

Manuel d'utilisation mysr_script et mysr_split

Louis BONDONNEAU

13 septembre 2015

Table des matières

1	Le processus	4
1.1	Refold	5
1.2	Scrunch	5
1.3	Template	6
1.4	TOAs	6
1.5	Tempo2	6
1.6	Seconde boucle	7
2	Le template simple	7
2.1	Les erreurs rencontrées	7
2.2	création des dossiers et sous dossiers pour le processus	7
2.3	Vérification des chemins	7
2.4	Boucle 1	8
2.4.1	refold 1, scrunch 1,scrunch DFTp 1	9
2.4.2	make template 1	9
2.4.3	make toa 1, tempo2_1	9
2.5	Boucle 2	10
2.6	Plots	11
3	Le multi_templates	12
3.1	Vérification des chemins	13
3.2	Lancement du programme	13
3.2.1	refold, split, scrunch, scrunch DFTp	14
3.2.2	make templates	15
3.2.3	make TOAs, tempo2	15
3.3	Plots	15
4	Le fichier SAUVE	18
5	Les dépendances	18

Liste des tableaux

Table des figures

1	Organigramme du processus.	4
2	Ouverture du script "mysr_script".	8
3	Choix du fichier paramètres.	8

4	Les premiers TOAs sous tempo2.	10
5	Variation du DM de B1237+25 avec plot_DMvar.	11
6	Comparaison des variations de DM avec subplot_DMvar de B1237+25.	12
7	Organigramme du processus de multi-templates.	13
8	Ouverture du script "mysr_split".	14
9	Templates en fréquences, lissés et avec un décalage en intensité pour B2111+46.	16
10	Templates en fréquences, lissés et superposés pour B2111+46.	16
11	Templates en fréquences, additionnés et avec un décalage en intensité pour B2111+46.	17
12	Templates en fréquences, additionnés et superposés pour B2111+46.	17

1 Le processus

Dans le but d'améliorer les paramètres d'un pulsar et ainsi pouvoir procéder à une analyse plus poussée, le logiciel bash "myspr_script.sh" est écrit. Il est basé sur les scripts de Lucas Guillemot. Ce script est composé de deux boucles elles-mêmes composées de plusieurs sous-parties représentées sur la figure 1 que nous allons expliquer ici.

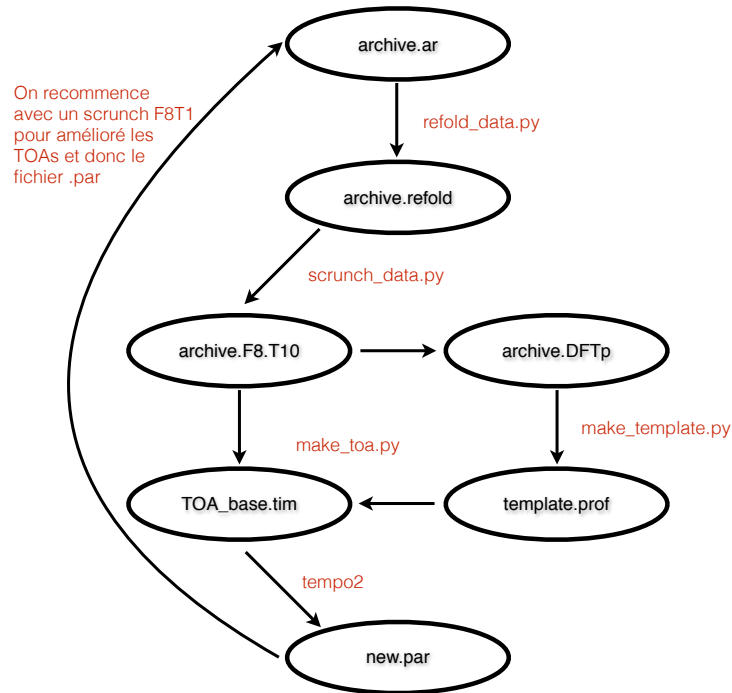


FIGURE 1 – Organigramme du processus.

