Bonnes pratiques en ingénierie de données en radio-oncologie

Mémoire

Gabriel Couture

Maîtrise en physique - avec mémoire
Maître ès sciences (M. Sc.)

Québec, Canada

Bonnes pratiques en ingénierie de données en radio-oncologie

Mémoire

Gabriel Couture

Sous la direction de:

Philippe Després, directeur de recherche

Résumé

Les travaux présentés dans ce mémoire visent à identifier et appliquer de bonnes pratiques quant à la gestion de données en santé, et plus précisément en radio-oncologie. Ce domaine comporte de nombreux défis en lien avec les données dont l'augmentation rapide du volume, de la variété et de la complexité des données. C'est pourquoi les développements en lien avec la gestion de données en santé doivent s'appuyer sur de bonnes pratiques d'ingénierie de données. Trois projets distincts en lien avec les données ont été abordés dans le cadre de ce mémoire.

Le premier concerne l'automatisation de la collecte de données en radio-oncologie. Un pipeline a été développé afin d'obtenir quotidiennement les indices dosimétriques des traitements de curiethérapie de prostate faits dans la journée. Ces indices sont ensuite stockés dans une base de données dédiée à la recherche sur le cancer de la prostate. Ces indices peuvent être obtenus par deux algorithmes de calcul de DVH. Une comparaison a été faîte avec un jeu de données de 20 cas de curiethérapie HDR de prostate. Celle-ci a permis d'identifier des différences entre chacun des algorithmes.

Le deuxième projet montre comment il est possible de concevoir des jeux de données massifs réutilisables dédiés aux analyses radiomiques. Des flots de travail permettant de conserver des données coûteuses générées dans le cadre d'analyses radiomiques ont été conceptualisés et implémentés. Ces flots, inspirés des principes FAIR, permettent d'assurer une meilleure traçabilité et de tendre vers des jeux de données réutilisables. Un flot qui permet à un spécialiste (ex. radio-oncologue) de tracer des segmentations a été implémenté et testé avec des logiciels libres, notamment le serveur DICOM *Orthanc* et *3D Slicer*.

Le dernier projet démontre l'apport de l'ingénierie de données en médecine personnalisée. Plus précisément, l'estimation des risques de cancer du sein pour des participantes à une étude d'envergure ont été obtenus par l'entremise de processus automatisés. Dans le cadre d'une étude sur le cancer du sein impliquant près de 2000 participantes, deux pipelines ont été développés. Le premier permet d'obtenir le risque de cancer du sein individuel des participantes en fonction de différents facteurs (habitudes de vie, historique familiale, marqueurs génétiques). Le deuxième pipeline génère des lettres personnalisées destinées aux participantes ainsi qu'à leur médecin traitant.

Ces projets démontrent la pertinence de bonnes pratiques quant à la gestion de données en santé. L'ingénierie de données présentée dans ce mémoire aura permis d'automatiser plusieurs opérations en lien avec les données en plus de concevoir des jeux de données réutilisables. Cette bonne gestion de données pave la voie vers de nouvelles pratiques et rend les activités scientifiques en santé plus efficaces.

Abstract

This work aims to identify and apply good practices in the management of health data, and more specifically in radiation oncology. This field has many data-related challenges including the rapidly increasing volume, variety and complexity of data. This is why developments related to health data management must be based on good data engineering practices. Three distinct data-related projects have been addressed in this thesis.

The first concerns the automation of data collection in radiation oncology. A pipeline has been developed to obtain daily dosimetric indices of prostate brachytherapy treatments performed during the day. These indices are then stored in a database dedicated to prostate cancer research. These indices can be obtained by two DVH calculation algorithms. A comparison was made with a dataset of 20 HDR prostate brachytherapy cases. This made it possible to identify the differences of each of the algorithms.

The second project shows how it is possible to design massive reusable datasets dedicated to radiomics analyses. Workflows to retain expensive data generated in radiomics analyzes have been conceptualized and implemented. These workflows, inspired by the FAIR principles, ensure better traceability and tend towards reusable data sets. A workflow that allows a specialist (e.g. radio-oncologist) to draw segmentations has been implemented and tested with free software, in particular with the DICOM server *Orthanc* and *3D Slicer*.

The last project demonstrates the contribution of data engineering in personalized medicine. Specifically, the breast cancer risk assessment of a large group of participants were obtained through automated processes. As part of a breast cancer study involving nearly 2000 participants, two data pipelines were developed. The first provides participants' individual breast cancer risk assessment based on various factors (lifestyles, family history, genetic markers). The second pipeline generates personalized newsletters for participants and their treating physician.

These projects demonstrate the relevance of good practices in health data management. The data engineering presented in this thesis will have made it possible to automate several data related operations in addition to designing reusable data sets. This good data management paves the way for new practices and makes health science activities more efficient.

Table des matières

Liste des tableaux

Liste des figures

Acronymes

AAPM Association Américaine des Physiciens en Médecine – *American Association of Physicists in Medicine* en anglais (AAPM) 26

AE *Application Entity* (AE) 18

API interface de programmation applicative – *Application Programming Interface* (API) en anglais – 20

AWS *Amazon Web Service* (AWS) 9

DAG graphe orienté acyclique – *Directed Acyclic Graph* (DAG) en anglais – 29

DICOM *Digital Imaging and Communications in Medicine* (DICOM) 2

DVH histogrammes dose-volume – *Dose-Volume Histogram* (DVH) en anglais – 24

GRPM Groupe de recherche en physique médicale de l'Université Laval (GRPM) 17

HDR haut débit de dose – *High Dose Rate* (HDR) – 24

HL7 *Health Level 7* (HL7) 3

OCB *Oncentra Brachy* (OCB) 28

OCP *Oncentra Prostate* (OCP) 26

PACS système d'archivage et de transmission d'images – *Picture Archiving and Communication System* (PACS) en anglais – 18

PRS score de risque polygénique, *Polygenic Risk Score* (PRS) en anglais, ... 49

REST *Representational State Transfer* (REST) 20

SCP *Service Class Provider* (SCP) 18

SCU *Service Class User* (SCU) 18

TPS logiciels de planification de traitement, *Treatment Planning System* (TPS) en anglais 13

VR représentation de donnée, *Value Representation* (VR) en anglais 15

Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius — and a lot of courage to move in the opposite direction.

Ernst Friedrich Schumacher

Remerciements

L'aventure qu'aura été cette maîtrise n'aura pas été simple, et sa complétude aura été possible grâce à plusieurs personnes. D'abord, je tiens à remercier mon directeur de recherche, Philippe Després, qui m'a offert cette possibilité qu'est la maîtrise, de la confiance qu'il m'a témoignée et de m'avoir donné de grandes libertés dans la poursuite de mes travaux de recherche. Les compétences et les connaissances que j'ai acquises au long de ce parcours me sont très précieuses, et je n'ai aucun doute de leur pertinence dans l'avenir.

Je tiens aussi à remercier Yannick Lemaréchal pour m'avoir aidé à terminer ce mémoire. Ses nombreuses lectures, révisions et commentaires m'auront permis de terminer cette maîtrise. Je remercie également Cédric Bélanger pour sa précieuse assistance par rapport à la fastidieuse tâche qu'est la collecte de données de traitement de curiethérapie. Je remercie pareillement Annie, Laurence, Jacques et Michel pour leurs encouragements et leurs commentaires en lien avec le projet PERSPECTIVE I&I.

Je tiens également à remercier les membres du groupe de recherche en physique médicale, qui m'ont assisté tout au long de mon parcours, mes amis, qui ont su m'écouter dans les moments difficiles, et mes proches, qui m'ont encouragé jusqu'à la fin. Finalement, je tiens à remercier ma meilleure amie et conjointe, Jasmine, de m'avoir épaulé et encouragé dans cette épreuve.

Introduction

Étymologiquement, l'informatique découle de l'information. Le terme «informatique» a été introduit en 1957 par Karl Steinbuch, un ingénieur allemand, dans l'essai *Informatik : Automatische Informationsverarbeitung* (Informatique : Traitement automatique de l'information en français) [1]. Il y est défini comme la manipulation d'informations. Un autre terme pour définir le domaine, *Datalogi*, a été introduit par le danois Peter Naur en 1966 [2]. Il a défini *Datalogi* comme étant la science de l'utilisation et du traitement logique de données. Les définitions données à ces termes suggèrent tous que l'information, c'est-à-dire la donnée, se trouve au coeur du domaine. Donald Knuth, professeur d'informatique et auteur de l'oeuvre légendaire *The Art of Programming*, souligne également que le domaine de l'informatique a en son coeur les données et leur manipulation [3]. En bref, les fondateurs et les experts du domaine jugent que l'innovation en informatique est intimement liée à la capacité de gérer les données, c'est-à-dire par l'augmentation de la capacité à les stocker et à les traiter.

Depuis quelques années, l'augmentation rapide et continue du volume et de la variété des données [4] a imposé un changement de paradigme chez les institutions et les entreprises. En effet, ces dernières se doivent de déployer des stratégies et des ressources afin d'adéquatement collecter, traiter et stocker les données. Nous pouvons d'ailleurs supposer qu'un grand nombre d'entreprises multinationales ayant eu du succès ces dernières décennies le doivent à leur capacité à gérer des quantités massives et variées de données quelle que soit la localisation géographique [5]. En effet, des entreprises tels que les GAFAM (Google, Apple, Facebook, Amazon et Microsoft) sont de bons exemples. Facebook réussit à gérer une quantité absolument massive de données : en date de 2019, à chaque minute, plus d'un demi-million de commentaires sont écrits, 293 000 statuts sont mis à jour et 136 000 photos sont chargées [6]. Toutes ces informations sont adéquatement stockées et accessibles dans la seconde à partir de n'importe quel ordinateur sur terre ayant une connexion internet. Pour Google, c'est l'aptitude à classer et à structurer l'immense et chaotique étendue d'information du *world wide web* qui leur ont permis d'atteindre leur succès. L'essor de ces entreprises est intimement lié à leur capacité de gestion de grands ensembles de données, et laisse supposer que la capacité d'une grande institution à opérer efficacement est reliée à la qualité de sa gestion de données. Puisque la médecine moderne utilise abondamment des données, il est naturel de supposer que la gestion optimale de celles-ci sera un de ses principaux vecteurs d'amélioration [7], notamment par

l'entremise de la médecine personnalisée [8; 9].

Malgré le fait que nous soyons dans une ère de données et d'informations, le domaine de la santé peine à améliorer sa capacité à les gérer. Les institutions de santé font face à une complexité croissante des données, accentuant du même ordre la difficulté de structuration de ces dernières. Elles peuvent être structurées ou non, stockées à une multitude d'endroits différents, parfois dans des formats peu ou pas documentés. Un établissement de santé comporte typiquement des dizaines voire des centaines de systèmes d'information pour gérer les données, certains d'entre eux étant développés à l'interne tandis que d'autres sont achetés de fournisseurs privés. Règle générale, chaque établissement administre les données à sa façon [10].

C'est dans le contexte de l'amélioration de la gestion de données en santé, et plus précisément en radio-oncologie, que s'inscrit ce mémoire. Les bonnes pratiques quant au stockage et au traitement des données y sont explorées, commentées et testées. Une revue des connaissances actuelles en ingénierie de données est d'abord abordée, telles que les données massives, les principes FAIR [11] et l'architecture de systèmes de gestion de données. Le standard *Digital Imaging and Communications in Medicine* (DICOM) [12], fondamental aux données d'imagerie et de radio-oncologie, est présenté.

Des applications faisant usage des connaissances d'ingénierie de données sont ensuite présentées. D'abord, au chapitre 2, il est question d'un pipeline de données dosimétriques. Celui-ci aborde la problématique de la gestion et de la consolidation des données dosimétriques (c.-à-d. indices dosimétriques) issues de traitements de radiothérapie dans un département de radio-oncologie. Le chapitre 3, quant à lui, présente des flots de travail de radiomique inspirés des principes FAIR. Ceux-ci permettent la réutilisation de données coûteuses (annotations, tracés de contour, *etc.*). Plus précisément, un flot ayant été implémenté et testé est présenté. Finalement, dans le chapitre 4, il est montré comment les techniques en ingénierie de données facilitent l'accès à la médecine personnalisée. En effet, dans le cadre d'une étude à grande échelle, les risques de cancer du sein de près de 2000 femmes ont été calculés, en fonction de leurs habitudes de vie, de marqueurs génétiques et de leur historique familiale.

Chapitre 1

Connaissances et concepts pertinents liés à l'ingénierie de données en radio-oncologie

La gestion adéquate des données implique de nombreux concepts : l'identification des enjeux de gestion de données, organisation des données et systèmes de gestion de données. À cela s'ajoute des concepts importants à la gestion de données spécifiques à la radio-oncologie, tels que les standards DICOM et *Health Level 7* (HL7) [13; 14] et les outils les entourant. Ce chapitre vise à consolider ces concepts afin qu'ils puissent être référés dans les chapitres subséquents.

Les enjeux de la gestion de données ont été explorés par les spécialistes des données massives, qui les ont décortiqués en plusieurs sous-groupes : les V (volume, vitesse, variété, *etc.*). Ces derniers sont abordés en premier dans ce chapitre. Les principes FAIR sont ensuite présentés. Ces principes consistent en un guide de bonnes pratiques quant à la gestion de données de recherche. Ils abordent certains enjeux dont la façon de les rendre accessibles et repérables, sous quel format, et sous quelles conditions elles peuvent être réutilisées. La section suivante concerne les solutions logicielles de systèmes de gestion de données, et plus particulièrement de leur architecture logicielle. Les notions présentées permettent de guider l'élaboration de systèmes de gestion de données. Il est question des différents éléments qui constituent les systèmes de gestion de données et comment ils interagissent.

Les précédents concepts concernent, à toutes fins pratiques, toutes les entreprises de gestion de données en recherche. Il est cependant crucial de présenter les aspects spécifiques aux données en radio-oncologie. Le standard DICOM est d'une importance majeure dans le domaine et est présenté dans ce chapitre. En effet, ce standard définit le format de données utilisé par les différentes modalités d'imagerie médicale et de radio-oncologie, et comment ces données sont organisées permettant de les partager au sein d'applications logicielles. Dans ce contexte, le serveur applicatif *Orthanc* est présenté. Ce dernier exploite pleinement le format DICOM et

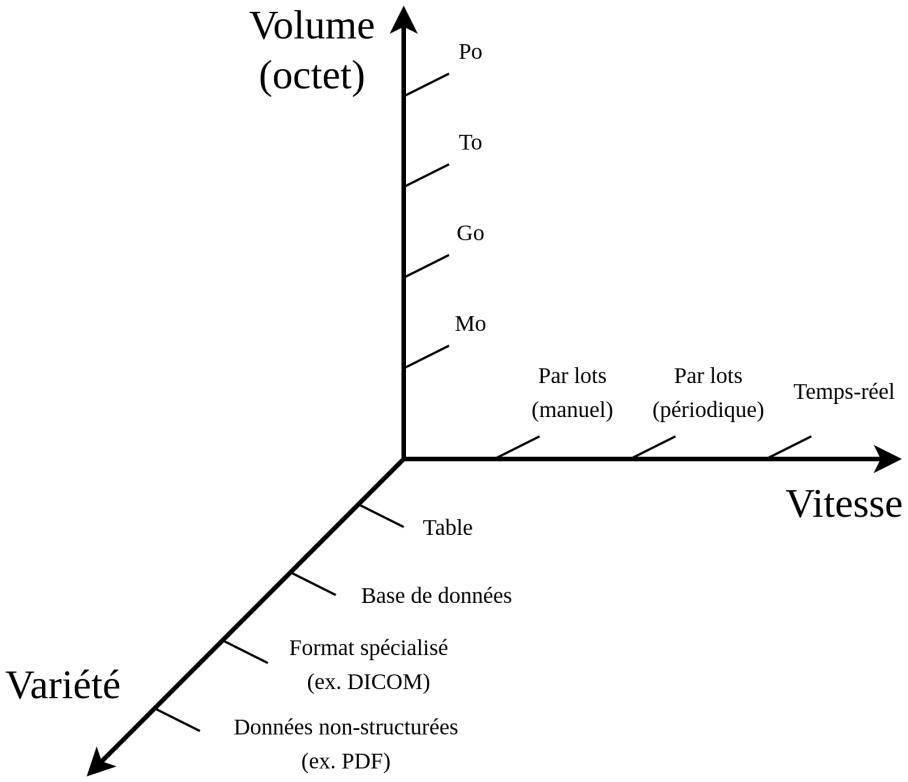


FIGURE 1.1 – Schéma des dimensions des données massives inspiré par les travaux de Laney [17].

permet de stocker, transmettre et valoriser les données.

1.1 Définition des données massives : les V

Le domaine des données massives, ou *big data* en anglais, a été défini de nombreuses façons au cours des dernières années. Dans les années 1990, la terminologie «données massives» référait aux problèmes impliquant des ensembles de données trop volumineux pour être traités [15] ou visualisés [16]. En 2001, Doug Laney, analyste chez *Gartner*, a publié un article nommé «*3-D Data Management : Controlling Data Volume, Velocity and Variety*». Pour Laney, un problème de gestion des données relève des données massives lorsqu'il soulève des enjeux au niveau de ce qu'il a appelé les 3 V : *Volume*, *Vitesse* de traitement et d'acquisition et *Variété* des données [17]. Un enjeu de gestion de données se positionne ainsi dans un espace en trois dimensions tel que montré à la figure 1.1. Chaque région de cet espace correspond à des choix de stratégies et d'outils. Également, plus un cas s'éloigne de l'origine, plus il est probable que des ressources importantes devront être consacrées à la gestion des données.

Volume

L'aspect volumétrique des données est la dimension la plus évidente. Il réfère aux difficultés de gestion d'ensemble de données volumineux, que ce soit au niveau de leur stockage ou de leur traitement. La dimension du volume de la figure 1.1 suggère que les défis croissent selon la quantité de données. Il est assez facile d'imaginer qu'un volume de quelques mégaoctets (Mo) de données se gère beaucoup mieux qu'un volume de plusieurs téraoctets (To). À partir d'un certain volume, le stockage et le traitement de données doivent être répartis sur plusieurs machines sous forme de grappes de serveurs (ou *cluster*). Un projet scientifique bien connu ayant été confronté à ces défis est le LHC (*Large Hadron Collider*). En effet, le système de stockage distribué pour les expériences du LHC contient plus de 200 Po de données brutes, sous la forme de plus de 600 millions de fichiers [18]. En plus d'être stockées, ces données doivent être facilement récupérables de n'importe quel endroit dans le monde afin d'être analysées. La gestion d'un tel ensemble de données représente de grands défis, pour lesquels le CERN a investi des ressources importantes [18], notamment en développant le système de stockage EOS [19]. Il s'agit d'un bel exemple démontrant l'importance – et la nécessité – de gérer adéquatement les données en fonction du volume.

Les solutions aux enjeux de volume sont généralement techniques et matérielles. Elles nécessitent des connaissances de technologies et d'infrastructures informatiques, mais peu du domaine d'application en tant que tel. Typiquement, le stockage et le traitement de grands volumes sont étalés sur plusieurs machines. Côté stockage, des technologies de stockage de fichiers, comme HDFS (*Hadoop Distributed File System*) [20], *GlusterFS* et *CephFS* [21], ou d'autres technologies de stockage d'objet (*Object-base Storage*) comme Ceph [22] permettent de stocker de très grands ensembles de données (plusieurs pétaoctets [21]). Plusieurs compagnies internationales ont d'ailleurs développé des applications de stockage afin de répondre à leurs besoins, notamment *Apache Cassandra* de *Meta* [23] et *BigTable* de *Google* [24]. Du côté du traitement, le parallélisme est à privilégier. Des technologies tel que Hadoop et Spark [25] permettent d'éclater le traitement de données sur plusieurs machines.

Vitesse

L'aspect de la vitesse réfère à l'enjeu du traitement et du stockage des données par rapport au temps. La dimension de la vitesse de la figure 1.1 présente trois niveaux de traitement de données. Chaque pas dans la graduation amène des défis et une demande de ressources et de connaissances supplémentaires. Notons également que l'aspect du traitement en parallèle des données est complémentaire à cette dimension.

Le premier pas est le traitement par lots manuels. Ceci réfère à un être humain qui démarre manuellement le processus de transfert ou traitement de données vers un système de stockage. Ce processus, qui est typiquement un script Python, Bash, ou autre, récupère, traite et stocke les données.

Le second pas dans la gradation de la vitesse est celui du traitement et du stockage par lots périodiques. Il s'agit de la même opération que la précédente, mais avec un outil de planification de tâche (*task scheduler* en anglais). Ces outils permettent de planifier le lancement d'un script de façon mensuelle, hebdomadaire, journalière, à toutes les heures, *etc.*. *Cron* [26], *Apache Airflow* [27] et *Luigi* [28] en sont quelques exemples. La figure 1.2 présente le fonctionnement de haut niveau d'un planificateur. La connaissance d'au moins un outil de planification de tâches est primordiale afin d'effectuer des traitements par lots périodiques. Il est aussi important d'intégrer la journalisation au sein des scripts exécutés périodiquement. En effet, puisqu'aucun être humain assiste à l'exécution, un système de journaux (*logs*) permet de retracer toute erreur ou anomalie. Par exemple, le planificateur *Airflow* permet de consulter à même sa plateforme les journaux des scripts. Un bon planificateur permet également l'envoi de notifications, tel un courriel, si une erreur se présente afin de signaler cette dernière à un intervenant. Parvenir à bien effectuer ces opérations par lots périodiques nécessite ainsi des connaissances supplémentaires que celles utilisées pour le traitement par lots manuels.

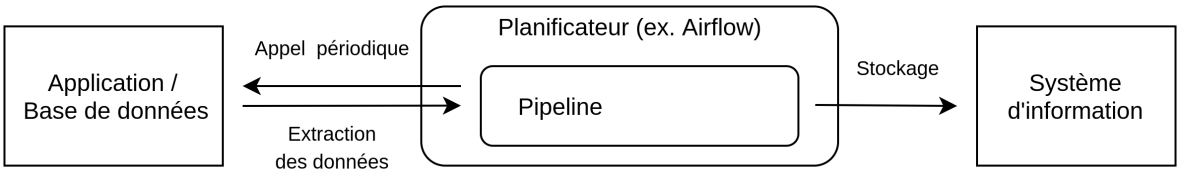


FIGURE 1.2 – Schéma du fonctionnement d'un planificateur de tâches. Les données sont récupérées grâce à des appels périodiques.

Le troisième pas est celui du traitement en temps réel. Il est parfois nécessaire que les nouvelles données se propagent dans les différents systèmes d'information à l'instant où elles sont collectées : des décisions critiques peuvent nécessiter une connaissance issue de données collectées à la dernière minute. Un autre exemple serait un cas où la collecte de données est si rapide que de traiter les données de façon périodique serait impossible dû au flot trop important. *Apache Kafka* [29] est probablement la plateforme de traitement en continu (*streaming* en anglais) la plus connue au moment d'écrire ce mémoire. La figure 1.3 présente le fonctionnement de haut niveau de ce genre de plateforme. La différence avec les traitements par lots périodiques est que le pipeline ne va pas extraire les données d'une application ou d'une base de données quelconque. C'est plutôt l'application qui pousse – ou publie – les données à la plateforme. Notons que ceci implique que l'application doit être conçue en conséquence.

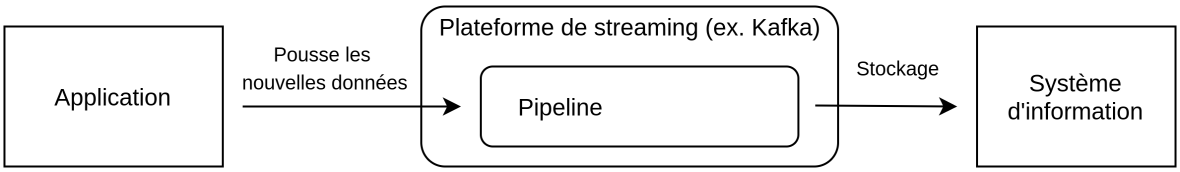


FIGURE 1.3 – Schéma du fonctionnement d'une plateforme de traitement continu (*streaming*). Les données sont directement poussées à la plateforme par l'application.

Variété

L'aspect de la variété des données réfère aux différents formats possibles. La figure 1.1 montre la gradation des défis de cette dimension. Il est, par exemple, assez intuitif de manipuler une table de données, comme un tableau *Excel* ou un fichier *CSV*. La manipulation de bases de données ajoute de la complexité à la gestion des données : une connaissance des bases de données, des schémas et de l'architecture des données devient nécessaire. Une difficulté de gestion supplémentaire survient lorsque des données ont un format spécialisé ou propriétaire. Des connaissances, outils et systèmes d'information spécialisés sont généralement nécessaires afin de manipuler et stocker ces données. C'est le cas du standard DICOM qui spécifie non seulement un format de données d'imagerie et de radio-oncologie, mais aussi comment les transférer. Comprendre et apprendre à manipuler ce standard est donc un enjeu important. À titre indicatif, l'ensemble des documents PDF contenant les spécifications du standard DICOM contenant ∼6600 pages au moment d'écrire ce mémoire [12]. Cette exhaustivité sur la spécification permet une standardisation complète et une utilisation simplifiée pour l'utilisateur qui la connaît.

Vient ensuite le dernier pas dans la gradation de la variété de la figure 1.1, qui concerne les données non structurées. Les données n'ont parfois pas de structures prédéfinies, ce qui rend particulièrement difficile leur traitement et leur analyse. On peut penser à des images, des vidéos, des fichiers PDF ou à du texte libre. Les rapports écrits à la main par des médecins, qui sont ensuite numérisés, sont un autre exemple. La récupération de ces données doit être faite par retranscription, soit par un être humain, impliquant *de facto* un risque d'erreur à la copie, ou par des algorithmes de reconnaissance optique de caractères, avec une justesse de transcription limitée. Par exemple, une étude menée par J. A. Mays et coll. a caractérisé l'erreur de retranscription de résultats de tests de glycémie dans un dossier de santé électronique par des cliniques ambulantes. 3,7% des saisies comportaient une erreur, dont 14,2% d'entre-elles étaient potentiellement dangereuses [30].

Extension aux 3 V

Depuis la publication de la définition des 3V par Doug Laney, les communautés scientifiques et techniques ont présenté d'autres extensions. En effet, pour plusieurs, les enjeux de gestion

de données ne se limitent pas par le Volume, la Vitesse et la Variété des données. On parle aujourd'hui de 4 [31], 5 [32; 33; 34], ou même 7 V [35; 36].

La *Véracité* est souvent avancée comme quatrième V par la communauté [33; 31; 37]. Elle réfère aux enjeux de confiance envers les données : leur source est-elle fiable ? Comment ont-elles été collectées ? Y a-t-il eu des biais dans la collecte ? Est-ce que les données représentent un portrait global, ou seulement une fraction du portrait ? Cette dimension renvoie essentiellement à la qualité des données.

La *Valeur* est souvent incluse dans les extensions des V. Selon Kaisler et coll. [32], Demchenko et coll. [33] et Shnain et coll. [37], la valeur des données est un aspect important : elle influence les priorités de développement, l'architecture des données et les données collectées. Cette dimension nécessite la collaboration d'experts du domaine d'application. Ce sont eux qui ont les connaissances nécessaires à l'estimation de la valeur des données. Un exemple intéressant est celui du traitement des évènements au LHC (*Large Hadron Collider*). La citation ci-dessous, tirée du site web du CERN, témoigne du fait que les données n'ont pas toutes la même valeur.

> ... parmi les quelque 600 millions d'événements que les détecteurs enregistrent par seconde, seuls 100 000 sont envoyés au Centre de calcul du CERN pour être reconstitués numériquement. Lors de la deuxième étape, des algorithmes plus sophistiqués traitent à nouveau les données, ne retenant que 100 à 200 événements intéressants par seconde [38].

Les scientifiques du CERN ne sont pas intéressés à conserver 600 millions d'événements à chaque expérience ; la très grande majorité d'entre eux sont connus et n'ont peu ou pas d'intérêt scientifique. Le CERN a donc développé et mis en place un pipeline de données qui trie et filtre ces évènements afin d'en garder que quelques centaines qui sont d'intérêt. Ici, les filtres dans le pipeline, soit les règles accordant de la valeur ou non aux évènements, sont un aspect absolument crucial au projet.

La *Visualisation* des données est une dimension présentée par Fiaz et coll. [39]. Les efforts d'ingénierie de données visent généralement à consolider des données afin d'en tirer de nouvelles connaissances. Pour que ces connaissances soient accessibles, elles doivent être communiquées. C'est dans ce contexte que Fiaz et coll. présente l'enjeu de la visualisation. Les défis de visualisation des données peuvent en effet se graduer selon plusieurs aspects, comme la rapidité de mise en disponibilité ou encore l'interactivité de la visualisation. Par exemple, des résultats pourraient être simplement présentés dans des graphiques générés manuellement ou périodiquement. La présentation des résultats pourrait aussi être en temps réel et tirer profit d'outils interactifs. C'est notamment le rôle des tableaux de bord, ou *dashboards* en anglais, tels que *Apache Superset* [40] et *Kibana* [41]. L'utilisation de tableaux de bord nécessite néanmoins un plus grand effort de développement et davantage de connaissances en ingénierie de données.

La *Variabilité* est une dimension qui concerne le changement d'une donnée ou d'un jeu de données [33; 32]. En d'autres mots, comment gérer le changement de la nature d'une donnée

collectée dans le temps. Imaginons un logiciel qui génère et stocke une donnée à chaque jour. Avec le temps, que ça soit par un changement de version du logiciel, de l'algorithme ou de la pratique, la nature de cette donnée pourrait changer. Ce changement pourrait ne pas être remarqué par les pipelines de données, et propager de mauvaises informations en aval. Ceci signifie que des mécanismes devraient être mis en place afin de s'assurer de monitorer la variabilité des données. De tels mécanismes pourraient être administratifs, c'est-à-dire que la source de données pourraient être révisée périodiquement. Un sens sémantique fort pourrait également être associé aux données grâce à des ontologies et terminologies reconnues.

La *Complexité* des données réfère au degré d'interconnexion des structures de données. Pour Kaisler et coll., la complexité mesure le niveau d'interconnexion et d'interdépendance dans les structures de données [32] où un simple changement dans la structure de données peut se répercuter à grande échelle dans le système. Les projets ayant un haut degré de complexité peuvent nécessiter d'importantes ressources humaines, autant au niveau du domaine d'expertise qu'au niveau technique. Il s'agit d'un enjeu particulièrement présent dans le domaine de la santé. Un patient va, au cours de sa vie, générer un grand nombre de données interreliées, souvent de types différents. L'ensemble de ces données correspond à son historique médical. Les historiques médicaux de différents individus peuvent même être croisés afin de générer des nouvelles connaissances. Cette pratique peut mener à des connaissances de santé publique intéressantes pour les décideurs. Elle est néanmoins compliquée à mettre en place dû à la grande complexité des systèmes en santé, tel que rapporté par Kannampallil et coll. [7].

La *Volatilité* réfère à la perte de pertinence des données en fonction de leur âge [36]. Sa conservation peut même engendrer des impacts négatifs : le coût des ressources consacrées à leur gestion peut excéder leur apport bénéfique. Dans les situations où de vieilles données doivent être préservées, mais qui ne sont que très rarement ou pas utilisées, des opérations d'archivage peuvent être envisagées. Les services d'archivage, tels que *Glacier* d'*Amazon Web Service* (AWS) [42], sont typiquement plus économiques que d'autres méthodes de stockage (ex. base de données relationnelles) puisqu'ils sont technologiquement conçus pour favoriser pour le stockage à long terme (écriture sur ruban/disque, faible impact énergétique) où peu d'opérations de récupération de données seraient effectuées.

