Machine learning for large observational datasets in healthcare

Maryan Morel

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAX013

Thèse de doctorat

Machine learning for large observational datasets in healthcare

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Ecole polytechnique

École doctorale n°574 Ecole Doctorale de Mathématique Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 6 Mai 2021, par

MARYAN MOREL

Composition du Jury :

Stanley Durrleman Directeur de recherche, ICM – Paris Brain Institute (ARAMIS)	Président
Jean-Philippe Vert Chercheur, Google Brain & Mines ParisTech (CBIO)	Rapporteur
Rodolphe Thiebaut Directeur de recherche, Université de Bordeaux (INSERM U1219)	Rapporteur
Mounia N. Hocine Professeure, Conservatoire National des Arts et Métiers (MESuRS)	Examinateur
Stéphane Gaïffas Professeur, Université Paris Diderot (LPSM)	Directeur de thèse
Emmanuel Bacry Directeur de recherche, Université Paris-Dauphine (CEREMADE)	Co-directeur de thèse
Marc Lavielle Directeur de recherche, INRIA Saclay	Invité

Centre de Mathématiques Appliquées
Ecole Polytechnique

Thèse de doctorat de l'Institut Polytechnique de Paris
en mathématiques appliquées

Directeurs de thèse: Emmanuel BACRY,
Stéphane GAÏFFAS

Machine learning for large observational datasets in healthcare

Maryan MOREL
maryan.morel@polytechnique.edu

Palaiseau, 2021

To my parents André & Rosa, To Marine,
and to all the health workers.

Acknowledgements

Je souhaite tout d'abord remercier mes directeurs de thèse, Stéphane Gaïffas et Emmanuel Bacry, qui m'ont guidé tout au long de ce travail. Merci Stéphane, pour les avalanches d'idées dont tu as le secret, pour tes astuces d'implémentation, pour ton talent à dénicher les dernières librairies Python ou Julia, et la découverte de quelques groupes de métal. Merci Emmanuel pour tes intuitions, ta capacité à clarifier des raisonnements parfois flous, pour avoir été à l'origine du partenariat sans lequel rien de tout cela n'aurait été possible, sans oublier tes recommandations gastronomiques. Merci à vous deux pour votre humour, votre humanité et votre générosité. Ces quelques lignes ne suffisent pas à exprimer toute ma gratitude pour vos enseignements, vos encouragements et les nombreuses opportunités que vous m'avez offertes durant ces quelques années passées à vos côtés.

Ce travail n'aurait pas été possible sans le soutien du centre de mathématiques appliquées de l'École Polytechnique, de l'école doctorale de mathématiques Hadamard et de la Caisse nationale de l'assurance maladie.

Je remercie particulièrement Jean-Philippe Vert et Rodolphe Thiebaut d'avoir accepté de rapporter ma thèse, j'en suis très honoré. Je leur sais gré de l'intérêt porté à mon travail et de leur lecture critique. J'exprime aussi ma reconnaissance à Mounia N. Hocine, Stanley Durrleman et Marc Lavielle pour leur participation à mon jury de thèse.

Un grand merci à Agathe, Anastasiia, Benjamin, Dian, Fanny, Moussa, Phong et Youcef avec qui j'ai eu la chance de partager l'écriture de plusieurs articles. Je garde un souvenir ému de nos débats et des longues soirées passées à finaliser les soumissions. Merci Agathe, tes idées ont souvent rendu infiniment simples des problèmes que je pensais complexes. Merci Anastasiia, ta persévérance et ta bonne humeur n'ont jamais cédé face à d'eventuels bugs, des résultats inattendus et à un co-auteur absorbé par sa rédaction de thèse. Merci Fanny pour m'avoir guidé dans les méandres du SNDS et de l'administration.

I also want to thank the fantastic engineers Xristos, Prosper, Sathiya, Daniel, Dian, Firas, Youcef, Søren, Philip, Kevin, Muhammad, and Angel, who played an essential part in `tick`'s or `scalpel`'s development. I cherished our endless debates on functional programming, Linux distributions, and niche programming languages. A special thank to товарищ Youcef, who has been a debugging wizard and the benevolent dictator behind `scalpel`'s codebase management.

Je n'oublie pas d'exprimer ma gratitude aux doctorants et postdoctorants avec qui j'ai passé de très bons moments durant ces dernières années. En particulier, Martin qui m'a fait connaître cette équipe. Merci à ceux qui m'ont précédé, Alain, Massil et Simon, ainsi que Yiyang, Peng, Marcello et Qing pour les moments partagés autour d'un café ou en école d'été. Merci également à Ling, Mioty et Matthews, dont le court passage parmi nous a initié une tradition d'exploration culinaire au sein de l'équipe.

Many thanks to all of you, magnificent nerds, for having discussed topics as weird as interesting, such as horses[1], the many ways of cooking ducks, mechanical keyboards, typography, coffee brewing, gaming, metal bands, computer vision applied to pigeon tracking, beers and all kind of fermented stuff, GPU-heated plankton crops which populated the office, and the unsuspected ties between football, sociology, and communism. I will truly miss these moments.

Merci à Stéphane, Dario Colazzo et Julie Josse avec qui j'ai effectué mes missions complémentaires d'enseignement. Cette expérience m'a beaucoup appris, et j'espère pouvoir continuer à partager mes connaissances dans la suite de mon parcours.

Je tiens également à remercier l'administration du CMAP et la DSI pour leur accompagnement durant ce doctorat. Plus particulièrement, merci à Nasséra, Sylvain et Pierre pour leur aide et leurs conseils lors de l'acquisition de serveurs et autres cartes graphiques.

Mes derniers remerciements, et non des moindres, s'adressent à l'ensemble de ma famille et de mes amis. Votre joie et votre présence à mes côtés tout au long de cette thèse ont été précieux. Merci à ceux d'entre vous qui se sont risqués à la relecture de certaines parties de cette thèse. J'espère qu'à la vue de ce travail, vous pardonnerez ces moments passés en compagnie d'un individu accaparé par son code et à l'humeur parfois changeante.

Un immense merci à mes parents, André et Rosa, pour votre confiance et votre soutien. Le choix de ce sujet de thèse n'est pas étranger aux valeurs que vous m'avez transmises, et qui m'ont accompagnées tout au long de ce travail.

Enfin, merci Marine, pour ton amour, ta force et tes encouragements. Je ne pense pas qu'il m'eut été possible de surmonter cette épreuve sans toi. J'attends avec impatience les prochaines aventures que l'avenir nous réserve, en espérant qu'elles soient les plus folles.

[1] Especially water poneys.

Works featured in this thesis

Published articles

[Bac+20] Emmanuel Bacry, Stéphane Gaiffas, Fanny Leroy, Maryan Morel, Dinh-Phong Nguyen, Youcef Sebiat, and Dian Sun. "SCALPEL3: a scalable open-source library for healthcare claims databases." In: *International Journal of Medical Informatics* (2020), p. 104203.

[Mor+20a] Maryan Morel, Emmanuel Bacry, Stéphane Gaiffas, Agathe Guilloux, and Fanny Leroy. "ConvSCCS: convolutional self-controlled case series model for lagged adverse event detection." In: *Biostatistics* 21.4 (2020), pp. 758–774.

Submitted articles

[Kab+20] Anastasiia Kabeshova, Maryan Morel, Emmanuel Bacry, and Stephane Gaiffas. *Which attention model and unsupervised pretraining strategy for electronic health records?* 2020. Submitted.

[Mor+20b] Maryan Morel, Benjamin Bouyer, Agathe Guilloux, Moussa Laanani, Fanny Leroy, Dinh Phong Nguyen, Youcef Sebiat, Emmanuel Bacry, and Stephane Gaiffas. *Screening anxiolytics, hypnotics, antidepressants and neuroleptics for bone fracture risk among elderly: a nation-wide dynamic multivariate self-control study using the SNDS claims database.* 2020. Submitted.

Oral communications

[Mor17a] Maryan Morel. *ConvSCCS: Convolutional Self-Controlled Case Series Model for Lagged Adverse Event Detection*. Data Science Summer School, Sept. 2017.

[Mor17b] Maryan Morel. *Learning temporal events relationships*. Facebook Core ML Data Science Seminar, Sept. 2017.

[Mor18] Maryan Morel. *ConvSCCS: Convolutional Self-Controlled Case Series Model for Lagged Adverse Event Detection*. Machine Learning for Healthcare, Aug. 2018.

[Mor19] Maryan Morel. *Screening adverse drug reactions using the French large observational healthcare database, SNIIRAM*. JSTAR 2019, Apr. 2019.

CONTENTS

Works featured in this thesis **iii**

Contents **v**

Introduction **1**

1 Use of large observational databases for research 1
1.1 Characterization of large observational databases in healthcare 2
1.2 Barriers to methodological research 5
1.3 Contribution: a framework for reproducible and fast data processing 8
2 Adverse drug reactions detection 14
2.1 Modeling challenges 14
2.2 Mathematical tools 17
2.3 Common methodologies 23
2.4 Selected approach 26
2.5 Contribution: Convolutional SCCS 29
2.6 Applications 32
2.7 Discussion 38
3 Learning representations for health data 40
3.1 Deep learning architectures for healthcare 42
3.2 Pre-training strategies. 48
3.3 Contribution: attention and pre-training strategies comparison. 50
3.4 Experiments 54

I SCALPEL3: a scalable open-source library for healthcare claims databases 57

I.1 Introduction 58
I.2 Background 59
I.3 Material and Methods 60
I.3.1 The SNDS database 61
I.3.2 SCALPEL3: a SCAlable Pipeline for hEaLth data 61
I.3.3 SCALPEL-Flattening: denormalization of the data 62
I.3.4 SCALPEL-Extraction: extraction of concepts 64

I.3.5 SCALPEL-Analysis: interactive manipulation and analysis of cohorts . . . 66
I.4 Results . . . 67
I.5 Discussion . . . 69
I.6 Conclusion . . . 72
I.7 Summary Table . . . 72
I.8 Declarations of interest . . . 73
I.9 Authors' contribution . . . 73
I.10 Acknowledgments . . . 74

Appendix
I.A Scalpel Analysis usage examples . . . 74
I.B List of SNDS databases currently denormalized. . . . 79
I.C List of available extractors . . . 80
I.D List of the available transformers . . . 81

II ConvSCCS: convolutional self-controlled case series model for lagged adverse event detection 83
II.1 Introduction . . . 84
II.2 Self-controlled case series models . . . 86
II.2.1 Conditional Poisson regression and SCCS models . . . 86
II.2.2 Risk screening . . . 88
II.3 ConvSCCS: an extension of SCCS models . . . 89
II.3.1 Discrete convolutional SCCS . . . 90
II.3.2 Penalised estimation . . . 91
II.4 Experiments . . . 92
II.4.1 Simulations . . . 93
II.4.2 Application on data from the French national health insurance information system . . . 96
II.5 Conclusion . . . 100

Appendix
II.A Likelihood in SCCS models . . . 101
II.B Discrete time SCCS . . . 102
II.C Numerical implementation . . . 103
II.D Software . . . 103
II.E Simulations details . . . 103

III Screening anxiolytics, hypnotics, antidepressants and neuroleptics for bone fracture risk among elderly. 111
III.1 Introduction . . . 112

III.2 Materials and methods . . . 114
III.2.1 Data Source . . . 114
III.2.2 Study design . . . 114
III.2.3 Case definition . . . 114
III.2.4 Exposure definition . . . 115
III.2.5 Statistical Analysis . . . 116
III.2.6 Sensitivity and subgroup analysis . . . 116
III.2.7 Software . . . 118
III.3 Results . . . 118
III.3.1 All fractures . . . 119
III.4 Discussion . . . 122
III.4.1 Key results . . . 122
III.4.2 Limitations . . . 123
III.4.3 Interpretation . . . 127
III.5 Conclusion . . . 129

Appendix
III.A Codes . . . 140
III.B Sensitivity analysis . . . 142
III.C SCCS assumption assessment . . . 186

IV Attention and unsupervised pre-training for EHR 187
IV.1 Introduction . . . 187
IV.2 Methods . . . 189
IV.2.1 Models architecture . . . 190
IV.2.2 Unsupervised pre-training . . . 193
IV.2.3 Supervised fine-tuning, losses and metrics . . . 193
IV.2.4 Hyper-parameters and training details . . . 194
IV.3 Experiments . . . 195
IV.4 Results . . . 196
IV.5 Conclusion . . . 197

Appendix
IV.A Encoders . . . 198
IV.A.1 Vanilla transformer . . . 198
IV.A.2 Linear transformer . . . 199
IV.A.3 Graph Attention Network . . . 199
IV.B Unsupervised Pre-training Strategies . . . 200
IV.B.1 Masked Language Model . . . 201
IV.B.2 Triplet loss . . . 201
IV.B.3 Contrastive Predictive Coding . . . 201

Conclusion **203**

A Résumé des contributions **207**
A.1 Utilisation de bases de données observationnelles 207
A.1.1 Contribution : un logiciel d'extraction rapide et reproductible de concepts médicaux 211
A.2 Détection d'effets indésirables médicamenteux 213
A.2.1 Défis méthodologiques 214
A.2.2 Approche retenue 216
A.2.3 Contribution : Convolutional SCCS 218
A.2.4 Applications 221
A.2.5 Discussion 224
A.3 Apprentissage de représentations en santé 226
A.3.1 Apprentissage profond en santé 228
A.3.2 Stratégies de préentraînement. 229
A.3.3 Contribution : comparaison de modèles d'attention et de méthodes de préentraînement. 231
A.3.4 Expériences 237

List of Figures **241**

List of Tables **245**

Bibliography **247**

List of acronyms **265**

Index **269**

INTRODUCTION

In the past twenty years, health insurance information systems and state agencies gathered data on citizens' healthcare consumption. This data is very rich and bears the promise of improving citizens' health, the healthcare system, and public health decision making. The work presented in this manuscript focuses on modeling health trajectories of French patients using claims data available in *Système National des Données de Santé* (SNDS, formerly known as *Système National d'Information Inter-Régimes* – SNIIR-AM). Access to SNDS resulted from a research partnership between *Ecole Polytechnique* and *Caisse Nationale de l'Assurance Maladie* (CNAM), the French agency managing the national health insurance system.

In that respect, the first contribution consisted in developing innovative ways of processing this large database (see Chapter I) as data volume and complexity initially hindered methodological research. The second contribution focused on improving longitudinal risk estimation of rare events (Chapter II). The resulting methodology successfully detected both long and short-term adverse drug reactions (ADRs) in applications detailed in Chapter II and III. Finally, extensive experiments were conducted to assess if pre-trained representations of medical event sequences could ease model estimation on multiple tasks (Chapter IV). This introduction gives a quick overview of these works, which are thoroughly developed in the following chapters.

1 Use of large observational databases for research

When statistical studies are performed to provide health safety information, randomized control trials (RCTs) are considered the "gold standard." RCTs consist of recruiting subjects with specific characteristics and dividing them into two similar groups. Subjects in one of the two groups receive treatment, while the others get a placebo. Subjects are then observed over a given period to assess the effectiveness and safety of the treatment. Thanks to tight control over the recruitment condition, group stratification, and randomization, RCTs provide a way to estimate unbiased treatment effects [Gro+04] and perform causal inference. However, they are costly to conduct and are unfeasible in some cases due to ethical concerns [Bee66; HS+79]. For example, an RCT involving a drug presumed harmful would put the treated group in danger willfully, which is unethical. Moreover, RCTs might suffer from small

sample size and short study periods as they are very costly to conduct. When it comes to rare or long-term outcomes, RCTs might fail to detect adverse drug reactions such as the association between pioglitazone and bladder cancer [Azo+12; Neu+12].

Observational studies can circumvent some of these issues. Indeed, observational studies do not control the subjects' assignment to treated and non-treated groups [Ros+10], which might be more ethical than RCTs in some cases. Furthermore, they can be conducted by re-purposing administrative data gathered by healthcare actors such as hospitals or insurance providers. The resulting datasets are often larger, longer, and much cheaper to acquire than RCTs data, with millions of patients followed over several years [Mad+14]. These databases' size allows the observation of sporadic events and specific sub-populations challenging to reach when using RCTs. Besides, observational data also provides a picture of real-life healthcare consumption, which might differ significantly from the tight design of RCTs [HA13]. As such, this alternative perspective might provide valuable insights for policymakers and practitioners. While observational studies have a long history, the use of large observational databases rose during the last fifteen years [Mad+14] thanks to their increased availability and advances in computational processing power.

1.1 Characterization of large observational databases in healthcare

Data acquired by re-purposing administrative data comes in two flavors: electronic health records (EHRs) and claims data. EHRs are produced by care providers to support and evaluate clinical care and the associated billing. They contain demographic information (such as birth date, gender, location, height, and weight) and sometimes observations regarding patients' living habits (e.g. smoker status or alcohol consumption) and medical history. More importantly, EHRs record information regarding the care provided, such as medical acts, diagnoses, vital signs, laboratory analysis results, imaging reports, and associated clinical observations or results. Because of its production process, EHRs data is often scattered across multiple care providers and might be hard to link with other records. Electronic health records can thus be seen as a very detailed and narrow perspective of a subject's interaction with healthcare services. Such data is available in the freely accessible MIMIC-III database [Joh+16], featured in numerous publications. Some EHRs can contain unstructured data such as medical imaging results or electrocardiograms.

In contrast, claims databases result from the aggregation of data primarily used for reimbursement purposes. Indeed, to be paid by insurance companies or agencies, healthcare providers must submit information supporting their services. Such data might consist of timestamped pharmacy claims of prescription drugs (e.g. drugs dispensed, quantity, manufacturer), acts, or exams performed on inpatients or outpa-

tients[2] and, eventually, the associated diagnoses. This information is far less detailed than the corresponding electronic health record. For example, a claim database might record that a subject underwent a physical exam at a specific date, but it will not record the associated exam results. Besides, it will only record reimbursed care and will not provide data regarding over-the-counter drugs, for example.

Such databases are maintained either by private companies (e.g. MarketScan Commercial Claims and Encounters [ACH08] provided by IBM) or state agencies (e.g. *Système National des Données de Santé* – SNDS [Tup+17a] provided by CNAM). Depending on the country, claims databases based on private insurance data might not reflect the general population because of socioeconomic biases and significant subject turnover as they change their subscription [Bro+10].

The French national system of health data (SNDS). A significant part of this thesis focused on developing tools and algorithms leveraging claims data from SNDS. French national health insurance consists of several insurance schemes depending on the beneficiaries' occupation. Local agencies handle their beneficiaries' reimbursements under the supervision of the CNAM, the national agency. Established in 2016, SNDS is an extension of SNIIR-AM, gradually developed since 1999. It aggregates data from multiple authorities such as hospitals and local agencies. SNDS gathers reimbursements data from most state health insurance schemes and their beneficiaries' demographic information. At its creation, it contained health reimbursements of 66 million inhabitants, which represents 98.8% of the French population [Tup+17a]. SNIIR-AM was initially used to monitor health expenditures and to evaluate health care utilization across the country. This database began to be used to conduct epidemiological studies in 2006, thanks to individual data availability. Since then, SNDS has led to many publications, some of which resulting in health policy changes [Neu+12; Tup+17a].

SNDS provides three years of history and twenty additional years of archived data submitted to an authorization from the national data protection authority (*Commission Nationale de l'informatique et des libertés* – CNIL). The quality of SNDS data results from mandatory logging of reimbursed care, three data validation stages, and pseudonymization routines. Thanks to its history length, high population coverage, and quality, the SNDS can be used to conduct epidemiological studies with high statistical power and almost exempt of representativity biases [Tup+17a]. Data contained in SNDS comes from two sources: *Données de Consommation Inter-Régimes* (Inter-scheme consumption data – DCIR) gathers outpatients health care billing and reimbursement information, while *Programme de Médicalisation des Systèmes d'Information* (medical information system program – PMSI) contains private and

[2]Inpatients are hospitalized persons. In opposite, outpatients are persons whose medical care does not require an overnight hospital stay.

public hospital data.

DCIR contains demographic information of the beneficiaries (date of birth, gender, date of death, the town of residence, and variables indicating if patients are beneficiaries of specific social subsidies depending on specific conditions or economic rules), and information on their eventual disabilities or long-term diseases. It also provides timestamped reimbursement information concerning drug purchases (coded with the Anatomical Therapeutic Chemical – ATC – classification system), medical procedures (coded with *Classification Commune des Actes médicaux* – French medical procedures classification – CCAM), laboratory analyses (coded with the classification of clinical pathology procedures, *nomenclature des actes de biologie médicale* – NABM) and medical products (*Liste des produits et prestations* – list of product and services – LPP).

PMSI is divided into four databases, *Médecine, Chirurgie, Obstétrique et Odontologie* (acute care ward – MCO), *Soins de Suite et Réadaptation* (rehabilitation care – SSR), *Hospitalisation À Domicile* (home to home care – HAD) and psychiatric care (PSY). In this thesis, we focus on the most stable and complete of these databases, that is PMSI-MCO. These databases contain pseudonymized hospital stays summaries, i.e., the starting and ending dates of the stays, and diagnoses (International Classification of Diseases, 10th revision – ICD-10), procedures, exams, and specific expenses. Contrary to EHR data, there is no information regarding the order or the temporality of the events happening throughout the hospital stay. PMSI also contains similar, more precisely timestamped information for outpatient consultations.

SNDS data access requires authorization from the CNIL, the French data protection authority. CNIL assesses the legal compliance and public interest of the projects applying for SNDS access. Five security rules protect SNDS data: (i) accessed data is pseudonymized, (ii) people accessing the data are strongly authenticated, (iii) their operations on the data are logged, (iv) audited periodically to check their compliance with security guidelines (v) taught to the data users.

To enforce these rules, users access SNDS data stored on CNAM Exadata [Ora08] servers through secured computers running SAS Enterprise Guide software [Sup76]. This setup has been adequate for the current SNDS uses, such as conducting epidemiological or economic studies using classical methodologies. However, it can hinder methodological research as it relies on closed-source, hard-to-customize software. Thanks to the research partnership between CNAM and Ecole Polytechnique, this thesis benefited from privileged access to an extract of SNDS data. An offline research cluster located in the CNAM datacenter hosted the data and ran the model estimations. Access and use of this cluster were subject to the security constraint described above.

At the time of writing of this thesis, a new governmental agency called Health Data Hub [Cug+18] is being created to ease methodological research on such data

by centralizing health data and providing accesses with similar freedom in terms of hardware and software infrastructure, under the CNIL's supervision.

As SNDS contains healthcare reimbursement logs for 66 million French patients, representing 20 billion raw events each year, data manipulation is challenging. A part of these events represents purely administrative cash flows, which do not bring additional information regarding beneficiaries' health and needs to be filtered out. The remaining events correspond to reimbursements that must be joined with tables giving medical details, such as the molecules composing the reimbursed drugs or details regarding diagnoses. Figure 1 gives a simplified overview of the SNDS structure, illustrating the necessity of join operations. Finally, once only relevant events are identified and then detailed by this joining process, they need to be combined to identify events that are meaningful for statistical or epidemiological analysis (see discussion on *phenotyping* Section 2.1 below).

1.2 Barriers to methodological research

Healthcare administrative data is not collected for research purposes. Codes used to justify reimbursements are not very good ontologies for representing practitioners' observations and actions on the patients [Alb+18]. Hence, even in the absence of coding errors, using raw codes to qualify the patients' health status is likely to provide mediocre estimates. Instead, raw codes should be translated to clinical concepts. This operation is called *phenotyping* and might be done manually or using machine learning.

Manual phenotyping is time-consuming to design and maintain. Indeed, it requires clinical experts, database experts, and data engineers to define and evaluate complex queries. These experts might also introduce their own bias by doing so. For example, identifying a bladder cancer in SNDS might be done crudely by keeping events with the ICD-10 code corresponding to bladder cancer ("C67"). Alternatively, a definition involving combinations of ICD-10 codes and surgical acts occurring in a well-defined time window leads to a more precise identification of bladder cancer events [Neu+12]. Misspecification in event identification algorithms can result in undetected events or false detections. Subsequent analyses are then likely to result in spurious conclusions.

Unfortunately, automatic phenotyping is still in its early stages and is not ready to be used in actual studies [Ban+18]. A middle-ground option can reduce event identification costs by using an existing mapping between codes and phenotypes. Large mapping databases such as PheCodes [Den+13] are publicly available and maintained by a college of experts. However, phenotyping remains one of the main bottlenecks of large-scale mining of large observational databases [HA13].

Most of the research effort trying to solve this issue rely on international or

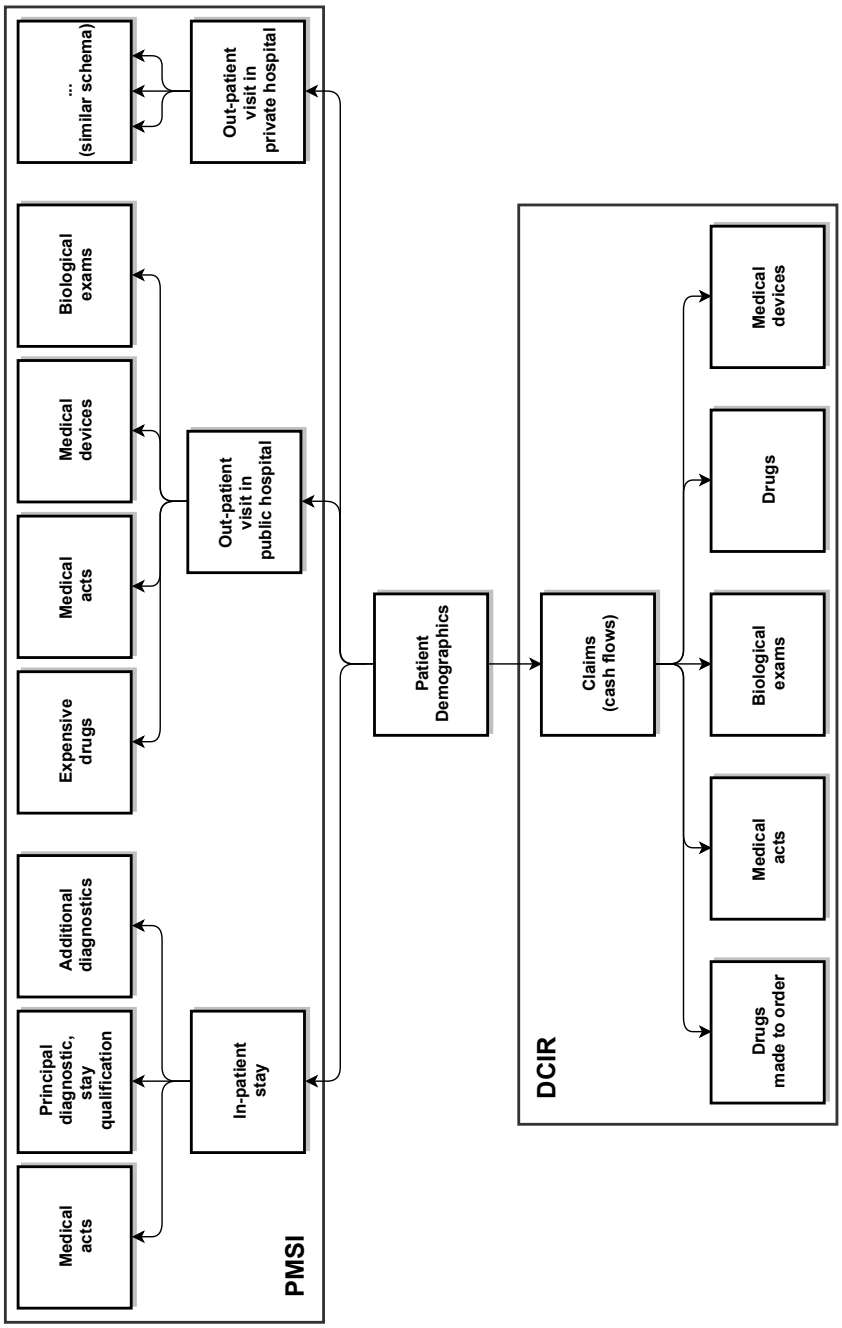


Figure 1 – Simplified structure of the SNDS database. This figure represents the main tables of the DCIR and PMSI MCO databases used in this thesis. Each rectangle represents a table, while arrows represent tables that can be joined together. This figure does not represent tables and sub-databases unnecessary to the applications presented in this work.

American vocabularies. Hence, SNDS cannot benefit from most of these works as it relies on coding vocabularies specific to France. Developing an efficient and reliable method to extract patients' care pathways from SNDS is thus the first and foremost challenge to solve. This extraction process is very complicated because of the database structure and size, which might hinder the reproducibility and reuse of research results.

Extracting meaningful patients care pathways involves two tasks. First, all the data corresponding to a set of patients need to be identified and collected. When the data is not normalized around the patients, this task requires several join operations, which can be very computationally intensive as the data volume increases. Second, medical concepts have to be correctly identified from administrative codes: this *phenotyping* task relies heavily on a combination of medical and database knowledge. The algorithms used to perform concept extraction from administrative data are either disclosed through scientific publications or shared as lengthy SQL queries of varying quality [Loo19]. As a result, building a study from scratch might be faster than reusing poorly documented code from previous works [Loo19; PDZ06]. Besides, accessing SNDS relies on proprietary software such as SAS [Sup76] or SPSS [IBM68]. While these tools are suitable to produce public health studies, they hinder methodological research as they do not interact readily with R or Python packages implementing state-of-the-art machine learning algorithms.

Several research programs produced tools to alleviate some of these issues. An extensive research effort aims at promoting data integration and interoperability by producing standard data models and terminologies to be shared across institutions (Observational Medical Outcomes Partnership Common Data Model (OMOP CDM) supported by the Observational Health Data Sciences and Informatics (OHDSI) research program [Hri+15], and the Informatics for Integrating Biology & the Bedside (i2b2) data model [Mur+10]).

Both models are normalized data models[3] centered around the patients, thus reducing the number of join operations required to access a specific patient history. However, the process of transforming an existing database to comply with such standards is costly, as it requires to build complex mappings. Establishing such a mapping for the SNDS database is still a work in progress [Dou+20].

In other fields, web-scale analytics have shifted from normalized SQL databases towards NoSQL technologies relying on distributed computing, denormalization, and columnar storage. Distributed computing produced gains in computational power by using low cost, commodity servers instead of expensive dedicated hardware [Bon+17]. To our knowledge, there is no implementation of a similar approach to extract medical concepts from large healthcare databases.

[3]These data models were specifically designed to operate within SQL databases.

1.3 Contribution: a framework for reproducible and fast data processing

SCALPEL3, an open-source framework, was developed during this thesis to answer reproducibility and scalability challenges posed by LOD concept extraction. This framework adopts an approach combining denormalization and distributed computing to large health databases. SCALPEL3 is divided into three components, as illustrated in Figure 2.

SCALPEL3 uses on Apache Spark [Zah+16], a robust and widely adopted distributed in-memory computation framework. Spark provides a powerful SQL-like high-level API and a more granular API to perform data operations. Spark can be used in combination with the Hadoop File System (HDFS) [Shv+10], which splits large files into small chunks, distributed and replicated[4] over a computing cluster. Large-file reading operations are then performed in a distributed manner, improving their speed and robustness to failures.

(i) SCALPEL-Flattening. As mentioned earlier, performing data analysis on SNDS requires many joins and can consequently be extremely slow. The data are denormalized to circumvent this issue. Sequential joins of the tables produce a big table in which each line corresponds to a patient identifier and a complete representation of an event.

Denormalizing a star-schema database results in a massive table due to values replications. The denormalized data is stored in Parquet [Voh16] files to avoid storage and computational issues. Well-integrated in the Spark ecosystem, Apache Parquet is an open-source columnar storage format implementing Google's Dremel [Mel+10] data model. Spark directly benefits from columnar storage data compression and query optimization [Arm+15]. A set of monitoring statistics are computed throughout the denormalization process to avoid data loss and complex debugging. SCALPEL-flattening can be used as a library through its Scala [Ode+04] application programming interface (API), or as a packaged application through and text file configurations. End-users looking for optimal reproducibility are encouraged to use the packaged version with versioned[5] text file configurations.

(ii) SCALPEL-Extraction provides fast extractions of medical concepts from the denormalized tables produced by SCALPEL-Flattening. By providing ready-to-use medical events, SCALPEL-extraction encapsulates SNDS technical knowledge. Nonetheless, it keeps medical data as raw as possible so that end-users have access to fine-grained data, which is critical when designing observational studies [Hon+18;

[4] By default, three replicas of each data chunk are spread across the cluster.

[5] Versionning can be performed with version control systems such as `git`, for example.

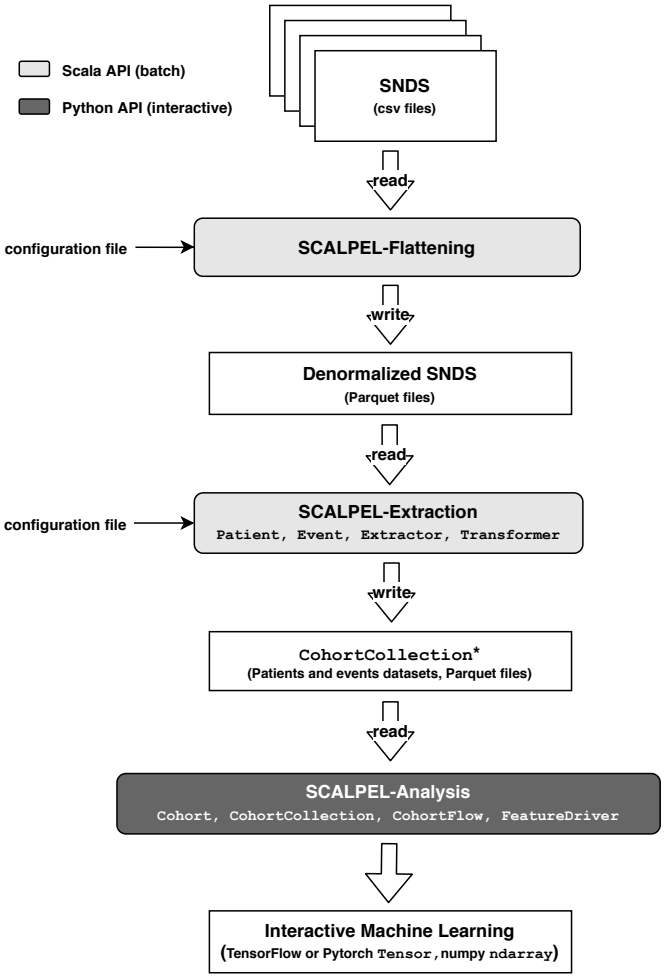


Figure 2 – SCALPEL3 workflow. SCALPEL3 consists of three independent open-source libraries plugged one after another. SCALPEL-Flattening, implemented in Scala/Spark, denormalizes the input database exported as CSV or Parquet files into a single big flat database. Then, SCALPEL-Extraction, implemented in Scala/Spark, extracts concepts from this flat database. Finally, SCALPEL-Analysis, implemented in Python/PySpark, loads extracted concepts to perform in-memory interactive analysis and feed machine learning algorithms.

Wan+16]. The extracted concepts are organized around two abstractions: `Patient` and `Event`. The `Patient` abstraction has a unique `patientID`, a `gender`, a `birthDate` and eventually a `deathDate`. The `Event` abstraction allows to represent any event associated to a patient. It can be punctual (e.g. medical act) or continuous (e.g. hospitalization).

All concepts are automatically extracted into `Patient` or `Event` objects by a set of `Extractors` and `Transformers`, designed to fetch the data in the relevant tables

and columns of the SNDS `Sources`.

As illustrated in Figure 3, `Extractors` successively refines data from the input (wide denormalized tables) by (1) identifying the relevant columns, (2) filtering out null values according to some columns, and (3) conform the extracted data to a standardized schema as illustrated in Figure 3. These three operations are very fast when performed on columnar data, as they exploit data sparsity[6] and consist of simple look-ups over hash tables containing columns metadata. An optional step filtering rows by value can occur before step (3). This operation is slower as it manipulates row values, but it typically occurs on small data since it happens near the end of the extraction process.

As SCALPEL-flattening, SCALPEL-Analysis provides a Scala API and a packaged mode using text configuration files.

(iii) SCALPEL-Analysis is implemented in Python/PySpark [Zah+16] since it is designed for interactive environments, such as Jupyter notebooks [Klu+16]. This module is based on the `Cohort` abstraction, defined as a set of `Patients` and their associated `Events` in a [`startDate`, `endDate`] time-window. Basic operations such as union, intersection, and difference can be performed between `Cohorts`, while a human-readable description is automatically added to the results. More granular control is kept available through accesses to the underlying Spark DataFrames (using Spark DataFrame API). This combination allows easy data engineering and fine-grained yet reproducible experiments.

International guidelines [Ben+15] regarding studies based on LODs insist on the explanation of cohort construction to highlight eventual population biases, motivating the `CohortFlow` abstraction. A `CohortFlowCohortFlow` is an ordered iterator defined as the following left fold operation

$$\text{foldl}(c : \texttt{CohortCollection}, \cap) := (((c_0 \cap c_1) \cap c_2) \cap \dots c_n)$$

assuming an input `CohortCollection` c of length n, where $\cap$ denotes an intersection of the `Cohorts`' patients. The `CohortFlow` iterator was designed to track transformation stages leading to a final `Cohort`. To ease this tracking, each intermediate `Cohort` is stored with a description of filtering rules used to produce the next `Cohort`.

Finally, the `scalpel.stats` sub-module produces descriptive statistics on a `Cohort` and their associated plots. For now, it contains more than 25 `Patient`-centric or `Event`-centric statistics, adding a custom one being very easy. When combined with `CohortFlow`, `scalpel.stats` computes various statistics at each analysis stage, helping to dectect biases induced by successive population filtering operations (see example in Appendix I.A of Chapter I).

[6] Null values are not represented in the data.

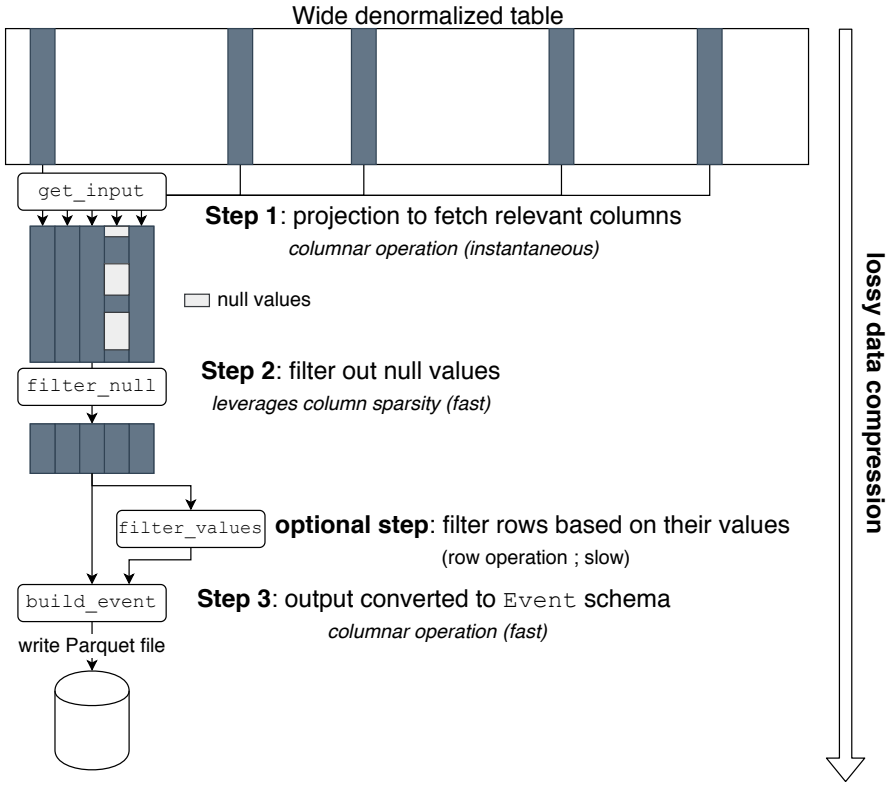


Figure 3 – `Extractor` design. `Extractors` implemented in SCALPEL-Extraction successively refines the input table (a large denormalized table) by taking advantage of fast columnar operations to produce ready-to-use medical events. Step 1 selects the relevant columns (equivalent to a hash table look-up) while Step 2 removes rows where null values are detected in specific columns, taking advantage of the sparsity of columnar representation (null values are not encoded in the data). Optionally, this extraction process filters out rows based on their values. Finally, Step 3 conforms the data to the `Event` schema, and is written to a Parquet file.

Knowledge reuse. SCALPEL-Flattening and SCALPEL-Extraction implement algorithms encapsulating SNDS expert knowledge. They implement flexible rules to assess data quality and extract predefined concepts that can be reused across studies. While we cannot guarantee that this growing library of concepts meets all the use-cases, it provides a good starting point when beginning a study. Even if SCALPEL does not automate the whole process, we believe it can dramatically accelerate research.

Reproducibility. Abstractions provided by SCALPEL3 can be used to study SNDS data by mostly using high-level operations, resulting in a smaller and more readable study-specific code. As a result, this code is easier to maintain and debug, while high-level operations provided by SCALPEL3 are tested and versioned using commonly used continuous integration tools. Besides, the use of text configuration files[7] allows for the reproducibility of flattening and extraction jobs while automated statistics reports monitoring operations performed on data. These tools greatly facilitate the reproducibility, maintainability, and the audit of conducted studies.

Scalability. SCALPEL3 was successfully used to perform the extraction of complex concepts for studies featured in Chapters II and III of this manuscript, featuring up to 14.5 million patients observed over three years (corresponding to more than 15 billion healthcare events and roughly 15 terabytes of data) in less than 49 minutes on a small 15 nodes HDFS cluster. Besides, SCALPEL3 scales almost linearly (provided the workers' resources are not shared) with the number of executors as illustrated if Figure 4 (see Chapter I for more details).

Adoption. SCALPEL3 reduces entry barriers to medical observational studies by providing ready-to-use concepts while easing data manipulation thanks to abstractions allowing concept extraction, high-level cohort manipulation, and production of data compatible with machine learning libraries formats. SCALPEL3 makes studies based on SNDS much simpler and more scalable than the existing framework [Tup+17a]. It is now used at the agency collecting SNDS data, at the French Ministry of Health and soon at the National Health Data Hub in France [Cug+18]. We believe that its use will continue to grow and that its genericity can address other LODs.

[7]Written in Human-Optimized Config Object Notation (HOCON) [Typ16]

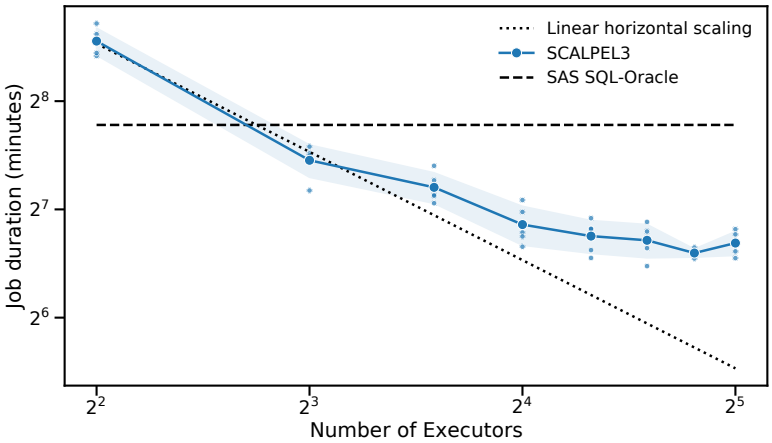


Figure 4 – SCALPEL-Extraction scaling experiments. The solid blue line represents the mean total running time (in seconds) of the benchmark extraction job (described in Chapter I) when varying the number of worker nodes used to perform the computation. Five small blue dots represent individual run time measurements for each experiment, while the big blue dot represents the corresponding means. Light blue bands represent one standard deviation computed over five runs. The dotted line corresponds to a theoretical performance assuming a perfect horizontal linear scaling (based on the single node performance). Dashed lines represent the runtime of similar queries on the SNDS SAS-Oracle infrastructure using a single run. Multiple runs were not performed on SAS-Oracle as computing resources allocation is dynamic and might differ over several runs. The scaling gains then slow down around 28 executors as at this point, the cluster resource used by the storage services (HDFS) comes into conflict with computation (SCALPEL3). Note that SCALPEL3 has been recently updated and should be slightly more performant.

2 Adverse drug reactions detection

Improving Adverse Drug Reaction (ADR) detection is one of the promises carried by the increased volume and availability of observational data [Sta+10]. ADRs can be defined as a harmful event following a single dose or prolonged use of a drug. An ADR can be related to the dose or not. Dose effects can be caused by supra-therapeutic doses (toxic effect resulting from an excessive dosage), sub-therapeutic doses (hyper-susceptibility), or at standard therapeutic doses (collateral effects, such an effect occurring in a non-targeted tissue) [AF03]. Dose relationships are tough to assess when using data from SNDS, as prescriptions are not known, and drug packaging is standardized [Tup+17a]. Hence, this thesis focuses on the temporality of ADRs. Indeed, while some ADRs can be time-independent (e.g. digoxin toxicity caused by potassium depletion [AF03]), many of them might happen either at the first dose (e.g. anaphylaxis after first penicillin use) or with some delay of varying length. Delayed effects might occur at the time of drug withdrawal (e.g. opiates) later on (e.g. carcinogenesis [AF03]). Individual susceptibility might heavily affect the occurrence and timing of ADRs [AF03]. Figure 5 represents examples of ADR risk patterns.

Historically, post-marketing surveillance relies on spontaneous reports from physicians and consumers [Sch+16], thus depending on human detection of adverse effects. Reports then trigger statistical confirmation studies, eventually using claims data such as [Neu+12]. Unfortunately, relying on human detection has been shown to result in ADRs under-reporting [Alv+98]. Indeed, when ADR events are rare, joining the dots might be very hard for a human observer. Data mining LODs might complement human detection by screening a vast amount of drug and reaction combinations to improve ADR detection.

2.1 Modeling challenges

Several methodological challenges specific to LODs such as SNDS complicate this task. Indeed, healthcare data is the result of three intertwined processes [Alb+18; Hag+14]:

(i) An epidemiological process, reflecting the physiology and pathophysiology of the observed patients.

(ii) A behavioral process related to the patients' lifestyles and healthcare utilization habits.

(iii) An institutional process related to the structure and the operation of the health-care system.

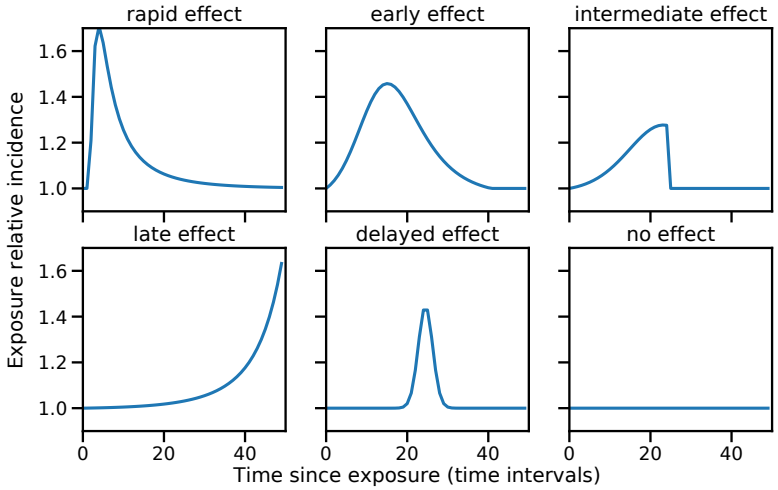


Figure 5 – Examples of risk patterns associated with adverse drug reactions. The probability of experiencing an event in a time interval is proportional to the corresponding area under the risk curve. Curves represent the relative risk of an event evolution after drug exposure start. Rapid effects occur at the first dose. Early reactions occur early in the treatment before decreasing due to the development of drug tolerance. An intermediate effect may or may not happen after some delay. If they do not occur after a fixed period, they will never occur. Late effect risk slowly increases over time. Delayed effect risk suddenly increases after some delay, e.g. at the time of drug withdrawal.

As a result, studies using this data should consider the peculiarities described in the next paragraphs.

Missing information and coding errors. While SNDS is very rich, it does not contain information that might be critical, depending on the conducted studies. Typical examples are socioeconomic characteristics (income, marital status), lifestyle habits (smoking status, alcohol consumption, nutrition), examination results, test results, over-the-counter drugs, drugs delivered during hospital stays, and prescriptions and accurate drugs dosage. SNDS records the cause of death since 2018. The absence of such information might cause biases depending on the statistical modeling strategies.

Besides, data can be inaccurate. While some errors can be random, the healthcare billing systems might lead to systematic errors. For example, French hospitals are paid based on a flat price corresponding to a stay's "main diagnosis." Therefore, they are incited to code health events in a specific way to optimize both their revenue and practitioners' time. The resulting recordings may thus conflict with the nominal

definition of a concept [HA13]. These errors can be correlated to individual healthcare providers, depending on their coding policy and working habits. In the case of SNDS, this issue might influence hospital stay data. However, as outpatient care is automatically recorded, it should be less affected by such coding biases [Tup+17a].

Pathways. Specific pathways might influence the results of studies based on observational data. When the studied molecules are often prescribed in a given sequential order, it is hard to separate their individual influence on an event of interest [Hri+16].

Reverse dynamics. Healthcare data capture beneficiaries' interactions with the healthcare system rather than a direct recording of their physiology, resulting in feedback loops and reversed dynamics [HA13]. Indeed, diseases precede their symptoms in terms of physiology. The data may record the symptoms (through exams or medical acts, for example) before the actual identification of the disease [HAP11].

Not-at-random sampling. The beneficiaries' events recording occurs when they interact with the healthcare system, i.e., data is only sampled when beneficiaries have health issues. As such, data sampling should not be considered random. In response, some studies impute data [Piv+14], use the information missingness as a feature[8] [Hag+14] or use flexible models, which is the approach developed in this thesis.

These issues might result in biases, the most pervasive one being the *indication bias* when it comes to observational studies. This bias occurs when an indication (e.g. fever) both prompts an exposure (e.g. paracetamol) and causes outcomes (e.g. asthma) [Aro+18]. Following this example, a study ignoring the fact that some viral infections causing fever increase the risk of developping asthma would wrongly associate asthma with paracetamol. Such biases are hard to avoid, especially when using SNDS as drug prescriptions are not recorded. For now, the only solution is to tailor the studies to address each database peculiarities [Mad+14]. The approach developed in this thesis relies on careful phenotyping and study designs, flexible model, and cautious interpretation to derive useful insights on many aspects of the healthcare system and its beneficiaries. However, causal inference is hindered by many unobserved confounding variables and the impossibility of taking actions on healthcare policies for research purposes.

[8] For example, the patient rate of visits can be a proxy for patient adherence and access to care

2.2 Mathematical tools

This section quickly introduces several mathematical tools used in this thesis. The works presented here aim at estimating temporal associations of diverse healthcare events. Statistical learning methods were used to establish general patterns structuring the data under study. The ADR detection problem was formulated as a supervised learning problem, aiming to predict a specific longitudinal event based on other longitudinal health events. The modeling of temporal dynamics borrows concepts from point processes theory and survival analysis. Estimating the resulting models' parameters relies on sparsity inducing penalties, proximal operators, and stochastic optimization.

Supervised learning. Given a training sample of *annotated* examples

$$\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\},$$

where $x_i \in \mathcal{X} \subset \mathbb{R}^d$ are d-dimensional input features and $y_i \in \mathcal{Y} \subset \mathbb{R}$ are values to be predicted, supervised learning consist in learning a prediction function $h : \mathcal{X} \to \mathcal{Y}$.

A *goodness-of-fit* function f is used to measure how well a statistical model fits a dataset $\mathcal{D}_n$ given the model parameters θ. For example, the negative log-likelihood function of a model, the quadratic loss, or the cross-entropy loss might serve as goodness-of-fit functions depending on the supervised task. In this thesis, a data sample corresponds to the history of patient i. Data samples are assumed to be generated independently. As a result, the goodness-of-fit functions are commonly decomposed as an average of individual losses f_i computed over each sample

$$f(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta),$$

where f_i implicitly depends on the data sample. Fitting the model to a dataset consist in solving the minimization problem

$$\min_{\theta \in \mathbb{R}^d} f(\theta).$$

Penalization and proximal operators. Statistical models considered in this thesis require the estimation of parameters $\theta \in \mathbb{R}^d$ given a *goodness-of-fit* function $f(\theta) : \mathbb{R}^d \to \mathbb{R}$. When the number of parameters d is large, there is a risk of model *overfitting*. In this case, the model is too closely adjusted to the training dataset and does not generalize well to other datasets. To prevent overfitting, the parameters' space can be constrained using a sparsity-inducing norm $g : \mathbb{R}^d \to \mathbb{R}$ leading to an optimization problem of the form

$$\min_{\theta \in \mathbb{R}^p} f(\theta) + \lambda g(\theta), \tag{1}$$

where $\lambda > 0$ is the penalization strength. A cross-validation procedure selects the best performing penalization strength λ according to some performance metric. The function f is assumed to be differentiable and $L-$smooth, i.e.

$$\|\nabla f(u) - \nabla f(v)\| \leq L\|u - v\|,$$

for any $u, v \in \mathbb{R}^d$ where $\|.\|$ is the Euclidean norm on $\mathbb{R}^d$ and $L > 0$ is the *Lipschitz constant*. The penalty g can be non-differentiable but it is assumed to be prox-capable, in the sense that its *proximal operator* (see definition below) can be easily computed. A range of methods can be used to solve Equation (1). Techniques known as *proximal methods* are particularly adapted to such problems thanks to their good convergence rates and scalability [Bac+12].

The *proximal operator* associated to λg is defined as

$$\mathbf{prox}_{\lambda g}(v) := \arg\min_{u \in \mathbb{R}^d} \left(g(u) + \frac{1}{2\lambda}\|u - v\|_2^2 \right), \tag{2}$$

for any $v \in \mathbb{R}^d$. The proximal operator is well-defined and unique since the objective in Equation (2) is strongly convex. If $\mathcal{C}$ is a closed nonempty convex set, it is interesting to remark that if g is the indicator function $I_{\mathcal{C}}(x)$ equal to 0 when $x \in \mathcal{C}$ and $+\infty$ otherwise, then the proximal operator associated to g consist in an orthogonal projection onto $\mathcal{C}$. The proximal operator can then be seen as a generalization of projection. Indeed, the point $\mathbf{prox}_{\lambda g}(v)$ is a compromise between minimizing λg and being near to v, weighted by the parameter λ. When combined with appropriate optimization algorithms, proximal operators can solve problems such as (1) in a reasonable amount of time [Bac+12], especially when g is non-differentiable. Proximal operators of penalties used in this thesis are quickly introduced below.

The *Lasso* penalty, also known as the ℓ_1-norm penalty, is used to induce sparsity by setting a number of coefficients v_j, $j = 1, \dots, d$ exactly equal to zero depending on the penalty strength λ. Denoting $(x)_+ = \max(x, 0)$, the proximal operator associated to $g_{\ell_1} = \|\cdot\|_1$ can be computed as follows [Bac+12]:

$$[\mathbf{prox}_{\lambda g_{\ell_1}}(v)]_j = \left(1 - \frac{\lambda}{|v_j|}\right)_+ v_j.$$

The ℓ_1/ℓ_2-norm penalty (or *group-Lasso*) [TVW05; YL06] is used to induce sparsity over groups of coefficients $j \in \mathcal{J}$, where $\mathcal{J}$ is a partition of $\{1, \dots, d\}$. To take the group size into account, λ is usually normalized by $\sqrt{\mathrm{Card}(j)}$. As the coefficient groups considered in this work share the same size, we ignore this normalization to simplify the notations. Depending on the penalty strength λ, all the coefficients belonging to some group $j^\star$ will be set exactly to zero. The penalty writes as follows

$$g_{\ell_1/\ell_2}(v) = \sum_{j \in \mathcal{J}} \|v_j\|_2,$$

and its proximal operator can be computed using the closed form [Bac+12]

$$[\mathbf{prox}_{\lambda g_{\ell_1/\ell_2}}(v)]_j = \left(1 - \frac{\lambda}{\|v_j\|_2}\right)_+ v_j,\ j \in \mathcal{J}.$$

The *Total Variation* (TV) penalty was first introduced in the image processing community [ROF92] to perform denoising by encouraging piecewise constant signals. When working on a single dimensional signals, it writes

$$g_{TV-1D}(v) = \sum_{i=1}^{p-1} |v_{i+1} - v_i|.$$

Although there is no closed-form expression to compute the proximal operator associated to this penalty, it can be computed efficiently (in $O(d)$) and exactly using [Con13].

Computing the proximal operator associated to a combination of several penalties is not straightforward. However, group-Lasso and Total Variation can be combined as follows. Considering integer intervals $j' \in \mathcal{J}' \subset [\![1, d]\!]$, and

$$[\mathbf{prox}_{\lambda_1 g_{TV-1D}}(v)]_{j'} = \sum_{i, i+1 \in j'} |v_{i+1} - v_i|,$$

the TV-Lasso proximal operator associated to

$$[g_{TVL}]_{j'} := v \to \lambda_1 [g_{TV-1D}(v)]_{j'} + \lambda_2 \left[g_{\ell_1/\ell_2}(v)\right]_{j'}$$

can be computed as

$$[\mathbf{prox}_{\lambda g_{TVL}}(v)]_{j'} = [\mathbf{prox}_{\lambda_2 g_{\ell_1/\ell_2}}([\mathbf{prox}_{\lambda_1 g_{TV-1D}}(v)]_{j'})]_{j'}$$

where $\lambda = (\lambda_1, \lambda_2)$ as shown in [Zho+12].

Optimization algorithms. Training the supervised learning models introduced in this thesis can be expressed as solving the following optimization problem

$$\min_{\theta \in \mathbb{R}^d} F(\theta) \quad \text{with } F(\theta) = f(\theta) + \lambda g(\theta), \quad \text{where } f(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta). \tag{3}$$

This paragraph introduces the basic ideas behind optimization methods used to minimize Equation (3). Algorithms such as second-order methods (e.g. Newton's method) will not be addressed here are they are hard to use in practice with high-dimensional datasets.

Batch gradient descent algorithms can be used whenever F is differentiable. From an initial guess θ_0, it iteratively updates the parameters in the opposing direction of the gradient

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta F(\theta_t),$$

where $\eta > 0$ is the *step size*, controlling the size of the updates. The choice of the step size is critical. A small step size can lead to very slow convergence, while a very large step size might result in poor convergence with updates repeatedly overshooting the minimum. Several methods (e.g. line search) aim to tune automatically the step size [Ber99; Nes83]. Several algorithms (e.g. ISTA [BT09]) have been developed to solve problems such as Equation (3) through the use of proximal operators, leading to the following update

$$\theta_{t+1} = \mathbf{prox}_{\lambda g/L}\left(\theta_t - \frac{1}{L}\nabla_\theta f(\theta_t)\right),$$

whenever F is differentiable and $L-$smooth, and g is prox-capable convex function.

Note that these algorithms do not exploit the fact that $f(\theta)$ is an average of functions. Moreover, ISTA requires computing the objective gradient on the *whole* dataset in order to perform a *single* update of θ_t. In practice, it might result in a very slow algorithm when the dataset does not fit in memory despite fast convergence rates [Rud16].

When f is an average of functions f_i associated with each training sample, *stochastic gradient descent*[9] (SGD) [RM51] might be more efficient. This method approximates the full gradient with the random variable $\phi_t = \nabla f_I(\theta_t)$. If I is uniformly distributed over $\{1, \dots, n\}$, ϕ_t is an unbiased estimator of the full gradient, i.e. $E(\phi_t) = \nabla f(\theta_t)$ [RM51].

Uniform SGD exploits this idea by performing an update of $\theta_{t+1} = \theta_t - \eta \nabla f_i(\theta_t)$ per data sample i. The iterate is then updated n times for each pass (or *epoch*) over the dataset when batch gradient descent would perform a single update. However, $\nabla f_i(\theta_t)$ does not converge towards zero as θ_t approaches the minimum, hindering the convergence towards a precise solution. The algorithm then overshoots the minimum repeatedly if the step size η is constant. Using decreasing step sizes can mitigate this issue [Rud16], though it can result in slow converge speed. Alternatively, using *mini-batches* might stabilize the congergence, by using $k \ll n$ samples per update instead of one to estimate ϕ_t, i.e.

$$\phi_t^k = \frac{1}{k} \sum_{i \in \mathcal{K}, |\mathcal{K}| = k} \nabla f_i(\theta_t).$$

The update ϕ_t^k has typically lower variance than ϕ_t, and can be computed efficiently by using parallel computation [Rud16].

[9] Also known as Robbins-Monro algorithm.

Other algorithms, such as stochastic variance reduced gradient (SVRG) [JZ13] use variance reduction techniques to converge towards a more precise minimum. They reach the precision of batch methods while using quick iterations alike SGD. Similar to batch algorithms, several stochastic algorithms have been adapted to solve problem (3) using proximal operators, such as Prox-SVRG [XZ14].

Point processes. Point processes model random occurrences of points (medical events in our case). They can be used to describe spatio-temporal phenomena, such as earthquakes [Oga99] or infectious diseases [Mey+18; Rei+18]. Some useful definitions and results related to point processes are introduced below. Further details on point processes can be found in [DV03].

Let us consider a set of distinct event times $\xi = \{t_1, \dots, t_n\}$ occurring in an interval $[0, T]$, where n is an integer random variable. A point process can be represented by considering the associated counting process $N(a, b] = \sum_{\tau \in \xi} \mathbb{1}_{\tau \in (a,b]}$ representing the number of event times in the interval $(a, b] \subset [0, T]$. To ease the notations, let us write $N_t = N(0, t], t \leq T$. The distribution of a counting process is characterized by a conditional intensity function

$$\lambda(t|\mathcal{F}_t) = \lim_{dt \to 0} \frac{\mathbb{P}(N_{t+dt} - N_t = 1|\mathcal{F}_t)}{dt},$$

representing the infinitesimal probability of an event occurrence at time t, given the information available up to time t, denoted $\mathcal{F}_t$. The conditional intensity is a very useful tool for statistical inference, as (i) it can be used to simulate the process [Oga81] and (ii) it can be used to express the likelihood of the process in a closed form. Indeed, the associated negative log-likelihood writes

$$\int_0^T \lambda(s|\mathcal{F}_s)ds - \sum_{k=1}^{N_T} \log \lambda(t_k|\mathcal{F}_{t_k}),$$

see [DV03].

Poisson Process. The simplest point process is the homogeneous Poisson point process with intensity $\lambda > 0$. It is defined by three properties:

(i) The number of events in each finite interval $(a, b]$ has a Poisson distribution with intensity $\lambda(b - a)$,

(ii) the number of events in disjoint intervals are independent random variables,

(iii) $N(0) = 0$.

If the intensity depends on time, the Poisson process is *inhomogeneous*, in which case $N(a, b]$ has Poisson distribution with intensity $\int_a^b \lambda(t)dt$. Note that $\lambda(t)$ might depend on exogenous longitudinal variables, but not on previous event times [DV03].

Survival analysis. Survival analysis models the expected time until one or more events of interest happen (*time-to-event*, TTE). For example, such events might be hard drive failure times [Dat19], or bladder cancer diagnoses [Neu+12]. Let us denote the random time of such an event T, and define the *survival function* as

$$S(t) = \mathbb{P}(t < T).$$

For some samples, the event of interest might not occur during their observation. These samples are said to be *right-censored*. In epidemiology, censoring might result from patients leaving the study or deceasing before they had a chance to experience the event. Censoring information prevents the model from incorrectly considering that the samples who have not yet experienced will never do.

Let us denote the time of censoring C. In practice, we observe the time $T \wedge C$ and $\delta = \mathbb{1}_{T \leq C}$ indicating if a sample is censored or not. Let us consider a dataset of n *i.i.d* samples $\{(T_i, C_i), i = 1, \dots, n\}$ and associated covariates $x_i \in \mathbb{R}^d$. Assuming that censoring times C_i and event times T_i are independent, let us define $\delta_i = \mathbb{1}_{T_i \leq C_i}$, $Y_i(t) = \mathbb{1}_{T_i \wedge C_i < t}$ and the counting process $N_i = \delta_i(1 - Y_i(t))$ for each sample. The intensity associated to $N_i(t)$ then writes

$$\alpha_i(t) = \lim_{h \to 0} \frac{\mathbb{P}(t \leq T \leq t + h | t \leq T)}{h} = -\frac{S'(t)}{S(t)}.$$

When assuming N_i to generate at most one event time, the intensity can be reduced to $\alpha_i(t) = \lambda_i(t)Y_i(t)$. The function $\lambda_i(t)$ is called *hazard function*, and the *cumulative hazard function* is defined as

$$\Lambda_i(t) = \int_0^t \lambda(s)ds.$$

Survival analysis aims to estimate either $S(t)$, $\lambda(t)$ or $\Lambda(t)$ and eventually to analyse the influence of covariates on these functions. One can use parametric, semi-parametric or non-parametric approaches to estimate these functions [Cox72; Mil11].

The Cox model. The Cox model, introduced in [Cox72], is a semi-parametric approach of modelling the hazard function as follows:

$$\lambda_i(t) = \lambda_0(t) \exp\left(x_i^\top \theta\right),$$

where λ_0 is a baseline hazard independent of the covariates x_i. Instead of modelling λ_0 using a parametric form and to estimate all the parameters using the model's *full*

likelihood, the Cox model only estimates θ relying on the *partial likelihood* of the model, by considering the *hazard ratios* of two patients,

$$\frac{\lambda_i(t)}{\lambda_j(t)} = \exp\left(\left(x_i - x_j\right)^\top \theta\right).$$

The conditional likelihood then writes

$$L_P(\theta) = \prod_{i=1}^{n} \left(\frac{\exp\left(x_i^\top \theta\right)}{\sum_{j \in R_i} \exp\left(x_j^\top \theta\right)} \right)^{\delta_i},$$

where $R_i = \left\{j : Y_j \geq Y_i\right\}$ denotes the samples who have experienced the event or the censoring time after sample i. While this model is very popular thanks to its capacity to model only the effect of covariates on the hazard function without requiring the estimation of the baseline risk, this approach is not straightforward to scale to a very large dataset due to the necessity of ordering patients to compute R_i [Ach+15].

Rare events and Zero-inflation. In survival analysis, events of interest are failure times. As such, they are supposed to happen only once. When confronted with rare events, a survival process might be *approximated* by a Poisson process of low intensity. While a Poisson process might be easier to estimate than a Cox model on large datasets, its estimation might be affected by the presence of many censored samples in the dataset, i.e. patients who have not experienced the event before their censoring time. In this case, the dataset is *zero-skewed*, and the model estimation is problematic. Indeed, when samples without events are overrepresented, the model tends to estimate almost null intensities and predict an overall absence of events. In the case of Poisson regression, using zero-inflated losses is a way to circumvent this issue. Such losses use a mixture of random variables. First, a Bernoulli random variable controls if an event will occur or not. Then, a Poisson random variable models the event count given it is not null [Zuu+09]. While there exist longitudinal extensions of such models [BW17], they are not very fit to problems where events are infrequent and do not happen more than once in the observed datasets. Besides, most of these longitudinal formulations rely on Markov processes. However, when events occur more than once, the Markov property is not likely to be verified in healthcare, as past disease occurrence might increase the probability of relapse (e.g. myocardial infarction).