1.1 Refold

Script : `python/refold_data.py`

Le Refold comme son nom l'indique permet de faire un premier repliement sur le temps. Il a donc besoin d'un fichier paramètre pour pouvoir placer un indicateur sur le début et la fin de chaque période de rotation. Il prépare les données pour les traitements qui suivent en y incorporant le fichier paramètres.

1. Entrées ;
 - Le premier fichier paramètre
 - venant des archives 'pav -e' (`./PSRNAME.par`)
 - ou du dossier courant
 - Les archives (`./data/links/*.ar`)
2. Sortie ;
 - Les archives `.refold` (`./data/refolded/backend/*.refold`)

1.2 Scrunch

Script : `python/scrunch_data.py`

Le scrunch (écrasement en temps et en fréquence) permet d'augmenter le SNR du signal tout en diminuant la taille des fichiers.

Dans la première boucle, il est décomposé en deux parties :

- Un scrunch F8T10 (8 canaux en fréquences et 10 subintégrations par observation. Dans le but de l'utiliser pour faire les premiers TOAs (Times Of Arrival), avec une bonne définition temporelle pour pouvoir fitter sur une observation et avoir une première approximation de la fréquence F_0 et du DM.
- Un scrunch DFTp (écrasement total sur Fréquences Time polarisations). Pour créer un template du pulsar et s'en servir pour déterminer les TOAs dans le scrunch F8T10.

Le script `scrunch_data.py` permet d'utiliser l'option `-dedisperse` donnant la possibilité de dedisperser le signal en sortie. En réalité, ce sont uniquement les indicateurs de chaque période qui sont alors déphasés pour absorber l'effet de dispersion du milieu.

1. Entrée :
 - Les archives `.refold` (`./data/refolded/backend/*.refold`)
2. Sorties :
 - Les archives `.scrunchedF8_tsub10` (`./data/scrunch/backend/*.scrunchedF8_tsub10`)
 - Les archives `.DFTp` (`./data/scrunch/backend/*.DFTp`)

1.3 Template

Script : python/make_templates.py

Le template est le profil type du pulsar. Il est utilisé pour déterminer les temps d'arrivées des observations. Il additionne les archives .DFTp et lisse le résultat.

Le script make_templates.py est utilisé avec plusieurs options.

- forcealign : aligne de force les impulsions.
- automax : détermine les profils à additionner en fonction de leurs SNR .

1. Entrée :

- Les archives .DFTp (./data/scrunch/backend/*.DFTp)

2. Sorties :

- png des profils additionnés
- les templates.prof additionnés et lissés (./smoothed_F8.tim)

1.4 TOAs

Script : python/make_toas.py

Make TOAs à pour but de calculer les temps d'arrivées des impulsions par une corrélation croisée entre le template et l'observation.

1. Entrées :

- Les archives .scrunchedF8_tsub10 (./data/scrunch/backend/*.scrunchedF8_tsub10)
- Le template (./smoothed_F8.tim)

2. Sortie :

- Le fichier (./TOAs_base_F8.tim)

1.5 Tempo2

Tempo2 est un outil permettant de fitter les paramètres d'un pulsar sur les temps d'arrivé, et d'éliminer par la même occasion les temps d'arrivé absurdes.

Le but de cette étape est d'aligner les TOAs en fittant en premier lieu sur une et par la suite sur tout les observations.

1. Entrées :

- Le fichier (./TOAs_base_F8.tim)
- Le fichier paramètre

2. Sorties :

- Le fichier nouveau fichier paramètre

1.6 Seconde boucle

Par la suite, on réitère le processus sur une nouvelle boucle avec le nouveau fichier paramètre, mais avec quelques différences :

- Le scrunch est fait sur F8T1 pour améliorer TOAs.
- À la fin de cette boucle, un fichier dmvals.dat est généré automatiquement à l'aide de la commande :

```
tempo2 -gr dm
```

Il contient les mesures de dispersion pour chaque observation.