La *Vulnérabilité* réfère aux enjeux de sécurité. Cette dimension a pris de l'importance ces dernières années, notamment à cause de cyberattaques en croissance et aux jeux de données de plus en plus complets et riches, donc plus intéressant pour des cybercriminels [43]. Les projets qui impliquent la manipulation et le stockage de données sensibles sont naturellement concernés par cette dimension de la gestion des données. En effet, une expertise technique en sécurité informatique devient importante, en plus d'une gouvernance claire et d'une gestion des accès.

1.2 Les principes FAIR

Les principes FAIR s'insèrent dans le contexte de bonne gestion de données. Ils ont été présentés dans l'article *The FAIR Guiding Principles for scientific data management and stewardship* paru en 2016 [11]. Ces principes consistent en un guide pour toute organisation ou tout groupe d'individus dans l'élaboration de systèmes de gestion de données, notamment pour des projets de recherches. En effet, la recherche étant intimement reliée à l'exploitation de données, il est primordial d'utiliser des systèmes qui permettent de bien les sauvegarder et les partager. Ceci est de plus en plus vrai depuis l'avènement des données massives et de l'apprentissage automatique, où les volumes et la complexité des données utilisées sont en croissances. Le domaine de recherche de Mark D. Wilkinson, auteur des principes FAIR, est un bon exemple. Son champ d'expertise est la génétique, où le coût de collecte des données est très élevé. En effet, le coût d'un séquençage de l'ADN d'un être humain était de plus de 1000 $ en 2019 [44]. L'opération est également reliée à une quantité importante de données : un segment du génome et les caractéristiques de l'individu pour qui l'ADN a été séquencé. Des métadonnées telles que l'appareil utilisé, la date du séquençage ou encore les outils utilisés peuvent également être conservées puisqu'elles pourraient s'avérer pertinentes. Ces enjeux ont mené à la confection du guide de gestion de données scientifiques que sont les principes FAIR. Ces principes, présentés dans [11], sont les suivants :

- ***Findable*** (Facilement trouvables)
  - F1. Un identifiant globalement unique et persistant est attribué aux (méta-)données
  - F2. Les données sont décrites avec des méta-données riches (définies au point R1)
  - F3. Les méta-données incluent clairement et explicitement l'identifiant des données qu'elles décrivent
  - F4. Les (méta-)données sont sauvegardées ou indexées dans une ressource consultable
- ***Accessible*** (Accessibles)
  - A1. Les (méta-)données sont récupérables par leur identifiant à l'aide d'un protocole de communication normalisé
    - A1.1 Le protocole est ouvert, gratuit et universellement implémentable
    - A1.2 Le protocole permet une procédure d'authentification et d'autorisation, si nécessaire
  - A2. Les méta-données sont accessibles, même lorsque les données ne sont plus disponibles
- ***Interoperable*** (Interopérables)

I1. Les (méta-)données utilisent un langage formel, accessible, partagé et largement applicable pour la représentation des connaissances

I2. Les (méta-)données utilisent des vocabulaires qui suivent les principes FAIR

I3. Les (méta-)données incluent des références qualifiées à d'autres (méta-)données

— ***Reusable*** (Réutilisables)

R1. Les (méta-)données sont richement décrites avec une pluralité d'attributs précis et pertinents

R1.1 Les (méta-)données sont publiées avec une licence d'utilisation des données claire et accessible

R1.2 Les (méta-)données sont associées avec leur provenance de façon détaillée

R1.3 Les (méta-)données respectent les normes communautaires applicables au domaine

1.3 Gestion de données et architectures

Les données dédiées aux analyses sont gérées au sein de systèmes qui permettent de les stocker, les transformer, les transférer et les exploiter. Ces systèmes tendent à être constitués des éléments typiques, tel que les entrepôts de données, les lacs de données, les magasins de données et les opérations ETL (*Extract, Transform, Load*) [45]. L'architecture de ces systèmes, c'est-à-dire la façon dont chacun des éléments énumérés plus haut est relié, est aussi souvent organisée sous des patrons typiques. Cette section développe sur les éléments typiques des architectures de gestion de données en plus des architectures.

1.3.1 Entrepôt de données

Un entrepôt de données, ou *data warehouse* en anglais, est une base de données où les données d'une organisation sont stockées dans une optique de servir à de l'analyse de données. Les données qui s'y trouvent sont structurées et suivent un schéma connu (SQL).

1.3.2 Lac de données

Un lac de données, ou *data lake* en anglais, est un répertoire ou un ensemble de systèmes d'information où les données d'une organisation sont stockées. La différence avec l'entrepôt de données est que les données sont stockées dans leur forme native, sans nécessairement de structure préétablie. Un lac de données pourrait, par exemple, prendre la forme de fichiers PDF ou CSV stockés dans un système de fichiers. Notons qu'il n'est pas requis que ces données soient prêtes à l'emploi, comme c'est le cas pour un entrepôt de données. Elles peuvent donc nécessiter un nettoyage avant l'utilisation.

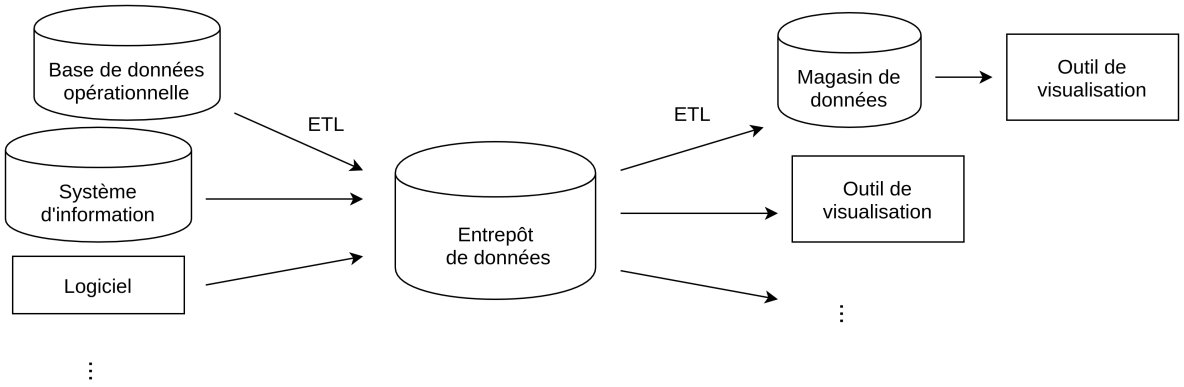


FIGURE 1.4 – Schéma typique de l'architecture autour d'un entrepôt de données.

1.3.3 Magasin de données

Un magasin de données, ou *data mart* en anglais, est une base de données spécialisée destinée à fournir des informations à un groupe donné. Il s'agit souvent d'un sous-ensemble d'un entrepôt de données ou de données initialement dans un lac de données qui ont été traitées. Les données qui se trouvant dans un magasin de données sont propres et prêtes à être analysées.

1.3.4 Opération ETL

L'opération ETL (*Extract, Transform, Load*), ou pipeline de données, est une opération qui exécute un transfert de données entre une source de données et un répertoire cible. L'opération consiste à extraire les données d'une source quelconque, les transformer via des règles prédéfinies et à charger les résultats dans un autre système d'information, comme un entrepôt de données par exemple. La transformation peut être un changement de format, des calculs, des filtres, *etc.*

1.3.5 Architectures de données massives

Les architectures typiques de projet de données massives et/ou d'analyse de données correspondent aux schémas 1.4 pour l'entrepôt de données et 1.5 pour le lac de données. Dans les deux cas, les données sont récupérées à partir de systèmes en production, soit des bases de données opérationnelles, des systèmes d'information, des logiciels qui produisent des données, *etc.* Dans le cas des lacs de données, les pipelines informatiques transfèrent les données au lac sans les transformer. Dans le cas des entrepôts de données, les pipelines informatiques peuvent transformer les données lors du transfert afin qu'elles respectent les critères de l'entrepôt.

Les enjeux amenés par ce genre de systèmes sont décrits par les V, présentés à la section 1.1. En effet, les entrepôts et les lacs de données sont typiquement volumineux (*Volume*). Ils impliquent également des pipelines qui transfèrent les données par lots manuels, par lots

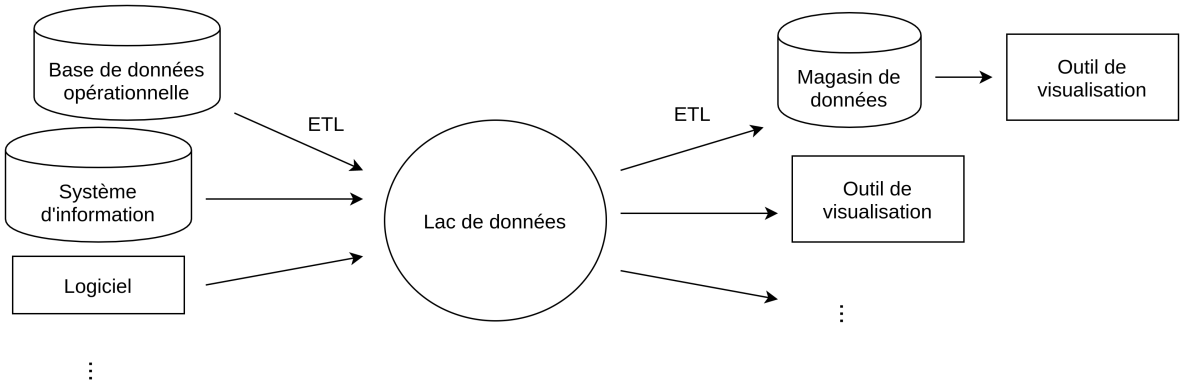


FIGURE 1.5 – Schéma typique de l'architecture autour d'un lac de données.

périodiques ou en temps-réel (*Vitesse*) et peuvent impliquer des données de différents formats (*Variété*). Les enjeux des autres V y sont également présents.

Le lac de données est particulièrement utile pour gérer des données de différents formats puisqu'il peut être constitué de répertoires de fichiers ou de différents systèmes d'information. Les pipelines qui transfèrent les données des systèmes opérationnels (à gauche à la figure 1.5) au lac de données sont généralement simples puisqu'ils n'effectuent typiquement qu'un transfert de données, sans changer le format. Ceci est en contraste avec l'entrepôt de données, où des pipelines peuvent transformer les données afin qu'ils correspondent au format préétabli de l'entrepôt de données. Les données stockées dans les entrepôts sont donc généralement plus propres dû au format imposé et à des règles de nettoyage. La quantité de travail pour construire un entrepôt est cependant plus importante.

1.3.6 Exemple au CHU de Québec

Un exemple de ce type de système est celui construit au CHU de Québec - Université Laval afin de consolider les données relatives aux traitements de curiethérapie de prostate. Les données souhaitées pour les analyses sont exportées sous le format DICOMRT (sous-ensemble de spécifications DICOM dédiées à la radiothérapie [12]) des logiciels de planification de traitement, *Treatment Planning System* (TPS) en anglais. Elles sont ensuite envoyées à une instance du serveur Orthanc [46] dédiée à la recherche et aux opérations de contrôle qualité. Le serveur Orthanc, qui est un système d'information DICOM, peut être considéré comme un des éléments du lac de données. Des fichiers PDF produits pendant la planification de traitement ont aussi été stockés dans un dossier partagé qui constitue un autre élément du lac de données.

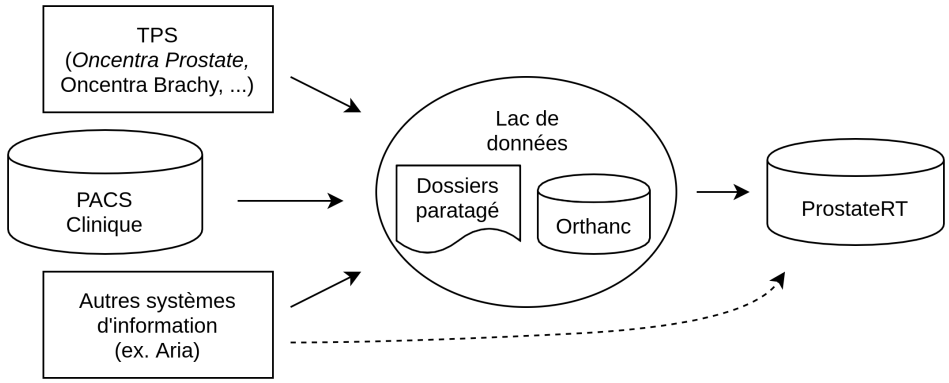


FIGURE 1.6 – Architecture de gestion des données dosimétriques des traitements de curiethérapie de prostate.

Une série de pipelines et d'opérations de saisie manuelle permet de peupler la base de données ProstateRT, un magasin de données. Notons sur la figure 1.6 que des données sont transférées de certains systèmes d'information au magasin de données directement. L'un des pipelines de cette architecture, soit celui calculant des indices dosimétriques, est décrit plus en détail au chapitre 2. Le magasin de données ProstateRT est un fichier *Access.db* placé dans un dossier partagé. Notons que cette base de données se prête mal aux utilisations complexes et à l'utilisation intensive des données [47]. Il s'agit d'un outil pertinent seulement lorsque les aspects des données à gérer se situent près de l'origine des dimensions discutées à la section 1.1. Cette solution n'est pas non plus conforme aux principes FAIR. En effet, les données ne sont pas facilement trouvables car le fichier *Access.db* est placé dans un dossier partagé (voir la section 1.2 pour plus de détails). Le choix de cet outil est justifié par des raisons historiques et légales. ProstateRT peut être utilisé par des personnes n'ayant peu ou pas de connaissances informatiques, ce qui rend sa facilité d'utilisation intéressante. Néanmoins, au moment d'écrire ce mémoire, les limitations discutées précédemment se font sentir, notamment lors de l'utilisation simultanée de plusieurs utilisateurs. Une base de données de type client-serveur permettrait d'outrepasser les limites de performance, en plus d'éviter d'avoir un fichier dans un dossier partagé.

1.4 Le standard DICOM

Le standard DICOM (*Digital Imaging and Communications in Medicine*) concerne la gestion informatique des données d'imagerie médicale [12; 48; 49; 50; 51; 52]. Le premier objectif de ce standard est de garantir une interopérabilité entre les logiciels développés par différents fournisseurs. Le standard DICOM se divise en deux aspects d'un point de vue d'ingénierie de données, soit l'aspect du format de données et celui de leur transmission.

1.4.1 Format DICOM

Le standard DICOM [49; 50; 51; 52] définit comment les données d'imagerie médicale et de radiologie doivent être structurées. Un fichier DICOM est ainsi composé de plusieurs modules, dont la présence peut être obligatoire, optionnelle ou conditionnelle, selon la modalité du fichier DICOM. La liste exhaustive se trouve sur le site Web du standard DICOM. Il est néanmoins possible de lister quelques modules d'intérêt dans le cadre de ce mémoire. Ces derniers sont présentés à la colonne de gauche du tableau 1.1. Chacun de ces modules est composé d'un ensemble d'éléments de données qui sont des conteneurs unitaires. Ces éléments sont identifiés par un *tag*. Quelques exemples sont présentés dans la colonne de droite du tableau 1.1. Les données qu'ils contiennent peuvent être des métadonnées de référencement, des paramètres d'un traitement, une image, *etc.* Un fichier DICOM est donc constitué d'un ensemble de modules, qui eux-mêmes sont constitués d'un ensemble d'éléments.

Modules DICOM	Tags DICOM
Patient	`PatientID`, `PatientName`, `PatientSex` ...
General Study	`StudyDate`, `ReferringPhysicianName`, `StudyInstanceUID`, `StudyInstanceUID`, `StudyDescription` ...
General Series, *RT Series*, *RT Struct*, *RT Plan* ...	`SeriesDate`, `SeriesDescription`, `SeriesInstanceUID`, `Modality` ...
SOP Common, *CT Image* ...	`SOPClassUID`, `SOPInstanceUI`, `PixelArray` ...

Tableau 1.1 – Modules DICOM d'intérêt.

Les éléments de données

Un élément de données est défini par un tag DICOM, c'est-à-dire un doublon de deux codes hexadécimaux, tel que présenté à la figure 1.7. Le premier code du doublon correspond au module mentionné plus haut alors que le deuxième correspond au tag en tant que tel [53]. Par exemple, le tag correspondant au nom du patient est `(0010,0010)`, alors que le tag correspondant à l'identifiant du patient est `(0010,0020)`. Ici, le premier code `0010` correspond au module *Patient* mentionné plus haut.

$$(\underbrace{0010}_{\text{Module}},\underbrace{0020}_{\text{Tag}})$$

Figure 1.7 – Structure d'un tag DICOM. Le tag montré en exemple est celui de l'identifiant de patient.

Chaque tag DICOM est lié a une représentation de donnée, *Value Representation* (VR) en anglais. Le VR correspond à un type de données prédéfini. Une liste des VR possibles se trouve sur le site Web du standard DICOM [54]. Par exemple, le tag du nom du patient, `(0010,0010)`, est associé au VR `"PN"`, pour *Person's Name*. Celui-ci impose un format pour la donnée. Tous

les tags qui réfèrent à un nom de personne sont de VR `PN`. Un autre exemple est le VR `"LO"`, ou *Long String* en anglais, qui correspond à une longue chaîne de caractères ne contenant pas les caractères d'échappement et ayant une limite de 64 caractères. Ce VR est utilisé par les tags où la donnée doit être une chaîne de caractère de format quelconque. C'est notamment le cas du tag `(0010,0020)`, celui de l'identifiant de patient, qui n'a pas un format imposé par le standard DICOM.

L'élément de donnée est donc un tag, un VR et la donnée qui correspond au format attendu. Par exemple, le tag qui contient le nom du patient prend la forme suivante : `"(0010,0010) PN NomˆPrénom"`. Notons ici que le VR `"PN"` impose que le prénom soit précédé du nom de famille. Il impose aussi la présence d'un accent circonflexe `"ˆ"` entre le nom et le prénom. Cette règle est définie dans le standard DICOM [54]. En somme, toute application qui peuple un élément de données d'un objet DICOM se doit d'assurer que la donnée soit attribuée au bon tag et dans le format définit par le VR. Ces règles rigides du standard DICOM sont ce qui permet aux fichiers DICOM d'être interopérables entre les différentes composantes logicielles qui les exploitent.

Hiérarchie DICOM

Il existe un concept de *hiérarchie DICOM* [52; 51], qui facilite la classification des objets DICOM. Celle-ci est divisée en quatre étages, *Patient, Étude, Série, Instance*, et permet de regrouper conceptuellement ce qui est fait en clinique. Une représentation de la hiérarchie est montrée à la figure 1.8.

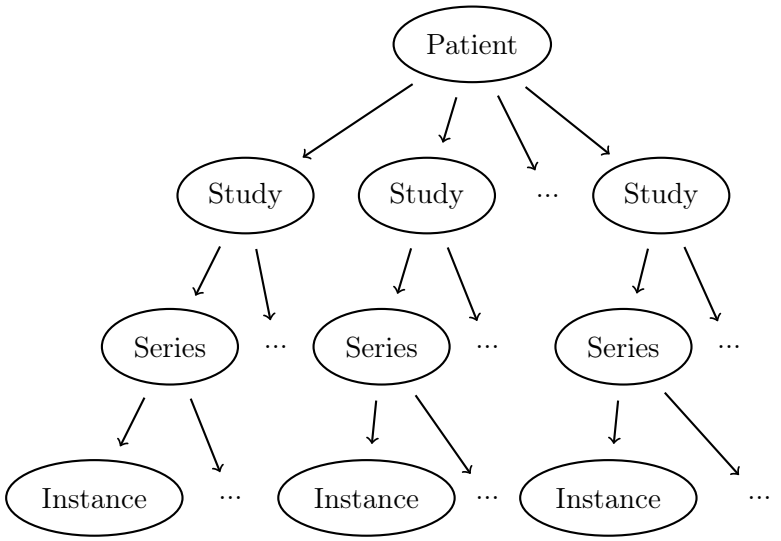


FIGURE 1.8 – Schéma de la hiérarchie DICOM, en ordre : Patient, Étude, Série et Instance

Dans un contexte hospitalier, le premier niveau, soit celui du patient, permet de regrouper toutes les études d'un patient dans un même groupe conceptuel. Ceci facilite aussi les re-

cherches du personnel médical, qui, avec les informations d'un patient, peut retrouver toutes ses études. Le deuxième niveau, celui de l'étude, regroupe des séries. Par exemple, un patient qui se rend à l'hôpital pourrait avoir plusieurs tests, tels qu'un TDM, une IRM, une échographie, *etc.* Chacun de ces tests correspond à une série dans le langage DICOM, et leur regroupement forme une étude s'ils ont tous été effectués dans la même visite à l'hôpital. L'étude permet donc au personnel médical de pouvoir retrouver les résultats de tests effectués dans une même visite. Vient ensuite le troisième niveau, celui de la série, qui correspond à une ou des instances d'une même modalité qui sont typiquement collectées lors d'une même acquisition. Par exemple, l'ensemble des images (c.-à-d. des instances) collecté dans un examen de tomodensitométrie forme une série. Autre exemple de série : un rapport structuré, qui est composé d'une seule instance et consiste à un ou des rapports faits par des médecins ou algorithmes. Finalement, le quatrième niveau, celui de l'instance, réfère à un fichier DICOM individuel. Ce fichier DICOM peut être, par exemple, une image individuelle d'un examen de tomodensitométrie, un rapport individuel, ou une planification de traitement de radiothérapie.

Chacun de ces différents niveaux est associé à un identifiant unique dans un fichier DICOM. L'identifiant du niveau Patient est le tag `(0010,0020)`, nommé *PatientID*. Cet identifiant est différent des autres, du fait que ce dernier est défini par l'institution qui génère les fichiers DICOM. Par exemple, le CHU de Québec génère des *PatientID* avec un préfixe `03HDQ` suivit d'un chiffre. Les autres identifiants, soit celui de l'étude (`(0020,000D)`, nommé *StudyInstanceUID*), celui de la série (`(0020,000E)`, nommé *SeriesInstanceUID*) et celui de l'instance (`(0008,0018)`, nommé *SOPInstanceUID*) sont tous les trois des identifiants uniques et pérennes. La structure des ces identifiants, présentée à la figure 1.9, est constitué d'un préfixe suivi d'un suffixe. Le préfixe, aussi nommé racine, est constant, et est émis pour une institution ou une compagnie. Le préfixe de la figure 1.9 est celui qui a été attribué au Groupe de recherche en physique médicale de l'Université Laval (GRPM) par la compagnie *Medical Connections*[1], qui peut émettre des préfixes libres de droit. La combinaison du préfixe à un suffixe doit être globalement unique. Notons qu'il est typique de joindre de l'information au début du suffixe. Une institution ou une compagnie pourrait, par exemple, assigner un numéro aux appareils, et faire débuter le numéro du suffixe par ce numéro, comme 28 dans l'exemple de la figure 1.9. Le contenu et le format du suffixe sont cependant arbitraires. Il est seulement nécessaire que le UID complet soit unique, composé que de chiffre et de points et qu'il ait au maximum 64 caractères.

$$\underbrace{1.2.826.0.1.3680043.10.424}_{\text{Préfixe}} . \underbrace{28.132468975015087642671881734187235 63}_{\text{Suffixe}}$$

FIGURE 1.9 – Structure d'un UID (*Unique Identifier*).

Il est important de souligner la différence entre l'identifiant de patients et ceux des études,

1. https://www.medicalconnections.co.uk/FreeUID/.

séries et instances : le premier étant un chiffre géré par une institution et est susceptible d'erreur de saisie manuelle ou d'avoir un doublon dans une autre institution, par exemple, ce qui ne le rend pas tout à fait fiable, alors que les autres identifiants sont universellement uniques. C'est pourquoi le niveau Patient dans la hiérarchie DICOM est parfois ignoré par certaines applications, qui considèrent les données relatives au patient que comme des métadonnées simples. Ceci est possible puisque le standard DICOM permet de considérer le niveau Étude comme premier niveau, et donc d'ignorer le niveau Patient.

1.4.2 Communication DICOM

Le standard DICOM couvre également l'aspect de la communication des objets DICOM précédemment mentionnées. La première version du standard a été formulée en 1985 [49], ce qui précède l'introduction du *World Wide Web* en 1989 [55]. L'aspect de la communication est divisé en services définis qui permettent de standardiser les différentes opérations de recherches et de stockage de fichier DICOM. L'énumération des services ne constitue pas l'objectif de ce travail, le lecteur intéressé peut consulter la liste de la description technique de ces opérations sur le site web du standard DICOM. Il est cependant possible d'énumérer les services pertinents à des travaux d'ingénierie de données en santé.

Le premier service d'intérêt est le C-ECHO. Ce service DICOM permet de vérifier la connexion entre deux *Application Entity* (AE). Un AE est un élément d'une application qui peut agir en tant que *Service Class User* (SCU) et/ou *Service Class Provider* (SCP)) [56]. Un SCU, essentiellement un «client» DICOM, est une application ou service qui émet des requêtes à des SCP, qui sont des services qui peuvent répondre à des requêtes. Par exemple, les système d'archivage et de transmission d'images – *Picture Archiving and Communication System* (PACS) en anglais – possèdent les fonctionnalités SCU et SCP. L'envoi et la réception d'une requête C-ECHO d'un SCU à un SCP confirme que ce dernier est accessible du SCU. Une représentation de ce service est montrée à la figure 1.10.

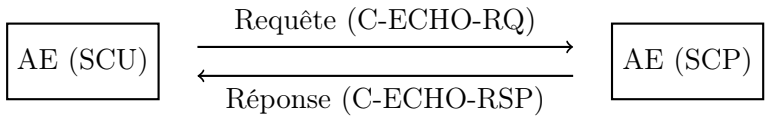


FIGURE 1.10 – Schéma d'une opération C-ECHO. La requête (C-ECHO-RQ) est répondue par la réponse (C-ECHO-RSP), qui confirme la connexion.

Un autre service communément utilisé dans diverses manipulations de données DICOM est le C-STORE. Cette opération permet de stocker un fichier DICOM dans un SCP donné (ex. un PACS). Essentiellement, un C-STORE pousse une requête contenant une ou des instances DICOM vers un SCP. Le SCP confirme par la suite au SCU (l'envoyeur) le stockage. Ce service est présenté à la figure 1.11.

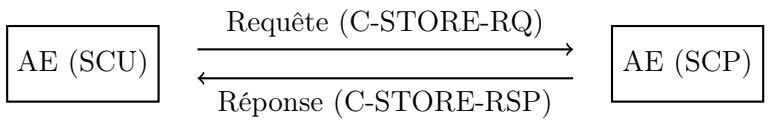


FIGURE 1.11 – Schéma d'une opération C-STORE. La demande de stockage et les données se trouvent dans la requête (C-STORE-RQ). La validation de la réussite du stockage se trouve dans la réponse (C-STORE-RSP).

Un autre service typiquement utilisé est le C-FIND. Ce dernier permet de faire des requêtes de recherche sur un SCP. Cette opération est utilisée pour trouver une correspondance avec un patient, une étude, une série ou une instance dans un SCP à partir de certains attributs présents dans la requête. Une représentation de cette opération est montrée à la figure 1.12. Il est possible de faire une recherche à chacun des niveaux de la hiérarchie DICOM mentionnée dans la section précédente, en autant que les identifiants des précédents niveaux soient dans la requête. Par exemple, une opération de recherche C-FIND au niveau Instance nécessite la présence de l'identifiant de la série (*SeriesInstanceUID*), de l'étude (*StudyInstanceUID*) et du patient (*PatientID*). Le niveau Patient est néanmoins exclu de cette logique lorsque le SCP suit le modèle du standard DICOM, où le niveau Étude est à la base de la hiérarchie [12].

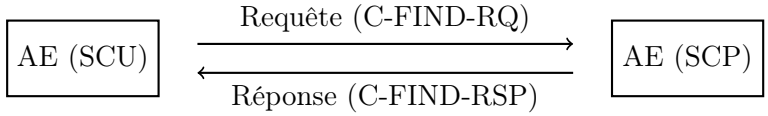


FIGURE 1.12 – Schéma d'une opération C-FIND. La requête (C-FIND-RQ) contient les informations qui permettent au SCP d'effectuer une recherche. Le résultat de la recherche se trouve dans la réponse (C-FIND-RSP).

Une autre opération DICOM pertinente aux travaux d'ingénierie de données en santé est le C-MOVE. Cette opération, montrée à la figure 1.13, vise à faire une copie des fichiers DICOM d'un AE (SCP) à un autre. Le C-MOVE est en fait une opération C-STORE encapsulée ; c'est une requête, envoyée d'un SCU vers un SCP, qui lui indique de faire une opération C-STORE vers un autre SCP. Cette opération est peu intuitive, mais s'avère utile pour faire des transferts de données complexes. Le AE cible peut être le même que le AE qui a émis le C-MOVE (le AE 3 de la figure 1.13 peut être le AE 1, ce n'est qu'une récupération de données dans ce cas-ci). Le AE cible peut aussi être un autre AE (exemple montré à la figure 1.13). Dans la deuxième situation, la machine qui envoie l'instruction de transfert de données ne reçoit jamais les données en tant que tel. Ceci est particulièrement important dans un contexte de recherche, où il n'est pas toujours souhaitable que les images transitent sur une machine personnelle. En contrepartie, l'entité qui a émis l'ordre n'a aucune information concernant l'état du transfert, si celui-ci s'est bien passé ou a échoué.

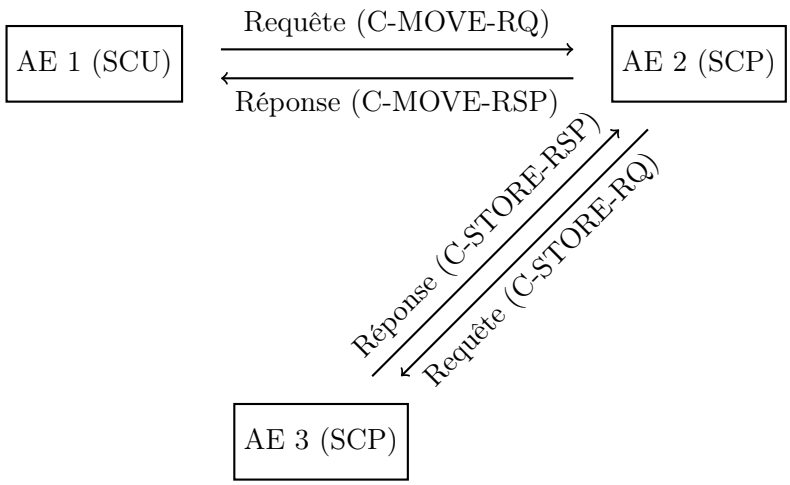


FIGURE 1.13 – Schéma d'une opération C-MOVE. La requête (C-MOVE-RQ) contient l'identité d'une ou plusieurs instances à transférer ainsi que l'identité du SCP cible, ici AE 3. La réponse (C-MOVE-RPS) valide au SCU (AE 1) que l'opération a été reçue. La réception d'un C-MOVE provoque l'opération C-STORE vers le SCP cible.

DICOM Web

DICOM Web est une extension au standard DICOM qui définit des spécifications quant à la communication d'objets DICOM via les technologies du Web (HTTP). Plusieurs définitions ont été introduites au fil des années afin de répliquer certaines opérations du standard traditionnel [57; 58], mais c'est en 2014 qu'une spécification complète a été formulée pour la première fois [58]. Celle-ci consiste en un ensemble de routes qui permettent de concevoir une interface de programmation applicative – *Application Programming Interface* (API) en anglais – de type *Representational State Transfer* (REST) afin d'accéder à des objets DICOM. REST consiste a un ensemble de contraintes pour un service web. Une API REST est donc un service web qui expose des ressources consommables par un client (c.-à-d. un utilisateur) via des appels HTTP [59]. Ces ressources sont accessibles via une route HTTP, tel que `http://mon-service-web/route-du-document-à-télécharger`.