2.3 Common methodologies

Besides the mathematical model formulation and estimation, the design of ADR detection studies is crucial, even more, when using LOD data. The Observational

Medical Outcomes Partnership (OMOP) pioneered an alert system based on vast amounts of claims data [Rya+12; Rya+13b]. Their approach relied on designs and models usually employed in observational studies. These designs can be divided into four categories.

Multiple groups designs

Such designs compare subjects who experienced the adverse effect (cases) with subjects who did not (controls).

Cohort studies compare groups of patients selected *according to their exposure* to some risk factor. For example, the risk associated with a drug might be studied with a *new-users* cohort. In this case, patients newly exposed to the drug of interest (DoI) is juxtaposed with a comparator population. The comparator group might be patients exposed to a drug of a different pharmacologic class sharing the DoI's indication; or patients with a diagnosis for the DoI's indication.

The *Case-control* method compares two population groups *according to the occurrence of an adverse effect*. The patients who experienced an adverse effect (the *cases*) are compared with the patients who did not (the *controls*). When performed on administrative databases, such designs are always nested within a cohort.

Comparing the different groups can be made by estimating odds ratios using a logistic regression model predicting the target event from drug exposure. Odds ratios are said *unadjusted* when estimated with univariate logistic regression predicting the target event from drug exposure. Odds ratios are *adjusted* when estimated using multivariate logistic regression to control confounding variables. When using a Cox model (described Section 2.2), the survival time and the incidence rates are estimated and compared between the patient groups.

These approaches are very sensitive to residual systematic differences between the studied groups. Thus, their performance heavily depends on measuring confusion factors or ensuring that the compared groups share similar characteristics (e.g. demographics, life habits, or existing diseases).

Single group designs

Such designs, called *Self-controlled designs*, include only *cases* who experienced the studied adverse event. They compare subjects considered to be at risk of experiencing the event (*risk periods*) with themselves when they are not at risk (*control* or *risk-free* periods). As each included patient is known to have experienced the studied event, statistical models are conditional to this event occurrence. These approaches are not sensitive to observed or unobserved covariates that are constant over time. However, self-controlled designs remain sensitive to systematic differences between risk periods and control periods. A few single group designs are introduced below.

The *Case-Crossover* method compares, for each individual, a single risk period immediately preceding the adverse event to one or several control periods, always preceding the risk period. The length of these risk and control periods are the same for all individuals. The association between drug exposure and the adverse effect is measured through *case crossover odds ratios* defined as the rate of exposure during the risk period divided by the rate of exposure during control periods. These rates can be estimated using a conditional logistic regression or a conditional Poisson model [AGT14].

Similarly, the *Self-Controlled Case Series* (SCCS) design relies on case data. However, instead of relying on time-periods common to all patients, risk and control periods are defined individually according to the information available during the whole observation period [FW06]. An observation period is defined according to assumptions often associated with the event of interest. Then, risk periods (or *exposure periods*) are characterized according to drug exposures times and a set of assumptions specific to the drugs or event under study. The control periods are defined as the periods when individuals are observed but not exposed. In opposition to case-crossover, this method is bi-directional as it uses information from both the periods preceding and following the event time. The relative risk of being exposed is estimated using a conditional Poisson model. The drug effect is then assessed by comparing the target event relative risk during exposure and control periods.

Self-controlled Cohort is a self-controlled design applied directly to a population as a whole in contrast with the previous designs modeling individual patients' trajectories [RSM13]. It estimates Incidence Rate Ratios (IRRs) as

$$IRR = \frac{(x_0/t_0)}{(x_1/t_1)},$$

where t_0 (resp. t_1) are the length of post-exposure (resp. pre-exposure) risk periods, and x_0 (resp. x_1) the number of adverse events observed during post-exposure (resp. pre-exposure) risk periods.

Hybrids

Other approaches borrow ideas from the two design families previously described. *Information Component Temporal Pattern Discovery* (ICTPD) is a variant of self-controlled cohort comparing patients with themselves (self-control) and assessing the existence of systematic differences by comparing case time intervals with equivalent periods in a control group (case-control). This approach adds a comparator group to self-controlled designs to control systematic differences between the risk and control periods. However, this approach remains sensitive to systematic differences between risk and control periods unique to the case group [Nor+13].

Others

Disproportionality analysis directly compares drug-event pairs co-occurrences using χ^2 tests to identify pairs which are more often reported together [Mon+11].

Longitudinal Gamma Poisson Shrinker is based on a similar idea, adapted to longitudinal data [Sch11]. Instead of merely counting events or non-events occurring within or outside drug exposure periods, it considers the length of drug exposure and non-exposed time to detect disproportionality. This approach is combined with *LEOPARD*, an algorithm comparing drug prescription rates in a fixed window, before and after the occurrence of a target event. It allows us to detect false positives caused by protopathic bias, i.e. situations when a drug is prescribed to cure the target event or its early manifestation instead of causing it [Fai15].

Comparison

OMOP developed benchmarks to evaluate these approaches' performance on ADRs detection when using claims databases [Rya+12; Rya+13b]. To perform these benchmarks, the researchers produced an ADRs database containing drug and adverse events pairs. They estimated different combinations of models and designs, varying hypotheses, and hyperparameters to produce binary answers for each (drug, reaction) pair. These answers were compared to a database [Rya+13a] listing positive and negative associations between molecules and reactions. These benchmarks concluded to a better performance of self-controlled designs over case-controlled designs. The scarcity of demographic and individual habits data in claims databases may explain this conclusion as it hinders control matching when using case-control designs. However, the evaluation method presented in [Rya+13b] has several shortcomings:

- Estimates are produced iteratively on drug and reaction pairs, which poses a high risk of obtaining estimates biased by unobserved confounding variables.
- Their ground truth ADR database has since been criticized for having misclassified some of the considered pairs [HAF16].
- The method used to choose between the many assumptions and hyperparameters is likely to overfit the ADR corpus, as they did not use an ADR testing set distinct from their training set.

2.4 Selected approach

Both human detection of ADR and tailored risk quantification studies are not scalable enough to perform large scale ADR screening. Indeed, the latter requires a tremendous amount of manual tuning to provide results, see for example [Neu+12].

Moreover, LOD-specific issues raised in Section 2.1 show that developing a fully automated ADR detection system on SNDS data would suffer from too many biases to be effectively used in practice [Mad+14]. However, the OMOP benchmarks described in Section 2.3 indicate that models combined with self-control designs might be robust enough to derive useful information from claims data when it comes to detecting ADRs. These observations motivated the development of a new model to improve ADR detection. The goals of this model are the following:

- To be easily interpretable and easy to use in order to foster its adoption by practitioners.
- To be as robust as possible to unobserved confounding factors.
- To handle sparse data and rare events as ADRs are mainly rare events.
- To ease exposure or risk period definition.
- To be flexible enough to use as little prior knowledge as possible.
- To be scalable to large populations and many drugs.

Let us assume that data is available from a global observation period $(a, b]$, where the time can be either calendar or measured by patients' age. Each patient $i = 1, \dots, m$ has an observation period $(a_i, b_i] \subset (a, b]$, in which we observe:

- the time occurrences $t_{i,1} < t_{i,2} < \cdots$ of the event of interest (also called *outcome* in what follows), or, equivalently a counting process N_i, defined as $N_i(t) = \sum_{k \geq 1} \mathbb{1}_{t_{i,k} \leq t}$ and $n_i = \int_{(a_i, b_i]} dN_i(t)$ the total number of outcomes of patient i,
- a vector of d longitudinal features

$$X_i = \big(X_i(t) = (X_i^1(t) \ \cdots \ X_i^d(t)) \ : \ t \in (a_i, b_i]\big),$$

  where in the context of drug safety studies, $X_i^j(t)$ gives us information about the exposure of patient i to drug j at time $t \in (a, b]$.

The modelization uses point processes as they are a natural tool to represent irregularly-sampled series of events while focusing on interpretability. Building upon the Cox model was ruled out, as this model is hard to scale [Ach+15]. Following the works on SCCS model [Far95; FW06; Sch+16], using a conditional Poisson process seemed to be a good starting point. As such, the model relies on the usual SCCS model key assumptions [FW06]. Namely, we assume that

(i) The features are exogenous, meaning that the counting process N_i does not have any influence on the features X_i;

(ii) The interval of observation $(a_i, b_i]$ is independent of N_i;

(iii) The process N_i is a Poisson process conditionally to $(X_i(t) \ : \ t \in (a_i, b_i])$.

Assumption (i) allows to condition on the full trajectory of the longitudinal features X_i in (4). In addition, thanks to (ii), the following derivations have to be understood conditionally to $(a_i, b_i]$. We may then define the conditional intensity of the process N_i as

$$\lambda_i(t, X_i) = \mathbb{P}(dN_i(t) = 1 \mid X_i) \tag{4}$$

for $t \in (a_i, b_i]$. Therefore, this model can be understood as a regression model, allowing to regress the outcomes in N_i on the longitudinal features X_i.

In order to study acute vaccine adverse effects, [FW06] considers the following model for the intensity:

$$\lambda(t, X_i) = \exp\big(\psi_i + \gamma_i + \phi(t) + X_i(t)^\top \beta\big),$$

where ψ_i is the baseline incidence of patient i and γ_i is a sum of non-temporal fixed and random individual effects. The parameter $\phi(t)$ is a time-dependent baseline which is common to all individuals. If age serves as the time scale, this term can help to capture age effects. The vector of parameters $\beta \in \mathbb{R}^d$ quantifies the effect of the longitudinal features $X_i(t)$ on the intensity. The idea of the SCCS method is to condition on both X_i and n_i. Usual arguments (see Chapter II) imply that the likelihood of $N_i|(X_i, n_i)$ of $i = 1, \dots, m$ independent patients is proportional to

$$\prod_{i=1}^{m} \prod_{k=1}^{n_i} \frac{\lambda_i(t_{i,k}, X_i)}{\int_{a_i}^{b_i} \lambda_i(s, X_i) ds} = \prod_{i=1}^{m} \prod_{k=1}^{n_i} \frac{\exp\big(\phi(t_{i,k}) + X_i(t_{i,k})^\top \beta\big)}{\int_{a_i}^{b_i} \exp\big(\phi(s) + X_i(s)^\top \beta\big) ds}. \tag{5}$$

Note that the conditioning with respect to n_i induced two notable properties of Equation (5):

- *Improved scalability:* the likelihood only depends on patients i such that $n_i \geq 1$ (while the "full" likelihood of $N_i|X_i$ does depend on patients i for whom $n_i = 0$). This is beneficial when studying rare adverse effects in large LODs.

- *Robustness to non-longitudinal confounders:* the non-longitudinal effects ψ_i and γ_i cancel out in the likelihood (Equation (5)). This trait makes SCCS models particularly robust to the patient's susceptibility.

These two properties mitigate issues related to missing variables and data scale, which is valuable when working with LODs such as claims databases. However, only relative incidences can be computed by taking the exponential of the corresponding coefficient, such as $\exp(\phi(t))$ for the baseline relative incidence.

2.5 Contribution: Convolutional SCCS

SCCS models were initially designed for vaccine safety studies [Far95], using the suspected ADR as the outcome. In this context, estimating the relative incidence of drug use requires defining related time-at-risk periods in which the suspected ADR might occur. The longitudinal features $X_i(t)$ then denote if the patient i is at risk or not at time t for a particular drug. One must then determine how long patients are at risk after each drug exposure and if this risk occurs either immediately or after some delay. Defining proper time-at-risk windows is a challenging problem when studying a single (drug, ADR) pair, which worsens even further when considering a set $(\text{drug}_1, \text{ADR}), \dots, (\text{drug}_d, \text{ADR})$ of such pairs. In the case of ADR screening over numerous drugs, such a methodology might even become infeasible.

Discrete SCCS. As LOD data is discrete (e.g. SNDS data is recorded daily), the intensity λ is assumed to be constant over time intervals $I_k = (t_k, t_{k+1}]$, $k = 1, \dots, K$ that form a partition of the observation interval $(a, b]$, for $i = 1, \dots, m$. Without loss of generality, I_k were chosen to be of constant length 1. In practice, the smallest granularity allowed by data is used as the length of these intervals. Hence, we can assume that $(a_i, b_i] \cap I_k$ is either $\varnothing$ or I_k for all $i = 1, \dots, m$, and $k = 1, \dots, K$, which means that the observation period of each individual is a union of intervals I_k. Denoting by $\lambda_{i,k}$ the value of $\lambda(t, X_i(t))$ for $t \in I_k$, and defining $y_{ik} := N_i(I_k)$, the discrete SCCS likelihood can be written as

$$L(y_{i1}, \dots, y_{iK} | n_i, X_i) = n_i! \prod_{k=1}^{K} \left(\frac{\lambda_{ik}}{\sum_{k'=1}^{K} \lambda_{ik'}} \right)^{y_{ik}}.$$

ADR flexible estimation. When prior knowledge on time-at-risk windows is not available, a simple method is to use a large window in order to be sure to capture the potential effect. However, this strategy typically "dilutes" the risk over the window, see [Xu+11], leading to a model unable to detect ADRs.

A different approach relies on fitting time-dependent parameters to estimate the risk of ADR over large risk windows. The model estimates a time-varying relative incidence function along with the risk window instead of assuming it to be constant. This approach was first used in [GWF16; GWF17; Sch+16], who used splines to model the drug effect as a function θ depending on longitudinal exposures. However, both [Sch+16] and [GWF16; GWF17] seem restricted to the study of a single (drug, ADR) pair at a time. This limitation can be problematic since SCCS is sensitive to time-varying confounders and benefits from studying multiple drugs at once, as shown by both [Sim+13] and [MRM16].

In order to derive a multivariate flexible model, we simplified the formulation of the effect of longitudinal features by using convolutions of low-granularity step

functions with point drug exposures. Assuming that the intensity is constant on each I_k, it writes

$$\lambda_{ik}(X_i) = \exp\Big(\psi_i + \gamma_i + \phi_k + \sum_{k'=a_i}^{k} X_{ik'}^\top \theta_{k-k'}\Big),$$

where X_{ik} stands for the value of $X_i(t)$ for $t \in I_k$, $\theta \in \mathbb{R}^{d\times K}$, where ψ_i is the baseline incidence of patient i and γ_i is a sum of non-temporal individual effects. The parameter ϕ_k is a time-dependent baseline which is common to all individuals, such as the effect of age.

We observe $l = 1, \dots, L_i^j$ starting dates of exposures c_{il}^j and introduce the features $X_{ik}^j = \sum_{l=1}^{L_i^j} \mathbb{1}_{k=c_{il}^j}$, which leads to the following intensity

$$\lambda_{ik}(X_i) = \exp\Big(\psi_i + \gamma_i + \phi_k + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k-c_{il}^j} \mathbb{1}_{[0,p]}(k - c_{il}^j)\Big). \tag{6}$$

The quantity $\exp(\theta_k^j)$ corresponds to the relative incidence of an exposure to drug j that occurs k time units after an exposure start. Finally, the likelihood writes

$$L(y_{ik}, \dots, y_{iK}|n_i, X_i) = \prod_{k=1}^{K}\left(\frac{\exp\left(\phi_k + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k-c_{il}^j} \mathbb{1}_{[0,p]}(k - c_{il}^j)\right)}{\sum_{k'=1}^{K}\exp\left(\phi_{k'} + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k'-c_{il}^j} \mathbb{1}_{[0,p]}(k' - c_{il}^j)\right)}\right)^{y_{ik}}$$

and depends only on the parameters θ for the exposures and the time-dependent baseline ϕ. Such a flexible risk modeling combined with binary exposures have been shown to provide optimal results when no prior knowledge on ADRs dynamics is available [GAB15].

Feature selection. This formulation of intensity (6) is flexible since it allows to capture an immediate effect in θ_0^j, or delayed ones using θ_k^j for $k \geq 1$. This flexibility comes at a cost: it increases the number of parameters to be estimated significantly, eventually leading to inaccurate estimations and dataset overfitting. Therefore, we introduce a penalization that regularizes the parameters and helps the interpretation of the estimated relative risks.

We introduce groups $\theta^j = [\theta_1^j \cdots \theta_p^j] \in \mathbb{R}^p$ of parameters quantifying the impact of exposures to drugs $j = 1, \dots, d$ at different lags $k = 1, \dots p$. To avoid an overlapping of the exposure effects, we assume that exposure starting times are far enough, that is $\min_{l,l'} |c_{il}^j - c_{il'}^j| > p$. We want to induce two properties on the relative risks of drugs exposures: a "smoothness" property over lags $k = 1, \dots, p$, namely we

want consecutive relative risks $\exp(\theta_k^j)$ and $\exp(\theta_{k-1}^j)$ to be close; and the possibility for a drug to have no effect, namely to induce that θ^j can be the null vector. This can be achieved with the following penalisation that combines total-variation and group-Lasso

$$\text{pen}(\theta) = \gamma_{\text{tv}} \sum_{j=1}^{J} \sum_{k=1}^{p-1} |\theta_{k+1}^j - \theta_k^j| + \gamma_{\text{gl}} \sum_{j=1}^{J} \|\theta^j\|_2. \tag{7}$$

The effect of the 1-D total-variation penalty is illustrated Figure 6.

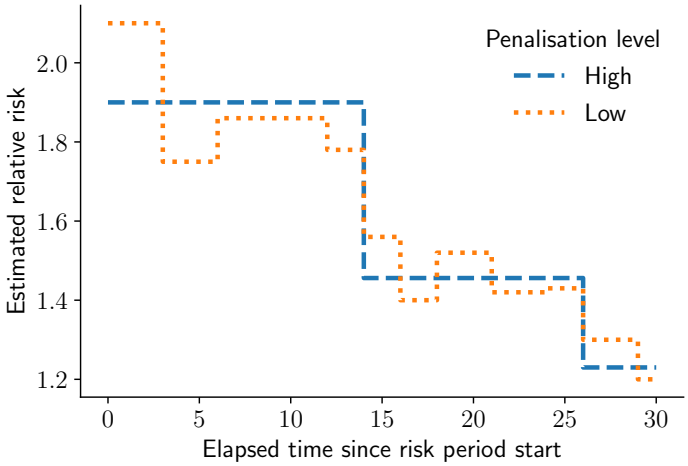


Figure 6 – Illustration of the Total Variation penalization effect. Assuming a risk period starting at 0 and lasting for 30 periods, ConvSCCS will estimate a 30-period piece-wise constant relative risk curve. The level of Total Variation penalization controls the total size of the jumps. A high (resp. low) level of penalization results in more (resp. less) restricted relative risk curves, illustrated by the small orange dashes (resp. long blue dashes) curve. The model's fitting algorithm aims to reach a good balance between the detail level and the estimated relative risk curves' smoothness.

Estimation. The penalised negative log-likelihood of our model then writes as follows:

$$-\ell(\phi, \theta) + \text{pen}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log\left(\frac{\lambda_{ik}(\phi, \theta)}{\sum_{k'=1}^{K} \lambda_{ik}(\phi, \theta)} \right) + \text{pen}(\theta), \tag{8}$$

where pen is given by (7) and λ_{ik} by Equation (6).

The objective (8) is convex and $\ell(\phi, \theta)$ is L-smooth. However, since the sparsity-inducing penalisation $\text{pen}(\theta)$ is not differentiable, we use a proximal first-order

method to minimise efficiently (8). Namely, we use the state-of-the-art SVRG algorithm from [XZ14], which is a fast stochastic proximal gradient descent algorithm, using a principle of variance reduction of the stochastic gradients. Finally, the hyperparameters γ_{tv} and γ_{gl} are selected using stratified V-fold cross-validation on the negative log-likelihood.

Interpretability. ConvSCCS estimates a relative risk curve $\exp(\theta^j)$ of length p for each feature $j = 1, \dots, d$. When using point exposures, these curves can easily be interpreted as the relative risk $k = 0, \dots, p$ periods after exposure start. Confidence intervals for these curves can be estimated using parametric bootstrap, as explained in Chapter II.

Implementation. ConvSCCS is available in `tick` [Bac+17b], an open-source machine learning library. This implementation provides a Python API, while computationally intensive operations are implemented in C++. Cross-validation is parallelized over multiple CPUs.

Performance on synthetic data. ConvSCCS was compared with the state-of-the-art SCCS models, namely SmoothSCCS [GWF16] and NonparaSCCS [GWF17]. Patient histories were simulated using random drug exposures (see Chapter II for more details) and relative risks represented in Figure 5. The performance was measured as the mean absolute error between the estimates and true values for the relative incidences θ^j and the longitudinal baselines ϕ. Figure 7 shows that fitting the effect of several drugs at the same time and using our penalization provides a better estimation accuracy than NonParaSCCS and SmoothSCCS, the improvement being larger for the estimation of drugs exposures risks profiles than for the baseline. Figure 8 gives the run times of all three procedures. ConvSCCS seems to scale better than both SmoothSCCS and NonParaSCCS when fitting a large number of feature such as $d = 14$ on $m > 2000$ cases. In small studies, however, when $d = 4$ for example, SmoothSCCS is the fastest algorithm, while NonParaSCCS is overall slower than the two other algorithms.

2.6 Applications

Existing works on ADRs screening such as [Rya+13b] evaluated the performance of their methodology by comparing their results to an adverse drug reaction database containing established positive and negative association [Rya+13a]. While this approach is convenient, ADRs databases' reliability has been criticized as there is evidence of misclassified associations [HAF16]. In place of this evaluation scheme,

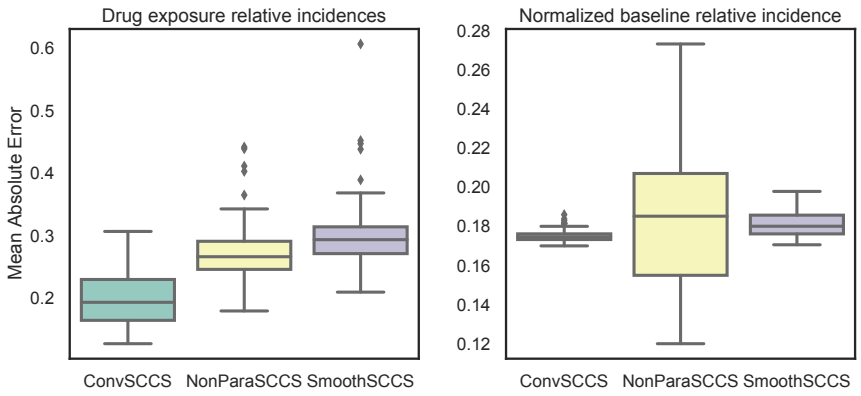


Figure 7 – Results on synthetic data using risk profiles illustrated in Figure 7) with $m = 4000$ cases. The boxplots represent the distribution of mean absolute error, computed over 100 simulated populations. *Left:* MAE distribution of the drug exposure relative incidences θ. *Right:* MAE distribution of the baseline relative incidences ϕ, constrained so that their integral is equal to one.

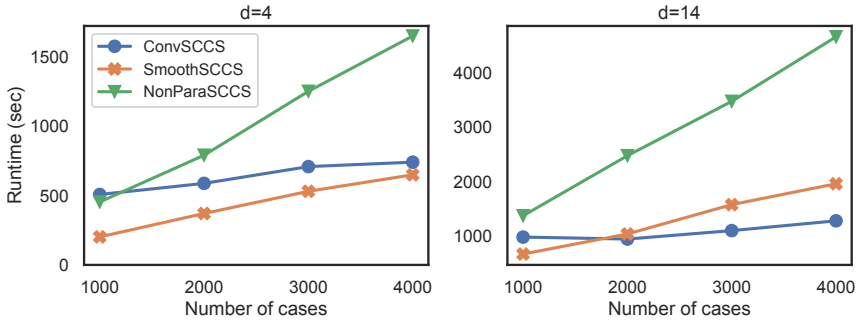


Figure 8 – Run times of ConvSCCS, SmoothSCCS and NonParaSCCS for 1000, 2000, 3000, 4000 cases. *Left:* run times on 4 features. *Left:* run times on 14 features. As SmoothSCCS and NonParaSCCS can only handle one feature at a time, we report the time required to fit them on each studied feature while ConvSCCS is fitted on all the features simultaneously. For each model, a fit includes cross-validation of the hyperparameters and estimation of confidence bands (see Chapter II for more details).

we evaluate our screening methodology by comparing our results to existing meta-analyses and results obtained with other methodologies. The two use-cases presented below considered homogeneous groups of molecules against an adverse event.

The first one focused on antidiabetic molecules exposure and bladder cancer, which is an ADR developing slowly over time. These associations were already

studied in [Neu+12] which is our baseline.

The second application focused on Anxiolytics, Hypnotics, Antidepressants, and Neuroleptics (AHANs) use. AHANs can induce changes in perception or drowsiness, causing falls and fractures among the elderly. Fall-related fractures are likely to occur more suddenly than ADRs such as cancers. This screening study was not aiming at reproducing existing results, but rather at extending the knowledge on these associations (see Chapter III) on which there is no consensus.

Note that while SNDS is often used to perform drug safety studies [Bez+17; Tup+17a], it has been used only very recently to perform ADR screening [Thu+20] using methods close to the ones described in [Rya+13b].

Antidiabetics and bladder cancer

Pioglitazone was withdrawn in France in 2011 due to an association with bladder cancer among men. This association has been observed using SNDS data in [Neu+12]. This study was tailored to SNDS data to minimize potential bias, at the cost of many *ad hoc* assumptions.

The study focused on French beneficiaries aged from 40 to 79 years on 2006/12/31 who filled at least one prescription for a glucose-lowering drug in 2006. The observation period ended on 2009/12/31. Using similar data, we used ConvSCCS to assess if the ADR (bladder cancer) was correctly detected by the model when using fewer assumptions than the initial study.

We used the same definition for the bladder cancer outcome as in [Neu+12]. The cohort contained 1699 cases of bladder cancer, which is roughly 400 missing cases in comparison to [Neu+12]. We also had less history before the follow-up to filter prevalent cases, due to French data regulation imposing patients information deletion after ten years[10]. More details about the cohort structure and construction can are available in Chapter II. Patients were exposed to a molecule as soon as they purchased a drug containing this molecule. Exposures were not limited in time. As diabetic patients use hypoglycemic agents continuously, exposure starting dates might be noisy. Time intervals were set to 30-days based on calendar time to take into account the small number of cases. The risk windows lasted $K = 24$ months after drug exposure start. The validity of ConvSCCS assumptions for this dataset is discussed in Chapter II.

Figure 9 displays the estimated relative risk curves (RRCs, or *relative incidence* curve) and 95% bootstrap confidence intervals for all investigated glucose-lowering drugs. Thanks to the penalization used in ConvSCCS, the estimated RRCs and confidence intervals are piece-wise constant on large steps: this is particularly interesting since it allows us to detect only significant variations of the relative risks.

[10]This deletion delay has been extended to twenty years since then.

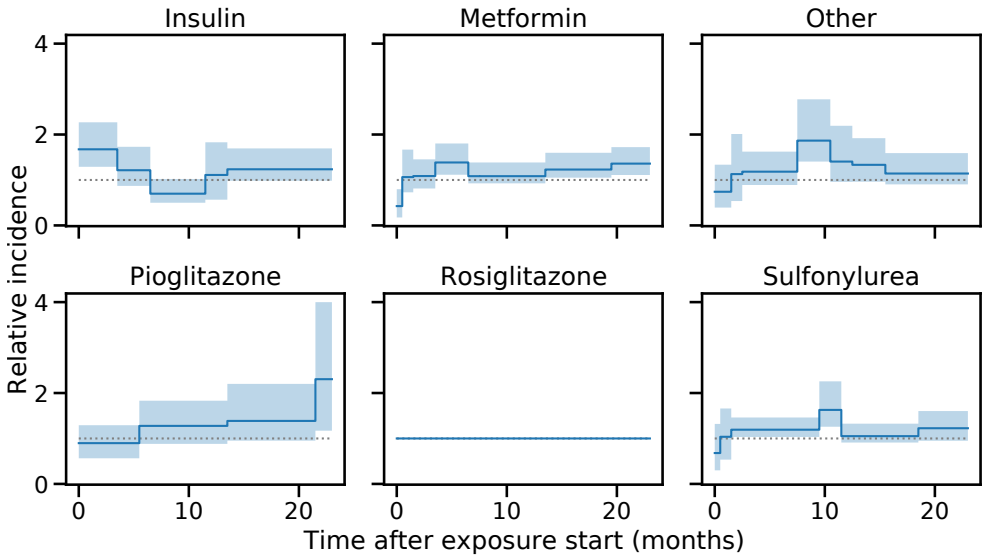


Figure 9 – Estimated relative incidence curves of glucose lowering drugs on the risk of bladder cancer. Dark blue curves represent the estimated relative incidence curves $k = 0, \dots, 23$ months after the beginning of an exposure. Light blue bands represent 95% confidence intervals estimated by the parametric bootstrap, with 200 bootstrap samples.

As shown in Figure 9, a strong positive association between pioglitazone and the risk of bladder cancer was recovered. The corresponding relative incidence increases over time from 6 to 24 months after exposure start. The values and breakpoints of this relative incidence curve are consistent with the results presented in [Neu+12] (see Chapter II for more details).

The results comparison for other hypoglycemic agents hazard ratios was more challenging since [Neu+12] did not estimate longitudinal risks for these molecules. While [Neu+12] did not found the other hypoglycemic agents to be statistically significant, our model cancels out the effect of rosiglitazone and finds that the other molecules are non-significant statistically during most of the lags after exposure start. However, sulfonylurea and "other" have significant positive estimates from lags 9 to 11, as well as insulin from lag 0 to 5. The shape of these three curves suggests there might be some colinearity issues between these three features. Indeed, the magnitude of their relative incidence curves seems to either match or be of opposite signs for similar lag values. Metformin seems to be non-significant overall, despite few coefficients suggesting a positive association. While these results are not a perfect match to [Neu+12], they show that our model might be useful when exploring quickly large sets of molecules with a reduced amount of data preprocessing, even when the con-

ditions are sub-optimal (noisy timestamps, possible feature endogeneity, and feature colinearity). Indeed, when compared to [Neu+12], our methodology is scalable in the number of drugs since it does not require the same precise preprocessing work.

Contribution: Screening AHANs association with fracture risk among the elderly

This second application aims at screening associations between AHANs and fractures among the elderly using SNDS data.

AHANs and fracture risk associations have already been investigated at different levels of granularity and scopes in numerous clinical and observational studies. Fractures among the elderly are a prominent public health issue as they are associated with high morbidity and mortality [Dea+10; Vri+18]. They can be caused by reduced bone mineral density or postural instability [All+05], both of which might be influenced by the use of AHANs. Meta-analyses, such as [Sep+18a] or [Woo+09] highlight how hard establishing a comprehensive mapping of fracture risk and molecules association can be, as most studies scope is limited to a single drug or drug class. To raise the level of evidence, [Sep+18a] calls for studies investigating pharmacological subgroups rather than large drug classes, as well as duration effects, which is the purpose of ConvSCCS.

Design. This study uses a self-control design on new-users, i.e. on patients who *started* to use AHANs during the observation period. Excluding patients already exposed to AHANs in the first year of observation prevented the risk dynamics estimated by the model to be affected by prevalent[11] drug use. Note that this strategy was hardly feasible in the study on bladder cancer described above, as diabetic patients' condition requires continuous antidiabetic use. Subjects entering the cohort had to:

(i) Be covered by the universal health insurance coverage, which is the case for 98.8% of France inhabitants [Tup+17a],

(ii) be 65 y.o. or older on the 1^{st} of January 2015,

(iii) receive their first outpatient target drug prescription at least 365 days after study starts on the 1^{st} of January 2014 to prevent prevalent users or to provide a sufficient wash-out delay.

Restriction to 65+ y.o. patients result in a more homogeneous population in terms of professional activity (retirees), behaviour (response to a fall, sports practice),

[11] Drug exposure starting before the observation period.

and characteristics (bone density), all of which might affect fracture risk. Patients meeting all these conditions entered the cohort on the 1^{st} of January 2014. They exited the cohort exit was defined as (i) death; or (ii) end of the study period, the 31^{th} of December 2017.

Drug exposures computation differed from the method used in the bladder cancer application. Indeed, existing literature on AHANs molecules suggests a rapid onset of fall-related ADRs. The length of the time intervals was set to one day to take this into account. Exposures were limited in time and preceded by pre-exposure risk periods, as illustrated in Figure 10. Such use of pre-exposure risk windows is not new [NN19; Pra+11; Req+20], especially when using flexible longitudinal models. This exposure construction allowed for multiple exposure periods for a single molecule within a patient's history. Fracture events were extracted following the query presented in [Bou+20]. Chapter III provides additional details regarding exposures and fracture identification.

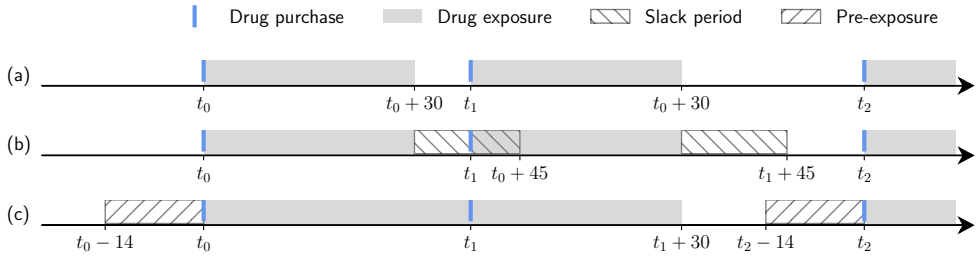


Figure 10 – Drug exposures computation. Exposures are assumed to last for 30 days (90 days for large drug packaging) after drug purchases (i). A slack period is added (ii) to account for slight variability in drug purchasing dates. Exposures which overlap with other exposures or other exposures' slack period are merged (iii). A 14-day pre-exposure period is then added before each exposure starting point (iii).

Relative risk curves interpretation. Pre-exposure relative risk curves (RRCs) are useful to assess biases resulting from LOD-specific care pathways. A pre-exposure RRC > 1 suggests an indication bias, when the molecule is likely to be prescribed in reaction to an event associated with the target event. On the contrary, a pre-exposure RRC < 1 might highlight protective environments such as hospitalizations preventing patients from experiencing a fracture. It typically occurs when a molecule is prescribed during a stay for post-hospitalization care. The drug delivery date then always follows a protective period. Both effects can mingle when the studied event has a chance to cause hospitalization and is an indication for the molecule. The resulting pre-exposure RRC is greater than one at the start of pre-exposure,

then decreases sharply to values below one. This interpretation was consistent with sensitivity analysis experiments restricting the fracture definition to a given severity level (see Chapter III for more details). Even though pre-exposure estimates do not prevent biases resulting from indication biases or protective environment, they create a useful context to understand screening results properly. This additional information might be valuable when designing further confirmation studies.

Results. The cohort selection process resulted in 126,567 fracture cases (as detailed in Chapter III). Dynamic risk estimation produced a broad set of relative risk curves, resulting in more informative results than point estimates providing yes/no answers. As such, this approach fosters human interpretation of data-mined patterns rather than pursuing a fully automated alert generation system.

Overall, the results were consistent with meta-analyses (Chapter III). For the sake of conciseness, this introduction only features the relative risk curves (RRCs) of antidepressants, represented Figure 11.

Antidepressants RRCs were consistent with the results presented in separate studies [Sep+18a; Ves09]. The increase in relative risks after exposure was smaller among tricyclic antidepressants (TCAs) than selective serotonin reuptake inhibitors (SSRIs), serotonin-norepinephrine reuptake inhibitors (SNRIs), and tetracyclic antidepressants (TTCAs). We also observed decreasing RRCs for citalopram, escitalopram, sertraline, mianserin, mirtazapine, and venlafaxine similarly to [Hub+03]. However, we estimated a constant RRC for amitriptyline while [Hub+03] found a decreasing RR. Amitriptyline pre-exposure RRC was above one, indicating a potential confounding by indication, perhaps resulting from its use in neuropathic pain management, especially after spinal cord injury [AJ17]. Aside from amitriptyline, pre-exposure RRCs were either non-significant or below 1, which suggests post-hospitalization prescriptions but no indication bias. This observation is consistent with SSRIs [Mor+13] and mirtazapine [Hon+07] being prescribed following a myocardial infarction.

2.7 Discussion

We showed that our approach mixing cautious study design and an easy-to-tune flexible statistical algorithm could be used to produce broad results highlighting eventual associations and indication or database-specific longitudinal biases. This approach is easy to implement as it relies on open-source, scalable libraries. It does not require much fine-tuning; it can handle large populations and many molecules; it relies on a few ascertainable assumptions and provides easily interpretable results. Cohort construction and exposure and event definitions help to mitigate some of the database biases, without injecting over-restrictive prior knowledge to retain model plasticity. Flexible longitudinal pre and post-exposure relative risk curves

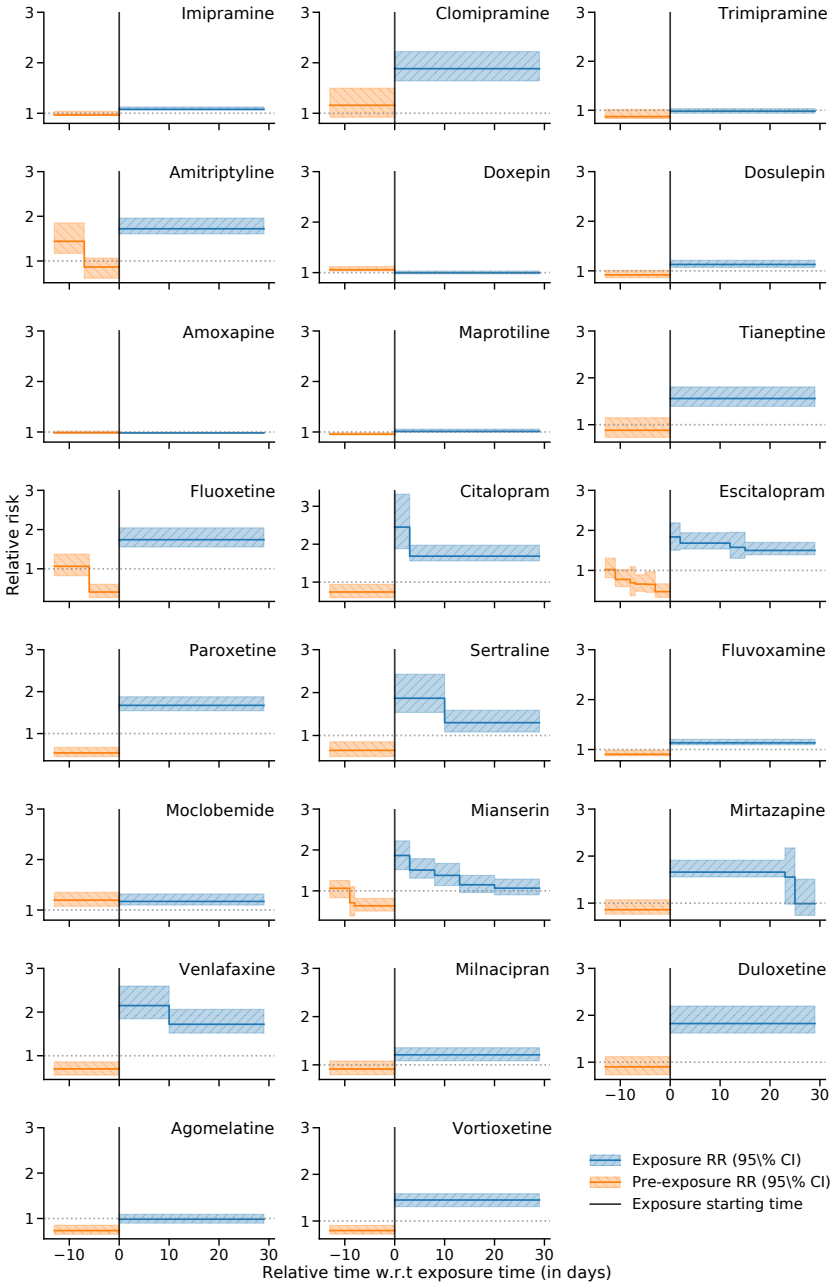


Figure 11 – Fracture relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Blue (resp. orange) solid lines represent post-exposure (resp. pre-exposure) relative risk, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

provide information on healthcare pathways, helping to highlight large observational databases specific biases. While the properties of our approach make it robust to some biases and can detect additional ones, its result should still be interpreted with care, and rely on the co-operation of medical experts and statisticians. Extensive sensitivity analysis, such as the one featured in Chapter III, greatly help to spark stimulating and fruitful discussions with health professionals. We believe it can perform risk detection on large sets of molecules effectively while contextualizing these risks to ease further confirmation studies.

3 Learning representations for health data

Labels in healthcare data can be scarce (e.g. rare diseases, see [MH20]) or expensive [Shi+18] to obtain. Even when using large databases, the relevant study populations might be small, depending on the task. For example, the cohort of 1.4 million diabetic patients featured above resulted in 1,699 cases of bladder cancer (Table II.E.1).

Small populations are likely to result in low-quality estimates, even more as the model complexity grows [RJ+91]. *Multitask learning* can be a way to bypass this issue when several tasks are related and can be performed using a shared representation. The improvement is twofold. First, learning on several tasks can increase the quantity of available labeled data as each task might bring new labeled samples. Second, each task might provide a different perspective of the studied phenomenon, resulting in more robust representations of data points [Car97]. Thus, a multitask model can eventually generalize better than single-task models.

However, models such as ConvSCCS are not readily adaptable to multitask setups. Indeed, ConvSCCS requires assumptions that might be incompatible across several tasks. For example, assumptions on exposures might conflict: finite repeated exposures performs well at short-term risk estimation, while infinite unique exposures are better for long-term risk estimation. Restrictive assumptions help control database-specific biases (e.g. coding errors, noise in timestamps) and model estimation on small datasets by introducing useful constraints. However, they hinder multitask-learning, which requires more flexible models able to learn rich and polyvalent representations.

In the last few years, recurrent *deep learning* models produced slight improvements on several tasks related to healthcare. These models seem to benefit from multitask learning [Har+19]. Contrary to the earlier approach, deep learning models learn from (almost) raw data and do not require restrictive assumptions defining concepts such as drug exposures. These algorithms consist of stacking several smaller differentiable models (*layers*). Ideally, each layer progressively learns to extract higher-level features from the previous layer's output [LBH15]. Such algorithms have the advantage of requiring fewer assumptions and data preprocessing, but they

require large amounts of data and are difficult to interpret [Cha+17]. Note that the training of deep learning algorithms is usually a non-convex problem. Hence, optimization algorithms must be adapted and can only reach a local optimum [LBH15; Rud16].

Despite the flexibility of deep learning models, a multitask model must be learned from scratch when a new task arises, which can be a significant drawback. *Unsupervised pre-training* can be an answer [Rad+19]. Unsupervised learning does not require human-labeled data, allowing to use more samples from large observational databases. Pre-training is a form of *transfer learning*[12], consisting of training a model on a *pretext task* designed to learn *useful* input data representations (see Section 3.2 below for examples). Here, "useful" means that the pre-trained representation could easily be adapted to other tasks (*downstream tasks*) unknown at training time. Note that contrary to multitask learning, it is possible to reuse pre-trained models on new tasks unanticipated at its pre-training time. The pre-trained model adaptation to downstream tasks can be made by adding one or several task-specific layers. Thanks to efficient pre-learned representations, the resulting model's training is faster and requires fewer data to reach similar performance than a model trained from scratch.

Unsupervised pre-training was an essential ingredient of recent successes in Natural Language Processing (NLP) [Dev+18; Rad+19], but also for time series [FDJ19] and computer vision, where deep encoders are pre-trained using self-supervised [DZ17] or *contrastive* [Che+20; OLV18] approaches. A parallel is often drawn between EHR data and NLP [Aya+20; SRB19], since both can be represented as sequences of tokens, corresponding to words or word pieces in NLP and medical codes in EHR. Recent attention models and pre-training strategies resulted in considerable improvements on many NLP tasks [Dev+18; Rad+19; You+18]. In particular, transfer learning has recently proved to be very effective for NLP, while it was already the case in computer vision [HR18].

Self-supervised learning is a recent form of *unsupervised learning*, which involves a pre-training step on a large unsupervised dataset, using a pretext task, followed by a fine-tuning step for specific supervised tasks. One of the most famous examples is Bidirectional Encoder Representations from Transformers (BERT) [Dev+18], with numerous extensions [Dai+19; Lan+19; Liu+19; Yan+19]. The work presented below tries to adapt some of these approaches to EHRs.

[12]Transfer Learning focuses on solving a machine learning problem by reusing knowledge gained from solving another problem.

3.1 Deep learning architectures for healthcare: From NLP to EHR.

While both text data and EHR are sequences of tokens with large vocabularies, EHR exhibit characteristics that do not exist in NLP:

(i) The order of tokens in texts is somewhat self-evident, while the ordering of tokens in EHR is specific to the medical practice. Temporal relationships between types of codes is a crucial component of EHR that does not exist in NLP.

(ii) As discussed in Section 2.1, EHRs are not direct recordings of the patients' physiology, but rather captures their interactions with the healthcare system, resulting in feedback loops and reversed dynamics [HA13].

(iii) EHR can contain much longer dependencies than text. For instance, a diabetes diagnosis is a risk factor all along with a patient's life, or some surgeries can prohibit other interventions, even decades later [Shi+18].

Formalisation. Each EHR can be considered as a sequence of timestamped events $z_i = (x_i, t_i)$, where $x_i \in \mathbb{R}^d$ are tokens representing medical codes and t_i are the associated timestamps. These sequences are first *embedded* to reduce the dimensionality of the feature space. Dense embeddings considered in this work are learned vectors of dimension $D \ll d$ corresponding to a specific token. Timestamps embedding can also use fixed representations based on multiple dilated sine waves [Vas+17]. The embedding of z_i is denoted e_i.

Sequence encoding. Tasks considered in this section *encode* event sequences into representations useful to learn several tasks. A part of the model called the *encoder* takes as input a sequence of event embeddings $\mathbf{e} = [e_1, \dots, e_n]$, where $e_i \in \mathbb{R}^D$ for $i = 1, \dots, n$ and outputs a sequence of contextualized embeddings of the same length. Several deep learning architectures described below can be used to perform sequence encoding. These architecture consist of multiple layers $l = 1, \dots, L$, computing *hidden states* $\mathbf{h}^l = [h_1^l, \dots, h_n^l]$ from the results of the previous layer, $\mathbf{h}^{l-1}$. The input of the first layer is the sequence embeddings $\mathbf{h}^0 = \mathbf{e}$.

Convolutional models. *Convolutional Neural Networks* (CNNs) can have good performance in time-series modeling [Tsa+17]. They can build a sequence representation by convoluting a weights tensor called *kernel* along the temporal axis. ConvSCCS presented earlier can be formulated as a single layer CNN using 1−dimensional

convolutions. CNN layers' structure (see Figure 12) makes them invariant to translation[Geh+17]. They can leverage Graphical Processing Units (GPUs) computational capabilities as their computation is parallelizable. However, their shift-invariance can be an issue when global order matters (e.g. when the recent past is more informative than the distant past) [Cho18].

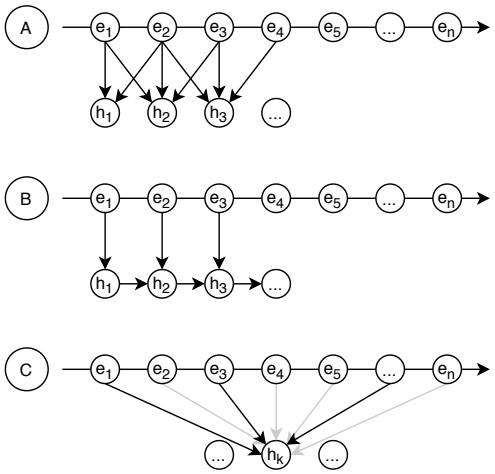


Figure 12 – Representations of three families of deep learning models for sequence modelling. The first layer is represented to illustrate the different sequence representation mechanisms. These layers compute hidden states $h = h_1, \dots, h_n$ from a sequence of embeddings $e = e_1, \dots, e_n$. **(A)**: Convolutional Neural Network (CNN) layer, using a kernel of dimension 3 and "same" zero-padding. Same zero-padding consist of adding zeros on both sides of the sequence S so that the sequence S' resulting from the convolution has the same length as S. Each representation h_k is computed from e_{k-1}, e_k, e_{k+1}, for $k = 1, \dots, n$. **(B)**: Recurrent Neural Network (RNN) layer. Each representation h_k is computed from h_{k-1} and e_k. **(C)**: Attention model layer. Each representation h_k is computed from sequence elements e_j, $j \in \mathcal{J}$ selected by an attention mechanism.

Recurrent neural networks. *Recurrent neural networks* (RNNs) update an internal state h_t recursively, based on their previous state h_{t-1} and the current sequence element e_t (see Figure 12). The structure of RNNs makes them suitable to process variable-length sequences with constant model size as they share weights across time. They have been used successfully to perform time-series prediction [HBB20] and sequence classification [Har+19; Kab+19]. However, they are computationally slow as they cannot fully leverage GPUs for training. Indeed, their recursive structure triggers regular memory I/O to load sequence chunks, quickly becoming a bottleneck and resulting in an under-utilization of the GPUs' computation capabilities. This issue grows with the length of the sequences [HS15].

Attention models. Models relying exclusively on attention mechanisms have been designed to represent sequences without processing the data in order (see Figure 12). Thanks to this feature, they can fully leverage GPUs' parallel computation capabilities resulting in faster training than RNNs. They have been used extensively in NLP and led to large pre-trained models such as BERT [Dev+18] or GPT [Rad+19]. Contrary to RNNs and CNNs, attention models leverage tokens order by embedding their index in the sequence (*positional embeddings*) instead of using their actual position.

Contrary to NLP, EHR sequences contain timestamps. Besides, these sequences are not sampled regularly like economic time series. This irregular sampling can cause issues when using RNNs, as the memory mechanisms used to update hidden states might not take temporal gaps into account. A similar issue arises with CNNs as their performance might suffer from their inability to learn a global order. These issues have been mitigated by performing various types of data interpolation to build regularly sampled time-series from sparse EHR data [Har+19; Tan+20]. However, even if EHR data is sparse, it can consist of very long sequences with a large feature space. Imputed time series can then be very costly in GPU memory. It might result in slow training when using a large dataset designed for self-supervised pre-training. The work featured in Chapter IV tries to use attention models using timestamp embeddings to express temporality in order to solve this issue. This approach would allow working on sparse EHRs sequences while leveraging parallel computations on GPUs. This aspect is critical when using large volumes of data to build reusable, pre-trained models.

The Transformer encoder proposed by [Vas+17] can be used to build a representation of an n-input sequence by stacking Multi-head Self-Attention (MSA) layers. Considering L layers, h heads and d-dimensional token representations, an MSA layer writes as follows:

$$\mathbf{Q}_i = \mathbf{H}\mathbf{W}_i^Q, \quad \mathbf{K}_i = \mathbf{H}\mathbf{W}_i^K, \quad \mathbf{V}_i = \mathbf{H}\mathbf{W}_i^V,$$

$$\mathbf{V}'_i = \text{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}}\right)\mathbf{V}_i, \tag{9}$$

$$\text{MSA}(\mathbf{H}) = [\mathbf{V}'_1, \dots, \mathbf{V}'_h]\mathbf{W}_i^O,$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d\times d_v}$ and $\mathbf{W}_i^O \in \mathbb{R}^{hd_v\times d}$ are learned parameters and $\mathbf{H} \in \mathbb{R}^d$ is the MSA input. The dot product between the queries $\mathbf{Q}_i$ and the keys $\mathbf{K}_i$ in Equation (9) weights the values $\mathbf{V}_i$ to which the MSA pays attention. MSA layers are stacked on L layers as follows

$$\mathbf{Z}^l = \text{LayerNorm}\left(\mathbf{H}^l + \text{MSA}(\mathbf{H}^l)\right)$$

$$\mathbf{H}^{l+1} = \text{LayerNorm}\left(\mathbf{Z}^l + \text{FFN}(\mathbf{Z}^l)\right)$$

where $\mathbf{H}^l \in \mathbb{R}^{n \times d}$ (resp. $\mathbf{H}^{l+1}$) is the input (resp. output) to the l^{th}-layer and FFN is a dense *feed forward network*[13]. The input to the encoder is $\mathbf{H}^0 = \mathbf{e}$, the n-sequence of token embeddings.

Linear transformer A well-known issue with models based on self-attention is their quadratic complexity w.r.t. the length of the sequence n. This complexity comes from the fact that the attention mechanism may focus on any event of the overall sequence. The EHR sequences lengths are approximately distributed according to a power law, which means that a significant proportion of health records has many events. Long sequences generally induce performance issues for Transformer-like models and can even lead to out-of-memory errors. Some recent approaches focused on dealing with long sequences without sacrificing efficiency. Towards this end, [Chi+19] introduced sparse factorizations of the attention matrix to reduce the self-attention complexity to $\mathcal{O}(n\sqrt{n})$. [KKL20] further reduced the complexity to $\mathcal{O}(n \log(n))$ using locality-sensitive hashing. Recently, [Kat+20] introduced the linear transformer model that reduces complexity to $\mathcal{O}(n)$ by using a kernel-based formulation of self-attention and the associative property of matrix products to calculate the self-attention weight. More precisely, the authors proposed to rewrite Equation (9) as follows:

$$\mathbf{V}_i' = \frac{\phi(\mathbf{Q}_i)^T \sum_{j=i}^{N} \phi(\mathbf{K}_j)\mathbf{V}_j^T}{\phi(\mathbf{Q}_i)^T \sum_{j=i}^{N} \phi(\mathbf{K}_j)},$$

where $\phi(x)$ is a feature map associated to a kernel $k(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$. Note that the feature map $\phi(\cdot)$ is applied row-wise to the matrices $\mathbf{Q}$ and $\mathbf{K}$.

Attention models on a graph. When using *Graph Attention Networks* (GATs) introduced in [Vel+17], each EHR sequence $\mathcal{X}_i$ is represented by a graph $\mathcal{G}_i$ (see Figure 13). The labels $\mathcal{Y}_i$ are not encoded in the graph. Considering an EHR sequence of length L, let us denote visit nodes of the graph $v_{t_0}, \dots, v_{t_L}$, and the event-modality nodes x_{d,m_d} where $d = 1, \dots, D$ indexes the event types and $m_d = 1, \dots, M_d$ their modality. To ease notation, we write d, m instead of d, m_d. Denoting e the edges of the graph,

$$\mathcal{G}_i \in \left\{ \{v\}_t, \{x\}_{d,m}, \{x\}_{e \to v}, \{e\}_{v_t \to v_{t'}, t<t'} \right\}$$

Times differences $\Delta_{t,t'} = t' - t$ can be stored on the edges $\{e\}_{v_t \to v_{t'}}$, while they are set to $\Delta = 0$ for edges $\{x\}_{e \to v}$. The graph is constructed by allowing visit nodes to

[13] A dense feed forward layer applies a linear transformation to the incoming data $h' = hA^\top + b$ followed by a non-linear function such as a sigmoid function

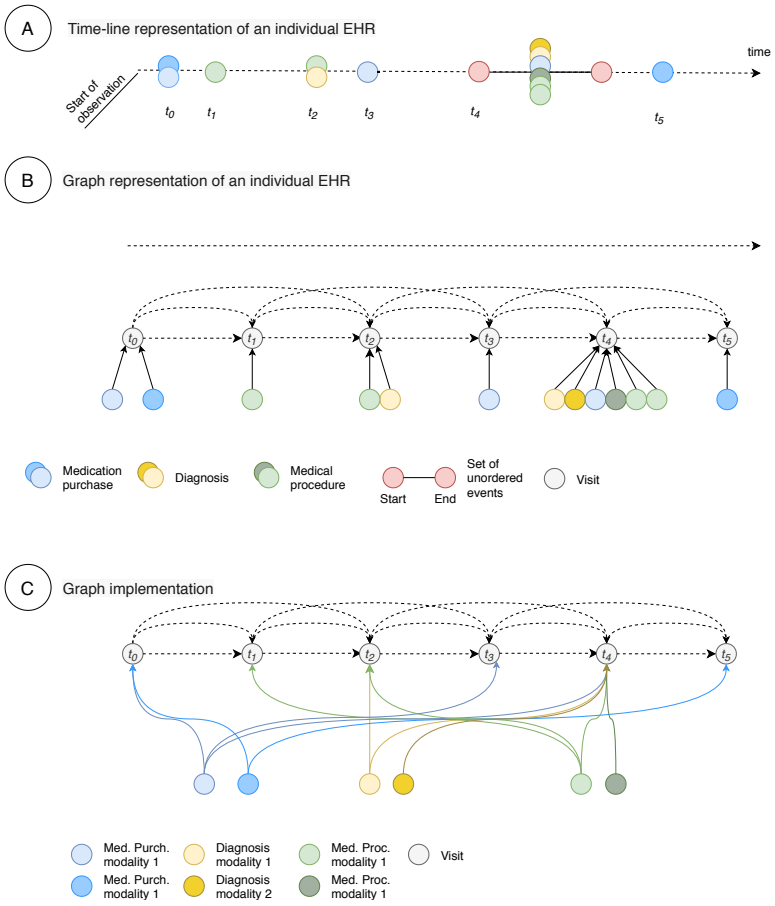


Figure 13 – Graph representation. A timestamped EHR sequence **(A)** can be represented by the graph **(B)**. A visit is created for each timestep in which medical events occur. Event nodes are created each time this event occurs during a visit. Visit nodes are initialized with the sum of the `[visit]` token embedding and the corresponding positional embedding, while event nodes are initialized with the embedding of the corresponding modality. In practice, the graph is implemented as depicted in panel **(C)** to improve the representation sparsity. For a given EHR sequence, event nodes are created only once per observed modality (*event-modality nodes*). Each visit node in which an event occurred can attend to the corresponding event-modality nodes. Note that there is no information flowing *into* these event-modality nodes when updating the graph through the layers. As such, there is no causality break nor data leak when using this representation. Event-modality nodes are *uniquely* learned by the embedding layer.

look back to the k previous visits by adding the edges $\{e\}_{v_t \to v_{t'}, t<t'}$. The model first embeds the nodes x using a dense embedding layer of dimension F

$$\overline{x} = \text{EMBEDDINGLAYER}(x).$$

Nodes v_t are initialized as follows

$$\overline{v}_t = \text{EMBEDDINGLAYER}(v) + \text{POSITIONALENCODING}(t),$$

where EmbeddingLayer(v) is common to all visits. Let us denote the graph with embedded nodes $\mathcal{G}_i^0 = \overline{\mathcal{G}}_i$. This representation is then updated successively by several GATLAYERS [Vel+17].

Let us ignore the distinction between visit and event-modality nodes for a moment, denoting $h = \{h_1, \dots, h_n\}, h_i \in \mathbb{R}^F$ the nodes of the graph. A GATLAYER takes an input graph $\mathcal{G}_i^n$, and produces a new graph $\mathcal{G}_i^{n+1}$ in which only nodes representation $h = \{h'_1, \dots, h'_n\}, h_i \in \mathbb{R}^{F'}$ have been updated while the edges remain unchanged. The attention layer is parametrized by $\mathbf{W} \in \mathbb{R}^{F' \times F}$, $a \in \mathbb{R}^{2F'}$. It writes

$$\alpha_{ij} = \frac{\exp\left(\text{LEAKYRELU}\left(a^\top \left[\mathbf{W}h_i || \mathbf{W}h_j\right]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LEAKYRELU}\left(a^\top \left[\mathbf{W}h_i || \mathbf{W}h_k\right]\right)\right)},$$

where $^\top$ denotes transposition, $||$ concatenation, $\mathcal{N}_i$ the k-order neighborhood of $h_i = \{h_j \in \mathcal{G}_i \ \ s.t. \ \ e_{h_j \to h_i}\}$. LEAKYRELU defines as

$$\text{LEAKYRELU}(x) = \begin{cases} x & if x > 0, \\ ax & otherwise, \end{cases}$$

where a is either fixed to a small value (such as 0.01) or learned by the model. Nodes are updated as

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} h_j\right)$$

where σ is a non-linearity. As in [Vas+17], the attention mechanism can use several attention head. While Self Attention described in [Vas+17] splits the input representation over the dimension F to feed the heads and concatenates the results of each head, GAT heads takes an inputs of size F average their outputs. The GATLAYER applies the update rule defined in Equation (3.1) to each node of the input graph. The embedded graph $\mathcal{G}_i^0$ flows through L GATLAYER [Vel+17].

$$\mathcal{G}_i^l = \text{GATLAYER}_l(\mathcal{G}_i^{l-1}), \qquad l = 1 \dots, L$$

to update the graph representation. Note that all nodes are updated *simultaneously*. Besides, as there are no edges directed towards nodes $\overline{x}$, the representation of event-modality nodes is not updated. *This could be done by adding edges* $e_{x_{d,m} \to x_{d,m}}$. The

final graph representation $\mathcal{G}_i^n$ can then be used to perform tasks such as node classification (LOS task on visit nodes) or graph classification (IHM task by pooling visit nodes).

3.2 Pre-training strategies.

Masked Language Model. Masked Language Model (MLM) has been introduced in [Dev+18] to learn language representations. It introduces two pretext tasks. The first one consists in selecting 15% of the sequence tokens randomly. They are modified as follows: 80% of these tokens are replaced by the `[MASK]` token, 10% of them are replaced by a random code, and another 10% remain unchanged. The pretext task consists in predicting the true token given the rest of the sequence. The second task consists in predicting the next sentence, given the current one. While the first pretext task can adapt to EHRs, the second one does not as the "sentence" concept is hard to define for healthcare data.

Triplet loss. Triplet loss training consists of predicting if several sub-sequences belong to a given sequence or not. The triplet loss was shown to produce good time series representations by employing an unsupervised causal model [FDJ19]. The sampling algorithm extracts random sub-sequences x^{ref} and x^{pos} (a positive example) of a given sequence y_i, and samples K of x^{neg} (negative examples) that are chosen at random in different random time series y_j with $j \neq i$. Then, on the one hand, the representation of x^{ref} should be close to that of x^{pos}, while on the other hand, the representation of x^{ref} should be distant from the ones of x^{neg}. This leads to the minimization of the triplet loss, given by

$$\mathcal{L}_{\text{triplet}} = -\log(\sigma(f(x^{\text{ref}}, \theta)^\intercal f(x^{\text{pos}}, \theta))) - \sum_{k=1}^{K} \log(\sigma(-f(x^{\text{ref}}, \theta)^\intercal f(x_k^{\text{neg}}, \theta))),$$

where σ is the sigmoid function and $f(\cdot, \theta)$ is a deep neural network encoder, where the parameters θ are to be trained.

Contrastive Predictive Coding. Contrastive Predictive Coding (CPC), as formulated in [OLV18], learns representations by training neural networks to predict the representations of "future" observations from those of "past" ones. The main idea of CPC consists of maximizing the mutual information between the encoded representations and not between the original labeled data. When predicting future information the authors propose to encode the target x (future) and context c (present) into a compact distributed vector representations via non-linear learned mappings. These

representations are designed to maximally preserve the *mutual information* (MI) of the original signals x and c defined as

$$I(x;c) = \sum_{x,c} p(x,c) \log \frac{p(x|c)}{p(x)}.$$

By maximizing the mutual information between the encoded representations (which is bounded by the MI between the input signals), CPC extracts the underlying latent variables that inputs have in common. The architecture of CPC is as follows: first, a non-linear encoder f_θ maps the input sequence of observations x_i to a sequence of latent representations $z_i = f_\theta(x_i)$. Next, an auto-regressive model g_{ar} summarizes all $z \leq t$ in the latent space and produces a context latent representation $c_{i,t} = g_{\text{ar}}(z_i \leq t)$. The authors model a density ratio which preserves the mutual information between $x_{i,t+k}$ and $c_{i,t}$ as follows:

$$f_k(x_{i,t+k}, c_{i,t}) = \exp\left(z_{i,t+k}^T W_k c_{i,t}\right).$$

Both the encoder and auto-regressive models are trained to jointly optimize a loss based on *Noise Contrastive Estimation* (NCE), which is called InfoNCE. Given a set $X = \{x_1, \dots, x_N\}$ of N random samples containing one positive sample from $p(x_{t+k}|c_t)$ and $N-1$ negative samples from the distribution $p(x_{t+k})$, we optimize

$$\mathcal{L} = -\mathbb{E}\Big[\log \frac{f_k(x_{i,t+k}, c_{i,t})}{\sum_{x_j \in X} f_k(x_j, c_t)}\Big].$$

This loss function encourages the prediction $\hat{z}_{i+k}$ to be most similar to the one positive sample z_{i+k} among a set of randomly selected negative samples z_l [Hén+19]:

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{i,k} \log\Big(\frac{\exp(\hat{z}_{i+k}^T z_{i+k})}{\exp(\hat{z}_{i+k}^T z_{i+k}) + \sum_l \exp(\hat{z}_{i+k}^T z_l')}\Big).$$

Related works.

As explained previously, adapting state-of-the-art architectures for NLP to structured EHR is a non-trivial task. Only a few relevant papers can be found in the literature on this topic. BEHRT [Li+20] develops pre-trained models to predict the occurrence of any disease in future visits. It uses positional embeddings to distinguish different visits and adds an age layer to imply temporal orders. However, BEHRT only uses disease sequences besides basic demographic information, discarding other medical information such as lab exams and drug consumption, which might hinder its reuse for other tasks. G-BERT [Sha+19] adapts MLM pre-training to align disease and drug representations within a single visit to predict medications from diseases and

conversely. However, they discard order and temporal information in the process, making this approach unusable to perform forecasting tasks. Med-BERT [Ras+20] adapts BERT for pre-training contextualized embedding models on a larger cohort and longer visit sequences compared to BEHRT and G-BERT. Interestingly, this paper introduces the pretext task of prolonged length of stay in hospital (LOS) and fine-tunes the model on two tasks concerning disease-prediction. However, Med-BERT only exploits diagnosis information and does not include the elapsed time between visits, leading to a significant loss of information.

Graph Convolutional Transformer introduced in [Cho+20] incorporates a self-attention mechanism. Medical visits are represented as graphs, of which edges are estimated by using self-attention. Self-attention is constrained to enforce specific chains of events such as observed symptoms cause diagnoses and diagnoses cause prescription. The representations of visits are computed with convolutional graph networks over the estimated graphs. This approach supposes to have access to fine-grained information in the dataset. However, data such as symptoms are not often available in EHRs.

3.3 Contribution: comparison of multiple attention models and pre-training strategies

This work brings new contributions with the evaluation of several transformer architectures combined with several unsupervised pre-training strategies for structured EHR. Pre-trained representations are fine-tuned with a single additional output layer for the considered specific downstream task. The experiments were performed using the freely accessible MIMIC-III database [Joh+16], that is featured in numerous publications, see [Har+19; Shi+18; Son+18] among many others. Experiments conducted in this work use current best practices for hyperparameters tuning (see Section IV.2.4).