2 Le template simple

2.1 Les erreurs rencontrées

Il y a tout d'abord certaines conditions à vérifier pour que les archives soient traitées sans erreur :

- Le nombre total de canaux doit être un multiple de 8.
- Le site référencé dans l'archive doit être dans la base de données utilisée par tempo2.

2.2 création des dossiers et sous dossiers pour le processus

Dans le but de faciliter cette étape, L.Guillemot a conçu un script intitulé setup_directoy.py.

```
python setup_directoy.py -quick -pulsar 0329+54 -destdir B0329+54  
-extdataext ar -extdatadir /data/.../B0329/
```

L'exécution de la commande crée un dossier B0329+54 contenant un sous-dossier data/Links dans lequel sont stockés les liens des archives.

2.3 Vérification des chemins

Avant de lancer le code, la première chose à faire est de vérifier si les variables des chemins sont bien initialisées

- DIR : (directory principal) contenant un dossier parfiles contenant lui-même un dossier new regroupant les nouveaux fichiers paramètres. Le dossier DIR contient aussi le dossier "python" avec les scripts prérequis.
- lucasscripts : chemin vers les utilitaires de L.Guillemot
- setuptempo2 : chemin vers le fichier de réglage pour l'affichage de tempo 2.

2.4 Boucle 1

cd B0329+54

```
bash mypsr_script.sh
```

```

      \ |          # SCRIPT menu.sh
       \|         # V: 0.9.1
        \|        # DATUM: 21.07.2015
         O         # 
        /|         # 
       / |         # 
      /  \|        # 
     /    \|       # 

site=FR606
-----
Select an option:

1) All the processing           4) SPLIT_processing
2) Start with a particular step 5) Plots
3) Only one step               6) Quit
#? 1
```

FIGURE 2 – Ouverture du script "myspr_script".

À l'ouverture du script, il est possible de choisir entre plusieurs options :

1. All the processing, pour commencer processing complet.
2. Start with a particular step, pour démarrer le processus à une étape particulière.
3. Only one step, seulement une étape du processus.
4. Plots, pour faire ou refaire certains graphiques.
5. Quit, pour fermer le programme.

On entre 1 pour démarrer le processus.

```
How would you want select your parameter file ?
1) Manually                      3) Automatic
2) chose in current directory    4) Return
#? 2
```

FIGURE 3 – Choix du fichier paramètres.

1. Manually : pour donner le chemin du fichier paramètres
2. Chose in current directory : choisir le fichier paramètres dans le dossier courant.
3. Automatic : un choix automatique dans le dossier parfiles.
 - (a) Fichier.par le plus récent dans le dossier parfiles/new/.
 - (b) Sinon dans le dossier courant.
 - (c) Ou alors on le récupère avec un vap -E.
4. Return

2.4.1 refold 1, scrunch 1,scrunch DFTp 1

Par la suite, le programme va débiter le refold et les deux scrunch. Ces étapes peuvent prendre beaucoup de temps, mais elles ne demandent aucune intervention.

2.4.2 make template 1

Ici, l'option "forcealigne" est activée dans le make_template pour forcer le logiciel à aligner les observations et ainsi pallier l'erreur sur la période.

La conception du template débute par l'addition des scrunchs DFTp.

Une fenêtre surgissante doit apparaître affichant et les observations ainsi que leurs MJD. De plus, un tableau avec le chemin de chaque observation ainsi que le MJD est affiché dans le terminal. C'est alors que l'on peut supprimer une observation en répondant par y (oui) ou par n (non) et en indiquant son chemin copié dans le tableau précédant.

Il est alors possible de recommencer l'opération autant de fois que l'on désire pour supprimer les observations corrompues.

À la fin de cette étape un fichier .prof a été créé dans le dossier courant.

2.4.3 make toa 1, tempo2_1

Les premiers TOAs sont générés à partir du profil. Ils sont automatiquement ouverts avec tempo2 figure 4.