Elles sont particulièrement intéressantes pour plusieurs raisons : elles facilitent entre autres le développement d'applications, elles exploitent les techniques de sécurité Web et elles permettent d'introduire la notion de *compte utilisateur* à un système DICOM. Ce dernier point est d'un intérêt particulier. En effet, dans la communication DICOM classique, un SCP «authentifie» la connexion d'un SCU en se basant sur l'adresse IP de ce dernier. Ainsi, c'est le système du SCU qui a la responsabilité d'identifier l'utilisateur, de limiter ses accès au besoin et de monitorer ses activités. Le SCP n'a pas cette possibilité puisqu'il n'a accès qu'à l'adresse IP (et l'AET) du SCU pour autoriser ou non la requête. Ceci est complètement différent pour un API REST qui suit le standard DICOM Web, où le serveur DICOM peut lui-même avoir sa notion de compte utilisateur afin de filtrer les requêtes.

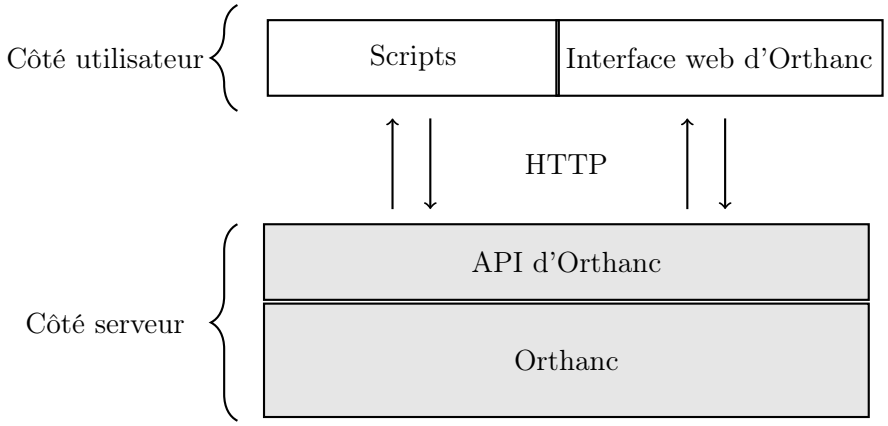


FIGURE 1.14 – Schéma des interactions possibles entre Orthanc et les utilisateurs.

L'ajout de DICOM Web au standard DICOM est donc prometteur dans un contexte où les données d'imagerie, notamment celles de recherche, pourraient être stockées dans des systèmes d'information accessibles par différents types d'utilisateurs avec des accès limités. Cependant, le standard semble peu adopté par les hôpitaux. Ceci s'explique probablement par une absence de nécessité quant à leur fonctionnement, puisque le standard DICOM traditionnel fonctionne encore très bien pour leurs activités quotidiennes.

1.5 Orthanc

1.5.1 Orthanc

Orthanc est un serveur DICOM en logiciel libre [46] issu du Département de physique médicale du Centre hospitalier universitaire de Liège. Bien qu'il existe d'autres serveurs DICOM libres (ex. *Dicoogle* [60], *EasyPACS* [61], *NeurDICOM* [62], *DCM4CHEE* [63; 64]), Orthanc se démarque par son interface Web et par une riche interface de programmation REST, ou API REST. La version 1.10.0 de l'API d'Orthanc contient en effet plus de 150 routes [2], ce qui permet à Orthanc d'être un outil de prédilection quant à la manipulation et la gestion de données DICOM, tant de façon programmatique que par l'interface Web d'Orthanc (voir figure 1.14). Enfin, un autre intérêt d'*Orthanc* est la possibilité d'ajouter ou d'implémenter des plugiciels.

1.5.2 PyOrthanc

Les possibilités offertes par la combinaison de scripts et d'Orthanc permettent d'accroître les capacités de manipulation d'objets DICOM. Dans le cadre des travaux effectués pendant cette maîtrise, une librairie Python qui facilite les manipulations de données via Orthanc a

2. Les spécifications des routes de l'API REST d'Orthanc se trouvent à l'adresse suivante : `https://api.orthanc-server.com/`, consulté le 21 mars 2022.

été développée et mise à disposition à la communauté [65]. Cette librairie, PyOrthanc, est un client Python pour toutes les routes d'Orthanc, ce qui permet de développer rapidement des scripts. Le lecteur intéressé peut consulter l'annexe A.1.

Chapitre 2

Pipeline de données dosimétriques

La technologie est un des fondements de la radio-oncologie. L'évolution constante depuis le début du XXème siècle de la pratique dans le domaine est intimement liée aux développements technologiques. Nous n'avons qu'à penser aux différentes modalités d'imagerie, telles que l'échographie, l'imagerie par rayons-X (radiographie, tomodensitométrie), la médecine nucléaire, et l'imagerie par résonance magnétique, ou encore aux appareils de traitement du cancer comme les accélérateurs linéaires. Outre les avancés techniques, un nouvel aspect promet d'avoir un impact important en radiothérapie : les données. En effet, l'exploitation et l'analyse de données massives ont le potentiel de faire progresser le domaine de la radio-oncologie – et le domaine de la santé en général. Selon Bibault et coll. [66], ces innovations en radio-oncologie prendront, entre autres, la forme de systèmes apprenants qui permettront d'optimiser la qualité des traitements. Il est cependant nécessaire de collecter et stocker adéquatement les données issues des traitements de radio-oncologie pour atteindre ces objectifs. C'est à cet effet que Lambin et coll. [8] indiquent quelles données doivent être collectées dans l'optique de concevoir des systèmes apprenants et des outils d'aide à la décision pour la radio-oncologie. Ces données sont :

— Les caractéristiques dîtes cliniques (statut de performance du patient, grade et stade de la tumeurs, résultats des tests sanguins, divers questionnaires répondus par le patient).

— Les caractéristiques du traitement (dose administrée, histogramme dose-volume, indices dosimétriques, etc.).

— Les caractéristiques tirées des images (volume de la tumeur, résultats d'analyses radiomiques, etc.).

— Les caractéristiques moléculaires issues des analyses génomiques.

C'est dans ce contexte qu'une infrastructure matérielle et logicielle est déployée au centre de recherche du CHU de Québec - Université Laval afin de consolider des données reliées aux traitements de curiethérapie à la prostate. Les données reliées à ces traitements sont rassemblées dans une base de données *MS Access*, qui est elle-même placée dans un dossier partagé. Les in-

dices dosimétriques qu'elle contient sont des données particulièrement pertinentes à rassembler dans un contexte de traitements radiatifs. La section suivante détaille pourquoi.

2.1 Pertinence des indices dosimétriques

Les indices dosimétriques sont un ensemble de valeurs quantitatives qui décrivent la qualité d'un traitement de radiothérapie. L'approbation d'un plan de traitement est d'ailleurs fortement influencée par ces données. Suite à la prescription du radio-oncologue, la planification débute par la collecte d'images de la région tumorale et de ses alentours. Ensuite, un radio-oncologue trace des contours pertinents afin de construire des structures virtuelles. Par la suite vient la planification de la dose administrée (positions et temps d'arrêts d'une source radioactive dans le cas de la curiethérapie) afin de maximiser l'irradiation de la tumeur tout en préservant autant que possible les organes sains. Une planification de dose typique de traitement de curiethérapie haut débit de dose – *High Dose Rate* (HDR) – de la prostate est montrée à la figure 2.1. Finalement, les indices dosimétriques sont obtenus à partir des histogrammes dose-volume – *Dose-Volume Histogram* (DVH) en anglais – qui sont eux-mêmes construits à partir de la dose prévue à l'étape précédente. L'approbation de la planification de traitement se base ainsi sur quelques indices dosimétriques choisis par le protocole de traitement. Par exemple, le protocole RTOG (*Radiation Therapy Oncology Group*) 0924 [68] de curiethérapie à haut débit de dose à la prostate recommande de respecter ces critères quant aux indices dosimétriques : $V_{100} > 90\%$ pour la prostate, $D_{10} < 118\%$ pour l'urètre et $V_{75} < 1$ cm^3 pour

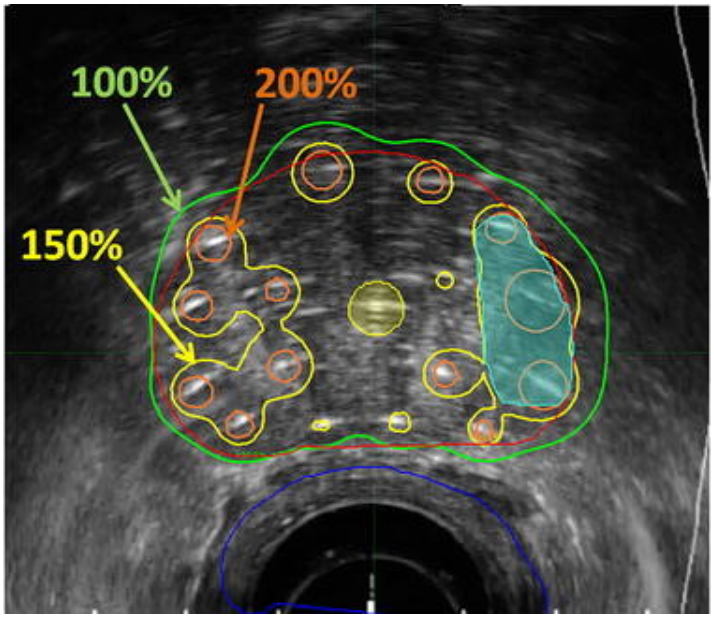


FIGURE 2.1 – Image d'une planification de curiethérapie HDR à la prostate typique provenant de [67]. Les isodoses de 100%, 150% et 200% correspondent au pourcentage de la dose de prescription.

le rectum et la vessie. Ici, les indices dosimétriques V_Y réfèrent au volume, relatif ou absolu, d'une structure (par ex. un organe) qui a reçu au moins une dose de Y (en Gy ou % de la dose de prescription), alors que les indices dosimétriques D_X réfèrent à la dose, relative ou absolue, administrée à au moins X (en cm^3 ou % du volume). Notons qu'en pratique, si les unités de Y et X n'ont pas été spécifiées, elles sont considérées comme relatives (%).

Chacune des étapes de la planification de traitement de radiothérapie produit des données DICOM. La figure 2.2 présente les fichiers créés selon de l'étape. La première étape est celle d'acquisition d'images d'une modalité donnée sous le format DICOM (ex. CT, IRM). Ensuite, les contours, donc les structures, sont stockés dans un fichier DICOM de modalité *RTStruct*. La planification de traitement est représentée dans un fichier DICOM de modalité *RTPlan*. On retrouve dans ce dernier les informations telles que la dose de prescription à la structure cible ou les paramètres de traitement selon le type de traitement (ex. curiethérapie, radiothérapie externe). Finalement, le résultat dosimétrique est stocké dans un fichier DICOM de modalité *RTDose*. Ce fichier contient essentiellement la grille de dose en trois dimensions calculée. Notons que, selon le principe de clés étrangères (*foreign key* en anglais) en bases de données relationnelles, chacun des fichiers DICOM créé réfère à ceux qui ont servi à leur génération. Ceci est possible grâce aux *StudyInstanceUID*, *SeriesInstanceUID* et aux *SOPInstanceUID*, qui permettent d'identifier les études, examens et instances (voir la section 1.4 pour avoir davantage d'information sur l'utilisation des *UID* dans le standard DICOM). Pour résumer, la grille dosimétrique (RTDose) se base sur la planification (RTPlan), qui elle-même réfère aux structures (RTStruct), qui pointe finalement sur l'examen d'imagerie. Ainsi, il est possible de retracer tout le processus menant à la construction de la dose administrée à partir du fichier RTDose. Notons que le standard DICOM possède un sous-module qui permet de stocker

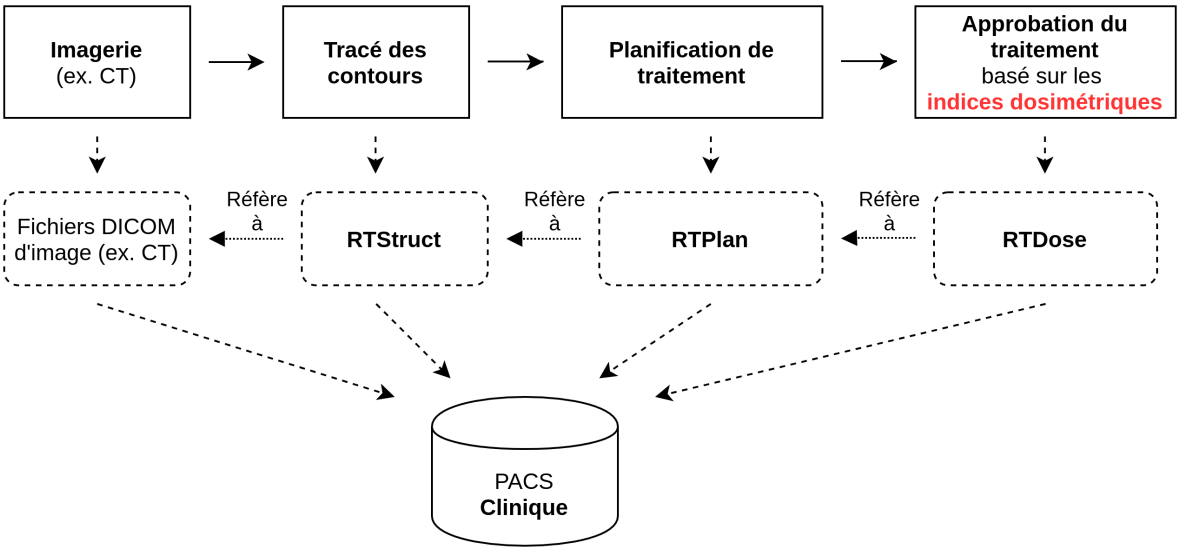


FIGURE 2.2 – Données DICOM produites le long du flot de la planification de traitement en radiothérapie.

les DVH, *RT DVH*, mais ce module n'est pas toujours utilisé par les TPS. C'est le cas du TPS *Oncentra Prostate* (OCP), qui n'utilise pas ce module. Les fichiers RTDose, RTPlan et RTStruct permettent toutefois de recalculer les DVH et, à partir de ces derniers, d'obtenir les indices dosimétriques.

Il est intéressant de stocker les indices dosimétriques puisqu'ils sont au coeur de la planification de traitement de radiothérapie. Les chercheurs peuvent les utiliser afin de mener des projets de recherche, et les cliniciens peuvent s'en servir afin d'avoir une vue d'ensemble sur la qualité des soins prodigués. En effet, comme mentionné plus haut, les indices dosimétriques sont des valeurs cruciales dans la planification d'un traitement de radiothérapie, et représentent une métrique importante en radiothérapie.

2.2 Calcul dosimétrique en curiethérapie

Le calcul dosimétrique consiste à déterminer la dose à des coordonnées données en fonction des sources [69]. En curiethérapie, les TPS se basent typiquement le Rapport du Groupe de Travail No.43 de l'Association Américaine des Physiciens en Médecine – *American Association of Physicists in Medicine* en anglais (AAPM), communément nommé TG-43, afin d'estimer la dose [70]. L'AAPM a développé le TG-43 afin d'uniformiser le calcul de dose en curiethérapie et d'établir une méthode destinée à la clinique. Celle-ci consiste à modéliser l'émission radioactive d'une source dans l'eau.

Le schéma présenté à la figure 2.3 présente le système de coordonnées utilisé dans le formalisme du TG-43.

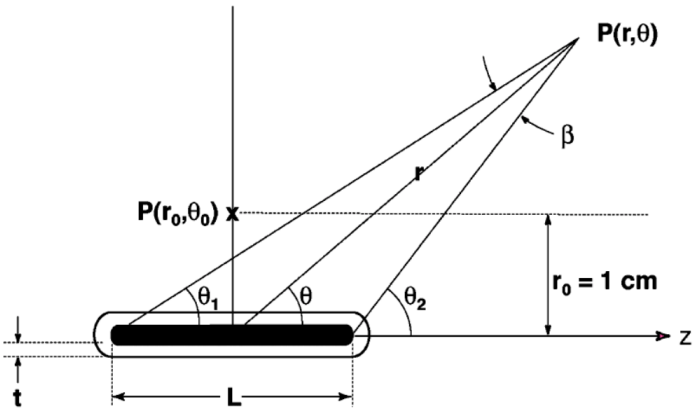


FIGURE 2.3 – Système de coordonnées utilisé par le formalisme du TG-43 [70] pour la dosimétrie en curiethérapie.

Ce système de coordonnées permet de définir le calcul du débit de dose, $\dot{D}(r, \theta)$, en un point

donné, $P(r,\theta)$, obtenu par l'équation suivante :

$$\dot{D}(r,\theta) = S_K \cdot \Lambda \cdot \frac{G_L(r,\theta)}{G_L(r_0,\theta_0)} \cdot g_L(r) \cdot F(r,\theta), \qquad [\text{cGy h}^{-1}] \tag{2.1}$$

où r est la la distance du centre de la source au point d'intérêt, θ est l'angle d'incidence entre le point d'intérêt et l'axe z, r_0 est la distance du point de référence $P(r_0,\theta_0)$ ($r_0 = 1$ cm et $\theta_0 = \pi/2$ dans le schéma 2.3) du centre de la source dans ce protocole. S_K, dite *Air Kerma Strength* en anglais, correspond au *KERMA* (*Kinetic Energy Released per unit MAss*) dans l'air multiplié par la distance d'éloignement de la source au carré, tel que montré dans l'équation

$$S_K = \dot{K}(d) \cdot d^2, \qquad [\text{U}] \tag{2.2}$$

où 1 U = 1 cGy cm^2 h^{-1}. $\dot{K}(d)$ est le taux de variation du *kerma* dans l'air et d une distance de la source. S_K est essentiellement un indicateur de la puissance d'une source. Sa valeur est, en pratique, mesurée en laboratoire. Λ est une constante du débit de dose issue de la relation entre la puissance de la source à la dose mesurée dans l'eau. Ses unités sont [cGy h^{-1} U^{-1}], donc [cm^{-2}]. La fonction géométrique $G_L(r,\theta)$ correspond à la variation relative de dose à une position donnée qui ne vient pas de la diffusion et de l'atténuation dans le milieu, mais à la décroissance de la fluence en r^{-2}. Il s'agit donc d'une correction due à la géométrie du système. Elle est donnée par

$$G_L(r,\theta) = \begin{cases} \frac{\beta}{Lr\sin\theta}, & \text{si} \quad \theta \neq 0, \\ (r^2 - L^2/4)^{-1}, & \text{si} \quad \theta = 0, \end{cases} \tag{2.3}$$

où L est la longueur de la source et β et θ les angles tels que montré dans la figure 2.3.

La fonction radiale $g_L(r)$ correspond à la diminution de la dose due à la l'atténuation et à la diffusion dans le milieu, alors que la fonction anisotropique, $F(r,\theta)$, correspond à une correction anisotropique due à la forme non-sphérique de la source. $g_L(r)$ et $F(r,\theta)$ dépendent toutes les deux du modèle de la source.

2.3 Calcul des DVH

Il devient possible de calculer les DVH suite à l'obtention des points de dose. Plusieurs étapes sont nécessaires afin d'obtenir un histogramme, et chacune d'entre elles apporte un enjeu technique. Les manufacturiers de TPS implémentent leur solution pour chacune d'entre elles, d'où des résultats potentiellement différents malgré l'utilisation des mêmes données à l'entrée. Nelms et coll. [71] les ont divisées de cette façon :

1. La transformation des contours, qui sont une série de points sur des plans 2D, en structures 3D.
2. L'interpolation entre les contours et la gestion des limites des structures (typiquement première et dernière tranche contenant une structure).

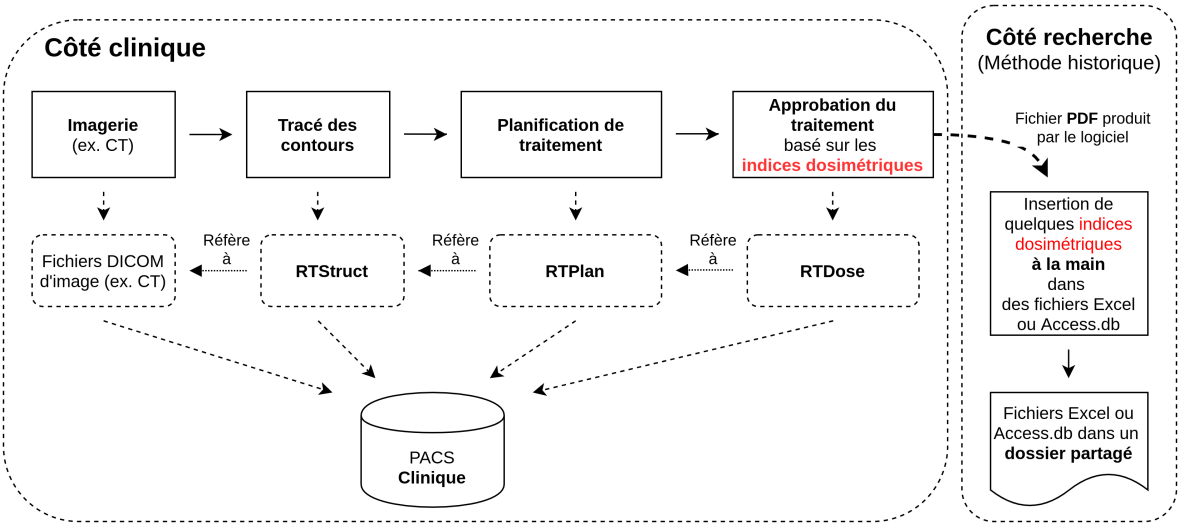


FIGURE 2.4 – Méthode historique de collecte des indices dosimétriques.

3. Résolution spatiale des points de dose. Par exemple, la grille de dose qui se trouve dans le fichier RTDose pourrait avoir une trop petite résolution (si elle est utilisé dans le calcul). Dans le cas d'utilisation de points de dose échantillonnés pour calculer les DVH, ceux-ci doivent permettre à l'algorithme de bien évaluer les régions où le gradient de dose est important ou dans les petits volumes.
4. Sur-échantillonnage de points de doses dans le régions où la résolution de la dose est insuffisante par rapport à la dimension de la structure (ex. l'urètre pour les traitement de curiethérapie de prostate).
5. Déterminer quels points de dose se trouvent dans chacune structure.
6. Déterminer la largeur des cases (*bin*) de l'histogramme.

2.4 Méthode « historique » de collecte d'indices dosimétriques

Au Service de radio-oncologie du CHU de Québec, les données issues de traitements de curiethérapie de la prostate et qui sont destinées à la recherche et à l'analyse sont consolidées dans une base de données *MS Access*, nommé *ProstateRT*, qui fait office de magasin de données. Celle-ci, mentionnée à la sous-section 1.3.6, contient notamment les indices dosimétriques des traitements approuvés. Historiquement, les données y étaient peuplées grâce à la saisie manuelle de quelques indices dosimétriques et du volume des structures, tous obtenus des TPS *Oncentra Prostate* (OCP) et *Oncentra Brachy* (OCB). La figure 2.4 présente le flot typique de collecte de données pour peupler la base de données *ProstateRT*.

Des scripts Python ont par la suite été développés afin d'extraire les indices dosimétriques des PDF produits par chaque TPS, OCP et OCB. Cette méthode, bien que fonctionnelle, est

confrontée à plusieurs problèmes : le format du fichier diffère d'un TPS à l'autre, il est probable que ce format change d'une version d'un TPS à l'autre, les indices dosimétriques présents sur le document peuvent être limités, le stockage, format et l'accès au fichier ne respectent pas les principes FAIR, les indices sont obtenus à partir de DVH pouvant être calculés par des algorithmes différents. Ces problématiques peuvent être évitées en partie ou en totalité en automatisant un recalculant les DVH avec un algorithme connu au travers un pipeline de calcul de DVH.

2.5 Développement d'un pipeline de calcul de DVH

Un pipeline de données calculant des DVH et des indices dosimétriques a été implémenté. Celui-ci prend la forme d'un graphe orienté acyclique – *Directed Acyclic Graph* (DAG) en anglais – pouvant être chargé dans la plateforme *Airflow* (voir l'annexe A.3 pour plus de détails). Le pipeline, schématisé à la figure 2.5, est constitué des trois étapes principales suivantes :

1. Interroger, analyser et récupérer des fichiers DICOM dans une instance du serveur DICOM Orthanc grâce à la bibliothèque Python *PyOrthanc* [65]
2. Calculer les DVH et les indices dosimétriques de chaque structure à partir des fichiers DICOM grâce à la bibliothèque Python *dicompyler-core* [72] (Annexe A.2 pour davantage de détails) ou grâce au microservice *Brachy-dose-calculation-microservice*, qui utilise la composante du logiciel gMCO [73] qui permet de reproduire le calcul de DVH fait par OCP et OCB.
3. Rassembler des métadonnées complémentaires, formater les données et écrire les résultats dans un fichier CSV qui sera ensuite chargé dans *ProstateRT*.

Une dernière étape manuelle, en dehors du pipeline, est nécessaire afin de peupler *ProstateRT* avec les nouveaux indices dosimétriques.

La première étape, implémentée avec *PyOrthanc* (Annexe A.1), peut être elle-même sous-divisée en cinq étapes :

1.1 Lire une liste d'identifiants de patients (PatientID), sous format de fichier texte, si fournie. Cette liste permet de traiter un sous-ensemble particulier de patients.

1.2 Construire les structures de données en arbre pour chaque patient (voir l'annexe A.1 pour avoir davantage d'information sur cette structure), où seules les études n'ayant pas déjà été traitées par le pipeline sont considérées et qui contient des fichiers RTStruct, RTPlan et RTDose.

1.3 Filtrer les arbres pour ne conserver que les cas où la planification de traitement a été faite avec les logiciels OCP ou OCB.

1.4 Pour chaque arbre, supprimer des séries RTPlan et RTStruct pour lesquelles le fichier RTDose ne réfère pas.

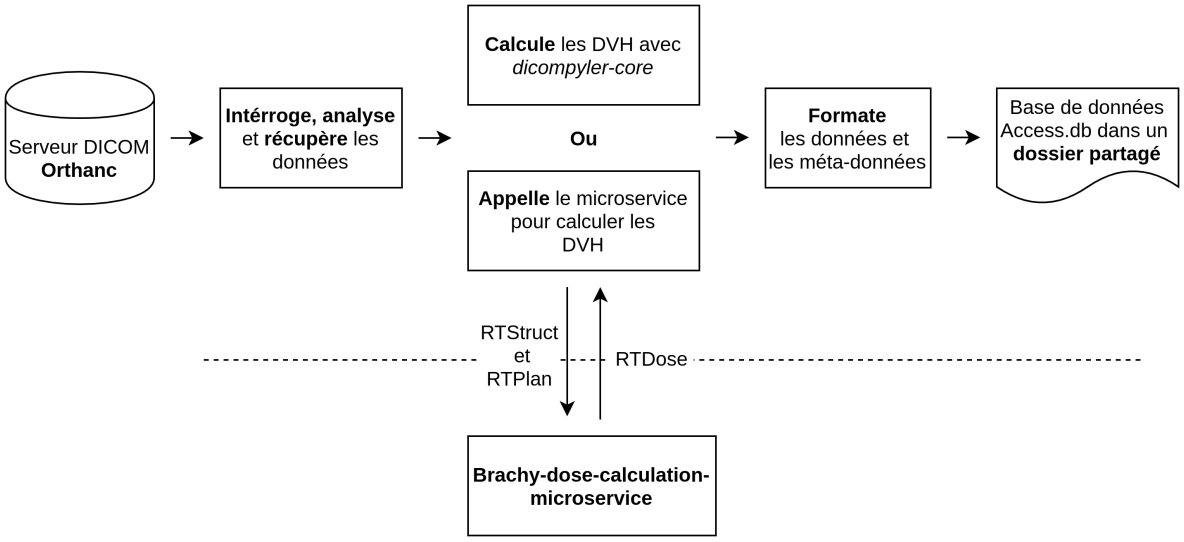


FIGURE 2.5 – Nouvelle méthode pour collecter les indices dosimétriques à des fins de recherche et d'analyse. Les DVH peuvent être calculés avec la bibliothèque Python *dicompyler-core* ou avec le microservice *Bracy-dose-calculation-microservice*.

1.5 Récupérer les fichiers DICOM qui correspondent aux instances restantes de chacun des arbres de patient et les stocker dans un système de fichiers.

La complexité de cette tâche est due au contexte d'accumulation des données dans le serveur DICOM (Orthanc) dédié à la recherche. En effet, il est commun lors de la planification de traitement de curiethérapie de verser les séries RTPlan et RTStruct sans qu'elles soient dans leur version finale, résultant ainsi en des études avec plusieurs fichiers DICOM de modalité RTStruct et RTPlan. Néanmoins, une seule série RTDose est versée par planification de traitement. Les métadonnées du fichier DICOM RTDose réfèrent aux séries RTPlan et RTStruct ayant servi à sa création, et plus précisément grâce à leur *SeriesInstanceUID*. La présence de ces métadonnées permet de sélectionner les séries désirées pour les calculs des indices dosimétriques (étape 1.4 mentionnée plus haut). Notons également que d'autres filtres sont appliqués puisque des fichiers DICOM issus de différentes sources se retrouvent dans le serveur DICOM, et non pas nécessairement celles provenant de traitements de curiethérapie. Ces filtres vérifient notamment qu'une étude ne contient qu'un seul fichier RTDose, et valident la présence des fichiers RTStruct et RTPlan qui lui correspondent. Il est aussi vérifié que ces fichiers ont été générés par les logiciels OCP ou OCB et que l'identifiant de patient (PatientID) débute par `O3HDQ` (correspondant à l'Hôtel-Dieu de Québec).

La deuxième tâche consiste à effectuer le calcul des DVH et d'extraire des indices dosimétriques à partir des fichiers DICOM précédemment récupérés (RTStruct, RTPlan et RTDose). Le calcul des DVH est effectué par la bibliothèque libre *dicompyler-core* [74] ou par gMCO [73]. La méthode utilisée par *dicompyler-core* se base sur la grille de dose présente dans le fichier RTDose qui a préalablement été exporté du TPS. *dicompyler-core* utilise donc la dose calculée

par le TPS qui a été projetée dans une grille ayant des pas uniformes. Elle superpose les voxels de cette grille sur les structures présentes dans le fichier RTStruct. Chaque voxel est associé à une dose. Si un centre d'un voxel se trouve dans le volume d'une structure, celui-ci est considéré dans le calcul. Il devient donc possible de construire le DVH d'une structure en effectuant un histogramme avec la dose correspondant aux voxels qui sont à l'intérieur de cette dernière. La section A.2 présente plus de détails à propos de *dicompyler-core*. L'avantage de *dicompyler-core* est qu'il est possible d'effectuer le calcul de DVH à partir de n'importe quels fichiers RTStruct et RTDose, que ce soit en curiethérapie ou radiothérapie externe.