Method

Apart from patient demographics (e.g., age, gender) and some other static features, a structured EHR consists, for each patient, of a sequence of medical events, such as a diagnosis, medication codes, or medical acts, for example.[14] Each event is timestamped with a precision called *time unit*, that depends on the database. Since the time unit is generally relatively large (an hour or a day), many events are co-occurring, i.e. share the same timestamp.

[14]Medical concepts used in EHR and associated codes usually come from pre-defined standards, such as the International Classification of Diseases (ICD)

Timeline versus graph representation. A patient EHR with several types of events is illustrated in Figure 13A using a timeline representation. Another representation illustrated in Figure 13C uses a directed graph representation, where successive nodes representing time units replace the timeline. Another set of nodes represent the events happening during a time unit. The edges correspond to existing structural associations between events such as next time unit event on the timeline, medical events associated with the same time unit, diagnosis (or treatment) events associated with a symptom event. Such a representation was introduced by [Cho+18] and inspired other works, such as [Cho+20] and [Het+19]. The graph representation used in this work is similar to that of [Het+19], but sequences of events are modeled through temporal point processes therein, while we use deep attention models. The choice of the representation is driven by the architecture used, as explained below.

Models architecture. The models considered in this work all share the same neural network architecture illustrated in Figure 14. Following the data flow, the architecture contains four components: an embedding component, an encoder, a pooler, and a final dense layer for operating a given task. We use a two-step training strategy: the encoder is first pre-trained in an unsupervised way using some pretext task, and the final dense layer is fine-tuned on some specific clinical supervised task.

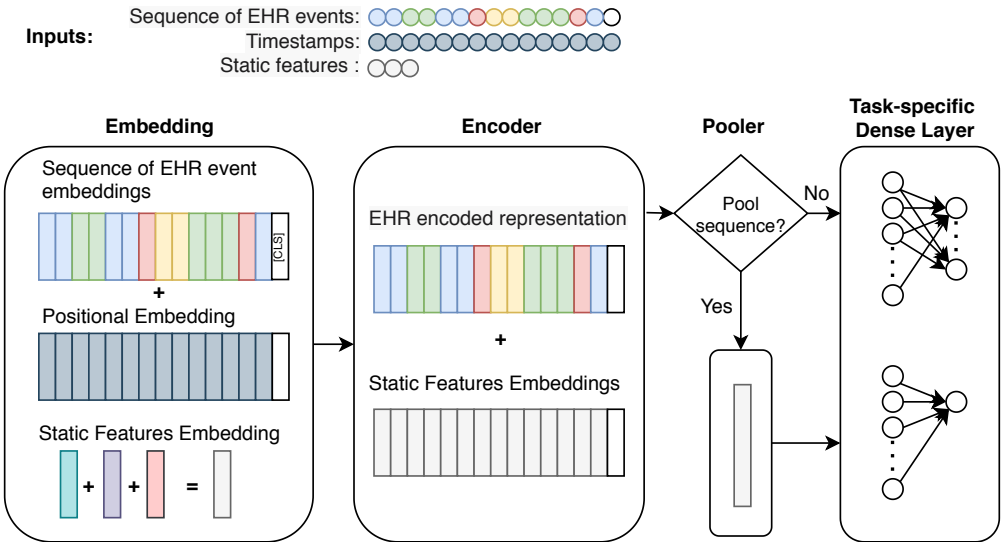


Figure 14 – Overview of the generic model architecture. The EHR representation (see Figure 13) is used as input data to an architecture with four components: an Embedding component, a attention-based encoder, a Pooler and a final dense layer for operating a given pre-training or downstream task.

Embedding component. Each event from the EHR representation corresponds to a code and/or numeric variables (see Figure 13A). These codes/variables are tokenized and embedded: each unique token is individually mapped to a low-dimensional embedding vector. The relative position (timestamp) of each event is encoded using fixed positional embeddings (*added* to each event embedding) following [Vas+17], in which the ordered position number is replaced by the elapsed time relative to the timestamp of the first event of the sequence. Finally, a unique token `[CLS]` is added at the end of the sequence and is embedded. Moreover, static features (including patient demographics, e.g. age, and gender) are also embedded and summed out into a single vector used as input to the encoder component, see Figure 14.

Encoder. An encoder is used to encode the whole EHR representation into a new sequence with the same length. The static features embedding is then added to each element of this new sequence. This work compares the following attention-based encoders:

(i) The *Vanilla Transformer* [Vas+17], which allows building a representation of an input sequence by stacking Multi-head Self-Attention (MSA) layers;

(ii) The *Linear Transformer* [Wu+20], which significantly reduces the memory footprint and scales linearly with respect to the sequence length compared to [Vas+17], allowing to feed entire sequences without length restrictions;

(iii) The *Graph Attention Network* (GAT) [Vel+17; Ye+19], which uses fully connected graphs with an self-attention mechanism which does not involves queries and keys as MSA. For this encoder, we use the graph representation described in Section 3.3 and Figure 13 C.

In each case, *causal* attentions are used: at any given position, the attention mechanism is not allowed to put attention on any data involving *future* positions, so that the output sequence of the encoder preserves causality.

Pooler. As explained in Section 3.3 below, two types of downstream supervised tasks can be considered: (i) tasks that operate on each event embedding of the sequence coming out of the encoder (e.g. length of stay prediction) and (ii) tasks that exploit the overall sequence (e.g. mortality prediction). A pooler is required only for (ii). We follow [Dev+18] where only the last element of the encoded sequence is kept (which explains the use of the `[CLS]` token above).

Unsupervised pre-training

As summarized in Figure 15, the unsupervised pre-training strategies described in Section 3.2 were considered:

(i) *Masked Language Modeling* (MLM) [Dev+18],

(ii) *Triplet Loss* [FDJ19],

(iii) *Contrastive Predictive Coding* (CPC or InfoNCE) [Che+20; Sun+19].

All architectures were trained from scratch and independently for each pre-training strategy.

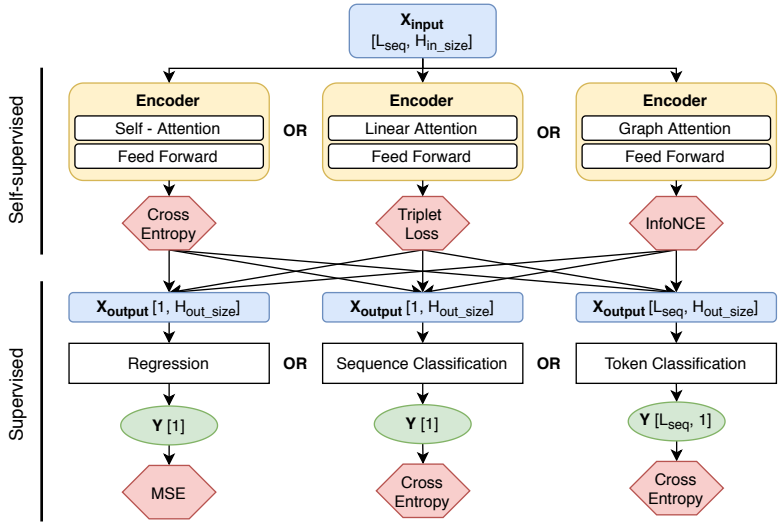


Figure 15 – Overview of the unsupervised pre-training and evaluation procedures. First, three encoders are pre-trained separately in an unsupervised manner. The obtained representation for each token in the sequence is then passed into the Pooler if the downstream supervised task requires it. In the next step, we add a classifier network on the top of the encoder, which is trained in a supervised manner separately for each downstream task.

Supervised fine-tuning, losses and metrics

All the combinations of encoder architectures and pre-training strategies are assessed using several clinical *supervised tasks* ([Har+19], see Section 3.4 below). For each model combinations and supervised tasks, the following training strategies are compared:

(i) Fine-tuning, using the supervised task, of the whole architecture, with the embedding component and encoder initialized with the weights learned during the pre-training phase;

(ii) fine-tuning of the final dense layer only (the embedding component and encoder weights are fixed to the pre-trained values);

(iii) end-to-end supervised training of the whole architecture, with random initialization of all the weights.

Depending on the supervised tasks, the following standard losses and assessment metrics were used: cross-entropy loss for binary and multi-class classification, with AUROC and AUPRC metrics for assessing binary classification tasks and Cohen's linear weighted kappa metric for multi-class classification. hyperparameters and training details are thoroughly described in Section IV.2.4 of Chapter IV.

3.4 Experiments

The experiments were conducted on MIMIC-III data. MIMIC-III (Medical Information Mart for Intensive Care) is a single-center database containing de-identified data about patients admitted to intensive care units (ICU) [Joh+16] between 2001 and 2012. Following [Har+19; Son+18], the experiments used a cohort of 33,798 unique patients with a total of 42,276 ICU stays. Population selection, features, and labels were generated following [Har+19] (see Table IV.3.1 in Chapter IV). Training, validation, and testing sets are respectively 70%, 15%, and 15% of the ICU stays, reusing the same sampling as [Har+19]. ICU stays with less than five events are excluded.

Three clinical prediction tasks[15] from [Har+19] were considered:

(i) *In-Hospital Mortality* (IHM): the outcome is a binary variable indicating wheter a patient dies during a given ICU stay or not. It is treated as a binary classification problem. True mortality labels are curated by comparing the times of death, hospital admission, and discharge. The mortality rate within the cohort is 13%.

(ii) *Length-of-Stay* (LOS): the outcome is the remaining time spent in ICU. It is bucketized into ten buckets ($\leq$ 1 day; 1; 2; ... ; 7 days, $[1, 2)$ weeks; $\geq$ 2 weeks) and is considered as a 10-class classification problem.

(iii) *Phenotyping* (PHE): the outcome is a category corresponding to one of 25 diseases. It is treated as a classification problem and called *acute care phenotyping*. The disease is predicted retrospectively from data about the ICU stay of a patient. The data contains 25 diseases, including 12 critical respiratory/renal

[15]We do not use the *decompensation* task, since it is highly correlated with IHM and leads to a highly unbalanced binary classification problem that does not provide more insights than the ones considered here.

failures, 8 chronic conditions such as diabetes or atherosclerosis, and 5 "mixed" conditions such as liver infections. Patients with multiple phenotypes are excluded.

Results

Combinations of the architectures and pre-training strategies detailed in Section 3.3 were compared using supervised tasks related to clinical prediction tasks (Section 3.4) on MIMIC-III. The corresponding metrics, computed on the test set, are reported in Table 1.

Table 1 – Test metrics obtained by all combinations of architectures and pre-training strategies (rows) on clinical prediction tasks (columns) using the MIMIC dataset.. Due to its underwhelming performances, GAT was not trained for all the tasks and training strategies to avoid computation waste. * The Length Of Stay (LOS) task in [Har+19] slightly differ from ours. They predict the remaining LOS at each hour, while our experiments do so each time there is a new patient measurement. Thus, performance comparison cannot be made directly between these two approaches.

Encoder	In-hospital mortality AUPRC/AUROC	Length of Stay Kappa	Phenotyping AUROC
End-to-end supervised			
Multi-task LSTM [Har+19]	**0.533/0.870**	**0.450***	**0.774**
Vanilla Transformer	0.394/0.809	0.535	0.736
Linear Transformer	0.355/0.790	**0.584**	0.676
GAT	0.132/0.528	0.218	0.503
MLM Pre-training			
Vanilla Transformer	**0.409/0.817**	**0.554**	**0.749**
Linear Transformer	0.344/0.785	0.405	0.708
GAT	0.154/0.572	–	–
Triplet Loss Pre-training			
Vanilla Transformer	**0.357/0.781**	0.451	**0.729**
Linear Transformer	0.330/0.774	**0.577**	0.686
CPC Pre-training			
Vanilla Transformer	**0.391/0.805**	0.466	**0.741**
Linear Transformer	0.333/0.770	**0.521**	0.675

According to this table, we first note that GAT shows inferior performance on all tasks, using pre-training or not. Increasing k, the number of past visits GAT could attend to did not result in any improvement. As aggregating events into visits according to a similar graph structure resulted in good representation in [Cho+20], this poor performance might be rooted in the attention mechanism. Indeed, GAT

uses an attention formulation relying only on node similarity rather than the query, key, values mechanism used in MSA. Besides the performance aspect, the graph formulation was very effective in GPU memory usage, allowing to process the longest sequences and use larger mini-batches. Moreover, attention on a graph is easy to implement since no ad-hoc masking is required to enforce causality, and it easily handles sequences of varying lengths. Blending this approach with an attention mechanism closer to MSA could be an exciting extension, as an analoguous approach resulted in promising results in NLP [Ye+19].

As explained in Section 3.1, the vanilla transformer could not handle long sequences under the memory constraints of a few GPUs due to its quadratic complexity in the sequences' length. In our experiences, limiting the length of the sequences seemed to hinder its performance. The linear transformer could handle longer sequences, but it did not result in performance improvements over standard MSA.

Fine-tuning with frozen encoder weights led to worse results than fine-tuning with unfrozen weights and are not reported here. We observed that it took only 5 to 15 epochs, depending on the task, to achieve good performances when performing fine-tuning with unfrozen encoder weights. The training for each architecture, pre-training strategy and prediction took less than 5 hours, except MLM, for which training could last up to two days. We observed that MLM improved the scores of end-to-end supervised Vanilla Transformer, while Triplet Loss and CPC pre-training led to minor improvements. Regarding triplet loss, the random sampling of triplets $x^{\text{ref}}, x^{\text{pos}}, x^{\text{neg}}$ might be an issue. Indeed, even simple models can quickly learn to choose between x^{pos} and x^{neg} when they are chosen at random. In this case, the average triplet loss quickly converges towards zero, resulting in very slow parameter updates [Wu+17]. Adapting the sampling strategy to EHR data could be a way of improving results on triplet loss.

Contrastive pre-training might not have revealed all of its capabilities in our experiments since it was understood only recently for computer vision problems [Che+20] that data-augmentation is a crucial ingredient in such unsupervised strategies. Building pertinent data-augmentation on EHR data remains, to the best of our knowledge, a fascinating open question that requires to be addressed by future works since it would, in our opinion, enable important advances in learning representations for health pathways in an unsupervised fashion.

Finally, this work was kept general enough to be used with claims data in place of EHR data. While many publications feature deep learning models on EHR data, only [Kab+19] uses such models on claims data. Claims databases such as SNDS are likely to benefit the most from pre-training because of its scale and exhaustivity. Pursuing this work's development on such a database could eventually lead to significant innovations in public health.

CHAPTER I

SCALPEL3: A SCALABLE OPEN-SOURCE LIBRARY FOR HEALTHCARE CLAIMS DATABASES

Objective: This article introduces SCALPEL3 (SCAlable Pipeline for hEaLth data), a scalable open-source framework for studies involving Large Observational Databases (LODs). It focuses on scalable medical concept extraction, easy interactive analysis, and helpers for data flow analysis to accelerate studies performed on LODs.

Materials and methods: Inspired from web analytics, SCALPEL3 rely on distributed computing, data denormalization and columnar storage. It was compared to the existing SAS-Oracle infrastructure by performing several queries on a dataset containing a three years-long history of healthcare claims of 13.7 million patients.

Results and Discussion: SCALPEL3 horizontal scalability allows handling large tasks quicker than the existing infrastructure while it has comparable performance when using only a few executors. SCALPEL3 provides a sharp interactive control of data processing through legible code, which helps to build studies with full reproducibility, leading to improved maintainability and audit of studies performed on LODs.

Conclusion: SCALPEL3 makes studies based on *Système National des Données de Santé* (SNDS) much easier and more scalable than the existing framework [Tup+17b]. It is now used at the agency collecting SNDS data, at the French Ministry of Health and soon at the National Health Data Hub in France [Cug+18].

Keywords: *Large observational database, Healthcare claims data, ETL, Scalability, Reproducibility, Interactive data manipulation*

I.1 Introduction

In the past decade, the volume of healthcare data and its accessibility rose quickly. For instance, in France, the SNDS claims database contained 86% of the French population in 2010 [Tup+10a] to reach 98.8% in 2015 [Tup+17b] leading to one of the world's largest health Large Observational Database (LOD) [Bez+17; Tup+17b]. The exhaustivity of LODs such as *Système National des Données de Santé* (SNDS) has proven useful for public health research, by improving the statistical power of algorithms using this data and by mitigating the sensitivity to selection biases [Tup+17b].

However, such an abundance of data comes at a cost: SNDS is a very complex database, with data spread across hundreds of tables and columns. Its scale makes data manipulation non-trivial. More importantly, using this data requires a tremendous amount of knowledge from SNDS experts. Many coding or data recording subtleties, such as data duplication caused by administrative complexity, might bewilder inexperienced users. Deriving proper health events definitions and extracting them accurately is, therefore, a difficult task, having important consequences on the derived studies [Han+13; Tup+17b]. These issues are of course not unique to SNDS but shared by many LODs [Mad+14].

This paper proposes an answer to this problem by introducing SCALPEL3 (SCALable Pipeline for hEaLth data), an open-source framework intending to reduce such entry barriers to LODs. This framework attempts to simplify medical concept extraction by providing a set of tools performing batch Extract-Transform-Load (ETL) tasks, while an interactive API eases the manipulation and the exploration of longitudinal cohorts. Thus, this research focuses on the following objectives:

(i) Design and implement a scalable tool allowing to extract and manipulate longitudinal patient data from large observational databases;

(ii) Simplify methodological research by reducing SNDS data complexity and by easing data loading into formats used by common machine learning libraries;

(iii) Foster reproducibility by monitoring the data flow and by following best practices for clean code;

(iv) Promote reusability and extensibility by documenting and publishing an open-source implementation of SCALPEL3.

The main concepts used by SCALPEL3 and some related works are presented in Section I.2. The LOD for which SCALPEL3 was initially designed for is described in Section I.3, together with SCALPEL3 methods and abstractions. The scalability of SCALPEL3 is evaluated in Section I.4, while Section I.5 discusses its strengths and limitations.

I.2 Background

LODs are not designed to perform medical research. Electronic Health Records (EHR) data directly supports clinical care and are used to justify care billing and reimbursement, while claims data are primarily used for reimbursement purpose. The data models and terminologies used in such databases were optimized to suit these particular goals, resulting in normalized data models built around hospital stays, transactions, or cash flows [Tup+17b]. Extracting meaningful patients care pathways from such data can be decomposed into two tasks. First, all the data corresponding to a set of patients need to be identified and collected. When the data is not normalized around the patients, this task requires several join operations which can be very costly in terms of computations as the data volume increases. Second, medical concepts have to be properly identified from administrative codes: this *phenotyping* task relies heavily on a combination of medical and database knowledge. The algorithms used to perform concept extraction from administrative data are either disclosed through scientific publications or shared as lengthy SQL queries [Loo19]. Their code or the description of the algorithms involved can vary in quality, hindering reuse, and reproducibility. As a result, building a study from scratch might be faster than reusing poorly documented code from previous works [Loo19; PDZ06]. Besides, access to LODs such as SNDS might rely on proprietary software such as SAS [Sup76] or SPSS [IBM68]. While these tools are suitable to produce public health studies, they hinder methodological research as they do not interact easily with R or Python packages that implement state-of-the-art machine learning algorithms. All of these challenges are complex to solve and exacerbated by the data volume at hand.

Related works

Several research programs produced tools in order to alleviate some of these issues. An important research effort aims at easing data integration and interoperability by producing standard data models and terminologies to be shared across institutions. Observational Medical Outcomes Partnership Common Data Model (OMOP CDM), which is supported by the Observational Health Data Sciences and Informatics (OHDSI) research program [Hri+15], and the Informatics for Integrating Biology & the Bedside (i2b2) data model [Mur+10], can be considered as the most pervasive data models developed for this purpose. OMOP CDM can be used to standardize EHR or claims data, while i2b2 is focused on EHR data.

Both models are centered around the patients, thus reducing the number of join operations required to access a specific patient history. They also rely on a normalized data model combined with SQL databases. A collection of open-source software has been developed on top of these models, implementing analytics or visualization tools [Hus+16]. These softwares can take the form of R libraries [Hus+16], or

compiled Java [SM14] programs with a graphical user interface. While making these softwares freely available is an important step to foster methodological research, they do not seem to be easily extensible or interoperable as they do not provide documented APIs to build new software upon it. Besides, the process of transforming an existing database in order to conform to such standards is costly, as it requires to build complex mappings between shared representations expressed through highly heterogeneous codes from one information system to the other. In the case of the SNDS database considered in this work, such a mapping is still work in progress [Dou+20].

In other fields, web-scale analytics have shifted from the use of normalized SQL databases towards NoSQL technologies relying on distributed computing, denormalization, and columnar storage. The use of distributed computing allowed gains in computational power using low cost, commodity servers instead of expensive dedicated hardware [Bon+17]. A work from OHDSI [Pow16] compared the ACHILLES software (R [Cor17], PostgreSQL [Pos96]) with Apache Spark [Zah+16] using common SQL requests. They observed performance gains for Spark even on a single server or small clusters, at the exception of requests leading to large network I/O, since such operations are known to be the slowest operations in a distributed computing framework because of network latencies and throughput. It can create bottlenecks when many data chunks are sent across the servers in the cluster to perform a join or a groupby operation (leading to so-called *shuffles*). Denormalization can be a way to circumvent this issue by performing a set of join operations beforehand, once and for all [Deh+15; LP14; Wei+08], reducing join operations to simple look-ups over a very large table. The data duplication resulting from such joins operations might lead to storage issues, which can be mitigated with the help of columnar storage formats [LP14; Mel+10] using compression strategies.

To the best of our knowledge, such an approach has not been implemented to perform ETL on large health databases. Prior works are either relying on SQL and normalized schemas [Jan+17; Ong+17] or applied to small datasets [Har+18]. This paper describes and implements such an approach for large health databases, as explained in the next section.

I.3 Material and Methods

This work focuses on (i) denormalizing the data in combination with columnar storage and distributed computing to perform concept extraction, (ii) providing a structured and re-usable concept library, and (iii) introduce useful abstractions to handle cohort data. Scalability issues are handled by (i), while (ii) and (iii) foster the reuse of code and knowledge across studies. This is achieved by reducing both study-specific code and database entry barriers by providing ready-to-use concepts. SCALPEL3 provides Scala [Ode+04] and Python APIs to ensure easy extension and

interoperability with numerous libraries. All the code supporting this paper is open source and freely available.

This paper is not about data integration from disparate sources, such as multiple EHR systems, but rather about an ETL based on batch distributed processing of a large, centralized claims database.

I.3.1 The SNDS database

This work was performed using the *Système National des Données de Santé* (SNDS), a large claims database containing pseudonymized data on 98.8% of the French population (66 million patients in 2015) [Bez+17; Tup+17b]. It contains time-stamped information about medical events leading to reimbursement (see Table 1 in [Tup+17b] for an exhaustive list of available data) in the last 3 years[1]. It contains more than 20 billion health events per year, representing roughly 70TB of data.

SNDS is composed of multiple "sub-databases", each one with a star schema. The central table records events leading to cash flows that need to be joined to many other tables to access medical information[2]. In this form, retrieving patient information for statistical studies is very costly in terms of computation and expert knowledge: targeted data can be spread across multiple databases, tens of tables, and hundreds of columns, and its identification requires a deep administrative knowledge of the French health-care reimbursement mechanisms. Mitigating these issues is precisely the motivation of the SCALPEL3 framework.

I.3.2 SCALPEL3: a SCAlable Pipeline for hEaLth data

SCALPEL3 is based on Apache Spark [Zah+16], a robust and widely adopted distributed in-memory computation framework. Spark provides a powerful SQL-like high-level API and a more granular API to perform data operations. It can be coupled with the Hadoop File System (HDFS) [Shv+10] replication system to accelerate large files reading and distribution over a computing cluster. SCALPEL3 is an open-source framework organized in the following three components.

SCALPEL-Flattening [Kum+19] denormalizes the data "once and for all" to avoid joining many tables each time the data of a patient is accessed. Its input is a set of CSV files extracted from the original SNDS database.

SCALPEL-Extraction [Pau+19] defines concepts extractors that process the denormalized data and transformers, that compute more complex events based on

[1] which can be extended up to 20 years under some restrictions.

[2] We work with two main sub-databases containing data relevant for public-health research. When working on drug safety studies, each of these two databases contains 8 relevant tables, representing approximately 5 billion lines per year when restricted to 65+ y.o. subjects.

extractors output. For example, extractors can fetch all drug dispenses or medical acts.

SCALPEL-Analysis [Seb+19] implements powerful and scalable abstractions that can be used for data analysis, such as easy ways to investigate data quality issues. It can load data into formats commonly used in machine learning, such as TensorFlow or PyTorch tensors or NumPy arrays.

As SCALPEL-Flattening and SCALPEL-Extraction perform batch operations, they need to read (resp. write) input (resp. output) data from the file-system (local or HDFS). They are implemented in Scala in order to access Spark's low-level API and take advantage of functional programming and static typing, resulting in rigorous automated testing (94% of the Scala code is covered by unit tests). Both can be configured through textual configuration files or be used as libraries. SCALPEL-Analysis is a python module implemented in Python & PySpark and designed for interactive use. It can be used in a Jupyter notebook [Klu+16] for instance. This workflow is illustrated in Figure I.1.

I.3.3 SCALPEL-Flattening: denormalization of the data

As mentioned earlier, performing data analysis on SNDS patients' health requires many joins and can consequently be extremely slow. To circumvent this issue, the data are denormalized by joining the tables sequentially to obtain a big table in which each line corresponds to a patient identifier and a wide representation of an event.

Denormalizing a star-schema database results in a really big table due to values replications. To circumvent storage and computation issues, the denormalized data is stored in Parquet [Voh16] files, an open-source columnar storage format implementing Google's Dremel [Mel+10] data model. Parquet is well-integrated in the Spark ecosystem [Arm+15], allowing us to take advantage of the columnar storage in terms of data compression and query optimization. SCALPEL-Flattening first converts the input CSV files containing exports of SNDS tables to Parquet files. Then, it recursively performs left joins with these tables, starting with the central table. Finally, it writes the results in a single Parquet file. To ensure the scalability of these big join operations, the input data can be automatically divided with respect to some time unit (such as years, months) before performing the join operations. In this case, the joins results are sequentially appended to the output parquet file. These operations are repeated for each SNDS sub-databases. The size of the temporal slicing used in the joins, the schema, and the joining keys can be tuned by the end-user through a configuration file, which defaults to the denormalization of tables containing only medical data (as opposed to econometric and administrative data). A set of statistics that monitors the denormalization process is automatically computed along the steps involved in it, in order to ensure that no loss of information occurs.

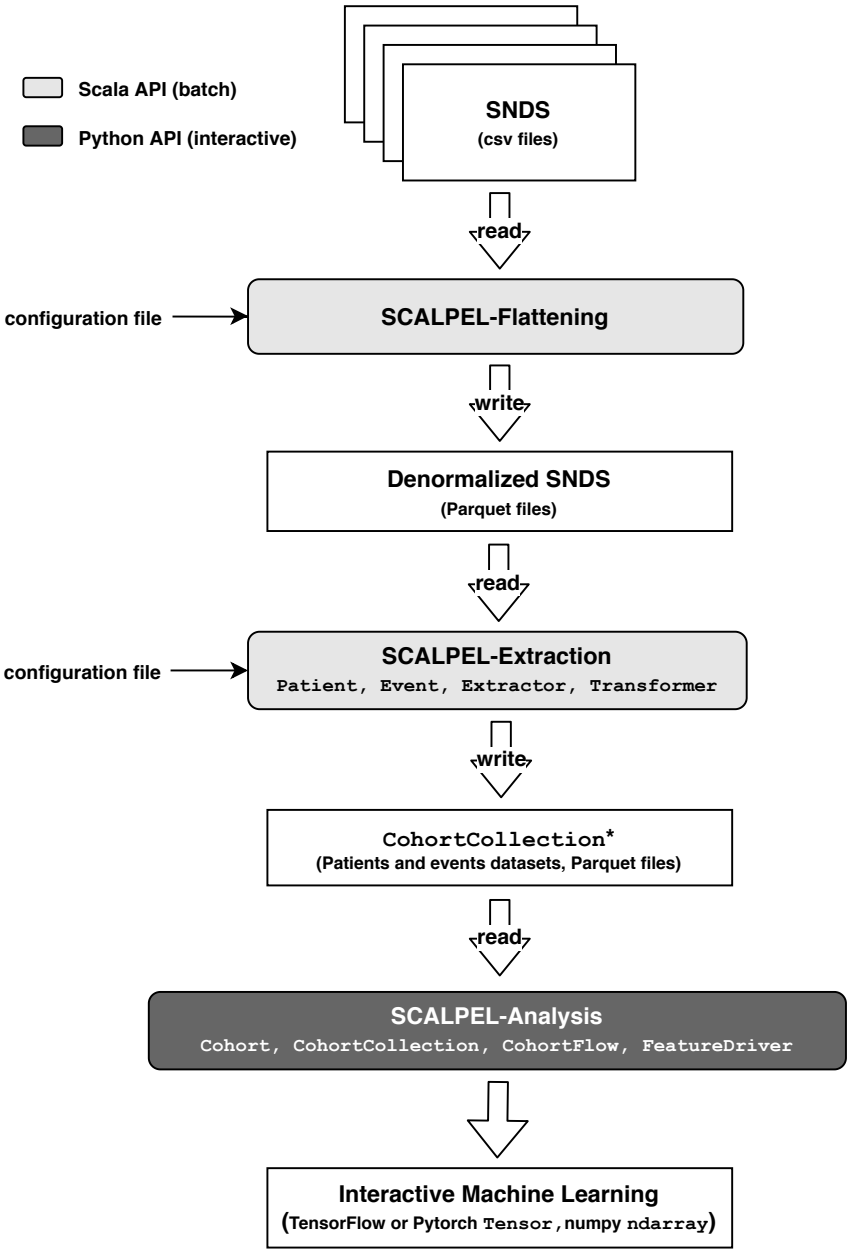


Figure I.1 – SCALPEL3 workflow. SCALPEL3 is made of three independent open-source libraries plugged one after another. SCALPEL-Flattening, which is implemented in Scala & Spark, denormalizes the input database exported as CSV or Parquet files into a single big flat database. Then, SCALPEL-Extraction, implemented in Scala & Spark, extracts concepts from this flat database. Finally, SCALPEL-Analysis, implemented in Python & PySpark loads extracted concepts to perform in-memory interactive analysis and feed machine learning algorithms.

I.3.4 SCALPEL-Extraction: extraction of concepts

SCALPEL-Extraction provides fast extractions of medical concepts from the denormalized tables produced by SCALPEL-Flattening. By providing ready-to-use medical events, SCALPEL-extraction encapsulates SNDS technical knowledge but keeps medical data as raw as possible, so that end-users have access to fine-grained data which is critical when designing observational studies [Hon+18; Wan+16]. The extracted concepts are organized around two abstractions: `Patient` and `Event`.

The `Patient` abstraction has a unique `patientID`, a `gender`, a `birthDate` and eventually a `deathDate`.

The `Event` abstraction allows to represent any event associated to a patient. It can be punctual (e.g., medical act) or continuous (e.g., hospitalization).

All concepts are automatically extracted into `Patient` or `Event` objects by a set of `Extractors` and `Transformers`, designed to fetch the data in the relevant tables and columns of the SNDS `Sources`.

The `Extractor` abstraction maps a `Row` of a `Source` to zero or many `Events`:

$$\texttt{Extractor}\colon \texttt{Row} \mapsto \texttt{List[Event]}.$$

`Extractors` successively refines data from the input (wide denormalized tables) by (1) identifying the relevant columns, (2) filtering out null values according to some columns and (3) conform the extracted data to a standardized schema. These three operations are very fast when performed on columnar data, as they exploit sparsity (null values are not represented in the data) and consist in simple look-ups over hash tables containing columns metadata. An optional step that filters rows by value can occur before step (3). This operation is slower as it manipulates row values, but since it is performed near the end of the extraction process, it typically occurs on small data. This process is illustrated Figure I.2.

Many extractors are available to fetch medical acts, diagnoses, hospital stays, among others, an example being the drug dispense `Extractor` which allows extracting events related to specific subsets of drugs and to output events at multiple levels of granularity (drug, molecule, ATC class, custom classes) as defined in a configuration file. This simple architecture makes it easy to add new `Extractors` and to answer to any extraction need.

The `Transformer` abstraction transforms a collection of `Events` related to a unique `Patient` into a list of more complex `Events` (complex diseases, drug exposures, ...):

$$\texttt{Transformer}\colon \texttt{List[Event]} \mapsto \texttt{List[Event]}.$$

A `Transformer` is based on specific algorithms requiring multidisciplinary knowledge from epidemiologists, statisticians, physicians, and SNDS experts [Tup+17b].

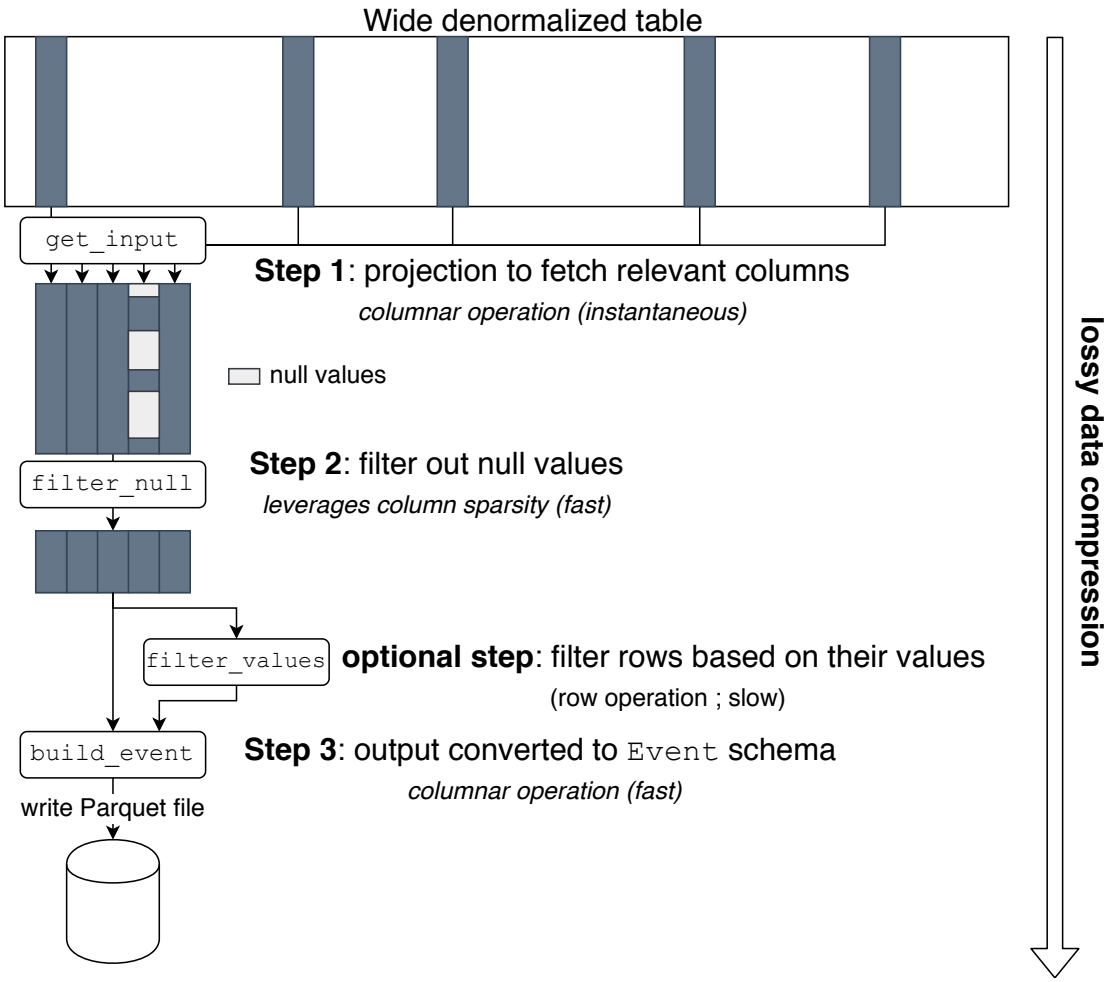


Figure I.2 – `Extractor` design. `Extractors` implemented in SCALPEL-Extraction successively refines the input table (a large denormalized table) by taking advantage of fast columnar operations to produce ready-to-use medical events. Step 1 selects the relevant columns (equivalent to a hash table look-up) while Step 2 removes rows where null values are detected in specific columns, taking advantage of the sparsity of columnar representation (null values are not encoded in the data). Optionally, this extraction process filters out rows based on their values. Finally, Step 3 conforms the data to the `Event` schema, and is written to a Parquet file.

`Transformers` usually combine events built by `Extractors` to build more complex events, such as computing drug exposures from timestamped drug dispenses. `Extractors` and `Transformers` can be used through a Scala API or controlled using a textual configuration file. Many `Transformers` used in several studies such as [Mor+20; Neu+12] are implemented and ready to use.

Besides Parquet files containing extracted events, SCALPEL-Extraction outputs metadata tracking the data used to build each type of extracted events. This file can be leveraged by SCALPEL-Analysis to build `Cohorts` and flowcharts, as explained below.

I.3.5 SCALPEL-Analysis: interactive manipulation and analysis of cohorts

While SCALPEL-Flattening and SCALPEL-Extraction are implemented in Scala & Spark for performance and maintainability, SCALPEL-Analysis is implemented in Python & PySpark [Zah+16] since it is designed for interactive environments, such as Jupyter notebooks [Klu+16]. SCALPEL-Analysis eases the manipulation and analysis of cohort data. It is based on the following abstractions:

The `Cohort` abstraction is a set of `Patients` and their associated `Events` in a time-window [`startDate`, `endDate`]. Basic operations such as union, intersection, and difference can be performed between `Cohorts`, while a human-readable description is automatically updated in the results. More granular control is kept available through accesses to the underlying Spark DataFrames (using Spark DataFrame API). This combination allows easy data engineering and fine-grained, yet reproducible, experiments.

The `CohortCollection` abstraction is a collection of `Cohorts` on which operations can be jointly performed. The `CohortCollection` has metadata that keeps the information about each `Cohort`, such as the successive operations performed on it, the Parquet files they are stored in and a git commit hash of the code producing the extraction from the `Source`.

International guidelines [Ben+15] regarding studies based on LODs insist on the explanation of cohort construction to highlight eventual population biases, motivating the following `CohortFlow` abstraction.

The `CohortFlow` abstraction is an ordered iterator defined as the following left fold operation

$$\text{foldl}(c : \texttt{CohortCollection}, \cap) := (((c_0 \cap c_1) \cap c_2) \cap \dots c_n)$$

assuming an input `CohortCollection` c of length n, where $\cap$ denotes an intersection of the `Cohorts`' patients. It is meant to track the stages leading to a final `Cohort`,

where each intermediate `Cohort` is stored along with textual information about the filtering rules used to go from each stage to the next one.

The `scalpel.stats` module produces descriptive statistics on a `Cohort` and their associated plots. For now, it contains more than 25 `Patient`-centric or `Event`-centric statistics, adding a custom one being very easy. Among other things, this module provides automatic reporting as text or graphical displays, with performance optimization through data caching. It can be combined with `CohortFlow` to compute various statistics at each analysis stage, to assess the biases induced along with successive population filtering operations. Flowcharts can easily be produced to track how many subjects were removed at each stage. Flowcharts can be produced either from a `CohortFlow`, or the metadata tracking the data extraction process produced by SCALPEL-Extraction. Examples are provided in Supplementary Material.

SCALPEL-Analysis also provides tools producing datasets in formats compatible with popular machine learning libraries. At the core of these tools is the `FeatureDriver` abstraction.

The `FeatureDriver` abstraction is used to transform `Cohorts` into data formats suitable for machine learning algorithms, such as `numpy.ndarray` [Har+20], or tensors from `tensorflow` [Mar+15] or `pytorch` [Pas+17] libraries. It is mainly a transformation of a Spark dataframe representation into a tensor-based format. `FeatureDrivers` perform several sanity checks, such as time-zone and event dates consistency, and can be easily extended by end-users, thanks to the PySpark API.

I.4 Results

Scaling experiments presented in this section were performed on a SNDS subset containing 13.7 million patients followed up to three years described in Table I.1. Data from this sample is structured data containing common data types (timestamps, integers, floats, small strings), normalized according to the SNDS data model. The testing data consisted in outpatient data (DCIR) and inpatient data excepted home hospitalization, rehabilitation centers and psychiatric hospitals (PMSI-MCO). Raw data was extracted from the SNDS by CNAM, the French agency that manages this database. Extracts were dumped on the testing cluster as a set of CSV files.

SCALPEL3 was tested on a Mesos [Hin+11] cluster of commodity servers with 14 worker nodes driven by 4 master nodes. Worker nodes resources amount to 224 2.4Ghz logical cores, 1.7Tb of RAM, and 448Tb of storage distributed over 88 spinning hard drives. These resources are shared over the cluster by HDFS [Shv+10] for data storage and by Spark for memory storage and computations. This cluster and the configuration of the jobs were not fine-tuned for the usage of SCALPEL3, but follow standard guidelines for cluster configuration for distributed computing with Spark.

Table I.1 – Characteristics of the dataset used for experiments. Results are produced on a subset of SNDS containing 13.7 million subjects, followed up to three years. The scope is restricted to outpatient data (DCIR) and inpatient data excepted hospitalization at home, rehabilitation centers and psychiatric hospitals (PMSI-MCO). The central fact table of DCIR records cash flows resulting from healthcare reimbursements to patients covered by the French national healthcare insurance. One line in this table correspond to one cash flow (such as the reimbursement of a drug bought following a prescription). The central fact table of PMSI-MCO records hospital stays. Events occurring during the stay are stored in dimension tables linked to this central table.

Count	DCIR	PMSI-MCO
Rows in the central table	10,579,545,716	35,375,046
Rows in the denormalized table	10,636,094,654	3,208,682,967
Patients	13,762,623	7,807,517
Drug reimbursements events	1,933,985,925	NA
Distinct drug codes	16,289	NA
Reimbursed medical acts events	210,847,422	97,484,303
Distinct medical acts codes	7254	7591
Diagnoses events	NA	120,212,253
Distinct diagnoses codes	NA	16,895
Source data set disk size (CSV, GB)	6,416.3	48.7
Source data set disk size (Parquet, GB)	572.7	5.9
Flattened data set disk size (Parquet, GB)	690.6	8.9

Denormalizing this dataset using SCALPEL-Flattening took about 6 hours using the 14 worker nodes. During the conversion of CSV tables to parquet files, worker nodes CPU and memory usage are maxed out on most worker nodes. During the join operations, resource usage is first dominated by network I/O to shuffle the data across the workers, followed by an increase in CPU and memory usage reaching two-thirds of the cluster capacity. Note that the current framework used for SNDS data cannot handle such denormalization so that there is no element of comparison for SCALPEL-Flattening with it.

SCALPEL-Extraction was evaluated on the following extraction tasks, that correspond to typical events required for public health research studying relations between fractures and some drug exposures: (a) extraction of patient demographics (gender, age, eventual date of death), (b) extraction of drug dispenses, (c) filtering of patients w.r.t their first date of drug use (prevalent drug users, 65 drugs), (d) computation of drug exposures based on drug dispenses dates, (e) extraction of reimbursed medical acts, (f) extraction of diagnoses, (g) identification of fractures using the algorithm described in [Bou+20] based on medical acts and diagnoses.

Indicative baseline performance was established by executing similar queries on the current SNDS infrastructure, based on SAS Enterprise Guide for analytics [Sup76], connected to an Oracle SQL database hosted on Oracle Exadata servers [Ora08]. This baseline performance was computed with a single run, as the current SNDS framework is designed to allocate resources dynamically each time a new query is submitted. The monitoring of resource usage on this SAS-Oracle infrastructure is not straightforward, since computations are divided between SAS and Oracle jobs, and since the resources of the Oracle Exadata infrastructure are divided across servers focused on storage or computation. At peak use (for task (c)), the Oracle job was using 10 CPUs supported by 4.9GB of PGA memory, while SAS was using 1 to 6GB of RAM.

An assessment of the horizontal scaling of SCALPEL3 is performed by varying the number of executors (4 logical cores and 25 GB RAM) to perform these queries. All the results are displayed in Figure I.3.

SCALPEL-Analysis aims at providing useful abstractions to ease cohort data manipulation. We provide in Supplementary Material, see Section I.A herein, examples that illustrate how these abstractions can be leveraged to perform typical data preparation in a few lines of code.

I.5 Discussion

SCALPEL-Extraction reaches performances similar to SQL-SAS based SNDS framework when using 6 executors (Figure I.3 (h)). It is consistently faster on tasks involving large data volumes or complex operations such as tasks (b), (c), (d), and (g). On the other hand, tasks involving the PMSI-MCO database (tasks (e) and (f)) exhibit poor performance. This is rooted in the flat table structure as PMSI-MCO is not sparse-by-block like DCIR (see the difference in the ratio of Rows in central table w.r.t. denormalized table in Table I.1). It results in performing more tests on row values and data shuffle than necessary when performing queries on PMSI-MCO. Performance on these tasks could be further improved by slightly modifying the join strategy in the flattening step to ensure PMSI-MCO sparsity by block.

The cost of data denormalization should be considered to be fixed as this operation is done once and for all. The denormalized data can then be updated incrementally when new data are fed into the cluster (typically a few times a year).

SCALPEL-Extraction scales almost linearly from 4 to 16 executors. The scaling gains then slow down, reaching peak performance at 28 executors (see Figure I.3). These diminishing returns can be caused by the cluster resource sharing between storage services (HDFS) and computation (SCALPEL3). As a result, SCALPEL3 resource usage can be in conflict with HDFS resources as soon as the number of

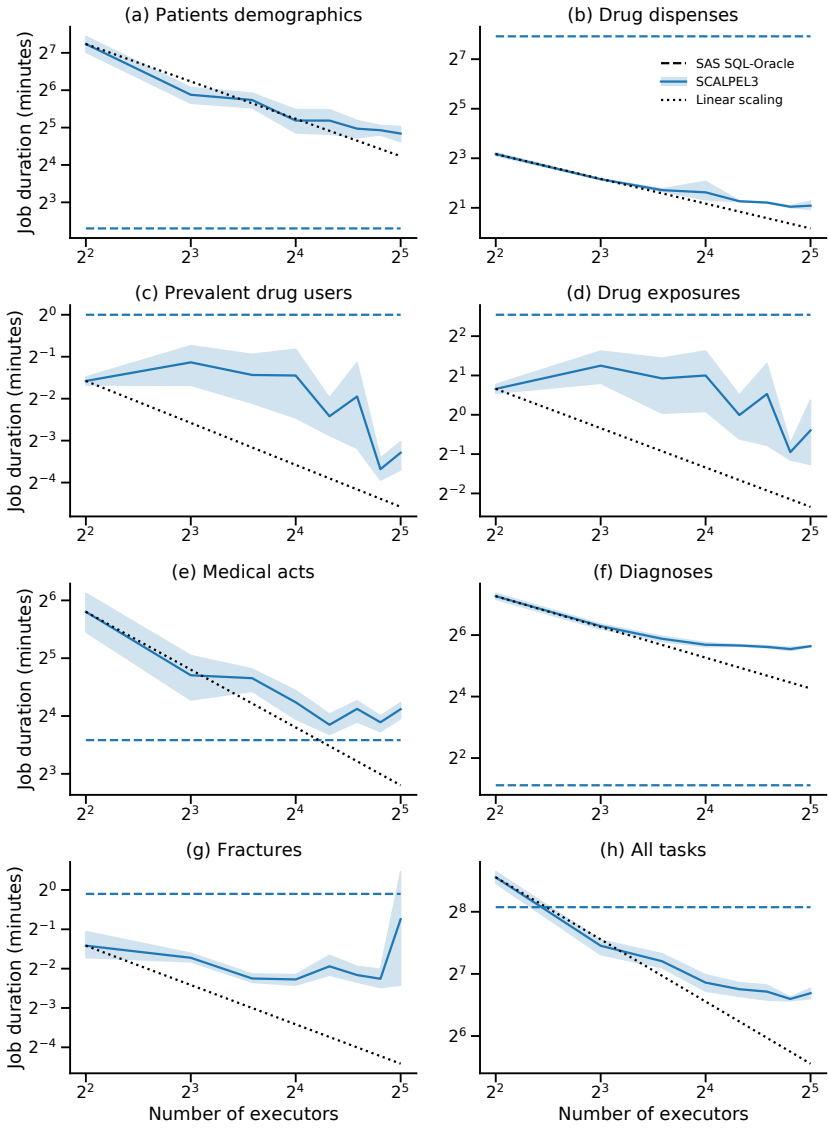


Figure I.3 – SCALPEL-Extraction scaling experiments. The blue solid line represents the mean total running time (in seconds) of queries (a) – (g) described in Section I.4 when varying the number of worker nodes used to perform the computation. Figure (h) represents the total running time of the (a) – (h) queries. Light blue bands represent one standard deviation computed over 5 runs. The dotted line corresponds to a theoretical performance assuming a perfect horizontal linear scaling (based on the single node performance). Dashed lines represent the runtime of similar queries on the SNDS SAS-Oracle infrastructure using a single run. Multiple runs were not performed on SAS-Oracle as computing resources are dynamically allocated for each queries and cannot be set beforehand.

nodes used by SCALPEL3 excess one-third of the cluster[3]. Splitting the cluster nodes between storage nodes and computation nodes could improve horizontal scalability. Note that for very small tasks (such as (c), (d), (g)), runtime is dominated by I/O operations and do not benefit particularly from additional CPUs.

Besides performance considerations, note that SCALPEL3 uses only open-source, free software and runs on commodity hardware, which is likely cheaper than Oracle Exadata servers and easier to scale if the data volume increases: a Spark cluster easily scales "horizontally" by adding more nodes.

The performance comparison between the two infrastructures is limited by (i) the impossibility to set the resources used by SAS-Oracle beforehand for these experiments does not allow for multiple runs and (ii) slight differences in query implementation caused by design differences such as columnar vs row orientation. Nonetheless, it shows that SCALPEL3 can be used as a viable open-source alternative running on commodity hardware while benefiting from horizontal scaling on very large jobs.

Besides, SCALPEL3 greatly improves the maintainability, audit, and reproducibility of studies using SNDS. First, continuous integration of code updates and large code coverage (94%) with unit testing is a big improvement in terms of maintainability over copy-pasted SQL snippets. Secondly, SNDS expertise encapsulation for events extraction is fully tested and maintained in SCALPEL3, so it eases extraction algorithms reuse for studies and lowers the entry-barrier to SNDS. Obviously, design and maintenance of SNDS concept extractors by a team of developers and SNDS specialists is a mandatory task, as the database contents are constantly evolving. Moreover, the relevance of extracted data (to answer a trade issue) requires some SNDS knowledge and is the responsibility of the user.

The combination of expert knowledge encapsulation (SCALPEL-Extraction) and interactive cohort manipulation (SCALPEL-Analysis) results in smaller and more readable user-code, leading to easily shared and reproducible studies, supported by data tracking and automated audit reports. Finally, SCALPEL3 allows producing datasets compatible with several Python machine learning libraries formats, fostering methodological research on SNDS data, which was not possible with the proprietary software that is currently used.

The choice of the Python language might help SCALPEL3 adoption among the data science and machine learning community, while it might hinder its use among public health researchers who are traditionally using proprietary statistical softwares or the R language. SCALPEL3 can be used in standalone mode[4] or in distributed mode[5] when working on large datasets. The knowledge and skills required to manage

[3]HDFS is configured to replicate the data across the worker nodes three times; HDFS performance is thus not much impacted if one-third of the nodes are not available at some point.

[4]Using a single large server.

[5]Using a computing cluster.

a computing cluster are not yet widespread which could also impede a large adoption of the distributed mode among small organizations.

Finally, while SCALPEL3 does not support international data standards yet, the development of vocabulary mapping tables in France was anticipated so as to ease future support of data standards such as OMOP-CDM [Rei+10] or FHIR [BS13] to SCALPEL3.

I.6 Conclusion

SCALPEL3 could be further improved by optimizing the flattening step, so as to ensure optimal block-sparsity of the resulting denormalized databases automatically. Besides, optimizing the cluster design to separate storage from computation as well as using YARN instead of Mesos to manage resources could help to improve its performance further by lowering data access times. Finally, using Apache ORC [Apa15] instead of Parquet could also lead to further performance improvements. Parquet was initially chosen over ORC because of better integration with Spark. ORC is now well-integrated in it and has been reported to have better performances and a higher compression factor on non-nested data.

I.7 Summary Table

- Strengths:
  - Expert knowledge encapsulation lowers entry barriers to SNDS use
  - Important improvement of query performance on sparse-by-block denormalized data
  - Horizontal scalability
  - Code versioning and rigorous testing
  - Low hardware cost
  - Open-source software
  - Inter-operates with rich ecosystems (Python, Scala) providing many machine learning and data analysis libraries
- Weaknesses:
  - Suppose familiarity with Python programming. While it can be assumed that most data scientists are fluent in Python, it might not be the case among the public health community.

  - Requests on flattened PMSI are too slow as it is not sparse-by-block. Improvements to the flattening are being developed to solve this issue.
  - SCALPEL3 concept extraction supposes continuous algorithms and code maintenance to ensure it is always up to date with eventual changes in SNDS structure and contents.

- Opportunities:
  - Reduce entry barriers by lowering the knowledge required to use SNDS data
  - Encapsulated knowledge and code versioning fosters reproducibility
  - Interoperability and open-source code foster methodological research
  - Open source software allows us to perform code audit and to have full control over the software and infrastructure.

- Threats:
  - Public health researchers working with proprietary software or the R language might not be Python-fluent.
  - Distributed use suppose the knowledge of cluster management in the information systems team.
  - Connectors to existing data standards not ready yet, ongoing effort.

I.8 Declarations of interest

None

I.9 Authors' contribution

This work was co-authored by Emmanuel Bacry (CEREMADE, CMAP), Stéphane Gaïffas (LPSM, ENS), Fanny Leroy (CNAM), Maryan Morel[6] (CMAP), Dinh-Phong Nguyen (CMAP, CNAM), Youcef Sebiat (CMAP), Dian Sun (CMAP). The authors contributed as follows:
Manuscript preparation: MM, EB, SG, DPN, YS, DS.
Concept and design of the data pipeline: YS, DS, MM.
Concept and design of the cohort manipulation library: YS, MM, DS, DPN.
Benchmarking: YS, FL, DS, MM.
Data sharing and critical review: DPN, FL.

[6]Corresponding author

I.10 Acknowledgments

We thank the engineers who worked on this project at some point: Firas Ben Sassi, Prosper Burq, Philip Deegan, Daniel De Paula e Silva, Angel Francisco Orta, Xristos Giatsidis, Sathiya Prabhu Kumar.

We also thank the people from CNAM or Polytechnique who were or are currently involved in the Polytechnique-CNAM partnership, namely, for CNAM: Muhammad Abdallah, Aurélie Bannay, Hélène Caillol, Anthony Du, Sébastien Dumontier, Mehdi Gabbas, Claude Gissot, Moussa Laanani, Mickaël Lechapelier, Clémence Martin, Anke Neumann, Cédric Pulrulczyk, Jérémie Rudant, Omar Sow, Kévin Vu Saintonge, Alain Weill, and for Polytechnique: Qing Chen, Agathe Guilloux, Anastasiia Nitavskyi, Yiyang Yu.

This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

Appendix

I.A Scalpel Analysis usage examples

This section presents a quick example of the SCALPEL-Analysis API. `In [ ]` and `Out [ ]` respectively indicate numbered input code and output results. Example `[1]` shows how to load a cohort collection from a json file produced by SCALPEL-Extraction. Examples `[2]`, `[3]`, `[4]` show how to access a cohort from a cohort collection and to count their subjects. Example `[5]` shows how to use algebraic manipulations over cohort to remove prevalent cases from a given population, and times this operation to show that it fast enough for interactive use. Example `[6]` highlight automatically generated captions for cohorts resulting from algebraic operations, while examples `[7]` and `[8]` show how to access subject and event data from a cohort. Examples `[9]` and `[10]` illustrate how to use SCALPEL-Analysis to define a `CohortFlow` from a sequence of cohorts, then using `scalpel.stats` to obtain statistics about the distributions of gender and age along the stages. In example `[9]`, excluding patients with a fracture does not introduce much changes in the gender and age distributions. In example `[10]` however, keeping only patients with fractures in the final stage leads to an older population, with an important change in the age distribution of women (a well-known phenomenon related to osteoporosis).

```
In [1]: from scalpel.core.cohort_collection import CohortCollection

        # metadata_path = '/path/to/some/metadata_file.json'
```

```
        cc = CohortCollection.from_json(metadata_path)
        print(cc.cohorts_names)
```

```
Out[1]: {'follow_up', 'acts', 'fractures', 'extract_hospital_stays',
         'filter_patients', 'liberal_acts', 'extract_patients', 'exposures',
         'diagnoses', 'drug_purchases'}
```

```
In [2]: base_population = cc.get('extract_patients')
        base_population.subjects.count()
```

```
Out[2]: 5186601
```

```
In [3]: exposed_subjects = cc.get('exposures')
        exposed_subjects.subjects.count()
```

```
Out[3]: 2666662
```

```
In [4]: fractured_subjects = cc.get('fractures')
         fractured_subjects.subjects.count()
```

```
Out[4]: 179072
```

```
In [5]: %%timeit
         # Select subjects in base population who were exposed but
         # have not experienced a fracture
         final_cohort = (exposed_subjects.intersection(base_population)
                                 ).difference(fractured_subjects)
         final_cohort.subjects.count()
```

```
11.3 s ± 4.5 s per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
Out[5]: 2542922
```

```
In [6]: final_cohort.describe()
```

```
Out[6]: 'Events are exposures. Events contain only subjects
         with event exposures with extract_patients without
         subjects with event fractures.'
```

```
In [7]: final_cohort.subjects.show()
```

```
+---------+------+-------------------+-------------------+
|patientID|gender|          birthDate|          deathDate|
+---------+------+-------------------+-------------------+
|    Alice|     2|1934-07-27 00:00:00|               null|
|      Bob|     1|1951-05-01 00:00:00|               null|
|   Carole|     2|1942-01-12 00:00:00|               null|
|    Chuck|     1|1933-10-03 00:00:00|2011-06-20 00:00:00|
|    Craig|     1|1943-07-27 00:00:00|2012-12-10 00:00:00|
|      Dan|     1|1971-10-07 00:00:00|               null|
|     Erin|     2|1924-01-12 00:00:00|               null|
|      Eve|     2|1953-02-21 00:00:00|               null|
+---------+------+-------------------+-------------------+
```

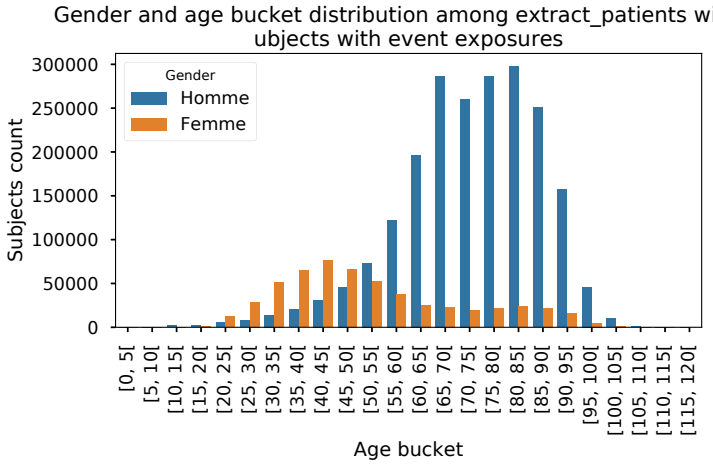
```
In [8]: final_cohort.events.show()
```

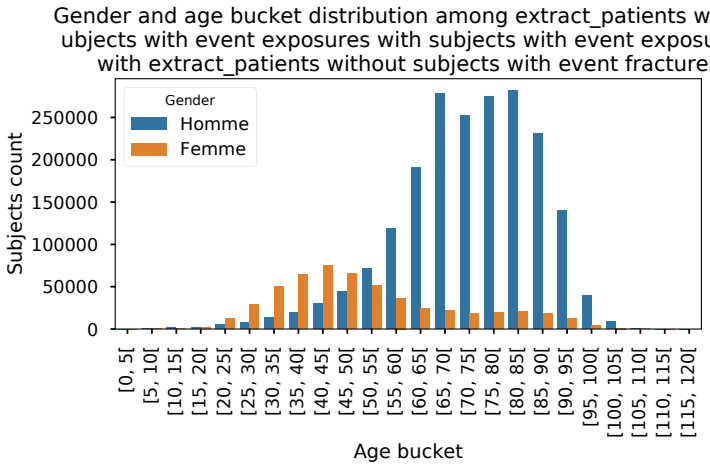
```
+---------+--------+-------+-----+------+-------------------+-------------------+
|patientID|category|groupID|value|weight|              start|                end|
+---------+--------+-------+-----+------+-------------------+-------------------+
|    Alice|exposure|   null|DrugA|   1.0|2013-08-08 00:00:00|2013-10-07 00:00:00|
|    Alice|exposure|   null|DrugB|   1.0|2012-09-11 00:00:00|2012-12-30 00:00:00|
|    Alice|exposure|   null|DrugC|   1.0|2013-01-23 00:00:00|2013-03-24 00:00:00|
|      Bob|exposure|   null|DrugB|   1.0|2014-03-04 00:00:00|2014-05-03 00:00:00|
|   Carole|exposure|   null|DrugB|   1.0|2010-01-25 00:00:00|2010-12-13 00:00:00|
|      Dan|exposure|   null|DrugA|   1.0|2012-11-29 00:00:00|2013-01-28 00:00:00|
|     Erin|exposure|   null|DrugC|   1.0|2010-09-09 00:00:00|2011-01-17 00:00:00|
|      Eve|exposure|   null|DrugA|   1.0|2010-04-30 00:00:00|2010-08-02 00:00:00|
+---------+--------+-------+-----+------+-------------------+-------------------+
```

```
In [9]: from scalpel.stats.patients import distribution_by_gender_age_bucket
        from scalpel.core.cohort_flow import CohortFlow

        flow = CohortFlow([base_population, exposed_subjects, final_cohort])

        for cohort in flow.steps:
            figure = plt.figure(figsize=(8, 4.5))
            distribution_by_gender_age_bucket(cohort=cohort, figure=figure)
```

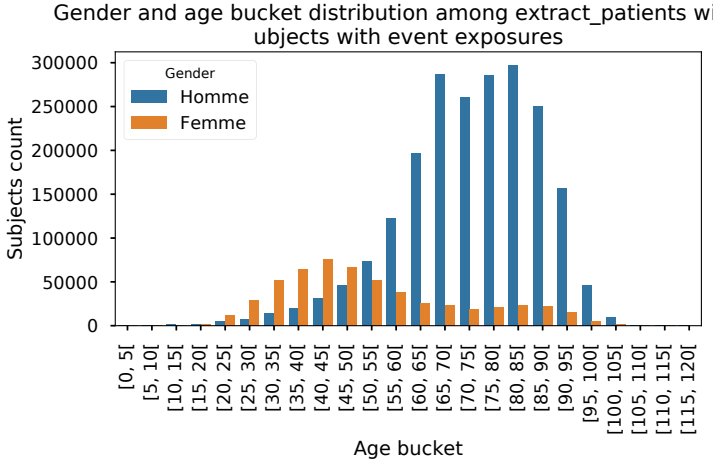


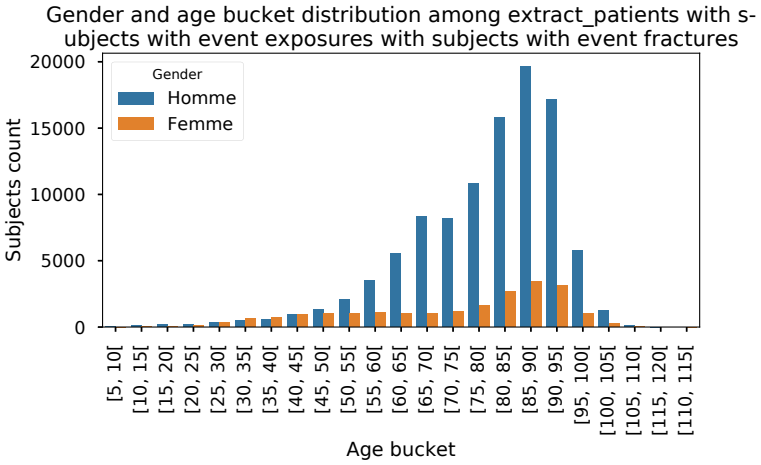



```
In [10]: from scalpel.stats.patients import distribution_by_gender_age_bucket
         from scalpel.core.cohort_flow import CohortFlow

         flow = CohortFlow([base_population, exposed_subjects, fractured_subjects])

         for cohort in flow.steps:
             figure = plt.figure(figsize=(8, 4.5))
             distribution_by_gender_age_bucket(cohort=cohort, figure=figure)
```




Gender and age bucket distribution among extract_patients with s-
ubjects with event exposures with subjects with event fractures
Gender
Homme
Femme
Subjects count
20000
15000
10000
5000
0
[5, 10[
[10, 15[
[15, 20[
[20, 25[
[25, 30[
[30, 35[
[35, 40[
[40, 45[
[45, 50[
[50, 55[
[55, 60[
[60, 65[
[65, 70[
[70, 75[
[75, 80[
[80, 85[
[85, 90[
[90, 95[
[95, 100[
[100, 105[
[105, 110[
[115, 120[
[110, 115[
Age bucket

I.B List of SNDS databases currently denormalized.

Table I.B.1 – List of SNDS sub-databases which are currently denormalized by SCALPEL-Flattening. IR_IMB_R and IR_BEN_R are tables and were simply converted to Parquet files.

Database	Contents
DCIR	Outpatients reimbursement data
PMSI	Hospital discharges
MCO	Acute ward
MCO CE	Acute ward outpatients treatment
SSR	Rehabilitation
SSR CE	Rehabilitation outpatients treatment
HAD	Home-to-home care
IR_IMB_R	Long term chronic diseases
IR_BEN_R	Patients socio-demographic information

I.C List of available extractors

Table I.C.1 – List of implemented event extractors. This list is meant to grow over time. More details are available in SCALPEL-Extraction [Pau+19] wiki on GitHub at `https://github.com/X-DataInitiative/SCALPEL-Extraction/wiki`.

Extractor	Source databases	Event Type
Medical acts		
CCAM	DCIR, MCO, MCOCE, SSR, SSRCE, HAD	Punctual
NGAP	DCIR, MCOCE	Punctual
CSARR	SSR	Punctual
Biological acts	DCIR	Punctual
Practitioner encounter		
Medical	DCIR	Punctual
Non-medical	DCIR	Punctual
Drug dispenses	DCIR	Punctual
Diagnoses		
Main	MCO, SSR, HAD	Punctual
Associated	MCO, SSR, HAD	Punctual
Linked	MCO, SSR, HAD	Punctual
Long-term chronic disease	IR_IMB_R	Longitudinal
Hospital stay	MCO	Longitudinal
Emergency visit	MCOCE	Punctual
SSR Stay	SSR	Longitudinal
Hospital takeover	SSR, HAD	Punctual
Main Takeover reason	HAD	Punctual
Associated Takeover reason	HAD	Punctual
Patient	IR_BEN_R, DCIR, MCO, SSR, HAD	Person

I.D List of the available transformers

Table I.D.1 – List of implemented transformers. This list is meant to grow over time. More details are available in SCALPEL-Extraction [Pau+19] wiki on GitHub at `https://github.com/X-DataInitiative/SCALPEL-Extraction/wiki`.

