

Service de médecine nucléaire
et imagerie moléculaire

pySchedVisu

Documentation

Balazs Laurenczy
CHUV
2019-12-23

Table des matières

- Qu'est-ce que pySchedVisu ?
- Que fait pySchedVisu ?
- Que contient un rapport de pySchedVisu ?
- Comment fonctionne pySchedVisu ?
- Où est-ce que pySchedVisu s'exécute ?
- Où se trouve pySchedVisu ?
- Comment installer pySchedVisu sur une machine?
- Comment paramétrer pySchedVisu ?
- Comment créer une nouvelle version de pySchedVisu
- Comment faire pour exécuter pySchedVisu de manière récurrente?

Qu'est-ce que **pySchedVisu** ?

- **pySchedVisu** est un logiciel de génération de rapport de planning des scanners PET et SPECT du service de médecine nucléaire du CHUV.
- Les rapports sont générés chaque semaine (ou chaque deux semaines) et contiennent l'utilisation des scanners, la durée et le nombre d'examens, etc.
- Ces rapports sont utiles pour voir l'utilisation *effective* des scanners (les images qui ont été vraiment prises), comparé à leur utilisation *planifiée*. Certaines statistiques d'utilisation (durée moyenne d'examen, nombre de trous, etc.) sont également utiles.

Que fait **pySchedVisu** ?

pySchedVisu fonctionne en 3 étapes:

1. **pySchedVisu** récupère les données nécessaires à la création des rapports dans le **PACS** (base de donnée d'images) du CHUV.
2. **pySchedVisu** traite ensuite ces données pour en extraire des informations pertinentes (heure de début et fin de chaque examen, type d'examen, etc.).
3. **pySchedVisu** condense et affiche finalement ces données sous la forme d'un rapport envoyé par e-mail.

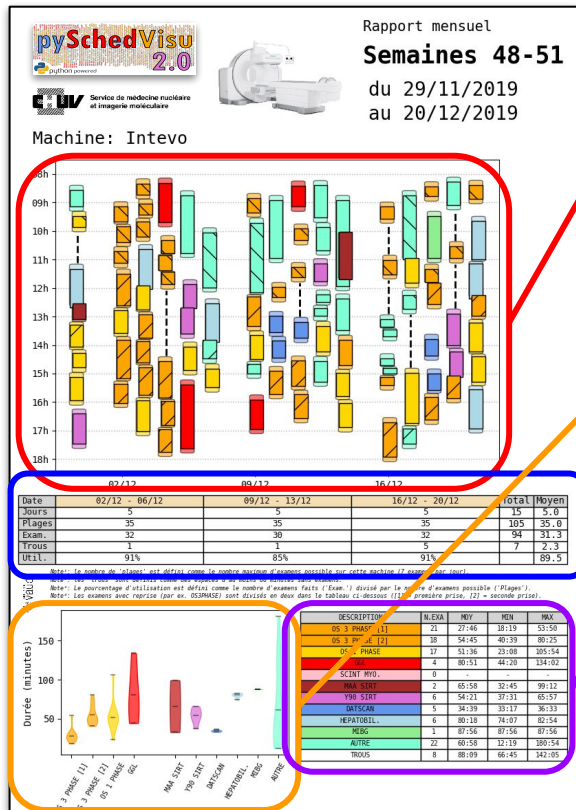
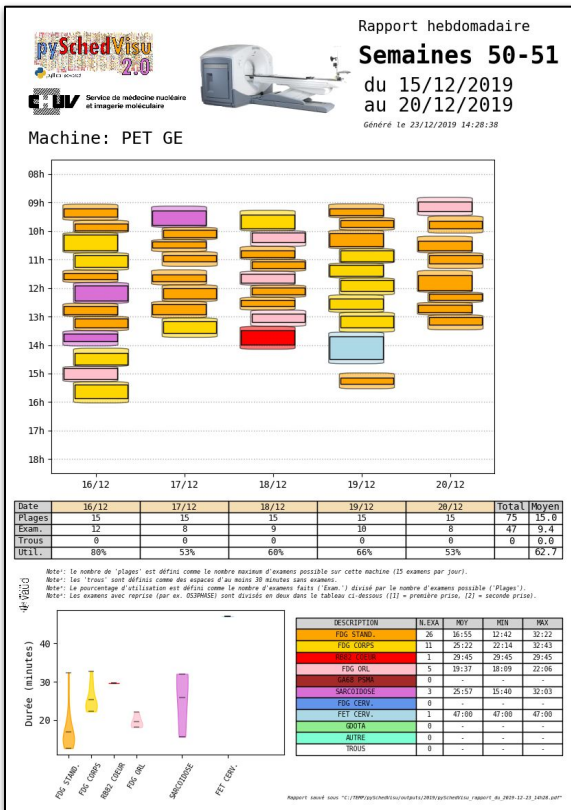
Que contient un rapport de **pySchedVisu** ?