Le microservice *Brachy-dose-calculation-microservice* a été développé dans le cadre de ce projet. Il s'agit d'un microservice Web muni d'une API REST qui reproduit le calcul fait par OCP et OCB. Pour ce faire, il se base sur la composante du logiciel gMCO [73] qui fait les calculs dosimétriques. Ce microservice utilise les structures (fichier RTStruct) et la planification de traitement (RTPlan) exportées du TPS OCP et produit un fichier RTDose dont le sous-module DICOM `RT DVH` est peuplé. Le microservice est présenté plus en détail à l'annexe A.4. L'avantage de cette méthode est que les résultats obtenus devraient être très proches de ceux de OCP et OCB, donc ceux obtenus en clinique. Cependant, cette méthode de calcul de DVH est limitée aux cas traités avec OCP et OCB.

Les DVH et les indices dosimétriques V200, V150, V130, V125, V120, V100cc, V100, V75, V50, V40, D100, D90, D50, D10, D5, D2cc, D1cc, D0.1cc et D0.01cc sont récupérés et stockés dans une base de données *SQLite*. Ces indices sont récupérés puisqu'ils ont été identifiés d'intérêt par les cliniciens et chercheurs au CHU de Québec - Université Laval. *SQLite* est utilisé afin d'avoir les résultats dans une base de données relationnelle qui tient en un seul fichier.

La troisième et dernière tâche se subdivise en trois étapes :

3.1 Récupérer les données dans la base de données *SQLite* mentionnée précédemment.

3.2 Filtrer les cas traitements de curiethérapie de prostate (jusqu'à ce moment, tous les cas de curiethérapie étaient traités, mais seuls ceux de prostate sont pertinents pour *ProstateRT*).

3.3 Stocker les résultats dans un fichier CSV ayant un format pouvant être chargé dans *ProstateRT*.

Au moment d'écrire ce mémoire, seuls les traitements de curiethérapie de prostate sont considérés. Néanmoins, ce pipeline de calcul de DVH est agnostique de la région cible lorsque la méthode de calcul *dicompyler-core* est utilisée. Le filtre de l'étape 3.2 pourrait donc être changé selon les traitements visés.

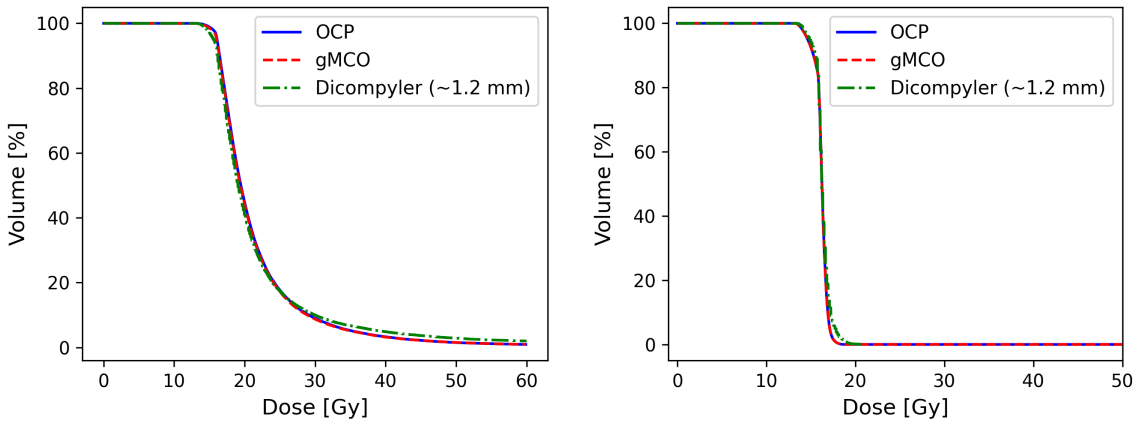
2.6 Méthodes de validation du pipeline

Les données de 20 traitements de curiethérapie de HDR de prostate (avec une dose de prescription de 15 Gy) ont été collectées afin de valider les indices dosimétriques produits par le pipeline. Les structures analysées sont la prostate, l'urètre et la vessie. Les DVH pour chaque structure ont été calculés avec le logiciel de planification de traitement OCP, avec gMCO et avec la bibliothèque *dicompyler-core* à des fins de comparaison. Les DVH ont d'abord été exportées d'OCP en fichiers textes. Les fichiers RTStruct, RTPlan et RTDose ont ensuite été exportés du TPS. Les fichiers RTDose ont été exportés avec une résolution de grille de près de 1,2 mm (parmi les grilles de dose générées par OCP, l'espacement entre les rangées est entre 1,194 mm et 1,275 mm, alors que l'espacement entre les colonnes est entre 1,147 mm et 1,234 mm, ce qui ne correspond pas nécessairement à la taille des voxels de la tomodensitométrie utilisée pour faire les calculs).

Les indices de doses (D_X) et de volumes (V_Y) sont obtenus des DVH exportés d'OCP et des DVH calculés avec gMCO et avec *dicompyler-core*. À des fins de comparaison, les valeurs de doses et de volumes sont en % (pourcentage de la dose de prescription et du volume total de la structure respectivement). Un indice D_X correspond à la dose qu'a reçu au moins X % du volume, alors qu'un indice V_Y correspond au volume ayant reçu au moins Y % de la dose (par rapport à la dose de prescription).

2.7 Comparaison entre les indices dosimétriques obtenus d'OCP et de *dicompyler-core*

Les figures 2.6a et 2.6b présentent respectivement des DVH d'une prostate et d'un urètre de cas typiques. Les courbes bleues correspondent aux DVH calculés avec OCP, les courbes rouges aux DVH calculés par gMCO et les courbes vertes par *dicompyler-core*. Les DVH d'OCP et gMCO sont visuellement similaires, alors que ceux de *dicompyler-core* présentent de légères différences, notamment dans la région de la queue (∼30 à ∼60 Gy pour 2.6a et ∼15 à ∼17.5 Gy pour 2.6b).



(a) DVH d'une prostate.

(b) DVH d'un urètre.

FIGURE 2.6 – Exemples de DVH calculés avec OCP, gMCO et avec dicompyler-core.

Les différences qualitativement visibles aux figures 2.6a et 2.6b impliquent des écarts quantitatifs entre les indices dosimétriques d'OCP, de gMCO et de *dicompyler-core*. Une distribution de ces écarts a été obtenue pour chacun des indices dosimétriques suivants :

— Prostate : D10, D20, D30, D90, V150, V100, V90

— Vessie : D10, D90, V75, V40

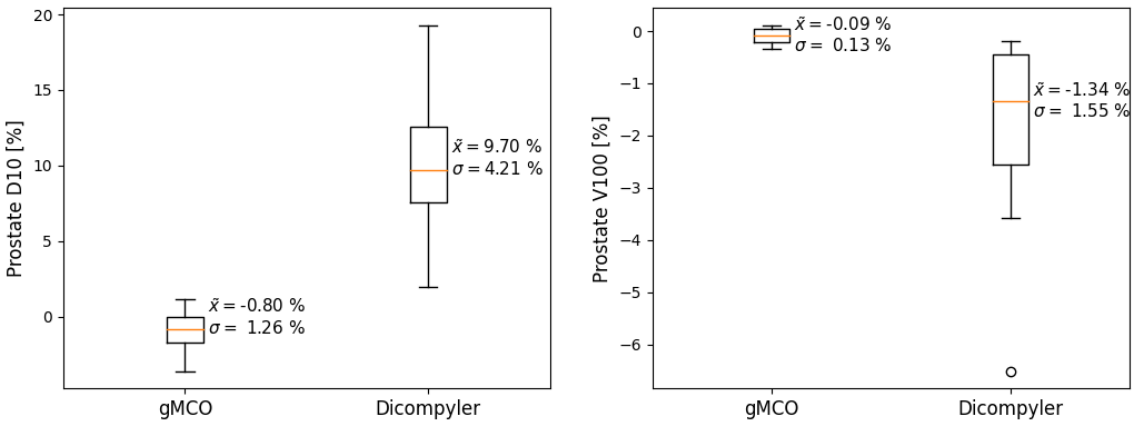
— Urètre : D10, D90, V120, V90

Le tableau 2.1 contient la moyenne des écarts absolus $|OCP_{indice\ [\%]} - gMCO_{indice\ [\%]}|$ et $|OCP_{indice\ [\%]} - Dicompyler_{indice\ [\%]}|$. Les unités des indices utilisés sont en %. Les écarts moyens des indices de doses (D_X) correspondent aux DVH observés (fig. 2.6a et 2.6b). En effet, les différences qualitativement visibles dans les queues des DVH entre OCP et *dicompyler-core* devraient se traduire en écarts plus grands pour les indices D_X où X est petit et les indices V_Y où Y est grand, ce qui est observé.

Les figures 2.7, 2.8 et 2.9 présentent des boîtes à moustaches des écarts $OCP_{indice\ [\%]} - gMCO_{indice\ [\%]}$ et $OCP_{indice\ [\%]} - Dicompyler_{indice\ [\%]}$ pour des indices pour la prostate, la vessie et l'urètre respectivement.

Structure	Indice	Écart moyen gMCO [%]	Écart moyen *dicompyler* [%]
Prostate (20 cas)	D10	1,20	9,89
	D20	0,70	2,99
	D30	0,38	1,76
	D90	0,10	1,79
	V150	0,30	1,19
	V100	0,13	1,65
	V90	0,05	0,44
Vessie (20 cas)	D10	0,24	2,47
	D90	0,02	1,48
	V75	0,09	0,82
	V40	0,20	3,80
Urètre (20 cas)	D10	0,00	1,54
	D90	0,08	1,46
	V120	0,02	1,77
	V90	0,05	0,64

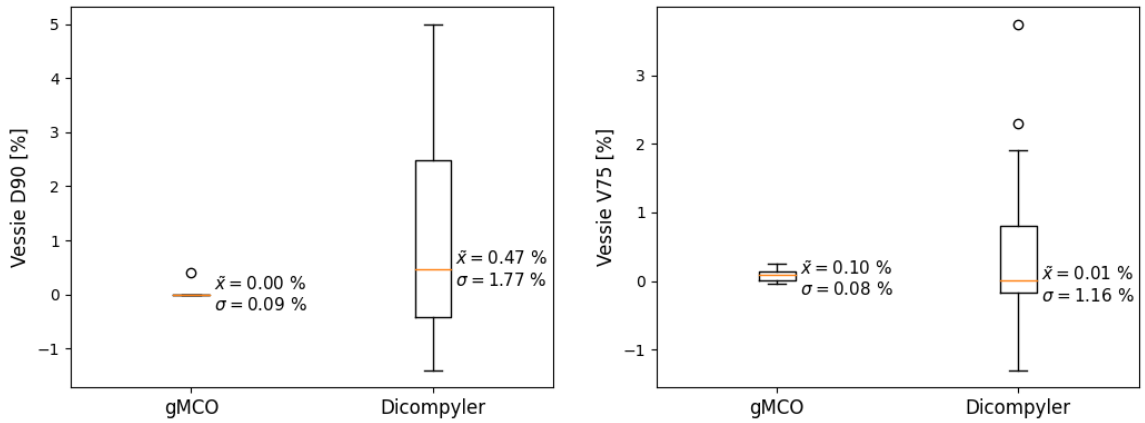
TABLEAU 2.1 – Écarts des indices dosimétriques obtenus par *dicompyler-core* et ceux obtenus du TPS OCP pour 20 cas de curiethérapie HDR.



(a) Écarts de l'indice D10.

(b) Écarts de l'indice V100.

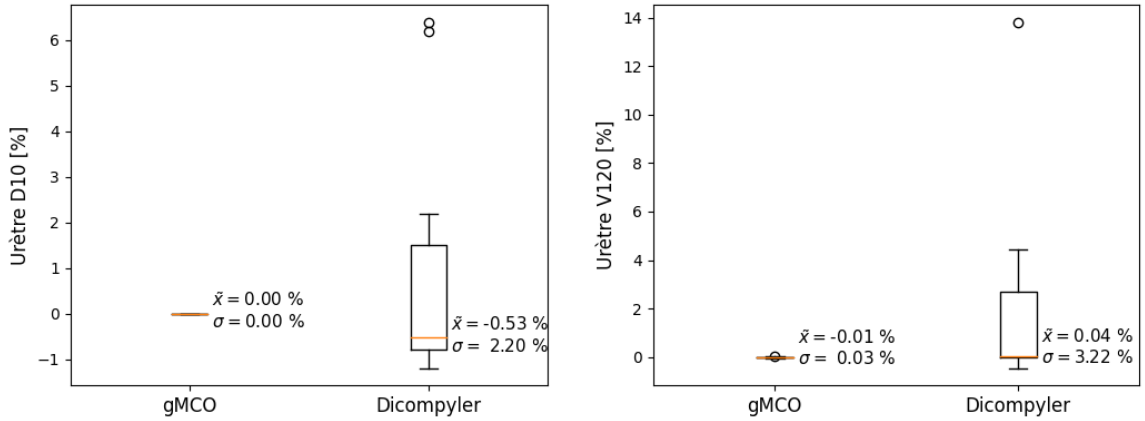
FIGURE 2.7 – Boîtes à moustaches des écarts entre $OCP_{indice\ [\%]} - gMCO_{indice\ [\%]}$ et $OCP_{indice\ [\%]} - Dicompyler_{indice\ [\%]}$ pour la prostate de 20 cas de curiethérapie HDR. Les valeurs de la médiane $\tilde{x}$ et de l'écart-type σ y sont présentées.



(a) Écarts de l'indice D90.

(b) Écarts de l'indice V75.

FIGURE 2.8 – Boîtes à moustaches des écarts $OCP_{indice\ [\%]} - gMCO_{indice\ [\%]}$ et $OCP_{indice\ [\%]} - Dicompyler_{indice\ [\%]}$ pour la vessie de 20 cas de curiethérapie HDR. Les valeurs de la médiane $\tilde{x}$ et de l'écart-type σ y sont présentées.



(a) Écarts de l'indice D10.

(b) Écarts de l'indice V120.

FIGURE 2.9 – Boîtes à moustaches des écarts entre $OCP_{indice\ [\%]} - gMCO_{indice\ [\%]}$ et $OCP_{indice\ [\%]} - Dicompyler_{indice\ [\%]}$ pour l'urètre de 20 cas de curiethérapie HDR. Les valeurs de la médiane $\tilde{x}$ et de l'écart-type σ y sont présentées.

2.8 Discussion sur les différences observées entre OCP, gMCO et *dicompyler-core*

Des différences qualitatives sont observées entre les DVH obtenus d'OCP et *dicompyler-core*, notamment dans la section de la queue, alors que les DVH issus d'OCP et gMCO sont visiblement similaires (voir les figures 2.6a et 2.6b). Ceci suggère que les algorithmes d'OCP et de *dicompyler-core* produits des résultats différents, ce qui est observé au tableau 2.1. Pour *dicompyler-core*, les indices D_X et V_Y où X est faible et Y est élevé ont typiquement des

écarts plus grands. Dans le cas de gMCO, les écarts avec OCP tendent aussi à grandir lorsque la valeur X diminue pour les indices D_X. Ces résultats indiquent que les plus grands écarts rencontrés entre OCP, gMCO et *dicompyler-core* ont lieu dans les petits volumes recevant une grande dose, en particulier pour la prostate. Notons également que les écarts entre gMCO et OCP sont typiquement d'un ordre de grandeur inférieur à ceux entre *dicompyler-core* et OCP. Puisque l'implémentation d'OCP n'est pas accessible, il est difficile de déterminer les causes de ces écarts. Il est cependant possible d'émettre des hypothèses en s'inspirant de la section 2.3, qui présente des enjeux pour l'implémentation d'un algorithme de calcul de DVH.

L'algorithme gMCO, tout en suivant le formalisme TG-43 (section 2.2), effectue un échantillonnage de points de dose aléatoire [73] dans les structures en se fiant aux données de planification du traitement de curiethérapie qui se trouvent dans le fichier RTPlan. Ceci diffère pour l'algorithme de *dicompyler-core*, qui utilise la grille de dose stockée dans le fichier RTDose. Notons qu'il n'est pas possible de déterminer si la grille de dose exportée par OCP représente bien la distribution de dose utilisée pour les calculs de DVH au sein de celui-ci. Également, il n'est pas possible d'effectuer du sur-échantillonnage de points de dose dans les régions où la résolution est insuffisante par rapport à la taille de la structure avec *dicompyler-core*. En effet, son algorithme est limité par les points de dose exportés sur une grille (à une résolution de ∼1,2mm pour les cas utilisés dans le cadre de ces travaux, soit la plus petite résolution disponible pour l'exportation de données dans OCP). Il n'est donc pas étonnant de voir de plus grands écarts dans les régions où le gradient de dose est important pour *dicompyler-core* que pour gMCO. Bien qu'il n'est pas possible de déterminer l'implémentation d'OCP, sa documentation [1] indique qu'elle utilise des techniques d'échantillonnage similaires à celles utilisées par gMCO.

Les figures 2.7, 2.8 et 2.9 montrent des écart-types beaucoup plus faibles pour les écarts entre gMCO et OCP contre ceux entre *dicompyler-core* et OCP (ex. urètre V120 : 0,03% contre 3,22%). Ces figures et le tableau 2.1 montre que gMCO produit des indices plus proches de ceux d'OCP que de *dicompyler-core*. Ceci est peu surprenant puisque l'algorithme de gMCO a été développé afin de reproduire les résultats d'OCP et d'OCB. Il est cependant spécialisé aux données de planification issues d'OCP et d'OCB, et donc à la curiethérapie. La bibliothèque *dicompyler-core* est en contrepartie plus générale. Elle utilise les points de dose présent dans le fichier RTDose. Il est donc possible d'obtenir des DVH de fichiers DICOM issus de traitements de curiethérapie et de radiothérapie externe.

1. http://www.pi-medical.gr/products/oncentra-prostate, consulté le 9 avril 2022

2.9 Conclusion sur l'automatisation du calcul des DVH et des indices dosimétriques

En résumé, un pipeline qui permet d'obtenir les indices dosimétriques a été implémenté avec le langage de programmation Python. Le pipeline, qui prend la forme d'un DAG *Airflow*, récupère quotidiennement les fichiers DICOM pertinents (RTStruct, RTPlan et RTDose), calcule les DVH et obtient les indices dosimétriques de chaque nouveau cas et exporte les résultats de façon à pouvoir les charger dans une base de données de recherche sur le cancer de la prostate. Ces indices deviennent ainsi accessibles pour les activités de recherche. Le calcul de DVH peut être fait soit par la bibliothèque de calcul de DVH *dicompyler-core* [72] ou grâce à une composante de gMCO [73] qui permet de reproduire le calcul de DVH fait dans le logiciel de planification de traitement OCP. Ces derniers sont accessibles à partir du microservice Web de calcul de DVH *Brachy-dose-calculation-microservice*, développé dans le cadre de ces travaux. La bibliothèque *dicompyler-core* peut également être utilisée directement dans du code Python du DAG.

Les indices dosimétriques de 20 cas de curiethérapie de prostate ont été obtenus avec gMCO et *dicompyler-core* et ont été comparés à ceux générés par le logiciel OCP afin de valider le calcul fait par le pipeline. Des différences ont été principalement observées pour les indices issus de *dicompyler-core*, notamment dans la région de la queue des DVH, ce qui signifie que les écarts des résultats produits sont plus grands dans les régions ayant reçu une dose élevée. Ceci implique que les indices D_X, où X est bas, et V_Y, où Y est élevé, devraient avoir des écarts plus grands, ce qui a été observé. Les écarts des indices obtenus avec gMCO sont typiquement d'un ordre de grandeur inférieur à ceux obtenus de *dicompyler-core*. Ceci s'explique par l'utilisation de gMCO de techniques d'échantillonnage adaptées à la curiethérapie (TG-43). La bibliothèque *dicompyler-core* utilise quant à elle des points de dose déjà échantillonnés et distribués sur une grille de dose de résolution finie. gMCO est donc un meilleur choix dans le contexte où l'objectif est de reproduire les indices obtenus en clinique. L'algorithme de gMCO est cependant limité aux données de planifications issues d'OCP, ce qui en fait un calculateur spécialisé. Dans un contexte où un calculateur plus général est nécessaire, *dicompyler-core* pourrait être un meilleur choix. Il pourraient également mieux performer avec des données issues de traitement de radiothérapie externe qu'avec des données issues de curiethérapie, puisque les gradients de dose sont plus faibles. Les limitations liées à la résolution de la grille de dose devraient être moins grandes.

Chapitre 3

Flots de travail de radiomique inspirés des principes FAIR

L'imagerie médicale a révolutionné l'étude qualitative des tissus et organes internes. En effet, la capacité de visualiser l'anatomie du corps humain en a fait un outil de diagnostic indispensable de la médecine moderne [75]. Plus récemment, plusieurs se sont intéressés à l'étude quantitative des images médicales. L'imagerie digitale, contrairement à l'imagerie sur film de radiologie, consiste en une matrice de valeurs numériques. Ces valeurs numériques peuvent être extraites à des fins d'analyses. Le domaine de la radiomique repose sur l'analyse de ces valeurs afin d'obtenir de nouvelles connaissances de nature biologique [9].

En pratique, la radiomique consiste à extraire des caractéristiques quantitatives d'une ou plusieurs images, d'appliquer divers traitements informatiques à ces caractéristiques, et finalement d'afficher et d'analyser les résultats quantitatifs qui pourraient mener à un diagnostic, à la prédiction du pronostic, à une association génotype-phénotype et autres.

Il est important d'appliquer de bonnes pratiques quant à la gestion des données reliées aux processus de la radiomique. Quelques ouvrages scientifiques soulèvent d'ailleurs ce point. Vallières et coll. [76] soulignent l'importance d'assurer une reproductibilité des résultats grâce à des jeux de données FAIR (section 1.2). Ils mentionnent des outils intéressants, dont notamment une ontologie reliée à la radiomique [77] et au TCIA (*The Cancer Imaging Archive*) [78], qui est un répertoire d'objets DICOM reliés au cancer. Il s'agit d'une plateforme où les chercheurs ou les cliniciens peuvent verser leurs données.

Il y a également mention de la plateforme TCIA par Limkin et coll. [79] qui souligne que les chercheurs devraient être encouragés à soumettre leurs données dans un répertoire centralisé, malgré les enjeux que cette pratique soulève (enjeux éthiques et de propriété intellectuelle). Ces défis peuvent néanmoins être surmontés. À titre d'information, la collection de radiologie de TCIA (c.-à-d. l'ensemble de jeux de données associé à la radiologie) contenait à elle seule plus

de 30,9 millions d'images en 2017 [80], ce qui témoigne du succès de la plateforme. La collection *NSCLC-Radiomics* [81] mérite également d'être mentionnée. C'est une collection dédiée aux activités de recherche en radiomique. Elle contient les objets DICOM CT, RTStruct et SEG de 422 patients.

Les travaux présentés dans ce chapitre visent à définir des bonnes pratiques quant à la gestion de données de radiomique, tant pour la production de données que pour leur stockage. Plus précisément, des flots de travail inspirés des principes FAIR qui permettent de conserver l'information coûteuse utilisée en radiomique y sont présentés. Ici, l'information coûteuse réfère autant aux calculs informatiques intensifs effectués sur les images qu'aux opérations nécessitant l'expertise et le travail de spécialistes, telle que le traçage de régions d'intérêt, qui sont dessinées sur des images médicales. Il est aussi important que ces informations coûteuses soient accompagnées de métadonnées qui décrivent comment elles ont été générées, c'est-à-dire par quelle personne ou par quel algorithme, à quel endroit, à quel moment et avec quel outil (par qui, où, quand, comment). À cet égard, l'utilisation du standard DICOM est particulièrement appropriée, puisqu'il permet aux flots de travail de capturer ces métadonnées et d'être ainsi conformes aux principes FAIR. Un flot de travail permettant de produire et conserver adéquatement des tracés de régions d'intérêt a notamment été implémenté comme preuve de concept. Celui-ci est détaillé dans la section 3.2.

3.1 Conceptualisation de flots de travail de radiomiques

Les principes FAIR, présentés à la section 1.2, sont un excellent point de départ pour la conception de flots de travail impliquant la génération et le stockage de données de recherche. Tel que mentionné précédemment, ils permettent de concevoir des jeux de données riches, bien documentés, facilement exploitables et réutilisables. Dans un contexte d'analyses radiomiques, la mise en pratique de ces principes est possible grâce au standard DICOM.

Le standard DICOM, présenté à la section 1.4, est particulièrement intéressant dans un contexte de radiomiques car il contraint l'utilisation de métadonnées riches qui permettent de retracer comment les données ont été générées (quand, où, par qui, avec quel outil ou appareil, *etc.*). Ceci fait en sorte qu'il est aligné sur des principes FAIR, et ce pour de nombreuses raisons :

1. Il est basée sur un système d'identifiants uniques et pérennes et des métadonnées riches, qui permettent de retrouver les données.
2. L'aspect de la communication du standard DICOM standardise processus d'accès aux données.
3. Les représentations des objets DICOM sont ouvertes et documentées.
4. Les métadonnées riches maximisent la réutilisation des données, ce qui les rend entre autres traçables (c'est-à-dire qu'il est possible de savoir d'où elles viennent, comment

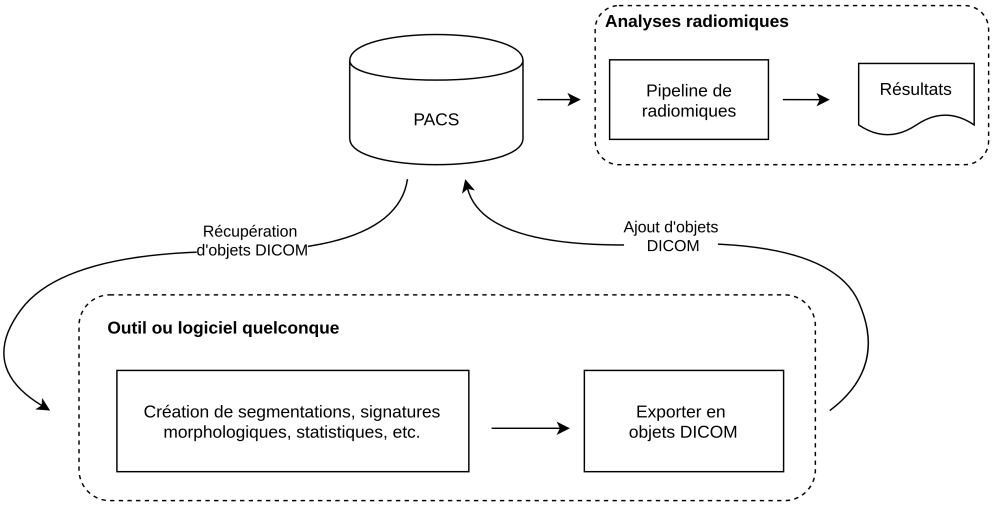


FIGURE 3.1 – Un flot de travail présentant le chemin des données. Les données sont stockées dans un PACS de recherche et sont récupérables par celui-ci. Les données générées sont placées dans un ou plusieurs objets DICOM, selon leur nature.

elles ont été générées, *etc.*).

Ceci démontre la pertinence d'exploiter le standard DICOM dans la manipulation et la confection de jeux de données d'imagerie médicale et de radio-oncologie.

Le flot de travail conçu vise à conserver non seulement les données radiomiques (par ex. segmentations faites par des spécialistes), mais aussi les métadonnées qui pourraient être importantes dans les activités de recherche ultérieures, comme les outils utilisés, le nom du spécialiste ou de l'algorithme, la date, etc. Le concept du flot de travail est illustré à la figure 3.1. Toutes les données du flot de travail (c'est-à-dire les données source et celles générées) prennent la forme d'objets DICOM, et sont stockées dans un PACS dédié aux activités de recherche. Ceci permet d'avoir un système d'information unique où les données sources et celles générées se retrouvent. Il est d'ailleurs important pour ce genre de flots de travail que les outils utilisés remplissent adéquatement les champs DICOM afin d'assurer le référencement des données.

Il est à noter que les données sources et générées ne sont pas anonymisées dans le PACS de recherche, contrairement aux objets DICOM au sein du TCIA. Plusieurs raisons expliquent ceci. Tout d'abord, dans un contexte de soumissions de projets de recherche impliquant des données médicales, il peut être nécessaire de pouvoir établir une liste de participant et de pouvoir y référer. De plus, l'anonymisation des données fait perdre certaines informations nominatives qui pourraient être pertinentes à préserver, tel que le nom de l'expert qui effectue les tracés sur les images. Il peut donc être préférable de conserver certaines métadonnées nominatives ou d'effectuer les opérations d'anonymisation lors de l'extraction de données pour les pipelines d'analyse de données. De cette façon, tous les objets DICOM sont regroupés en

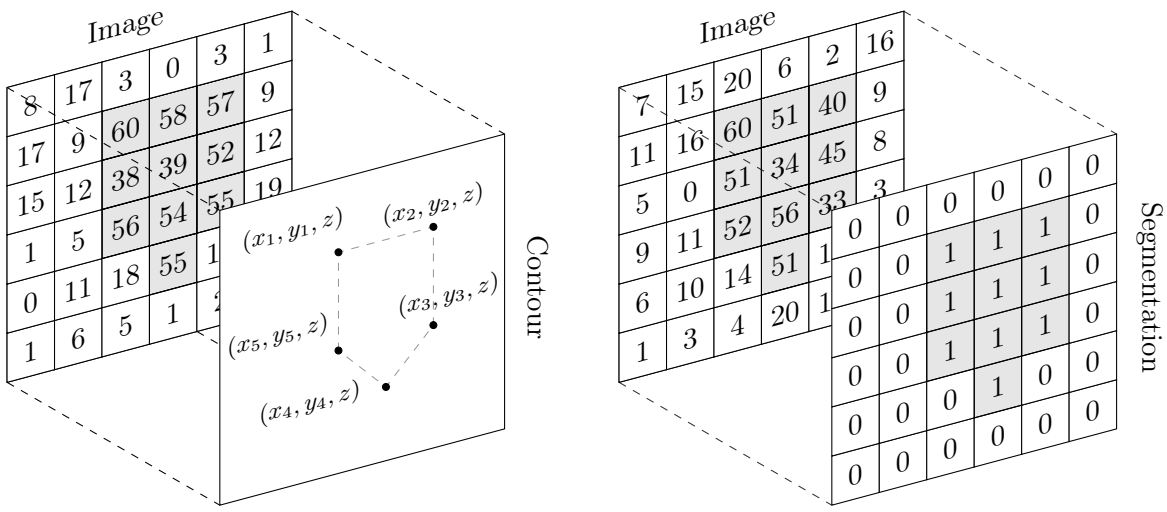
un seul endroit et forment une base de données qui peut être réutilisée par différents projets de recherche.

Les modalités DICOM (c.-à-d. les types d'objets DICOM) pertinents aux flots de travail en radiomique sont celles des images, des segmentations, des contours, et des rapports structurés. Les objets DICOM d'images consistent en les données sources dans le flot de travail conçu. Ces derniers contiennent notamment l'image brute, l'information sur le patient (c.-à-d. prénom, nom, âge, etc.), le nom de l'opérateur, les paramètres d'acquisition d'image, le nom de l'établissement, les identifiants DICOM, etc. Les identifiants DICOM sont particulièrement intéressant dans la conception d'un jeu de données de radiomique. En effet, ces dernières, sous le format RTStruct ou SEG, pointent vers les images sur lesquelles elles ont été générées avec des identifiants pérennes, tel que recommandé par les principes FAIR.