Transformer	Source events [optional]
Observation period	Patients, [Any]
Trackloss	Patients, [drug dispenses]
Follow-up	Patients, observation period, [trackloss, drug dispenses, diagnoses]
Drug prescription	Drug dispenses
Drug interaction	Drug dispenses
Exposure	
Limited in time	Drug dispenses, Follow-up, [drug interaction]
Unlimited	Drug dispenses, Follow-up, [drug interaction]
Outcomes	
Fractures per body site	Medical acts, diagnoses
Bladder cancer	Medical acts, diagnoses
Infarctus	Diagnoses
Heart failure	Diagnoses

CHAPTER II

CONVSCCS: CONVOLUTIONAL SELF-CONTROLLED CASE SERIES MODEL FOR LAGGED ADVERSE EVENT DETECTION

With the increased availability of large electronic health records (EHRs) databases comes the chance of enhancing health risks screening. Most post-marketing detection of adverse drug reaction (ADR) relies on physicians' spontaneous reports, leading to under-reporting. To take up this challenge, we develop a scalable model to estimate the effect of multiple longitudinal features (drug exposures) on a rare longitudinal outcome. Our procedure is based on a conditional Poisson regression model also known as self-controlled case series (SCCS). To overcome the need of precise risk periods specification, we model the intensity of outcomes using a convolution between exposures and step functions, which are penalised using a combination of group-Lasso and total-variation. Up to our knowledge, this is the first SCCS model with flexible intensity able to handle multiple longitudinal features in a single model. We show that this approach improves the state-of-the-art in terms of mean absolute error and computation time for the estimation of relative risks on simulated data. We apply this method on an ADR detection problem, using a cohort of diabetic patients extracted from the large French national health insurance database (SNIIRAM), a claims database containing medical reimbursements of more than 53 million people. This work has been done in the context of a research partnership between Ecole Polytechnique and CNAMTS (in charge of SNIIRAM).

Keywords: *Conditional Poisson Model, Self-Controlled Case Series, Risk screening, Penalisation, Scalability, Total Variation.*

II.1 Introduction

In recent years, there has been a rapid increase in health data volume and availability. Large observational databases (LODs) such as claims databases contain electronic health records (EHRs) of millions of patients. One way to leverage this data is adverse drug reaction (ADR) detection. ADRs are adverse outcomes caused by drugs which might not have been detected during prelicensing studies. ADRs can be related to multiple factors such as dose or time effects or even to patients' susceptibility due to genetic variation, gender, age, etc. [AF03]. This paper focuses on time effects, i.e. on the relationship between ADR occurrences and occurrences of other past events (e.g. drug purchases), since it is known that some ADRs can be identified years after commercialisation [Dow+17].

While LODs have been used to investigate ADRs after spontaneous reports, a more extensive use could improve ADR detection by generating hypotheses directly from the data using screening strategies [Tri+09]. In recent years, this perspective led to an increased research effort involving the use of LODs [Hri+15].

However, using LODs for ADR screening is not a trivial task. This kind of data can be quite heterogeneous, in terms of data types, structure, granularity and quality, due to fragmentation across multiple institutions for example. Several research projects are focusing on mitigating these issues. The Observational Medical Outcomes Partnership (OMOP, [Mar+12]), and later the Observational Health Data Sciences and Informatics (OHDSI, [Hri+15]) produced data models standards and methodologies allowing to improve EHR homogeneity across several institutions across several countries. In this work, we focus on the large French national claims database SNIIRAM [Tup+10b]. Its data is collected, harmonised and curated from multiple institutions across the country by CNAMTS, resulting in a country-wide claims database containing information on 83% of the French population. This database might be less biased than many LODs due to its large population coverage and quite accurate thanks to the automation of large parts of the data recording and cleaning processes.

A first challenge comes from the scale of the data. Indeed, LODs allows to study millions of patients across several years, hence it requires the use of scalable algorithms. The scalability must also be thought in terms of the number of drugs the patients are exposed to. When using LODs for risk screening, prior knowledge on the potentially problematic drugs might be scarce, consequently, the number of combination of drugs and outcomes to consider is potentially very large.

Many other challenges comes from the fact that EHR data tends to reflect the healthcare system rather than the patients' physiology. Indeed, EHR data are likely to contain non-random errors, record gaps, misleading timestamps and uncontrolled confounding [HA13]. For example, as the diagnoses result from clinical findings, raw timestamps could suggest that diseases follow their effects [HAP11]. As a result,

mapping complex, raw EHR data to clinical conditions is a very hard task, and is a research field by itself. While our work does not solve this problem, we hope to alleviate some it by using LODs to perform ADR screening.

There does not seem to be a clear consensus about which methods should be preferred when working with EHRs. However, models based on a self-control strategy, such as univariate self-control case series model [Far95] or temporal pattern discovery algorithms [Nor+10] seems to perform better empirically than cohort and case-control methods [Rya+13b]. The poor performance of case-control methods can be explained by the lack of proper "metadata" about patients (smoker status, wealth, etc.) in LODs, which are used to find proper controls in case-control studies. Besides, self-control methods might be more robust to unobserved confounders than cohort methods as they ignore non-longitudinal confounders [Far95].

We focus on Self-Controlled Case Series (SCCS) models, originally developed for vaccine safety studies [Far95], since then applied in post-marketing studies using LODs [Gau+17]. The SCCS model scales quite well since it is fitted on cases only. Moreover, as explained below, its goodness-of-fit function cancels out non-longitudinal confounders, which reduces potential non-longitudinal biases. Thus, an SCCS model helps with the scalability and unobserved confounding issues described earlier. However, an SCCS model relies heavily on the definition of a time-at-risk period, which makes it hard to use in multivariate settings.

Previous attempts to solve this problem relied on the use of splines to provide a more flexible modelling of drug effects [GWF16; GWF17; Sch+16]. However, the use of splines makes the estimation of the model more complicated, resulting in models able to fit the effect of a single drug in addition to a temporal baseline. This can be problematic when performing ADRs screening, as SCCS is sensitive to temporal confounders, and thus, to the omission of longitudinal features.

This paper introduces a new approach in the framework of SCCS models that addresses the three challenges mentioned previously:

- it considers several longitudinal features at the same time (longitudinal drug exposures),
- it cancels out non-significant drug effects automatically
- it learns automatically and in a flexible way the significant drug effects, with no precise knowledge on a time-at-risk period,
- it runs faster than comparable algorithms when studying many drugs at a time.

Hence, it provides an important extension to the usage of SCCS models, allowing to *study multiple exposures at the same time, while requiring much less attention to the definition of time-at-risk periods.* An application of this methodology is described in

Section II.4.2 below, and leads to a scalable approach with respect to the number of drugs. On the one hand, it does not require a high precision work when preparing the dataset (as done in [Neu+12]). On the other hand, it is not thought as a replacement of such approaches but rather as a screening method to identify potential problematic drugs that might require specific subsequent investigations (using [Neu+12] types of approach).

The paper is organised as follows. We first describe SCCS models in Section 2 and construct our method in Section 3. Numerical experiments are given in Section 4. It includes in Section II.4.1 experiments on simulated data, with a comparison to state-of-the-art methods from the SCCS literature. In particular, these simulations as designed to reproduce some of the problems met with the data used in Section II.4.2, in order to test the robustness of our algorithm compared to the state-of-the-art. Section II.4.2 gives an application of our method on a LOD from the French national health insurance information system (SNIIRAM, a database built around medical reimbursements of more than 53 million people). Our model produces consistent results with a population-based cohort study [Neu+12] when estimating the effect of pioglitazone (a hypoglycemic agent) on the risk of bladder cancer. A conclusion is given in Section 5, and mathematical and numerical details are provided in Supplementary Material.

II.2 Self-controlled case series models

SCCS models allow to estimate the impact of longitudinal features (such as time-varying exposures to drugs) on the occurrence intensity of events of interest (such as dates of adverse events), see [Far95]. An interesting particularity with this family of methods is that individuals form their own controls: individuals who do not experience the event of interest are not used to fit the model. This construction relies on the property of order statistics of the Poisson process and the statistical output of such models is an estimation of the *relative incidence* of the longitudinal features, i.e. the relative increase of the outcomes intensity.

II.2.1 Conditional Poisson regression and SCCS models

Data is available from a global observation period $(a, b]$, where the time can be either calendar or measured by the age of individuals. Each patient $i = 1, \dots, m$ has an observation period $(a_i, b_i] \subset (a, b]$, in which we observe:

- the time occurrences $t_{i,1} < t_{i,2} < \cdots$ of the event of interest (also called *outcome* in what follows), or, equivalently a counting process N_i, defined as $N_i(t) = \sum_{k \geq 1} \mathbb{1}_{t_{i,k} \leq t}$ and $n_i = \int_{(a_i, b_i]} dN_i(t)$ the total number of outcomes of patient i,

- a vector of d longitudinal features

$$X_i = \big(X_i(t) = (X_i^1(t) \ \cdots \ X_i^d(t)) \ : \ t \in (a_i, b_i]\big),$$

  where in the context of drug safety studies, $X_i^j(t)$ gives us information about the exposure of patient i to drug j at time $t \in (a, b]$.

The model developed in this paper relies on the usual SCCS model key assumptions [FW06]. Namely, we assume that

(1.) The features are exogenous, meaning that the counting process N_i does not have any influence on the features X_i;

(2.) The interval of observation $(a_i, b_i]$ is independent of N_i;

(3.) The process N_i is a Poisson process conditionally to $(X_i(t) \ : \ t \in (a_i, b_i])$.

Assumption (1.) allows to condition on the full trajectory of the longitudinal features X_i in (4). In addition, thanks to (2.), the following derivations have to be understood conditionally to $(a_i, b_i]$. We may then define the conditional intensity of process N_i as

$$\lambda_i(t, X_i) = \mathbb{P}(dN_i(t) = 1 \mid X_i) \tag{II.1}$$

for $t \in (a_i, b_i]$. This model can be, therefore, understood as a regression model, allowing to regress the outcomes in N_i on the longitudinal features X_i.

In order to study acute vaccine adverse effects, [FW06] considers the following model for the intensity:

$$\lambda(t, X_i) = \exp\big(\psi_i + \gamma_i + \phi(t) + X_i(t)^\top \beta\big),$$

where ψ_i is the baseline incidence of patient i and γ_i is a sum of non-temporal fixed and random individual effects. The parameter $\phi(t)$ is a time-dependent baseline which is common to all individuals. If age is used as the time scale, this term can help to capture age effects. The vector of parameters $\beta \in \mathbb{R}^d$ quantifies the effect of the longitudinal features $X_i(t)$ on the intensity. The idea of the SCCS method is to condition on both X_i and n_i. Usual arguments (see Section II.A in Supplementary Material) imply that the likelihood of $N_i|(X_i, n_i)$ of $i = 1, \dots, m$ independent patients is proportional to

$$\prod_{i=1}^{m} \prod_{k=1}^{n_i} \frac{\lambda_i(t_{i,k}, X_i)}{\int_{a_i}^{b_i} \lambda_i(s, X_i) ds} = \prod_{i=1}^{m} \prod_{k=1}^{n_i} \frac{\exp\big(\phi(t_{i,k}) + X_i(t_{i,k})^\top \beta\big)}{\int_{a_i}^{b_i} \exp\big(\phi(s) + X_i(s)^\top \beta\big) ds}. \tag{II.2}$$

Note that the conditioning with respect to n_i induced two notable properties of (5):

- *Improved scalability:* the likelihood only depends on patients i such that $n_i \geq 1$ (while the "full" likelihood of $N_i|X_i$ does depend on patients i for whom $n_i = 0$). This is beneficial when studying rare adverse effects in large LODs.

- *Robustness to non-longitudinal confounders:* the non-longitudinal effects ψ_i and γ_i cancel out in the likelihood (5). This makes SCCS models particularly robust to the patient's susceptibility.

These two properties are appealing when working with LODs such as claims databases, as it helps to mitigate issues related to missing variables and the data scale. However, only relative incidences can be computed by taking the exponential of the corresponding coefficient, such as $\exp(\phi(t))$ for the baseline relative incidence.

SCCS models were initially designed for vaccine safety studies [Far95], using the suspected ADR as the outcome. In this context, estimating the relative incidence of drug use requires defining related time-at-risk periods in which the suspected ADR might occur. The longitudinal features $X_i(t)$ are then used to express the fact of being at risk or not at time t for a particular drug. One must then determine for how long patients are at risk after each exposure to a drug, and if this risk occurs either immediately or after some amount of time. Defining proper time-at-risk windows is a hard problem when studying a single (drug, ADR) pair, which worsens even further when considering a set $(\text{drug}_1, \text{ADR}), \dots, (\text{drug}_d, \text{ADR})$ of such pairs. In the case of ADR screening over multiple drugs, such a methodology might even become inappropriate.

II.2.2 Risk screening

When prior knowledge on time-at-risk windows is not available, a simple method is to use a large window in order to be sure to capture the potential effect. However, this strategy typically "dilutes" the risk over the window, see [Xu+11], leading to a model unable to detect ADRs. Existing works propose to relax the time-at-risk window definition while trying to overcome this risk dilution. It is proposed in [Xu+11] to select an optimal risk window by testing several window sizes, in a data-driven fashion. However, this method is difficult to adapt for ADR screening when considering d drugs and q risk windows at the same time, since it requires to fit q^d models.

A different approach relies on fitting time-dependent parameters in order to estimate the risk of ADR over large risk windows. The model estimates a time-varying relative incidence function all along the risk window instead of assuming it to be constant. This approach is used in [Sch+16], where the drug effect is a function θ of the accumulated exposures. It uses a discrete model with daily granularity, assuming that the integral of $X_i(t)$ over one day is equal to 1 when the patient is exposed to the studied drug. Accumulated exposures up to time t is measured by

$\int_{a_i}^{t} X_i(s)\mathrm{d}s$, where $X_i(t)$ is univariate, and expresses the exposure to a single drug at time t, leading to the following model for the intensity:

$$\lambda_i(t, X_i) = \exp\Big(\psi_i + \gamma_i + \phi(t) + \theta\Big(\int_{a_i}^{t} X_i(s)\mathrm{d}s\Big) + X_i(t)\beta\Big),$$

where the function θ is estimated using natural cubic splines. As the splines are not regularised, this model might be prone to overfitting. Alternatively, [GWF16] use a convolution to model drug effects, writing the intensity as

$$\lambda_i(t, X_i) = \exp\big(\psi_i + \gamma_i + \phi(t)\big) \int_{a_i}^{t} X_i(s)\theta(t-s)\mathrm{d}s.$$

In this model, $X_i(t)$ is either a point exposure $X_i(t) = \delta_{c_i}(t)$ where δ_{c_i} stands for a Dirac mass at date $c_i \in \mathbb{R}^+$, or a continuous exposure to a constant quantity x, namely $X_i(t) = x\mathbb{1}_{(c_i,b_i]}(t)$. In the former case, the intensity can be expressed as

$$\lambda_i(t, X_i) = \exp\left(\psi_i + \gamma_i + \phi(t)\right)\theta(t - c_i). \tag{II.3}$$

The function θ is estimated using M-splines (in order ensure positivity) in [GWF16; GWF17], while the age effect ϕ is estimated by step functions in [GWF16] and by splines in [GWF17]. The considered model could deal with multiple point exposures c_i for the drug, given that the maximum time gap between successive exposures is smaller than the support of θ, but the authors have not developed this point.

Both [Sch+16] and [GWF16; GWF17] seem restricted to the study of a single (drug, ADR) pair at a time. This can be problematic since SCCS is sensitive to time-varying confounders and benefits from studying multiple drugs at once as shown by both [Sim+13] and [MRM16]. In order to fit an SCCS model using several drugs at the same time, [GWF17] propose to extend their work by modelling additional drugs effect with step functions instead of splines. However, such functions are basically not regularised, which can result in overfitting, and are very sensitive to the chosen number of steps.

II.3 ConvSCCS: an extension of SCCS models

We now introduce our ConvSCCS model. It is an extension of the classical SCCS model in several directions. First, it allows considering exposures to several drugs. More importantly, our model is time-invariant thanks to a convolutional structure. Hence it can learn the potential effects of the drug exposures even without prior definition of precise time-at-risk periods.

More specifically, we construct a model that estimates the effect of longitudinal features using convolutions of low-granularity step functions with point drug exposures. The low-granularity leads to an over-parametrised model with poor estimation accuracy. We solve this issue in Section II.3.2 below by using a penalisation technique that combines total-variation and Group-Lasso penalties. The second will perform an automatic variable selection, while the first enforces longitudinal effects to be piece-wise constant over larger steps whenever statistically relevant. As illustrated in Section II.4, this leads to improvements over current state-of-the-art methods, and provides interpretable results on the observational database considered in this paper, see Section II.4.2.

II.3.1 Discrete convolutional SCCS

We assume that, for $i = 1, \dots, m$, the intensity λ is constant over time intervals $I_k = (t_k, t_{k+1}]$, $k = 1, \dots, K$ that form a partition of the observation interval $(a, b]$. Without loss of generality, we choose I_k to be of constant length 1. In practice, we use the smallest granularity allowed by data. Hence, we can assume that $(a_i, b_i] \cap I_k$ is either $\emptyset$ or I_k for all $i = 1, \dots, m$, and $k = 1, \dots, K$, which means that the observation period of each individual is a union of intervals I_k. Denoting by $\lambda_{i,k}$ the value of $\lambda(t, X_i(t))$ for $t \in I_k$, and defining $y_{ik} := N_i(I_k)$, the discrete SCCS likelihood can be written as

$$L(y_{i1}, \dots, y_{ik} | n_i, X_i) = n_i! \prod_{k=1}^{K} \left(\frac{\lambda_{ik}}{\sum_{k'=1}^{K} \lambda_{ik'}} \right)^{y_{ik}},$$

where we use the convention $0^0 = 1$, i.e. only the exposition period $(a_i, b_i]$ contributes to the likelihood, and since $N_i(I_k) = \lambda_{ik} = 0$ whenever $I_k \cap (a_i, b_i] = \emptyset$, see Section II.B of Supplementary Material for more details. We consider an intensity given by

$$\lambda_i(t, X_i) = \exp\Big(\psi_i + \gamma_i + \phi(t) + \int_{a_i}^{t} X_i(s)^\top \theta(t - s) \mathrm{d}s\Big).$$

Since the intensity is constant on each I_k, it can be rewritten as

$$\lambda_{ik}(X_i) = \exp\Big(\psi_i + \gamma_i + \phi_k + \sum_{k'=a_i}^{k} X_{ik'}^\top \theta_{k-k'}\Big),$$

where X_{ik} stands for the value of $X_i(t)$ for $t \in I_k$ and $\theta \in \mathbb{R}^{d \times K}$. We observe $l = 1, \dots, L_i^j$ starting dates of exposures c_{il}^j and introduce the features $X_{ik}^j = \sum_{l=1}^{L_i^j} \mathbb{1}_{k = c_{il}^j}$,

which leads to the following intensity

$$\lambda_{ik}(X_i) = \exp\Big(\psi_i + \gamma_i + \phi_k + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k-c_{il}^j} \mathbb{1}_{[0,p]}(k - c_{il}^j)\Big). \tag{II.4}$$

The quantity $\exp(\theta_k^j)$ corresponds to the relative incidence of an exposure to drug j that occurs k time units after an exposure start. Finally, the likelihood is equal to

$$L(y_{i1}, \dots, y_{ik} | n_i, X_i) = \prod_{k=1}^{K} \left(\frac{\exp\left(\phi_k + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k-c_{il}^j} \mathbb{1}_{[0,p]}(k - c_{il}^j)\right)}{\sum_{k'=1}^{K} \exp\left(\phi_{k'} + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k'-c_{il}^j} \mathbb{1}_{[0,p]}(k' - c_{il}^j)\right)} \right)^{y_{ik}} \tag{II.5}$$

and depends only on the parameters θ for the exposures and the age effects ϕ.

II.3.2 Penalised estimation

This formulation of intensity (6) is flexible since it allows to capture an immediate effect in θ_0^j, or delayed ones using θ_k^j for $k \geq 1$. This flexibility comes at a cost: it increases significantly the number of parameters to be estimated, which might lead to inaccurate estimations and to overfitting of the dataset. To that end, we introduce a penalisation technique which allows handling this issue, and which provides interpretable estimations of the relative risks as a byproduct.

We introduce groups $\theta^j = [\theta_1^j \cdots \theta_p^j] \in \mathbb{R}^p$ of parameters quantifying the impact of exposures to drugs $j = 1, \dots, d$ at different lags $k = 1, \dots p$. To avoid exposure effects overlapping, we assume that exposure starting times are far enough, that is $\min_{l,l'} |c_{il}^j - c_{il'}^j| > p$. We want to induce two properties on the relative risks of drugs exposures: a "smoothness" property along lags $k = 1, \dots, p$, namely we want consecutive relative risks $\exp(\theta_k^j)$ and $\exp(\theta_{k-1}^j)$ to be basically close; and the possibility for a drug to have no effect, namely to induce that θ^j can be the null vector. This can be achieved with the following penalisation that combines total and group-Lasso

$$\text{pen}(\theta) = \gamma_{\text{tv}} \sum_{j=1}^{J}\sum_{k=1}^{p-1} |\theta_{k+1}^j - \theta_k^j| + \gamma_{\text{gl}} \sum_{j=1}^{J} \|\theta^j\|_2 \tag{II.6}$$

over the groups θ^j for $j = 1, \dots, d$, where $\gamma_{\text{tv}} \geq 0$ and $\gamma_{\text{gl}} \geq 0$ are respectively levels of penalisation for the total-variation and the group-Lasso. The group-Lasso introduced in [YL06] acts like the lasso at the group level: depending on γ_{gl}, it can cancel out a full block θ^j. Total-variation penalisation is known to consistently estimate change points for the estimation of the intensity of a Poisson process, see [AGG15].

We write the penalised negative log-likelihood of our model as follows:

$$-\ell(\phi, \theta) + \text{pen}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log \left(\frac{\lambda_{ik}(\phi, \theta)}{\sum_{k'=1}^{K} \lambda_{ik}(\phi, \theta)} \right) + \text{pen}(\theta), \tag{II.7}$$

where pen is given by (7) and where we recall that

$$\lambda_{ik}(\phi, \theta) = \exp\Big(\phi_k + \sum_{j=1}^{d} \sum_{l=1}^{L_i^j} \theta^j_{k - c_{il}^j} \mathbb{1}_{[0,p]}(k - c_{il}^j) \Big).$$

The function (8) is convex and $\ell(\phi, \theta)$ is gradient-Lipschitz. However, since the sparsity-inducing penalisation $\text{pen}(\theta)$ is not differentiable, we use a proximal first-order method to minimise efficiently (8). Namely, we use the state-of-the-art SVRG algorithm from [XZ14], which is a fast stochastic proximal gradient descent algorithm, using a principle of variance reduction of the stochastic gradients.

Finally, the hyper-parameters γ_{tv} and γ_{gl} are selected using a stratified V-Fold cross-validation on the negative log-likelihood.

II.4 Experiments

In this section, we compare ConvSCCS with the state-of-the-art, namely SmoothSCCS [GWF16] and NonparaSCCS [GWF17], that are described below, see also Section II.2.2 for further details.

ConvSCCS is the method introduced in this paper: an extension of SCCS models allowing to fit the effect of *several* drugs on an ADR in a flexible way, see also Table II.4.1 below. ConvSCCS is available in our open-source `tick` library, see Section II.C in Supplementary Material for details.

SmoothSCCS is introduced in [GWF16], which uses splines to model the effect of a *single* drug exposure to a disease and step functions to model the effect of age. We use the `SCCS` R package implementation, available at `http://statistics.open.ac.uk/sccs/r.htm`. We use 12 knots and six groups of age as suggested in [GWF16]. Since this model is designed to fit (drug, ADR) pairs, we fit the model on each drug successively.

NonparaSCCS is introduced in [GWF17] which uses splines to model both the effect of drug exposure and age. We use the same R package and settings as the ones described for SmoothSCCS.

We did not include [Sch+16] as we have not found any open source implementation of this work. We have not tried to use [Sim+13] since we do not have precise priors on relevant risk periods in the context of ADR screening.

Table II.4.1 – Comparison of SCCS methods with ConvSCCS. MSCCS is introduced in [Sim+13], ESCCS in [Sch+16], while SmoothSCCS and NonParaSCCS are respectively introduced in [GWF16; GWF17]. Regularised models are constrained to avoid overfitting, the constraint being controlled by hyper-parameters. The models can either fit multiple features at a time or be limited to study only one feature at a time. We do not consider SmoothSCCS and NonParaSCCS as able to study multiple features properly since only one feature can be regularised.

Algorithm	Regularised	Multiple features	Multiple exposures	Flexible effect
MSCCS	yes	yes	yes	no
ESCCS	no	no	accumulated	yes
SmoothSCCS	yes	no	no	yes
NonParaSCCS	yes	no	no	yes
ConvSCCS	yes	yes	yes	yes

II.4.1 Simulations

The performances of our model against SmoothSCCS and NonParaSCCS are compared in a simulation study. For this purpose, multivariate longitudinal exposures and outcomes are simulated, with a correlation structure between exposures.

Simulation of longitudinal features. The simulation of correlated longitudinal features is a difficult task, for which we use Hawkes processes, see [HO74], which is a family of counting process with an autoregressive intensity, see Section II.E of the Supplementary Material for more details. Our simulation setting has been chosen so that it generates correlated exposures, as it is the case with actual exposures from the LOD considered in this paper.

Simulation of relative risks. We assume that all simulated adverse outcomes can take place at most 50 time intervals after the first exposure. We consider two sets of relative risk profiles from [GWF17] and [AF03]. These sets are precisely described in Section II.E of the Supplementary Material, and contain several types and shapes of risks profiles.

Simulation of outcomes. We simulate $m = 4000$ patients' exposures over $K = 750$ time intervals. The observation periods are set to $[0, b_i]$, where $b_i = K - e_i$ and e_i are is from an exponential distribution with intensity $1/250$. Intensities λ_{ik} are set to zero for all $k > b_i$. The outcomes are simulated according to a multinomial distribution $\text{Mult}(1; p_{i,0}, \dots, p_{iK})$ where $p_{ik} = \lambda_{ik} / \sum_{k'=1}^{K} \lambda_{ik'}$.

Sensitivity analysis. We perform extensive simulations to test the robustness of our model to bias sources specific to EHR data using the following scenarios, namely not-at-random missing data, noisy timestamps, missing longitudinal features, see Section II.E of the Supplementary Material for more details.

Performance measure. The performance of the different models is computed using the mean absolute error (MAE) between the estimated relative incidence and the true risk profile, see Section II.E of the Supplementary Material for details. For both sets of relative risk profiles, we simulate $m = 4000$ cases and simulate 100 datasets for each scenario.

Results. Boxplots representing the MAE distribution over the 100 simulated datasets are represented in Figures II.4.1 and II.4.2. In Set 1 of relative exposures, which is an "easy" setting (4 features and 8 non-zero correlations, see Supplementary Material), the gain resulting from studying several drugs at a time seems to be balanced by the bias resulting from using step functions when fitting smooth risk profiles. Indeed, as shown by Figure II.4.1, the estimation errors of drug exposures relative risks are similar across the three considered models. For the baseline estimation, NonParaSCCS performs better than ConvSCCS and SmoothSCCS since the use of splines results in a better approximation than the step functions with six groups of age.

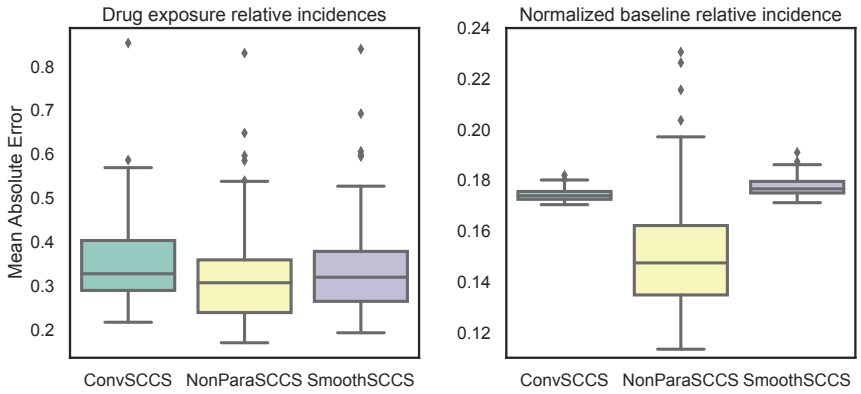


Figure II.4.1 – Simulations results using Set 1 or risk profiles (see Figure II.E.2) with $m = 4000$. The boxplots represent the distribution of mean absolute error as defined in Section II.4.1, computed over 100 simulated populations. *Left:* MAE distribution of the drug exposure relative incidences. *Right:* MAE distribution of the baseline relative incidences, constrained so that their integral is equal to one.

In Set 2 of relative exposures, which is a more difficult setting (14 features, with 24 non-zero correlations, see Supplementary Material), ConvSCCS outperforms both SmoothSCCS and NonParaSCCS. We observe in Figure II.4.2 that fitting the effect of several drugs at the same time and using our penalisation provides a better estimation accuracy than NonParaSCCS and SmoothSCCS, the improvement being larger for the estimation of drugs exposures risks profiles than for the baseline. This illustrates the benefits of fitting several drugs at the same time in the context of an SCCS model. Figure II.4.3 gives the run times of all three procedures. ConvSCCS seems to scale better than both SmoothSCCS and NonParaSCCS when fitting a large number of feature such as $d = 14$ on $m > 2000$ cases. In small studies, however, when $d = 4$ for example, SmoothSCCS is the fastest algorithm, while NonParaSCCS is overall slower than the two other algorithms. According to its improved performance and scalability when studying several drugs, ConvSCCS seems to be a useful model for ADR screening on LODs.

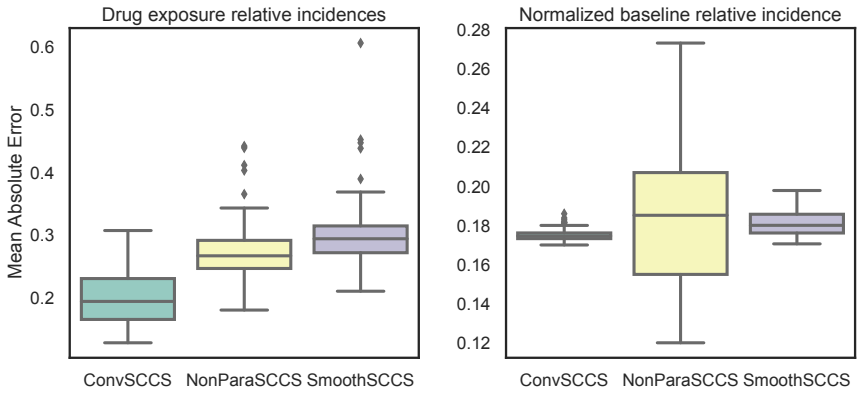


Figure II.4.2 – Simulations results using Set 2 or risk profiles (see Figure II.E.3) with $m = 4000$. The boxplots represent the distribution of mean absolute error as defined in Section II.4.1, computed over 100 simulated populations. *Left:* MAE distribution of the drug exposure relative incidences. *Right:* MAE distribution of the baseline relative incidences, constrained so that their integral is equal to one.

The sensitivity analysis shows that the model is robust to small to moderate noise in timestamps, but its performances degrade with large noise (see Figure II.E.4 in Supplementary Material). With large noise, the model over-penalizes (with the group-Lasso), resulting in a constant relative incidence for each feature. In such situations, reducing the granularity might help to reduce the noise level, but it might also dilute the risk. The model does not seem particularly sensitive to not-at-random missing data or to slightly correlated missing features, see Figures II.E.5, II.E.6 and II.E.7 in

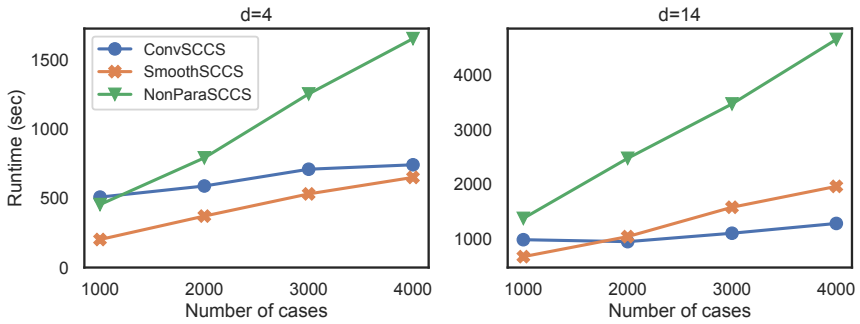


Figure II.4.3 – Run times of ConvSCCS, SmoothSCCS and NonParaSCCS described Section II.4 for 1000, 2000, 3000, 4000 cases. *Left:* run times on 4 features. *Left:* run times on 14 features. As SmoothSCCS and NonParaSCCS can only handle one feature at a time, we report the time required to fit them on each studied feature while ConvSCCS is fitted on all the features simultaneously. For each model, a fit includes cross-validation of the hyper-parameters and estimation of confidence bands.

Supplementary Material for more details.

II.4.2 Application on data from the French national health insurance information system

We investigate the association between glucose-lowering drugs and the risk of bladder cancer in France with data from the SNIIRAM/PMSI database. Using similar data, a significant association between pioglitazone (glucose-lowering drug) and bladder cancer was reported in [Neu+12]. As a result of this study, the use of pioglitazone was suspended in France in June 2011. Note that other studies, such as [Lew+15] did not conclude to a significant effect on this particular association.

The SNIIRAM/PMSI database. The data was extracted from the French national health insurance information system (*Système National d'Information Inter-régimes de l'Assurance Maladie* (SNIIRAM), see [Tup+10b]) linked with the French hospital discharge database (*Programme de Médicalisation des Systèmes d'Information* (PMSI), see [ATI] website), in the context of a research partnership between Ecole Polytechnique and CNAMTS. The full SNIIRAM/PMSI database is an SQL database containing hundreds of tables built around medical reimbursements of more than 53 million people (its size is between 150 and 200 TB). Our team set up a 15 nodes Spark cluster and developed an ETL (Extract Transform Load) pipeline to transform the data into a single patient-centric table that can be used to build features that feed

various statistical inference algorithms.

Cohort, ADR and expositions definitions. The cohort includes patients covered by the general insurance scheme aged 40 to 79 years on 2006/12/31 who filled at least one prescription for a glucose-lowering drug in 2006. The end of the observation period was set on 2009/12/31. The glucose-lowering drugs investigated are insulin, metformin, sulfonylurea, pioglitazone, rosiglitazone, and other oral hypoglycemic agents.

All patients with any bladder cancer-related events in the six months before follow-up start have not been included. So although the depth of the data was 48 months, the cohort was followed for up to 42 months. The considered outcomes can then be treated as incident cases. We use the same definition for the bladder cancer outcome as in [Neu+12], which adds particular procedures to a hospital discharge diagnosis (ICD-10-C67). The cohort contains 1699 patients with bladder cancer. Note that we have roughly 400 cases missing in comparison to [Neu+12], and less history prior to follow-up to filter prevalent cases, due to French data regulation imposing patients information to be deleted after ten years. More details about cohort structure can be found in Table II.E.1 in supplementary materials.

We consider that patients are exposed to a molecule as soon as they purchase a drug containing this molecule. Once a patient has been exposed, she is considered as exposed until the end of her follow-up. There is a potential bias concerning drug exposures. Indeed, diabetic patients use hypoglycemic agents continuously. As a result, exposure starting dates might exhibit noisy timestamps.

ConvSCCS. We apply ConvSCCS to the cohort with bladder cancer, and use the smallest available granularity: 30-days time intervals based on calendar time and consider a risk window of 24 months. We do not use age-related features and consider its effect to be part of patients' baseline cancelled out during the model estimation.

ConvSCCS Assumptions (2) and (3) (see Section II.2.1) are considered to be unviolated for the following reasons. Bladder cancer times and the observation period do not seem to be correlated: among 1699 cases, we observe only 52 censoring times occurring between 2 and 35 months after outcome times. We thus consider, following [Far+11], that the model performance should not be affected. Assumption (3) is valid when working on rare non-recurrent events [FW06]. Using the same outcome definition as [Neu+12], we find 1699 cases over roughly 1.5 million patients. The construction of this outcome also constrains it to occur only once over the 4 years of observation. It considers successive bladder cancer events as multiple recordings of the same cancer, which is sensible regarding the study length. Hence, it seems reasonable to consider bladder cancer as a rare, non-recurrent event, and thus, Assumption (3), following [FW06].

Concerning Assumption (1), we observe a small shift in the distribution of new exposures to Insulin and Others after the outcome date among the studied cases. If this shift is caused by the outcome time, it would violate the feature exogeneity Assumption (1). However, it is also a characteristic of diabetes care pathways in France: diabetic patients often begin their treatment with metformin, and then switch to another group of molecules later on if it fails to regulate their diabetes, and so on, with Insulin being one of the last options. Timestamps might also be noisy, since diabetic patients are continuously exposed to hypoglycemic agents. As a result, most of the patients in the cohort are already exposed at the beginning of the follow-up (30% to 70% of the exposures start at beginning of the follow-up depending on the molecule). This might introduce noise in the timestamps, as we do not really know for how long patients have been exposed, and we have shown in our simulation study that ConvSCCS is sensitive to noisy timestamps (see the sensitivity analysis in Section II.4). However, this problem met in the data is standard would affect any other method similarly. Despite these unavoidable problems with the data, ConvSCCS is able to detect, as explained below the stronger adverse effect of pioglitazone pointed out in [Neu+12].

We selected the best hyper-parameters γ_{tv}^* and γ_{gl}^* using stratified 3-fold cross-validation, with random search. Bootstrap confidence intervals are computed with 200 bootstrap samples obtained with the parametric bootstrap on the unpenalised likelihood. We refit the model using the support of the parameters obtained with the penalised procedure before using the bootstrap. Cross-validation and 95% bootstrap confidence intervals computation took 188 seconds using a single thread of an Intel Xeon E5–2623 v3 3.00 GHz CPU.

Study in [Neu+12]. The exposure to pioglitazone is measured in terms of duration from the first purchase, categorised in three intervals. The exposure to other lowering drugs starts when the patient buys the drug two times in a 6-month window, setting the beginning of the exposure in the middle of the 6-month window. A multivariate Cox model to estimate the bladder cancer hazard ratios for glucose-lowering drugs (time-dependent) exposures, adjusted for age using groups of 5 years and gender, was used.

Results. The estimated relative incidences and 95% bootstrap confidence intervals for all investigated glucose-lowering drugs are represented in Figure II.4.4. Thanks to the penalisation used in ConvSCCS, the estimated relative incidences and confidence intervals are piecewise constant on large steps: this is particularly interesting since it allows to detect only significant variations of the relative risks.

As shown in Figure II.4.4 we recover a strong positive association between pioglitazone and the risk of bladder cancer, which consistently increases over time

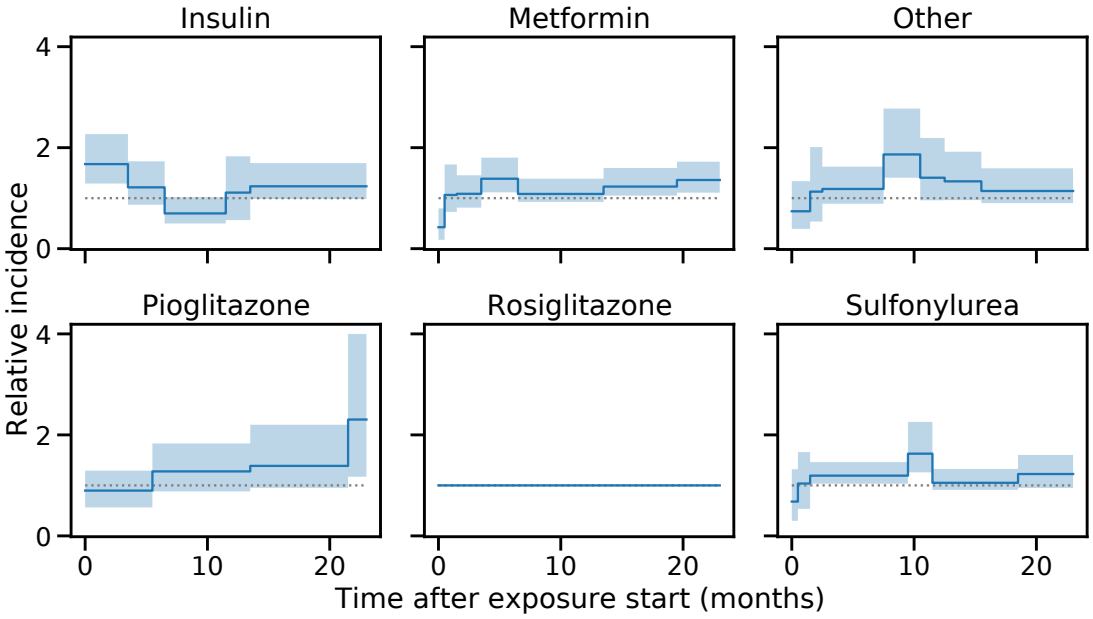


Figure II.4.4 – Estimated relative incidences of glucose lowering drugs on the risk of bladder cancer. Blue curves represent the estimated relative incidences $k = 0, \dots, 23$ months after the beginning of exposure. Light blue bands represent 95% confidence intervals estimated by the parametric bootstrap, with 200 bootstrap samples.

from 6 to 24 months after exposure start. Since our model estimates longitudinal effects of exposures, we compare ourselves with the duration of pioglitazone use estimates in [Neu+12] in the paragraph below. Our model estimated a hazard ratio of 0.89 ([0.56, 1.29]) for the first 6 months after exposure to pioglitazone, 1.27 ([0.88, 1.83]) between 6 and 14 months after pioglitazone exposure start. [Neu+12] found a hazard ratio of 1.05 ([0.82, 1.36]) for pioglitazone exposure of less than 12 months. For exposure greater than 12 months, they estimated a hazard ratio of 1.34 ([1.02, 1.75]) and 1.36 ([1.04, 1.79]) while our model found 1.39 ([0.95, 2.2]) from 14 to 22 months after pioglitazone exposure start and 2.3 ([1.17, 4.0]) from 22 to 24 after pioglitazone exposure start. Our results regarding pioglitazone are thus overall consistent with [Neu+12].

The comparison for other hypoglycemic agents hazard ratios is more difficult since [Neu+12] does not estimate longitudinal risks for these molecules. While other hypoglycemic agents are non statistically significant in [Neu+12], our model cancels out the effect of rosiglitazone and find the other molecules non statistically significant during most of the lags after exposure start. However, sulfonylurea and “other” have

positive significant estimates from lags 9 to 11, as well as insulin from lag 0 to 5. The shape of these three curves suggests there might be some colinearity issues between these three features, since the magnitude of their relative incidence curves seems to either match or be of opposite signs and magnitude in similar lag values. Metformin seems to be non-significant overall, despite few coefficients suggesting a positive association. While these results are not a perfect match to [Neu+12], they show that our model might be useful when exploring quickly large sets of molecules with a reduced amount of data preprocessing, even when the conditions are sub-optimal (noisy timestamps, possible feature endogeneity, and feature colinearity). Indeed, in contrary to [Neu+12] approach, our methodology is scalable in the number of drugs since it doesn't require the same precise preprocessing work.

II.5 Conclusion

In this paper, we introduced ConvSCCS, a multivariate SCCS method with a flexible risk formulation. Our approach is based on a discrete-time version of the SCCS model [Far95], enjoying its scalability and automatic adjustment for time-independent confounders. Classical SCCS models usually require a precise prior definition of risk windows, which might be unavailable in an adverse drugs reaction screening context. Our model circumvents this problem by modelling exposures-related relative incidences with low-granularity step functions, on which we apply total-variation penalisation. ConvSCCS shows improvements in precision and computational speed compared to the state-of-the-art in moderate to high dimension. It relies on the usual SCCS assumptions: the outcomes are distributed as a Poisson process, conditionally to the longitudinal features that are assumed to be exogenous, and observation periods of the subjects should be independent from outcome times. ConvSCCS exhibits robustness to a departure from the above mentioned SCCS assumptions, as illustrated in extensive numerical experiments, but remains sensitive to a large noise level in timestamps, which can be problematic depending on the data source quality.

An other important advantage of ConvSCCS is its ability to consider exposures to multiple drugs simultaneously in the model. ConvSCCS is, therefore, a flexible tool which could be used for future ADR screening based on LODs. An application of ConvSCCS is provided on a cohort of diabetic patients studied in [Neu+12], and it is able to recover the ADR detected by the authors.

Acknowledgements

This research benefited from the CNAMTS-Polytechnique research partnership, and from the Data Science Initiative of Ecole Polytechnique. We thank Aurélie Bannay, Hélène Caillol, Joël Coste, Claude Gissot, Anke Neumann, Jérémie Rudant, and Alain Weill for their insights and their help with the understanding of the SNIIRAM database. We also would like to thank the research engineers who have been working on this partnership, first of all Youcef Sebiat for the code review and revision work and also Firas Ben Sassi, Prosper Burq, Xristos Giastidis, Sathiya Kumar, Daniel de Paula.

Authors' contribution

This work was co-authored by Maryan Morel, Emmanuel Bacry, Stéphane Gaïffas, Agathe Guilloux and Fanny Leroy. MM conducted the experiments. All the authors contributed to the research plan, drafted the manuscript, reviewed the manuscript and approved the final version.

Appendix

II.A Likelihood in SCCS models

From [Dal03], the Poisson likelihood of a single patient i can then be written as

$$L_i(n_i; t_i|X_i) = e^{-\int_{a_i}^{b_i} \lambda_i(s,X_i)ds} \prod_{k=1}^{n_i} \lambda_i(t_{ik}, X_i),$$

and the total number of events $n_i = N_i([a_i, b_i])$ follows a Poisson distribution

$$\mathbb{P}(n_i|X_i) = \frac{(\int_{a_i}^{b_i} \lambda_i(s,X_i)ds)^{n_i}}{n_i!} e^{-\int_{a_i}^{b_i} \lambda_i(s,X_i)ds}.$$

Conditioning the likelihood by the total number of events and on the covariates histories leads to the SCCS likelihood of a patient history

$$\begin{aligned} L_i(t_i|n_i, X_i) &= \frac{L_i(n_i; t_i|X_i)}{\mathbb{P}(n_i|X_i)} \\ &= \frac{e^{-\int_{a_i}^{b_i} \lambda_i(s,X_i)ds} \prod_{k=1}^{n_i} \lambda_i(t_{ik}, X_i)}{e^{-\int_{a_i}^{b_i} \lambda_i(s,X_i)ds} \frac{(\int_{a_i}^{b_i} \lambda_i(s,X_i)ds)^{n_i}}{n_i!}} \\ &= n_i! \prod_{k=1}^{n_i} \frac{\lambda_i(t_{ik}, X_i)}{\int_{a_i}^{b_i} \lambda_i(s, X_i)ds}, \end{aligned}$$

where we used the convention $\prod_{k=1}^{0} \ldots = 1$ (i.e., the likelihood is equal to 1 if a patient does not have any event, namely $n_i = 0$). The likelihood of m patients can therefore be expressed, up to constants independent on the intensities, as

$$L \propto \prod_{i=1}^{m} \prod_{k=1}^{n_i} \frac{\lambda_i(t_{ik}, X_i)}{\int_{a_i}^{b_i} \lambda_i(s, X_i)ds}.$$

II.B Discrete time SCCS

We assume that, for $i = 1, \ldots, m$, the intensity $\lambda(t, X_i(t))$ is constant over time intervals $I_k = (t_k, t_{k+1}]$, $k = 1, \ldots, K$ that form a partition of the observation interval $(a, b]$. We choose I_k to be of constant length, chosen without loss of generality equal to 1. In practice, we use the smallest granularity allowed by data. We therefore can assume that $(a_i, b_i] \cap I_k$ is either $\emptyset$ of I_k for all $i = 1, \ldots, m$, and $k = 1, \ldots, K$, which means that the observation period of each individual is a union of intervals I_k. The discrete-time likelihood writes

$$\begin{aligned} L(t_i; n_i|X_i) &= \exp\Big(\sum_{k=1}^{K} \int_{I_k} \log(\lambda(s, X_i(s)))\mathrm{d}N_i(s) - \sum_{k=1}^{K} \int_{I_k} \lambda(s, X_i(s))\mathrm{d}s \Big) \\ &= \exp\Big(\sum_{k=1}^{K} \log(\lambda_{ik}) N_i(I_k) - \sum_{k=1}^{K} \lambda_{ik} \Big), \end{aligned}$$

where $\lambda_{i,k}$ is the value of $\lambda(t, X_i(t))$ for $t \in I_k$, where $N_i(I_k) = \int_{I_k} dN_i(t)$ and where we used $\int_{I_k} dt = 1$ and the fact that $N_i(I_k) = 0$ and $\lambda_{ik} = 0$ if $I_k \cap (a_i, b_i] = \emptyset$. The distribution of the total number of events for patient i is given by

$$\mathbb{P}(n_i|X_i) = \frac{\big(\int_{a_i}^{b_i} \lambda(s, X_i(s))\mathrm{d}s\big)^{n_i}}{n_i!} e^{-\int_{a_i}^{b_i} \lambda(s, X(s))\mathrm{d}s} = \frac{\big(\sum_{k=1}^{K} \lambda_{ik}\big)^{n_i}}{n_i!} e^{-\sum_{k=1}^{K} \lambda_{ik}},$$

which leads to

$$L(t_i|n_i, X_i) = \frac{L(n_i; t_i|X_i)}{\mathbb{P}(n_i|X_i)} = \frac{\exp\big(\sum_{k=1}^K \log(\lambda_{ik}) N_i(I_k) - \sum_{k=1}^K \lambda_{ik}\big)}{\frac{(\sum_{k=1}^K \lambda_{ik})^{n_i}}{n_i!} e^{-\sum_{k=1}^K \lambda_{ik}}}$$
$$= n_i! \prod_{k=1}^K \Big(\frac{\lambda_{ik}}{\sum_{k'=1}^K \lambda_{ik'}}\Big)^{N_i(I_k)},$$

where we use the convention $0^0 = 1$, i.e. only the exposition period $(a_i, b_i]$ contributes to the likelihood, and since once again $N_i(I_k) = \lambda_{ik} = 0$ whenever $I_k \cap (a_i, b_i] = \varnothing$. Then, defining $y_{ik} := N_i(I_k)$, the previous equation can be rewritten as

$$L(y_{i1}, \dots, y_{ik}|n_i, X_i) = n_i! \prod_{k=1}^K \Big(\frac{\lambda_{ik}}{\sum_{k'=1}^K \lambda_{ik'}}\Big)^{y_{ik}}.$$

II.C Numerical implementation

We use the state-of-the-art SVRG algorithm from [XZ14] for the minimization of our penalized negative log-likelihoof. It is known to typically lead to faster convergence than quasi-newton algorithms, such as L-BFGS-B, see [LN89], while allowing to deal with non-smooth objectives. Solving (8) requires to compute the proximal operator (see [Bac+12] for a definition) of pen(θ). This can be done very fast numerically: pen(θ) can be separated into two separate proximal operators for total-variation and group-Lasso, see [Zho+12]. The proximal operator of group-Lasso is explicit and given by group soft-thresholding, see [Bac+12], while the prox of total-variation is not, but can be computed very efficiently with the fast algorithm from [Con13].

II.D Software

Our model is implemented in the `Tick` library, see [Bac+17a], which is a `Python` library focused on statistical learning for time dependent systems. It is open-source and available at `https://github.com/X-DataInitiative/tick`. The implementation is done in `C++`, with a Python API, and is thoroughly documented at `https://x-datainitiative.github.io/tick/`.

II.E Simulations details

About the simulation of longitudinal features. Let us give some details on the way we simulated longitudinal features using Hawkes processes.

Namely, we simulate dates of purchases $\{t_i^j\}_{i\geq 1}$, of drugs $j = 1, \dots, d$ using a multivariate Hawkes process $N_t = [N_t^1 \cdots N_t^d]$, for $t \geq 0$, where $N_t^j = \sum_{k\geq 1} \mathbb{1}_{t_k^j \leq t}$ for any $t \geq 0$. The process N_t is a multivariate counting process, whose components N^j have intensities

$$\lambda_t^j = \mu_j + \sum_{j'=1}^{d} \sum_{k\geq 1} A_{j,j'} \alpha \exp(-\alpha(t - t_k^{j'})) \tag{II.8}$$

for $j = 1, \dots, d$. This corresponds to a Hawkes process with so-called *exponential kernels*. The $\mu_j \geq 0$ are called *baselines* intensities, and correspond to the exogenous probability of being exposed to drug j. In the matrix $A = [A_{j,j'}]_{1\leq j,j'\leq d}$, called the *adjacency matrix*, the entry $A_{j,j'} \geq 0$ quantifies the impact of past exposures to drug j' on the exposition intensity to drug j and $\alpha > 0$ is a memory parameter. A single matrix A is simulated for the whole population, but a new one is generated in each round of simulation. Recalling that the simulated events t_i^j correspond to the purchase date of drugs (this is the only information available in the LOD described in Section II.4.2 below), we consider that a patient is exposed to a drug j at time t_1^j.

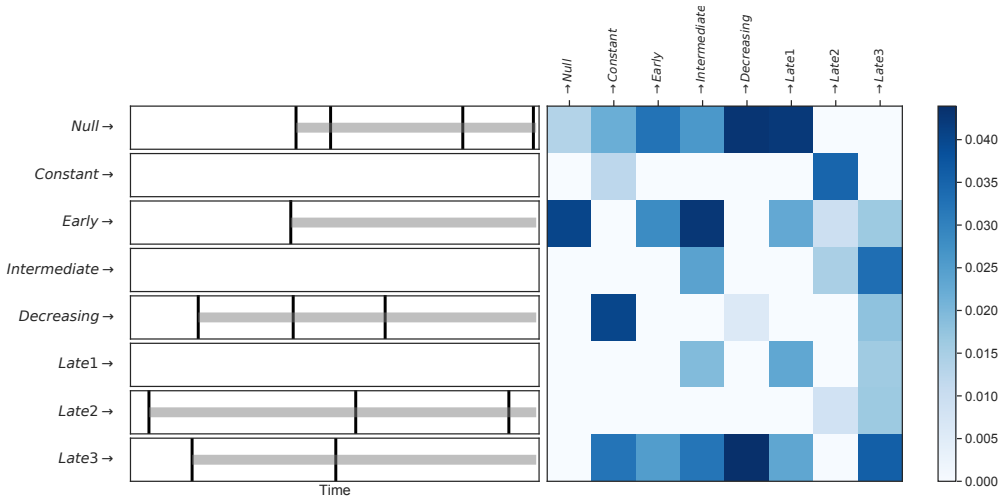


Figure II.E.1 – *Left*: example of simulated dates of drugs purchases (vertical black lines). Exposure starts at the date of the first purchase (gray horizontal lines). *Right*: an example of generated adjacency matrix A for longitudinal feature simulation using the Hawkes process. This matrix encodes the correlation structure of exposures to drugs. To ease the reading, this figure represents the transposed adjacency matrix $A^\top$. For example, a purchase of a 'null' drug increases the probability of purchasing a 'Late3' drug. In *Left* and *Right* we simulate potential exposures to 8 drugs, each of them have a different risk profile (named "null", "constant", "early", etc.). These profiles are described in Section II.E of the supplementary material.

We sample μ_j using a uniform distribution on $[0, 5 \times 10^{-3}]$, which will produce unbalanced exposures in the simulations, and set $\alpha = 0.5$. The diagonal entries $A_{j,j}$ are equal to μ_j, and we sample q non diagonal entries using a uniform distribution $[0, 5 \times 10^{-3}]$, while setting all other entries to zero. We set $d = 4,\ q = 8$ in the first experiment, $d = 14,\ q = 24$ in the second experiment. We normalize A so that its largest singular value is 0.1, in order to ensure that the process does not generate too much events. Simulation is achieved through the thinning technique, see [Oga81], and easily achieved using the `tick` library, see [Bac+17a]. An example of simulated matrix A is illustrated in Figure II.E.1. Our simulation setup allows to generate realistic exposures, since it can reproduce the following phenomena that are typically observed in LODs:

- Depending on the drug, a patient using it has a higher probability to use it again in the future: this is quantified by the value of the diagonal entries $A_{j,j}$;
- Some drugs are often purchased at the same time, because of the underlying medical treatment: a patient using drug j' has a higher probability to use drug j, which is quantified by $A_{j,j'}$;
- Most of the patients use only a subset of all available drugs during their observation period, so several entries of A are zeros.

About the risk profiles. We provide below a precise description of the two sets of risks profiles considered in our simulations.

- Set 1 of risk profiles corresponds to the ones used in [GWF17], and are represented in Figure II.E.2. We use a lower order of magnitude than [GWF17], resulting in risk profiles with maximum between 1.5 and 2 matching the magnitudes encountered in our application. The first risk profile is unimodal, the second has a constant effect, two others are continuously decreasing. In this set, risk profiles length matches $p = 50$.
- Set 2 of risk profiles represent effects described in [AF03]: rapid, early, intermediate, late and delayed effect, see Figure II.E.3, with magnitudes similar to Set 1. It contains the four shapes from Set 1, and a null risk, a unimodal risk with a sharp drop and three shapes of continuously increasing risks. This set contains risk profiles for which the optimal risk period is smaller than $p = 50$. We generate 7 features with "Null" risk profile, and one feature for each other risk profile, resulting in 14 features.

Following [GWF17], we use for all patients a baseline relative incidence given by $\phi(t) \propto 8 \sin(.01t) + 9$ (see the right-hand side of Figure II.E.2) which can be thought as the effect of age whenever each patient has the same age.

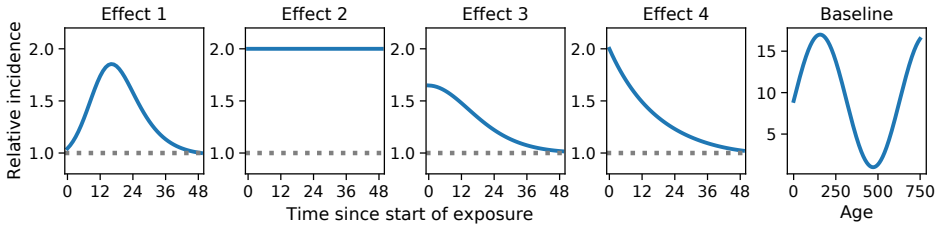


Figure II.E.2 – *Left.* Set 1 of relative risk profiles. The effect of these relative incidences starts with the exposure, and last 50 time periods. The effect on the individual risk is multiplicative. *Right.* Temporal baseline used in all simulations.

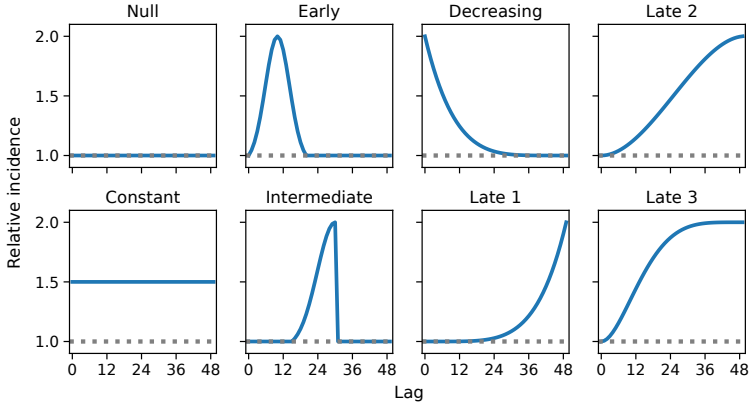


Figure II.E.3 – Set 2 of relative risk profiles. The effect of these relative incidences starts with the exposure, and last at most 50 time periods. The effect on the individual risk is multiplicative. Note that we include 7 features with the "null" risk profile in addition to one feature with each other risk profile in Set 2, to simulate datasets in which there are irrelevant features.

About the performance measure. As defined in Section II.3.1, relative incidence of drug j, k periods after exposure start is defined as $\hat{r}_k^j = \exp(\hat{\theta}_k^j)$, $k = 0, \dots, p$ in our model. In [GWF16; GWF17], the estimated relative incidence is defined as $\hat{r}_k^j = \hat{\theta}^j(k) > 0$ for $k = 0, \dots, p$, see Equation (II.3). Denoting the ground truth relative incidence r^*, the MAE is given by

$$MAE = \frac{1}{dK} \sum_{j=1}^{d} \sum_{k=1}^{K} |r_k^{j*} - \hat{r}_k^j|.$$

Since we assume that all the patients are affected by the baseline in the same way, its order of magnitude cannot be properly estimated by the models. In order to be

able to compare baseline relative risks, we constrain their integral to be equal to one as [GWF17].

Regarding the sensitivity analysis We consider three scenarii for the perturbations:

1. *Not-at-random missing data.* We simulate a hidden feature correlated to other longitudinal features using a Hawkes process. For each simulated timestamp of this feature, patients' data is masked for a time period of length drawn uniformly in $[0, max\_length]$. Outcomes are simulated using the non-perturbed data, while exposures provided to the model are computed using the censored timestamps. We vary the max_length parameter to assess the sensitivity of ConvSCCS to this perturbation.

2. *Noisy timestamps* We add a random noise draw uniformly in $[0, max\_length]$ to the features. We vary the max_length parameter to assess the sensitivity of ConvSCCS to this perturbation.

3. *Missing longitudinal feature.* We simulate more features. Outcomes are simulated taking these features into account, while they are not used when fitting the model. In a first scenario, we vary the number of hidden features at constant relative incidence magnitude. In a second scenario, we vary the the relative incidence magnitude of two hidden features.

All these experiments were performed using 2000 simulated cases.
Figures II.E.4 to II.E.7 present the results of this sensitivity analysis.

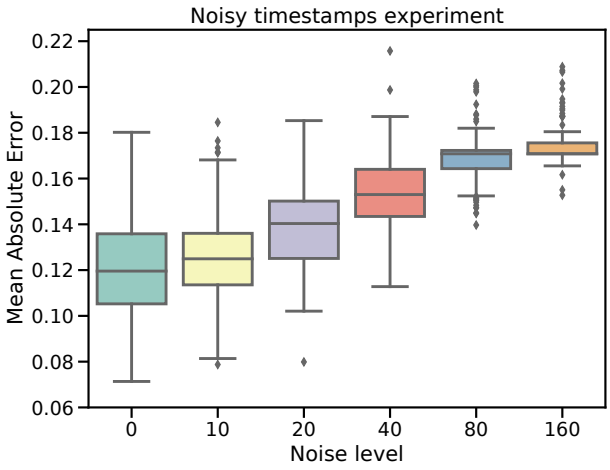


Figure II.E.4 – Sensitivity analysis adding a noise drawn uniformly in $[0, noise\_level]$ to features timestamps using Set 2 of risk profiles (see Figure II.E.3) with $m = 2000$. The boxplots represent the distribution of mean absolute error as defined in Section II.4.1, computed over 100 simulated populations.

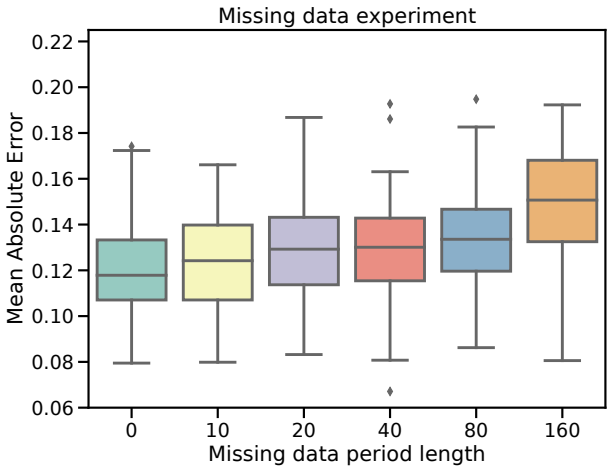


Figure II.E.5 – Sensitivity analysis simulating not-at-random missing data. A hidden feature timestamps are simulated in the same way as regular features. At each time of this feature, other features data is masked for a period of *missing data period length*. Other features are simulated using Set 2 or risk profiles (see Figure II.E.3) with $m = 2000$. The boxplots represent the distribution of mean absolute error as defined in Section II.4.1, computed over 100 simulated populations.

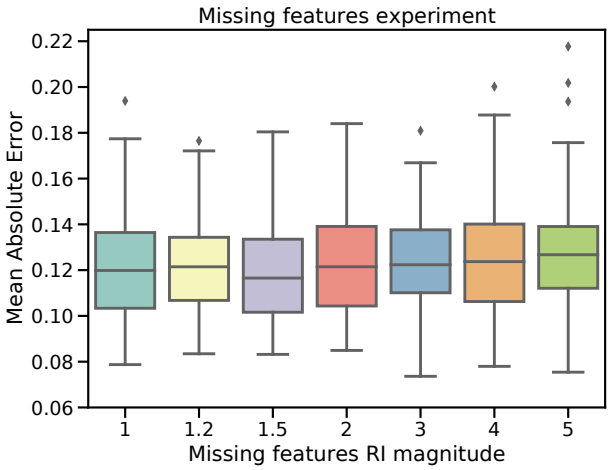


Figure II.E.6 – Sensitivity analysis simulating missing longitudinal features. Simulations results using Set 2 or risk profiles plus two hidden features (see Figure II.E.3) with $m = 2000$. The order of magnitude of hidden features relative incidence vary from 1 to 5. The boxplots represent the distribution of mean absolute error as defined in Section II.4.1 computed over 100 simulated populations.

Figure II.E.7 – Sensitivity analysis simulating missing longitudinal features. Simulations results using Set 2 or risk profiles plus hidden features using similar risk profiles (see Figure II.E.3) with $m = 2000$. The number of hidden features vary from 0 to 14. The boxplots represent the distribution of mean absolute error as defined in Section II.4.1 computed over 100 simulated populations.

Regarding the cohort structure. Table II.E.1 compile demographic and drug consumption data of the cohort used in this study.

Table II.E.1 – Demographics and glucose-lowering drug use of the cohort of French diabetic patients covered by the general insurance scheme (i.e., in the SNIIRAM database), aged 40–79 years and followed from 2006 to 2009.

