# Guide d'utilisation des scripts de préparation des données pour l'utilisation de l'interface Parcours patients

## Table des matières



## Introduction

Ce document est un guide pratique pour l'utilisation des scripts de préparation des données en vue de les utiliser dans l'interface d'analyse du parcours des patients. Il est complémentaire du document "Tutoriel" qu'il est recommandé de lire en amont afin de comprendre le contexte général et les raisonnements sous-jacents.

Des jeux de données fictifs sont fournis pour tester les scripts. Ces données sont fictives, les numéros d'identification des patients, des unités médicales et les dates ont été recodés. Il est normal de ne pas obtenir le même résultat qu'avec les données fournies en exemple pour le fonctionnement de l'interface.

Pour préparer votre propre base de données, vous allez avoir besoin des deux scripts fournis, codés sur Python 3.6, mais également de la librairie Java SPMF. Vous pouvez télécharger cet outil sous forme d'exécutable ou directement son code source sur la page dédiée :

http://www.philippe-fournier-viger.com/spmf/index.php?link=download.php

L'organisation des scripts entre eux et de l'outil est illustrée dans la figure 1.



Lucile TRUTT
2018

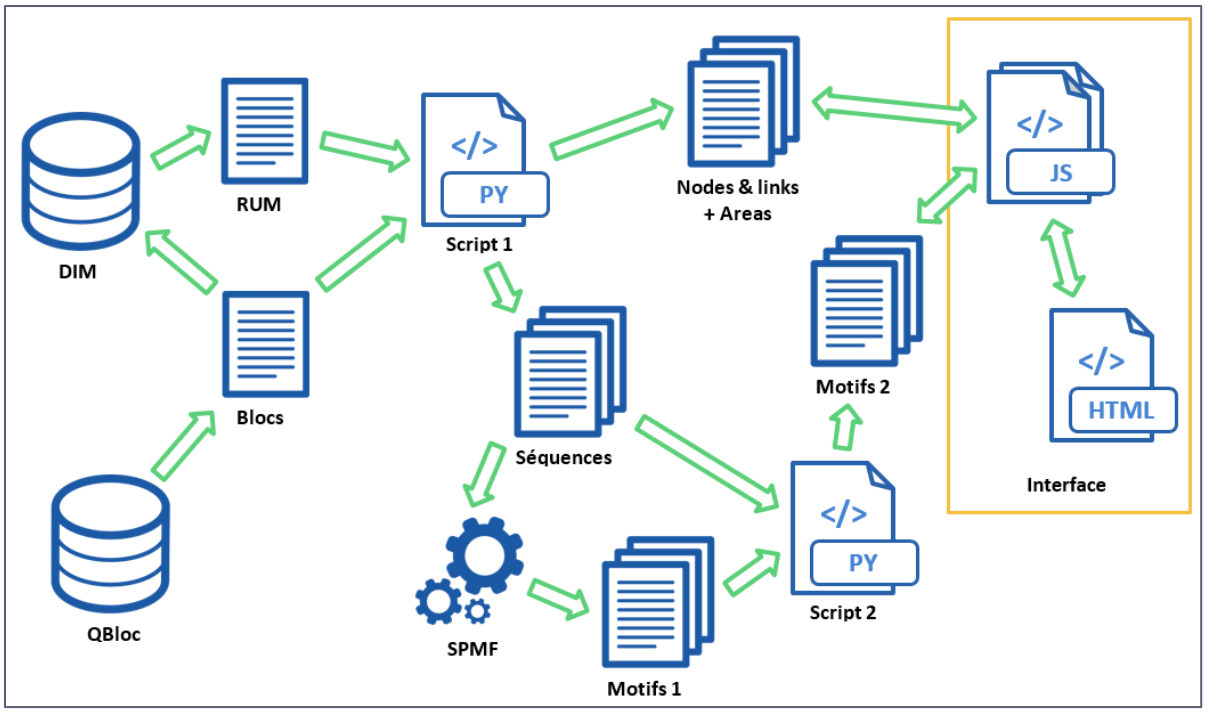


**Figure 1 : Organisation des scripts et des fichiers lors de l'étape de préparation des données et de leur utilisation par l'interface graphique**

# Données de départ

L'originalité de ce travail est de fonctionner à partir de deux sources :

- Base de données médico-administratives de centre hospitalier (DIM),
- Logiciel métier de gestion des blocs opératoires (QBloc).