Les modalités RTStruct et SEG sont toutes les deux des conteneurs pour des traçages de régions d'intérêt. La différence entre ces deux objets se trouve dans leur façon de structurer les données. Les contours d'un objet RTStruct sont représentés par des points (positions (x, y, z)) sur les plans des images. La figure 3.2a présente un schéma d'un contour tracé sur une image, qui est elle-même représentée en pixel. Dans un objet RTStruct, ces points sont inscrits dans le tag DICOM `ContourData` sous la forme d'une chaîne de caractères tel que "x_1\$y_1$\$z_1$\$x_2$\$y_2$\$z_2$\$x_3$\$y_3$\$z_3$\...". Notons également que chaque contour (c.-à-d. chaque ensemble de points définis dans un tag `ContourData`) est accompagné des identifiants uniques des images sur lesquelles le contour a été tracé. Ces identifiants, soit les `SOPInstanceUID` des images, se trouvent dans les tags `ReferencedSOPInstanceUID`, qui se trouvent eux-même dans la séquence `ContourImageSequence`.

La région d'intérêt d'un objet DICOM SEG prend la forme d'un masque, donc d'une matrice de la même dimension que l'image et qui contient une information booléenne à savoir si un pixel donné fait parti de la région d'intérêt ou non. Le schéma de la figure 3.2b présente cette structure. Les matrices de segmentation se trouvent dans le tag DICOM `PixelData`, sous la forme d'une liste de matrices (c.-à-d. d'une liste de masques). La séquence contenue dans le tag `ReferencedInstanceSequence` contient la correspondance entre les positions de la liste et les `SOPInstanceUID` des images.

La modalité SR (rapport structuré, ou *Structured Report* en anglais) permet finalement de conserver de l'information qui ne pourrait pas l'être dans les objets DICOM mentionnés précédemment, dû à une absence de tags DICOM pour une information donnée. Sa structure de données consiste en un arbre où chaque noeud peut contenir des chaînes de caractères quelconques, ce qui la rend plus flexible. Les objets SR accueillent typiquement des rapports ou constats médicaux. Par exemple, un objet SR contenant l'historique d'un patient (date de diagnostic ou de décès, rapport de diagnostic) peut être joint à une image. Un rapport de diagnostique peut même pointer vers une image spécifique, mentionnant par exemple que



(a) Schéma d'un contour d'un objet RTStruct. (b) Schéma d'une segmentation d'un objet SEG.

FIGURE 3.2 – Schémas présentant une structure de données de contour trouvée dans un objet DICOM RTStruct (à gauche) et une autre de segmentation trouvée dans un objet DICOM SEG (à droite).

telle image a servi à l'établissement du diagnostic. Notons que les images mentionnées dans un rapport peuvent être référées grâce à leur tag `SOPInstanceUID`. Cette référence à l'image où l'observation s'est produite avec son identifiant unique et pérenne est en parfait accord avec les principes FAIR.

3.2 Implémentation d'un flot de travail

Une implémentation du flot de travail présenté dans la section 3.1 a été réalisée et testée au CHU de Québec avec des données d'imagerie dédiées à un projet de recherche en radiomique et suit le flot présenté à la figure 3.1. Cette implémentation, montrée à la figure 3.3, repose sur plusieurs outils libres, soit Orthanc [46], 3D Slicer [82], l'extension SlicerRT [83] et l'extension Quantitative Reporting [84]. Le fait que ces applications soient libres rend possible à toutes les institutions d'implémenter ce flot de travail sans engendrer de coût liés aux licences logicielles.

Orthanc, décrit à la section 1.5.1, est le PACS de recherche dans ce flot de travail. Il gère le stockage et la mise à disposition des objets DICOM. Aucun de ces objets est anonyme dans le cadre de ce flot de travail.

3D Slicer est un logiciel libre qui permet de visualiser et d'analyser des images médicales. L'application permet également de manipuler plusieurs types d'objets, tel que des contours, des annotations, des segmentations, des surfaces, *etc.* [82]. Via ses modules, Slicer peut communiquer avec un PACS par communication DICOM traditionnelle ou via une implémentation DICOMWeb pour toutes les opérations de communication (récupération, envoi, *etc.*). Ceci lui

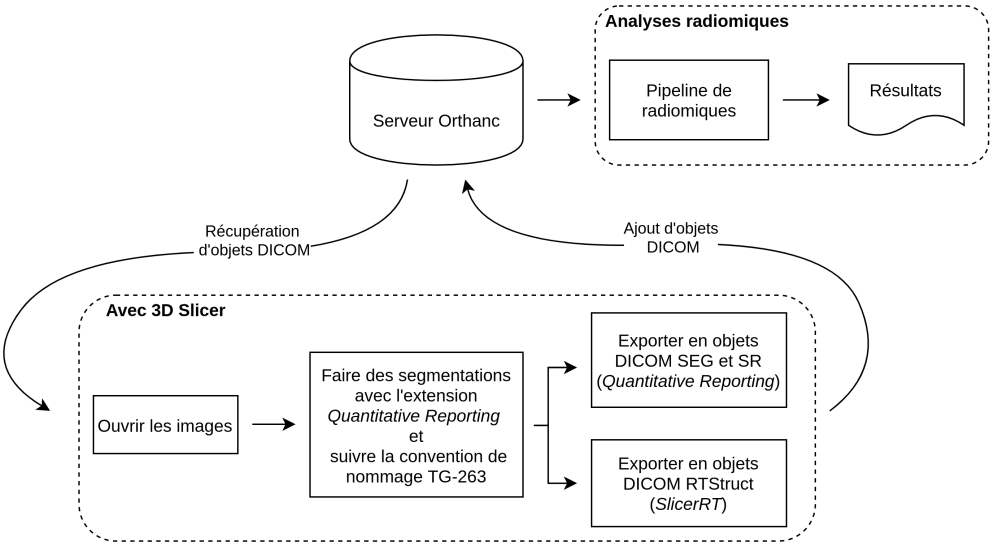


FIGURE 3.3 – Flot de travail conçu avec les logiciels libres Orthanc et 3D Slicer qui permet de tracer, créer et gérer des contours (RTStruct), des segmentations (SEG) et des rapports structurés (SR).

permet entre autres d'effectuer des requêtes sur un PACS, Orthanc dans ce cas-ci, afin de récupérer des images. Des extensions sont également nécessaires afin d'exporter les contours tracés au sein de Slicer en objets DICOM. L'extension SlicerRT [83] ajoute des fonctionnalités de manipulation d'objets DICOM reliés à la radiothérapie. Dans ce flot de travail, c'est la fonctionnalité d'exportation de régions d'intérêt sous le format d'objet RTStruct qui est pertinent.

L'extension Quantitative Reporting [84] permet quant à elle l'exportation des régions d'intérêt en objets SEG. Cette extension a été développée expressément pour la conservation d'un maximum d'informations lors du tracé de régions d'intérêt. L'exportation produit également un objet SR, où se trouvent des informations supplémentaires à celles trouvées dans un objet SEG seul. On y retrouve notamment des annotations (si ajoutées) et des mesures quantitatives reliées aux volumes [84]. Notons que lors du nommage des structures, il est recommandé de suivre une nomenclature préétablie, telle que celle fournie par l'ontologie SNOMED CT [85], qui concerne la terminologie clinique. Dans le cadre de ce travail, la nomenclature suivie pour le nommage des structures est celle donnée par le TG263 [86], qui a établi une nomenclature pour les structures en radio-oncologie spécifiquement.

3.3 Résultats

Le flot de travail implémenté permet l'accumulation d'images, de segmentation, de contours et de rapports structurés contenant de l'information d'intérêt. Ces données, toutes au format DICOM, forment une source de données exploitables par des pipelines d'analyse de radiomique.

Le flot de travail est documenté dans un manuel d'utilisation et prend la forme d'un guide. Celui-ci est présenté à l'annexe A.5 et est présent sur l'instance *Gitlab* du GRPM [1]. En suivant ce guide, un spécialiste peut reproduire le flot de données présenté, et ainsi construire un jeu de données réutilisable. Il est cependant nécessaire que l'infrastructure informatique et logicielle soit préalablement installée.

Ce flot de travail a été fait en collaboration avec Danahé LeBlanc dans le cadre de son projet de maîtrise, *Analyse radiomique du cancer de la prostate pour la prédiction du pronostic des patients avec un grand risque de récidive* [87]. Les travaux sur le flot de travail ont également conduit au flot de données présenté à la figure 3.4. Ce projet vise à concevoir un lac de données d'images et de segmentations. Il est fait en collaboration entre le groupe de recherche du Pr. Philippe Després et l'Institut universitaire de cardiologie et de pneumologie de Québec (IUCPQ). Selon des estimations préliminaires faites lors de la planification du projet, il est prévu que de 4000 à 8000 examens de tomodensitométrie soit déposés dans le lac.

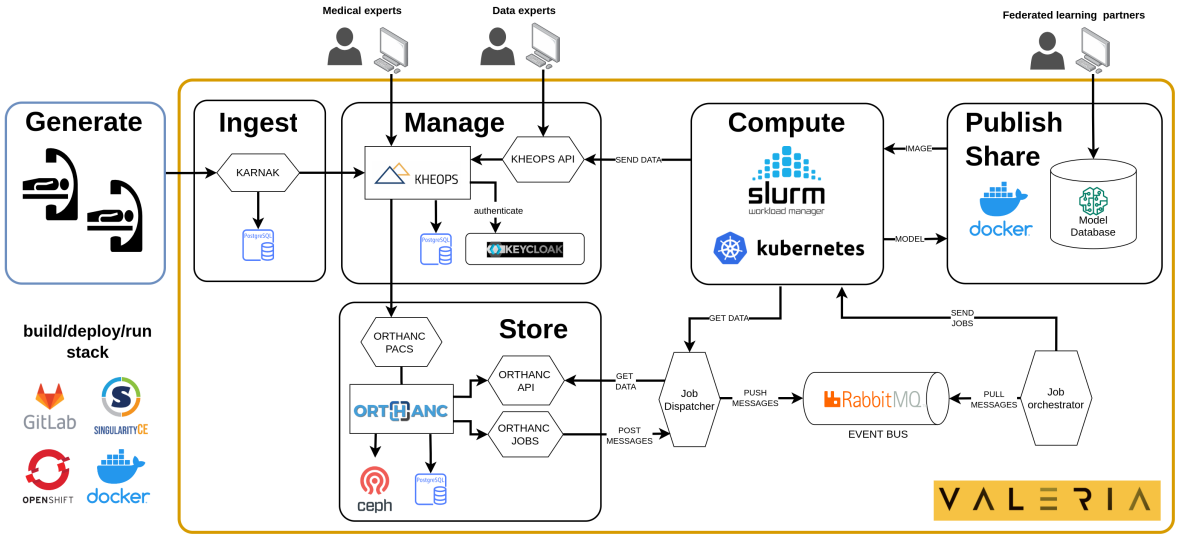


FIGURE 3.4 – Diagramme du flot de données DICOM à l'IUCPQ, comprenant notamment des images, des segmentations et des rapports structurés.

3.4 Discussion et conclusion sur le flot de travail

Le flot de travail comporte plusieurs caractéristiques suivant les principes FAIR. Tout d'abord, les données sont facilement trouvables. Cela est possible grâce au standard DICOM, sur lequel repose une panoplie d'outils tel que des systèmes d'information (le PACS Orthanc dans le cadre de ce travail). De plus, si le nommage des structures suit les nomenclatures suggérées (TG263), il devient facile de lancer des requêtes pour récupérer les données de structures précises.

1. https://gitlab.chudequebec.ca/gacou54/dicomsegmentationworklow

Les données respectent également l'accessibilité, grâce à la standardisation et l'universalité du format DICOM. Cela permet, entre autres, de les héberger sur un système d'information sécurisé où il est possible d'accorder des accès aux utilisateurs. Le standard DICOM leur permet aussi d'être lues et consommées par un large éventail d'outils, tels que `DCMTK` [88], `pydicom` [89] et `dcm4che` [64] pour ne donner que quelques exemples. En fait, toute application qui implémente le standard DICOM ou utilise une bibliothèque dédiée peut avoir accès aux données.

L'interopérabilité est aussi respectée, dû au standard DICOM et au vocabulaire utilisé dans le nommage des structures. Comme mentionné dans le dernier paragraphe à propos de l'accessibilité, la structure définie par DICOM permet à un éventail d'outil d'exploiter les données. Aussi, l'utilisation du vocabulaire contrôlé du TG263 renforce le critère d'interopérabilité du jeu de données. Notons néanmoins que le vocabulaire du TG263 semble relativement peu adopté par les différents outils en radio-oncologie, très probablement dû à son introduction récente (2018). La nomenclature semble toutefois susciter de l'intérêt et le développement de nouveaux outils dans les deux dernières années [90; 91; 92].

Finalement, ce flot de travail permet de créer un jeu de données réutilisable, ce qui en fait tout son intérêt. En effet, l'accumulation de données au format standardisé et accompagnée de riches métadonnées dans un répertoire centralisé maximise la réutilisation. Les données comprennent les objets DICOM des images, des contours (RTStruct) et segmentation (SEG), et un rapport structuré (SR). Le résultat est similaire au TCIA. D'éventuels projets de radiomique ont ainsi accès à toutes les données nécessaires pour leurs analyses.

Une différence notable entre les données issues de ce flot de travail et celles de répertoires centralisés tel que TCIA est qu'elles ne sont pas anonymes. Tel que mentionné à la section 3.1, ceci permet d'identifier les données des participants à l'étude. Ainsi, si des tracés de contours ont été effectués pour des données d'un patient donné dans une étude antérieure, ces derniers peuvent être réutilisés. C'est ce qu'on entend par conservation des informations coûteuses ; le matériel (c.-à-d. les données) utilisé par un projet de radiomique n'a pas à être recréé à chaque fois. Cependant, une meilleure pratique serait d'anonymiser le jeu de données, et potentiellement conserver cette information nominative comme le nom de l'expert ayant tracé les contours, et de garder une clé qui le lie les objets DICOM anonymes aux données nominatives.

Malgré les avantages offerts par ce flot de travail, des enjeux éthiques pourraient limiter son adoption selon les politiques et cadres réglementaires en vigueur dû à l'utilisation et à la production de données non-anonymes. Cependant, il serait simple d'ajouter un module d'anonymisation en aval, lors de l'extraction des données pour les analyses radiomiques. Dans le cadre d'un projet de recherche, il peut être nécessaire de ne travailler que sur des données dépourvues d'information identificatoire, ou encore de détruire toutes les données issues des

activités de recherche à la fin du projet. Ces pratiques font perdre le travail important que représente la création de jeux de données de qualité, et vont à l'encontre des principes FAIR. Le flot de travail développé est un moyen technique qui permet d'éviter cette perte. Il sera important de repenser la gestion des données de recherche d'établir un meilleur cadre de gestion de données quant aux obligations éthiques.

Chapitre 4

Développement de pipelines de données destinés à une étude d'évaluation personnalisée du risque de cancer du sein

La connaissance des facteurs de risque de diverses maladies pour chaque individu d'une population représente un atout important tant pour les individus que pour un système de santé. D'une part, un individu qui se sait prédisposé à une certaine maladie pourra prendre des actions en conséquence, par exemple, en changeant ses habitudes de vie ou en monitorant davantage sa santé. D'autre part, avec cette connaissance, le système de santé pourra optimiser certaines de ses pratiques. Ainsi, un examen de dépistage récurrent, coûteux ou qui présente un risque pour la santé pourrait être effectué à de plus grands intervalles pour des individus considérés à faible risque. Ceci permettrait non seulement de réduire des coûts, mais aussi des risques inutiles. À l'inverse, pour les individus à risque accru, des examens effectués à de plus courts intervalles pourraient dépister une maladie à un stade moins avancé. Ce dépistage précoce peut améliorer le pronostic de l'individu, et potentiellement éviter des traitements demandant plus de ressources au système de santé.

Au Québec, le ministère de la Santé et des Services sociaux a démarré en mai 1998 le Programme québécois de dépistage du cancer du sein (PQDCS) pour les Québécoises âgées de 50 à 69 ans [93]. Le programme invite ces dernières à passer un examen de mammographie de dépistage du cancer du sein à tous les deux ans. Cependant, puisque le risque de cancer du sein varie d'une femme à l'autre [94; 95; 96; 97], certaines pourraient bénéficier d'une fréquence d'examen de dépistage accrue. L'étude *Personalized Risk Assessment for Prevention and Early Detection of Breast Cancer : Integration and Implementation* (PERSPECTIVE I&I) [98] permettra d'évaluer une approche d'évaluation personnalisée du risque de cancer du sein. Dans

le cadre de cette étude, des données ont été collectées sur les habitudes de vie, l'historique familial de cancer et certains marqueurs génomiques de participantes âgées de 40 à 69 ans afin de calculer un risque personnalisé de cancer du sein sur les dix prochaines années. Le risque obtenu permet d'associer la participante à une catégorie de risque de cancer du sein. Selon sa catégorie, un plan d'action de dépistage sera proposé. Ce plan pourrait proposer de procéder au dépistage offert par le PQDCS, d'augmenter les fréquences des mammographies de dépistage ou encore de les débuter avant 50 ans ou d'ajouter l'imagerie par résonance magnétique (IRM) ou l'échographie.

Le calcul de ce risque est possible grâce à l'algorithme *Breast and Ovarian Analysis of Disease Incidence and Carrier Estimation Algorithm* (BOADICEA) [95]. Ce dernier, qui est développé à l'Université de Cambridge, permet de calculer les risques de cancer du sein à partir des informations collectées par PERSPECTIVE I&I. L'algorithme BOADICEA est rendu accessible via le site Web `https://canrisk.ca/`, qui expose à la fois une interface graphique et une API REST [96; 97].

Le traitement manuel du volume de données impliqué dans cette étude peut représenter un frein important aux opérations d'analyse. En effet, dans le cadre de la phase II de l'étude PERSPECTIVE I&I et en date du 4 octobre 2022, 1949 participantes ont été recrutées. Ce nombre exclu les 44 participantes ayant participé au projet pilote de cette étude. En raison de la nature de l'étude, une quantité importante et variée de paramètres (près de 800 variables) devait être collectée pour chacune des participantes, ce qui nécessite la saisie et le traitement de nombreuses informations. Ces opérations, coûteuses en temps et susceptibles d'erreurs de saisie humaine, peuvent être significativement accélérées grâce à des pipelines dédiés.

Deux pipelines de données destinés à accélérer les opérations de l'étude PERSPECTIVE I&I ont été développés. Le premier pipeline vise à estimer le risque de cancer du sein de chaque participante en interrogeant l'API REST BOADICEA avec des données récoltées dans des formulaires. Le deuxième vise à produire des lettres informatives, au format PDF, qui sont envoyées aux participantes ainsi qu'à leur médecin attitré.

4.1 Les données de l'étude PERSPECTIVE I&I

Un formulaire électronique destiné aux participantes a été développé avec la plateforme *REDCap* [99; 100] dans le cadre de l'étude PERSPECTIVE I&I. *REDCap* est une application Web qui permet de créer des formulaires et d'héberger les données récoltées. Au cours de cette étude, les données étaient recueillies en continu. Suite à la réception des données d'une participante, le risque devait être calculé et les résultats devaient ensuite être stockés. Le jeu de données n'était donc pas statique, mais en évolution. Ce contexte influence l'architecture de la solution : les données de chaque participante doivent pouvoir être traitées individuellement et en continu.

Les données récoltées par les formulaires électroniques sont stockées dans *REDCap* sous la forme de tables de données – exportables au format CSV. Celles-ci sont divisées en deux groupes, le premier étant les informations relatives à la participante à l'étude, que nous qualifions de *données personnelles*, et le deuxième les informations concernant la famille biologique de la participante, les *données familiales*.

Les données personnelles regroupent plusieurs tables au sein de *REDCap* :

— La table *Informations sur le risque de cancer du sein* contient les données des caractéristiques physiques et des habitudes de vie, telles que la grandeur, le poids, le moment de l'arrêt des menstruations, le nombre d'enfants, la consommation d'alcool, *etc.*

— La table *Histoire personnelle de cancer* contient des données relatives au cancer (sein, ovaire, pancréas ou autre).

— La table *Densité mammaire* contient des informations quant aux tissus mammaires, telles que la classification BI-RADS [101].

— La table *Score PRS* contient les données relatives au score de risque polygénique, *Polygenic Risk Score* (PRS) en anglais, qui sont pertinentes pour la prédiction du risque de cancer du sein [94].

Les données familiales partagent une structure commune pour les informations concernant les différents membres de la famille biologique de la participante. Cela comprend l'année de naissance du proche, si il ou elle a déjà eu un cancer, si oui le type, l'âge du diagnostic et l'âge au décès, s'il y a lieu. Il existe une table pour les filles, les fils, les frères, les sœurs, les neveux, les nièces, les oncles, les tantes, les parents et les grand-parents. La table qui identifie les jumeaux identiques peut être ajoutée au groupe de données familiales malgré qu'elle ait un format différent des autres.

Les données personnelles et familiales doivent être transformées et insérées en tant que valeur à la clé `pedigree_data` de l'objet JSON montré à l'annexe A.7. Une représentation est montrée à la figure 4.1. La valeur du `pedigree_data` correspond à une table accompagnée d'un en-tête. Cet en-tête contient des informations relatives à la participante tirées des données personnelles, tels que le nombre d'enfants, les habitudes de consommation d'alcool, le score PRS, *etc.* La table montrée au schéma 4.1 contient des informations tirées des données familiales (et des données personnelles pour la rangée de la participante), dont notamment l'âge, la présence de cancer et l'âge au diagnostic le cas échéant.

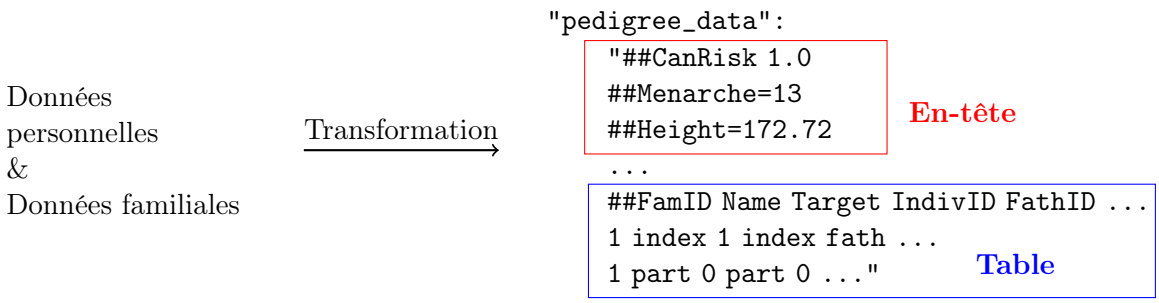


FIGURE 4.1 – Schéma représentant le transfert de format de données des formulaires à celui attendu par l'algorithme BOADICEA. L'en-tête du *pedigree* est en rouge alors que le corps – ou table – est en bleu.

4.2 Infrastructure logicielle

L'infrastructure logicielle de ce projet est déployée et gérée par PULSAR [102], une équipe qui relève du secteur Recherche et gestion de la recherche de la Direction des technologies de l'information de l'Université Laval. Dans le cadre de travaux préalables aux développements des pipelines de calcul du risque de cancer du sein et de production des lettres, cette équipe a déployé une instance de l'outil de collecte de données *REDCap* [99; 100], une instance de l'entrepôt de données *OPAL* [103], une instance de l'outil d'orchestration de pipelines informatiques *Airflow* [27] et un système de fichiers pour déposer les lettres générées. Le service Web BOADICEA est quant à lui hébergé et géré par l'Université de Cambridge. La figure 4.2 présente un schéma de cette architecture.

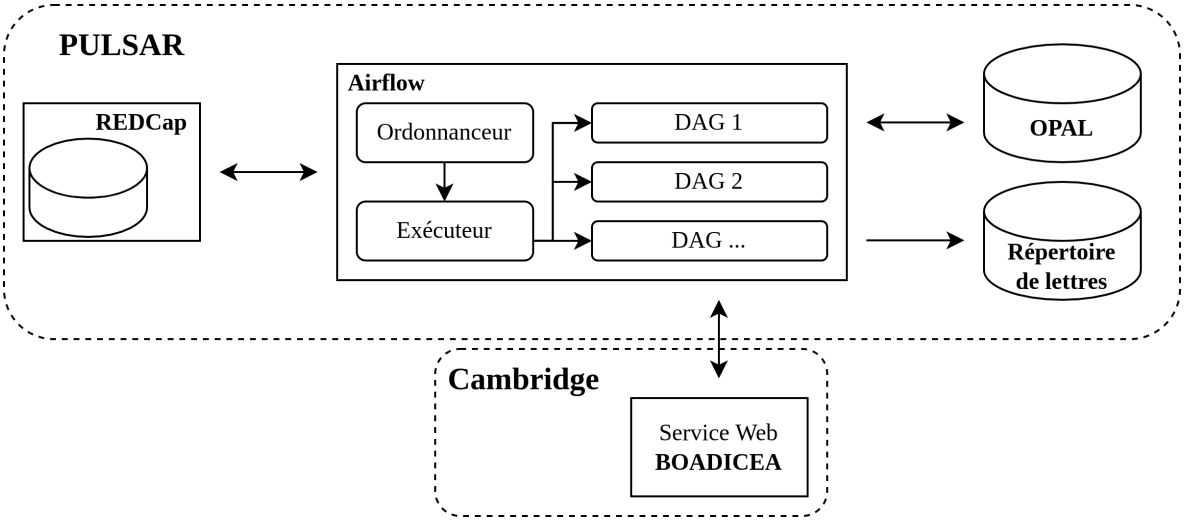


FIGURE 4.2 – Architecture des services utilisés dans le cadre du calcul du risque du cancer du sein. *REDCap*, *Airflow*, *OPAL* et le système de fichiers des lettres sont gérés par PULSAR. Le service Web BOADICEA est géré par l'Université de Cambridge.

L'instance de *REDCap* permet de collecter les données via le formulaire développé par l'équipe

PERSPECTIVE I&I. Cette instance contient donc les données brutes, dont certaines nominatives. L'instance d'*OPAL* contient quant à elle une copie des données dénominalisées de celles qui se trouvent dans *REDCap*. Deux raisons expliquent cette duplication de données :

1. Les données d'une participante sont transférées dans *OPAL* seulement lorsqu'elles sont considérées comme complètes. Un DAG vérifie que les variables nécessaires au calcul BOADICEA ont été associées à une valeur avant d'effectuer l'opération de transfert.
2. *OPAL* agit en tant qu'entrepôt de données nettoyées et dénominalisées. Ceci pourra permettre qu'elles soient réutilisables par d'autres projets de recherche.

Finalement, l'instance d'*Airflow* permet de lancer des pipelines de données – ou DAG, tels que ceux de transfert de données de *REDCap* à *OPAL*, de calcul du risque de cancer du sein et de production de lettres à envoyer aux participantes et à leur médecin traitant.

4.3 Pipelines du calcul des risques de cancer du sein et de production des lettres informatives

Le schéma présenté à la figure 4.3 montre le chemin fait par les données dans le pipeline du calcul du risque de cancer du sein. Ce pipeline quotidien, qui est exécuté par *Airflow* la nuit, débute par l'extraction des données dénominalisées de l'entrepôt de données *OPAL*. Les étapes suivantes traitent individuellement chaque participante. La deuxième étape vise à déterminer si la participante a déjà eu un résultat de risque de cancer du sein en interrogeant l'instance de *REDCap* via son API REST. L'accès à cette API est permis grâce à un jeton d'authentification. Si un résultat de risque existe, le traitement pour cette participante est interrompu. Dans le cas inverse, les données sont transformées en objet JSON consommable par l'API REST BOADICEA et envoyé à ce dernier via un appel HTTP POST. La réponse, un objet JSON contenant les résultats, est transformée selon le format attendu par *REDCap* et y est versée. Il est à noter que les objets JSON envoyés et reçus sont tous les deux conservés sous leur forme brute dans une variable dans *REDCap* afin de maintenir une trace des données exactes envoyées et reçues à l'API REST BOADICEA.

Le schéma de la figure 4.4 montre quant à lui les étapes qui permettent de produire deux lettres informatives, dont la première destinée à la participante et la deuxième à son médecin. Le pipeline de production de lettres, aussi démarré la nuit via *Airflow*, débute par l'extraction des données de *REDCap*. Il est à noter que ceci contraste avec le pipeline précédent, qui récupère ses données d'*OPAL*. Cela est dû à la nécessité d'utiliser des données nominatives afin de produire les lettres. Ces dernières contiennent entre autres le nom, la date de naissance et l'adresse civique de la participante. La deuxième étape vise à déterminer si la participante a déjà eu ses lettres validées par l'équipe PERSPECTIVE I&I. Cette validation prend la forme d'une variable booléenne à même *REDCap*. Si les lettres pour la participante ont déjà été validées, le traitement est arrêté. Si ce n'est pas le cas, les lettres destinées à la participante

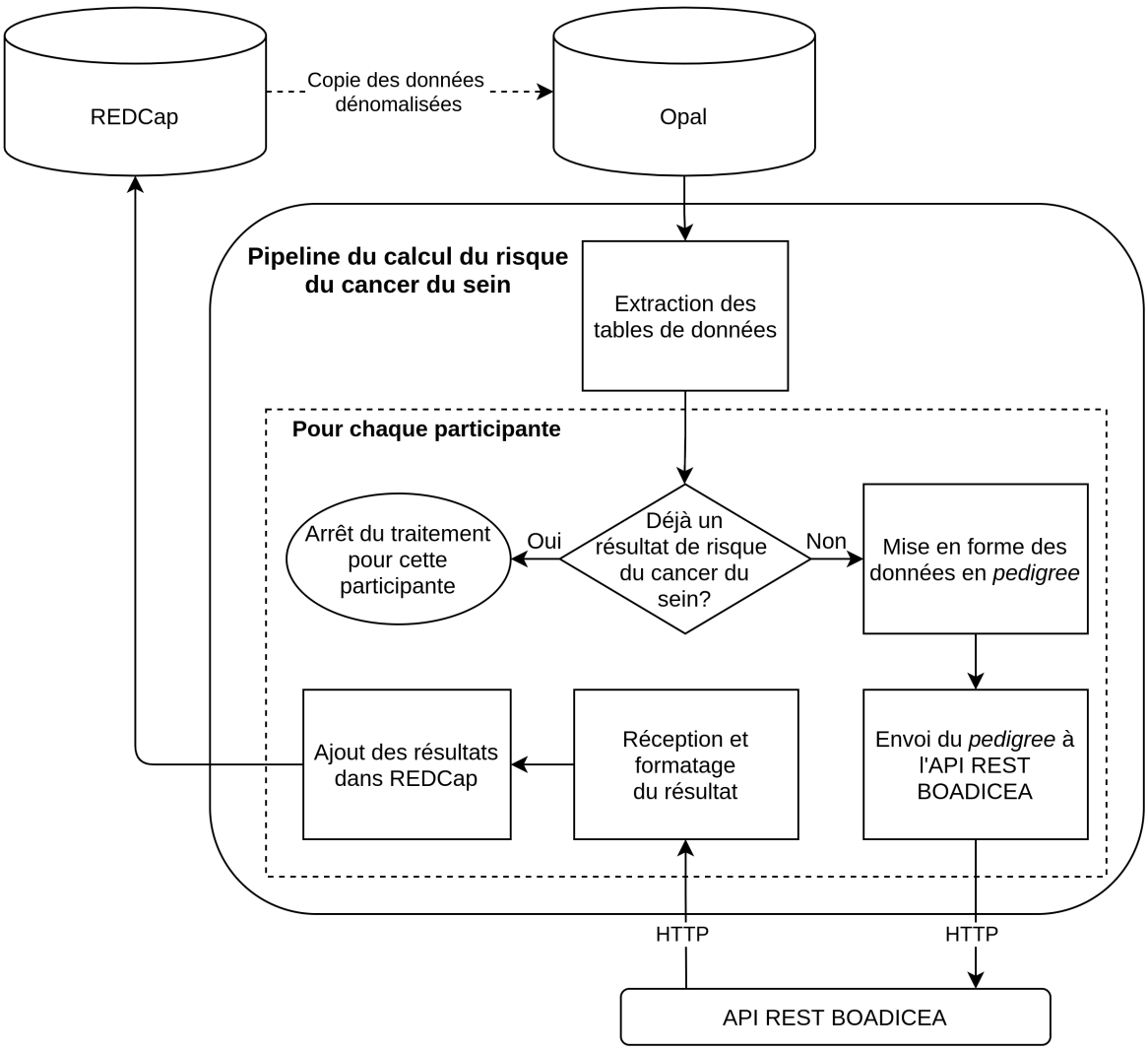


FIGURE 4.3 – Diagramme du pipeline du calcul du risque de cancer du sein.

et à son médecin sont produites. Notons que les lettres sont générées au format PDF à partir de fichiers *docx* (Microsoft Word) grâce au logiciel *Libre Office* [104].