Characteristics	**Overall study population**
N	1,428,637
Men	771,647
Bladder cancers	1,699
Age (years)	
40–44	54,989
45–49	94,986
50–54	160,388
55–59	238,611
60–64	238,394
65–69	223,721
70–74	232,100
75–79	185,448
Number of patients exposed to glucose-lowering drugs (a patient can appear in several lines)	
Insulin	343,912
Other OHA	434,352
Rosiglitazone	157,346
Metformin	1,043,967
Pioglitazone	158,619
Sulfonylurea	836,572
Number of patients exposed to a single glucose-lowering drug (each patient appears at most in a single line)	
Insulin	102,021
Other OHA	34,927
Rosiglitazone	2,239
Metformin	208,331
Pioglitazone	4,486
Sulfonylurea	145,509

CHAPTER III

SCREENING ANXIOLYTICS, HYPNOTICS, ANTIDEPRESSANTS AND NEUROLEPTICS FOR BONE FRACTURE RISK AMONG ELDERLY.

A NATION-WIDE DYNAMIC MULTIVARIATE SELF-CONTROL STUDY USING THE SNDS CLAIMS DATABASE.

Background and Purpose Existing screening works provide point estimates for drug-outcome pairs risk assessment. We propose a flexible approach based on dynamic risk estimates to support alert generation while providing additional information on risk qualification (delay, shape) and LOD-specific biases. We illustrate this approach by studying the longitudinal effect of anxiolytic, hypnotic, antidepressant, and neuroleptic molecules on fractures using SNDS, a French large healthcare claims database.

Methods We follow French new users who were 65 y.o. or older in 2014 for up to four years. We use ConvSCCS, a flexible longitudinal model based on self-control case series. This model alleviates several observational claims data issues and does not require precise assumptions on risk timings. The presence of eventual indication biases is assessed by estimating dynamic pre-exposure relative risks.

Results Pre-exposure risk estimates suggest the presence of confounding by indication in anxiolytics, hypnotics and neuroleptics estimates, while it is not the case for antidepressants. Tricyclic antidepressants exhibit lower relative risk than other antidepressants. Zolpidem relative risk is consistently higher than Zopiclone across all sensitivity analyses.

Conclusion This approach complements existing screening methods as well as clinical or observational risk quantification studies by providing granular and dynamic risk estimates for many molecules using a single model. It could be used to map molecules and adverse events, pointing out the presence of eventual biases or associations for further investigation.

Keywords: *Large Observational Database, Dynamic analysis, Adverse drug reaction screening, Elderly, Fracture, Anxiolytics, Hypnotics, Antidepressants, Neuroleptics.*

Key Points

- Our screening methodology estimates pre and post-exposure dynamic relative risk to go beyond existing approaches relying on point estimates.
- We perform a screening on all anxiolytic, hypnotic, antidepressant and neuroleptic (AHAN) molecules for bone fracture using a single flexible model on four years of SNDS data.
- Our results shed lights on the dynamics associating AHANs to fractures in claims data, and are consistent with the fragmented, existing literature.

III.1 Introduction

Observational healthcare data volume and availability increased over the last years carrying the hope of improving Adverse Drug Reactions (ADRs) screening. ADRs screening is defined as "drug-related risk identification and alert generation" in Bezin et al. [Bez+17]. In this setup, there are no precise hypotheses regarding suspected ADRs, and the screening algorithm is designed to assess several drug exposures effect on an identified event of interest.

Observational data peculiar characteristics and ADRs various dynamics [GAB15] make this task particularly difficult. Indeed, observational healthcare data, such a Electronic Healthcare Records (EHRs) and claims data are often collected for economic concerns rather than epidemiological purposes. As a result, such data reflect as much the data recording process and care providers economic considerations as patients' health status [HA13], and can mis-represent some populations due to eventual geographic or healthcare affordability constraints [Mad+14].

First screening approaches exploiting claims data relied on re-purposing classical models to study many drug-outcome *pairs* (see examples in Ryan et al. [Rya+13b]

or Thurin et al. [Thu+20]). They produced simple statistical models and designs, contrasting with the tailored analyses observational data would require to handle its specific biases [Mad+14]. More recent works alleviate some of these issues, either thanks to careful designs [Tol+17] or mixed effects bayesian models [Gib+19]. However, these approaches produce point estimates relying on strong temporal dynamics assumptions.

In this work, we use ConvSCCS [Mor+20], a recent flexible conditional Poisson model (also known as Self-Control Case Series – SCCS) which does not require precise assumptions on risk timings, provide dynamic risk estimates, and allows for estimating many molecules associated risk within a single model. Combined with a careful study design, we aim to provide detailed information on the underlying dynamics of several drug exposures association with a target event.

We use this methodology to screen potential associations between anxiolytic, hypnotic, antidepressant and neuroleptic (AHAN) molecules use and fractures among the elderly using data from the *Système National des Données de Santé* (SNDS, formerly known as SNIIRAM), a French large observational database containing most of the population's healthcare claims and hospital discharges. While SNDS is often used to perform drug safety studies [Bez+17; Tup+17a], it has been used only very recently to perform ADR screening Thurin et al. [Thu+20].

AHANs and fracture risk associations have already been investigated at different levels of granularity and scopes in numerous clinical and observational studies. Fractures among the elderly are a prominent public health issue as they are associated with high morbidity and mortality [Dea+10; Vri+18]. They can be caused by reduced bone mineral density or postural instability [All+05], both of which might be influenced by the use of AHANs. Meta-analyses, such as Seppala et al. [Sep+18a] or Woolcott et al. [Woo+09], reviewed a very large corpus of papers investigating such associations. These works highlight how hard establishing a broad mapping of fracture risk and molecules association can be, as most studies scope is limited to a single drug or drug class. To raise the level of evidence, Seppala et al. [Sep+18a] calls for studies investigating pharmacological subgroups rather than large drug classes, as well as duration effects, which is precisely what we are trying to achieve.

We aim to assess the capabilities of our approach to estimate the duration effects of all the molecules belonging to AHAN classes using a single statistical model while assessing the presence of eventual database-specific biases.

III.2 Materials and methods

III.2.1 Data Source

This study is based on data from the SNDS, a nation-wide claims and hospital discharge database containing of 98.8% of French population healthcare reimbursements [Bez+17; Tup+17a]. When working on adult subjects, this database has virtually no turnover apart from subjects moving abroad, resulting in almost no censorship due to loss in follow-up. Besides basic demographic information (gender, birth date), it contains timestamped outpatients dispensed drugs, procedures and long-term diseases, and inpatients procedures and diagnoses [Tup+17a].

III.2.2 Study design

The study was conducted as a self-control study on new-user data. To enter the cohort, subjects had to be (1) covered by the universal health insurance coverage, which is the case for 98.8% of France inhabitants [Tup+17a], (2) 65 y.o. or older on 1 January 2015, (3) receive their first outpatient target drug prescription at least 365 days after study start on 1 January 2014 to prevent prevalent users or to provide a sufficient wash-out delay. Restriction to 65+ y.o. patients result in a more homogeneous population in terms of professional activity (retirees), behaviour (response to a fall, sport practice) and characteristics (bone density), all of which might have an effect on fracture risk. Cohort entry was 1 January 2014 when all these conditions were met. Cohort exit was defined as (1) death; or (2) end of the study period, 31 December 2017.

As we performed the analysis with an SCCS model, only cases were used to fit the statistical model. We used a one-year time-window (entry condition (3)) as a control period common to all subjects. To ensure we do not bias fracture risk during the control period, patients with a history of fracture during this first year were not excluded.

III.2.3 Case definition

We extracted fracture events following the algorithm presented in Bouyer et al. [Bou+20]. Fractures from public and private hospitalisations were extracted using International Classification of Diseases 10^{th} revision (ICD10) codes of stay diagnoses, while non-hospitalised fractures were extracted based on outpatients' medical procedures using CCAM (French Common Classification of Medical Acts) codes matching plastered or orthopedic immobilisation and fracture reduction. Extracted events were categorised by fracture site and severity. Fracture severity was computed as an index ranging from 1 to 4: (1) there was no hospital admission due to the fracture, (2) fractured patients were hospitalized but did not have surgery, (3) fractured patients were hospitalised and a surgery dedicated to the fracture was performed and (4) indicates that the patient died during the hospital stay following its fracture.

As a fracture can generate multiple events in the healthcare system, we considered same-site fracture events within a 4-month window following the first event as the initial fracture subsequent events. To be consistent with our statistical analysis (see Section III.2.5), we only studied the first fracture event of each subject.

III.2.4 Exposure definition

AHANs dispensations were identified using codes from the Anatomical Therapeutic Chemical (ATC) classification system. We selected psychotropic drugs excepted those commonly used for acute psychiatric indications or anaesthesia, listed in Appendix, Tables III.A.1 to III.A.4.

We focused on fractures resulting from falls, and thus on short or mid-term adverse reactions to AHANs use [All+05]. We used binary indicators of exposure starting times in combination with a flexible model, which has been shown to be the best modeling assumption when the ADRs' true forms and prescribed doses are unknown [GAB15], which is the case in SNDS [Bez+17; Tup+17a].

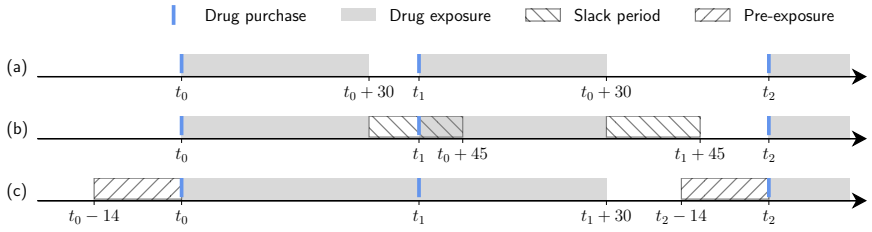


Figure III.2.1 – Illustration of drug exposures computation. Exposures are assumed to last for 30 days (90 days for large drug packaging) after drug purchases (i). A slack period is added (ii) to account for slight variability in drug purchasing dates. Exposures which overlap with other exposures or other exposures' slack period are merged (iii). Once the merging is done, 14-day pre-exposure periods are added before each exposure starting points (iii).

Exposure start and end time were computed as follows: (1) drug exposure was considered to start at the drug dispensation date. (2) subjects were assumed to use at most one drug dose per day, resulting in 30 or 90-day exposures depending on the drug packaging. (3) When a given exposure plus a 15-day slack period overlapped with another exposure, they were merged, thus considered as a single exposure. This slack period was used to account for small drug adherence variability across patients. The size of this delay was chosen to fit most regular users while remaining conservative, given the short half-life of the molecules under study [Wis+17]. (4) Pre-exposure risk periods covering 14 days preceding exposure start were defined to account for eventual confounding by indication.

As a result, patients could have been exposed multiple times to each molecule. Exposure and pre-exposure define two distinct sets of risk periods for each molecule, the reference period being non-exposed time. Note that pre and post-exposure risk periods were not stopped during hospitalisations as our statistical model requires risk periods of a minimal fixed length (see Section III.2.5). Exposure computation is illustrated Figure III.2.1.

III.2.5 Statistical Analysis

We aim to detect eventual association between fracture risk and drug exposure without precise prior knowledge on risk timing or shape. To do so, we used ConvSCCS [Mor+20], a flexible conditional Poisson model (also known as Self-Control Case Series – SCCS) allowing to estimate longitudinal variations of the risk resulting from each exposure. As in any SCCS model, patients are their own control and the model is robust to non-longitudinal confounding [Mor+20]. The new-user cohort design allowed us to use the first year of study as the control period, which also alleviated the prevalent user bias stated in Madigan et al. [Mad+14]. Such design performs well on claims databases [MSR13; Rya+13b] which do not contain enough information on patients' demographics and life habits to find matching control patients [Tup+17a].

ConvSCCS estimates a longitudinal relative risk curve (RRC) for each studied molecule, modeling the risk dynamics during risk periods. The length of those RRCs can be at most the minimal length of the considered exposure type [Mor+20]. Our risk periods definition resulted in 14-day pre-exposure and 30-day post-exposure RRCs for each molecule. We used daily data, which is the lowest temporal granularity available in SNDS. To avoid obtaining noisy estimates, RRCs variation were penalised as described in Morel et al. [Mor+20]. Group Lasso penalisation cancels out RRCs close to one, providing feature selection. Total Variation penalisation controls RRCs discontinuities, leading the model to automatically select an optimal risk periods partition, as illustrated in Figure III.2.2.

Reported 95% confidence intervals were estimated using parametric bootstrap as described in Morel et al. [Mor+20], and statistical power was approximated as described in Wasserman [Was13].

III.2.6 Sensitivity and subgroup analysis

To assess the robustness of our results, the following analyses were carried out:

(1) *Single fractures*: exclusion of patients with more than one fracture or hospital admission with fracture diagnosis over the observation period. Multiple fractures might reflect patients affected by osteoporosis or fractures resulting from

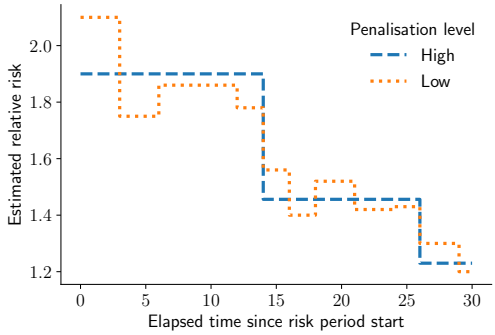


Figure III.2.2 – Illustration of the Total Variation penalisation effect. Assuming a risk period starting at 0 and lasting for 30 days, ConvSCCS will estimate a 30-day piece-wise constant relative risk curve. The total size of the jumps is controlled by the level of Total Variation penalisation. A low (resp. high) level of penalisation results in more (resp. less) detailed relative risk curves, illustrated by the orange small dashes (resp. blue long dashes) curve. The aim of the model fitting algorithm is to reach a good balance between the detail level and the smoothness of the estimated relative risk curves.

severe crashes. Approximately 16% of the patients experienced two or more fractures during the observation period (see Table III.3.2).

(2) *65–85 y.o.*: analysis on the 65–85 y.o. subgroup. The average age of the elderly moving into retirement homes is 85 years old [Mul17], in which case drug purchase data might be less precise [Tup+17a].

(3) *Epileptic patients exclusion*: this condition is an indication for some of the molecules under study thus leading to an eventual confounding [Dea+10; Sep+18b].

(4) *Gender*: analysis of men and women subgroups. Bone density variations related to gender might lead to differences in fracture risks [Sch+04]. Differences between men and women subgroups regarding associations between fractures and antidepressants have been reported in Vermeeren [Ver04].

(5) *Additional control drugs*: add exposures to other drugs which might have an influence on fractures. Additional molecules or group of molecules were opioids, proton pump inhibitors [Sep+18b]; loop diuretics, digitalis, digoxin [Vri+18]; and anti-hypertensive drugs [Dea+10]. Exposures to these molecules or groups of molecules were simply used as additional features, not to filter prevalent users.

(6) *Fracture severity*: restriction to a subset of fractures depending on their severity. We restricted the fracture definition to severity 1, 1 or 2 and 3. The severity 4 fractures subgroup was not considered as a very high correlation between death date and event date violates a ConvSCCS assumption (see discussion in Section III.4.2).

(7) *Specific fracture sites*: restriction to hip, wrist or spine fracture.

Sensitivity analysis resulted in numerous RRCs, the all-fracture scenario being the reference analysis. To ease results reading and interpretation, we report the differences between reference RRCs and those estimated in sensitivity experiments using the following method: (i) We consider only RRCs for which at least one value is significantly different from reference estimates based on 95% bootstrap confidence intervals. (ii) Some sensitivity analysis experiments result in smaller datasets with only a few patients exposed to some molecules. RRCs estimated with low power ($< .2$) are excluded from the comparison, to avoid considering estimates to be "unstable" when a poor estimate is caused by the lack of data. (iii) We then compute the mean relative error,

$$\bar{re} = T^{-1} \sum_{t=1}^{T} \frac{\theta_t - \theta_t^{ref}}{\theta_t^{ref}},$$

between the RRCs $\theta = (\theta_1, \dots, \theta_T)$ selected in step (i) and the corresponding reference estimates $\theta^{ref} = (\theta_1^{ref}, \dots, \theta_T^{ref})$, where T is the length of the considered risk period.

III.2.7 Software

Cases and exposures extraction from SNDS was performed using the SCALPEL3 library [Bac+20], while the statistical model was fitted using the Tick library [Bac+17b]. Both libraries are open-source and freely available. The code used to produce the results presented in this paper is available at `https://github.com/X-DataInitiative/AHANScreening`.

III.3 Results

From a source population containing 13,762,623 patients of 65 y.o. or older, we extracted 126,567 fracture cases among 1.969,587 patients exposed to AHANs between 1 January 2015 and 31 December 2017 but not exposed in 2014 (see flow chart Figure III.3.1). An overview of the studied cohort demographic characteristics is presented in Tables III.3.1 and III.3.2. Exclusion of subjects not exposed to AHANs did not result in major demographic changes (see Table III.3.2). Restriction of the cohort to cases only led to an over-representation of 85+ y.o. patients (57% vs. 32%),

women (72% vs. 58%), antidepressants (44% vs. 31%) and neuroleptics (21% vs. 7%) users as compared to the study population (see Table III.3.1).

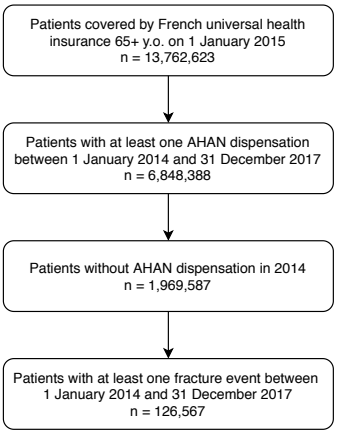


Figure III.3.1 – Flow chart of the study population. Subjects were first selected based on their consumption of Anxiolytics, Hypnotics, Antidepressants or Neuroleptics (AHANs). The cohort was then restricted to fracture cases.

Table III.3.1 – Demographics and Anxiolytics, Hypnotics, Antidepressants or Neuroleptics use by fracture adverse event. The first column reports the number of 65+ y.o. new users patients cohort in the listed subgroups. The second (resp. third) column reports the number of patients with at least one fracture (resp. hip fracture) during the observation period within the study cohort and the associated subgroups. Figures in parenthesis represent the relative size (%) of a subgroup with respect to its population (n).

	Study population (%)		Fracture cases (%)		Hip fracture cases (%)	
n	1,969,587	(100.0)	126,567	(100.0)	46,699	(100.0)
Women	1,136,695	(57.7)	90,340	(71.4)	33,370	(71.5)
Age (years)						
[65–70[	576,854	(29.3)	17,123	(13.5)	2,755	(5.9)
[70–75[	395,883	(20.1)	14,872	(11.8)	3,373	(7.2)
[75–80[	366,227	(18.6)	19,858	(15.7)	6,102	(13.1)
[80–85[	316,085	(16.1)	27,895	(22.0)	11,053	(23.7)
[85–90[	204,989	(10.4)	27,544	(21.8)	13,060	(28.0)
[90–95[	93,195	(4.7)	16,252	(12.8)	8,685	(18.6)
[95–100[	13,594	(0.7)	2,585	(2.0)	1,411	(3.0)
> 100	2,760	(0.1)	438	(0.4)	260	(0.6)
Exposed to anxiolytics	1,381,068	(70.1)	83,581	(66.0)	30,013	(64.3)
Exposed to hypnotics	519,548	(26.4)	38,291	(30.3)	14,682	(31.4)
Exposed to antidepressants	603,511	(30.6)	50,937	(40.3)	20,493	(43.9)
Exposed to neuroleptics	144,303	(7.3)	18,464	(14.6)	9,554	(20.5)

Our model produces a set of two relative risk curves (RRCs) for each molecule. The post-exposure RRC express the evolution of the relative risk $t = [0, \dots, 30]$

Table III.3.2 – Demographics of fractured patients. The first column reports the number of 65+ y.o. SNDS French patients who experienced a fracture during the observation period for each population subgroup. The second column reports the number of these patients who were also exposed to one of the Anxiolytic, Hypnotic, Antidepressant or Neuroleptic (AHAN) molecules under study. The last column reports similar figures when restricting the population to new users (i.e. patients who were not users during the first year of study). Figures in parenthesis represent the relative size (%) of a subgroup with respect to its population (n).

	Fractured 65+ y.o. patients (%)		Patients exposed to AHANs (%)		2015 new-users (%)	
n	729,647	(100.0)	513,303	(100.0)	126,567	(100.0)
with hip fracture	263,402	(36.1)	194,827	(38.0)	46,699	(36.9)
with multiple fractures	112,162	(15.4)	84,175	(16.4)	20,342	(16.1)
Women	549,795	(75.4)	398,847	(77.7)	90,340	(71.4)
Age (years)						
[65–70[	98,289	(13.5)	58,179	(11.3)	17,123	(13.5)
[70–75[	83,654	(11.5)	54,025	(10.5)	14,872	(11.8)
[75–80[	110,948	(15.2)	77,931	(15.2)	19,858	(15.7)
[80–85[	153,538	(21.0)	113,022	(22.0)	27,895	(22.0)
[85–90[	159,192	(21.8)	119,223	(23.2)	27,544	(21.8)
[90–95[	101,589	(13.9)	75,146	(14.6)	16,252	(12.8)
[95–100[	18,644	(2.6)	13,303	(2.6)	2,585	(2.0)
> 100	3,793	(0.5)	2,474	(0.5)	438	(0.3)

periods after exposure start. The coefficient associated with $t = 0$ represent the instantaneous relative risk, i.e. the risk associated to the day of exposure initiation. Pre-exposure RRCs describe relative risk dynamics during the two weeks preceding exposure start, with $t = [-1, \dots, -14]$. More details regarding RRCs interpretation are provided in Section III.4. Estimated relative risk curves (RRCs) are compiled Figure III.5.1 to III.5.8. Please note that the longitudinal variation of the risk is controlled by the model penalisation and is not the result of explicit assumptions.

III.3.1 All fractures

Fractures RRCs before and after the drug exposure start are compiled in Figures III.5.1 to III.5.4. With the exception of diazepam, anxiolytics post-exposure RRCs (Figure III.5.1) are either flat, between 1.2 and 1.7, or decreasing over 30 days, starting between 1.8 and 2.5 to fall between 1 and 1.5. Diazepam post-exposure RRC is also decreasing, starting much higher at 6.0 to level at 2, thirty days after exposure start. Other than buspirone, all anxiolytics pre-exposure risks are almost always significantly higher than 1, with three distinct profiles: increasing pre-exposure RRC (diazepam), decreasing RRCs starting between 1.9 and 3 before plummeting to 1 in 5 to 10 days and constant RRCs between 1.1 and 2.7.

Hypnotics RRCs (Figure III.5.2) are either constant, with similar profiles, or U-shaped in the case of zopiclone and zolpidem, for which post-exposure relative risk decreases in 10 to 15 days before increasing slightly. Zolpidem post-exposure relative risks are higher than zopiclone's. Both molecules have sharp decreasing pre-exposure RRCs. Temazepam and bromides RRCs are non-significant (at 95%).

Antidepressants RRCs (Figure III.5.3) can be separated in two groups. The first group, consisting of selective serotonin reuptake inhibitors (SSRIs), serotonin and norepinephrine reuptake inhibitors (SNRIs) and tetracyclic antidepressants (TTCAs) exhibit decreasing post-exposure RRCs for most of the molecules. Those RRCs range from 1.5 to 2.5. SSRIs relative risk stays high (> 1.5) during 30 days following exposure start, while RRCs of TTCAs are non-significant 12 days after exposure start for mianserin, and 22 days after exposure start for mirtazapine. Among these molecules, fluoxetine, paroxetine, and duloxetine have a constant relative risk of similar magnitude, while fluvoxamine, moclobemide and milnacipran exhibit very low relative risks (between 1 and 1.2). These molecules pre-exposure RRCs tend to be either decreasing or flat, but are always lower than 1 in the few days preceding the exposure starting time.

The second group, made of tricyclic antidepressants (TCAs) and other antidepressants has constant post-exposure RRCs, either ranging from 1.5 to 2.0 or non-significant. Those molecules have flat, non-significant pre-exposure RRCs, excepted amitriptyline and agomelatine. Amitriptyline's pre-exposure relative risks start at around 1.5 in the 14 to 7 days before exposure starts, and non-significant 7 to 0 days before exposure start, while agomelatine's is constant, lower than one.

Most of neuroleptics RRCs are constant over the considered risk periods. However, cyamemazine, haloperidol, and risperidone stand out, with post exposures starting between 1.7 and 2.4, decreasing towards a relative risk between 1.2 and 1.5, 30 days after exposure start. These three molecules exhibit similar pre-exposure RRCs, starting around 1.5 and decreasing to a non-significant relative risk or slightly less than in the case of cyamemazine. Other neuroleptics RRCs are either non-significant or constant, with a relative risk ranging from 1.1 to 1.6. They exhibit non-significant pre-exposure RRCs excepted for loxapine and tiapride, whose pre-exposure RRCs start around two, to non-significant levels a few days before exposure starting time.

Hip fracture

Restricting the study population to hip fractures resulted in 46,699 cases. RRCs are represented Figures III.5.5 to III.5.8. The smaller number of cases seems to result in a slight loss of power, leading to non-significant RRCs in some cases (such as clomipramine), flatter RRCs (escitalopram) or wider confidence intervals (mirtazapine).

Diazepam post-exposure RRC is now significantly lower, but still high starting at

2.5 to level-off around 1.5 after one week. Estimated RRC increases for dosulepin with a RR of 1.6 (1.2 previously) and duloxetine with RRC around 2.4 (1.7 previously). Other molecules post-exposure RRCs are overall stable with respect to the all-fracture analysis. A noticeable decrease in pre-exposure relative risk can be observed, especially among anxiolytics (Figure III.5.5) and hypnotics (Figure III.5.6), resulting in several non-significant pre-exposures RRCs. It is not the case for neuroleptics (Figure III.5.8) for which pre-exposure does not vary nor increases in the case of haloperidol.

Sensitivity analysis

The relative differences between RRCs of the reference all-fracture analysis and those estimated in sensitivity experiments are reported in Figure III.5.9. RRCs excluded from the comparison due to low power ($< .2$) are hatched on the graphical representation. Detailed relative risk estimates for these sensitivity analyses are provided in Supplementary Materials, Figures III.B.26 to III.B.37. Pre-exposure and post-exposure RRCs are overall stable with respect to population design variations (experiments 1 to 4 defined in Section III.2.6), with more variability when restricting the population to men. Adding control drugs slightly shifted downwards anxiolytics and Zopiclone and Zolpidem post-exposure RRCs, and their pre-exposure RRCs even more. Changing target event definition using specific fracture sites or fracture severity introduce some variations in post-exposure estimates, and heavily affects pre-exposure relative risks. Defining the target event as wrist fracture or low severity fractures leads to a smaller population ($n = 9,722$), resulting in low power estimates.

III.4 Discussion

III.4.1 Key results

We presented a methodology designed to perform large scope screening studies using claims data. It relies on using ConvSCCS [Mor+20], a flexible SCCS model, with binary drug exposures. The flexibility of the model allows to estimate risk dynamics without prior assumption on the risk shape, and prevent risk dilution when the risk window is larger than the actual risk [Mor+20]. Binary exposures encode the starting times of exposures, which has been show to be an optimal choice when combined with a flexible model in situations where reliable prior knowledge is unavailable [GAB15]. A new-user design prevents the risk dynamics estimated by the model to be affected by prevalent drug use, starting before the observation period. In addition to post-exposure risk, we also estimate pre-exposure flexible risk

curves to highlight the presence of eventual biases linked to specific care pathways or database-specific biases.

We applied this approach on observational claims data from SNDS [Tup+17a]. To our knowledge, only [Thu+20] performed ADRs screening on the full scale SNDS database, using a methodology based on point estimates, similarly to Ryan et al. [Rya+13a]. Dynamic risk estimation produced more detailed information than binary answers sought by screening algorithms based on point estimates. Rather than pursuing a fully automated alert generation system, our approach fosters human interpretation of data-mined patterns.

We evaluate our screening approach by studying AHANs for bone fracture risk. Some works on ADRs screening such as Ryan et al. [Rya+13b] evaluated the performance of their methodology by comparing their results to an adverse drug reaction database [Rya+13a] containing established positive and negative association. While this approach is appealing because of its convenience, the reliability of such datasets have been criticized by Hauben, Aronson, and Ferner [HAF16] as some associations appear to have been mis-classified. In place of this evaluation scheme, we evaluate our screening methodology by comparing our results to existing works on AHANs. We compare our relative risk estimates and dynamics to meta-analyses and to results obtained with other methodologies.

Overall, results from the main analysis (presented in Figures III.5.1 to III.5.8) seem consistent with meta-analyses compiled in Table III.4.1. Our estimates for benzodiazepines were slightly higher than pooled odds ratios (ORs) when considering all studies [Blo+13; Sep+18a; Woo+09], but they were close to pooled ORs restricted to studies providing adjusted ORs [Sep+18a]. Note that grouping individual molecules into large categories might result in an averaging effect, smoothing out risk estimates [Ver04]. Results for each molecule class are discussed below.

III.4.2 Limitations

Confounding by indication SNDS does not allow to make a distinction between a drug effect and its indication, which might bias the estimated associations [Tup+17a]. We used pre-exposure RRCs to assess biases resulting from LOD-specific care pathways. Such use of pre-exposure risk windows is not new [NN19; Pra+11; Req+20], especially when using flexible dynamic models. A pre-exposure RRC above one might indicate the presence of indication bias. In this case, the molecule is likely to be prescribed in reaction to the target event occurrence. On the contrary, pre-exposures RRCs below one might highlight protective environments such as hospitals, preventing patients to experience the studied event. It highlights situations where patients are prescribed a molecule during a hospital stay and buy the said molecule at discharge. Both effects can be mixed when the studied event is likely to cause an

Table III.4.1 – Meta-analyses and reviews on fracture and Anxiolytics, Hypnotics, Antidepressants or Neuroleptics (AHANs) eventual association. (*CI: Confidence Interval, OR: Odds Ratio, RR: Relative Risk, TCA: tricyclic antidepressant, SSRI: selective serotonin reuptake inhibitor*)

Study	Methodology	Target event	Results
Bloch et al. [Blo+13]	Meta-analysis	Falls	Heterogeneity across reviewed studies on hypnotics. Pooled ORs (95% CI): benzodiazepines 1.61 (1.35–1.93), hypnotics 1.53 (1.40–1.68), antidepressants 1.59 (1.43–1.75), antipsychotics 1.37 (1.16–1.61).
Graham et al. [Gra+11]	Review	Hip or femur fractures	Antipsychotic drug use OR varying from 1.2 to 3 depending on studies. No dose relationship, schizophrenia is a confounding factor.
Seppala et al. [Sep+18a]	Meta-analysis	Falls	Heterogeneity across reviewed studies. Pooled ORs on all studies (95% CI): benzodiazepines 1.38 (1.17–1.63), antidepressants 1.35 (1.28–1.42), antipsychotics 1.43 (1.15–1.77). Pooled ORs on adjusted studies: benzodiazepines 1.81 (1.05–3.16), TCAs 1.41 (1.07–1.86), SSRIs 2.02 (1.85–2.20), antipsychotics 1.54 (1.28–1.85).
Vestergaard [Ves09]	Review	Fractures	Only Amitryptiline is associated with fracture risk among TCA (RR = 1.3), SSRIs are associated with an increased risk overall, especially in the 14 first days post-exposure (RR = 6.3 tapering off to 1.3 after 42 days).
Woolcott et al. [Woo+09]	Meta-analysis	Falls	Heterogeneity across reviewed studies. Pooled ORs (95% CI): benzodiazepines 1.57 (1.43–1.72), hypnotics 1.47(1.35–1.62) antidepressants 1.68 (1.47–1.91), neuroleptics 1.59 (1.37–1.83).

hospitalization, resulting in a pre-exposure RRC starting above one and decreasing sharply (see zolpidem for example). This interpretation was consistent with sensitivity analysis experiments restricting fracture definition to a given severity level. Sensitivity analysis results summarised Figure III.5.9 showed that pre-exposure RRCs were lower than the all-fracture study when considering only fractures requiring surgery (severity 3), and conversely when considering fractures which did not require surgery (severity 1 or 2) or did not require hospitalisation (severity 1). While pre-exposure estimates do not prevent biases resulting from such dynamics, they allow for contextualising screening results thus helping to design further confirmation studies.

Comorbidities and unobserved confounding Potential biases linked to impaired vision, low BMI, physical or instrumental disability, cognition impairment, Parkinson's disease or rheumatic diseases [Dea+10] have a slow evolution. They might result in almost-static individual effects, which should not have a significant impact on our results thanks to self-controlled designs ability to ignore unmeasured non-longitudinal biases [Far95]. Results were robust to the exclusion of 1678 epileptic patients as shown in Figure III.5.9 (see Figures III.B.9 to III.B.41 for more details). Depression might also be considered as a comorbidity [Dea+10] and lead to an eventual confounding by indication but antidepressant pre-exposure RRCs did not suggest the presence of short-term indication biases.

Fracture definition We studied the first fracture event of each case rather than studying recurrent fractures. Results were robust to the exclusion of the 20,342 patients who experienced more than one fracture over the observation period (see Figure III.5.9). We also controlled for an eventual measurement bias by restricting the analysis to hip, wrist, or spine fractures. Due to a small number of cases, restriction to wrist fracture resulted in low power estimates for many molecules. Restriction to spine fractures resulted in higher anxiolytic and hypnotic pre and post-exposure RRCs indicating a stronger indication bias for these molecules in this subgroup, especially in the case of diazepam. Conversely, restriction to hip fractures resulted in lower pre-exposure RRCs and a lower diazepam post-exposure RRC, while other post-exposure RRCs were comparable with the reference analysis.

Population selection Restriction to women resulted in slightly lower post-exposure RRCs and lower pre-exposure RRCs, and conversely for men, which might be explained by differences in fractures site repartition between men and women as a larger proportion of men experienced spine (+6%) or ribs (+3%) fractures. Post-exposure RRCs were robust to the exclusion of 85+ y.o. subjects.

Model assumptions ConvSCCS relies on three assumptions: (1) exposure times are independent of outcome times, (2) the observation period of each patient is independent of its outcome times, (3) outcome times follow a Poisson process conditionally on the exposure times. Assumption (3) is verified by design as we consider only the first fracture event. Assumption (2) was assessed by looking at the distribution of the gaps between event times and time to death [Whi+18]. In our reference analysis, 7.35% of the cases event times were eventually correlated to time of death, which seems reasonable. Excluding 85+ y.o. patients or high severity fractures reduced this proportion while producing in similar results (see Figure III.C.1 in Supplementary materials for more details). Assumption (1) is not likely to be verified as pre-exposure RRCs suggested the presence of confounding by indication. While pre-exposures help to capture this effect at least partially, we cannot rule out an eventual bias in post-exposure RRCs.

Statistical power When our model estimation procedure leads to flat RRCs for some molecules, it does not necessarily mean that the risk is actually flat. It can be the result of a lack of statistical power when too few patients are exposed to a molecule and risk variations are small. RRCs estimated to be constant over the risk period can be interpreted as an "average risk" over the risk period, similarly to regular SCCS models. In addition, low statistical power might lead to non-significant relative risks for some molecules. In these cases, we conclude to an absence of detection rather than an absence of risk. While using a large observational database brings more cases, and thus more power, longitudinal screening of many molecules is data intensive. Indeed, it relies on the estimation of many parameters as it cannot take advantage of precise prior assumptions regarding RRCs shapes.

Scalability While the statistical model has no particular issue in terms of scalability, relying on a new-user design might result in too few available subjects when working on many molecules with only a few years of data.

Long term ADRs The study described in this paper focuses on short term associations, and cannot detect long term associations in its current form. However, the statistical model used in this study can do so when adopting different exposure and risk window definitions [Mor+20].

III.4.3 Interpretation

Anxiolytics All anxiolytics exhibited a positive association with fracture risk, either constant or decreasing over time. Decreasing of post-exposure RRCs follow two scenarios: (1) the molecules can be prescribed for short-duration treatments

(< 30 days, including a drug withdrawal phase [LTD09]), in this case, the effect might disappear at the end of the treatment. The decline of the risk can be smooth, as withdrawal might be implemented by slowly decreasing the doses patients are using. (2) There can be some form of tolerance as described in Vermeeren [Ver04]. The tolerance can be pharmacokinetic when it results from a lesser absorption with use, pharmacodynamic when the response to the molecule decreases with use, or behavioural when the brain gradually learns to overcome drug-induced impairments. As the estimated curves express an averaged effect over the studied population, they can express dynamics resulting from both scenarios. Several anxiolytic pre-exposure RRCs (such as clorazepate potassium or clobazam) indicated a potential indication bias [Fai15]. In some cases such as oxazepam or etifoxine, a sharp decrease in pre-exposure risk highlighted the presence of care pathways in which fracture is probably followed by a hospital admission of two to ten days and a subsequent anxiolytic prescription. Such pathways might occur when anxiolytics are used to manage anxiety following the event leading to the fracture, or patient agitation in the case of fractures that cannot be immobilised such as head or torso fractures. This was confirmed by sensitivity analysis experiments restricted to specific fracture severity levels (see Figure III.5.9; and Figures III.B.26 to III.B.37 in Supplementary Materials for more details). Similar dynamics of anxiolytic prescriptions following car crashes or fracture events were also observed in other studies [Gib+09; Req+20].

Our risk estimates were consistent with meta-analysis adjusted pooled Odds Ratios (ORs) [Sep+18a], excepted for diazepam, for which estimated RRC was considerably higher than other benzodiazepines in the all-fracture analysis. While a similarly high diazepam RR has also been found in studies focusing on car crashes [Gib+09], it is likely to be the result of a strong indication bias in our case. Indeed, pre-exposure RRC of diazepam was much higher (peaking at 9) than what can be observed for other anxiolytics. Results from our sensitivity analysis showed that this bias was particularly important when restricting the study to spine fractures with a peak RR around 42 (see Figure III.5.9; or Figure III.B.42 in Supplementary Materials for more details). This strong association can be explained by prescriptions of diazepam aiming to control spasticity after spinal cord injury [CL07; MTC91]. These biases almost disappeared when restricting the study to hip fractures. Adding control drugs (experiment 5) also resulted in a lower diazepam pre-exposure RRC, suggesting a potential co-prescription bias with opioids (see Figure III.5.9; or Figure III.B.23 in Supplementary Materials for more details). Note that such co-prescription bias might be also affect other anxiolytics and hypnotics pre-exposure RRCs (see Figure III.5.9). As a result, diazepam relative risk was probably overestimated when spine fractures were included in cases definition and opioids exposure were not controlled in the model.

Hypnotics Hypnotic benzodiazepines exhibited lower RRCs than anxiolytic benzodiazepines which can be consistent with their recommended use, at bedtime. Their side effects such as drowsiness or dizziness might be less likely to lead to fractures than anxiolytic benzodiazepines which are used during daytime [Ver04].

However, zopiclone and zolpidem association with fracture risk was similar or even higher than anxiolytics. Those two molecules have been extensively investigated as a group ("Z-drugs") or individually [Tre+17]. However, they seem to have been compared only once in Pierfitte et al. [Pie+01] when it comes to fracture association. Pierfitte et al. [Pie+01] also find odds ratio twice as high for zolpidem compared to zopiclone's, but they relied on a very small sample (less than 70 patients exposed to each molecule, among whom there are at most 15 cases) which might result in large confidence intervals and possibly low power.

Despite variations in pre-exposure risk, zolpidem post-exposure RRC was always found to be higher than zopiclone's across all sensitivity analysis experiments. This might be explained by zolpidem's sharper plasma concentration-time curve compared to zopiclone [Dro04] and more impairing reported side effects, such as "strong visual disturbances and changes in perception" for zolpidem while "tiredness, dry mouth, metallic taste" were reported for zopiclone [Dro04]. This result might also indicate a misuse of zolpidem [MFL16] in France.

Antidepressants Antidepressants RRCs were consistent with the results presented in reviews [Sep+18a; Ves09]. The increase in relative risks after exposure was smaller among tricyclic antidepressants (TCAs) than selective serotonin reuptake inhibitors (SSRIs), serotonin norepinephrine reuptake inhibitors (SNRIs) and tetracyclic antidepressants (TTCAs). We also observed decreasing RRCs for citalopram, escitalopram, sertraline, mianserin, mirtazapine, and venlafaxine similarly to Hubbard et al. [Hub+03]. However, we estimated a constant RRC for amitriptyline while Hubbard et al. [Hub+03] found a decreasing RR. Amitriptyline pre-exposure RRC was above one, indicating a potential confounding by indication, perhaps resulting from its use in neuropathic pain management, especially after spinal cord injury [AJ17]. Aside from amitriptyline, pre-exposure RRCs were either non-significant or below 1, which suggest post-hospitalization prescriptions but no indication bias. Such care pathways are likely as SSRIs [Mor+13] or mirtazapine [Hon+07] might be prescribed in reaction to post-myocardial infarction for example.

Neuroleptics Similarly to [Pra+11], neuroleptic pre-exposure RRCs suggested the presence of indication bias which might be explained by the use of some neuroleptics in neuropathic pain management [NS19]. Our post-exposure risk estimates were consistent with Pratt et al. [Pra+11], while our pre-exposure risks were slightly lower. We have not found a clear pattern relating estimated RRCs to neuroleptic sub-classes

or their mechanism of action.

Hip fracture Restricting the study to hip fractures resulted in a sharp decrease of pre-exposure RRCs with respect to the reference analysis. It can be explained by a larger proportion of hospitalized fractures when restricting the cases to hip fractures, shifting the whole RRC downward. However, pre-exposure RRCs decrease was not as important when restricting the study to hospitalized fractures (severity 3), suggesting another source of pre-exposure RRC diminution. This subgroup age repartition is slightly skewed towards older subjects (see Table III.3.1) for whom benzodiazepine prescription are not recommended by the French Health Authority (HAS) [San07], probably resulting in a lesser confounding by indication.

III.5 Conclusion

We showed that our approach mixing cautious study design and an easy-to-tune flexible statistical algorithm can be used to produce large scope results highlighting eventual associations and indication or database-specific dynamic biases. Our approach is easy to implement as it relies on open-source, scalable libraries. It does not require much fine-tuning, it can handle large populations and many molecules, it relies on a few ascertainable assumptions and provides easily interpretable results. Our cohort construction and exposure and event definitions help to mitigate some of the database biases, without injecting over-restrictive prior knowledge to retain model plasticity. Flexible dynamic pre and post-exposure relative risk curves provide information on healthcare pathways, helping to highlight large observational databases specific biases. While the properties of our approach make it robust to some biases and can detect additional ones, its result should still be interpreted with care, and rely on the co-operation of medical experts and statisticians. We believe it can be used effectively to perform risk detection on large sets of molecules while contextualizing these risks so as to ease further confirmation studies.

Acknowledgements

We thank the engineers who worked with us on this project: Muhammad Abdalla, Philip Deegan, Angel Francisco Orta, Kévin Vu Saintonge, Dian Sun. We thank Claude Gissot for his help in the realisation of this work. We also thank Joel Coste and Jérémie Rudant for their help in the early stages of this work, Marie Laurent and Olivier Pallanca for their insights regarding results interpretation.

Authors' contribution

This work was co-authored by Maryan Morel, Benjamin Bouyer, Agathe Guilloux, Moussa Laanani, Fanny Leroy, Dinh Phong Nguyen, Youcef Sebiat, Emmanuel Bacry and Stephane Gaiffas. MM, AG, ML, FL, DPN, YS, EB, SG contributed to the research plan. MM, ML, FL, DPN, YS contributed to the study design. MM drafted the manuscript, YS performed data extraction, MM and YS conducted the analyses. MM, BB, ML, FL, DPN and YS contributed to the interpretation of the results. All authors reviewed the manuscript and approved the final version.

Compliance with Ethical Standards

Funding This research benefited from the CNAMTS-Polytechnique research partnership, and from the Data Science Initiative of Ecole Polytechnique.

Conflict of Interest M. Morel, B. Bouyer, A. Guilloux, M. Laanani, F. Leroy, D. P. Nguyen, Y. Sebiat, E. Bacry and S. Gaiffas declare that they have no conflict of interest.

Ethical approval Use of SNDS observational data was approved by the CNIL

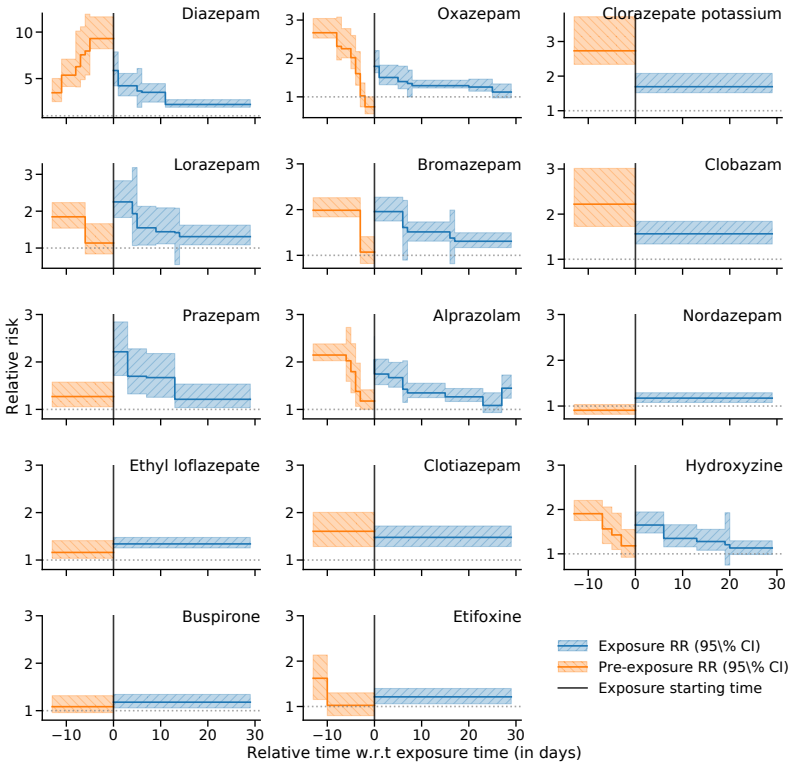


Figure III.5.1 – Fracture relative risk curves estimated before and after anxiolytics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

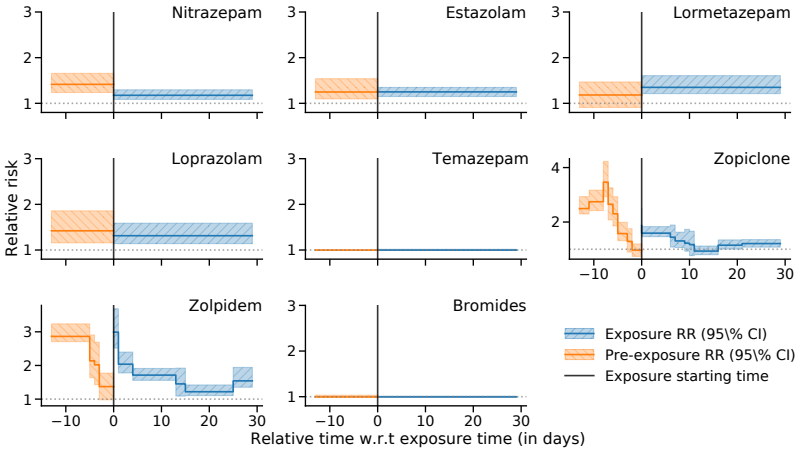


Figure III.5.2 – Fracture relative risk curves estimated before and after hypnotics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

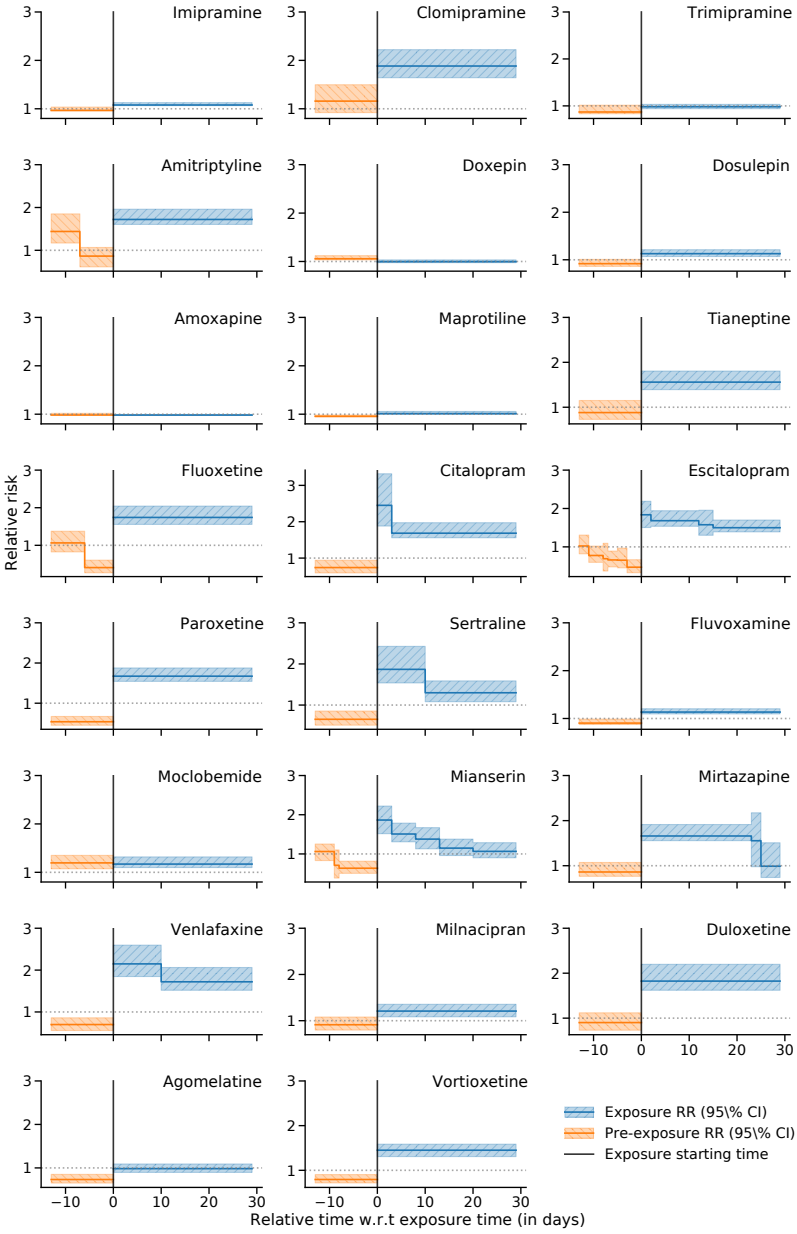


Figure III.5.3 – Fracture relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

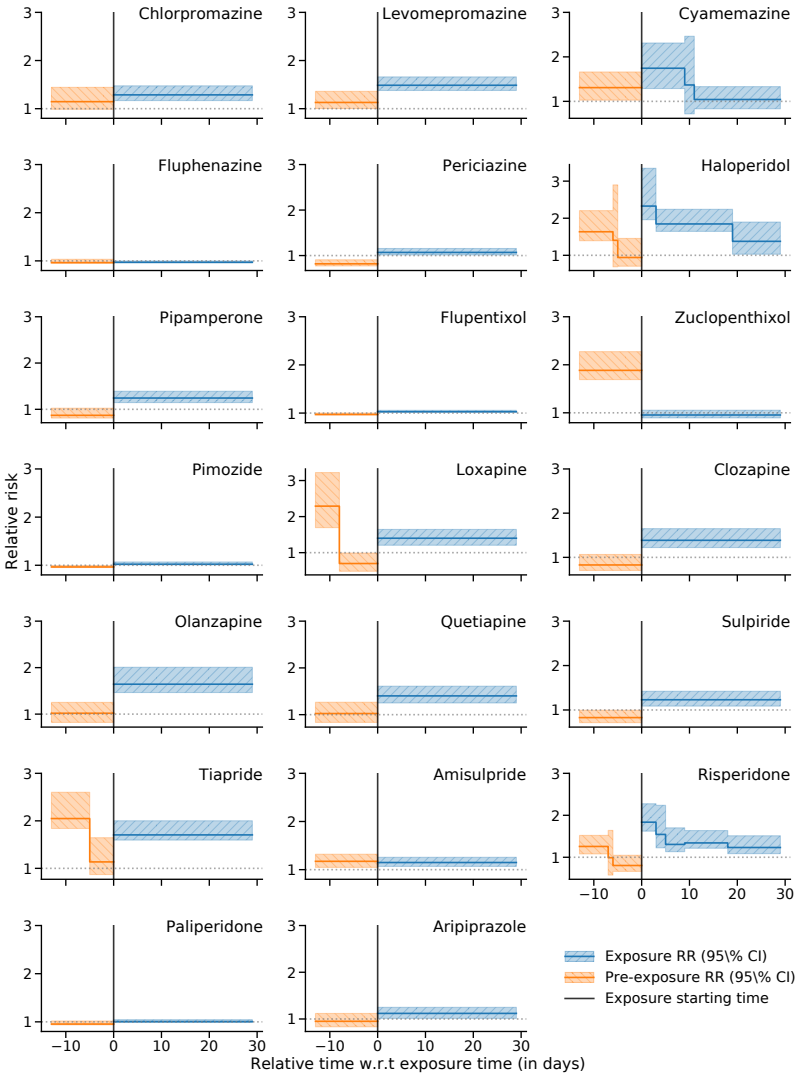


Figure III.5.4 – Fracture relative risk curves estimated before and after neuroleptics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

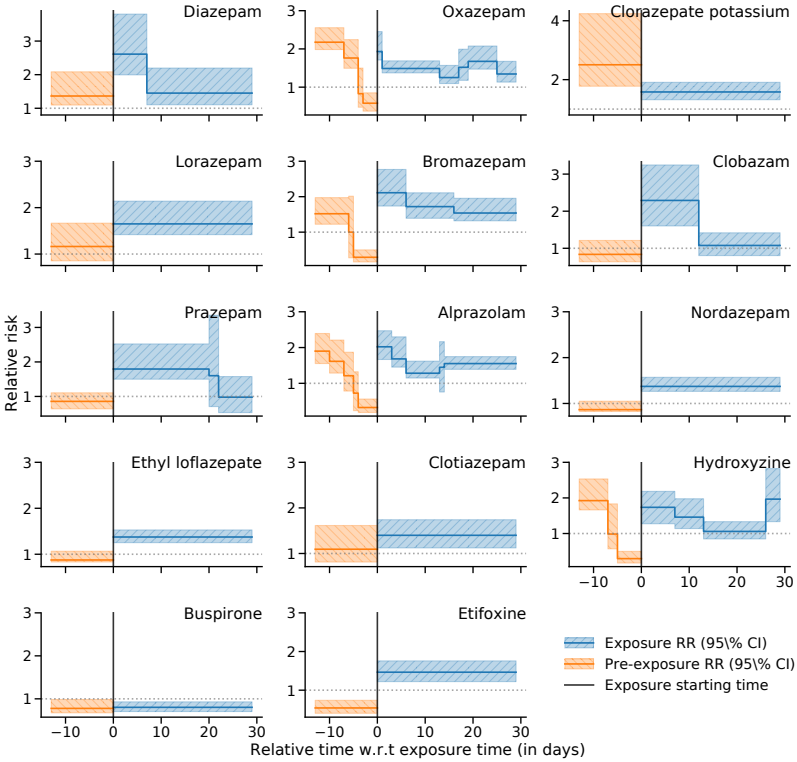


Figure III.5.5 – Hip fracture relative risk curves estimated before and after anxiolytics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

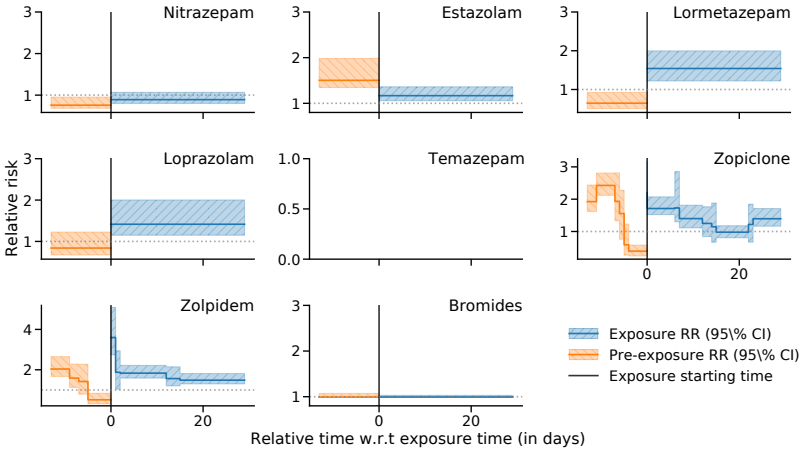


Figure III.5.6 – Hip fracture relative risk curves estimated before and after hypnotics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

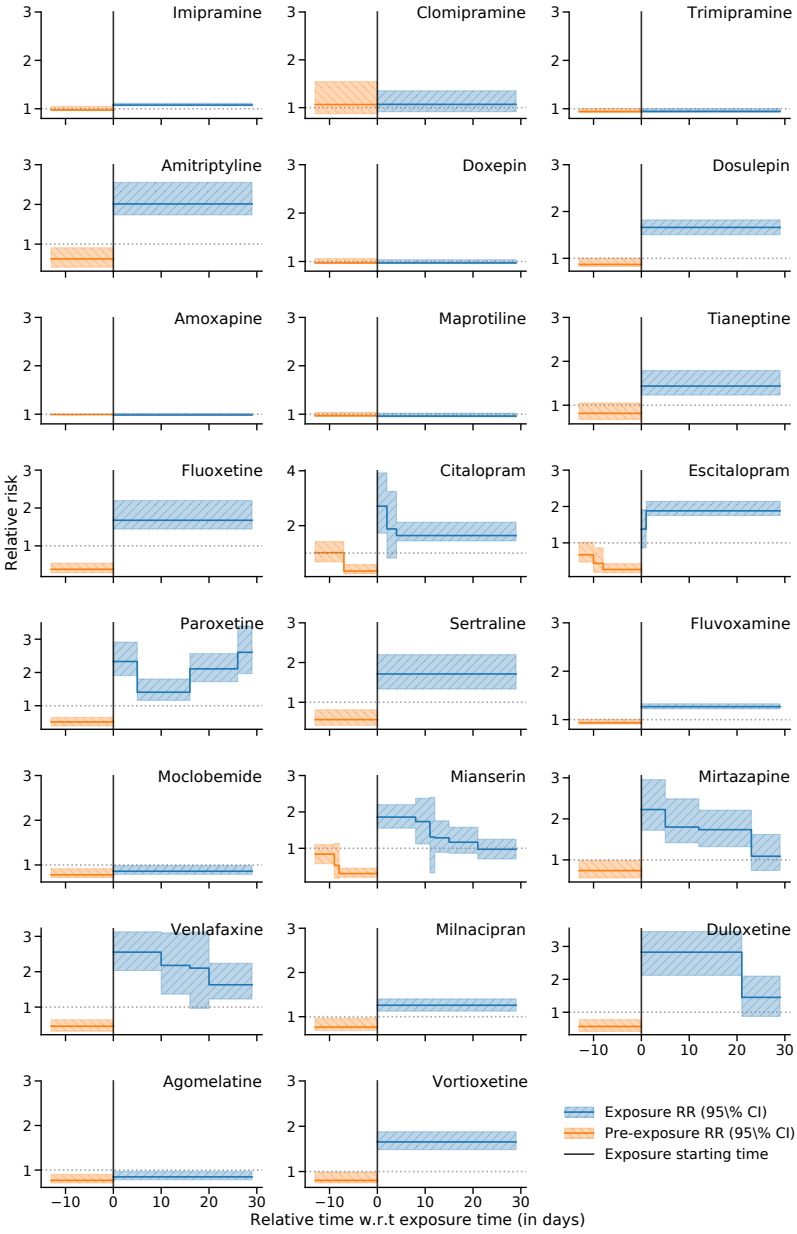


Figure III.5.7 – Hip fracture relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

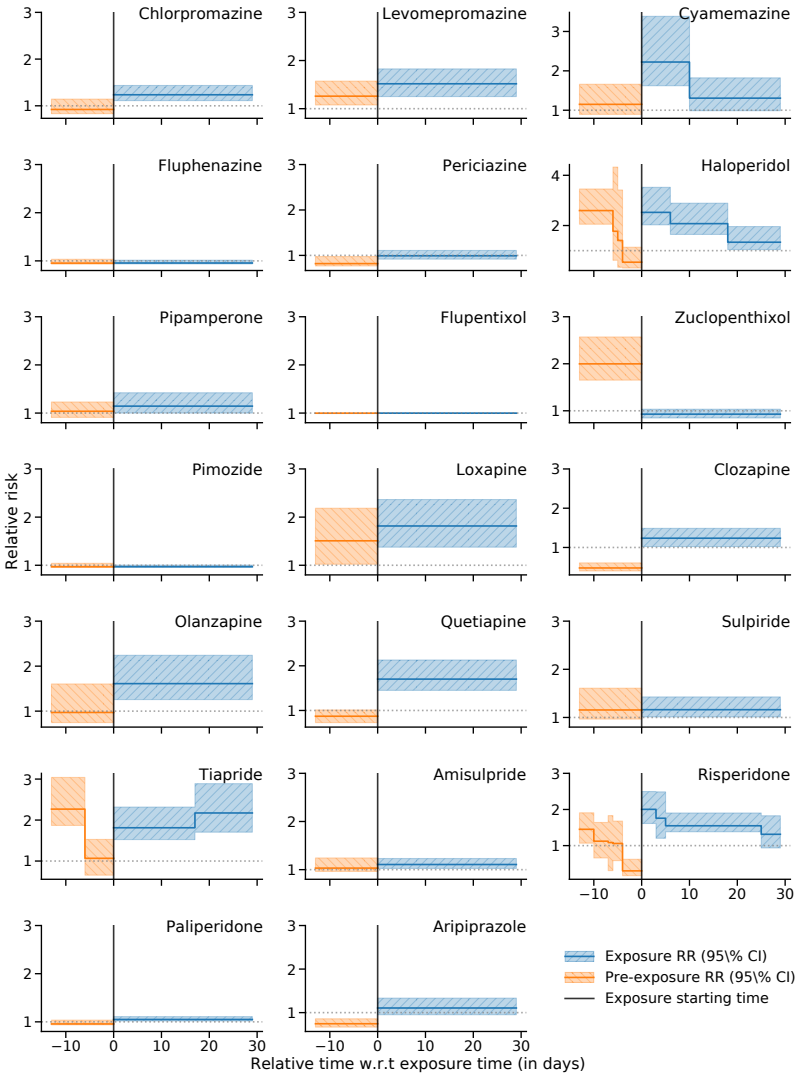


Figure III.5.8 – Hip fracture relative risk curves estimated before and after neuroleptics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

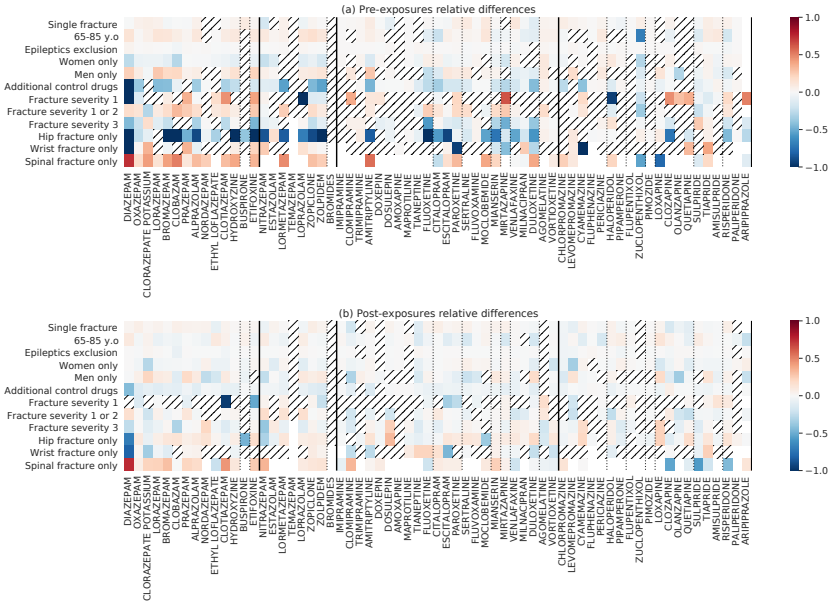


Figure III.5.9 – Summary of the significant changes in terms of mean relative difference over sensitivity analysis experiments. The top (resp. bottom) heatmap represents the relative errors of pre-exposure (resp. post-exposures) relative risks. To ease the reading, the mean relative difference between two relative risk curves are reported only when there is at least one coefficient of these curves being significantly different at 95% confidence with a power greater than 0.2. The darkest squares indicate the most variable results, and the hatched squares indicate relative risk curves for which power is less than 0.2. Errors reported in red (resp. blue) means that the estimated risk is higher (resp. lower) in the experiment than in the all-fracture, reference analysis.

Appendix

III.A Codes

Table III.A.1 – Anxiolytics: Anatomical Therapeutic Chemical (ATC) codes beginning with N05B*, N05CF*, N05CM11, N05CM16 and N05CX. Midazolam was excluded, at it is mostly used as pre-medication for minor surgery [Ver04].

Molecule	ATC class	Chemical class	ATC Code
DIAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05BA01
OXAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05BA04
CLORAZEPATE POTASSIQUE	Benzodiazepine derivatives	Benzodiazepines	N05BA05
LORAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05BA06
BROMAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05BA08
CLOBAZAM	Benzodiazepine derivatives	Benzodiazepines	N05BA09
PRAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05BA11
ALPRAZOLAM	Benzodiazepine derivatives	Benzodiazepines	N05BA12
NORDAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05BA16
ETHYLE LOFLAZEPATE	Benzodiazepine derivatives	Benzodiazepines	N05BA18
CLOTIAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05BA21
HYDROXYZINE	Diphenylmethane derivatives	Benzene and substituted derivatives	N05BB01
BUSPIRONE	Azaspirodecanedione derivatives	Diazinanes	N05BE01
ETIFOXINE	Other anxiolytics	Benzoxazines	N05BX03

Table III.A.2 – Hypnotics: Anatomical Therapeutic Chemical (ATC) codes beginning with N05CD*

Molecule	ATC class	Chemical class	ATC Code
NITRAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05CD02
ESTAZOLAM	Benzodiazepine derivatives	Benzodiazepines	N05CD04
LORMETAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05CD06
TEMAZEPAM	Benzodiazepine derivatives	Benzodiazepines	N05CD07
LOPRAZOLAM	Benzodiazepine derivatives	Benzodiazepines	N05CD11
ZOPICLONE	Benzodiazepine derivatives	Benzodiazepines	N05CF01
ZOLPIDEM	Benzodiazepine derivatives	Benzodiazepines	N05CF02
BROMURES	Other	Homogeneous halogens	N05CM11

Table III.A.3 – Antidepressants: Anatomical Therapeutic Chemical (ATC) codes beginning with N06A*, excepted Oxitriptan

Molecule	ATC class	Chemical class	ATC Code
IMIPRAMINE	TCA	Benzazepines	N06AA02
CLOMIPRAMINE	TCA	Benzazepines	N06AA04
TRIMIPRAMINE	TCA	Benzazepines	N06AA06
AMITRIPTYLINE	TCA	Dibenzocycloheptenes	N06AA09
DOXEPINE	TCA	Benzoxepines	N06AA12
DOSULEPINE	TCA	Benzothiepins	N06AA16
AMOXAPINE	TCA	Benzoxazepines	N06AA17
MAPROTILINE	TCA	Anthracenes	N06AA21
TIANEPTINE	TCA	Fatty Acyls	N06AX14
FLUOXETINE	SSRI	Benzene and substituted derivatives	N06AB03
CITALOPRAM	SSRI	Benzene and substituted derivatives	N06AB04
ESCITALOPRAM	SSRI	Benzene and substituted derivatives	N06AB10
PAROXETINE	SSRI	Piperidines	N06AB05
SERTRALINE	SSRI	Tetralins	N06AB06
FLUVOXAMINE	SSRI	Aralkylketone derivative	N06AB08
MOCLOBEMIDE	MAOI RIMA	Benzene and substituted derivatives	N06AG02
MIANSERINE	Tetracyclic	Benzazepines	N06AX03
MIRTAZAPINE	Tetracyclic	Benzazepines	N06AX11
VENLAFAXINE	SNRI	Phenol ethers	N06AX16
MILNACIPRAN	SNRI	Unclassed	N06AX17
DULOXETINE	SNRI	Naphtalenes	N06AX21
AGOMELATINE	Other	Carboxylic acids and derivatives	N06AX22
VORTIOXETINE	Other	Piperidines	N06AX26

Table III.A.4 – Neuroleptics: Anatomical Therapeutic Chemical (ATC) codes beginning with N05A* excepted Veralipride, Lithium and Chlorproethazin, as they are mostly used as a mood stabiliser to treat bipolar disorders or schizo-affective disorders rather than depression.