Le travail ayant fait suite à une demande de l'équipe de chirurgie thoracique et cardio-vasculaire (CTCV), le premier jeu de données est extrait de la base QBloc, appelé "Blocs". De cette extraction sont récupérés les Identifiants Permanents Patients (IPP) et les dates-heures (DH) de présence au bloc opératoire des patients.

Une première étape de manipulation des données, non détaillée ici car très simple, a permis de les mettre en forme pour être cohérentes avec celles qui seront extraites de la base DIM en considérant le bloc opératoire comme une unité médicale (UM) à part entière et ne faisant pas partie du service de CTCV :

- Création de codes "0000" et de libellés "Bloc CTCV" fictifs pour correspondre aux numéros d'UM, de service, de pôle et de type d'autorisation PMSI,
- Organisation des colonnes comparable à la future extraction PMSI :

| Année | IPP | Code UM | Lib. UM | DH entrée UM | DH sortie UM | Code autoris. | Lib. autoris. | Code service | Lib. service | Code pôle | Lib. pôle |
|---|---|---|---|---|---|---|---|---|---|---|---|

Ce jeu de données est ensuite utilisé comme référence pour réaliser une requête sur la base DIM. A partir de la base hospitalière des Résumés d'Unité Médicale (RUM), ont été extraites les lignes pour lesquelles :

- Les IPP correspondent à ceux du fichier Blocs,
- Les années PMSI des séjours (année du jour de sortie) corresponde à l'année de l'intervention chirurgicale, l'année précédente ou l'année suivante.

Cette extraction a ensuite été enrichie des codes et libellés de service, de pôle et type d'autorisation PMSI de chacune des UM. Les entrées et sorties des UM sont également sous la forme de dates-heures. La structure des données de sortie du fichier "RUM" est la suivante :

| Année | IPP | Code Lib. UM | DH entrée UM | DH sortie UM | Code autoris. | Lib. autoris. | Code Lib. service | Code Lib. pôle |
|---|---|---|---|---|---|---|---|---|

(Nous n'avons pas fait de sélection plus fine sur les numéros de séjour car le CHU de Nantes étant un établissement multi-sites géographiques, il était souhaitable pour ce travail de ne pas considérer des transferts comme un nouveau séjour tel que recommandé depuis 2016. Nous avons choisi arbitrairement de prendre les séjours un an avant et après pour augmenter les chances de récupérer la totalité des séjours concernés tout en limitant la taille du jeu de données pour garder un temps d'exécution raisonnable. Ces délais peuvent tout à fait être modifiés selon vos propres critères.)

Les scripts utilisent les numéros de colonnes plutôt que leur nom, vous pouvez donc les nommez comme vous le souhaitez ou utiliser d'autres niveaux d'agrégation, mais veillez à conserver une structure semblable.

# Script 1

Les détails des manipulations de données et des structures sont donnés en commentaires au fur et à mesure dans le script. Des affichages réguliers dans la console permet de surveiller l'avancée du script, de connaître l'étape en cours, d'avoir un aperçu du format des données à ce moment et de ressortir des erreurs des algorithmes que python ne peut pas reconnaitre.

Les listes des années et des niveaux d'agrégation sont précisées en tout début de script. Vous pouvez les modifier pour que les résultats soient cohérents avec vos propres données.

De façon succincte, les principales étapes du script 1 sont :

- Préparation :
  - Import et remise en forme des bases RUM et Blocs,
  - Création des venues (séjours éventuellement multi-sites) et tri des venues comportant une intervention chirurgicale en CTCV,
  - Concaténation des bases RUM et Blocs, découpage des RUM à cheval et nettoyage,
  - Export de la liste des libellés correspondants aux différents codes "lsLib.json".
- Module 1 (*325*):
  - Création des positions : code UM + rang par rapport au premier passage au bloc opératoire pour la première version, code UM + rang par rapport au dernier passage au bloc opératoire pour la seconde,

    - Compte des effectifs et de la durée moyenne de séjour (DMS) à chaque position,
    - Agrégation et mise en forme des fichiers JSON "links" et "nodes".
- Module 2 (*633*):
    - Création des séquences : suite des codes UM et des durées des RUM pour chaque venue.
    - Export au format SPMF des fichiers "sqc-…" et au format JSON des fichiers "sqc-…duree"
- Module 3 (*714*):
    - Agrégation des UM peu rencontrées (seuil de 5% des venues, modifiable) dans une unité "Autre",
    - Compte des UM de chaque journée à l'aide d'indices par rapport au jour d'entrée
    - Export des 90 premiers jours sous forme de csv "area…"

Les fichiers "nodes", "links" et "lsLib" sont uniques pour toute la base de données. Les fichiers "sqc-…" et "area…" sont séparés en fichiers individuels par année et par niveau d'agrégation pour permettre leur utilisation indépendante par SPMF pour les premiers, et pour améliorer le temps d'exécution des scripts de l'interface pour les seconds.