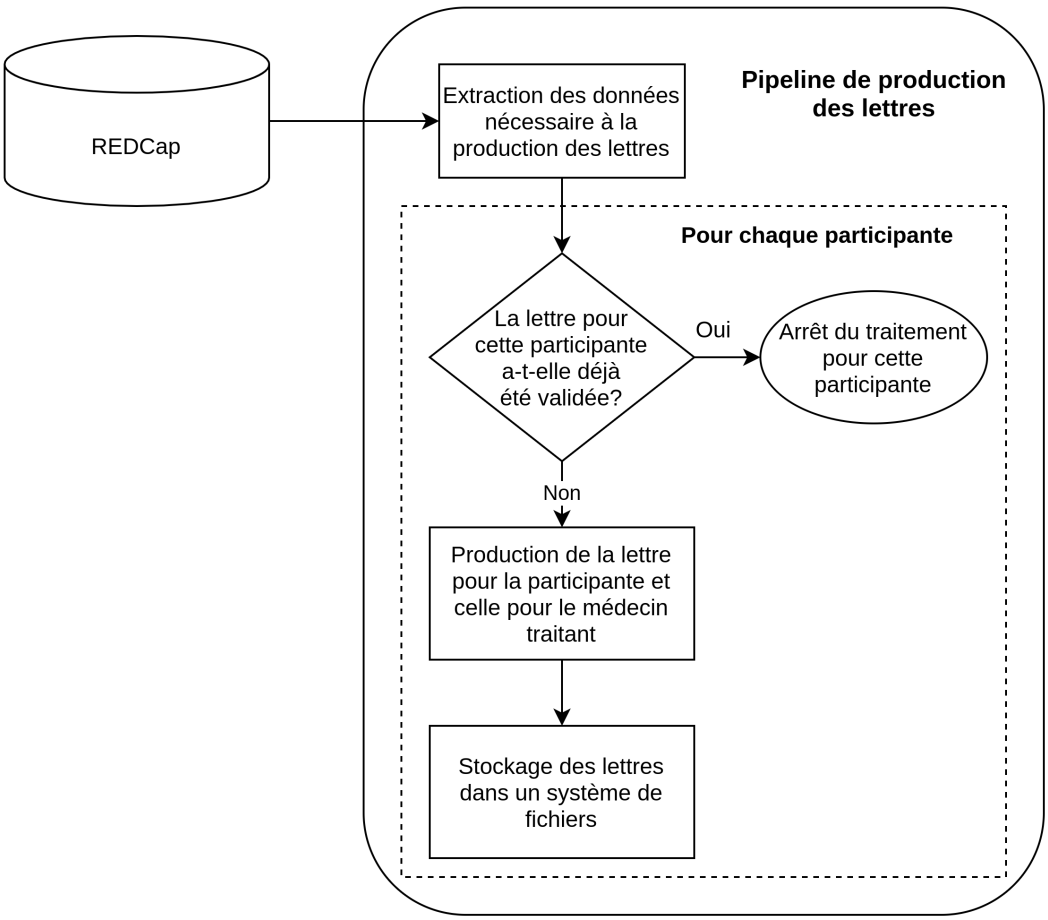


FIGURE 4.4 – Diagramme du pipeline de production des lettres.

Les lettres sont ensuite stockées dans un système de fichiers sécurisé supervisé par PULSAR. L'équipe PERSPECTIVE I&I peut finalement valider les lettres produites. Cette validation consiste à vérifier si les lettres finales d'une participante donnée ont le format et le contenu attendus, c'est-à-dire que l'insertion du contenu dans le gabarit de la lettre n'a pas déplacé les éléments visuels et que le contenu, tels que les noms ou dates, n'ont pas d'erreur. Si les lettres sont valides, l'équipe PERSPECTIVE I&I l'indique dans la variable booléenne précédemment mentionnée.

Les pipelines du calcul du risque de cancer du sein et de production des lettres ont été implémentés avec le langage de programmation Python, un choix motivé principalement par deux raisons. Tout d'abord, il s'agit du langage de programmation par défaut de la plateforme *Airflow*. De plus, il possède un riche écosystème de bibliothèques qui a permis de faciliter l'implémentation de chacune des étapes des pipelines, notamment :

1. *pandas* [105], qui permet de manipuler les données relationnelles,
2. *requests* [106], qui permet d'interagir avec l'API REST BOADICEA et l'API REST d'*OPAL*,
3. *PyCap* [107], qui permet d'interagir facilement avec l'API REST de *REDCap*,
4. *docx-mailmerge* [108], qui permet de peupler les champs d'un gabarit de lettre au format *docx* avec les informations des participantes.

4.4 Résultats

En date du 4 octobre 2022, sur les 1949 participantes recrutées au courant phase II de l'étude, 89 se sont retirées sans conservation des données, 64 se sont retirées avec conservation des données et 164 ont abandonnées (c.-à-d. n'ont pas terminé le processus de collecte de données). Une catégorie de risque de cancer du sein a été attribuée à 1626 participantes (6 n'ont pas encore eux leur risque calculé). Les catégories sont déterminées selon le risque, en pour mille (‰), d'avoir un cancer du sein dans les dix prochaines années. Les seuils de risque qui déterminent la catégorie sont présentés à l'annexe A.6.

Les proportions des participantes parmi les catégories sont présentées au tableau 4.1.

Catégorie de risque	Nombre de participantes [-]	Proportion de participantes [%]
Près de celui de la population générale	1211	74,45
Intermédiaire	293	18,02
Élevé	122	7,50

Tableau 4.1 – Participantes par catégories de risque de cancer du sein.

4.5 Discussion et conclusion sur les méthodes utilisées pour le calcul du risque de cancer du sein

Les pipelines développés permettent d'automatiser une grande partie des processus, soit le calcul des risques de cancer du sein et la production de lettres destinées aux participantes et aux médecins qui font leur suivi. Certaines opérations manuelles sont toutefois nécessaires afin d'exercer un contrôle qualité, notamment pour valider les données qui entrent dans le pipeline de calcul du risque et valider les lettres résultantes.

Les choix d'outils se sont avérés être en majorité pertinents. Les données qui se trouvent dans l'instance de *REDCap* sont facilement accessibles et peuvent aisément être mises à jour grâce au client Python *PyCap* dédié. L'instance d'*Airflow* permet de planifier le lancement des pipelines à chaque nuit. Le recours à *OPAL* a par contre été contre-productif dans le cadre de cette étude. En effet, en théorie, l'avantage d'extraire les données directement d'*OPAL* est que les informations d'une participante donnée ont été validées et sont prêtes pour le calcul de risque. Néanmoins, le travail d'accès aux données dans *OPAL* a de beaucoup surpassé les efforts qui auraient été nécessaires pour utiliser les données directement de *REDCap*, opération pour laquelle le code existe déjà dans le DAG de transfert entre *REDCap* et *OPAL*. La logique aurait donc pu simplement être réutilisée et n'aurait donc pas nécessité un travail supplémentaire pour accéder aux données dans *OPAL*. Cette dernière tâche fut particulièrement ardue. Le logiciel en ligne de commande (CLI) pour accéder à *OPAL*, *opal*[1] comportait beaucoup de bogues, ce qui rendait son utilisation ardue et imprévisible. La solution alternative a été d'appeler directement l'API REST d'*OPAL* via des appels HTTP. Malheureusement, cette API n'étant pas documentée, il fut nécessaire d'inspecter le code. Un client Python serait un outil fort utile à des développements subséquents afin de faciliter l'accès à *OPAL*. Notons également que les deux pipelines accèdent tout de même à *REDCap* puisque les résultats du risque de cancer y sont chargés et que le pipeline des lettres utilise des données nominatives. N'utiliser que *REDCap* comme source de données n'aurait donc pas nécessité d'efforts supplémentaires (il est à noter que le requis OPAL a été imposé par l'équipe PULSAR). Il est tout de même pertinent de conserver les données dans OPAL puisqu'il s'agit d'une plateforme intéressante pour la réutilisation de données par d'autres projets, un des objectifs d'un entrepôt de données. OPAL permet en effet de gérer de façon granulaire les accès et permissions liés aux jeux de données, jusqu'au niveau de la variable.

Le langage de programmation Python s'est quant à lui avéré à être un bon choix. En effet, les librairies Python énumérées dans la section 4.3 ont toutes été utilisées dans la conception des deux pipelines dans presque chaque étape de ces derniers. Le fait que le langage Python soit nativement lié à l'outil *Airflow* a également permis de réduire le temps de développement.

Les DAG de calcul du risque de cancer du sein et de production des lettres permettent de capturer et d'identifier facilement les erreurs. En effet, lors d'une erreur un courriel est envoyé à l'équipe PULSAR, qui peut par la suite notifier l'équipe PERSPECTIVE I&I. Une erreur produite par le calcul BOADICEA écrit également le message à même une variable dédiée dans les données de la participante dans *REDCap*. Ceci permet à l'équipe PERSPECTIVE I&I de facilement comprendre la cause de l'erreur et apporter la correction nécessaire à la donnée en cause. Le DAG du calcul du risque recalculera le risque la journée suivante. Le design du pipeline de production des lettres s'est aussi avéré pertinent pour la gestion d'erreurs et

1. Documentation du logiciel en ligne de commande *opal*, https://opaldoc.obiba.org/en/latest/python-user-guide/commands.html

de mauvais format de lettre (par exemple, un nom de participante ou de médecin trop long pourrait changer la disposition des éléments de la lettre). De par sa conception, le pipeline reproduit les lettres de toutes les participantes à chaque nuit, tant que les lettres pour une participante précise n'ont pas été validées, tel qu'indiqué dans le schéma de la figure 4.4. Ainsi, si une erreur quelconque s'introduit, tel qu'un mauvais format ou d'une faute de frappe dans le nom du médecin, l'équipe PERSPECTIVE I&I peut corriger l'erreur et attendre à la journée suivante pour avoir les lettres corrigées. Une fois que les lettres d'une participante donnée ont le format et le contenu attendu, l'équipe PERSPECTIVE I&I l'indique dans une variable au sein de *REDCap*, ce qui les empêche d'être à nouveau créées. Ensuite, une fois les lettres validées, les lettres sont transférées dans un dossier dédié. Un pipeline développé par PULSAR vient finalement récupérer les lettres destinées aux participantes de ce dossier afin de les charger dans leur plateforme de santé durable. Les participantes peuvent ainsi les consulter via un tableau de bord personnalisé. Les lettres dédiées aux médecins sont quant à elles faxées par l'équipe PERSPECTIVE I&I.

Finalement, il important de noter que le format atypique de l'objet JSON à envoyer au service Web BOADICEA a significativement augmenté la complexité du projet. En effet, ce dernier, montré à l'annexe A.7, possède un format spécifique au projet. Aucun outil basé sur un standard ne permet de lire la forme attendue par la valeur associée à la clé `pedigree_data`, ce qui a nécessité le développement d'un lecteur et d'un créateur (respectivement *parser* et *builder* en anglais). Cette contrainte aurait pu facilement être évitée par une utilisation adéquate du format JSON, ou par la simple utilisation du format CSV.

Conclusion

Les travaux effectués dans le cadre de ce mémoire visaient à identifier et implémenter de bonnes pratiques quant à la gestion de données en santé, et plus précisément en radio-oncologie. L'établissement de bonnes pratiques est d'actualité en santé, un domaine où les défis reliés aux données sont nombreux et ayant trait par exemple à la gestion de systèmes préexistant, à une hausse importante de la volumétrie, à la variété des données ou encore à des enjeux éthiques. Malgré ces défis, les données en santé nous permettent d'envisager de remarquables avancées, de l'automatisation de la collecte et de l'analyses de données, à l'assemblage de jeux de données prêts à l'emploi pour des activités scientifiques, ou encore à la médecine personnalisée.

Les bonnes pratiques d'ingénierie de données ont un caractère très général jusqu'à très spécifique selon le domaine d'application. Le chapitre 1 aborde ces sujets, en premier lieu avec l'identification des enjeux reliés à la gestion des données massives (les V) : le volume à gérer, la vitesse d'ingestion et du traitement de données nécessaire, la nature des données à manipuler, *etc.* De plus, les principes FAIR [11] sont désormais incontournables quand il est question de gestion de données dans un contexte de recherche scientifique. Ces derniers visent à rendre les données dédiées aux activités de recherche **F**acilement trouvables, **A**ccessibles, **I**nteropérables et **R**éutilisables.

Ensuite, viennent les enjeux d'architecture de systèmes de gestion de données, c'est-à-dire de la structure des systèmes où les données sont stockées, transformées et exploitées. Conceptuellement, il existe plusieurs types de répertoire de données. Les entrepôts de données, par exemple, sont de larges ensembles de données structurées dédiées à l'analyse d'une institution ou d'une entreprise. De façon similaire, le lac de données regroupe des données dédiées à l'analyse, mais qui ne sont pas structurées. Finalement, le magasin de données est un sous-ensemble de données structurées spécifiques, qui sont dédiées à un secteur d'activité précis. Une institution peut avoir plusieurs de ces répertoires, et transférer les données par des opérations ETL, c'est-à-dire des opérations d'extraction, transformation et chargement (*Extract*, *Transform*, *Load* en anglais).

Viennent ensuite les enjeux de gestion de données reliés au domaine d'application, la radio-oncologie dans le cadre de ces travaux. Le standard DICOM [12] est en ce sens d'une importance majeure. Il régit autant la structure de données en imagerie médicale et en radio-

oncologie que de leur communication (c.-à-d. transfert et recherche de données). Les logiciels qui permettent de stocker et interagir avec ces données sont des PACS. Dans le cadre des activités de recherche au sein du GRPM, le PACS libre *Orthanc* est utilisé.

La qualité d'un traitement de radiothérapie peut être évaluée en partie par les indices dosimétriques. C'est pourquoi un pipeline de données qui permet le calcul automatique des indices dosimétriques a été implémenté avec le langage de programmation Python. Ce dernier est présenté au chapitre 2. L'exécution quotidienne du pipeline permet d'obtenir les indices des traitements de curiethérapie faits dans la journée. Ces indices peuvent être obtenues par deux algorithmes de calcul de DVH, soit *dicompyler-core* et gMCO. Une comparaison a été faîte avec un jeu de données de 20 cas de curiethérapie HDR de prostate. Celle-ci a permis de déterminer que gMCO produit des indices beaucoup plus proches de ceux obtenus en clinique que ceux produits par *dicompyler-core*. Ceci est dû à l'algorithme de gMCO, qui utilise des techniques d'échantillonnage spécialisées à la curiethérapie (TG-43). Il est donc plus intéressant pour concevoir un jeu de données dont les indices sont proches de ceux approuvés en clinique. Il est cependant limité aux cas de curiethérapie fait avec OCP ou OCB. L'algorithme *dicompyler-core*, qui utilise la grille de dose exportée par un TPS, est plus général, et peut être utilisé avec des données dosimétriques issues de divers TPS.

La production et la collecte de données en santé sont particulièrement coûteuses. Par exemple, tel que présenté au chapitre 3, les études de radiomique nécessitent des images médicales et souvent des contours, sous la forme de segmentations (DICOM SEG) ou de structures (DICOM RTStruct). La génération de ces derniers requiert des ressources importantes car ils sont tracés par des spécialistes. C'est dans ce contexte que des flots de génération de données de radiomique, inspirés des principes FAIR, ont été idéalisés. Plus précisément, un flot de production de segmentations et de structures a été implémenté et testé. Ce flot permet de constituer un jeu de données qui respecte davantage les principes FAIR, entre autres de conserver l'information coûteuse.

Finalement, l'ingénierie de données permet de facilité les activités liées à la médecine personnalisée. L'évaluation individuelle du risque de cancer du sein pour une femme est un bon exemple. L'étude PERSPECTIVE I&I [2] est un projet de grande envergure (1949 participantes) qui a pour objectif de calculer le risque de cancer du sein de participantes grâce à un grand nombre de facteurs (habitudes de vie, marqueurs génétiques, historique familiale, *etc.*). Dans ce contexte, deux pipelines de données ont été développés afin d'automatiser certaines tâches particulièrement laborieuses. Ces derniers sont présentés au chapitre 4. Le premier vise à récupérer les facteurs précédemment mentionnés, de lancer le calcul des risques de cancer du sein et de stocker les résultats. Le deuxième, à partir de ces résultats, produit des lettres au format PDF dédiées aux participantes et à leur médecin traitant. Les deux opérations ETL se sont avérées d'une grande pertinence : elles réduisent le risque d'erreur de saisie, accélèrent

2. https://etudeperspective.ca/

significativement l'obtention de résultats et génèrent des résultats uniformes. Cependant, plusieurs défis ont été rencontrés lors du développement de ces pipelines. D'abord, une confusion de la structure des systèmes de données s'est introduite en début de projet, notamment avec l'utilisation de deux sources de données dupliquées (*REDCap* et OPAL). Aussi, le service web BOADICEA, utilisé pour faire le calcul du risque de cancer du sein, nécessitait l'envoi de données sous une forme non standardisée. Les décisions ayant mené à ces défis étaient néanmoins hors du contrôle du développeur des pipelines de données. Des recommandations d'amélioration sont énoncées à la fin du chapitre 4.

Bibliographie

[1] Karl Steinbuch. Informatik : Automatische informationsverarbeitung. *SEG-Nachrichten (Technische Mitteilungen der Standard Elektrik Gruppe)–Firmenzeitschrift*, 4 :171, 1957.

[2] Peter Naur. The science of datalogy. *Communications of the ACM*, 9(7) :485, 1966.

[3] Donald E Knuth. Computer science and its relation to mathematics. *The American Mathematical Monthly*, 81(4) :323–343, 1974.

[4] Tom Coughlin. 175 zettabytes by 2025. https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025. Accédé : 22 juin 2022.

[5] Infrastructure mondiale aws. https://aws.amazon.com/fr/about-aws/global-infrastructure/. Accédé : 22 juin 2022.

[6] Dan Noyes. Top 20 Facebook Statistics - Updated October 2019, oct 2019.

[7] Thomas G Kannampallil, Guido F Schauer, Trevor Cohen, and Vimla L Patel. Considering complexity in healthcare systems. *Journal of biomedical informatics*, 44(6) :943–947, 2011.

[8] Philippe Lambin, Ruud GPM Van Stiphout, Maud HW Starmans, Emmanuel Rios-Velazquez, Georgi Nalbantov, Hugo JWL Aerts, Erik Roelofs, Wouter Van Elmpt, Paul C Boutros, Pierluigi Granone, et al. Predicting outcomes in radiation oncology—multifactorial decision support systems. *Nature reviews Clinical oncology*, 10(1) :27–40, 2013.

[9] Philippe Lambin, Ralph TH Leijenaar, Timo M Deist, Jurgen Peerlings, Evelyn EC De Jong, Janita Van Timmeren, Sebastian Sanduleanu, Ruben THM Larue, Aniek JG Even, Arthur Jochems, et al. Radiomics : the bridge between medical imaging and personalized medicine. *Nature reviews Clinical oncology*, 14(12) :749–762, 2017.

[10] Dan LeSueur. 5 Reasons Healthcare Data Is Unique and Difficult to Measure. https://www.healthcatalyst.com/insights/5-reasons-healthcare-data-is-difficult-to-measure, jan 2019.

[11] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, and Others. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

[12] National Electrical Manufacturers Association. Digital imaging and communications in medicine (dicom) standard, nema ps3 / iso 12052, http ://www.dicomstandard.org/, 2021.

[13] Donald W Simborg. The case for the hl7 standard. *Computers in Healthcare*, 9(1) :39–40, 1988.

[14] Robert H Dolin, Liora Alschuler, Calvin Beebe, Paul V Biron, Sandra Lee Boyer, Daniel Essin, Elliot Kimber, Tom Lincoln, and John E Mattison. The hl7 clinical document architecture. *Journal of the American Medical Informatics Association*, 8(6) :552–569, 2001.

[15] Steve Lohr. The origins of 'big data' : An etymological detective story. *New York Times*, 1(1), 2013.

[16] Gil Press. A very short history of big data. *Forbes Tech Magazine, May*, 9, 2013.

[17] Doug Laney. 3D Data Management : Controlling Data Volume, Velocity, and Variety. *Application Delivery Strategies*, 949(February 2001) :4, 2001.

[18] Esra Ozcesmeci. Le lhc repousse les frontières de l'informatique. https://home.cern/fr/news/news/computing/lhc-pushing-computing-limits. Accédé : 29 avril 2020.

[19] Elvin A. Sindrilaru, Andreas J. Peters, Geoffray M. Adde, and Dirk Duellmann. EOS developments. In *Journal of Physics : Conference Series*, 2017.

[20] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The Hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010*, 2010.

[21] Giacinto Donvito, Giovanni Marzulli, and Domenico Diacono. Testing of several distributed file-systems (HDFS, Ceph and GlusterFS) for supporting the HEP experiments analysis. In *Journal of Physics : Conference Series*, 2014.

[22] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D.E. Long, and Carlos Maltzahn. CEpH : A scalable, high-performance distributed file system. In *OSDI 2006 - 7th USENIX Symposium on Operating Systems Design and Implementation*, 2006.

[23] Apache Cassandra. Apache cassandra. *Website. Available online at http ://planetcassandra. org/what-is-apache-cassandra*, 13, 2014.

[24] Jeffrey Dean Fay Chang, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable : A distributed storage system for structured data. *ACM Trans. Comput. Syst*, 26(2) :4, 2006.

[25] Tomasz Wiktorski. Spark. In *Data-intensive Systems*, pages 85–97. Springer, 2019.

[26] Ron Peters. cron. *Expert Shell Scripting*, pages 81–85, 2009.

[27] Pramod Singh. Airflow. In *Learn PySpark*, pages 67–84. Springer, 2019.

[28] Jeremy Leipzig. A review of bioinformatic pipeline frameworks. *Briefings in bioinformatics*, 18(3) :530–536, 2017.

[29] Nishant Garg. *Apache kafka*. Packt Publishing Birmingham, UK, 2013.

[30] James A Mays and Patrick C Mathias. Measuring the rate of manual transcription error in outpatient point-of-care testing. *Journal of the American Medical Informatics Association*, 26(3) :269–272, 2019.

[31] Barna Saha and Divesh Srivastava. Data quality : The other face of Big Data. In *Proceedings - International Conference on Data Engineering*, 2014.

[32] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, and William Money. Big data : Issues and challenges moving forward. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2013.

[33] Yuri Demchenko, Paola Grosso, Cees De Laat, and Peter Membrey. Addressing big data issues in Scientific Data Infrastructure. In *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013*, pages 48–55, may 2013.

[34] Yuri Demchenko, Cees De Laat, and Peter Membrey. Defining architecture components of the Big Data Ecosystem. In *2014 International Conference on Collaboration Technologies and Systems, CTS 2014*, 2014.

[35] Punam Bedi, Vinita Jindal, and Anjali Gautam. Beginning with big data simplified. In *2014 International Conference on Data Mining and Intelligent Computing, ICDMIC 2014*, 2014.

[36] Muhammad Fahim Uddin, Navarun Gupta, et al. Seven v's of big data understanding big data to extract value. In *Proceedings of the 2014 zone 1 conference of the American Society for Engineering Education*, pages 1–5. IEEE, 2014.

[37] Ammar Hameed Shnain, Hiba Jasim Hadi, Sarah Hadishaheed, and Azizahbt Haji Ahmad. Big Data and Five V'S Characteristics. *International Journal of Advances in Electronics and Computer Science*, 2015.

[38] Stockage des données. https://home.cern/fr/science/computing/processing-what-record. Accédé : 18 mai 2020.

[39] A. S. Syed Fiaz, N. Asha, D. Sumathi, and A. S. Syed Navaz. Visualization : Enhancing big data more adaptable and valuable. *International Journal of Applied Engineering Research*, 2016.

[40] Maxime Beauchemin. Apache superset, 2021.

[41] Clinton Gormley and Zachary Tong. *Elasticsearch : the definitive guide : a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.

[42] Classes de stockage amazon s3 glacier. https://aws.amazon.com/fr/s3/storage-classes/glacier/. Accédé : 23 juin 2022.

[43] MingJian Tang, Mamoun Alazab, and Yuxiu Luo. Big data for cybersecurity : Vulnerability disclosure trends and dependencies. *IEEE Transactions on Big Data*, 5(3) :317–329, 2017.

[44] Kris A. Wetterstrand. Dna sequencing costs : Data. https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data. Accédé : 26 juin 2020.

[45] Huang Fang. Managing data lakes in big data era : What's a data lake and why has it became popular in data management ecosystem. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 820–824. IEEE, 2015.

[46] Sébastien Jodogne. The Orthanc Ecosystem for Medical Imaging. *Journal of Digital Imaging*, may 2018.

[47] Access specifications. https://support.office.com/en-us/article/access-specifications-0cf3c66f-9cf2-4e32-9568-98c1025bb47c. Accédé : 29 mai 2020.

[48] Charles Parisot. The dicom standard. *The International Journal of Cardiac Imaging*, 11(3) :171–177, 1995.

[49] Peter Mildenberger, Marco Eichelberg, and Eric Martin. Introduction to the dicom standard. *European radiology*, 12(4) :920–927, 2002.

[50] Mario Mustra, Kresimir Delac, and Mislav Grgic. Overview of the dicom standard. In *2008 50th International Symposium ELMAR*, volume 1, pages 39–44. IEEE, 2008.

[51] Bernard Gibaud. The dicom standard : a brief overview. *Molecular imaging : computer reconstruction and practice*, pages 229–238, 2008.

[52] WD Bidgood Jr and Steven C Horii. Introduction to the acr-nema dicom standard. *Radiographics*, 12(2) :345–355, 1992.

[53] National Electrical Manufacturers Association. Digital imaging and communications in medicine (dicom) standard - main concepts, nema ps3 / iso 12052, https ://www.dicomstandard.org/concepts, 2021.

[54] Dicom - value representation (vr). https://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect_6.2.html. Accédé : 21 mars 2022.

[55] Stephanie Sammartino McPherson. *Tim Berners-Lee : Inventor of the World Wide Web*. Twenty-First Century Books, 2009.

[56] Bankman Isaac. Handbook of medical image processing and analysis, 2008.

[57] Brian C Green. Web access to dicom objects ("wado"), 2009.

[58] Otech img. Dicomweb, what is it and how can it be used ?, 2018.

[59] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.

[60] Carlos Costa, Carlos Ferreira, Luís Bastião, Luís Ribeiro, Augusto Silva, and José Luís Oliveira. Dicoogle-an open source peer-to-peer pacs. *Journal of digital imaging*, 24(5) :848–856, 2011.

[61] Easypacs. http://mehmetsen80.github.io/Easypacs. Accédé : 23 juin 2022.

[62] Roman Baygildin. Neurdicom. https://github.com/reactmed/neurdicom. Accédé : 23 juin 2022.

[63] Dcm4che. https://github.com/dcm4che/dcm4chee-arc-light. Accédé : 23 juin 2022.

[64] G Zeilinger. dcm4che, a dicom implementation in java. *URL (07.07. 2009) : http ://www. dcm4che. org*, 2006.

[65] Gabriel Couture and Philippe Després. PyOrthanc, sep 2019.

[66] Jean-Emmanuel Bibault, Philippe Giraud, and Anita Burgun. Big data and machine learning in radiation oncology : state of the art and future prospects. *Cancer letters*, 382(1) :110–117, 2016.

[67] Alexandru Nicolae, Niranjan Venugopal, and Ananth Ravi. Trends in targeted prostate brachytherapy : from multiparametric mri to nanomolecular radiosensitizers. *Cancer nanotechnology*, 7 :6, 07 2016.

[68] Mack Roach III, I-Chow Hsu, Het al Chung, Gerard Morton, Leonard G Gomella, Robert E Wallace, Ben Movsas, Deborah Watkins Bruner, Andrea M Barsevick, Deborah Citrin, et al. Radiation therapy oncology group rtog 0924. androgen deprivation therapy and high dose radiotherapy with or without whole-pelvic radiotherapy in unfavorable intermediate or favorable high risk prostate cancer : a phase iii randomized trial, 2018.

[69] E Panitsa, JC Rosenwald, and C Kappas. Developing a dose-volume histogram computation program for brachytherapy. *Physics in Medicine & Biology*, 43(8) :2109, 1998.

[70] Ravinder Nath, Lowell L Anderson, Gary Luxton, Keith A Weaver, Jeffrey F Williamson, and Ali S Meigooni. Dosimetry of interstitial brachytherapy sources : recommendations of the aapm radiation therapy committee task group no. 43. *Medical physics*, 22(2) :209–234, 1995.

[71] Benjamin Nelms, Cassandra Stambaugh, Dylan Hunt, Brian Tonner, Geoffrey Zhang, and Vladimir Feygelman. Methods, software and datasets to verify DVH calculations against analytical values : Twenty years late(r). *Medical Physics*, 42(8) :4435–4448, 2015.

[72] Aditya Panchal, Gabriel Couture, Gertsikkema, Nicolas Galler, Hideki Nakamoto, David C Hall, and Akihisa Wakita. Dicompyler/dicompyler-core v0.5.5, jun 2019.

[73] Cédric Bélanger, Sylviane Aubin, Luc Beaulieu, and Éric Poulin. Commissioning of gpu-based multi-criteria optimizer combined with plan navigation tools for high-dose-rate brachytherapy. *Journal of Contemporary Brachytherapy*, 14(1).

[74] Hubert Y Pan, Lukasz M Mazur, Neil E Martin, Charles S Mayo, Lakshmi Santanam, Todd Pawlicki, Lawrence B Marks, and Benjamin D Smith. Radiation oncology health information technology : Is it working for or against us ? *International Journal of Radiation Oncology Biology Physics*, 98(2) :259–262, 2017.

[75] Herb Brody. Medical imaging. *Nature*, 502(7473) :S81–S81, 2013.

[76] Martin Vallières, Alex Zwanenburg, Bodgan Badic, Catherine Cheze Le Rest, Dimitris Visvikis, and Mathieu Hatt. Responsible radiomics research for faster clinical translation, 2018.

[77] Bioportal : Radiomics ontology. https://bioportal.bioontology.org/ontologies/RO. Accédé : 27 juin 2022.

[78] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia) : maintaining and operating a public information repository. *Journal of digital imaging*, 26(6) :1045–1057, 2013.

[79] EJ Limkin, Roger Sun, Laurent Dercle, EI Zacharaki, Charlotte Robert, Sylvain Reuzé, Antoine Schernberg, Nikos Paragios, Eric Deutsch, and Charles Ferté. Promises and challenges for the implementation of computational medical imaging (radiomics) in oncology. *Annals of Oncology*, 28(6) :1191–1206, 2017.

[80] Fred Prior, Kirk Smith, Ashish Sharma, Justin Kirby, Lawrence Tarbox, Ken Clark, William Bennett, Tracy Nolan, and John Freymann. The public cancer radiology imaging collections of the cancer imaging archive. *Scientific data*, 4(1) :1–7, 2017.