Molecule	ATC class	Chemical class	ATC Code
CHLORPROMAZINE	Phenothiazines with aliphatic side-chain	Benzothiazines	N05AA01
LEVOMEPROMAZINE	Phenothiazines with aliphatic side-chain	Benzothiazines	N05AA02
CYAMEMAZINE	Phenothiazines with aliphatic side-chain	Benzothiazines	N05AA06
FLUPHENAZINE	Phenothiazines with aliphatic side-chain	Benzothiazines	N05AB02
PERICIAZINE	Phenothiazines with aliphatic side-chain	Benzothiazines	N05AC01
HALOPERIDOL	Butyrophenone derivatives	Organooxygen compounds	N05AD01
PIPAMPERONE	Butyrophenone derivatives	Organooxygen compounds	N05AD05
FLUPENTIXOL	Thioxanthene derivatives	Benzothiopyrans	N05AF01
ZUCLOPENTHIXOL	Thioxanthene derivatives	Benzothiopyrans	N05AF05
PIMOZIDE	Diphenylbutylpiperidine derivatives	Benzene and substituted derivatives	N05AG02
LOXAPINE	Diazepines, oxazepines, thiazepines and oxepines	Benzoxazepines	N05AH01
CLOZAPINE	Diazepines, oxazepines, thiazepines and oxepines	Benzodiazepines	N05AH02
OLANZAPINE	Diazepines, oxazepines, thiazepines and oxepines	Benzodiazepines	N05AH03
QUETIAPINE	Diazepines, oxazepines, thiazepines and oxepines	Benzothiazepines	N05AH04
SULPIRIDE	Benzamides	Benzene and substituted derivatives	N05AL01
TIAPRIDE	Benzamides	Benzene and substituted derivatives	N05AL03
AMISULPRIDE	Benzamides	Benzene and substituted derivatives	N05AL05
RISPERIDONE	Other	Pyridopyrimidines	N05AX08
PALIPERIDONE	Other	Pyridopyrimidines	N05AX13
ARIPIPRAZOLE	Other	Diazinanes	N05AX12

III.B Sensitivity analysis

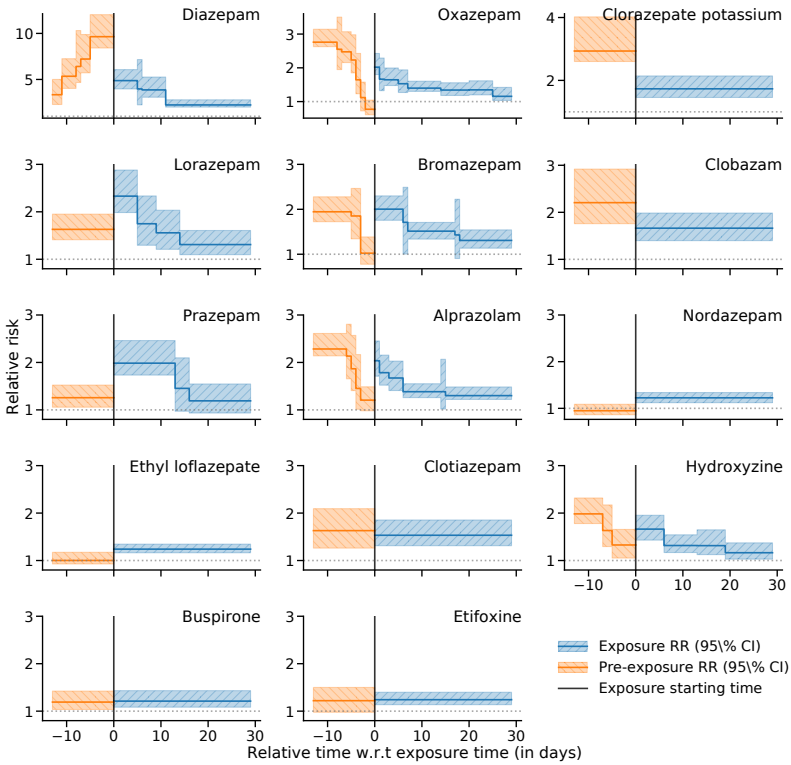


Figure III.B.1 – Fracture relative risk curves estimated before and after anxiolytics exposure on patients having experienced only one fracture during the observation period. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

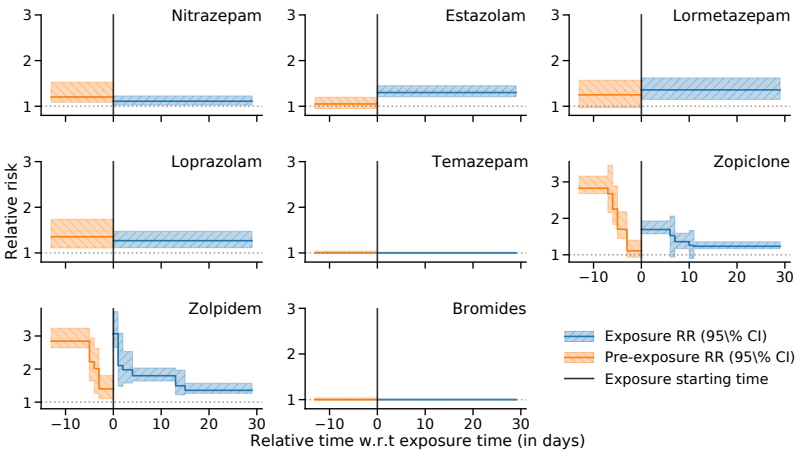


Figure III.B.2 – Fracture relative risk curves estimated before and after hypnotics exposure on patients having experienced only one fracture during the observation period. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

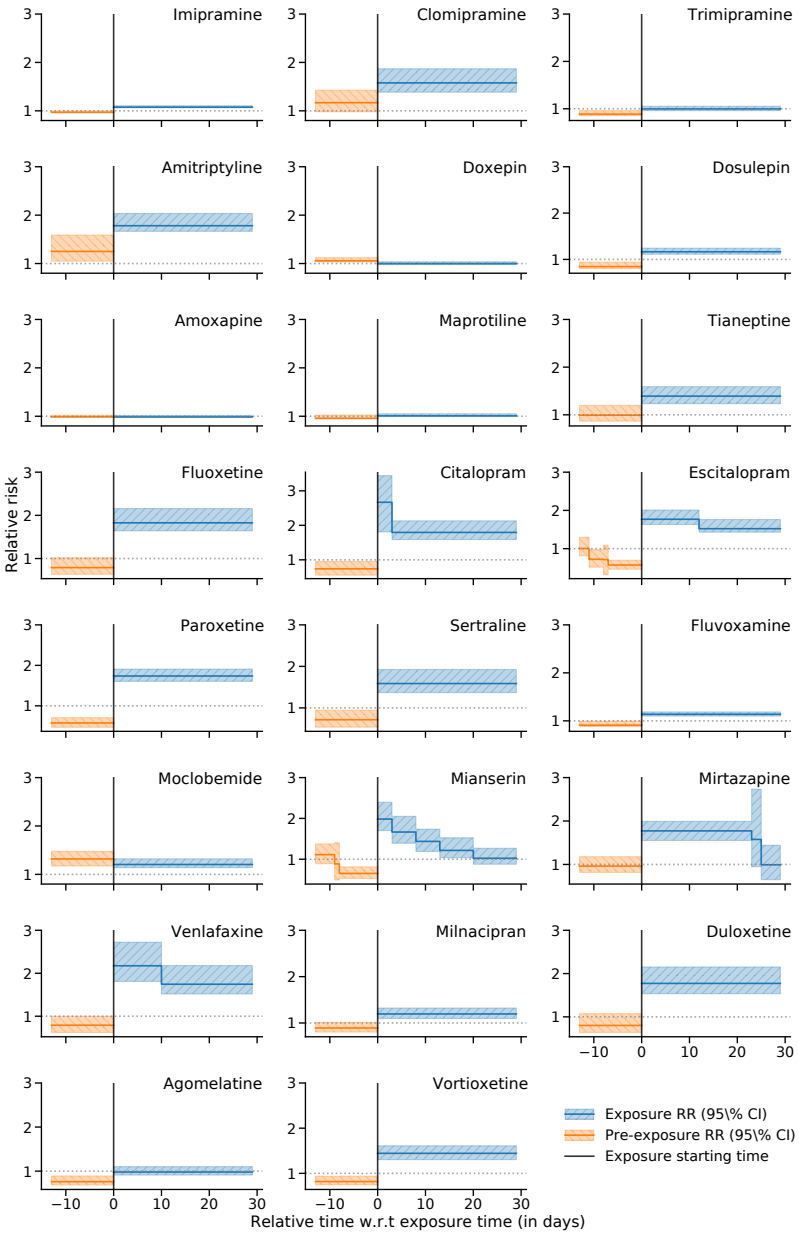


Figure III.B.3 – Fracture relative risk curves estimated before and after antidepressant exposure on patients having experienced only one fracture during the observation period. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

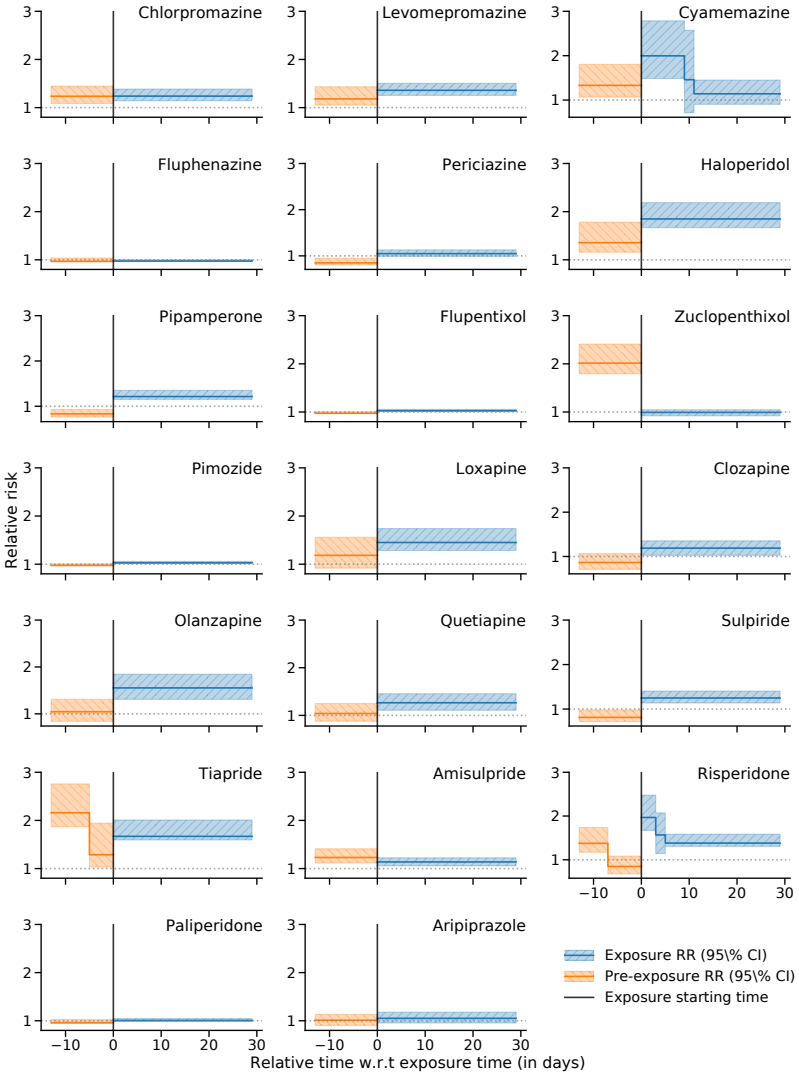


Figure III.B.4 – Fracture relative risk curves estimated before and after neuroleptics exposure on patients having experienced only one fracture during the observation period. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

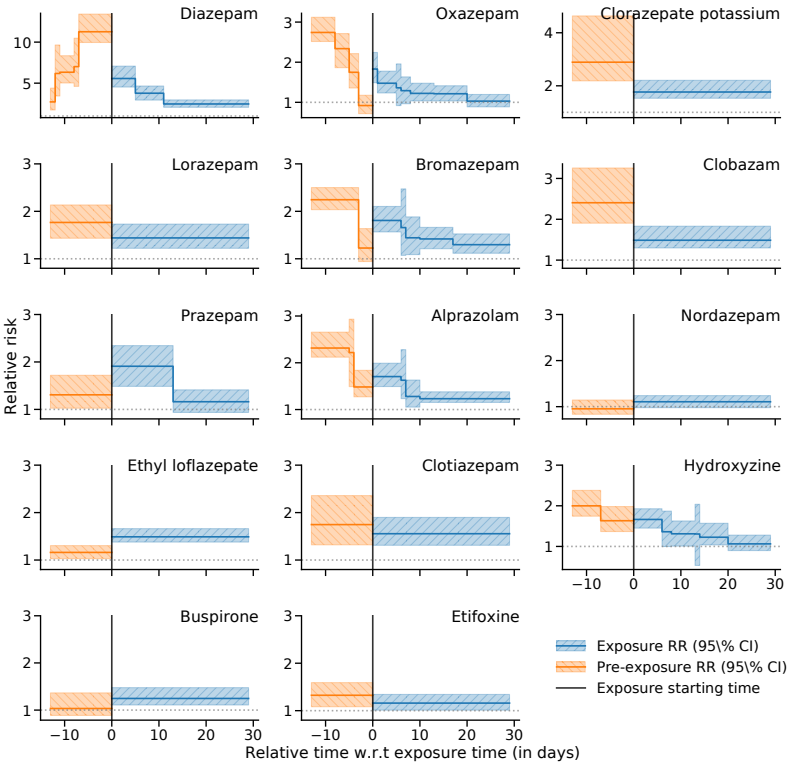


Figure III.B.5 – Fracture relative risk curves estimated before and after anxiolytics exposure on $65-85$ y.o. patients. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

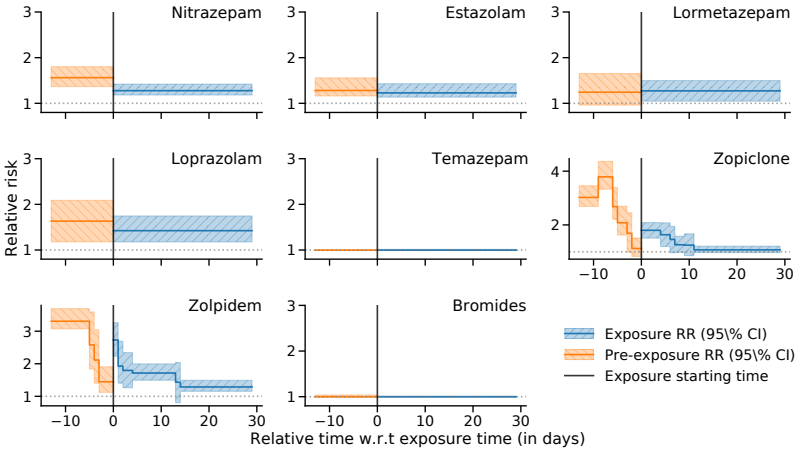


Figure III.B.6 – Fracture relative risk curves estimated before and after hypnotics exposure on $65-85$ y.o. patients. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

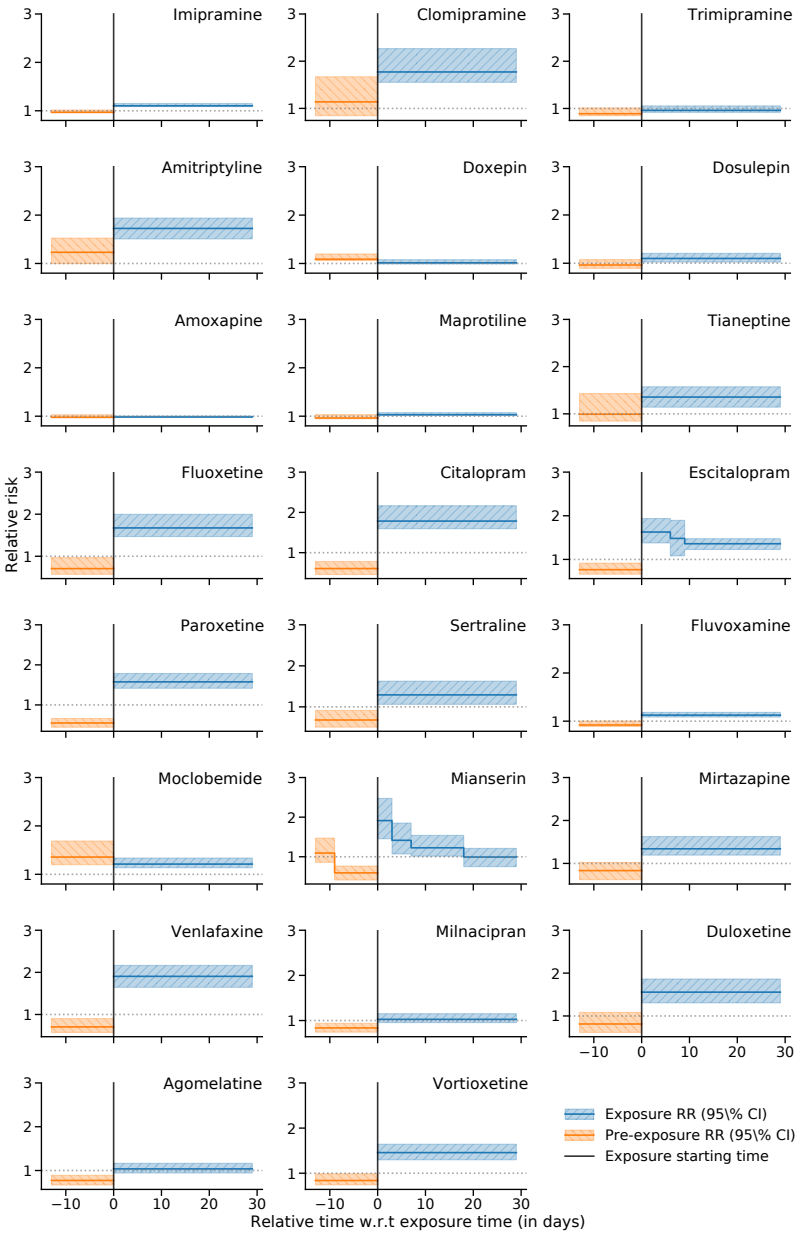


Figure III.B.7 – Fracture relative risk curves estimated before and after antidepressant exposure on $65-85$ y.o. patients. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respórange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

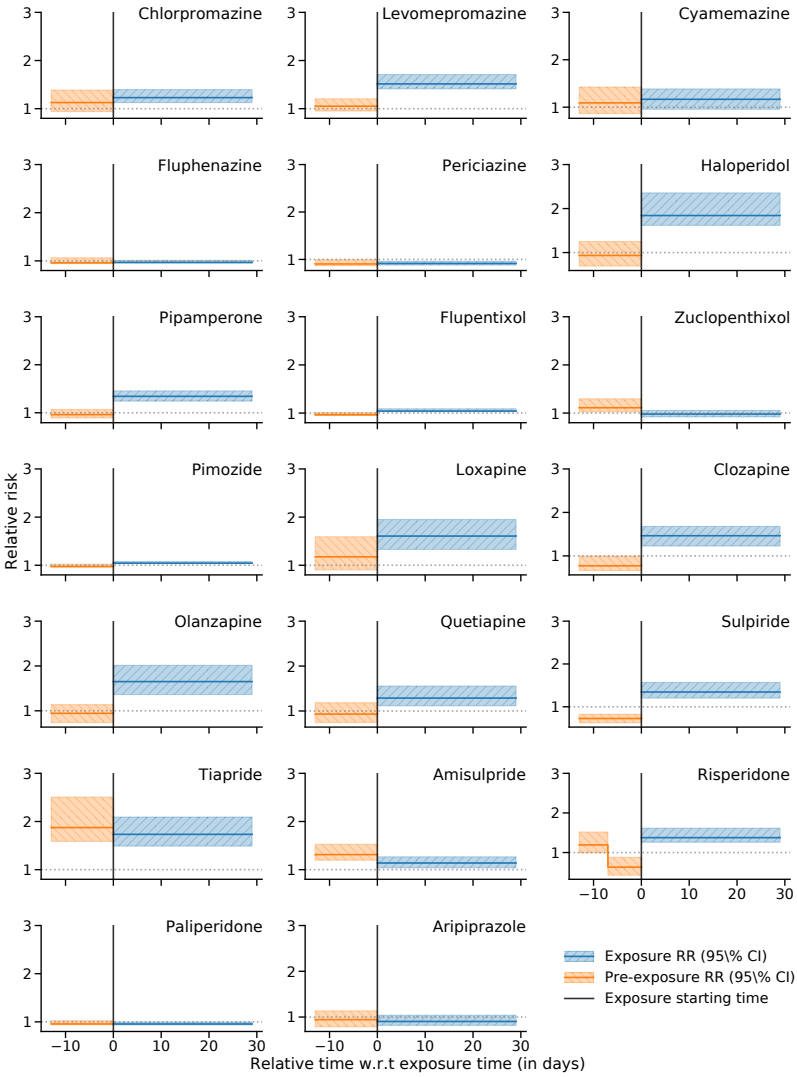


Figure III.B.8 – Fracture relative risk curves estimated before and after neuroleptics exposure on $65-85$ y.o. patients. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

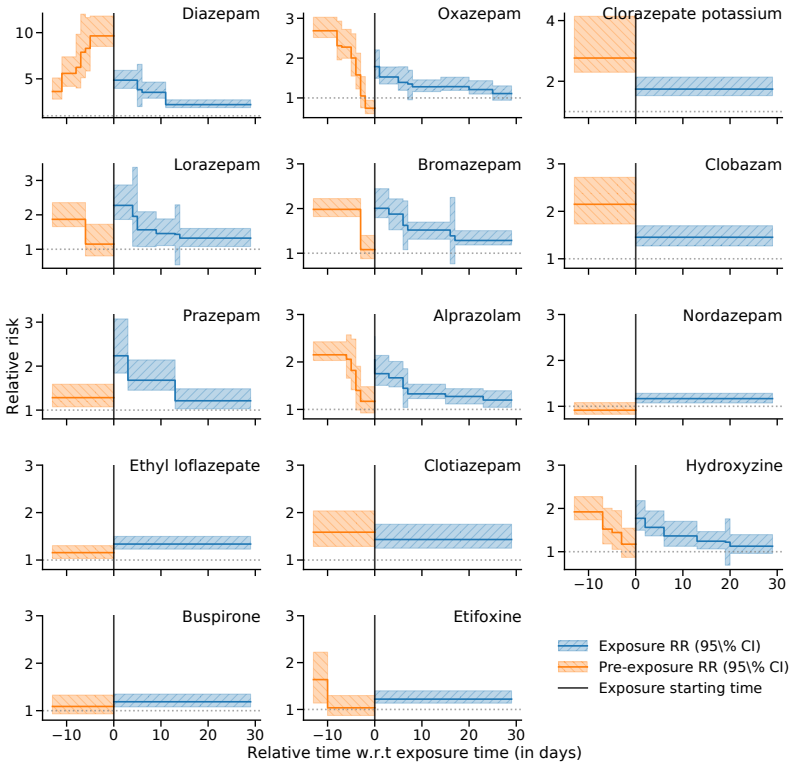


Figure III.B.9 – Fracture relative risk curves estimated before and after anxiolytics exposure on non-epileptic patients. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

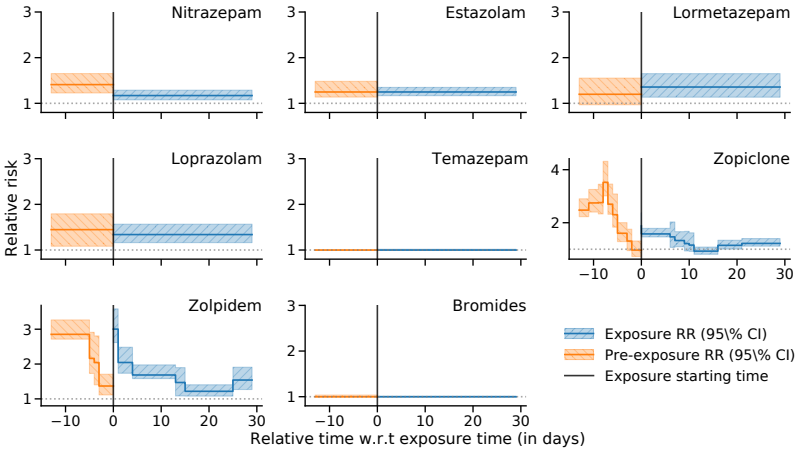


Figure III.B.10 – Fracture relative risk curves estimated before and after hypnotics exposure on non-epileptic patients. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

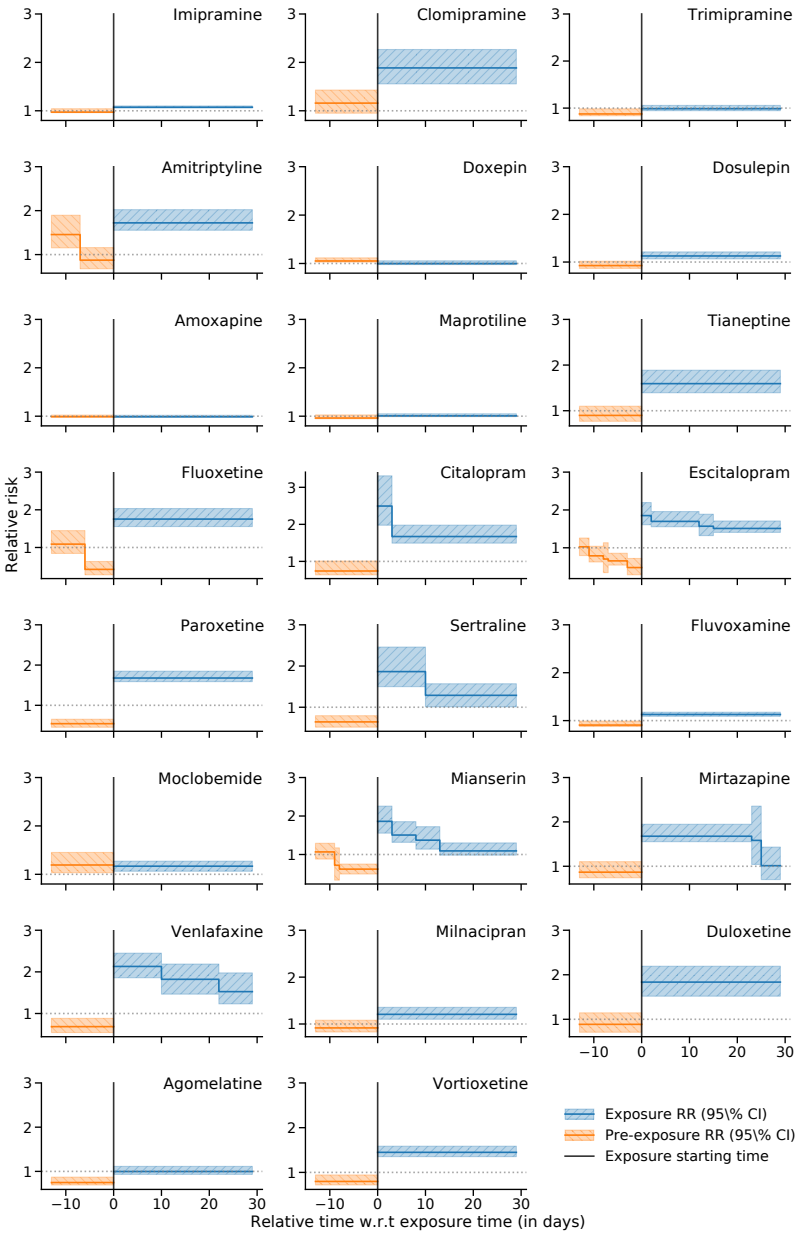


Figure III.B.11 – Fracture relative risk curves estimated before and after antidepressant exposure on non-epileptic patients. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

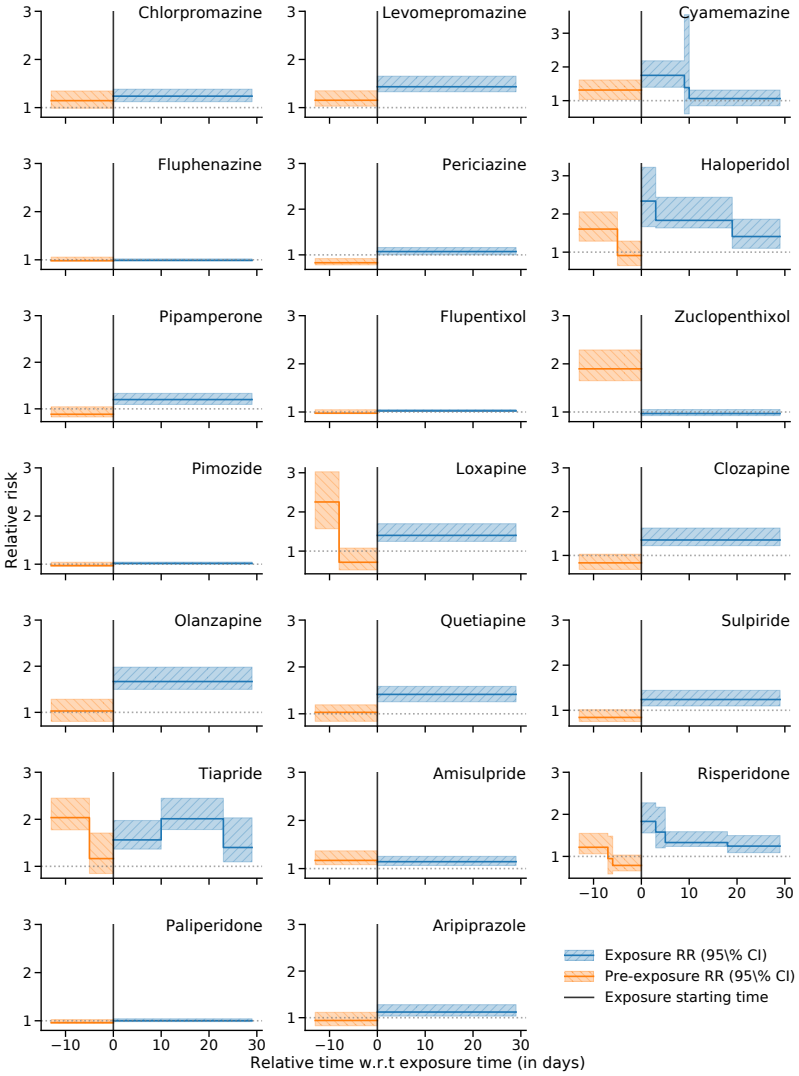


Figure III.B.12 – Fracture relative risk curves estimated before and after neuroleptics exposure after epileptic patients exclusion. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

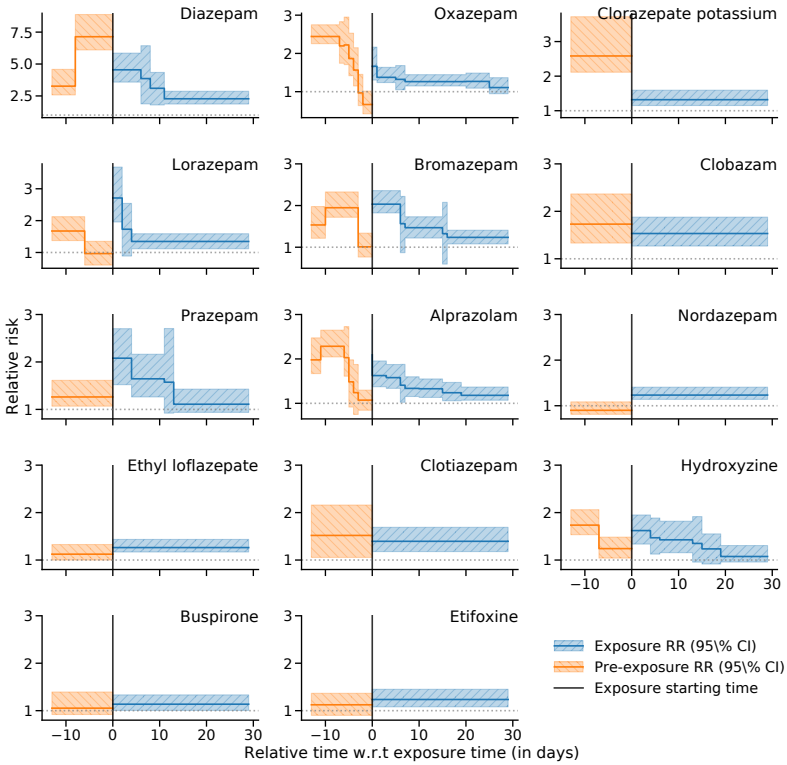


Figure III.B.13 – Fracture relative risk curves estimated before and after anxiolytics exposure on women only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp̀pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

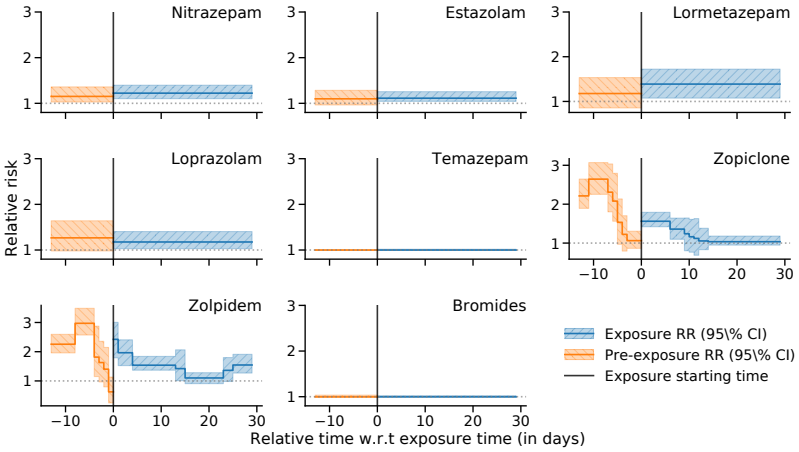


Figure III.B.14 – Fracture relative risk curves estimated before and after hypnotics exposure on women only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resppre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

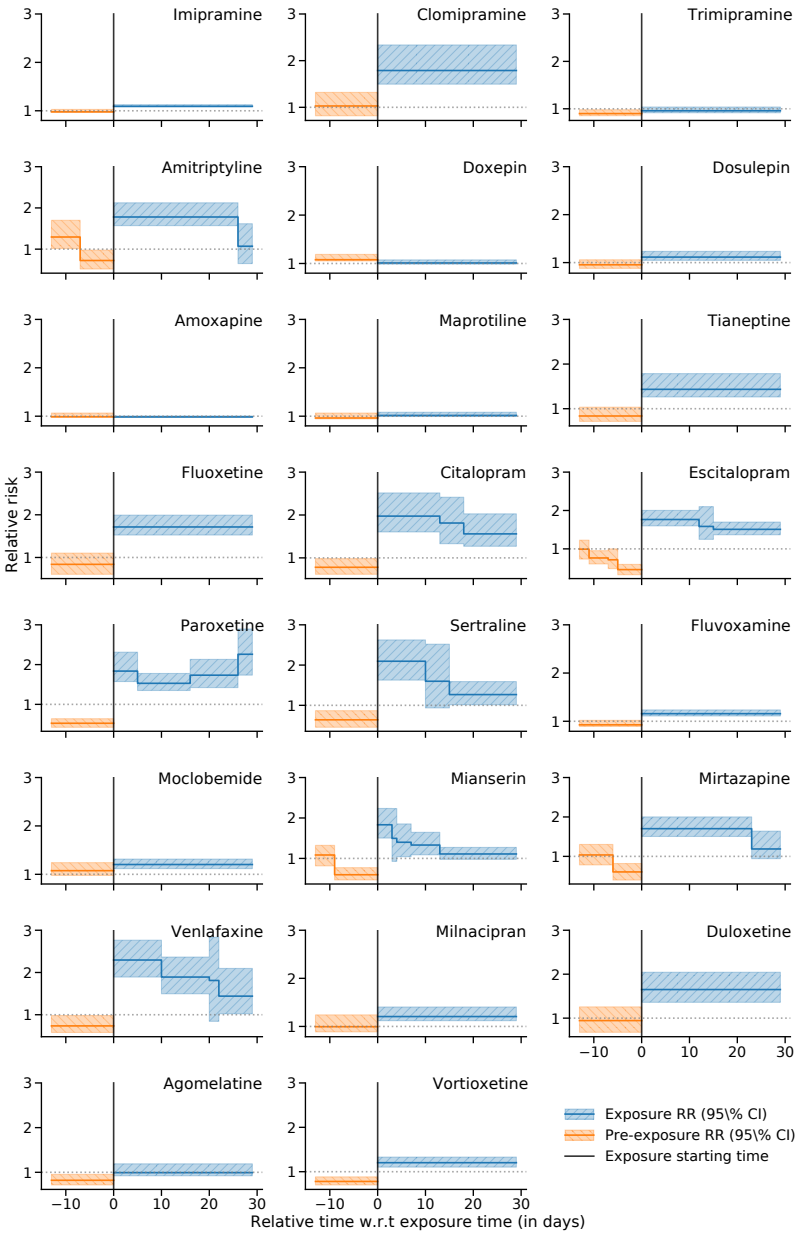


Figure III.B.15 – Fracture relative risk curves estimated before and after antidepressant exposure on women only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

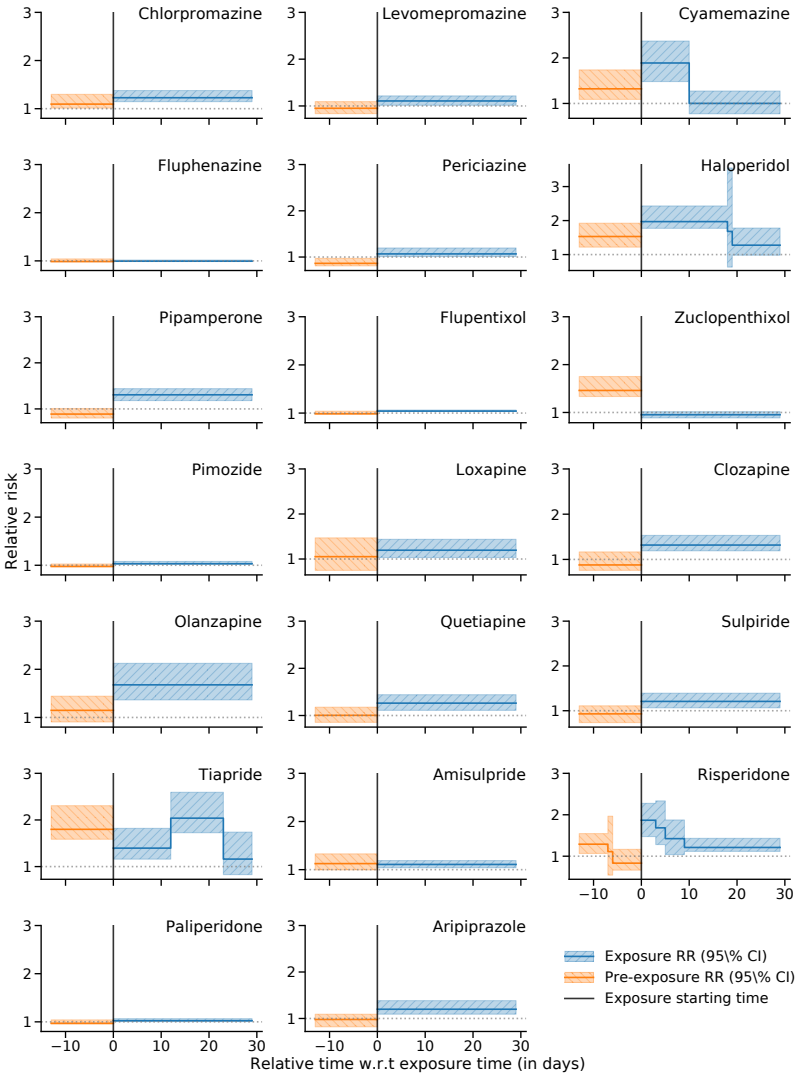


Figure III.B.16 – Fracture relative risk curves estimated before and after neuroleptics exposure on women only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp̀pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

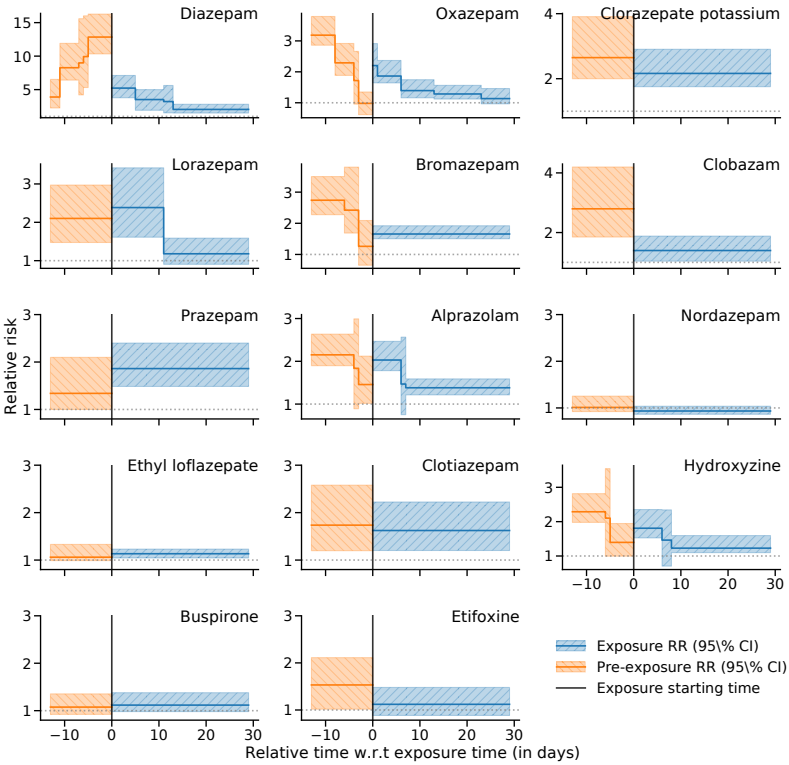


Figure III.B.17 – Fracture relative risk curves estimated before and after anxiolytics exposure on men only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp̀pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

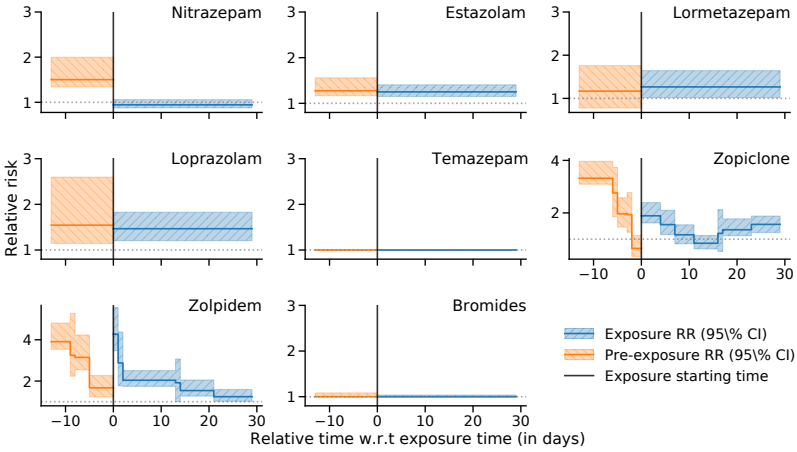


Figure III.B.18 – Fracture relative risk curves estimated before and after hypnotics exposure on men only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

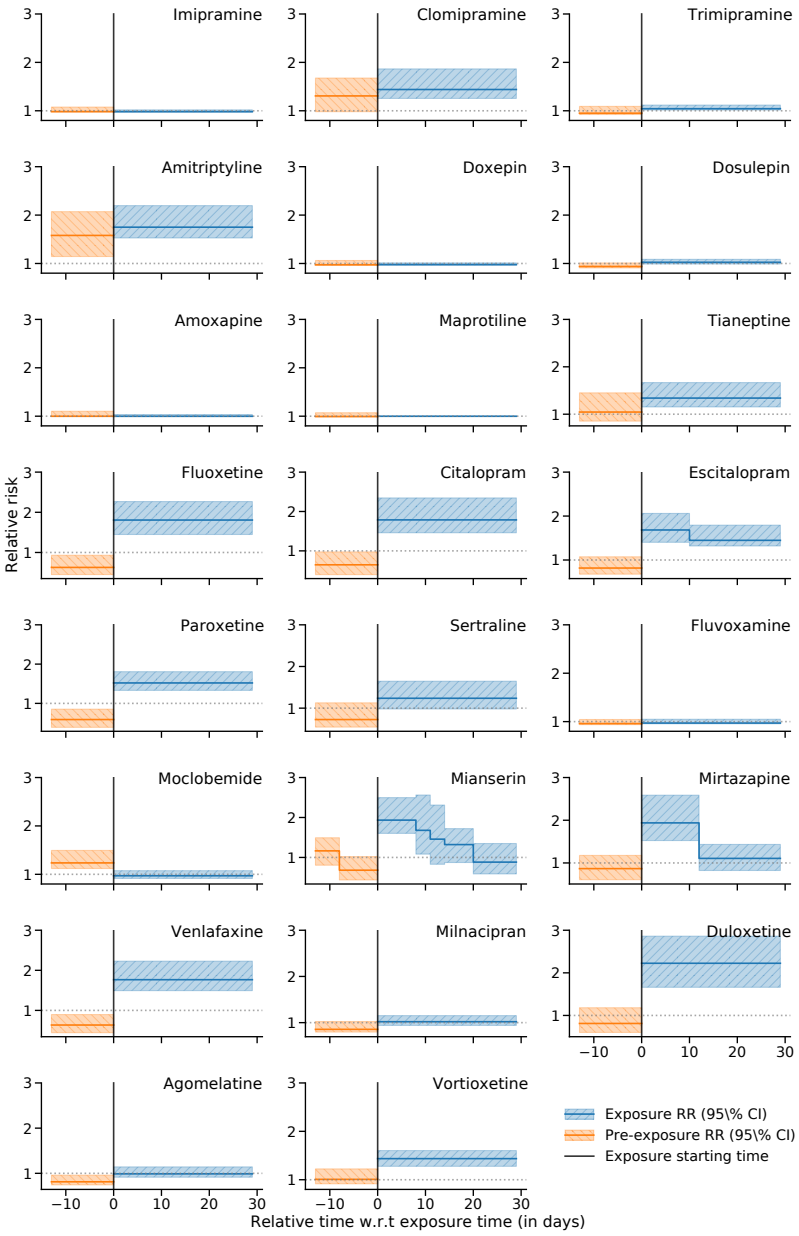


Figure III.B.19 – Fracture relative risk curves estimated before and after antidepressants exposure on men only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resppre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

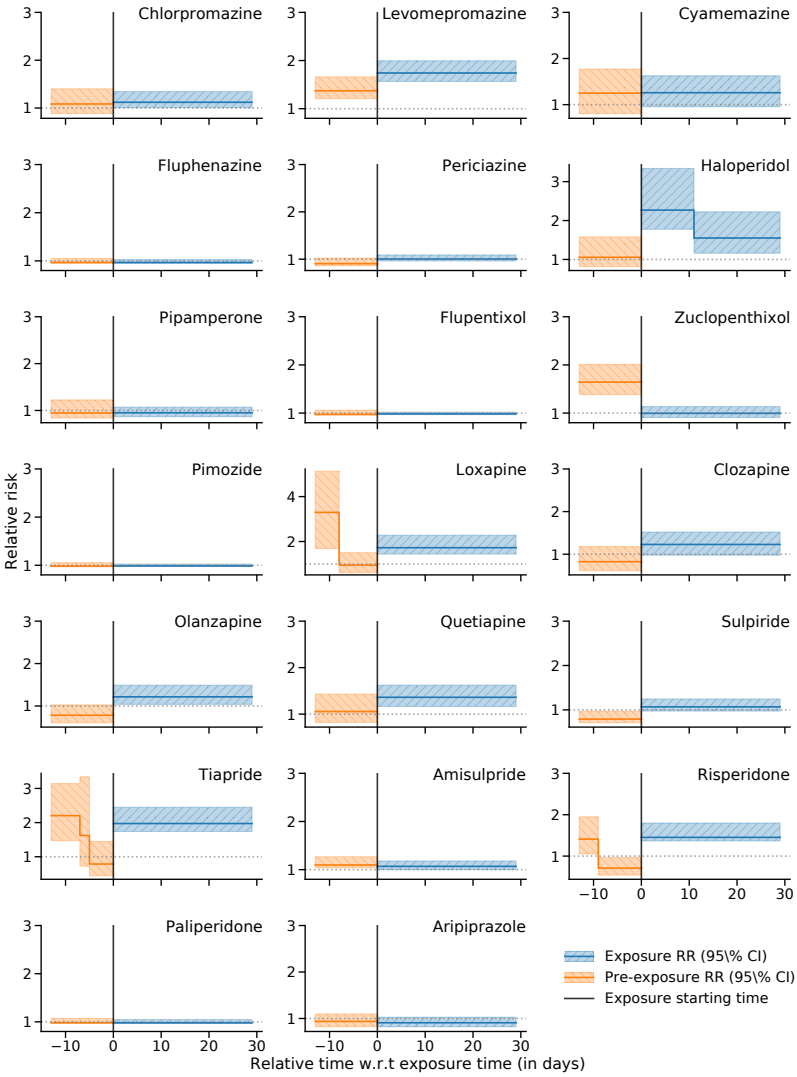


Figure III.B.20 – Fracture relative risk curves estimated before and after neuroleptics exposure on men only. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp̀pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

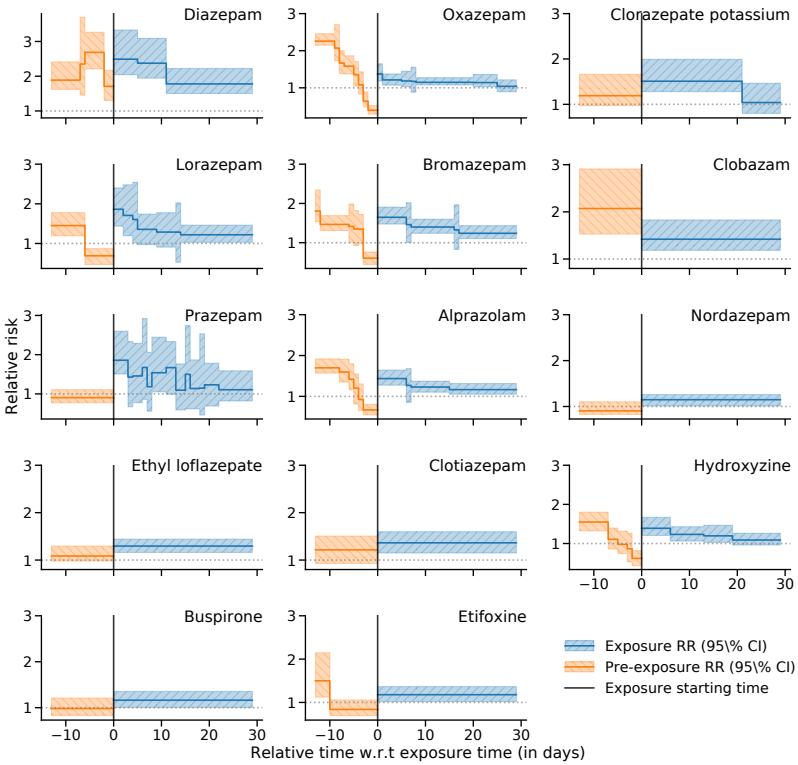


Figure III.B.21 – Fracture relative risk curves estimated before and after anxiolytics exposure when adding additional drugs as control variables. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

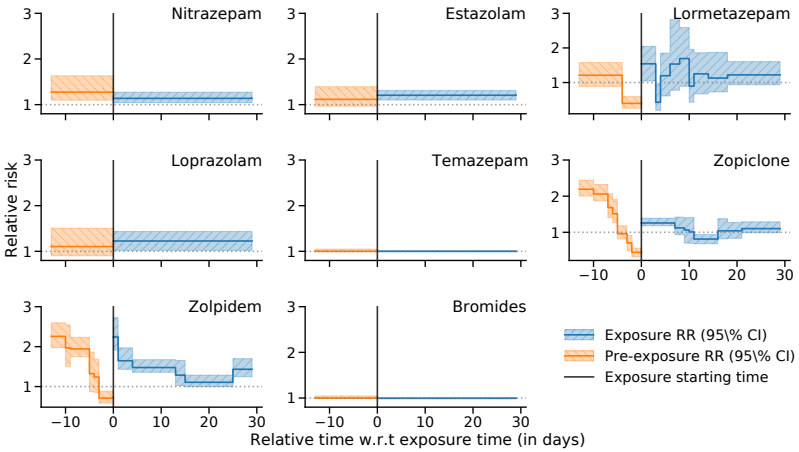


Figure III.B.22 – Fracture relative risk curves estimated before and after hypnotics exposure when adding additional drugs as control variables. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

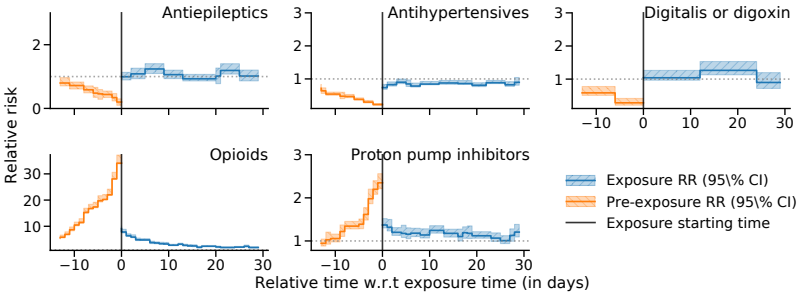


Figure III.B.23 – Fracture relative risk curves estimated before and after exposure to control drugs when adding additional drugs as control variables. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

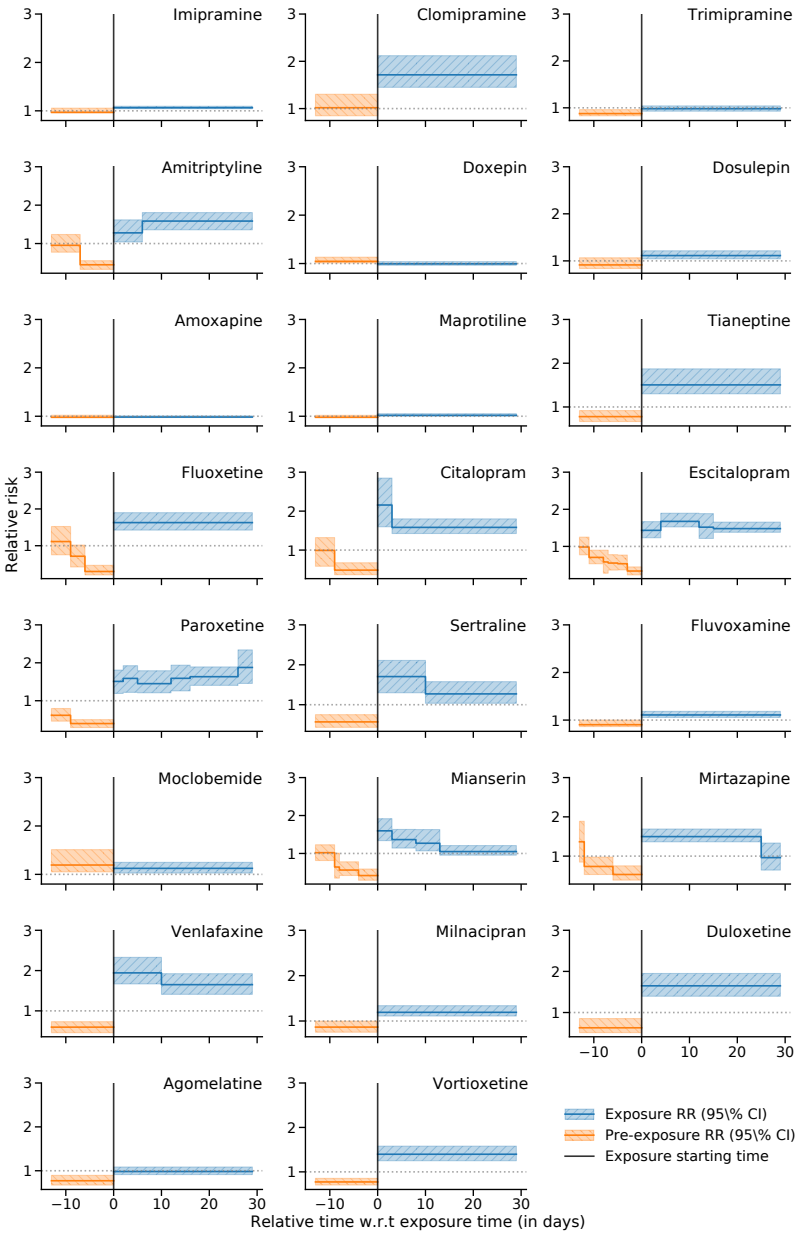


Figure III.B.24 – Fracture relative risk curves estimated before and after antidepressant exposure when adding additional drugs as control variables. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

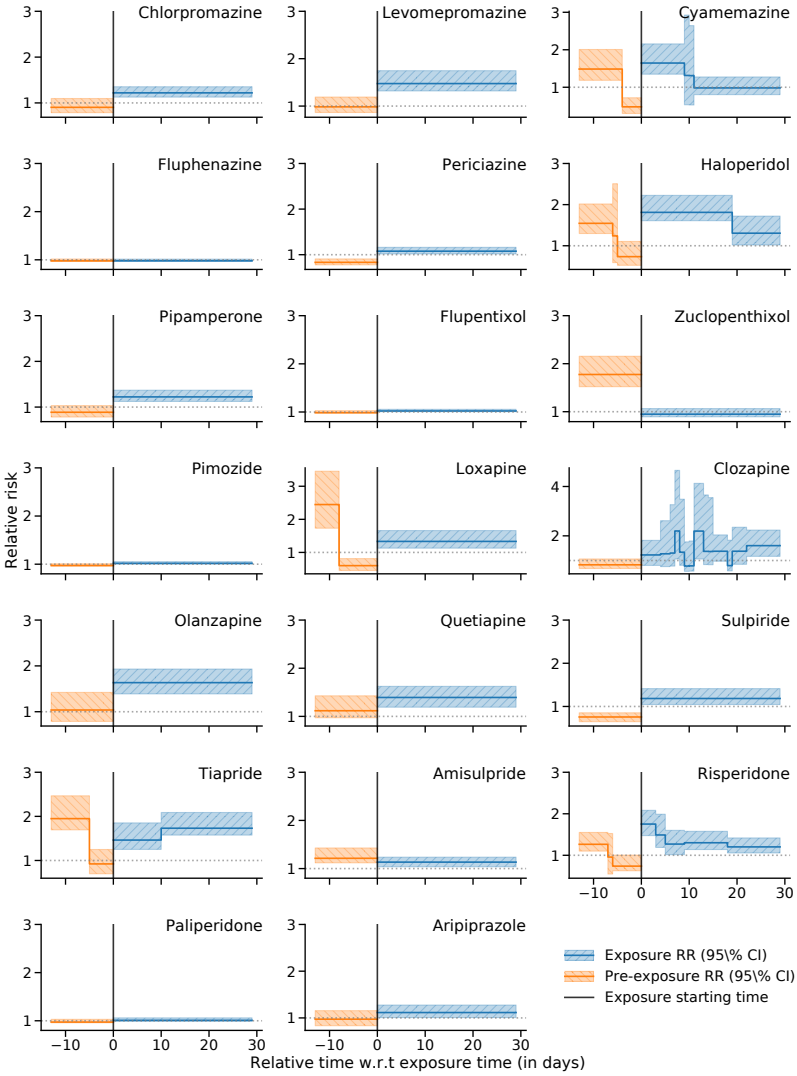


Figure III.B.25 – Fracture relative risk curves estimated before and after neuroleptics exposure when adding additional drugs as control variables. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

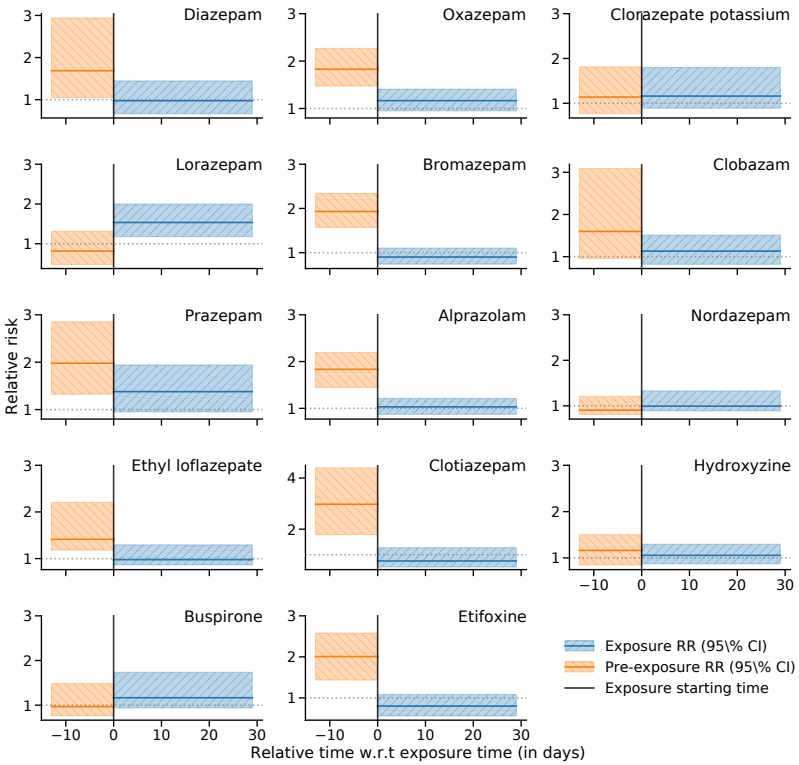


Figure III.B.26 – Non-hospitalised fracture relative risk curves estimated before and after anxiolytics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

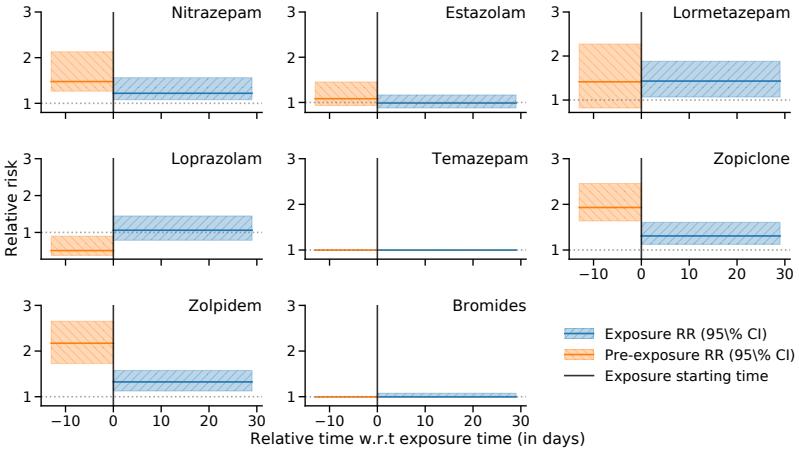


Figure III.B.27 – Non-hospitalised fracture relative risk curves estimated before and after hypnotics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resppre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

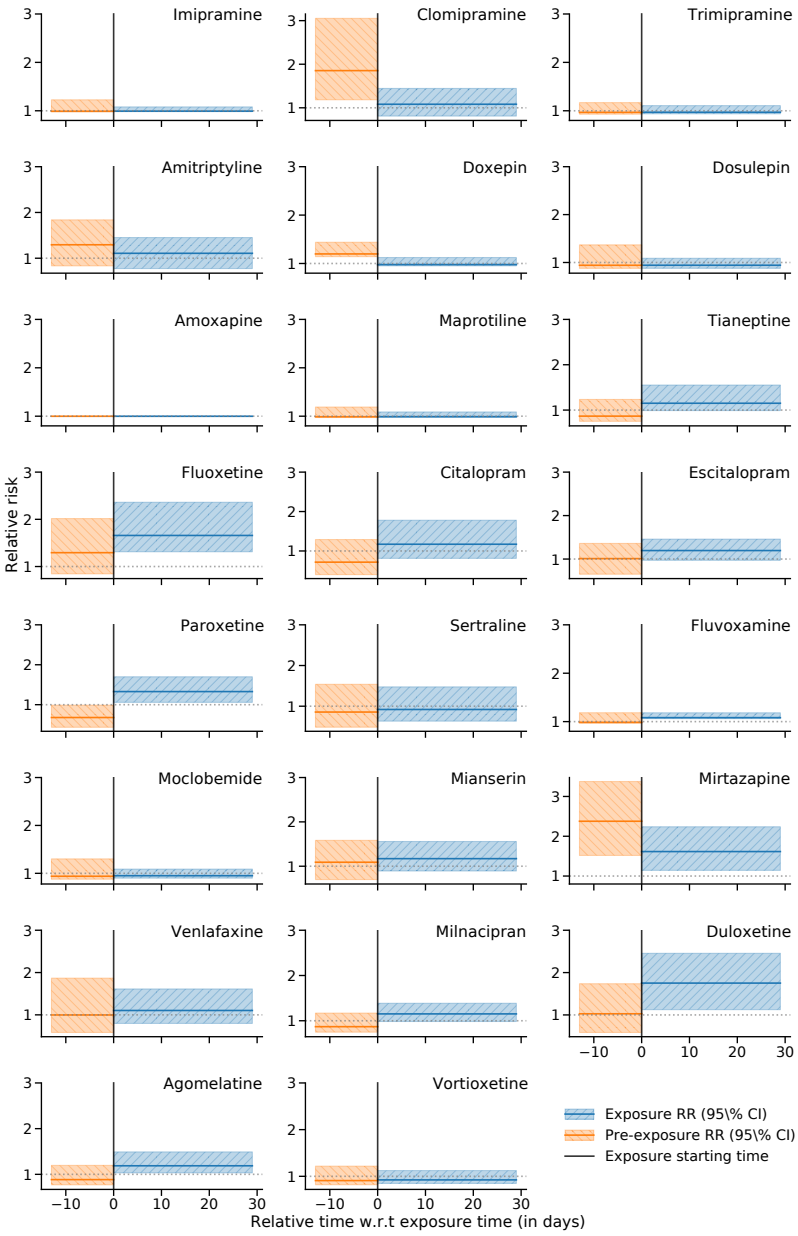


Figure III.B.28 – Non-hospitalised fracture relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