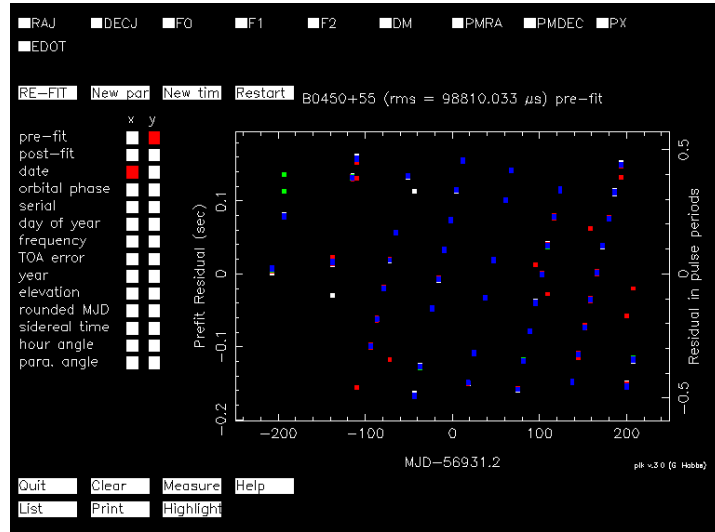


FIGURE 4 – Les premiers TOAs sous tempo2.

Bien évidemment, ce sont les premiers TOAs alors les paramètres sont souvent assez mauvais comme on le voit sur la figure 4. Dans cette boucle, les observations ont 10 sub-intégrations de façon à pouvoir vérifier la justesse de la période à cette échelle.

Il faut enregistrer le nouveau fichier paramètres avec une extension .par. À ce moment, le logiciel permet de recommencer le tempo 2 avec le nouveau fichier paramètres pour les vérifier ou les ajuster.

La première boucle est terminée avec en sortie un fichier paramètres.

2.5 Boucle 2

Dans cette seconde boucle, les seules différences avec la première boucle sont le fait qu'il y a une seule sub-intégration dans chaque observation et la désactivation du "forcealign" dans le make_template.

À la fin de cette seconde boucle, le fichier paramètres de sortie est jugé comme optimal pour nos observations.

Ici il est possible de sortir un fichier .tim de tempo2 pour pouvoir tracer la variation du DM.

2.6 Plots

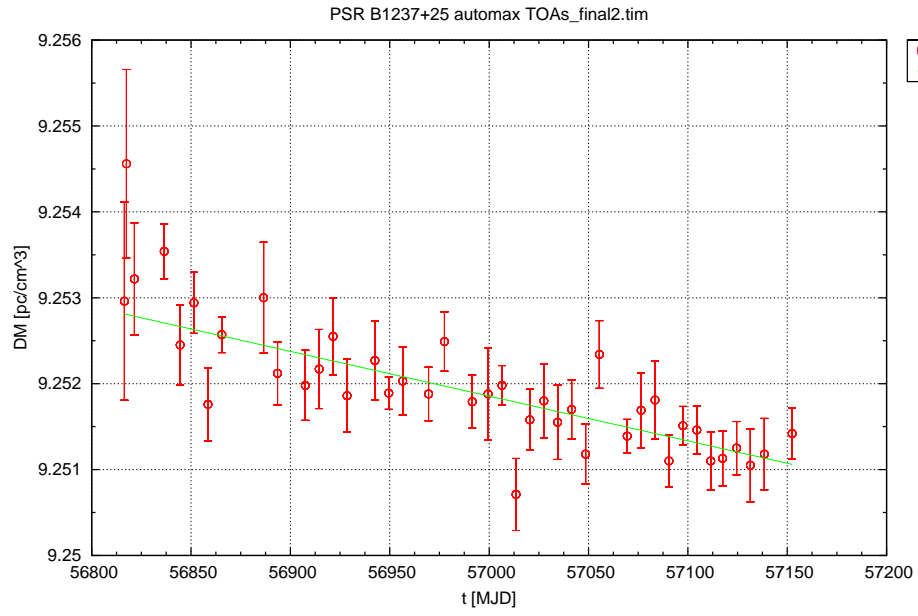


FIGURE 5 – Variation du DM de B1237+25 avec plot_DMvar.