Chaque rapport contient plusieurs pages, avec une page par machine et par “fréquence”. Il y a en ce moment 5 machines, et 7 “fréquences” (hebdomadaire, bimensuel, mensuel, trimestriel, semestriel, annuel et longue durée). Il y a donc 35 pages en tout.

Chaque page du rapport contient 4 sections principales:

1. Le **planning** “effectif”, sous forme d’agenda avec des rectangles représentant chaque examen.
2. Un tableau de statistiques de **comptage** pour chaque jour/semaine/mois/année.
3. Un tableau de statistiques sur la **durée** de chaque type d’examen
4. Un graphique de type “**violin plot**”, représentant visuellement les durées de chaque type d’examen.
5. Des **informations générales** sur le type de rapport (nom de la machine, dates, etc.)

Exemple de rapports pySchedVisu



1. Planning effectif

2. Tableau de comptage

3. Violin plot

4. Tableau de statistiques des durées d'examen

Comment fonctionne **pySchedVisu** ? (1)

pySchedVisu est un logiciel Python qui est exécuté chaque semaine à l'aide du Planificateur de Tâches de Windows. Il utilise également la librairie [dcm4che](#) afin de communiquer avec le PACS. **pySchedVisu** utilise également un certain nombre de modules Python afin de communiquer avec le PACS, de manipuler les données et de créer le rapport final. Ces modules sont listés dans le fichier `requirements.txt`.

Comme dit plus haut, **pySchedVisu** fonctionne en 3 étapes:

1. Récupération des données
2. Extraction des données
3. Création du rapport

Ces trois sections se trouvent dans 3 scripts Python différents et sont décrits plus bas.

Comment fonctionne pySchedVisu ? (2)

pySchedVisu exécute d'abord le script `main.py`, qui organise et coordonne l'exécution du programme entier, de la récupération des données à l'envoi de l'email contenant le rapport. Ce script appelle donc les autres scripts `retrieve_data.py`, `extract_data.py`, `create_report.py`. Le script `utils.py`, contient des fonctions d'aide.

La première étape de chaque étape ("`retrieve`", "`extract`", "`report`") consiste à déterminer les dates et jours qui doivent être traités. En effet, le programme ne traite pas à nouveau tous les jours / toutes les données à chaque exécution, mais il sauvegarde le résultat de chaque étape afin qu'il n'y ait plus besoin de les refaire. Ainsi, si un jour a déjà été traité, il est simplement recharger depuis le disque, sans être re-téléchargé ou re-traité.

pySchedVisu utilise un fichier de configuration (`config.ini`) et produit des `logs` qui peuvent être utiles pour comprendre les erreurs et obtenir des statistiques (nombre d'examens, temps d'exécution, etc.)

Comment fonctionne pySchedVisu ? (3)

pySchedVisu traite des données venant du PACS, dont la nomenclature peut être compliquée, c'est pourquoi il est nécessaire de donner quelques brèves définitions:

- Un **examen** (`study` dans le code) est l'ensemble des **séries** qui ont été faites pour un patient à une date donnée. Parfois, un patient a eu plusieurs examens dans la même journée, on parle alors de *reprise*.
- Une **série** (`series` dans le code) est un ensemble **d'images**, une partie d'examen effectuée par les scanners. Typiquement, un **examen** comporte plusieurs **séries** (CT de reconnaissance, examen PET, etc.) qui sont espacés entre eux, mais appartiennent au même **examen**.
- Une **image** (`image` dans le code) est un ensemble de pixels et de méta-données ("**headers**"). Certaines informations clés ne se trouvent qu'à ce niveau.
- Un **scanner** (`machine` dans le code) est un ensemble de scanners, souvent un ensemble PET-CT ou SPECT-CT.