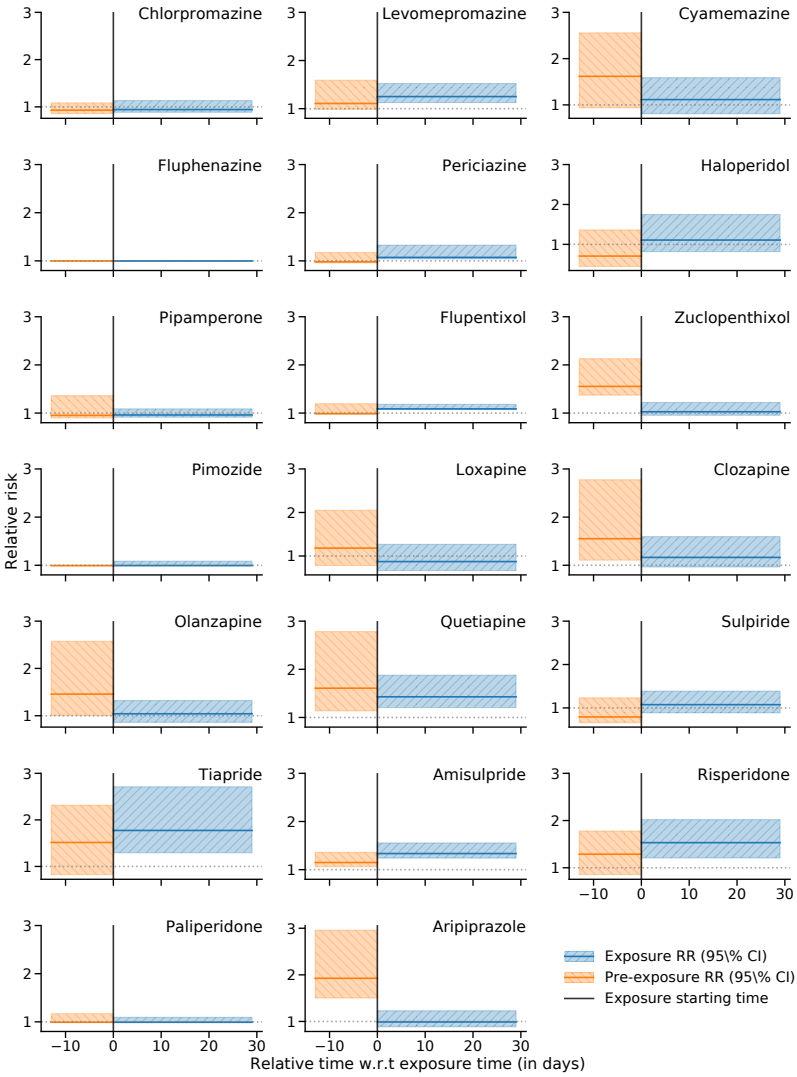


Figure III.B.29 – Non-hospitalised fracture relative risk curves estimated before and after neuroleptics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respórange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

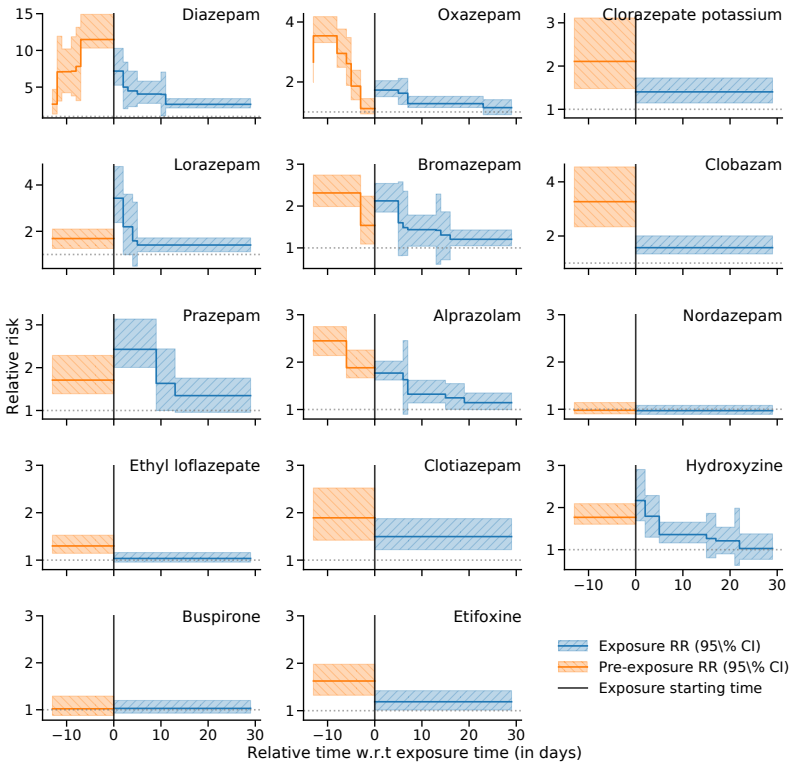


Figure III.B.30 – Fracture without surgery relative risk curves estimated before and after anxiolytics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

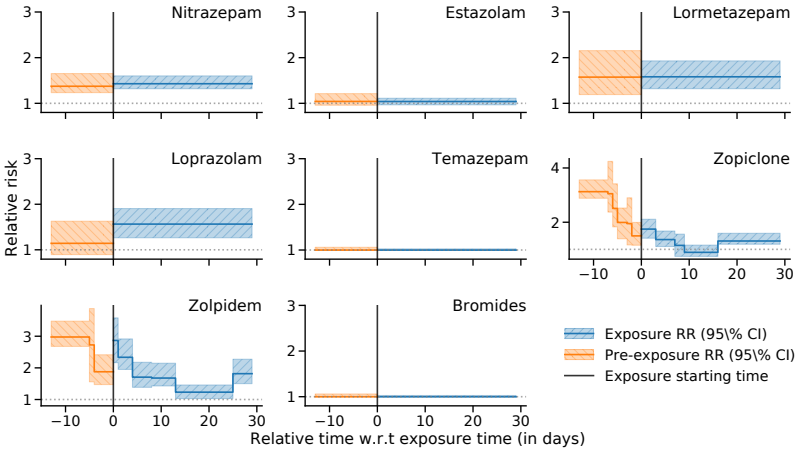


Figure III.B.31 – Fracture without surgery relative risk curves estimated before and after hypnotics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resppre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

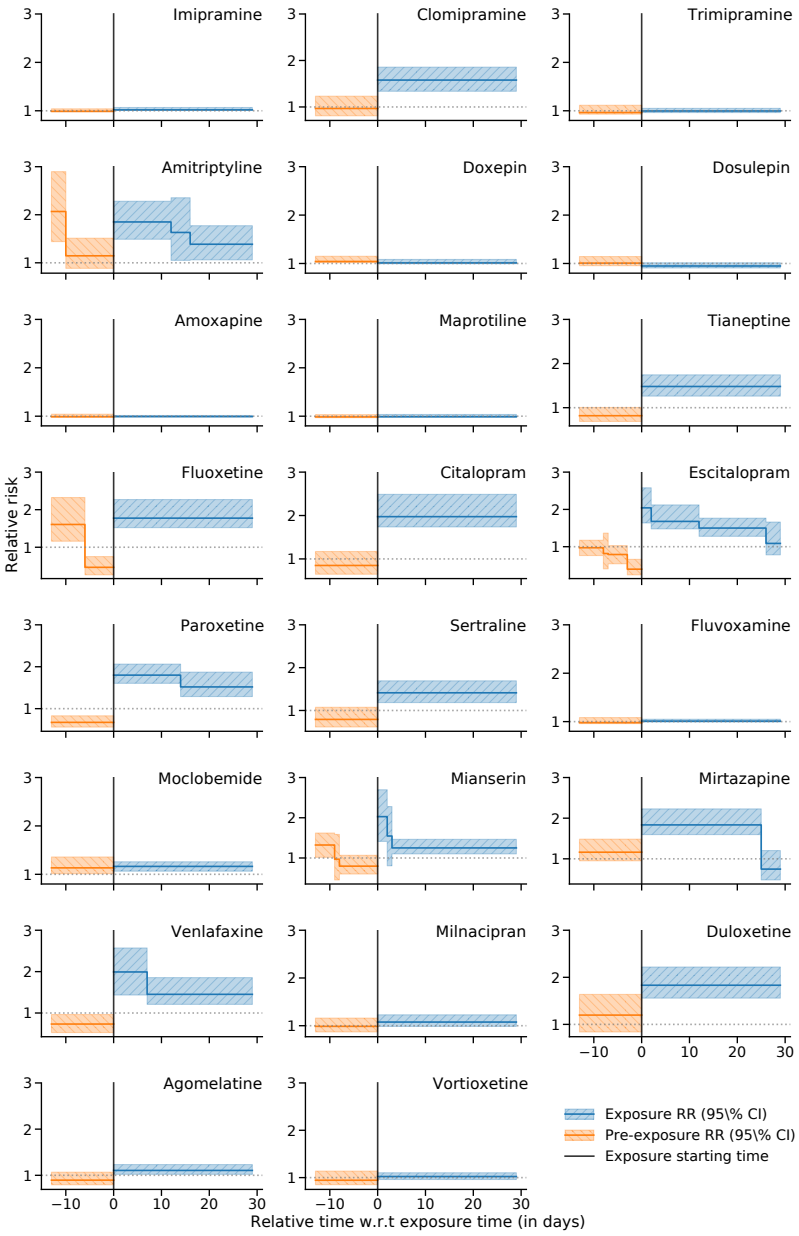


Figure III.B.32 – Fracture without surgery relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respȯrange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

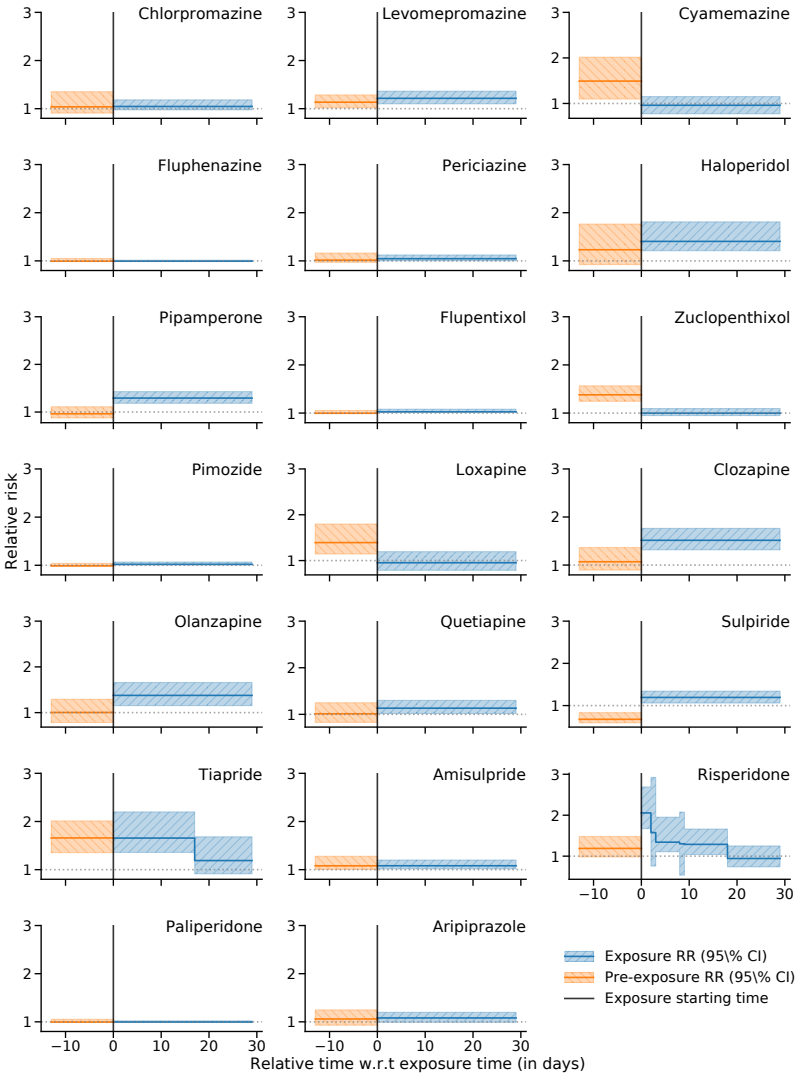


Figure III.B.33 – Fracture without surgery relative risk curves estimated before and after neuroleptics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respórange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

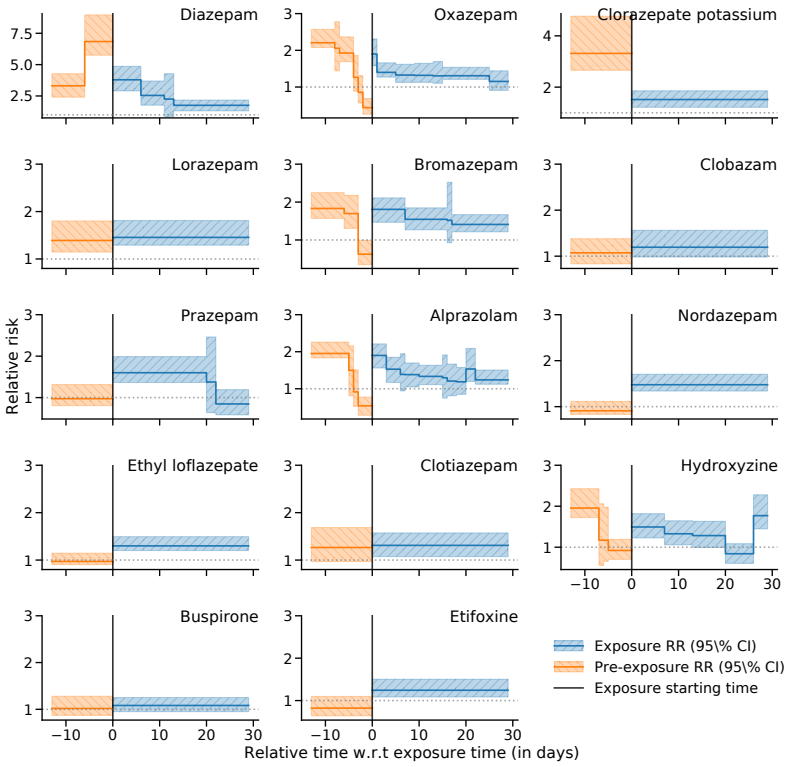


Figure III.B.34 – Fracture relative risk curves estimated before and after anxiolytics exposure after epileptic patients exclusion. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

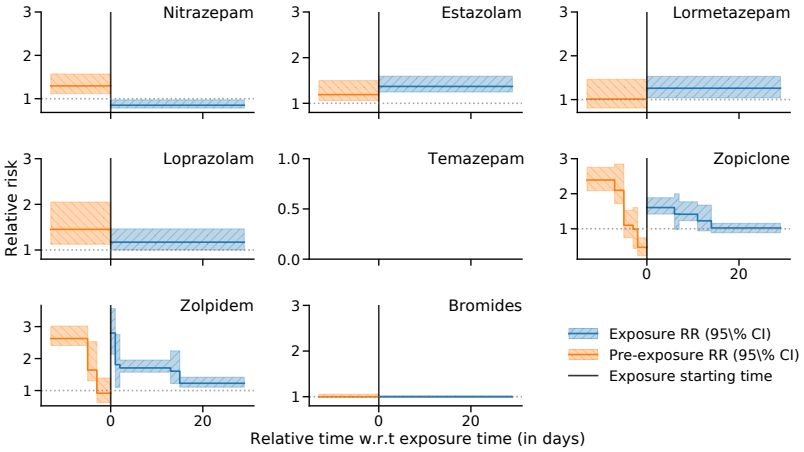


Figure III.B.35 – Fracture requiring surgery relative risk curves estimated before and after hypnotics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resppre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

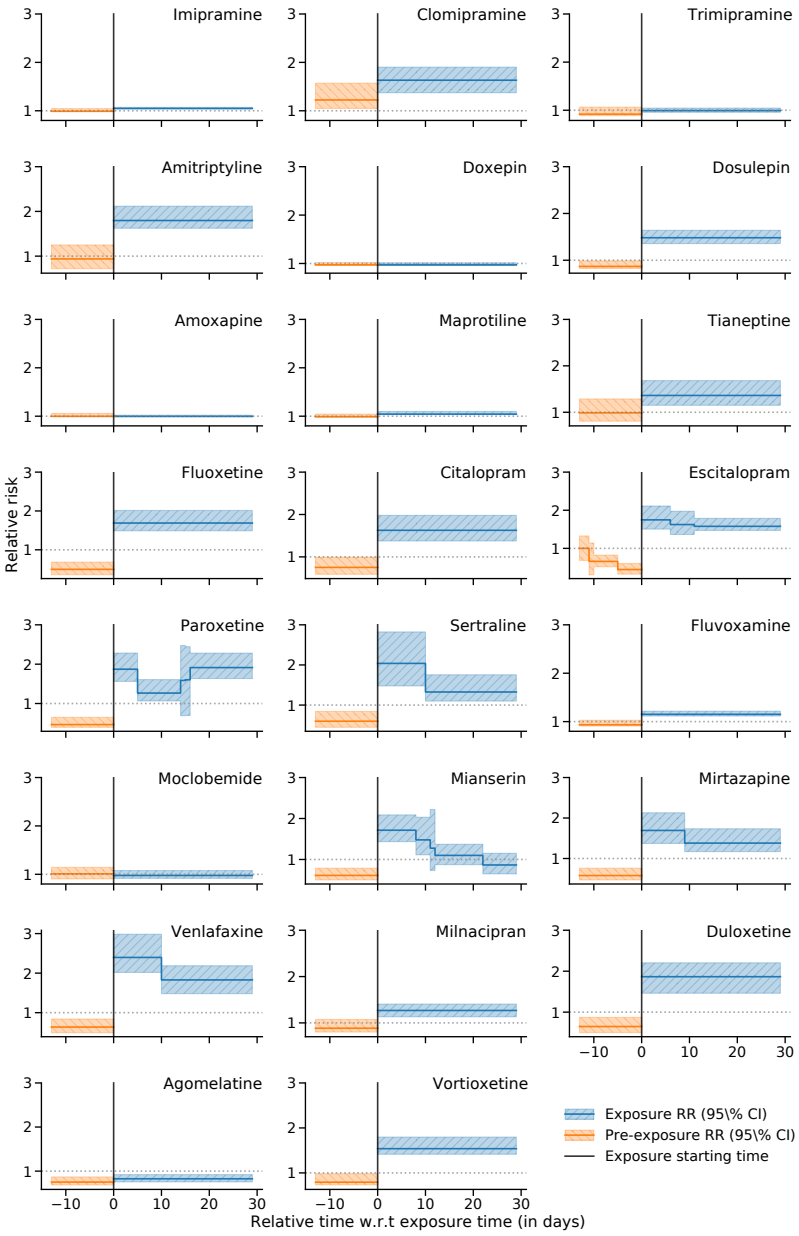


Figure III.B.36 – Fracture requiring surgery relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respórange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

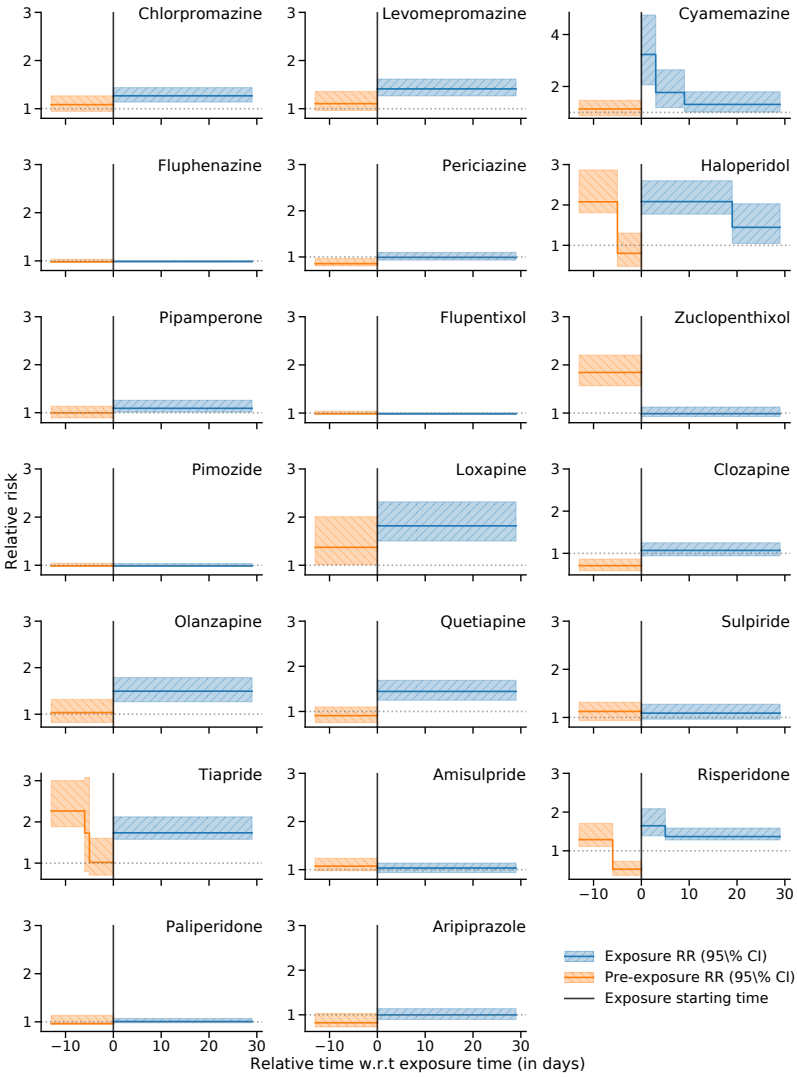


Figure III.B.37 – Fracture requiring surgery relative risk curves estimated before and after neuroleptics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (respórange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

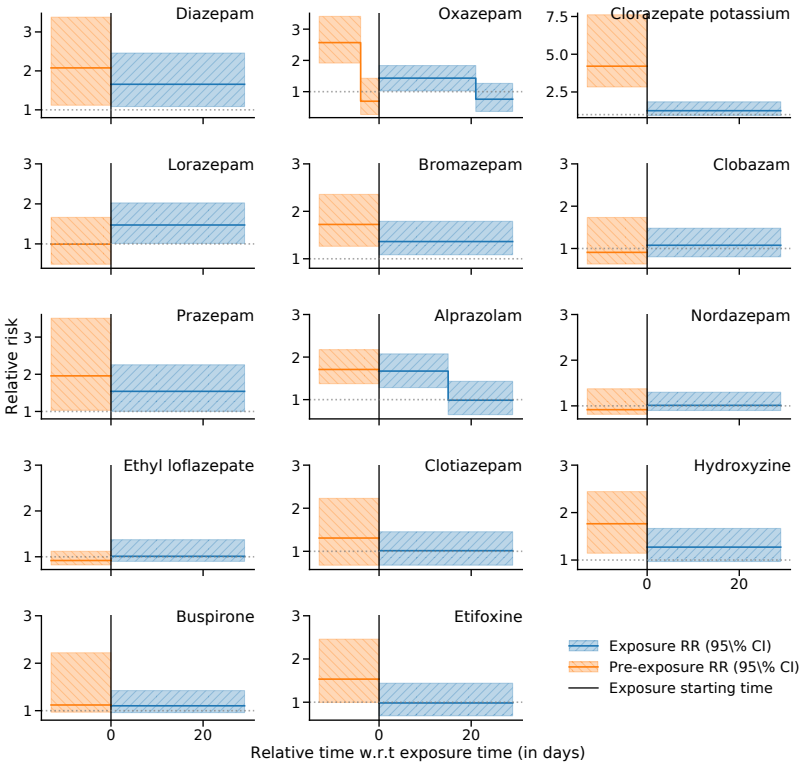


Figure III.B.38 – Wrist fracture relative risk curves estimated before and after anxiolytics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

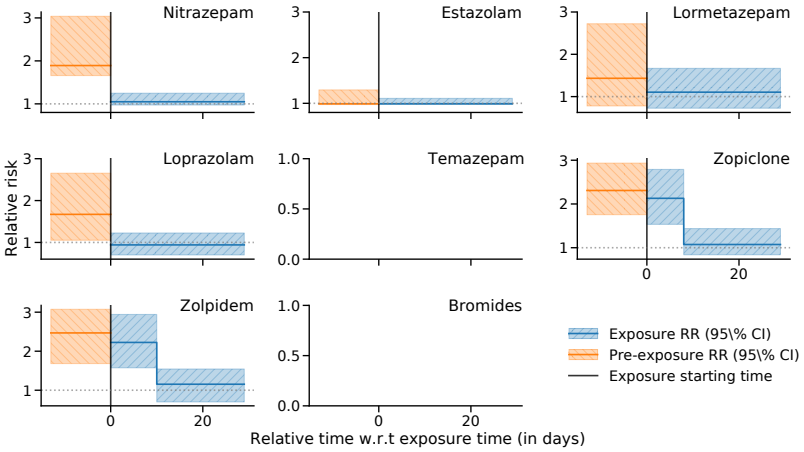


Figure III.B.39 – Wrist fracture relative risk curves estimated before and after hypnotics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

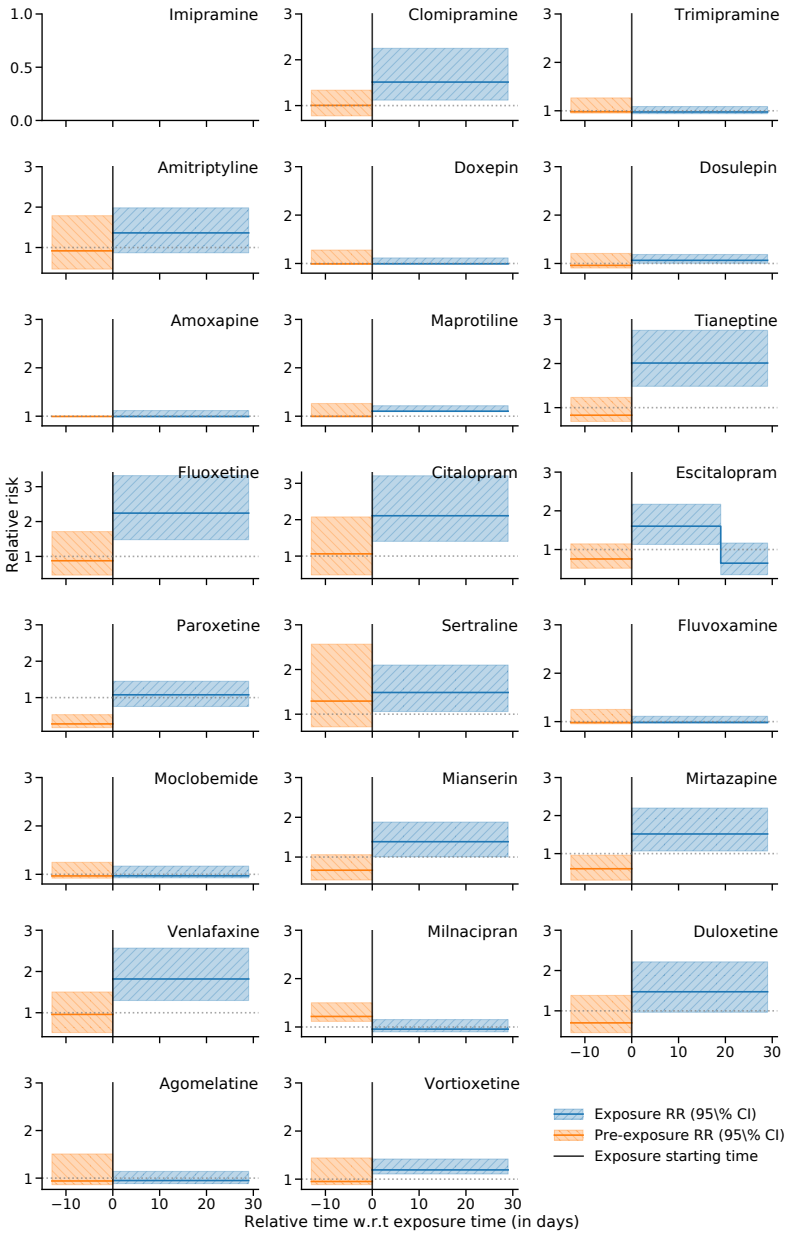


Figure III.B.40 – Wrist fracture relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

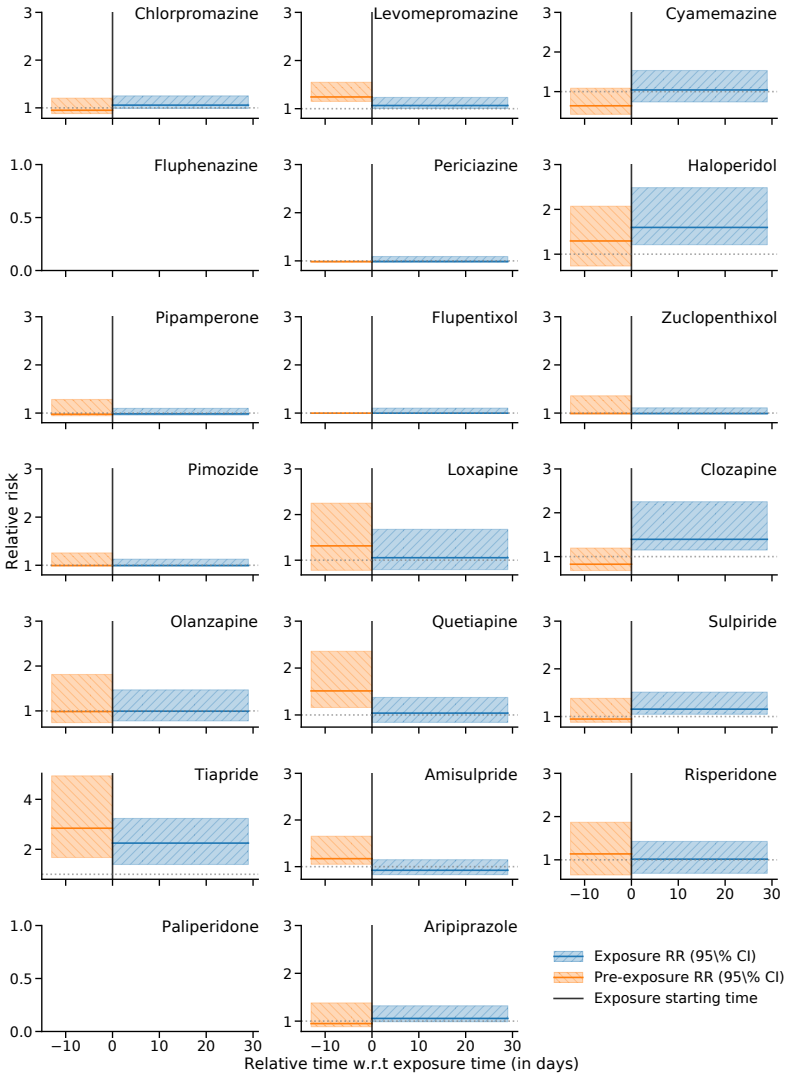


Figure III.B.41 – Wrist fracture relative risk curves estimated before and after neuroleptics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

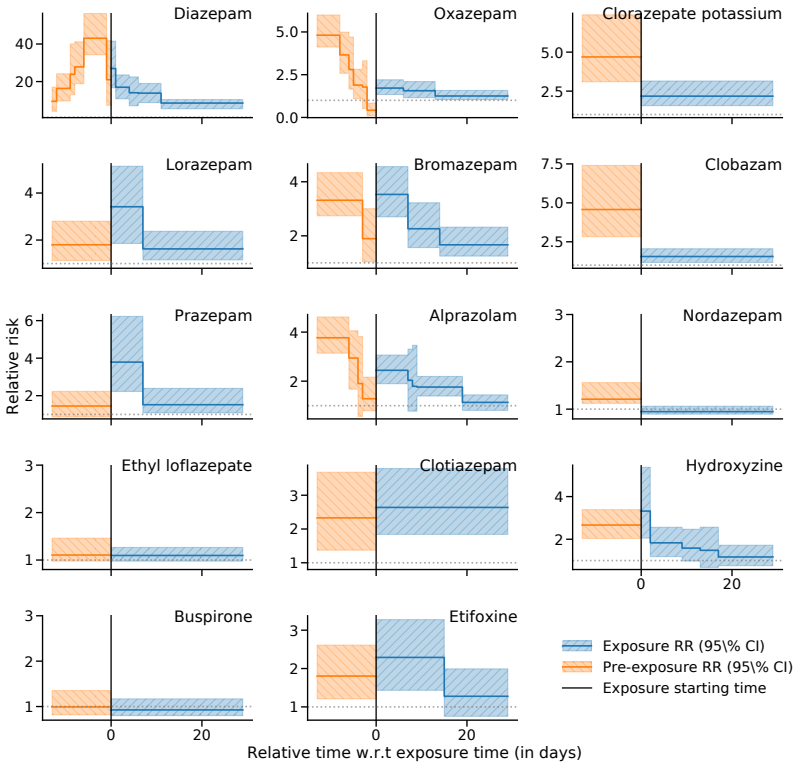


Figure III.B.42 – Spine fracture relative risk curves estimated before and after anxiolytics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

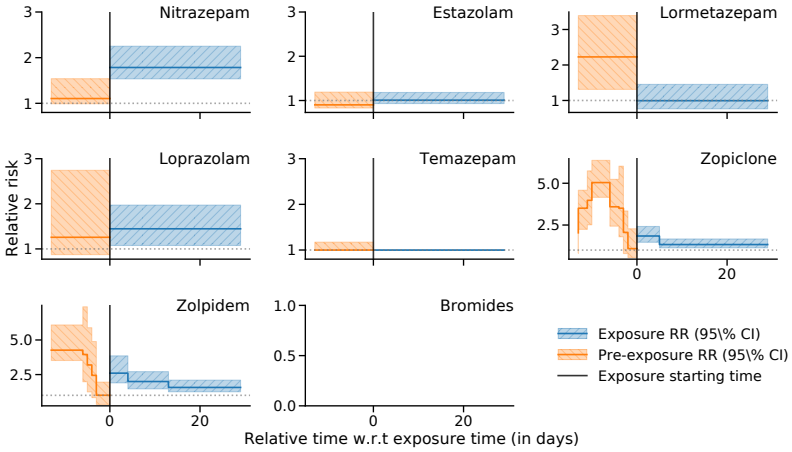


Figure III.B.43 – Spine fracture relative risk curves estimated before and after hypnotics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

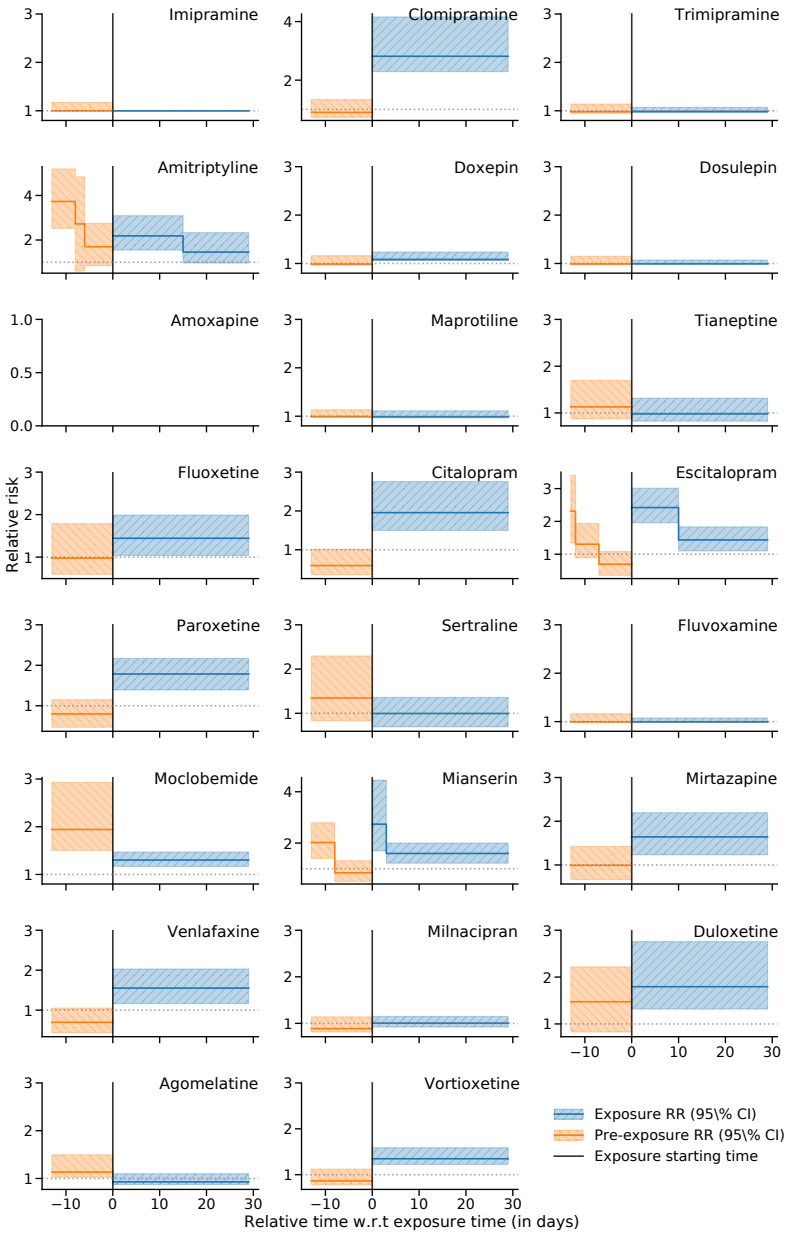


Figure III.B.44 – Spine fracture relative risk curves estimated before and after antidepressant exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched). Molecules considered in the three first rows are tricyclic antidepressants, followed by selective serotonin reuptake inhibitors in rows 4 and 5, and serotonin-norepinephrine reuptake inhibitor row 7.

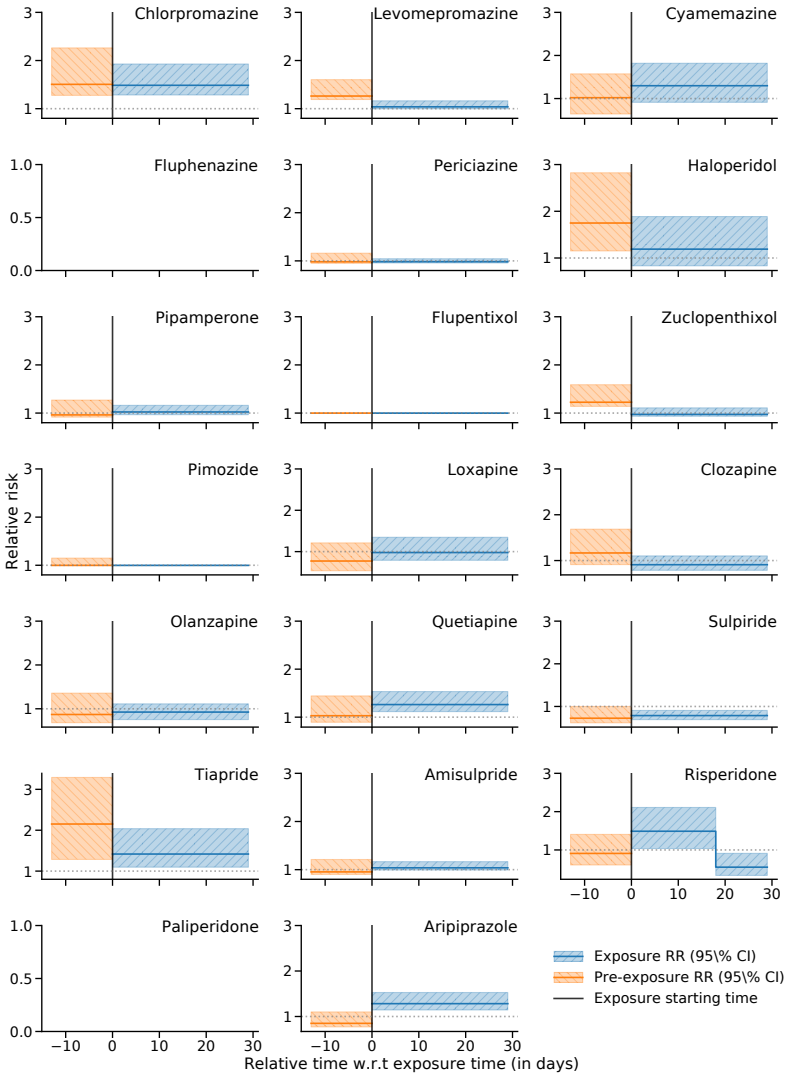


Figure III.B.45 – Spine fracture relative risk curves estimated before and after neuroleptics exposure. Exposure time is represented by the vertical black bar at $x = 0$. Post-exposure (resp. pre-exposure) relative risk is represented in blue (resp. orange) solid lines, with 95% Confidence Intervals (CI) depicted in blue right hatched bands (resp. orange left hatched).

III.C SCCS assumption assessment

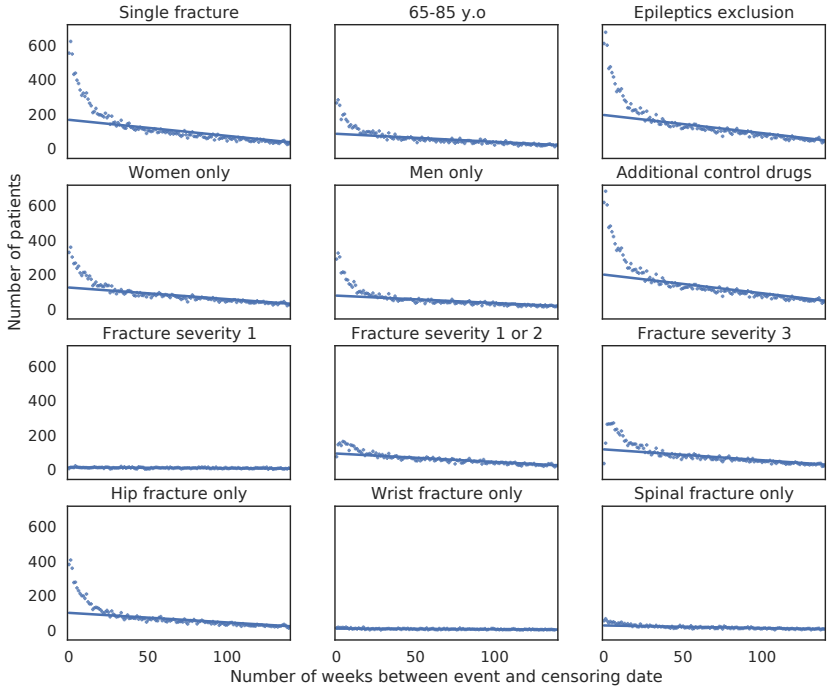


Figure III.C.1 – Robust regression (Huber) of censoring times versus event times. The horizontal axis represents the number of weeks between event date and censoring date, and the vertical axis the corresponding number of patients. Markers represent the patient counts corresponding to each week bucket, and the solid line represents the robust regression line. The closer the points are to the regression lines, the more likely the assumption of independence between event dates and observation dates is. Patients above the regression line are susceptible to have a death event correlated with a fracture event. In the all-fracture (reference) study featured in this paper, 9300 patients (7.35%) death event might be considered to be correlated to their fracture times. Note that the negative gradient of the regression line is explained by the events repartition in the study: if events are assumed to be uniformly distributed across the four years, there are fewer chances to observe events separated by 200 weeks than 25 weeks.

CHAPTER IV

WHICH ATTENTION MODEL AND UNSUPERVISED PRE-TRAINING STRATEGY FOR ELECTRONIC HEALTH RECORDS?

Motivated by recent advances in NLP (Natural Language Processing), this paper explores several unsupervised pre-training strategies and several transformer architectures for predictive modeling using structured electronic health records (EHR). We use the MIMIC-III dataset to predict in-hospital mortality, length-of-stay and phenotypes using physiological measurements and demographic information of patients admitted in intensive care units. We show that regular transformer models do not achieve good performances on several tasks despite using various pre-training strategies, while a graph representation combined with an attention mechanism and unsupervised pre-training turns out to efficient both in terms of computation time and performance.

Keywords: *Electronic Health Records, Transformers architectures, Unsupervised pre-training*

IV.1 Introduction

This paper focuses on *structured* EHR data, where we observe time-stamped sequences of medical codes (e.g. for diagnoses, medications, procedures) that describes the health pathways of patients. A parallel is often drawn between such data and NLP [Aya+20; SRB19], since both can be represented as sequences of tokens, corresponding to words or word pieces in NLP and to medical codes in EHR. Recent attention models and pretraining strategies resulted in considerable improvements on many NLP tasks [Dev+18; You+18]. In particular, transfer learning was proved, in the last few years, to be very effective for NLP, while this fact was already established

in computer vision [HR18]. A recent trend is a form of *unsupervised learning*, called *self-supervised* learning, which involves a pre-training step on a large unsupervised dataset, using a *pretext task*, followed by a fine-tuning step for specific supervised tasks. One of the most popular example is Bidirectional Encoder Representations from Transformers (BERT) [Dev+18], with numerous extensions [Dai+19; Lan+19; Liu+19; Yan+19].

Unsupervised pre-training. Obtaining labeled healthcare data can be either expensive [Shi+18] or scarce (e.g. rare diseases, see [MH20]), which makes end-to-end supervised training of deep architectures impossible in such cases. An answer to this issue can be *unsupervised pre-training*, which is a key ingredient of the recent successes in NLP mentioned above, but also for time series [FDJ19] and computer vision, where deep image encoders are pre-trained using self-supervised [DZ17] or *contrastive* [Che+20; OLV18] approaches.

Contributions. This is where this paper brings new contributions: we explore several transformer architectures together with several unsupervised pre-training strategies for structured EHR. Many combinations are compared using several downstream supervised tasks, in order to provide insights about the best general-purpose combination of an architecture and of a pre-training strategy. Pre-trained representations are fine-tuned with a single additional output layer, for the considered specific downstream task. This is performed using the freely accessible MIMIC-III database [Joh+16], that is featured in numerous publications, see [Har+19; Shi+18; Son+18] among many others. In particular, this paper includes many of the best practices for hyper-parameters tuning (see Section IV.2.4).

From NLP to EHR. While both text data and EHR are sequences of tokens from large vocabularies, let us highlight the following specific issues with EHR that do not exist in NLP. (1) The order of tokens in text is somewhat self-evident, while the ordering of tokens in EHR is specific to the medical practice. Temporal relationships between types of codes is a crucial component of EHR that does not exist in NLP. (2) EHR are not always direct recordings of the physiologies of the patients, but rather captures of their interactions with the healthcare system, resulting in feedback loops and reversed dynamics [HA13]. For instance, data might exhibit biological exams, followed by a treatment, other biological exams, a diagnosis code, and finally another treatment. In physiology, the disease precedes the symptoms, but the data might show the symptoms first (through exams or acts for example), followed by the actual identification of the disease [HAP11]. (3) EHR can contain much longer dependencies compared to text, for instance a diabetes diagnosis is a risk factor all

along a patient's life, or some surgeries can prohibit other interventions, even decades later [Shi+18].

Related works. As explained previously, adapting state-of-the-art architectures for NLP to structured EHR is a non-trivial task. To the best of our knowledge, only few relevant papers can be found in literature on this topic. BEHRT [Li+20] develops pre-trained models to predict the occurrence of any disease in future visits. It uses positional embeddings to distinguish different visits and adds an age layer to imply temporal orders. However, BEHRT only uses disease sequences besides basic demographic information, discarding other medical information such as lab exams and drugs consumption, which might hinder its reuse for other tasks. G-BERT [Sha+19] adapts MLM pretraining to align disease and drug representations within a single visit in order to predict medications from diseases and conversely. However, they discard order and temporal information in the process, making this approach unusable to perform forecasting tasks. Med-BERT [Ras+20] adapts BERT for pre-training contextualized embedding models on a larger cohort and longer visit sequences compared to BEHRT and G-BERT. Interestingly, this paper introduces the pretext task of prolonged length of stay in hospital (LOS) and fine-tunes the model on two tasks concerning disease-prediction. However, Med-BERT only exploits diagnosis information and does not include the elapsed time between visits, which can lead to an important a loss of information.

[Cho+20] introduce Graph Convolutional Transformer, which incorporates a self-attention mechanism. Medical visits are represented as graphs, which edges that are estimated by using self-attention. Self-attention is constrained to enforce specific chains of events such as observed symptoms cause diagnoses, diagnoses cause prescription, etc. The representations of visits are computed with convolutional graph networks over the estimated graphs. However, this approach supposes to have access to fine-grained information in the dataset, while symptoms information is for instance not often available in an EHR.

IV.2 Methods

Apart from patient demographics (e.g., age, gender, etc.) and some other static features, a structured EHR consists, for each patient, of a sequence of medical "events", such as diagnosis, medication codes, medical acts, etc.[1] These events can be part of an hospital stay or can be events from a "city" medical consultation. Each event is time-stamped with a precision called *time unit*, that depends on the database.

[1]Medical concepts used in EHR and associated codes are usually taken from pre-defined standards, such as the International Classification of Diseases (ICD)

Since the time unit is generally rather large (an hour or a day), many events are co-occurring, i.e. share the same time-stamp.

Timeline VS graph representation. A patient EHR with several types of events is illustrated in Figure IV.2.1A using a time-line representation. Another representation illustrated in Figure IV.2.1B and C uses a directed graph representation, where the time-line is replaced by successive time unit events and where the edges correspond to existing structural associations between events: next time unit event on the time-line, medical events associated to the same time unit, diagnosis (or treatment) events associated to a symptom event, etc. Time unit events with no associated medical events are not coded. Such a representation was introduced by [Cho+18] and inspired other works, such as [Cho+20] and [Het+19]. The graph representation used in this paper is similar to that of [Het+19], but sequences of events are modelled through temporal point processes therein, while we use transformer architectures in this paper. The choice of the representation is driven by the architecture used, as explained in the next Section.

IV.2.1 Models architecture

The models considered in this paper all share the same neural network architecture illustrated in Figure IV.2.2. Following the flow of the data, it contains four components: an embedding component, an encoder, a pooler and a final dense layer for operating a given task. We use a two-step training strategy: the encoder is first pre-trained in an unsupervised way using some pretext task, and the final dense layer is fine-tuned on some specific clinical supervised task.

Embedding component. Each event from the EHR representation corresponds to a code and/or numeric variables (see Figure IV.2.1A). These codes/variables are tokenized and embedded: each unique token is individually mapped to a low-dimensional embedding vector. The relative position (timestamp) of each event is encoded using fixed positional embeddings (*added* to each event embedding) following [Vas+17], in which we replace the ordered position number by the elapsed time relative to the timestamp of the first event of the sequence. Finally, a special token `[CLS]` is added at the end of the sequence and is embedded as well. Moreover, static features (including patient demographics, e.g. age and gender) are also embedded and summed out into a single vector, that is used as input to the Encoder component, see Figure IV.2.2.

Encoder. An encoder is used to encode the whole EHR representation into a new sequence with the same length. The static features embedding is then added to each

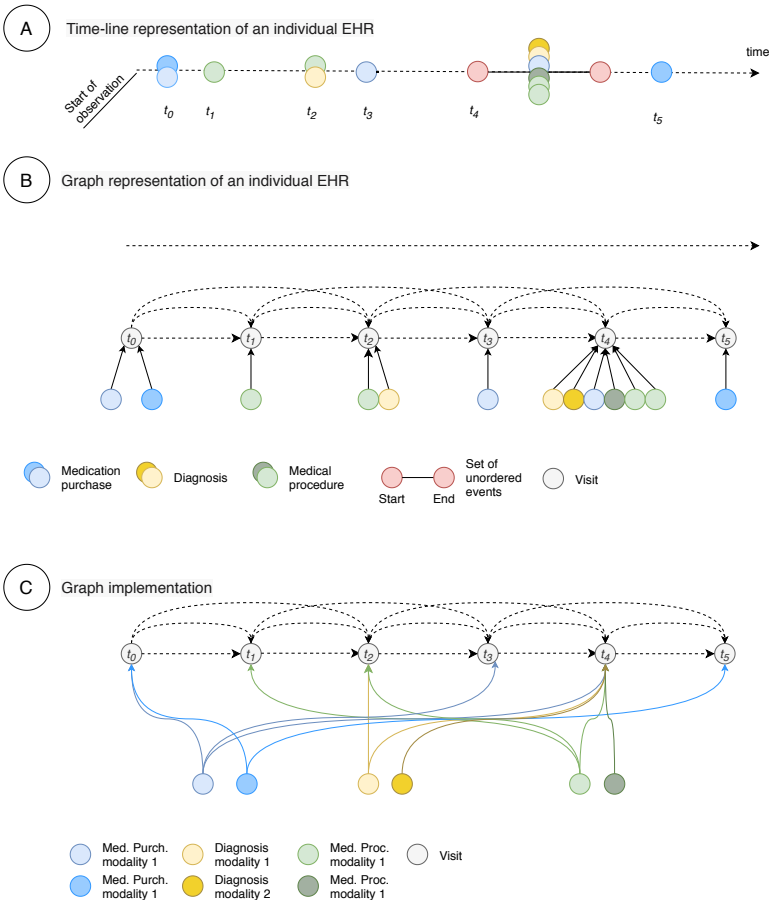


Figure IV.2.1 – Graph representation. A timestamped EHR sequence **(A)** can be represented by the graph **(B)**. A visit is created for each timestep in which medical events occur. Event nodes are created each time this event occurs during a visit. Visit nodes are initialized with the sum of the `[visit]` token embedding and the corresponding positional embedding, while event nodes are initialized with the embedding of the corresponding modality.
In practice, the graph is implemented as depicted in panel **(C)** to improve the representation sparsity. For a given EHR sequence, event nodes are created only once per observed modality (*event-modality nodes*). Each visit node in which an event occurred can attend to the corresponding event-modality nodes.
Note that there is no information flowing *into* these event-modality nodes when updating the graph through the layers. As such, there is no causality break nor data leak when using this representation. Event-modality nodes are *uniquely* learned by the embedding layer.

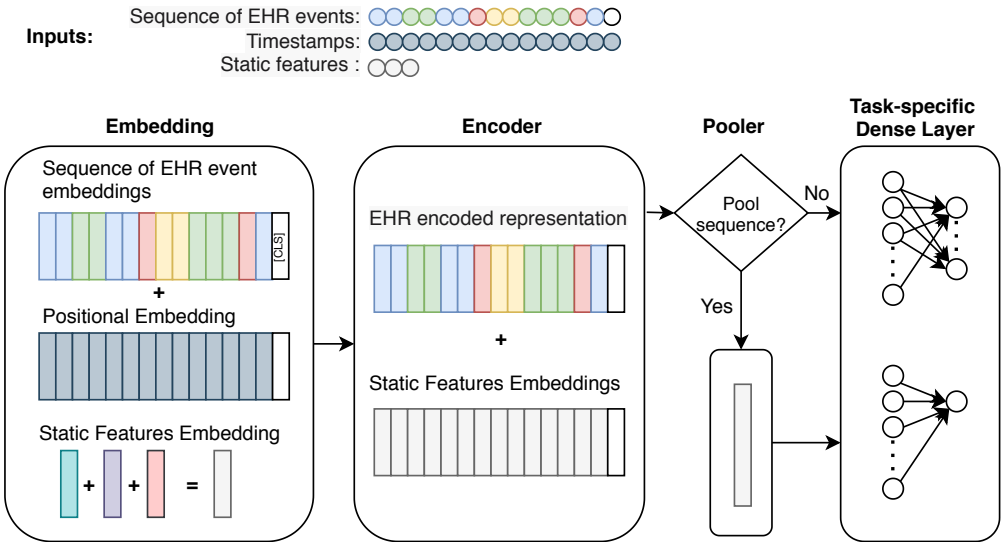


Figure IV.2.2 – Overview of the generic model architecture. The EHR representation (see Figure IV.2.1) is used as input data to an architecture with four components: an Embedding component, a Transformer-based Encoder, a Pooler and a final dense layer for operating a given pre-training or downstream task.

element of this new sequence.

We use and compare the following attention-based encoders. (1) The *Vanilla Transformer* [Vas+17], which allows to build a representation of an input sequence by stacking Multi-head Self-Attention (MSA) layers; (2) The *Linear Transformer* [Wu+20], which significantly reduces the memory footprint and scales linearly with respect to the sequence length compared to [Vas+17], allowing to feed entire sequences without length restrictions; (3) The *Graph Attention Network* (GAT) [Vel+17; Ye+19], which uses fully-connected graphs with a self-attention mechanism which does not involves queries and keys as MSA. For this encoder, we use the graph representation described in Section IV.2 and Figure IV.2.1 C).

In each case, *causal* attentions are used: at any given position, the attention mechanism is not allowed to put attention on any data involving *future* positions, so that the output sequence of the Encoder preserves causality. Detailed descriptions of each encoder are provided in Supplementary Material.

Pooler. As explained in Section IV.2.3 below, two types of downstream supervised tasks can be considered: (1) tasks that operate on each event embedding of the sequence coming out of the encoder (e.g. length of stay prediction) and (2) tasks that exploit the overall sequence (e.g. mortality prediction). A pooler is required only

for (2). We follow [Dev+18] where only the last element of the encoded sequence is kept (which explains the use of the `[CLS]` token above).

IV.2.2 Unsupervised pre-training

We consider the following strategies for unsupervised pre-training of the encoder: (1) *Masked Language Modeling* (MLM) [Dev+18], (2) *Triplet Loss* [FDJ19] and (3) *Contrastive Predictive Coding* (CPC or InfoNCE) [Che+20; Sun+19] that are summarized in Figure IV.2.3. A more detailed description of each approach is provided in Supplementary Material. In our experiments, an architecture is trained from scratch and independently for each pre-training strategy.

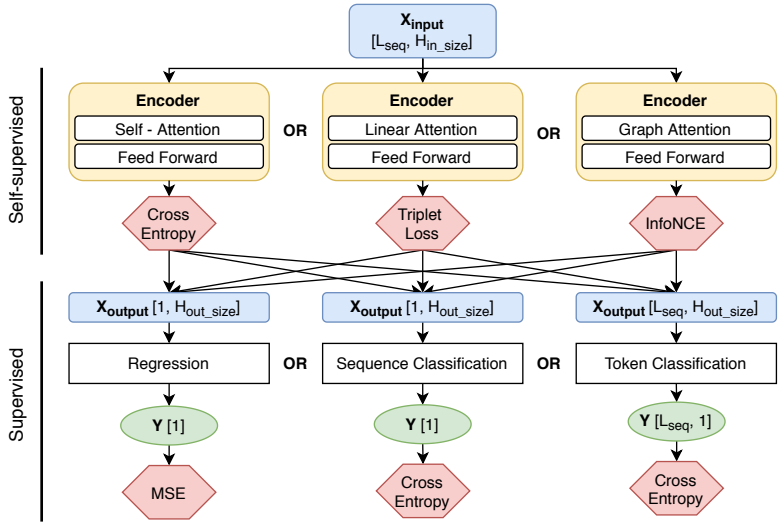


Figure IV.2.3 – Overview of the unsupervised pre-training and evaluation procedures. First, three Encoders are pre-trained separately in a unsupervised manner. The obtained representation for each token in the sequence is then passed into the Pooler if the downstream supervised task requires it. In the next step, we add a classifier network on the top of the Encoder which is trained in a supervised manner separately for each downstream task.

IV.2.3 Supervised fine-tuning, losses and metrics

All the combinations of encoder architectures and pre-training strategies are assessed using several clinical *supervised tasks* ([Har+19], see Section IV.3 below). For each model combinations and supervised tasks, we use and compare: (1) fine-tuning, using the supervised task, of the whole architecture, with the embedding component and encoder initialized with the weights learned during the pre-training phase;

(2) fine-tuning of the final dense layer only (the embedding component and encoder weights are fixed to the pre-trained values); (3) end-to-end supervised training of the whole architecture, with random initialization of all the weights.

Depending on the supervised tasks, we use the following standard losses and assessment metrics: cross-entropy loss for binary and multi-class classification, with AUROC and AUPRC metrics for assessment of binary classification tasks and Cohen's linear weighted kappa metric for multi-class classification.

IV.2.4 Hyper-parameters and training details

All hyper-parameters described below were tuned using cross-validation on the validation set. We describe below the resulting best-performing choices for each architecture. For Vanilla Transformers, we restrict the number of events in a sequence to 1024, and use hidden units of dimension 256, output units of dimension 512, 6 hidden layers and 4 attention heads. Linear Transformers use the same hyper-parameters as Vanilla Transformers except for the dimensions of the hidden units (128), the output units (256) and the use of GeLU [HG16] instead of ReLU activations. We use the implementations provided by [Kat+20] available at `https://github.com/idiap/fast-transformers`. The Graph attention network uses the graph representation of EHR where each token node is only connected to its corresponding time unit node and each time unit node is connected to the three previous time unit nodes. We use 4 layers with 2 heads for the first and fourth layers and 4 heads for the second and the third.

In all cases, we use dropout with $p = 0.1$ and batch normalisation[2]. We use the Rectified Adam (RAdam) [Liu+20] optimiser in combination with the Lookahead optimization algorithm [Zha+19] with learning rate 10^{-3} and momentum 0.9. We use a maximum of 500 epoch using early stopping with a patience of 25 epochs and tolerance 10^{-4}.

Each model is trained on a *single* GPU, using mini-batches of 512 sequences. When the batch size is too large for the GPU memory, gradient accumulation is used to counterbalance the reduction of the batch size. For CPC pre-training, we follow [Che+20] and use larger batch sizes (1024) in order to have more diversity in the sampled negative examples (more details on CPC can be found in Supplementary Material). For the three fine-tuning approaches described in Section IV.2.3, all the training hyper-parameters are kept the same besides the mini-batch size (128).

[2]dropout is applied after each layer and layer normalization is used after each attention layer, excepted for the `[CLS]` Pooler

IV.3 Experiments

In this section we describe the data used in our experiments (MIMIC-III) together with the considered supervised tasks. MIMIC-III (Medical Information Mart for Intensive Care) is a large single-center database containing de-identified data about patients admitted to intensive care units (ICU) [Joh+16] between 2001 and 2012. Following [Har+19; Son+18], we use a cohort of 33,798 unique patients with a total of 42,276 hospital admissions and ICU stays. Population selection, features and labels are generated following [Har+19], where patient data is divided into separate episodes containing both time-series of events and episode-level outcomes. This results in 17 longitudinal features, among which 4 are categorical. Continuous features are bucketized using inter-decile intervals computed on the training set. The sample sizes for each task are given in Table IV.3.1. Training, validation and testing sets are respectively 70%, 15% and 15% of the ICU stays, reusing the same sampling as [Har+19]. ICU stays with less than 5 events are excluded.

Table IV.3.1 – Number of patients and ICU stays from MIMIC-III used in our experiments. ICU stays with less than 5 events are excluded.

Cohort	# patients	# ICU stays	# Excluded ICU stays
Pre-training, LOS, PHE	33 597	41 702	192
IHM	18 064	21 079	60

We consider 3 clinical prediction tasks[3] from [Har+19].

(i) *In-Hospital Mortality* (IHM): the outcome is a binary variable indicating wheter a patient dies during a given ICU stay or not. It is treated as a binary classification problem. True mortality labels are curated by comparing the times of death, hospital admission, and discharge. The mortality rate within the cohort is 13%.

(ii) *Length-of-Stay* (LOS): the outcome is the remaining time spent in ICU. It is bucketized into ten buckets (≤ 1 day; 1; 2; ... ; 7 days, $[1, 2)$ weeks; ≥ 2 weeks) and is considered as a 10-class classification problem.

(iii) *Phenotyping* (PHE): the outcome is a category corresponding to one of 25 diseases. It is treated as a classification problem and called *acute care phenotyping*. The disease is predicted retrospectively from data about the ICU stay of a patient. The data contains 25 diseases, including 12 critical respiratory/renal

[3]We do not use *decompensation*, since it is highly correlated with IHM and leads to a highly unbalanced binary classification problem which do not provides more insights than the ones considered here.

failures, 8 chronic conditions such as diabetes or atherosclerosis, and 5 "mixed" conditions such as liver infections. Patients with multiple phenotypes are excluded.

IV.4 Results

All the combinations of the considered architectures (Section IV.2.1) and pre-training strategies (Section IV.2.2) are compared using supervised tasks (Section IV.2.3) related to clinical prediction tasks (Section IV.3) on MIMIC-III. The corresponding metrics, computed on the test set, are reported in Table IV.4.1.

Table IV.4.1 – Test metrics obtained by all combinations of architectures and pre-training strategies (rows) on clinical prediction tasks (columns) using the MIMIC dataset.. Due to its underwhelming performances, GAT was not trained for all the tasks and training strategies to avoid computation waste. * The Length Of Stay (LOS) task in [Har+19] slightly differ from ours. They predict the remaining LOS at each hour, while our experiments do so each time there is a new patient measurement. Thus, performance comparison cannot be made directly between these two approaches.

Encoder	In-hospital mortality AUPRC/AUROC	Length of Stay Kappa	Phenotyping AUROC
End-to-end supervised			
Multi-task LSTM [Har+19]	**0.533/0.870**	**0.450***	**0.774**
Vanilla Transformer	0.394/0.809	0.535	0.736
Linear Transformer	0.355/0.790	**0.584**	0.676
GAT	0.132/0.528	0.218	0.503
MLM Pre-training			
Vanilla Transformer	**0.409/0.817**	**0.554**	**0.749**
Linear Transformer	0.344/0.785	0.405	0.708
GAT	0.154/0.572	–	–
Triplet Loss Pre-training			
Vanilla Transformer	**0.357/0.781**	0.451	**0.729**
Linear Transformer	0.330/0.774	**0.577**	0.686
CPC Pre-training			
Vanilla Transformer	**0.391/0.805**	0.466	**0.741**
Linear Transformer	0.333/0.770	**0.521**	0.675

According to this table, we first note that GAT shows inferior performance on all tasks, using pre-training or not. Increasing k, the number of past visits GAT could attend to did not result in any improvement. As aggregating events into visits

according to a similar graph structure resulted in good representation in [Cho+20], this poor performance might be rooted in the attention mechanism. Indeed, GAT uses an attention formulation relying only on node similarity rather than the query, key, values mechanism used in MSA. Besides the performance aspect, the graph formulation was very effective in GPU memory usage, allowing to process the longest sequences and use larger mini-batches. Moreover, attention on a graph is easy to implement since no ad-hoc masking is required to enforce causality, and it easily handles sequences of varying lengths. Blending this approach with an attention mechanism more similar to MSA could be an exciting extension, as a similar approach resulted in promising results in NLP [Ye+19].