[81] H. J. W. L. Aerts, L. Wee, E. Rios Velazquez, R. T. H. Leijenaar, C. Parmar, P. Grossmann, and P. ... Lambin. Data from nsclc-radiomics, the cancer imaging archive [data set], 2019. https ://doi.org/10.7937/K9/TCIA.2015.PF0M9REI.

[82] Steve Pieper, Michael Halle, and Ron Kikinis. 3D Slicer. In *2004 2nd IEEE International Symposium on Biomedical Imaging : Macro to Nano*, 2004.

[83] Csaba Pinter, Andras Lasso, An Wang, David Jaffray, and Gabor Fichtinger. SlicerRT : Radiation therapy research toolkit for 3D Slicer. *Medical Physics*, 2012.

[84] Andriy Fedorov, David Clunie, Ethan Ulrich, Christian Bauer, Andreas Wahle, Bartley Brown, Michael Onken, Jörg Riesmeier, Steve Pieper, Ron Kikinis, et al. Dicom for quantitative imaging biomarker development : a standards based approach to sharing clinical data and structured pet/ct analysis results in head and neck cancer research. *PeerJ*, 4 :e2057, 2016.

[85] Michael Q Stearns, Colin Price, Kent A Spackman, and Amy Y Wang. Snomed clinical terms : overview of the development process and project status. In *Proceedings of the AMIA Symposium*, page 662. American Medical Informatics Association, 2001.

[86] Charles S Mayo, Jean M Moran, Walter Bosch, Ying Xiao, Todd McNutt, Richard Popple, Jeff Michalski, Mary Feng, Lawrence B Marks, Clifton D Fuller, et al. American association of physicists in medicine task group 263 : standardizing nomenclatures in radiation oncology. *International Journal of Radiation Oncology* Biology* Physics*, 100(4) :1057–1066, 2018.

[87] Danahé LeBlanc. Analyse radiomique du cancer de la prostate pour la prédiction du pronostic des patients avec un grand risque de récidive. Master's thesis, Université Laval, 2021.

[88] Marco Eichelberg, Joerg Riesmeier, Thomas Wilkens, Andrew J Hewett, Andreas Barth, and Peter Jensch. Ten years of medical imaging standardization and prototypical implementation : the dicom standard and the offis dicom toolkit (dcmtk). In *Medical Imaging 2004 : PACS and Imaging Informatics*, volume 5371, pages 57–68. SPIE, 2004.

[89] Darcy Mason. Su-e-t-33 : pydicom : an open source dicom library. *Medical Physics*, 38(6Part10) :3493–3493, 2011.

[90] Rex A Cardan, Elizabeth L Covington, and Richard A Popple. An open source solution for improving tg-263 compliance. *Journal of applied clinical medical physics*, 20(9) :163–165, 2019.

[91] Romaana Mir, Sarah M Kelly, Ying Xiao, Alisha Moore, Catharine H Clark, Enrico Clementel, Coreen Corning, Martin Ebert, Peter Hoskin, Coen W Hurkmans, et al. Organ at risk delineation for radiation therapy clinical trials : Global harmonization group consensus guidelines. *Radiotherapy and Oncology*, 150 :30–39, 2020.

[92] William C Sleeman IV, Joseph Nalluri, Khajamoinuddin Syed, Preetam Ghosh, Bartosz Krawczyk, Michael Hagan, Jatinder Palta, and Rishabh Kapoor. A machine learning method for relabeling arbitrary dicom structure sets to tg-263 defined labels. *Journal of Biomedical Informatics*, 109 :103527, 2020.

[93] Ministère de la Santé et des Services sociaux. Programme québécois de dépistage du cancer du sein (pqdcs). https://www.quebec.ca/sante/conseils-et-prevention/depistage-et-offre-de-tests-de-porteur/programme-quebecois-de-depistage-du-cancer-du-sein. Accédé : 29 juin 2022.

[94] Yiwey Shieh, Donglei Hu, Lin Ma, Scott Huntsman, Charlotte C Gard, Jessica WT Leung, Jeffrey A Tice, Celine M Vachon, Steven R Cummings, Karla Kerlikowske, et al. Breast cancer risk prediction using a clinical risk model and polygenic risk score. *Breast cancer research and treatment*, 159(3) :513–525, 2016.

[95] AJ Lee, AP Cunningham, KB Kuchenbaecker, N Mavaddat, DF Easton, and AC Antoniou. Boadicea breast cancer risk prediction model : updates to cancer incidences, tumour pathology and web interface. *British journal of cancer*, 110(2) :535–545, 2014.

[96] Stephanie Archer, Chantal Babb de Villiers, Fiona Scheibl, Tim Carver, Simon Hartley, Andrew Lee, Alex P Cunningham, Douglas F Easton, Jennifer G McIntosh, Jon Emery, et al. Evaluating clinician acceptability of the prototype canrisk tool for predicting risk of breast and ovarian cancer : A multi-methods study. *PLoS One*, 15(3) :e0229999, 2020.

[97] Tim Carver, Simon Hartley, Andrew Lee, Alex P Cunningham, Stephanie Archer, Chantal Babb de Villiers, Jonathan Roberts, Rod Ruston, Fiona M Walter, Marc Tischkowitz, et al. Canrisk tool—a web interface for the prediction of breast and ovarian cancer risk and the likelihood of carrying genetic pathogenic variants. *Cancer Epidemiology, Biomarkers & Prevention*, 30(3) :469–473, 2021.

[98] Étude perspective sur le dépistage du cancer du sein. https://etudeperspective.ca/. Accédé : 29 juin 2022.

[99] Paul A Harris, Robert Taylor, Brenda L Minor, Veida Elliott, Michelle Fernandez, Lindsay O'Neal, Laura McLeod, Giovanni Delacqua, Francesco Delacqua, Jacqueline Kirby, et al. The redcap consortium : Building an international community of software platform partners. *Journal of biomedical informatics*, 95 :103208, 2019.

[100] Paul A Harris, Robert Taylor, Robert Thielke, Jonathon Payne, Nathaniel Gonzalez, Jose G Conde, et al. A metadata-driven methodology and workflow process for providing translational research informatics support. *J Biomed Inform*, 42(2) :377–81, 2009.

[101] Margaret M Eberl, Chester H Fox, Stephen B Edge, Cathleen A Carter, and Martin C Mahoney. Bi-rads classification for management of abnormal mammograms. *The Journal of the American Board of Family Medicine*, 19(2) :161–164, 2006.

[102] Pulsar. https://pulsar.ca/. Accédé : 28 juin 2022.

[103] Dany Doiron, Yannick Marcon, Isabel Fortier, Paul Burton, and Vincent Ferretti. Software application profile : Opal and mica : open-source software solutions for epidemiological data management, harmonization and dissemination. *International journal of epidemiology*, 2017.

[104] The Document Foundation. Libreoffice writer, 2020.

[105] The pandas development team. pandas-dev/pandas : Pandas, Feb 2020.

[106] Kenneth Reitz. Requests is a simple, yet elegant, http library. https://pypi.org/project/requests/.

[107] Scott Burns and Aaron Browne. Version 1.0, may 2014.

[108] Booke Haarsma. Performs a mail merge on docx (microsoft office word) files. https://pypi.org/project/docx-mailmerge/.

[109] Cédric Bélanger, Songye Cui, Yunzhi Ma, Philippe Després, J. Adam M. Cunha, and Luc Beaulieu. A GPU-based multi-criteria optimization algorithm for HDR brachytherapy. *Physics in Medicine and Biology*, 2019.

[110] Darcy Mason. SU-E-T-33 : Pydicom : An Open Source DICOM Library. *Medical Physics*, 38(6) :3493, 2011.

Annexe A

Annexe

A.1 PyOrthanc

Orthanc, un serveur DICOM libre, peut être utilisé dans de nombreuses situations où des données sous le format DICOM doivent être manipulées. Son API REST (voir 1.5.1) permet notamment d'interagir avec ce serveur par programmation. En effet, il suffit d'envoyer des requêtes HTTP afin de récupérer, stocker ou supprimer des objets DICOM, ou encore d'exécuter certaines opérations, comme anonymiser un fichier DICOM. La fonctionnalité d'envoi de requêtes HTTP est supportée par Python, ce qui permet à des scripts écrits dans ce langage de programmation d'interagir avec une instance du serveur Orthanc. Il est néanmoins nécessaire de formater les requêtes pour qu'elles soient interprétables par Orthanc, en plus de connaître les routes qui peuvent être appelées. Afin d'éviter de dupliquer le développement de code Python pour interagir avec Orthanc, une librairie facile d'utilisation qui accomplit ce rôle a été développée, soit PyOrthanc.

PyOrthanc est une librairie Python qui permet de manipuler Orthanc grâce à des appels HTTP. Il s'agit d'un client en Python qui couvre toutes les routes de l'API d'Orthanc. Celle-ci a de nombreuses fonctionnalités ; elle permet notamment d'importer des fichiers DICOM

```python
from pyorthanc import Orthanc

orthanc = Orthanc('http://adresse_du_serveur_orthanc')
orthanc.setup_credentials('nom-utilisateur', 'mot-de-passe')  # Si nécessaire

with open('A_DICOM_INSTANCE_PATH.dcm', 'rb') as file_handler:
    orthanc.post_instances(file_handler.read())
```

Segment 1 – Exemple montrant comment importer un fichier DICOM dans Orthanc via PyOrthanc.

dans un serveur Orthanc, tel que présenté dans le segment de code 1. À la ligne 3, un objet `Orthanc` est créé avec l'adresse HTTP en paramètre. Il est par la suite possible d'inclure les informations d'identification dans cette instance, comme montré à la ligne 4, si l'identification est nécessaire. Finalement, il est possible d'envoyer le fichier DICOM, sous forme d'octets, au serveur Orthanc grâce à la méthode `post_instances`. Notons que grâce à cette méthode, il n'est pas nécessaire de formater la requête HTTP pour effectuer cette action, comme cela aurait été le cas sans PyOrthanc.

Le segment de code 2 présente des exemples qui permettent de récupérer de l'information sur les objets DICOM qui se trouvent dans un serveur Orthanc. Ces exemples montrent comment il est possible de descendre dans la hiérarchie des niveaux DICOM (patient, étude, série, instance, voir 1.4 pour plus de détail) afin d'accéder aux données de chacun de ces niveaux.

```
1  from pyorthanc import Orthanc
2
3  orthanc = Orthanc('http://adresse_du_serveur_orthanc')
4
5  # Pour récupérer les données des patients
6  for patient_id in orthanc.get_patients():
7      patient_info = orthanc.get_patient_information(patient_id)
8
9      patient_name = patient_info['MainDicomTags']['name']
10     ...
11
12 # Pour récupérer les données des études d'un patient
13 for study_id in patient_info['Studies']:
14     study_info = orthanc.get_study_info(study_id)
15
16     study_date = study_info['MainDicomTags']['StudyDate']
17     ...
18
19 # Pour récupérer les données des séries d'une étude
20 for series_id in study_info['Series']:
21     series_info = orthanc.get_series_info(series_id)
22
23     modality = series_info['MainDicomTags']['Modality']
24     ...
25
26 # Pour récupérer les données des instances d'une série
27 for instance_id in series_info['Instances']:
28     instance_info = orthanc.get_instance_information(instance_id)
29     ...
```

Segment 2 – Exemples de cas d'utilisation de la librairie PyOrthanc.

Il est également possible, avec PyOrthanc, de générer une structure de données en arbre qui suit la hiérarchie des niveaux DICOM. Cette structure de données est montrée à la figure A.1. Selon la profondeur dans cet arbre, un noeud peut être un objet `Patient`, `Study`, `Series` ou `Instance`. Chacun de ces objets contient un identifiant pointant vers la ressource qui leur correspond dans un serveur Orthanc. Cette structure facilite les interactions complexes avec les objets DICOM, notamment pour les opérations de filtrage de données.

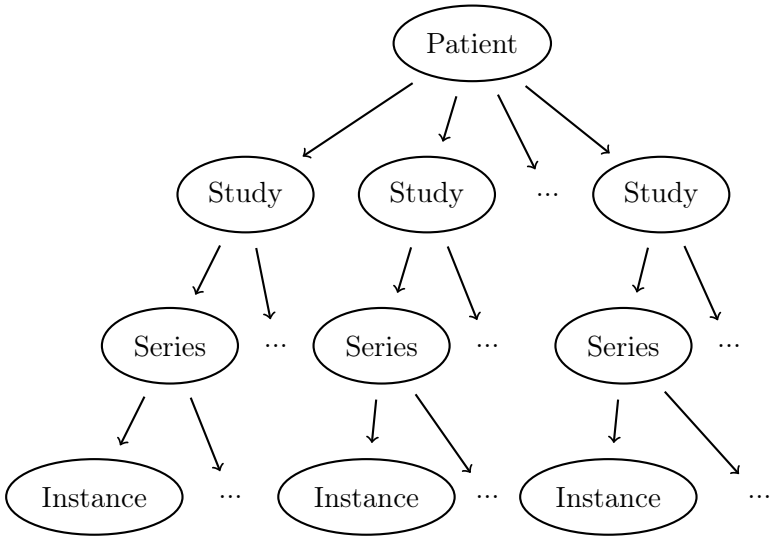


FIGURE A.1 – Arbre d'un patient généré avec PyOrthanc.

Le segment de code 3 présente un exemple de construction de la structure de données mentionnée plus haut. La fonction permettant de la générer, `build_patient_forest`, a comme argument des filtres, soit `patient_filter`, `study_filter`, `series_filter` et `instance_filter`. Ces filtres doivent être des *callables* Python, comme une fonction, qui retourne un booléen. Les fonctions aux lignes 3 et 6 en sont des exemples. L'arbre est construit en conséquence. Dans cet exemple, les filtres passés en argument aux lignes 13 et 14 font en sorte que les arbres résultant ne sont que des patients ayant «Gabriel» dans leur nom, et que les seules séries conservées sont celle de modalité «CT». Il est par la suite de manipuler d'avantage, sans les avoir préalablement récupérées. En effet, chaque méthode de type *getter* des objets `Patient`, `Study`, `Series` et `Instance` fait un appel HTTP au serveur Orthanc. Par exemple, l'exécution de la méthode `study.get_date()` envoie une requête au serveur Orthanc pour récupérer la date de l'étude.

```python
from pyorthanc import Orthanc, build_patient_forest, Patient, Series

def is_named_gabriel(patient: Patient) -> bool:
    return 'Gabriel' in patient.get_name()

def is_ct(series: Series) -> bool:
    return series.get_modality() == 'CT'

orthanc = Orthanc('http://adresse-du-serveur-orthanc')

patient_forest = build_patient_forest(
    orthanc,
    patient_filter=is_named_gabriel,
    series_filter=is_ct,
)

for patient in patient_forest:
    patient.get_name()
    patient.get_zip()  # Récupère tous les fichiers DICOM reliés au patient
    ...

    for study in patient.get_studies():
        study.get_date()
        study.get_referring_physician_name()
        ...

        for series in study.get_series():
            ...
```

Segment 3 – Exemple de construction d'arbres de patients.

Les fonctionnalités de communications DICOM avec d'autres PACS, soit C-Echo, C-Find, C-Store, C-Move, etc. (la section 1.4 détaille davantage la communication DICOM), sont également accessibles par PyOrthanc, ce qui permet entre autres de programmer des opérations complexes de transfert de données. Le segment de code 4 présente un exemple de communication DICOM où une requête, qui correspond à l'opération DICOM C-Find, est envoyée à un autre PACS, et est suivie d'un transfert de données, qui correspond à l'opération DICOM C-Move, entre ce PACS vers un autre.

```python
from pyorthanc import RemoteModality, Orthanc

orthanc = Orthanc('http://adresse-du-serveur-orthanc')

remote_modality = RemoteModality(orthanc, 'une_modalité_pré_e')

# Requête (C-Find)
data = {'Level': 'Study', 'Query': {'PatientID': '*'}}
query_response = remote_modality.query(data=data)

# Transfert (C-Move) des résultats de la
# précédente requête à une modalité donnée.
remote_modality.move(query_response['QUERY_ID'], 'une_modalité_donnée')
```

Segment 4 – Exemple de communication DICOM effectuant une requête C-Find à un PACS et un transfert de données (C-Move) à un autre.

A.2 *Dicompyler-core*

Dicompyler-core [72] est une librairie libre qui permet de calculer des DVH à partir des fichiers DICOM RTStruct et RTDose. Elle utilise la grille de dose présente dans le RTDose et elle la superpose aux structures présentes dans le RTStruct. Le calcul de DVH est donc effectué avec une dose déjà planifiée et avec une grille à pas constant. Le segment de code 5 présente un exemple de calcul de DVH. Dans celui-ci, le DVH de la structure "2" est obtenu. Il est finalement possible d'obtenir les indices dosimétriques, accessibles en tant qu'attributs.

```python
from dicompylercore import dvhcalc

# Calculer le DVH
dvh = dvhcalc.get_dvh("rtstruct.dcm", "rtdose.dcm", 2)

dvh.max, dvh.min, dvh.D2cc
# (3.0899999999999999, 0.029999999999999999, dvh.DVHValue(2.96, 'Gy'))
```

Segment 5 – Extrait de code où un DVH est calculé à partir des fichiers RTStruct et RTDose.

Dicompyler-core permet d'interpoler (c.-à-d. ajouter) des segments entre les contours, un enjeu pour le calcul de DVH présenté à la section 2.3. Le segment de code 6 présente comment effectuer cette interpolation.

```
# Calculer le DVH
dvh = dvhcalc.get_dvh("rtstruct.dcm", "rtdose.dcm", 2,
                      interpolation_segments_between_planes=1)
```

Segment 6 – Extrait de code montrant comment interpoler un segment entre les contours.

A.3 *Apache Airflow*

Apache Airflow[1] est une plateforme logicielle pour créer, planifier et surveiller des flux d'opérations. Plus précisément, elle permet de déployer des séquences d'opérations sous la forme de scripts Python ou Bash qui seront exécutées de façon périodique, quotidiennement ou hebdomadaire par exemple. *Airflow* contient également une interface Web permettant de visualiser l'état des flux d'opérations, notamment en collectant les journaux produits par ces derniers. Les flux d'opérations au sein d'*Airflow* prennent la forme de graphiques acycliques dirigés, nommés DAG (*Directly Acycled Graph*). Un exemple de DAG est présenté au segment de code 7.

```
from airflow import DAG
from airflow.operators.python_operator import PythonOperator

from une_librairie import taches

dag = DAG(
    dag_id='Un_id',
    schedule_interval='@daily'
)

premiere_tache = PythonOperator(
    python_callable=taches.premiere_tache,
    dag=dag
)
deuxieme_tache = PythonOperator(
    python_callable=taches.deuxieme_tache
    dag=dag
)

premiere_tache >> deuxieme_tache
```

Segment 7 – Exemple de DAG pouvant être déployé dans la plateforme *Airflow*.

Le DAG du segment de code 7 contient deux tâches, soit celles des lignes 11 et 25. Ces tâches

1. https://airflow.apache.org/

contiennent un *Callable* Python, comme une fonction, qui est appelée à chaque exécution du DAG. La séquence d'exécution des tâches est définie à la ligne 20, où le symbôle `>` indique l'ordre. Par exemple, si nous avions trois tâches à exécuter dans le DAG, l'ordre serait écrit comme `premiere_tache >> deuxieme_tache >> troisieme_tache`. Dans cet exemple du segment de code 7, le DAG est planifié pour être exécuté quotidiennement (ligne 8).

A.4 *brachy-dose-calculation-microservice* et *brachy-dose-calculation-client*

brachy-dose-calculation-microservice est un service Web, ou micro-service, qui permet de calculer les DVH d'une planification de traitement de curiethérapie. Plus précisément, cette application est bâtit sur l'algorithme gMCO[109] (*GPU-basedmulti-criteria*), qui a pour objectif de calculer les DVH pour des cas de curiethérapie à la prostate fait avec le logiciel de planification de traitement *Oncentra Prostate*. Afin d'être performant, gMCO utilise un GPU (*Graphical Processing Unit*) pour ses calculs. Cette caractéristique le contraint à être utilisé que sur de machines étant équipée avec ces unités. Puisqu'il est peu intéressant d'équiper un grand nombre de machines d'un GPU, une interface REST (*Representational state transfer*) a été construite. Cette interface, implémentée en Python avec le framework web *Falcon*[2], permet d'utiliser les fonctionnalités de calcul de DVH de gMCO à partir de n'importe quelle application sur le réseau clinique. Grâce à cette interface web, il devient possible d'obtenir un fichier RTDose avec le sous-module DICOM DVH peuplé en envoyant une requête HTTP contenant les fichiers DICOM RTStruct et RTPlan à *brachy-dose-calculation-microservice*.

2. https://falcon.readthedocs.io/en/stable/

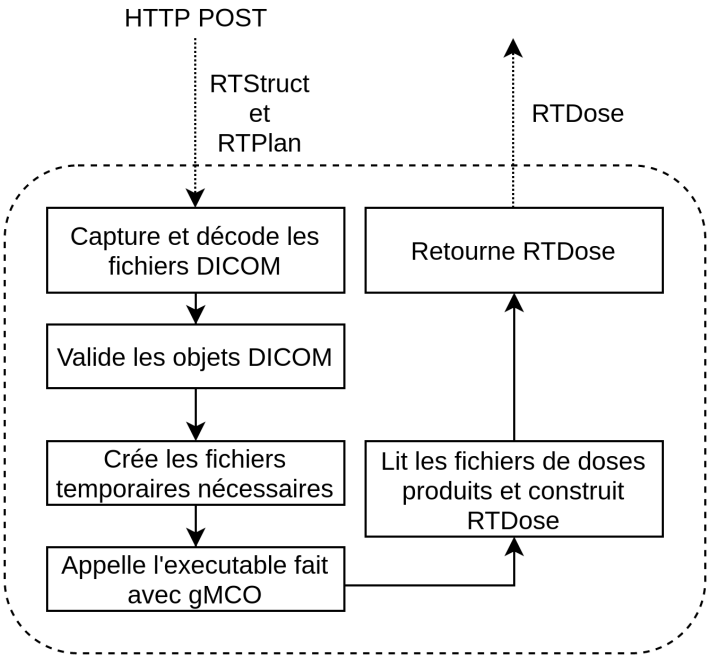


FIGURE A.2 – Schéma du flot d'exécution de l'application *brachy-dose-calculation-microservice*

La séquence d'exécution à l'intérieur de *brachy-dose-calculation-microservice* est montrée à la figure A.2 ; 1) À la réception d'une requête HTTP de type POST, les données brutes des fichiers RTPlan et RTStruct sont capturées et décodées avec la librairie *pydicom*[110] ; 2) Les objets DICOM validés, c'est-à-dire qu'il est vérifié s'ils contiennent les données nécessaire afin de calculer les DVH ; 3) Plusieurs fichiers temporaires nécessaire à l'exécutable fait avec gMCO sont créés ; 4) l'exécutable fait avec gMCO est appelé. Ce dernier lit les fichiers temporaires créés à l'étape précédente et créé un nouvel ensemble de fichier contenant les données de dose (ex. fichiers de texte contenant des DVH) ; 5) Les fichiers produits par l'exécutable sont lus et permettre de créer un fichier DICOM RTDose, en peuplant notamment le sous-module DICOM DVH ; 6) Le nouveau fichier RTDose est placé dans une réponse qui est renvoyée au client. Il est à noter qu'au moment d'écrire ce mémoire que la grille de dose n'est pas écrite dans le fichier RTDose présent dans la réponse.

brachy-dose-calculation-client est une librairie Python afin d'interagir avec le service Web *brachy-dose-calculation-microservice*. Elle permet d'y envoyer facilement les objets DICOM RTStruct et RTPlan. Le segment de code 9 présente deux exemples. Dans le premier, à la ligne 6, seuls les chemins des fichiers DICOM RTStruct et RTPlan doivent être donnés en arguments à la méthode `calculate_dose`, qui retourne un objet `pydicom.FileDataset` de modalité RTDose, qui est une représentation d'un fichier DICOM RTDose. Dans le deuxième exemple d'utilisation, à la ligne 13, ce sont les objets `pydicom.FileDataset` de modalité RTStruct et RTPlan qui sont donnés en arguments.

```python
import pydicom
from brachy_dose_calculation_client import BrachyDoseCalculationClient

client = BrachyDoseCalculationClient('http://service-url')

rtdose, response = client.calculate_dose('rtstruct-path.dcm', 'rtplan-path.dcm')
print(rtdose.Modality) # RTDose

# ou
rtstruct = pydicom.dcmread('rtstruct-file-path.dcm')
rtplan = pydicom.dcmread('rtplan-file-path.dcm')

rtdose, response = client.calculate_dose(rtstruct, rtplan)
print(rtdose.Modality) # RTDose
```

Segment 8 – Exemples d'utilisation la librairie *brachy-dose-calculation-client* afin d'obtenir un fichier RTDose

A.5 Guide de segmentations

A.5.1 Authors

— Gabriel Couture : gabriel.couture.4@ulaval.ca

— Danahé LeBlanc : danahe.leblanc.1@ulaval.ca

A.5.2 Introduction

The Digital Imaging and Communications in Medicine (DICOM) is a standard for storing, communicating and managing medical imaging data and related data. The DICOM standard is both a file type and a communication standard. For the file type, you can think of it as a much more general and rich format of '.png' or '.jpg' file types. Indeed, the '.png' or '.jpg' file types contain the image data and context metadata. In the same fashion, the DICOM file contains the resulting data of a modality data acquisition and metadata. Each modality (CT, IRM, US, SEG, RTDOSE, RTSTRUCT, etc) has its own DICOM module format. For more information on the format, you can navigate through the DICOM format documentation there : https://dicom.innolitics.com/ciods.

Segmentation is a core concept of radiation therapy. Therefore, it is no surprise that there are dedicated DICOM modules. A cool thing about the DICOM format is that when you create a Segmentation DICOM object from CT images for instance, the Segmentation file will keep the patient and context metadata so that your Segmentation DICOM file is linked to the CT DICOM files. When storing your newly created Segmentation file on a PACS, the

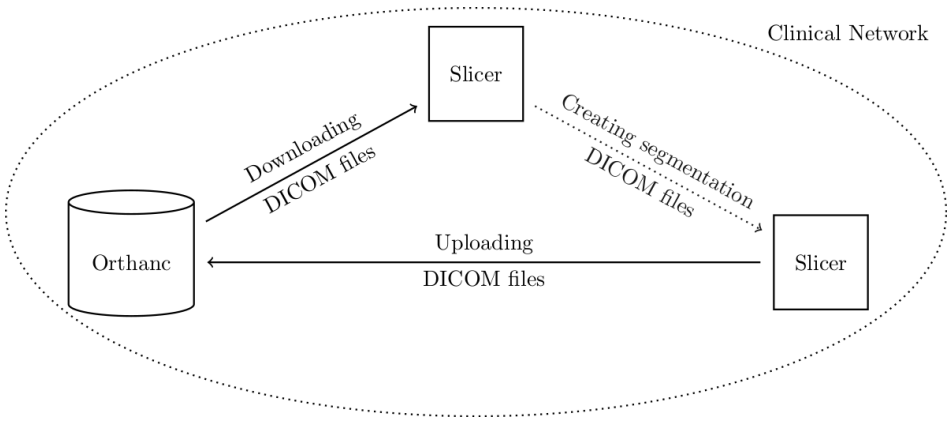


FIGURE A.3 – Segmentation workflow in a nutshell

Segmentation will be stored along with the CT's Study. The related data is therefore gathered in the same place.

We usually *don't want to do segmentation on anonymous patient*. As mentioned above, a newly created Segmentation DICOM object will keep meta-data from the CT/MRI/... DICOM files. Doing segmentation on an anonymous patient will more likely result in the loss of precious information. As the segmentation process is quite costly in time, we wish to keep data as much as possible. This allows the reuse of the data in different research projects.

A.5.3 Getting and Uploading patient DICOM files

The used DICOM server – or research PACS – at GRPM (*Groupe de Recherche en Physique Médicale*) is an Orthanc server. Orthanc is an open-source mini-PACS that allows us to store and manipulate DICOM files. We want, as much as possible, that all DICOM files generated by our research operations to be listed in.

A.5.4 Getting the DICOM files

Typically, a research project that use patient data has a list of patient IDs. At the Hotel-Dieu de Québec, these IDS usually start with the '03HDQ' prefix. You can have access to the DICOM files of a patient through the Orthanc web interface (fig. A.4). To access it, open your web browser and go to the address `https://oncoweb:8042`. You will notice that credentials are required. Note that this website is only accessible through the clinic network. Indeed, as Orthanc contains non-anonymous data, we want to keep the patient data inside the clinical environment. This means that the data can not be exported outside (e.g. to a laptop that will be reconnected to other networks).

Once on the page, enter the patient ID in the corresponding input field. After, hit the "Do lookup", it will output the matching studies. Then, select the desired study. After clicking on it, a interface similar the the fig. A.5 will show up. It is now possible to download the

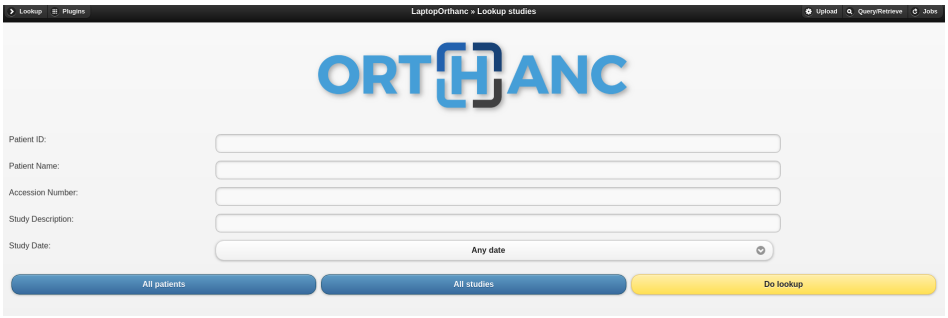


FIGURE A.4 – Orthanc home page

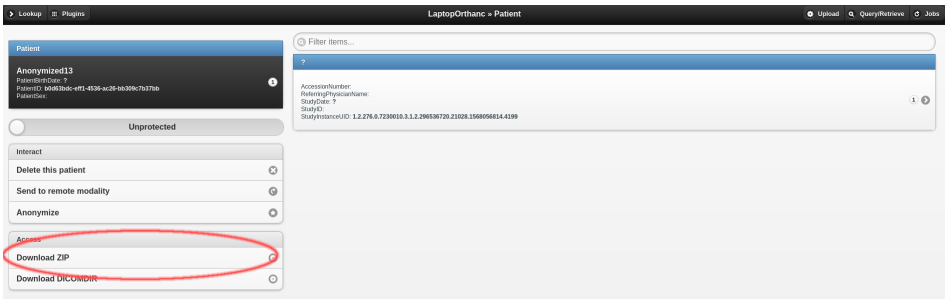


FIGURE A.5 – Download patient interface

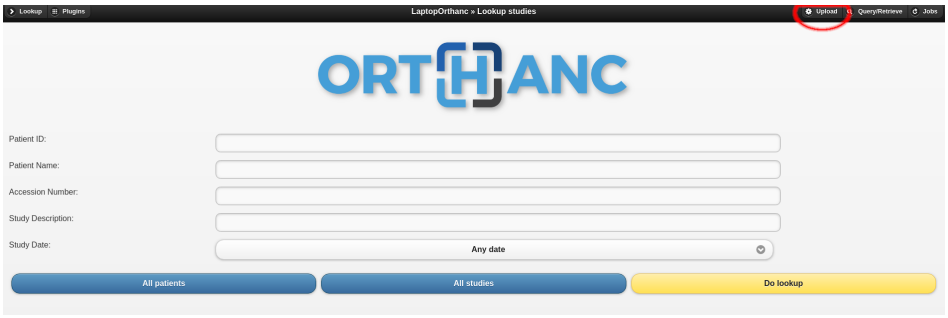


FIGURE A.6 – Orthanc upload button

studies by clicking on "Download ZIP". Once downloaded, extract the files in the ZIP file at the desired path on your computer. The result is directories that contain the DICOM files.