Comment fonctionne la récupération des données de pySchedVisu ?

Les données nécessaires à la génération du rapport sont stockées dans le “header” des fichiers DICOMs, eux-mêmes stockés dans le [PACS](#). Afin de les récupérer, pySchedVisu effectue les étapes suivantes (script `retrieve_data.py`):

1. Pour chaque jour nécessaire, trouver toutes les examens de type CT, PT et NM effectués ce jour-là (fonction `find_studies_for_day`). Il y en a typiquement entre 25 et 50, toutes machines confondues.
2. Pour chaque examen, trouver toutes les séries qu'elle contient (fonction `find_series_for_studies`). Il y en a typiquement entre 100 et 400 par jour, toutes machines et examens confondus.
3. Pour chaque série, trouver les informations nécessaires (fonction `fetch_info_for_series`):
 - a. Télécharger les fichiers [DICOM](#) associés à chaque série. En fonction de la modalité, ceci peut être un ou plusieurs fichiers, car pour les séries CT & PT, il est suffisant de trouver la première et la dernière image d'une série, alors que pour les séries NM, tout la série doit être téléchargée.
 - b. Lire et stocker les informations nécessaires se trouvant dans le header du fichier DICOM. Les données sont stockées sous forme de DataFrame ([pandas](#)) dans des fichiers binaire [Pickle](#).

Comment fonctionne l'extraction des données de pySchedVisu ?

Les données de toutes les séries sont maintenant stockées pour chaque jour dans un fichier `XXXXX.pkl` (un fichier par jour, dossier `C:\TEMP\pySchedVisu\data`).

La prochaine étape consiste à réunir toutes les séries en un grand DataFrame et les regrouper en examens. Pour ce faire, pySchedVisu effectue les étapes suivantes (script `extract_data.py`):

1. Charger les séries de chaque jour (fonction `load_data_from_files`)
2. Créer un consensus sur le noms des scanners (fonction `mark_machine_group`)
3. Marquer les reprises (fonction `mark_retakes`)
4. Regrouper les séries en examen (`df_series` devient `df_studies`)
5. Créer un consensus sur les descriptions des examens (fonction `create_description_consensus`)
6. Ajouter des annotations aux examens (fonctions `add_preparation_times` et `add_time_to_prev_and_next`)
7. Stocker les deux DataFrame de séries et d'examens (sous `df_series.pkl` et `df_studies.pkl`)

Comment fonctionne la création du rapport de pySchedVisu ?

Tous les examens sont maintenant stockés ensemble dans le fichier `studies.pkl` (dossier `C:\TEMP\pySchedVisu\data`). Ces examens doivent maintenant être affichés dans un rapport ayant une page par scanner et par “fréquence” (rapport hebdomadaire, mensuel, etc.). Pour ce faire, **pySchedVisu** effectue les étapes suivantes (script `create_report.py`):

1. Déterminer les dates de début et fin pour chaque “fréquence” (fonction `create_report`)
2. Créer un fichier PDF multi-pages (module `PdfPages` de `matplotlib`)
3. Pour chaque machine
 - a. Pour chaque “fréquence”
 - i. Créer une page contenant:
 1. Les informations de la page (fonctions `create_header` et `create_notes`)
 2. L’agenda `create_schedule`)
 3. Le tableau de comptage sous l’agenda (fonction `create_daily_table`)
 4. Le “violin” plot (fonction `create_violin`)
 5. Le tableau des statistiques (fonction `create_stat_table`)

Où est-ce que **pySchedVisu** s'exécute ?

Actuellement (31 décembre 2019), **pySchedVisu** est installé comme tâche hebdomadaire sur le poste HOS 51499. **pySchedVisu** s'exécute tous les samedis matins à 01:00 du matin.

Afin que l'exécution s'effectue correctement chaque semaine, il est nécessaire que la machine ne soit pas éteinte et que l'utilisateur qui est inscrit comme propriétaire de la tâche dans le planificateur (actuellement David Viertl) doit avoir une session ouverte sur la machine.