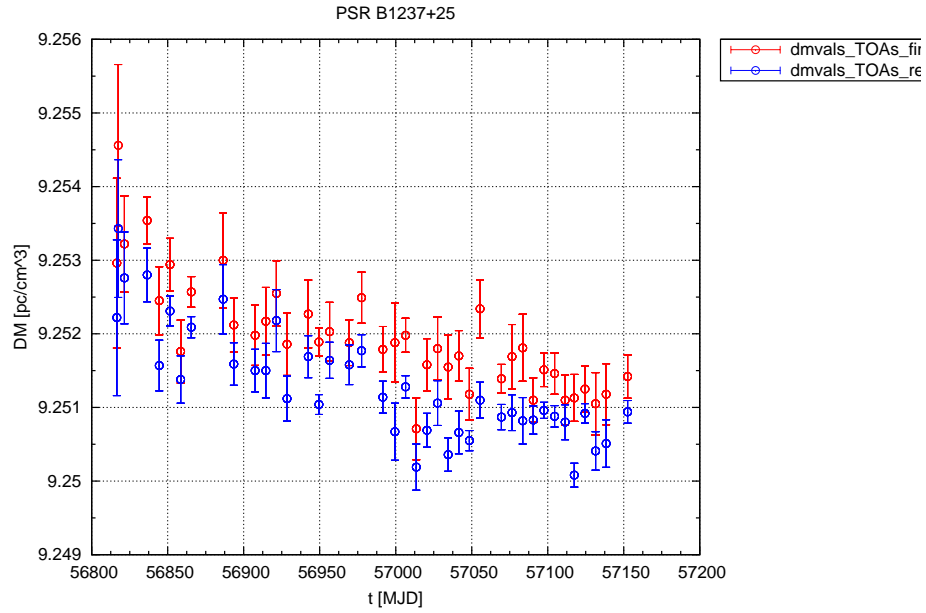


FIGURE 6 – Comparaison des variations de DM avec subplot_DMvar de B1237+25.

Ce logiciel regroupe quelques options pour tracer les variations de DM dans le menu "Plots".

1. plot_DMvar : trace une variation de DM avec une régression linéaire (figure 5).
2. subplot_DMvar : permet de superposer deux variations de DM pour les comparer (figure 6).

3 Le multi_templates

C'est étape consiste en un même processus que les boucles un et deux. Cependant, elle démarre à partir des archives.refold de la seconde boucle (figure 7).

De plus, elle calcule les TOAs à partir de plusieurs templates répartis dans le spectre. Le logiciel utilisé est intitulé "mysr_split.sh".

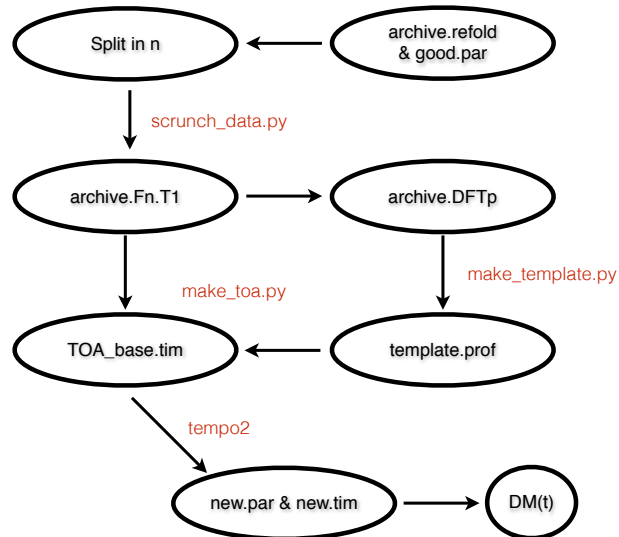


FIGURE 7 – Organigramme du processus de multi-templates.

3.1 Vérification des chemins

Avant de lancer le code, la première chose à faire est de vérifier si les variables des chemins sont bien initialisées

- DIR : (directory principal) contenant un dossier parfiles contenant lui même un dossier new regroupant les nouveaux fichiers paramètres ainsi que la variable. Le dossier DIR contient aussi le dossier "python" avec les scripts prérequis.
- lucasscripts : chemin vers les utilitaires de L.Guillemot
- setuptempo2 : chemin vers le fichier de réglage pour l’affichage de tempo 2.

