POLY_c', 'Fe_uv_norm', 'Fe_uv_FWHM', 'Fe_uv_shift', \
    'Fe_op_norm', 'Fe_op_FWHM', 'Fe_op_shift']
conti_para = np.array([op_data2[name][0] for name in \
    conti_keyword])
conti_model = continuum_all(op_data2_spec.wave_prereduced, \
    conti_para)

# power law + poly continuum model
pl_model = conti_para[0]*(op_data2_spec.wave_prereduced/3000.0)\ \
    **conti_para[1]
```

```

poly_model = F_poly_continuum(op_data2_spec.wave_prereduced,\n
                             conti_para[2:5])
pl_poly_model = pl_model + poly_model

# complex line lists
op_linelist = np.array(['HALPHA', 'HALPHA_BR', 'NII6585',\n
                       'SII6718', 'HBETA', 'HBETA_BR', 'HEII4687',\n
                       'HEII4687_BR', 'OIII5007', 'OIII5007C', 'CAII3934',\n
                       'OII3728', 'NEV3426', 'MGII', 'MGII_BR', 'CIII_ALL',\n
                       'CIII_BR', 'SIIII1892', 'ALIII1857', 'NIII1750', '\n
                       'CIV', 'HEII1640', 'HEII1640_BR', '\n
                       'SIIIV_OIV', 'OI1304', 'LYA', 'NV1240'])

# fitted complex line lists
comp_lst = np.array([op_data2[hdn][0] for hdn in op_data2.names\n
                     if ('complex_name' in hdn) and ('local' not in hdn)])
ncomp = len(comp_lst)

# fitted individual line lists
line_lst = np.array([hdn[:-6] for hdn in op_data2.names if\n
                     ('_scale' in hdn) and ('err' not in hdn)])

# calculate the fitted gaussian line fit models
line_flux = np.zeros(len(op_data2_spec.wave_prereduced))
for l in range(len(line_lst)):\n
    line_gauss = np.array([op_data2[line_lst[l]+'_scale'], \n
                          op_data2[line_lst[l]+'_centerwave'], \n
                          op_data2[line_lst[l]+'_sigma']])
    line_flux+=Manygauss(np.log(op_data2_spec.wave_prereduced)\n
                         , line_gauss)

```

5.5 Smoothing.

```

import numpy as np
import pandas as pd
def smooth(y, window):\n
    # Convolves y with window
    box = np.ones(window)/window
    y_smooth = np.convolve(y, box, mode='same')
    return y_smooth

def remove_outliers(spikey, baseline, cutoff):\n
    # Removes data from spikey that is > cutoff from baseline.
    np_spikey = np.array(spikey)
    np_baseline = np.array(baseline)

```

```
cond_delta = (abs(np_spikey-np_baseline) > cutoff)
np_remove_outliers = np.where(cond_delta, np.nan, np_spikey)
return np_remove_outliers

baseline = smooth(op_data2_spec.flux_prereduced,31)
remove_spike = remove_outliers(op_data2_spec.flux_prereduced\
, baseline, 30)
interpolate = pd.DataFrame(remove_spike).interpolate()
smooth = smooth(interpolate.to_numpy().ravel(),31)
```