Où se trouve pySchedVisu ?

Le code de pySchedVisu se trouve dans le dépôt de code GitHub suivant:



<https://github.com/blaurenczy/pyschedvisu>

Le programme compilé (exécutable sans Python) se trouve également dans divers dossiers du répertoire NUC:

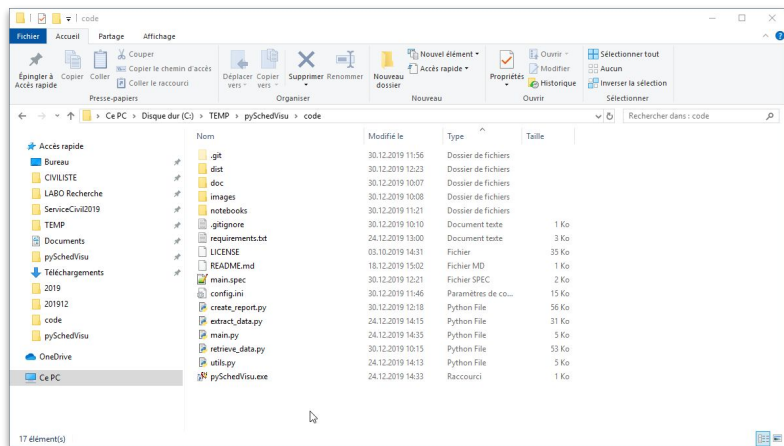
1. N:\NUC\COMMUN\CIVILISTE\SchedVisu\pySchedVisu
2. N:\NUC\NUC_QUALITE\pySchedVisu

Le code et le programme compilé se trouvent également dans le dossier C:\TEMP\pySchedVisu du poste où le programme s'exécute actuellement (2019-12-31)

Comment installer **pySchedVisu** sur une machine?

Pour installer (déployer) **pySchedVisu**, il vous faut effectuer les étapes suivantes:

1. Récupérer la dernière version de **pySchedVisu**, sous forme de fichier ZIP s'appelant **pySchedVisu_v2_1.zip** (ou une autre version plus récente). [Voir ici](#) pour plus d'informations.
2. Extraire le logiciel dans votre dossier **C:\TEMP** de manière à ce que le dossier contenant tous les fichiers soit appelé **C:\TEMP\pySchedVisu**.
3. Modifier le fichier de configuration (**config.ini**) afin que les paramètres correspondent à la nouvelle machine (paramètres de connexion au PACS, chemins d'accès aux fichiers, etc.). [Voir ici](#) pour plus de détails (et [surtout ici](#)).
4. Créer une tâche dans le Planificateur de Tâches Windows. [Voir ici](#) pour plus de détails.



Comment paramétrer pySchedVisu ?

pySchedVisu fonctionne grâce à un fichier de configuration (`config.ini`), grâce auquel un grand nombre de paramètres peuvent être changés. Ainsi, de nombreux changements peuvent être faits, sans avoir besoin de toucher au code Python (et donc sans avoir besoin de re-compiler ou de changer l'exécutable).

Tous les paramètres du fichier de configuration sont documentés dans le fichier lui-même. Certains paramètres méritent toutefois une explication plus détaillée. Ils sont décrits dans les pages qui suivent.

Comment paramétrer **pySchedVisu** ? (main 1)

```
[main]
```

```
report_range = hebdomadaire,bimensuel,mensuel,trimestriel,semestriel,annuel,longueduree
```

pySchedVisu crée un rapport pour différentes périodes de temps (la dernière semaine, le dernier mois, etc.). Ce paramètre permet de changer lesquelles de ces périodes se trouvent dans le rapport. Par exemple, en utilisant la ligne suivante:

```
report_range = hebdomadaire,mensuel,annuel
```

Seul le rapport de la dernière semaine, du dernier mois, et de l'année en cours seront inclus dans le rapport.

Note: afin de définir de nouvelle période de temps (les 3 dernières semaines par exemple), des modifications au code doivent être effectués. Ce paramètre permet seulement d'inclure ou non des périodes déjà définies.