Des manipulations supplémentaires sont laissées dans le script en commentaires pour permettre la création d'autres jeux de données pour des utilisations externes à l'interface de visualisation ou des calculs différents :

- Export de la liste des RUM (*145*) ou des RUM et blocs après tri et nettoyage (*243*),
- Repérage des RUM de moins de 1, 2 ou 3 heures (*225*)
- Indice de largeur des nœuds du diagramme de Sankey directement proportionnel à la DMS plutôt que par classes (*446*)
- Export de la structure administrative des UM concernées (*735*),
- Export de la base agrégée comme pour le module 3 mais avec un compte par heures plutôt que par jours, avec ou sans regroupement dans l'unité Autre (*851*).

Les fichiers "nodes", "links", "area…" et "lsLib" pourront être directement utilisés par les scripts JavaScript de l'interface pour les modules 1 et 3 après avoir été placés dans le dossier "db". Les séquences pour le module 2 nécessitent d'autres manipulations telles que décrites ci-après.

# Préparation du module 2

## SPMF

Nous décrirons ici l'utilisation de SPMF par l'intermédiaire de son interface graphique. Si vous souhaitez l'intégrer dans un outil complet ou que vous préférez utiliser la librairie directement, vous pouvez utiliser le code source, mais cette option ne sera pas détaillée ici.

L'algorithme à utiliser est celui appelé HirateYamana, dans la catégorie sequential pattern mining. Celui-ci permet d'intégrer une contrainte temporelle de distance entre les items. De cette manière, les motifs ne concernent que des positions directement adjacentes dans le parcours et représentent ainsi des mouvements réels. N'ayant qu'un seul item par lot ("itemset"), le paramétrage sera simplement de minimum = 0 et maximum = 1 pour la distance entre items. Sauf limite du nombre de mouvements demandée expressément, la longueur du motif peut être réglée de 0 à 100 pour tous les récupérer. Il

n'y a pas de support minimum à demander ni de filtre à appliquer, il sera fait ultérieurement et tous les parcours doivent être pris en compte pour les calculs effectués dans le script suivant.

Réalisez l'analyse pour chacun des fichiers "sqc-…" (sans extension et sans "duree" à la fin) et enregistrez les résultats de la même façon, dans des fichiers "spm-…" sans extension, en ajoutant le niveau d'agrégation et l'année dans chacun des noms. (Fichiers "Motifs 1" sur le schéma.)

Synthèse des paramètres à indiquer :

- Algorithm : HirateYamana
- Input file : "sqc-…"
- Output file : "spm-…"
- Minsup : 0
- Min time interval : 0
- Max time interval : 1
- Min whole time interval : 0
- Max whole time interval : 100 (modifiable)

## Script 2

Ce script récupère 3 types de fichiers :

- Les fichiers "spm-…" construits par SPMF
- Les fichiers "sqc-…duree" construits par le premier script python
- Le fichier "lsLib.json" construit par le premier script python (attention si vous l'avez déjà placé dans les dossiers de l'interface)

Les détails des manipulations de données et des structures sont donnés en commentaires au fur et à mesure dans le script. Des affichages réguliers dans la console permet de surveiller l'avancée du script, de connaître l'étape en cours, d'avoir un aperçu du format des données à ce moment et de ressortir des erreurs des algorithmes que python ne peut pas reconnaitre.

Les listes des années et des niveaux d'agrégation sont précisées en tout début de script. Vous pouvez les modifier pour que les résultats soient cohérents avec vos propres données.

De façon succincte, les principales étapes du script 2 sont :

- Calculs :
    - De la durée moyenne de séjour dans chacune des unités du motif,
    - Du nombre d'apparitions du motif ("occurrence", différent du "support" donné par SPMF et correspondant au nombre de venues où le motif apparait),
    - De la "confidence" de règles d'association au sein des motifs
- Filtrage des motifs sur la présence du bloc
- Export au format JSON des fichiers "motifs-…"

Les fichiers ainsi créés peuvent être placés dans le dossier "db" de l'interface pour être utilisés dans le module 2. (Fichiers "Motifs 2" sur le schéma.)