A.5.5 Uploading DICOM files

Uploading DICOM files is easy too. From Orthanc's home page, click on the 'Upload' button (fig. A.6).

Once on the upload page, you can upload DICOM files by clicking on the "Select files to upload ..." button (fig. A.7). Then, select your files.

You should see the name of the file in the pendent section (fig. A.8). Then, click on the "Start the upload" button and it's done! Orthanc will automatically store your DICOM files

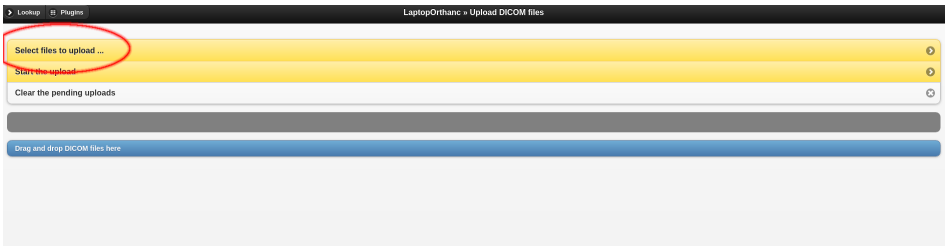


FIGURE A.7 – Orthanc's uploading interface

FIGURE A.8 – Orthanc's uploading interface with pending files

according to their metadata.

A.5.6 Getting patient files with Python

Python is a programming language with which you build scripts. Scripts are very powerful when it comes to performs complex operations, such as retrieving a large batch of patients. Orthanc's REST API help us to do those operations. For convenience, a Python library that call PyOrthanc has been made, *PyOrthanc* `https://https://pypi.org/project/pyorthanc/`. It is possible to install *PyOrthanc* directly with pip with the command line :

```
pip install pyorthanc
```

Assuming you have a list of patient IDs, you may put them in a `json` file, which is a file format that plays well with scripts. Your `json` file may look like this :

```
[
 "FIRST_PATIENT_ID",
 "SECOND_PATIENT_ID",
 "THIRD_PATIENT_ID"
]
```

Then, a script similar to this will allow you to download and store the DICOM files of your patients. In this example, only the *CT* DICOM files will be retrieved.

```
    import json
    import pyorthanc

    # Get our patients files
    with open('path/to/your/patient-ids.json') as file_handler:
           IENT_IDS = json.load(file_handler)

    # Building an Orthanc object
    orthanc = pyorthanc.Orthanc('http://oncoweb:8042')
    orthanc.setup_credentials('pacs', 'pacs')

    # This parses Orthanc to find correspondences.
    # It will keep patient that are in the PATIENT_IDS list and the series
    # that are CT.
    # Note that the build_patient_forest function can take a while to run
    # on Orthanc server with a lot of patients.
    # Indeed, the function indexes patients/studies/series/instances
    # that are allowed considering the filters.
    # Adding specific "filter" can greatly reduce the computing time.
    patients = pyorthanc.build_patient_forest(
            orthanc=orthanc,
            patient_filter=lambda p: p.get_id() in PATIENT_IDS,
            series_filter=lambda s: s.get_modality() == 'CT'
    )

    # Then download and write the patients
    pyorthanc.retrieve_and_write_patients_forest_to_given_path(
            patients,
            './path/where/dicom/will/be/write'
    )
```

Segment 9 – Exemples d'utilisation la librairie *brachy-dose-calculation-client* afin d'obtenir un fichier RTDose

A.5.7 Using Slicer3D to create segmentation and to create SEG and SR filer

Slicer3D is an open source software platform for medical imaging, image processing, segmentation creation and three-dimensional visualization (https://www.slicer.org/). The recommended version is the latest stable release (4.10.2 built 2019-05-22/30) which uses Python 2.7.13. This version ensures the compatibility with the 'QuantitativeReportng' extension (https://qiicr.gitbooks.io/quantitativereporting-guide/).

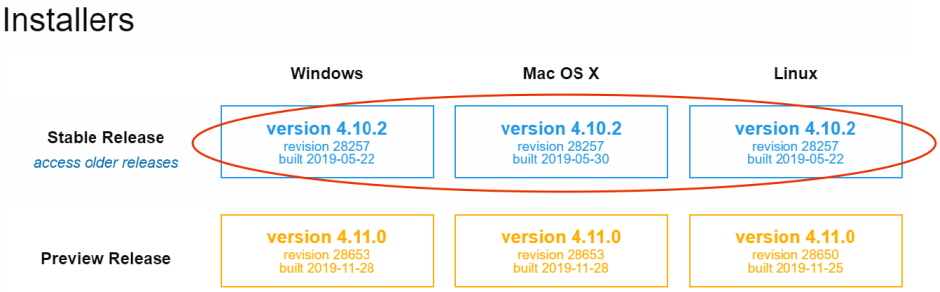


FIGURE A.9 – Download page for the latest stable release of Slicer3D for different platforms

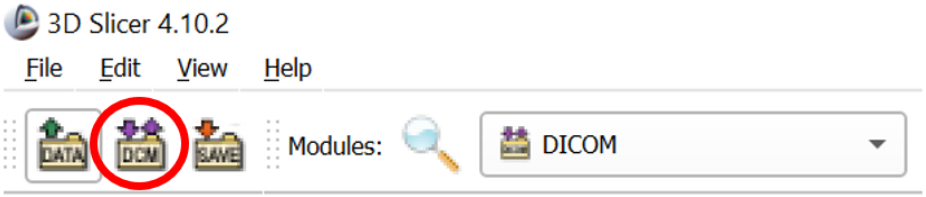


FIGURE A.10 – Shortcut icon to open the DICOM Browser

As discussed in the introduction, there are different DICOM modalities that allow to keep the segmentations and metadata related to them (creator name, image used for segmentation, etc.). All the information is therefore available in a DICOM Structured Report (SR) with a referenced DICOM segmentation (SEG). This section therefore presents how to use Slicer3D to create segmentations as well as to create SR and SEG files.

A.5.8 Installation

Download the latest stable release from `http://download.slicer.org` : 4.10.2 build 2019-05-22/30. To install the extension follow the instructions in the "Installation and Upgrade" section : `https://qiicr.gitbooks.io/quantitativereporting-guide/docs/install.html`.

A.5.9 Import and Load DICOM files

To load and import DICOM files, go to 'DICOM' module and click on "Show DICOM Browser" or click on the icon presented in fig. A.10.

The DICOM Browser is shown in fig. A.11.

1. Import : all DICOM files in the selected folder (including subfolders) will be parsed and basic information from file headers will be stored in Slicer's DICOM database. If enabled, then Slicer will make a copy of the imported files into the database folder.
2. Patient list : shows patients in the database. Studies available for the selected patient(s) are listed in study list. Multiple patients can be selected

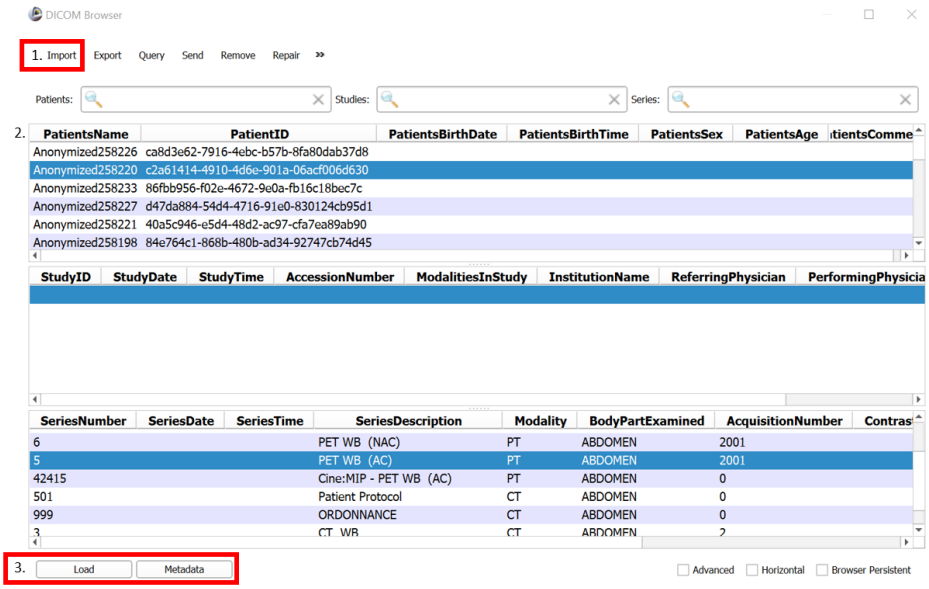


FIGURE A.11 – Slicer's DICOM Browser

3. Series list : shows list of series (images, structure sets, segmentations, registration objects, etc.) available for selected studies.
4. Metadata : click this button to see all the DICOM tags stored in file headers of selected series.
5. Load : click to load currently selected loadables into slicer.

A.5.10 Using the 'Quantitative Reporting' extension

When you first use the extension, it asks you to enter your name. The desired form is the following : `FIRSTNAMELASTNAME`. In this way, when recording segmentations, and other information, it is ensured that the person who added the information to the patient is known. The DICOM tag created is (0070,0084) which is called "Content Creator's Name Attribute". The creator's name can be changed when saving data. The one entered during the first use is only the default one.

To use the previously downloaded DICOM dataset in Slicer3D in the 'Quantitative Reporting' extension, two steps highlighted in fig. A.12 must be completed :

1. It is first necessary to create a new table in the "Mesurement report" tab.
2. In the drop-down menu "Master volume" now available, select the desired master volume to enable editing.

Now it is possible to notice that the extension header, as shown in the example in fig. 11, includes the data set information. The different interface components of the extensions are des-

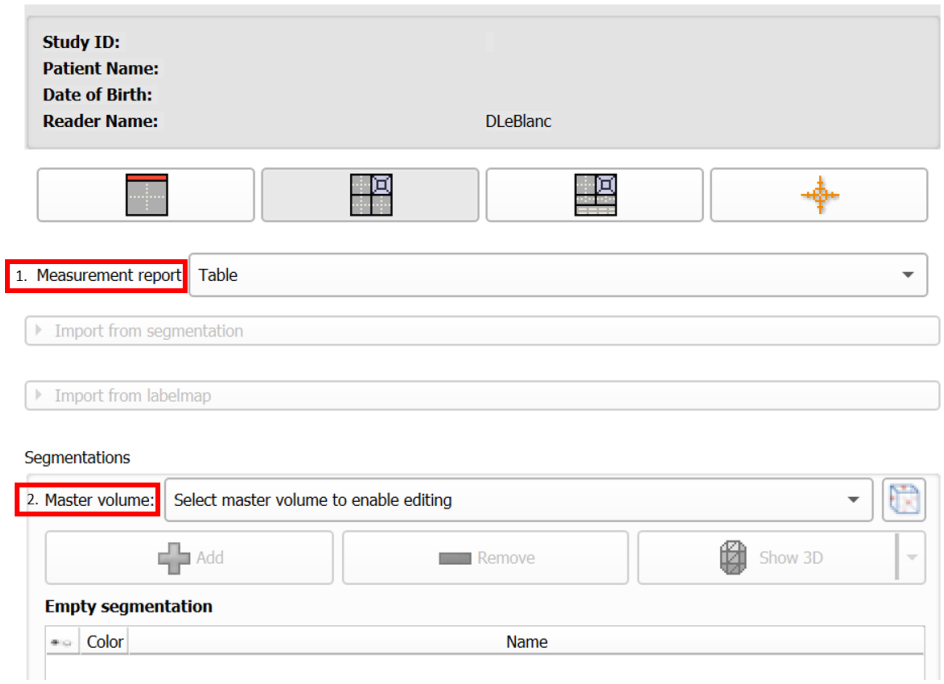


FIGURE A.12 – Initial configuration of the extension Quantitative Reporting, when no dataset is connected.

cribed in the "User Interface" section of the extension's home site : https://qiicr.gitbooks.io/quantitativereporting-guide/content/docs/user-interface.html.

A.5.11 Create segmentations

More precisely, here are the sections that are more interesting for the creation of segmentations.

3. This segmentation zone is based on the same functions of the on the following website : https://slicer.readthedocs.io/en/latest/user_guide/module_segmenteditor.html. See the tutorials for examples of how to apply the functions. Slicer3D base module : 'Segment editor'. All functions are described.

- To use an existing segmentation, refer to section *Import an existing Segmentation.*

4. You can either save a report and continue later or complete the current report. Either way a DICOM Structured Report (SR) with a referenced DICOM Segmentation (SEG) will be created and stored into the Slicer DICOM database. Be sure to change the creator's name if it is not the default one.

- Save Report : Will create the partially completed DICOM Structured Report, which could be continued at a later time.
- Complete Report : Will create the completed DICOM Structured Report representing the final version which usually wouldn't be modified afterwards.

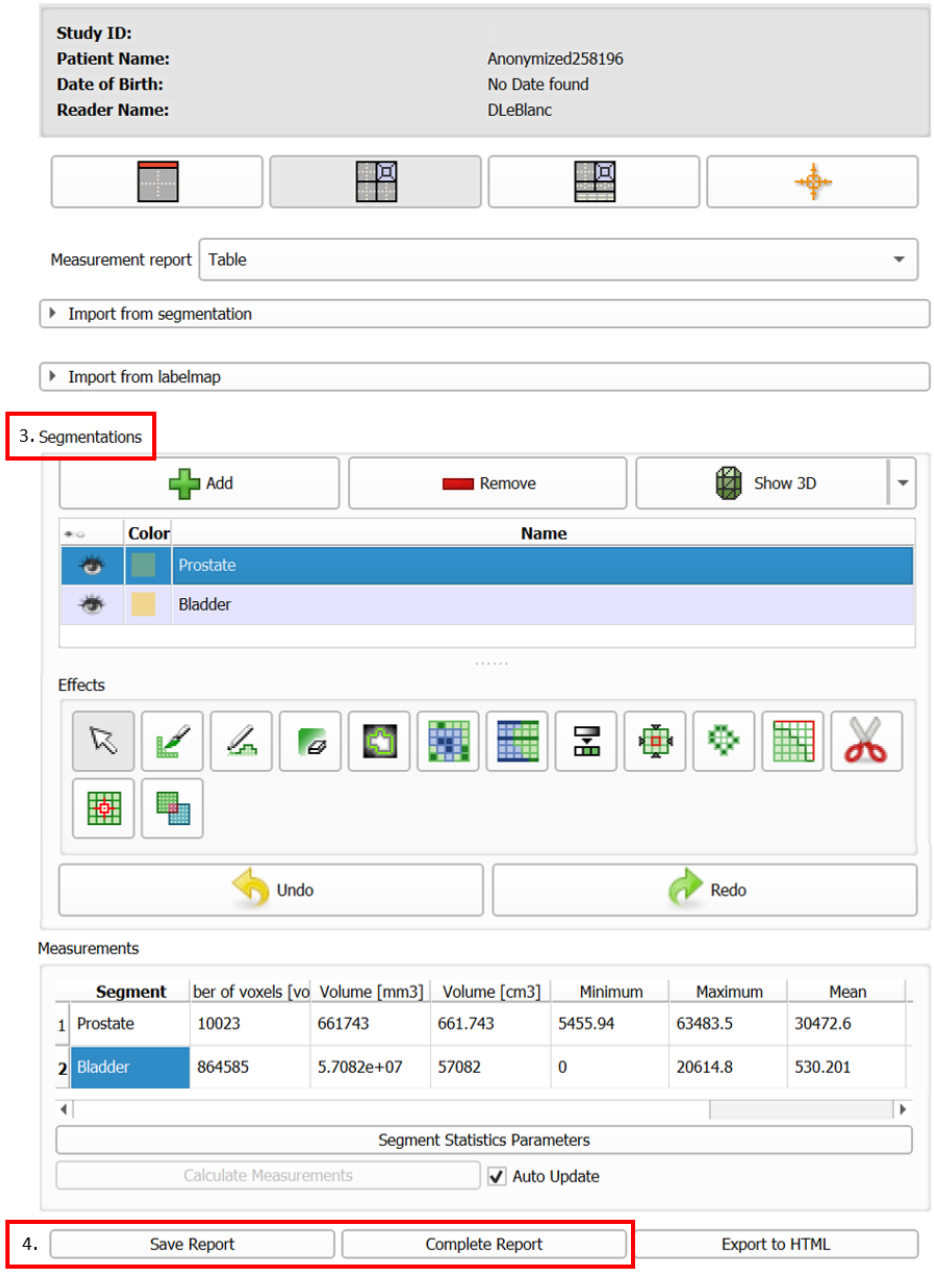


FIGURE A.13 – Interface of the Quantitative Reporting extension.

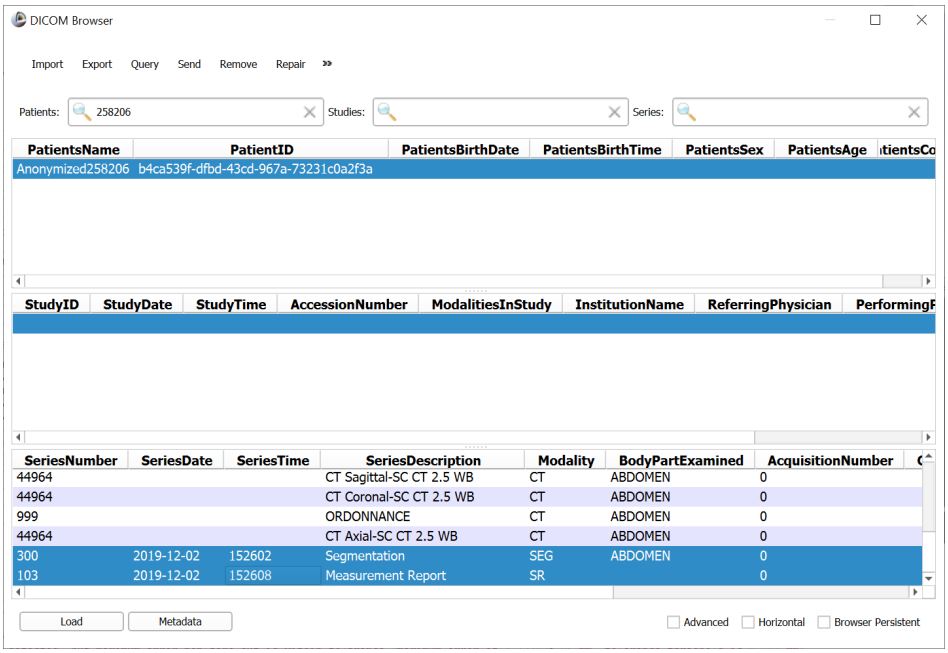


FIGURE A.14 – Presence of SR and SEG reports in the DICOM Browser.

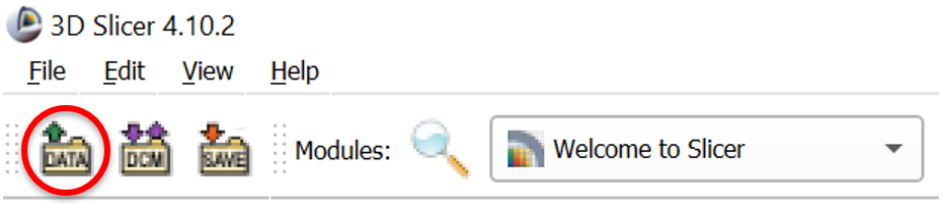


FIGURE A.15 – Shortcut icon to open window to add data to scene.

When the report is successfully saved, it is possible to find the SR and SEG in the DICOM Browser, as shown in fig. A.14.

A.5.12 Import an existing segmentation

It is possible to import existing segmentations in order to create the desired DICOM files. On the other hand, the "Import from segmentations" section creates an error when saving the report. The solution is therefore to create a labelmap and import it from the "Import from labelmap" section.

First, load the desired segmentation with the "add data into the scene" window by pressing the button in fig. A.15. Then drag and drop the segmentation onto the patient's study (See fig. A.16 for before and after)

In the *Segmentations* module presented in fig. A.17, there is a section called "export/import models and labelmaps". Choose the *export* mode as operator and the *labelmap* mode as output mode. In advanced section, it is possible to link the segmentation to the correct image modality.

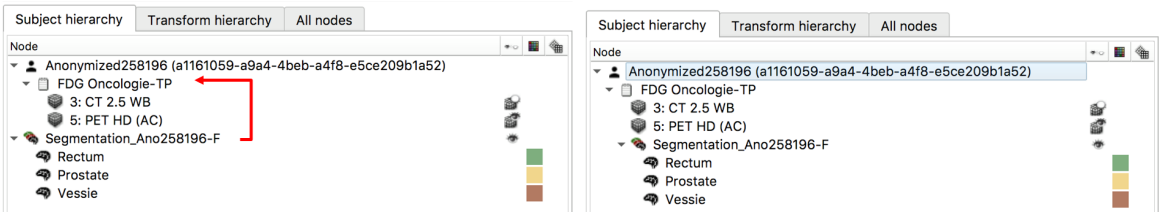


FIGURE A.16 – Illustration of the data after downloading : (Left) before adding the segmentation to the study (the red arrow representsthe drag-and-drop movement) (Right) after adding.

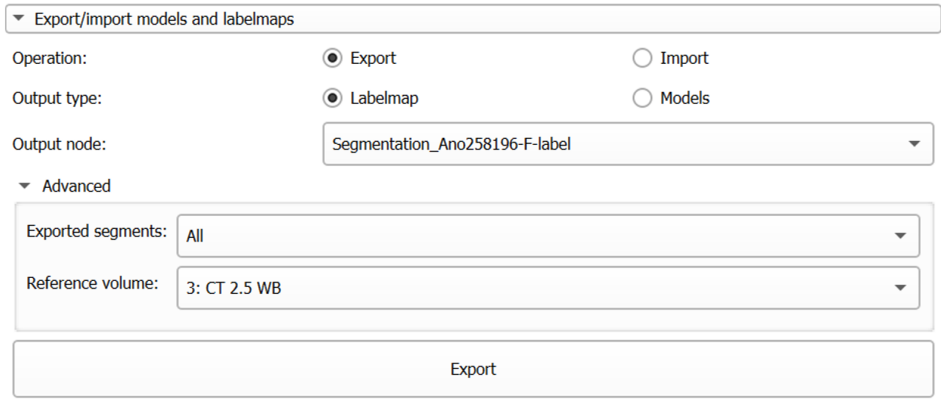


FIGURE A.17 – Section "Export/import models and labelmaps" of module Segmentations.

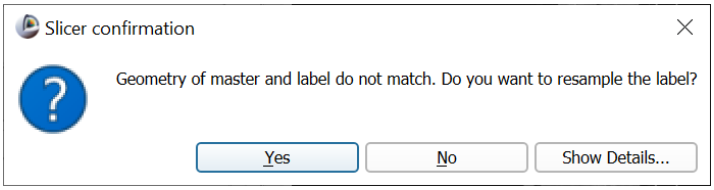


FIGURE A.18 – Message obtained when no master volume is assigned to the segmentation when exporting in labelmap.

If this is not done, when importing into the *Quantitative Reporting* extension, the window in fig. A.18 will appear. Just click on "No" to keep the segment informations (names and colors).

A.5.13 Export and create DICOM files

To obtain the DICOM files and thus be able to send them to Orthanc, the procedure is quite simple. There are two ways to open the DICOM Export :

— Press the "Export" button in the DICOM Browser (See *Import and Load DICOM files* section for information on how to open it)

— In the 'Data' module, right-click on the desired study for export and select "Export to Dicom..."

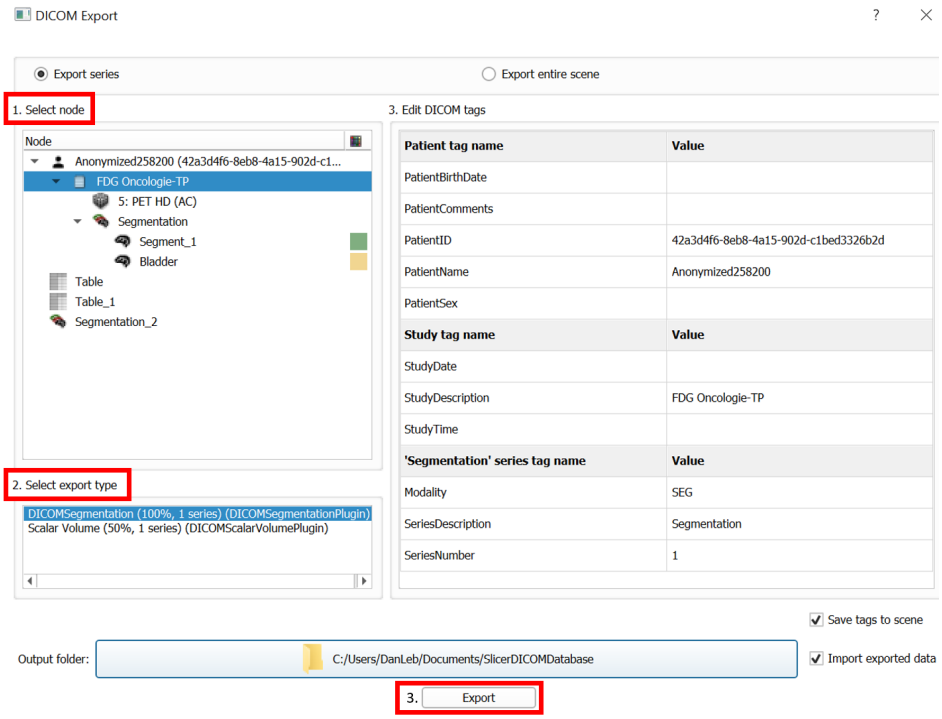


FIGURE A.19 – Interface of the DICOM Export window.

As shown in fig. A.19, the "Export series" mode is selected to select only SR and SEG files that are connected to the correct modality of image.

1. It is necessary to select the desired study.
2. Select the DICOMSegmentation to export. It contains the SR and the SEG files.
3. It is possible to choose the location of the DICOM files to download and export them with the "Export" button.

Folders containing ".dcm" files are now created !

A.6 Catégories de risque de cancer du sein

Le tableau A.1 présente comment PERSPECTIVE I&I [98] a définit les catégories de risque de cancer du sein. Une catégorie est associée à une participante selon son âge et son risque d'être d'avoir un cancer du sein dans les dix prochaines années, en pour mille (‰).

Âge [années]	Risque près de la population générale [‰]	Risque intermédiaire [‰]	Risque élevé [‰]
40	$x < 20$	$20 \leq x < 36$	$x \geq 36$
41	$x < 22$	$22 \leq x < 38$	$x \geq 38$
42	$x < 23$	$23 \leq x < 41$	$x \geq 41$
43	$x < 25$	$25 \leq x < 43$	$x \geq 43$
44	$x < 26$	$26 \leq x < 46$	$x \geq 46$
45	$x < 28$	$28 \leq x < 48$	$x \geq 48$
46	$x < 29$	$29 \leq x < 50$	$x \geq 50$
47	$x < 30$	$30 \leq x < 52$	$x \geq 52$
48	$x < 31$	$31 \leq x < 54$	$x \geq 54$
49	$x < 32$	$32 \leq x < 56$	$x \geq 56$
50	$x < 33$	$33 \leq x < 58$	$x \geq 58$
51	$x < 34$	$34 \leq x < 60$	$x \geq 60$
52	$x < 35$	$35 \leq x < 62$	$x \geq 62$
53	$x < 37$	$37 \leq x < 64$	$x \geq 64$
54	$x < 38$	$38 \leq x < 67$	$x \geq 67$
55	$x < 40$	$40 \leq x < 70$	$x \geq 70$
56	$x < 42$	$42 \leq x < 72$	$x \geq 72$
57	$x < 44$	$44 \leq x < 76$	$x \geq 76$
58	$x < 46$	$46 \leq x < 79$	$x \geq 79$
59	$x < 48$	$48 \leq x < 83$	$x \geq 83$
60	$x < 50$	$50 \leq x < 86$	$x \geq 86$
61	$x < 51$	$51 \leq x < 89$	$x \geq 89$
62	$x < 53$	$53 \leq x < 92$	$x \geq 92$
63	$x < 55$	$55 \leq x < 95$	$x \geq 95$
64	$x < 56$	$56 \leq x < 97$	$x \geq 97$
65	$x < 57$	$57 \leq x < 99$	$x \geq 99$
66	$x < 58$	$58 \leq x < 100$	$x \geq 100$
67	$x < 58$	$58 \leq x < 100$	$x \geq 100$
68	$x < 58$	$58 \leq x < 100$	$x \geq 100$
69	$x < 57$	$57 \leq x < 99$	$x \geq 99$
70	$x < 57$	$57 \leq x < 98$	$x \geq 98$

TABLEAU A.1 – Catégories de risque de cancer du sein telles que défini par PERSPECTIVE I&I. Le risque, en pour mille (‰), correspond au risque d'avoir un cancer du sein dans les dix prochaines années.

A.7 Pedigree


```
{
  "mut_freq": "UK",
  "cancer_rate": "Canada",
  "user_id": "end_user_id",
  "pedigree_data":
    "##CanRisk 1.0
    ##Menarche=13
    ##Parity=3
    ##First_live_birth=24
    ##OC_use=F:10
    ##MHT_use=E
    ##BMI=19.766
    ##Alcohol=21.46
    ##Menopause=52
    ##Height=172.72
    ##PRS_BC=alpha=0.4406
    ##FamID Name Target IndivID FathID MothID Sex MZtwin Dead Age Yob BC1 BC2 OC PRO PAN Ashkn BRCA1 BRCA2 PALB2 ATM CHEK2 RAD51D RAD51C BRIP1 ER:PR:HER2:CK14:CK56
    1 index 1 index fath moth F 1 0 60 1960 0 0 0 0 0 1 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 part 0 part 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 moth 0 moth mgfath mgmoth F 0 1 75 1935 0 0 75 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 mgmoth 0 mgmoth 0 0 F 0 1 55 1895 55 55 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 mgfath 0 mgfath 0 0 M 0 1 65 1885 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 fath 0 fath pgfath pgmoth M 0 1 65 1925 65 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 pgmoth 0 pgmoth 0 0 F 0 1 0 0 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 pgfath 0 pgfath 0 0 M 0 1 0 0 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 dau1 0 dau1 part index F 2 0 37 1984 25 35 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 dau2 0 dau2 part index F 2 0 35 1986 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 son1 0 son1 part index M 0 1 25 1985 0 0 0 25 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 sis1 0 sis1 fath moth F 1 1 65 1950 55 55 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 sis2 0 sis2 fid1 moth F 0 0 69 1952 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 fid1 0 fid1 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 bro1 0 bro1 fath moth M 0 0 66 1955 0 0 0 45 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 nie1 0 nie1 fid2 sis1 F 0 0 49 1972 AU AU 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 fid2 0 fid2 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 nie2 0 nie2 bro1 mid1 F 0 1 15 1974 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 mid1 0 mid1 0 0 F 0 0 0 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 nep1 0 nep1 fid3 sis2 M 0 0 45 1976 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 fid3 0 fid3 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0
    1 patunc1 0 patunc1 pgfath pgmoth M 0 1 85 0 0 0 0 0 0 0 0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0 0:0:0:0:0",
}
```


Segment 10 – Exemple de l'objet JSON consommable par l'API REST BOADICEA