3.2 Lancement du programme

```
bash mypsr_split.sh
```


3.2.2 make templates

Ce make templates est un peu particulier, car il crée un template pour chaque canal. L'étape est donc répétée sur le nombre de canaux que possède le signal. À la fin de cette étape des fichiers .prof sont créés dans le dossier courant.

3.2.3 make TOAs, tempo2

Les TOAs sont déterminés à partir des templates précédents. Un tempo2 se lance pour vérifier une dernière fois les paramètres et créer le fichier ".tim" qui sera utilisé pour calculer les variations du DM.

Le processus s'arrête à cette étape, il est possible de recommencer le tempo2 ainsi que de calculer et tracer les variations du DM dans le menu Plots à partir du ".tim".

3.3 Plots

Ce logiciel regroupe quelques options pour tracer différents graphiques "Plots".

1. plot_smoothed_split_templates : Trace les templates en fréquences avec un décalage (figure 9).
2. plot_smoothed_split_templates_superpose : Superpose les templates en fréquences (figure 10).
3. plot_Add_split_templates : Trace les profils additionnés (non lissés et non normalisés) en fréquences avec un décalage (figure 11).
4. plot_Add_split_templates_superpose : Superpose les profils additionnés (non lissés et non normalisés) en fréquences (figure 12).
5. tempo2 : Tracer les TOAs (figure 4).
6. plot_DMvar : trace une variation de DM avec une régression linéaire (figure 5).
7. subplot_DMvar : permet de superposer deux variations de DM pour les comparer (figure 6).

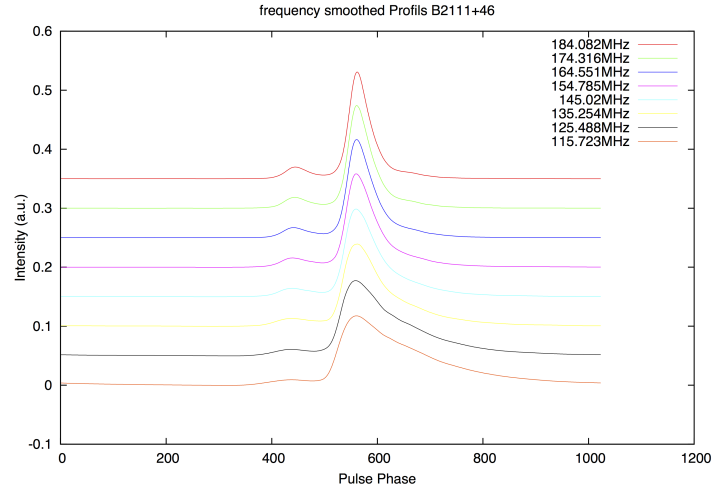


FIGURE 9 – Templates en fréquences, lissés et avec un décalage en intensité pour B2111+46.

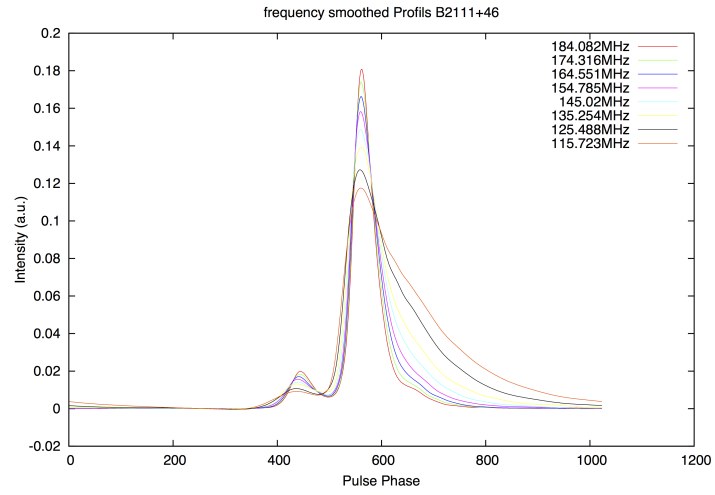


FIGURE 10 – Templates en fréquences, lissés et superposés pour B2111+46.