Comment paramétrer **pySchedVisu** ? (main 2)

```
[main]  
  
report_year_start = 2016
```

pySchedVisu récupère les données depuis le début de l'année 2016. Ce paramètre permet de changer l'année à partir de laquelle les données sont récupérées et incluses dans le rapport.

*Note: Lors du changement de ce paramètre vers une date antérieure à 2016, des nouvelles données peuvent être requise (par exemple tout 2015), ce qui peut surcharger le PACS, ainsi que potentiellement nécessiter un supplément de temps d'exécution de **pySchedVisu**.*

Comment paramétrer **pySchedVisu** ? (main 3)

```
[main]
```

```
debug_level = WARNING
```

pySchedVisu crée des fichiers de logs (normalement stockés dans `C:/TEMP/pySchedVisu/logs`). Ces logs contiennent uniquement les informations les plus importantes sur l'exécution du logiciel. Des logs plus ou moins détaillés peuvent être obtenus en changeant ce paramètre pour un autre niveau (`DEBUG`, `INFO`, `WARNING`, `ERROR`)

*Note: un logging de bas niveau (`DEBUG` ou même `INFO`) peut causer une augmentation significative de la taille des fichiers de logs, ainsi qu'un potentiellement ralentissement de l'exécution de **pySchedVisu**.*

Comment paramétrer pySchedVisu ? (email)

```
[email]
# HTML content of the email
body = ...
# subject of the email
subject = Rapport hebdomadaire pySchedVisu
# comma separated list of recipients
recipients_email = Prenom.Nom@chuv.ch,exemple@gmail.com
```

pySchedVisu envoie les rapports par e-mails. Ces paramètres permettent de changer le sujet, le texte et les destinataires de ces e-mails.

Les destinataires doivent être spécifiés comme une liste séparée par des virgules, sans espaces inutiles.

Le texte de l'e-mail peut contenir des balises HTML. Il contient également des balises de remplacement de texte (par exemple `{__REPORT_PATH__}`) qui ne doivent pas être changées (à moins d'être certain de comprendre ce que l'on fait).

Comment paramétrer pySchedVisu ? (path)

```
[path]
dcm4che_path = C:/TEMP/pySchedVisu/dcm4che-5.19.1/bin
dicom_temp_dir = C:/TEMP/pySchedVisu/DICOMs
log_dir = C:/TEMP/pySchedVisu/logs
data_dir = C:/TEMP/pySchedVisu/data
studies_db_save_path = C:/TEMP/pySchedVisu/data/studies.pkl
series_db_save_path = C:/TEMP/pySchedVisu/data/series.pkl
output_dir = N:\NUC\NUC_QUALITE\pySchedVisu
```

pySchedVisu stocke des fichiers (logs, DICOMs, Pickle, etc.) et utilise des ressources (dcm4che). Le logiciel a besoin de savoir où ces éléments se trouvent, mais le logiciel n'a pas nécessaire qu'ils soient toujours tous au même endroit prédéfini. Il est donc **possible de changer n'importe lequel de ces chemins**, *pour peu qu'il existe et qu'il soit possible d'y accéder*.

Veuillez respecter les conventions de notations: pas de / final à la fin des chemins des dossiers et n'utilisez des \ que si c'est nécessaire.

Comment paramétrer **pySchedVisu** ? (PACS)

```
[PACS]
local_ae_title = CIVILISTENUC2
local_host = 155.105.54.51
local_port = 104
remote_ae_title = csps1FIR
remote_host = 155.105.3.105
remote_port = 104
```

pySchedVisu a besoin de se connecter au PACS afin de télécharger les fichiers DICOMs contenant les données nécessaires pour créer le rapport. **Les paramètres de connexions sont différents pour chaque machine!** Ainsi, il est nécessaire de récupérer/établir ces paramètres pour chaque nouvelle installation. Veuillez entrer en contact avec le service de PACS TRM de Radiologie pour obtenir ces identifiants pour une nouvelle machine (*rad.trm.pacs at chuv.ch*).

Comment paramétrer **pySchedVisu** ? (machines)

Comment paramétrer **pySchedVisu** ? (draw)

Comment créer une nouvelle version de pySchedVisu

Comment faire pour exécuter **pySchedVisu** de manière récurrente?