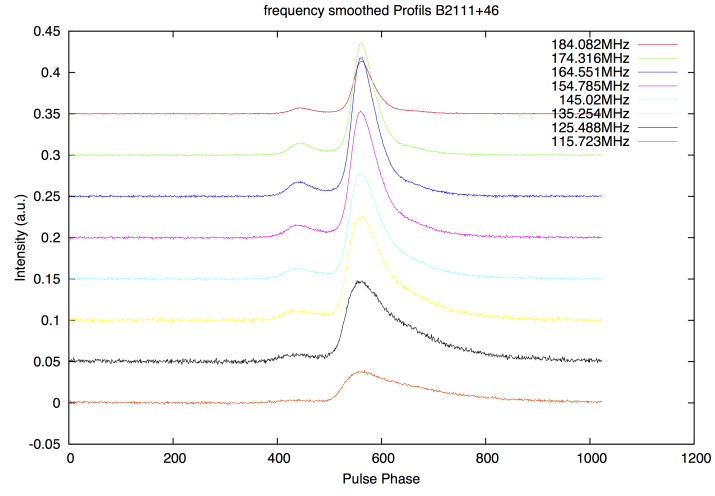


FIGURE 11 – Templates en fréquences, additionnés et avec un décalage en intensité pour B2111+46.

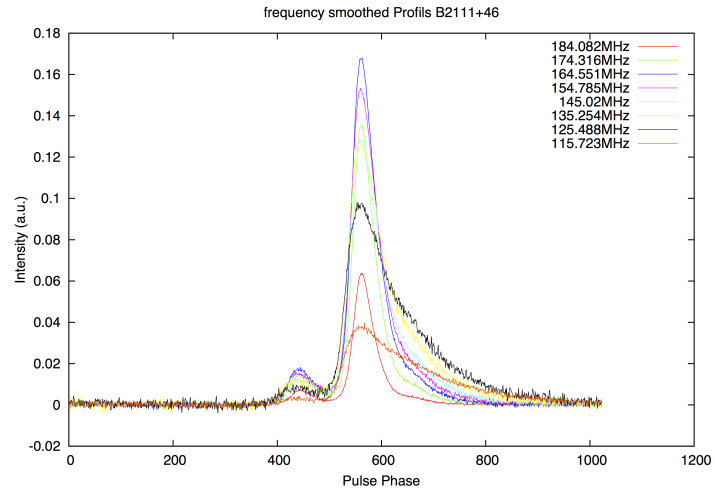


FIGURE 12 – Templates en fréquences, additionnés et superposés pour B2111+46.

4 Le fichier SAUVE

Dans le dossier de chaque pulsar est créé un dossier SAUVE qui regroupe des informations sur la précision des calculs, le fichier paramètre de la seconde boucle ainsi que les différents graphiques tracés.

plus en détails :

- Les Plots avec les profils lissés, additionner les variations de DM...
- Un fichier "results.txt" avec les erreurs médianes et moyennes sur les TOAs ainsi que le DM.
- Les templates de la première et la seconde boucle.
- Un fichier "phase.txt" avec le décalage en phase des templates dans le multi_template ainsi que leurs fréquences respectives.

5 Les dépendances

- PSRCHIVE
- L.Guillemot, refold_data.py
- L.Guillemot, scrunch_data.py
- L.Guillemot, make_templates.py
- L.Guillemot, make_toas.py
- L.Bondonneau, median.py : calcul de la médiane ainsi que la moyenne sur les TOAs
- L.Bondonneau, median_dmvals.py : calcul de la médiane ainsi que la moyenne sur les DMs
- L.Bondonneau, jump.py : pour ajouter un jump dans un .tim
- python 2
 - numpy
- gnuplot
- gwenview
- display