As explained in Section 3.1, the vanilla transformer could not handle long sequences under the memory constraints of a few GPUs due to its quadratic complexity in the sequences' length. In our experiences, limiting the length of the sequences seemed to hinder its performance. The linear transformer could handle longer sequences, but it did not result in performance improvements over standard MSA.

Fine-tuning with frozen encoder weights led to worse results than fine-tuning with unfrozen weights and are not reported here. We observed that it took only 5 to 15 epochs, depending on the task, to achieve good performances when performing fine-tuning with unfrozen encoder weights. The training for each architecture, pre-training strategy and prediction took less than 5 hours, except MLM, for which training could last up to two days. We observed that MLM improved the scores of end-to-end supervised Vanilla Transformer, while Triplet Loss and CPC pre-training led to minor improvements.

IV.5 Conclusion

This work proposes an extensive study of the combination of different transformer-based encoder architectures and unsupervised pre-training strategies inspired by recent advances in NLP and computer vision. Regarding triplet loss, the random sampling of triplets $x^{\text{ref}}, x^{\text{pos}}, x^{\text{neg}}$ might be an issue. Indeed, even simple models can quickly learn to choose between x^{pos} and x^{neg} when they are chosen at random. In this case, the average triplet loss quickly converges towards zero, resulting in very slow parameter updates [Wu+17]. Adapting the sampling strategy to EHR data could be a way of improving results on triplet loss. Contrastive pre-training might not have revealed all of its capabilities in our experiments since it was understood only recently for computer vision problems [Che+20] that data augmentation is a crucial ingredient in such unsupervised strategies. Building pertinent data-augmentation on EHR data remains, to the best of our knowledge, a fascinating open question that requires to be addressed by future works since it would, in our opinion, enable important advances in learning representations for health pathways in an unsupervised fashion.

Acknowledgements

This research benefited from the National Health Insurance Fund (CNAM) partnership with Data Science initiative at École polytechnique.

Authors' contribution

This work was co-authored by Anastasiia Kabeshova, Maryan Morel, Emmanuel Bacry and Stéphane Gaïffas. AK and MM conducted the experiments and drafted the manuscript. All the authors contributed to the research plan, reviewed the manuscript and approved the final version.

Appendix

IV.A Encoders

The encoder takes as input a sequence of token embeddings $\mathbf{e} = [e_1, \dots, e_n]$, where $e_i \in \mathbb{R}^D$ for $i = 1, \dots, n$ and outputs a sequence of contextualized embeddings with the same length.

IV.A.1 Vanilla transformer

The Transformer encoder proposed by [Vas+17] can be used to build a representation of an n-input sequence by stacking Multi-head Self-Attention (MSA) layers. Considering K layers, h heads and d-dimensional token representations, an MSA layer writes as follows:

$$\mathbf{Q}_i = \mathbf{H}\mathbf{W}_i^Q, \quad \mathbf{K}_i = \mathbf{H}\mathbf{W}_i^K, \quad \mathbf{V}_i = \mathbf{H}\mathbf{W}_i^V, \tag{IV.1}$$

$$\mathbf{V}'_i = \text{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}}\right)\mathbf{V}_i, \tag{IV.2}$$

$$\text{MSA}(\mathbf{H}) = [\mathbf{V}'_1, \dots, \mathbf{V}'_h]\mathbf{W}_i^O, \tag{IV.3}$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d\times d_v}$ and $\mathbf{W}_i^O \in \mathbb{R}^{hd_v\times d}$ are learned parameters and $\mathbf{H} \in \mathbb{R}^d$ is the MSA input. This structure is stacked on K layers as follows

$$\mathbf{Z}_k = \text{LayerNorm}(\mathbf{H}^k + \text{MSA}(\mathbf{H}^k))$$

$$\mathbf{H}^{k+1} = \text{LayerNorm}(\mathbf{Z}^k + \text{FFN}(\mathbf{Z}^k)$$

where FFN is a dense feed forward network and $\mathbf{H}^k \in \mathbb{R}^{n \times d}$ (resp. $\mathbf{H}^{k+1}$) is the input (resp. output) to the kth-layer. The input to the encoder is $\mathbf{H}^1 = \mathbf{e}$, the n-sequence of token embeddings.

IV.A.2 Linear transformer

A well-known issue with models based on self-attention is their quadratic complexity w.r.t the length of the sequence n. This comes from the fact that the attention mechanism may focus on any event of the overall sequence. The length of sequences in an EHR are approximately distributed according to a power law, which means that a significant proportion of health records have a very large number of events. This generally induces poor performances for Transformer-like models and can even lead to out-of-memory errors. Some recent approaches focused on dealing with long sequence without sacrificing efficiency. Towards this end, [Chi+19] introduced sparse factorizations of the attention matrix to reduce the self-attention complexity to $\mathcal{O}(n\sqrt{n})$. Locality-sensitive hashing can further reduce self-attention complexity to $\mathcal{O}(n \log(n))$ [KKL20]. Recently, [Kat+20] introduced the linear transformer model that reduces complexity to $\mathcal{O}(n)$ by using a kernel-based formulation of self-attention and the associative property of matrix products to calculate the self-attention weight. More precisely, the authors proposed to rewrite Equation (IV.2) as follows:

$$\mathbf{V}'_i = \frac{\phi(\mathbf{Q}_i)^T \sum_{j=i}^{N} \phi(\mathbf{K}_j)\mathbf{V}_j^T}{\phi(\mathbf{Q}_i)^T \sum_{j=i}^{N} \phi(\mathbf{K}_j)}, \tag{IV.4}$$

where $\phi(x)$ is a feature map associated to a kernel $k(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$. Note that the feature map $\phi(\cdot)$ is applied row-wise to the matrices $\mathbf{Q}$ and $\mathbf{K}$.

IV.A.3 Graph Attention Network

An alternative solution to Linear Transformer for reducing the quadratic complexity problem is to use a directed graph attention mechanism. The main idea is, somehow, to be able to specify the (limited) context accessible to each event (and more generally to each node at each step of the attention mechanism) using a directed graph as illustrated on the Figure IV.2.1B and C. This process is very similar to Graph Attention Network [Vel+17] or BP-transformer [Ye+19].

In our case, the MIMIC-III EHR database is organized in visits. A visit corresponds to a stay of a patient in an ICU. The directed graph is built as follows. We code each visit v using a node which stores the starting and ending dates (t_v^{start}, t_v^{end}) of the visit.

Each visit node is linked with direct edges to one or more child nodes corresponding to various events that occurred during this visit (drug prescription, medical exam, medical procedure, diagnosis, etc). Visits with no events are not coded in the graph. Each visit is linked to other visits with directed edges, whose weights express the number of days between the two visits. Assuming the visits are not overlapping and sorted according to their starting date (i.e. $v_i, v_j, t_{v_i}^{start} < t_{v_j}^{start}$), the elapsed time between visits is given by $w_{v_i \to v_j} = t_{v_i}^{start} - t_{v_j}^{start}$.

The *context* $\mathcal{V}(u)$ of a node u is defined as follows : (i) if u is a child node of a visit node, its context is the corresponding visit node and (ii) if u is a visit node its context will be the k previous visit nodes.

The input to the encoder is a set of node features, $h = \{h_1, \dots, h_N\}, h_i \in \mathbb{R}^F$, where N is the number of nodes, and F is the number of features in each node. The encoder produces a new set of node features $h' = \{h'_1, \dots, h'_N\}, h'_i \in \mathbb{R}^{F'}$ as its output. In [Vel+17] the graph structure is injected into the mechanism by performing masked attention. For each node i, and for each node j in the neighborhood of node i, one computes

$$\alpha_{i,j} = \frac{\text{LeakyReLU}(a^\top [\mathbf{W}h_i || \mathbf{W}h_j])}{\sum_{j' \in \mathcal{V}(i)} \text{LeakyReLU}(a^\top [\mathbf{W}h_i || \mathbf{W}h_j])}, \tag{IV.5}$$

where $||$ stands for concatenation and $\mathbf{W} \in \mathbb{R}^{d^2}$, $a \in \mathbb{R}^{2d}$, $h_j \in \mathbb{R}^d$ are parameters to be trained. Then, given a non-linearity σ, a single layer performs the following node update

$$h'_i = \sigma\left(\sum_{j \in \mathcal{V}(i)} \alpha_{i,j} \mathbf{W} h_j\right). \tag{IV.6}$$

The network builds nodes representations by stacking such layers. A multi-head version of this attention can be implemented by performing these operations k times in parallel at each layer, and then by performing an average of the representations in the last layer before applying the non-linearity σ.

In practice, GAT self-attentional layers can easily be parallelized over edges and the computation of output features can be parallelized over all nodes. The time complexity of an attention head is $O(|V|d^2 + |E|)$ where $|V|$ and $|E|$ are the number of nodes and edges in the graph. Note that using multi-heads multiplies the required storage by h. We used a sparse implementation of GAT available in the DGL library, which is supposed to be $O(|V| \times |E|)$ in terms of storage complexity.

IV.B Unsupervised Pre-training Strategies

We describe in this section the supervised and unsupervised strategies used in the paper.

IV.B.1 Masked Language Model

We apply MLM as in [Dev+18]. We select 15% of the codes at random, and modify them according to the following probabilities: 80% of chance that a code is replaced by the `[MASK]` token, 10% of chance that the code is replaced by a random code, and another 10% chance that it was kept unchanged.

IV.B.2 Triplet loss

To the best of our knowledge, the triplet loss for unsupervised learning has never been applied for pre-training purposes in health applications despite its proven effectiveness [Erh+10]. One approach is to use an unsupervised causal model, as proposed by [FDJ19]. The sampling algorithm extracts random sub-sequences x^{ref} and x^{pos} (a positive example) of a given sequence y_i, and samples K of x^{neg} (negative examples) that are chosen at random in different random time series y_j with $j \neq i$. Then, on the one hand, the representation of x^{ref} should be close to that of x^{pos}, while on the other hand, the representation of x^{ref} should be distant from the ones of x^{neg}. This leads to the minimization of the triplet loss, given by

$$\mathcal{L}_{\text{triplet}} = -\log(\sigma(f(x^{\text{ref}}, \theta)^\intercal f(x^{\text{pos}}, \theta))) - \sum_{k=1}^{K} \log(\sigma(-f(x^{\text{ref}}, \theta)^\intercal f(x_k^{\text{neg}}, \theta))),$$

where σ is the sigmoid function and $f(\cdot, \theta)$ is a deep neural network encoder, where the parameters θ are to be trained.

IV.B.3 Contrastive Predictive Coding

Contrastive Predictive Coding (CPC), as formulated in [OLV18], learns representations by training neural networks to predict the representations of "future" observations from those of "past" ones. The main idea of CPC consists of maximizing the mutual information between the encoded representations and not between the original labeled data. When predicting future information the authors propose to encode the target x (future) and context c (present) into a compact distributed vector representations (via non-linear learned mappings) in a way that maximally preserves the mutual information (MI) of the original signals x and c defined as

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)}. \tag{IV.7}$$

By maximizing the mutual information between the encoded representations (which is bounded by the MI between the input signals), CPC extracts the underlying latent

variables that inputs have in common. The architecture of CPC is as follows: first, a non-linear encoder f_θ maps the input sequence of observations x_i to a sequence of latent representations $z_i = f_\theta(x_i)$. Next, an auto-regressive model g_{ar} summarizes all $z \leq t$ in the latent space and produces a context latent representation $c_{i,t} = g_{\text{ar}}(z_i \leq t)$. The authors model a density ratio which preserves the mutual information between $x_{i,t+k}$ and $c_{i,t}$ as follows:

$$f_k(x_{i,t+k}, c_{i,t}) = \exp\left(z_{i,t+k}^T W_k c_{i,t}\right). \tag{IV.8}$$

Both the encoder and auto-regressive models are trained to jointly optimize a loss based on NCE, which is called InfoNCE. Given a set $X = \{x_1, \dots, x_N\}$ of N random samples containing one positive sample from $p(x_{t+k}|c_t)$ and $N-1$ negative samples from the distribution $p(x_{t+k})$, we optimize

$$\mathcal{L} = -\mathbb{E}\Big[\log \frac{f_k(x_{i,t+k}, c_{i,t})}{\sum_{x_j \in X} f_k(x_j, c_t)}\Big]. \tag{IV.9}$$

The objective set by this loss function is for the prediction $\hat{z}_{i+k}$ to be most similar to the one positive sample z_{i+k} among a set of randomly selected negative samples z_l [Hén+19]:

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{i,k} \log\Big(\frac{\exp(\hat{z}_{i+k}^T z_{i+k})}{\exp(\hat{z}_{i+k}^T z_{i+k}) + \sum_l \exp(\hat{z}_{i+k}^T z_l')}\Big). \tag{IV.10}$$

CONCLUSION

The works presented in this thesis aim to push forward the use of large observational databases in healthcare. While such databases are very rich and become more available, they pose numerous technical and methodological challenges. In the previous chapters, some of them were addressed through contributions to several fields.

Chapter I presents SCALPEL3, an open-source framework accelerating medical concept extraction. This framework provides scalability gains thanks to a combination of data denormalization and columnar storage on a distributed architecture of commodity hardware. A growing library of medical event extractors encapsulates the knowledge needed to fetch relevant information from a large volume of administrative data. Continuous integration combined with careful unit and functional tests ensures the robustness of these extractors. A suite of automated statistics helps to monitor data manipulations, preventing data loss and errors. Finally, powerful high-level abstractions ease the interactive manipulation of cohort data. Studies using SCALPEL3 require only a few lines of readable code, resulting in easier debugging and better reproducibility. This framework is now used at the agency collecting SNDS data, at the French Ministry of Health, and soon at the National Health Data Hub in France.

Chapter II introduces ConvSCCS, a new model designed to highlight associations between a counting process and several longitudinal features. The model relies on a conditional Poisson process, resulting in robustness to non-longitudinal confounding variables. Using a convolution between step functions and temporal events gives the model sufficient flexibility to fit various relationships between the target process and multiple longitudinal features. Finally, a combination of group lasso and total variation penalties provides automatic feature selection and eases interpretability.

ConvSCCS was used to perform adverse drug reaction detection using data from the *Système National de Données de Santé* (SNDS), a massive observational database. It was first tested by recovering a known association between bladder cancer and a specific antidiabetic molecule in Chapter II. This model was then used to screen numerous anxiolytics, hypnotics, antidepressants, and neuroleptics for fracture risk among the elderly (Chapter III). This study results confirm previous results on antidepressants while revealing exciting biases by estimating pre-exposure and post-exposure longitudinal risk curves. These biases are tied to the studied molecules'

prescription context and bring a new light on previous results on anxiolytics and hypnotics.

Even if the volume of available healthcare data is growing, labels remain either rare or costly to obtain. Chapter IV tried to build generic representations using self-supervised learning to leverage existing troves of unlabeled healthcare data. This work presents several strategies to pre-train attention models on EHR data. These approaches were tested through several experiments using MIMIC-III data. While the resulting performance is not satisfying yet, this work sets a fertile ground for future research to grow.

The methods proposed in this thesis bring answers to some of the challenges posed by repurposing administrative healthcare data, with practical applications to drug safety. They also set sound foundations for future research leveraging observational healthcare data.

Future research could pursue the development of useful representations for irregularly sampled longitudinal data. While the graph representation of electronic health records has not resulted in good performances, it has interesting properties in computational cost and memory usage. As graph representations led to good results in non-longitudinal tasks [Cho+20], there should be a way to derive performant and effective graphs models in a longitudinal setting.

The models tested in Chapter IV rely solely on attention mechanisms performing successive aggregations of vectorized representations. Such models were consistently less performant than recurrent neural networks (RNNs) combined with imputation strategies. There might be a structural reason for this: attention models might struggle to learn concepts such as the derivatives of longitudinal measurements to capture their evolution. The computations behind the imputations and the recurrent neural networks might be emulated with graphs and graph convolutions such as EDGECONV [Wan+19]. This research path could eventually lead to a model with similar or better performances and computational properties than RNNs combined with imputation.

Furthermore, leveraging existing hierarchies in medical codes such as ICD-10 [Sha+19], as well as external information such as molecules properties [Wis+17] could lead to even richer representations.

Besides, while several works are building patient history representations, all of them rely on EHR data. One of the initial goals behind Chapter IV was to assess if good representations could be derived from claims data. Regrettably, this aspect was not developed due to a lack of human and computational time. While claims data have different properties than EHRs, claims datasets are usually much larger. Such large amounts of data could be a corpus of choice to pre-train very large models.

Finally, there is important work to be done on patient records simulation or at least deformation. As the dynamics behind EHR and claims data are quite hard to

derive, there are currently no robust techniques to perform data augmentation or generate synthetic healthcare data. Such tools are likely to be an essential missing part of the current efforts to build unsupervised patient records representations.

Appendix A

Résumé des contributions

Durant les vingt dernières années, plusieurs organismes étatiques ou privés ont accumulé des données sur les consommations de soins individuelles. La richesse de ces données en fait un outil de choix pour guider les politiques en santé publique. Cette thèse se concentre sur la modélisation de parcours de soins, à partir des données du Système National des Données de Santé (SNDS). L'accès à ces données résulte d'un partenariat de recherche conclu entre l'École Polytechnique et la Caisse Nationale de l'Assurance Maladie (CNAM). Cette dernière gère entre autres la collecte et la consolidation du SNDS.

La première contribution de ce travail consiste en l'élaboration et la publication de SCALPEL3, un système en libre accès qui facilite l'utilisation de bases de données observationnelles massives (BDOMs) à des fins de recherche (cf. Chapitre I). La seconde contribution consiste en un modèle permettant l'estimation de risques longitudinaux pour des évènements rares (Chapitre II). Ce modèle a été appliqué avec succès à la détection d'effets secondaires médicamenteux à court et à long terme (cf. Chapitre II and III). Enfin, la dernière partie de cette thèse évalue l'efficacité de différents modèles d'attention et stratégies de préentraînement pour l'apprentissage de représentations de parcours de soins de façon non supervisée (Chapter IV). Ce chapitre a pour but de résumer succinctement ces travaux, plus amplement détaillés dans les chapitres précédents.

A.1 Utilisation de bases de données observationnelles

Les essais randomisés contrôlés (ERCs) sont souvent considérés comme une référence lorsqu'il s'agit de produire des études statistiques en santé publique. Ils permettent de contrôler finement de nombreux biais lors de l'estimation des effets des traitements, grâce à une méthodologie éprouvée [Gro+04]. Cependant, les ERCs sont coûteux, et ne peuvent pas toujours être mis en place en raison de contraintes éthiques [Bee66; HS+79]. Par exemple, un ERC qui mettrait volontairement en danger un groupe de patients contreviendrait aux principes éthiques en vigueur. Par ailleurs, les ERCs peuvent être limités par leurs tailles d'échantillons, et ne sont souvent pas conduits sur de longues périodes en raison de leur coût. Ils peuvent ainsi

échouer dans la détection d'effets secondaires médicamenteux de long terme tels que l'association entre le pioglitazone et le cancer de la vessie [Azo+12 ; Neu+12].

Les études observationnelles peuvent contourner quelques uns de ces problèmes. En effet, la réutilisation de données historiques permet parfois de résoudre certains problèmes éthiques posés par les expériences sur des sujets vivants [Ros+10]. Les études observationnelles peuvent être menées sur des données administratives collectées par différents acteurs de la santé, comme les hôpitaux ou les systèmes d'assurance maladie. Les jeux de données qui en résultent couvrent souvent une large population avec un long historique, et sont beaucoup moins coûteux à produire que les données d'ERC [Mad+14]. Les données observationnelles permettent également d'observer des évènements rares, ou des sous-populations parfois difficiles à atteindre dans le cas des ERC. Enfin, elles fournissent une image réelle de la consommation de soins, qui peut compléter les conclusions d'ERCs au design très contrôlé [HA13]. Ainsi, l'utilisation de données observationnelles peut fournir des informations utiles pour l'élaboration de politiques de santé publique. Bien que les données observationnelles soient utilisées depuis longtemps, l'utilisation de bases de données massives ne s'est largement développée que durant les quinze dernières années [Mad+14] grâce aux progrès technologiques en informatique.

Les données observationnelles se divisent en deux catégories. Premièrement, les dossiers médicaux informatisés (*electronic health records*, EHRs) fournissent une image très détaillée, mais limitée dans le temps du parcours des patients. En effet, ils contiennent de nombreuses informations concernant la physiologie des patients ainsi que certaines de leurs habitudes de vie (consommation d'alcool ou de tabac par exemple) en plus de variables démographiques comme le genre ou l'âge. MIMIC-III (*Medical Information Mart for Intensive Care*) [Joh+16] en est un exemple, qui a donné lieu à de nombreuses publications. Les EHRs peuvent parfois contenir des données issues de capteurs, comme par exemple des electrocardiogrammes ou des données d'imagerie. Deuxièmement, des bases de données administratives telles que le SNDS contiennent des informations concernant des remboursements de soins. Ces données sont initialement collectées afin de permettre la comptabilité du système d'assurance maladie en France. Elles consistent en de nombreux évènements datés, qu'il s'agisse de délivrances de médicaments remboursés, d'actes médicaux et de diagnostics[1]. Ces informations sont bien moins détaillées que dans le cas des dossiers médicaux informatisés. Par exemple, elles ne contiennent pas de mesures physiologiques ou d'informations sur les habitudes de vie des patients ni sur les soins non remboursés.

Le système national de données de soins (SNDS). Le développement d'outils et d'algorithmes permettant d'utiliser les données observationnelles du SNDS constitue une part importante du travail effectué dans le cadre de cette thèse.

[1] Dans le SNDS, les diagnostics ne sont disponibles que pour les patients hospitalisés.

Pour constituer le SNDS, la CNAM collecte et vérifie l'ensemble des données fournies par les caisses primaires d'assurance maladie ainsi que certaines données hospitalières. Le SNDS contient ainsi les données de remboursements de soins d'environ 98,8 % de la population française [Tup+17a]. Initialement destinée à contrôler les dépenses en santé et l'utilisation des infrastructures de soins, cette base de données a commencé à être utilisée pour produire de nombreuses études épidémiologiques à partir de 2006 [Neu+12 ; Tup+17a].

Grâce à un historique de trois ans[2] et une grande rigueur dans la collecte des données, le SNDS permet de conduire des études sur une population quasi exhaustive [Tup+17a].

Les données du SNDS proviennent des Données de Consommation Inter-Régimes (DCIR) qui contient les remboursements des soins de ville, ainsi que du Programme de Médicalisation des Systèmes d'Information (PMSI) qui contient des données hospitalières. Ces deux bases de données sont massives, et possèdent une structure complexe. Le DCIR est normalisé autour de la notion de remboursement, alors que le PMSI l'est autour de la notion de séjour hospitalier. Les différents évènements médicaux qui y sont enregistrés sont codés à l'aide d'ontologies spécifiques telles que la Classification Internationale des Maladies (CIM). Le PMSI est lui-même divisé en quatre bases de données en fonction du type de service hospitalier. Les travaux présentés ici se concentrent sur le sous-ensemble Médecine, Chirurgie, Obstétrique et Odontologie (MCO) du PMSI.

L'accès au SNDS suppose une autorisation de la Commission Nationale de l'Informatique et des Libertés (CNIL), qui évalue la rigueur et l'intérêt des projets requérants. Par mesure de sécurité, le SNDS est stocké sur des serveurs Exadata dans les centres informatiques (*datacenters*) de la CNAM, et l'accès se fait par des postes sécurisés utilisant le logiciel SAS Enterprise Guide [Sup76]. Bien que ces modalités d'accès permettent la réalisation d'études épidémiologiques et économiques, elles sont très contraignantes d'un point de vue méthodologique. Grâce au partenariat conclu entre l'École Polytechnique et la CNAM, il a été possible d'utiliser un groupement de serveurs (*computer cluster*) de recherche situé dans le centre informatique de la CNAM.

Chaque année, le SNDS enregistre environ 20 milliards de traces de remboursements. Le traitement de ces données peut donc s'avérer délicat en raison de leur volume et de leur complexité. En effet, comme il s'agit d'informations administratives, l'obtention des évènements médicaux nécessite un traitement qui suppose de nombreuses jointures comme illustré par la Figure A.1.1. Par ailleurs, l'identification de ces concepts nécessite également une très bonne connaissance de subtilités administratives et techniques relatives au SNDS, qui peuvent rapidement représenter une barrière à l'exploitation de ces données.

[2] Auxquels s'ajoutent vingt ans d'historique, accessible dans certains cas.

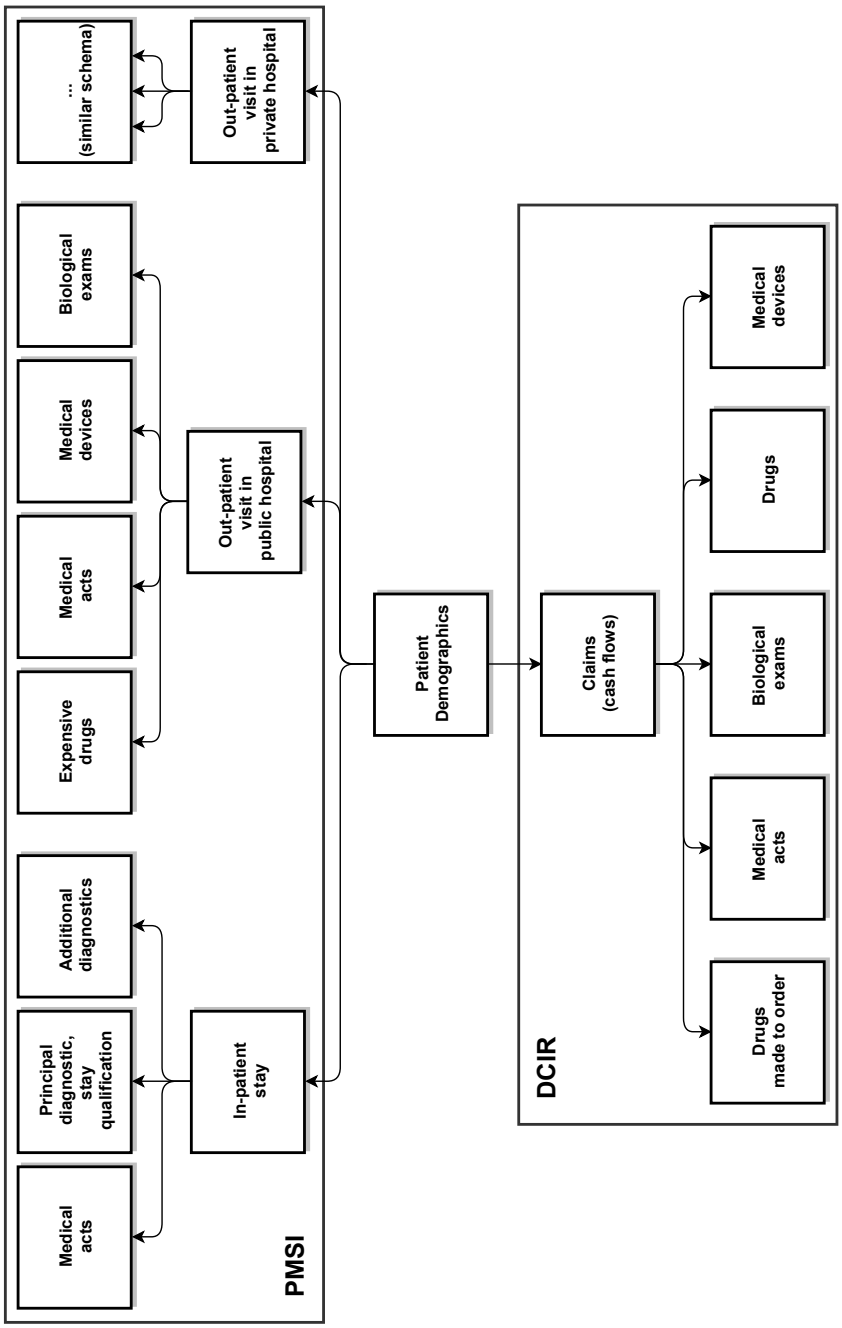


FIG. A.1.1 – Structure simplifiée du SNDS. Cette illustration représente les tables principales du DCIR et de PMSI MCO. Chaque rectangle représente une base de données, tandis que les flèches montrent quelles sont les jointures qui peuvent être réalisées.

L'extraction de concepts médicaux ne peut pour le moment pas être réalisée à l'aide d'algorithmes d'apprentissage [Ban+18], et requiert donc un traitement manuel assez lourd (c.f. l'identification des cancers de la vessie dans [Neu+12]). Une erreur dans l'identification des évènements médicaux peut ajouter un bruit non négligeable dans les données, avec les conséquences négatives que cela peut avoir sur la performance de modèles d'apprentissage.

Concernant le SNDS, cette tâche est aujourd'hui principalement réalisée à l'aide de programmes complexes sous SQL et SAS.

A.1.1 Contribution : un logiciel d'extraction rapide et reproductible de concepts médicaux

En premier lieu, cette thèse a contribué au développement de SCALPEL3, une suite logicielle facilitant l'extraction de concepts médicaux du SNDS ainsi que la manipulation de données de cohorte.

Contrairement aux approches existantes [Hri+15; Mur+10] qui reposent sur des modèles de données normalisés, nous avons fait le choix de dénormaliser la base de données afin de ne procéder aux jointures qu'une seule fois. La table qui résulte de ces jointures est très volumineuse. Afin de pouvoir la manipuler aisément, elle est stockée en utilisant un format orienté colonne ([Voh16]), ce qui permet de bénéficier d'un facteur de compression important lorsque des valeurs sont répétées sur plusieurs lignes [Mel+10]. Enfin, le traitement des données est implémenté de façon à pouvoir être distribué sur un cluster de calcul. Ces techniques qui ont déjà prouvé leur efficacité pour l'analyse de données de navigation internet [Bon+17] sont ici adaptées aux bases de données massives en santé. La structure de cette suite logicielle est illustrée Figure A.1.2, et développée dans le Chapitre I.

Un ensemble d'extracteurs de concepts médicaux affine successivement la version dénormalisée du SNDS de la façon suivante :

(i) Identification des colonnes pertinentes.

(ii) Filtrage des valeurs nulles, et éventuellement par valeurs.

(iii) Conversion des résultats en un schéma préétabli.

Ces trois opérations sont très rapides à effectuer sur des données orientées colonne, et peuvent être distribuées facilement. Un ensemble de `Transformers` peut par la suite combiner les données extraites pour former des évènements médicaux plus complexes. Un ensemble d'extracteurs et de `Transformers` permettent d'encapsuler une grande partie des connaissances nécessaires à l'identification de concepts médicaux, réduisant ainsi l'une des barrières à l'utilisation de données SNDS. Ce code

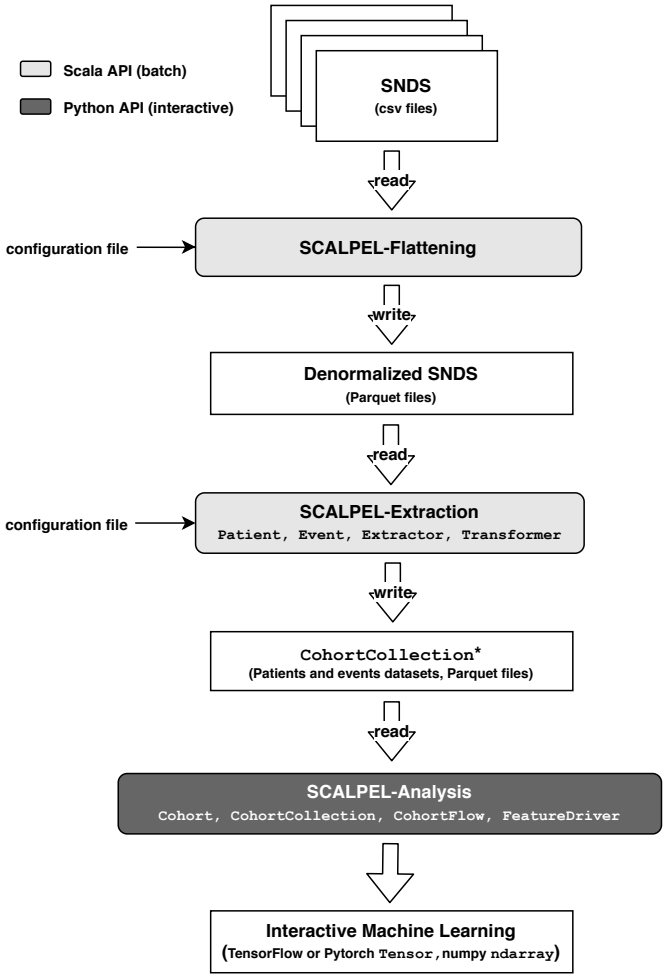


FIG. A.1.2 – Architecture de SCALPEL3. SCALPEL3 est divisé en trois librairies indépendantes à source ouverte, qui peuvent être utilisées indépendamment ou en combinaison. SCALPEL-Flattening, implémentée en Scala/Spark, dénormalise les bases de données en entrées (au format CSV) et produit une seule table stockée au format Parquet. Ensuite, SCALPEL-Extraction, implémenté en SCALA/Spark, extrait des concepts médicaux de cette table. Enfin, SCALPEL-Analysis implémentée en Python/PySpark permet de manipuler ces concepts de façon interactive, et d'alimenter des algorithmes d'apprentissage automatique.

est testé à l'aide de tests unitaires et fonctionnels afin de garantir la qualité des résultats. Ces tests, associés à la gestion rigoureuse des versions de SCALPEL3, facilitent notamment la reproductibilité des résultats obtenus avec cette suite logicielle.

Par ailleurs, SCALPEL3 fournit plusieurs abstractions de haut niveau détaillées dans le Chapitre I. Ces abstractions permettent de réaliser des opérations telles que des unions, intersections et différences entre des cohortes de patients. Ainsi, le

volume de code nécessaire à la production d'études statistique s'en trouve réduit. Ce code est donc plus facilement compréhensible et maintenable, favorisant ainsi sa diffusion. Enfin, un sous-module implémente de nombreux indicateurs statistiques, qui permettent de contrôler les opérations effectuées par SCALPEL3 tout au long du processus.

L'ensemble des opérations décrites ci-dessus est basé sur Apache Spark. Celles-ci peuvent donc être distribuées sur un cluster de calcul, en bénéficiant d'une scalabilité horizontale quasi linéaire (cf. Chapitre I).

Pour conclure, SCALPEL3 facilite la production d'études basées sur le SNDS tout en fournissant des gains en scalabilité. Cette suite logicielle est maintenant utilisée à la CNAM, à la Direction de la Recherche, des Études, de l'Évaluation et des Statistiques (Drees), et bientôt au sein de la Plateforme des Données de Santé (*Health Data Hub*).

A.2 Détection d'effets indésirables médicamenteux

L'amélioration de la détection d'effets indésirables médicamenteux (EIM) est une des promesses portées par l'utilisation de BDOMs [Sta+10]. Un EIM peut être défini comme la survenance d'un évènement indésirable suite à l'utilisation prolongée ou non d'un médicament. La survenance d'un EIM peut avoir des temporalités variées et être lié à une dose spécifique ou non [AF03]. Établir un effet lié à un dosage en utilisant les données du SNDS est ardu. En effet, l'estimation précise des doses utilisées par les patients est complexe, puisque les prescriptions ne sont pas consignées et que les quantités de médicaments délivrées ne sont pas individualisées[3] [Tup+17a]. Ainsi, cette thèse s'intéresse principalement à la temporalité des EIMs. Tandis que certains EIMs peuvent être indépendants du temps (p. ex. toxicité de la digoxine causée par une carence en potassium [AF03]), de nombreux EIMs peuvent survenir dès la première prise (p. ex. anaphylaxie après une prise de pénicilline) ou avec un délai plus ou moins important. Par exemple, les EIMs peuvent se produire à l'arrêt du traitement (p. ex. opiacés) ou avec un délai de plusieurs mois (p. ex. carcinogenèse) [AF03]. La vulnérabilité individuelle des patients peut également influencer la survenance d'EIMs. La Figure A.2.1 représente quelques exemples de risques longitudinaux d'EIMs.

Historiquement, la détection des EIMs après mise sur le marché repose sur des signalements effectués par les professionnels de soins ou les patients [Sch+16]. Ces signalements sont ensuite étudiés a posteriori par des études statistiques telles que [Neu+12]. Malheureusement, cette méthode qui repose sur des moyens hu-

[3] La quantité de médicament délivrée n'est pas égale à la quantité prescrite, en raison de la standardisation des quantités par boite.

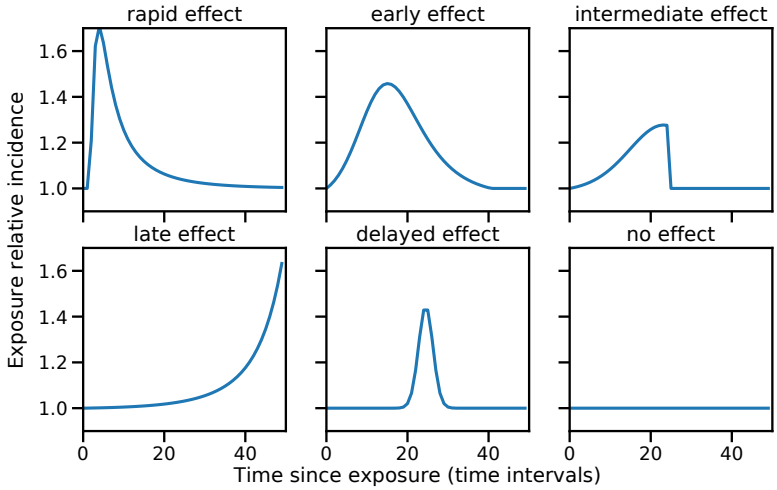


FIG. A.2.1 – Exemples de risques associés aux effets secondaires médicamenteux (EIM). La probabilité d'occurrence d'un EIM est proportionnelle à l'aire sous les courbes de risque correspondantes.

mains ne parvient à détecter qu'un sous-ensemble d'EIMs [Alv+98]. En effet, lorsque les EIMs sont rares, il peut être difficile pour un observateur humain de faire des rapprochements aboutissant à une alerte.

A.2.1 Défis méthodologiques

Plusieurs défis méthodologiques spécifiques aux BDOMs comme le SNDS compliquent la détection d'EIMs. En effet, les données de soins résultent de trois processus imbriqués [Alb+18 ; Hag+14] :

(i) Un processus épidémiologique, qui reflète la physiologie et la pathophysiologie des patients.

(ii) Un processus comportemental, lié aux habitudes de consommation de soins des patients et à leur hygiène de vie.

(iii) Un processus institutionnel, lié à la structure et au fonctionnement du système de soins.

Ainsi, l'utilisation de ce type de données doit prendre en compte quelques spécificités détaillées ci-dessous.

Informations manquantes et erreurs de codage. Bien qu'il soit très riche, le SNDS ne contient pas certaines catégories d'informations qui peuvent s'avérer cruciales selon l'objectif mené. On peut citer par exemple certaines caractéristiques socioéconomiques (revenu, statut marital), le style de vie (consommation d'alcool ou de tabac, pratique sportive, nutrition), les résultats d'examen, ou l'utilisation de médicaments non remboursés. Par ailleurs, les données enregistrées peuvent être imprécises. Par exemple, le codage d'un évènement de santé peut différer entre deux établissements de soins, pour des raisons purement administratives [HA13 ; Tup+17a].

Parcours spécifiques. La présence de parcours de soins spécifiques peut affecter les résultats d'une étude. En effet, si certaines molécules sont toujours délivrées dans un ordre bien précis, il peut être complexe de séparer leurs effets individuels [Hri+16].

Dynamiques inversées. Les données de soins capturent les interactions des bénéficiaires plutôt que leur physiologie. Il peut en résulter une inversion des dynamiques observées [HA13]. En effet, alors que les maladies précèdent leurs symptômes du point de vue physiologique, les données enregistrent les symptômes avant l'identification de la maladie [HAP11].

Échantillonnage non aléatoire. Les évènements de soins ne sont enregistrés que lorsque les patients interagissent avec le système de soins, c.-à-d. les données ne sont collectées que lorsque les patients ont des problèmes de santé.

En réponse à ces problèmes, il est possible d'utiliser des stratégies d'imputation spécifiques [Piv+14], d'utiliser les informations manquantes comme une information en soi[4] [Hag+14] ou d'utiliser des modèles flexibles. Cette troisième approche est celle retenue par cette thèse.

Ces spécificités méthodologiques peuvent produire de nombreux biais, le plus répandu d'entre eux étant peut-être le *biais par indication* dans le cadre des études observationnelles. Ce biais se produit lorsqu'une indication (p. ex. fièvre) cause à la fois une exposition (p. ex. paracétamol) et un effet néfaste (p. ex. asthme) [Aro+18]. Suivant cet exemple, ignorer le fait que certaines infection virales causant la fièvre augmentent le risque de développer de l'asthme pourrait conduire une étude à identifier une association fallacieuse entre le paracétamol et l'asthme.

De tels biais sont difficiles à identifier et à contrôler, notamment lorsque les prescriptions ne sont pas enregistrées comme dans le cas du SNDS. L'unique solution

[4]Par exemple, le nombre de visites d'un patient peut être utilisé comme un proxy pour son adhérence et son accès au système de soins

consiste pour le moment à adapter finement le design des études aux spécificités des données [Mad+14]. L'approche développée dans cette thèse s'appuie donc sur une identification précise des évènements médicaux, l'utilisation de modèles flexibles et une interprétation prudente des résultats.

Les outils mathématiques utilisés dans l'introduction de la thèse et développés tout au long des chapitres ne seront pas rappelés ici dans un souci de concision. Il en va de même pour les modèles et designs expérimentaux usuels décrits en introduction.

A.2.2 Approche retenue

Ni les signalements spontanés, ni les études de quantification de risque comme par exemple [Neu+12] ne peuvent être utilisés à grande échelle. Par ailleurs, les problèmes méthodologiques développés ci-dessus freinent le développement d'un système de détection d'EIM complètement automatisé [Mad+14]. Enfin, des tests de performance réalisés par l'*Observational Medical Outcomes Partnership* [Rya+12; Rya+13b] suggèrent que les approches dans lesquelles les sujets sont leurs propres témoins sont plus robustes que les approches qui comparent des groupes de sujets appariés. Ces observations ont guidé le développement d'un nouveau modèle afin d'améliorer la détection d'EIM. Les objectifs de ce modèle sont les suivants :

- Être facilement interprétable pour favoriser son adoption.
- Être suffisamment robuste pour limiter les biais causés par des informations manquantes dans le SNDS.
- Être capable d'utiliser des données creuses pour l'étude des évènements rares.
- Utiliser une notion simple d'exposition aux molécules.
- Utiliser les hypothèses les moins contraignantes possibles pour être applicable à plusieurs sujets d'étude.
- Pouvoir étudier de nombreuses molécules sur de grandes populations.

Supposons que les données sont disponibles sur une période d'observation globale $(a, b]$. Chaque patient $i = 1, \dots, m$ est associé à une période d'observation individuelle $(a_i, b_i] \subset (a, b]$, durant laquelle sont observés :

- Les temps d'occurrence $t_{i,1} < t_{i,2} < \cdots$ d'évènements d'intérêt (aussi appelés *outcomes* par la suite), ou de façon équivalente, un processus de comptage N_i, défini comme $N_i(t) = \sum_{k \geq 1} \mathbb{1}_{t_{i,k} \leq t}$ et $n_i = \int_{(a_i, b_i]} dN_i(t)$ le nombre total d'outcomes du patient i,

- un vecteur de d variable exogènes (features) longitudinales

$$X_i = \big(X_i(t) = (X_i^1(t) \ \cdots \ X_i^d(t)) \ : \ t \in (a_i, b_i]\big),$$

où $X_i^j(t)$ fournit des informations sur l'exposition du patient i à la molécule j au temps $t \in (a, b]$.

On se base sur la théorie des processus ponctuels, qui sert à modéliser des séries de points échantillonnés de façon irrégulière [Dal03]. Les travaux existants sur les modèles *Self-Controlled Case Series* (SCCS) [Far95 ; FW06 ; Sch+16] nous ont servi de point de départ. Ce type de modèle repose sur un schéma d'expérience où les patients sont leurs propres contrôles, et s'écrit comme un processus de Poisson conditionnel. Il repose sur les hypothèses suivantes [FW06] :

(i) Les variables explicatives sont exogènes, c.-à-d. le processus de comptage N_i n'influe pas sur les variables X_i ;

(ii) L'intervalle d'observation $(a_i, b_i]$ est indépendant de N_i ;

(iii) Le processus N_i est un processus de Poisson conditionnellement à $(X_i(t) \ : \ t \in (a_i, b_i])$.

Comme détaillé dans le Chapitre II, la vraisemblance du modèle SCCS s'écrit

$$\prod_{i=1}^{m} \prod_{k=1}^{n_i} \frac{\lambda_i(t_{i,k}, X_i)}{\int_{a_i}^{b_i} \lambda_i(s, X_i) ds} \tag{A.1}$$

où

$$\lambda_i(t, X_i) = \mathbb{P}(dN_i(t) = 1 \mid X_i) \tag{A.2}$$

est l'intensité conditionnelle du processus N_i pour $t \in (a_i, b_i]$. Ce modèle peut être compris comme un modèle de régression des outcomes dans N_i sur les variables longitudinales X_i. On suppose que cette intensité est multiplicative, c.-à-d. qu'elle s'exprime comme un produit. Il est intéressant de constater que le conditionnement par n_i génère deux propriétés intéressantes :

- *Scalabilité :* la vraisemblance ne dépend que des patients i pour qui $n_i \geq 1$, ce qui permet de travailler sur un échantillon réduit sans perte de puissance statistique [Far+11]. Cette propriété est particulièrement intéressante lors de l'étude d'évènements rares.

- *Robustesse aux variables non observées non longitudinales :* lorsque l'intensité $\lambda_i(t, X_i)$ peut s'exprimer comme un produit, les effets non longitudinaux s'annulent dans l'écriture de la vraisemblance. Cette propriété rend les modèles SCCS particulièrement robustes aux variables non observées comme la vulnérabilité individuelle des patients. En revanche, cela signifie également que les risques estimés sont relatifs à un risque de base non-estimé.

A.2.3 Contribution : Convolutional SCCS

L'utilisation des modèles SCCS nécessite des hypothèses concernant les périodes durant lesquelles les patients sont considérés comme étant à risque ou non [Far95]. Dans ce contexte, les variables longitudinales $X_i(t)$ expriment si le patient i est à risque ou non au temps t, pour une molécule et un EIM donnés. La définition de ces périodes de risques à partir des dates de remboursement de médicament nécessite des hypothèses assez fortes sur le délai d'occurrence des EIMs après le début de traitement. En cas de mauvaise définition de ces périodes, le modèle ne peut tout simplement pas estimer le risque recherché. La définition de telles périodes de risque est un problème complexe, qui devient presque impossible à résoudre lorsque l'on considère un ensemble de plusieurs molécules.

Pour résoudre ce problème, on se base sur une version discrétisée du modèle SCCS. On suppose que l'intensité λ est constante sur des intervalles de temps $I_k = (t_k, t_{k+1}]$, $k = 1, \dots, K$ qui forment une partition de $(a, b]$, pour $i = 1, \dots, m$. Pour simplifier les notations, on suppose que les intervalles I_k sont de longueur 1. En écrivant $\lambda_{i,k}$ la valeur de $\lambda(t, X_i(t))$ pour $t \in I_k$, et en définissant $y_{ik} := N_i(I_k)$, la vraisemblance du modèle SCCS discret s'écrit :

$$L(y_{i1}, \dots, y_{iK} | n_i, X_i) = n_i! \prod_{k=1}^{K} \left(\frac{\lambda_{ik}}{\sum_{k'=1}^{K} \lambda_{ik'}} \right)^{y_{ik}}.$$

Courbes de risque. Lorsqu'il est difficile de définir une période de risque, on peut être tenté d'utiliser une période plus grande que nécessaire pour être certain de détecter l'EIM étudié. Cette stratégie a cependant pour conséquence de diluer le risque sur l'ensemble de la fenêtre considérée [Xu+11], menant à un modèle incapable de détecter le moindre EIM.

On retient une approche différente, qui consiste à modéliser le risque par une courbe dépendante du temps plutôt que par un seul paramètre. Cette approche a déjà été utilisée dans [GWF16 ; GWF17 ; Sch+16], qui utilisent des splines pour modéliser l'effet des prises de médicament sur les intensités. Toutefois, ces modèles sont restreints à l'étude d'une molécule à la fois. Cela peut se révéler problématique, car les modèles SCCS sont sensibles à l'omission de variables longitudinales confondantes [MRM16 ; Sim+13].

Afin de formuler un modèle multivarié, on simplifie la formulation de la courbe de risque en utilisant des fonctions escalier au lieu des splines. L'intensité peut être définie par une convolution entre ces fonctions escalier et des indicateurs ponctuels d'exposition. En supposant l'intensité constante sur chaque I_k, celle-ci s'écrit

$$\lambda_{ik}(X_i) = \exp\left(\psi_i + \gamma_i + \phi_k + \sum_{k'=a_i}^{k} X_{ik'}^{\top} \theta_{k-k'} \right),$$

où X_{ik} est la valeur moyenne de $X_i(t)$ dans $t \in I_k$, $\theta \in \mathbb{R}^{d\times K}$, ψ_i est le risque du base du i et γ_i une somme d'effets individuels non longitudinaux. Le paramètre ϕ_k est un risque de base dépendant du temps, commun à tous les individus (par exemple le risque lié à l'âge).

On observe $l = 1, \dots, L_i^j$ dates de début d'exposition c_{il}^j et on introduit les variables $X_{ik}^j = \sum_{l=1}^{L_i^j} \mathbb{1}_{k=c_{il}^j}$, pour aboutir à l'intensité

$$\lambda_{ik}(X_i) = \exp\Big(\psi_i + \gamma_i + \phi_k + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k-c_{il}^j} \mathbb{1}_{[0,p]}(k - c_{il}^j)\Big). \tag{A.3}$$

La quantité $\exp(\theta_k^j)$ correspond au risque relatif d'une exposition à la molécule j, k périodes après le début de l'exposition. La vraisemblance s'écrit donc

$$L(y_{ik}, \dots, y_{iK}|n_i, X_i) = \prod_{k=1}^{K}\left(\frac{\exp\left(\phi_k + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k-c_{il}^j} \mathbb{1}_{[0,p]}(k - c_{il}^j)\right)}{\sum_{k'=1}^{K}\exp\left(\phi_{k'} + \sum_{j=1}^{d}\sum_{l=1}^{L_i^j} \theta^j_{k'-c_{il}^j} \mathbb{1}_{[0,p]}(k' - c_{il}^j)\right)}\right)^{y_{ik}}$$

et ne dépend que des paramètres associés aux variables longitudinales, à savoir θ pour les expositions et ϕ pour le risque de base longitudinal. Lorsque peu d'information sur la temporalité des EIMs est disponible a priori, combiner des courbes de risques flexibles et des indicateurs binaires d'exposition est la méthode de détection de risque la plus performante [GAB15].

Sélection de variables. La flexibilité de cette formulation de l'intensité (A.3) a un coût, puisqu'elle augmente fortement le nombre de paramètres à estimer. Cela peut notamment conduire à un phénomène de surentraînement du modèle. On utilise donc une stratégie de régularisation pour contraindre les paramètres, tout en aidant par ailleurs l'interprétation des courbes de risque. Pour ce faire, on combine les régularisations Group-Lasso et variation totale. On considère les groupes $\theta^j = [\theta_1^j \cdots \theta_p^j] \in \mathbb{R}^p$ de paramètres qui correspondent aux courbes de risques pour chacune des molécules étudiées $j = 1, \dots, d$ à des retards différents $k = 1, \dots p$. La régularisation s'écrit

$$\text{pen}(\theta) = \gamma_{\text{tv}} \sum_{j=1}^{J}\sum_{k=1}^{p-1} |\theta_{k+1}^j - \theta_k^j| + \gamma_{\text{gl}} \sum_{j=1}^{J} \|\theta^j\|_2. \tag{A.4}$$

L'effet de la variation totale unidimensionnelle introduit une contrainte sur la variabilité des courbes de risque, comme illustré dans la Figure A.2.2. Le group Lasso permet quant à lui d'effectuer une sélection de variables, en annulant les courbes associées à des molécules sans effet.

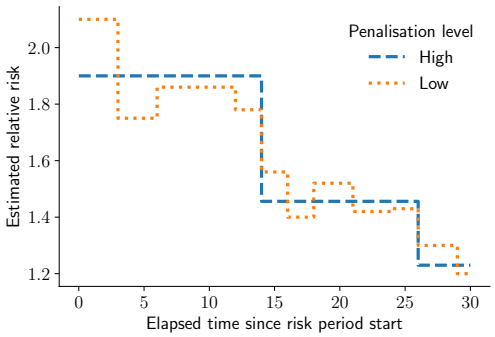


FIG. A.2.2 – Illustration de l'effet de la régularisation variation totale. En supposant une période de risque allant de 0 à 30 périodes, ConvSCCS estime une courbe de risque avec au plus 30 discontinuités. Le niveau de régularisation contrôle la taille totale des sauts de la courbe de risque. Un fort (resp. faible) niveau de pénalisation conduit à une courbe de risque plus (resp. moins) restreinte dans ses variations, comme illustré par la courbe en tirets oranges (resp. longs tirets bleus). Le but de la sélection de modèles est d'atteindre un bon équilibre entre le niveau de détail de la courbe de risque et sa régularité.

Estimation. La log-vraisemblance négative pénalisée s'écrit donc

$$-\ell(\phi, \theta) + \text{pen}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log\left(\frac{\lambda_{ik}(\phi, \theta)}{\sum_{k'=1}^{K} \lambda_{ik}(\phi, \theta)}\right) + \text{pen}(\theta), \tag{A.5}$$

où pen est donnée par l'équation (7) et λ_{ik} par l'équation (6). Cet objectif est convexe, mais la régularisation pen(θ) n'est pas différentiable. On résout ce problème de minimisation en utilisant un algorithme proximal du premier ordre, à savoir l'algorithme SVRG introduit dans [XZ14]. Les hyperparamètres γ_{tv} et γ_{gl} sont sélectionnés par validation croisée en utilisant la log-vraisemblance négative comme métrique.

Interprétatilité. ConvSCCS estime des courbes de risque relatifs $\exp(\theta^j)$ de longueur p pour chaque variable longitudinale $j = 1, \dots, d$. Si ces variables expriment des dates de début d'exposition, elles peuvent facilement être interprétées comme le risque relatif $k = 0, \dots, p$ périodes après le début des périodes d'exposition. Des intervalles de confiance peuvent être estimés par bootstrap paramétrique, comme expliqué dans le Chapitre II.

Performance sur des données synthétiques. ConvSCCS a été comparé à l'état de l'art des modèles SCCS flexibles, nommément SmoothSCCS [GWF16] et NonparaSCCS [GWF17]. À l'aide de données simulées, on montre dans le Chapitre II

que ConvSCCS obtient de meilleures performances lorsque le nombre de variables explicatives s'accroit, pour un temps de calcul comparable ou moindre.

A.2.4 Applications

ConvSCCS a d'abord été appliqué à la détection d'une association connue entre une molécule antidiabétique et le cancer de la vessie [Neu+12] (voir Chapitre II). Par la suite, ce modèle a été appliqué à la détection d'association entre l'utilisation d'anxiolytiques, d'hypnotiques, d'antidépresseurs, de neuroleptiques (AHANs) et les fractures causées par des chutes chez les personnes âgées (c.f. Chapitre III).

Antidiabétiques et cancer de la vessie

Cette étude cherche à reproduire les résultats obtenus dans [Neu+12] avec une cohorte similaire. En utilisant moins d'hypothèses, des intervalles de trente jours et des périodes de risque de $K = 24$ intervalles après le début des expositions, ConvSCCS estime les courbes présentées dans la Figure A.2.3.

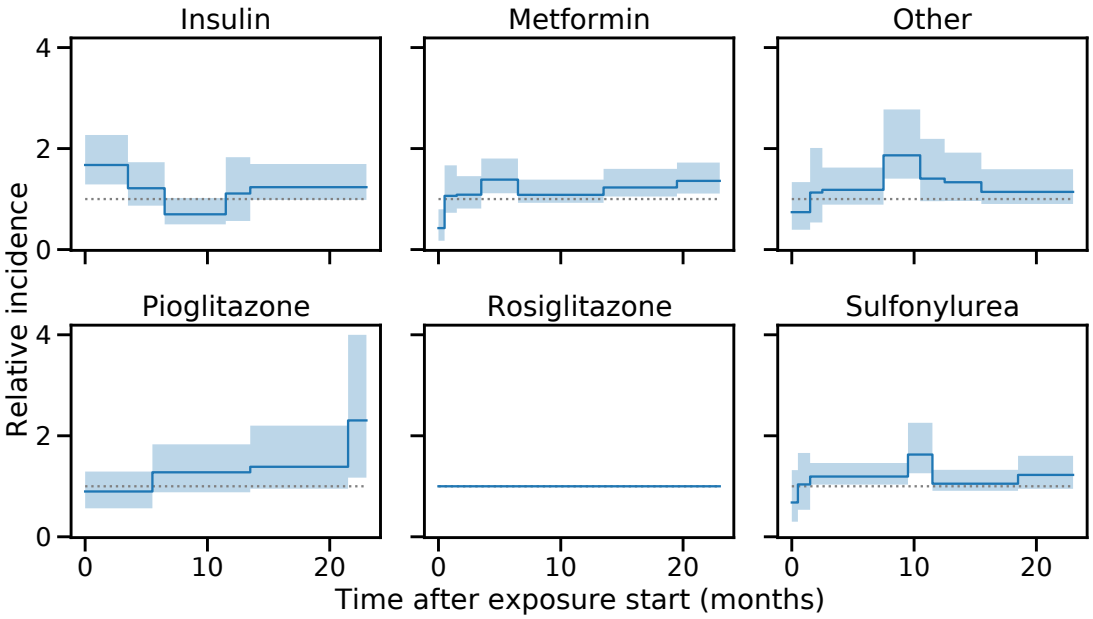


FIG. A.2.3 – Courbes de risques relatifs des antidiabétiques pour le cancer de la vessie. Les courbes bleu foncé représentent les courbes de risques relatifs estimées $k = 0, \dots, 23$ mois après le début d'une exposition. Les bandes bleu clair représentent les intervalles de confiance à 95 % estimés par bootstrap paramétrique, avec 200 tirages.

Comme illustré dans la Figure A.2.3, ConvSCCS détecte l'association forte entre le pioglitazone et le cancer de la vessie. La courbe de risque relative associée augmente avec le temps, avec un risque relatif supérieur à un de 6 à 24 mois après le début de l'exposition. Les valeurs ainsi que les points de rupture de cette courbe correspondent aux résultats obtenus dans [Neu+12] (c.f. Chapitre II). Les résultats concernant les autres antidiabétiques ne sont pas directement comparables, car [Neu+12] n'estime pas l'évolution des risques associés avec le temps. L'analyse conduite dans [Neu+12] ne détecte pas de risques significatifs pour ces molécules. ConvSCCS ne détecte pas d'effet significatif pour le rosiglitazone. Le risque associé aux autres molécules n'est pas significatif pour la plupart des périodes après le début d'exposition. En revanche, les courbes associées aux sulfonylurées et à la catégorie "autres" présentent un risque relatif positif des périodes 9 à 11, de même que l'insuline pour les périodes 0 à 5 après début d'exposition. La forme de ces trois courbes suggère la présence éventuelle de colinéarité des variables d'exposition associées. En effet, l'ordre de grandeur de ces courbes d'incidence est soit quasiment identique, soit opposé pour des intervalles donnés. Tandis que ces résultats ne sont pas parfaitement les mêmes que ceux de [Neu+12], ils montrent que ConvSCCS permet de détecter un EIM même lorsque les conditions d'application ne sont pas idéales. En effet, dans cette application, les dates de début d'exposition sont soumises à un aléa, certaines variables pouvant avoir un effet endogène quand d'autres semblent colinéaires (c.f. Chapitre II pour plus de détails). Notre méthode permet d'étudier un grand nombre de molécules simultanément, puisqu'elle peut être utilisée sans nécessiter de travail complexe de préparation des données pour formuler les expositions.

Contribution : Recherche d'associations entre les AHANs et le risque de fracture chez les personnes âgées.

Les associations entre certains AHANs et le risque de fracture chez les personnes âgées ont déjà été étudiées à différents niveaux de détails par plusieurs études cliniques et observationnelles. Les fractures chez les personnes âgées sont associées à une augmentation du risque de mortalité et sont à ce titre un sujet majeur en santé publique [Dea+10 ; Vri+18]. Ces fractures peuvent être causées par une perte de densité osseuse ou une instabilité posturale [All+05], qui peuvent toutes deux résulter d'une exposition à certains AHANs. Plusieurs méta-analyses, telles que [Sep+18a] ou [Woo+09] ont souligné la difficulté d'établir une cartographie précise des associations entre le risque de fractures et ces molécules. En effet, la plupart des études existantes se limitent à quelques molécules limitant ainsi les comparaisons de risque. Afin d'améliorer l'état des connaissances sur le sujet, [Sep+18a] suggère l'étude longitudinale de ces groupes pharmacologiques à l'échelle des molécules, ce qui est l'objectif de notre analyse.

Le design de cette étude utilise une cohorte de nouveaux utilisateurs[5] de plus de 65 ans. Les détails sur la construction de cette cohorte sont précisés dans le Chapitre III.

Deux périodes de risque ont été associées aux expositions aux AHANs. Une période de préexposition permet de détecter un risque lié au contexte de l'exposition, et une période post-exposition permet de quantifier l'évolution du risque après le début du traitement (c.f. Figure A.2.4). L'utilisation de périodes de préexposition n'est pas nouvelle et est souvent combinée à l'utilisation de modèles flexibles [NN19 ; Pra+11 ; Req+20]. L'identification des évènements de fracture a été réalisée selon la méthode présentée dans [Bou+20]. Le Chapitre III fournit des détails additionnels sur le calcul des expositions et des fractures.

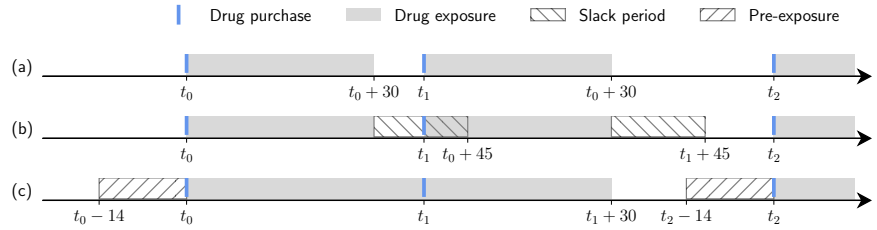


FIG. A.2.4 – Calcul des expositions aux AHANs. Les expositions sont supposées avoir une durée de 30 jours (90 jours pour les délivrances en gros conditionnement) après la date de délivrance du médicament (i). Une période tampon est ajoutée (ii) à chaque exposition pour tenir compte d'irrégularités dans les dates d'achats des traitements au long cours. Les expositions qui se superposent à d'autres expositions ou aux périodes tampons d'autres expositions sont assimilées à l'exposition initiale (iii). Enfin, une période de préexposition de 14 jours est ajoutée avant la date de début d'exposition (iii).

Interprétation des courbes de risques relatifs. Les courbes de risques relatifs (CRRs) des préexpositions sont utiles pour la détection de biais spécifiques aux parcours de soins des patients. Une CRR > 1 suggère la présence d'un biais par indication, c.-à-d. lorsque la prescription de la molécule est déclenchée par une indication pré-existante, liée à la fracture. À l'inverse, une CRR < 1 peut indiquer que le patient se trouve dans un environnement protecteur. Par exemple, un patient déjà hospitalisé a probablement moins de risque de subir une fracture qu'un patient non-hospitalisé. Ces deux effets peuvent être simultanés. Ce phénomène est discuté plus amplement dans le Chapitre III. Ainsi, bien que les CRR de préexposition ne suppriment pas les biais liés à des parcours spécifiques, elles permettent de mettre

[5] Les nouveaux utilisateurs sont les patients non exposés aux AHANs au moment du début de l'étude.

en perspective le reste des résultats obtenus. Cette information additionnelle peut notamment se révéler utile lors de la production d'études confirmatoires.

Résultats. L'estimation de risques dynamiques a produit un grand nombre de CRRs, fournissant des résultats plus informatifs que des estimations ponctuelles ou non longtidunales. Cette approche favorise notamment l'interprétation humaine de résultats qui condensent un large volume de données, en lieu et place d'un système d'alerte complètement automatisé.

De manière générale, les résultats obtenus sont cohérents avec les méta-analyses (Chapitre III). On ne citera ici que les résultats sur les antidépresseurs (Figure A.2.5) à titre d'exemple.

Les CRRs des antidépresseurs sont cohérentes avec les résultats présentés dans les revues de littérature [Sep+18a ; Ves09]. En effet, l'augmentation du risque relatif après une exposition aux tricycliques est plus faible que celle qui est observée dans le cas des inhibiteurs sélectifs de la recapture de la sérotonine, des inhibiteurs de la recapture de la sérotonine et de la noradrénaline, et des tétracycliques. On observe également des CRRs décroissantes pour le citalopram, l'escitalopram, la sertraline, la miansérine, la mirtazapine et la venlaflaxine, ce qui est cohérent avec [Hub+03]. La CRR associée à une préexposition à l'amitriptyline est supérieure à 1, ce qui suggère un biais par indication. Ce biais peut notamment résulter de l'utilisation de cette molécule dans le cas de douleurs neuropathiques, notamment après une atteinte de la mœlle épinière [AJ17]. Les CRRs de préexposition des autres molécules sont soit non-significatives, soit inférieures à 1. Dans ce dernier cas, ce résultat suggère d'éventuelles prescriptions en sortie d'hôpital. Cette observation est cohérente avec l'utilisation des inhibiteurs sélectifs de la recapture de la sérotonine [Mor+13] et de la mirtazapine [Hon+07] après un infarctus du myocarde.

A.2.5 Discussion

Cette approche mêlant l'utilisation d'un algorithme flexible à grande échelle et une construction de cohorte méticuleuse permet de produire rapidement des résultats riches en information, qui révèlent des associations tout en précisant leur contexte. Ce modèle ne nécessite pas un grand travail d'ajustement, peut analyser de grandes populations et de nombreuses molécules et fournit des résultats facilement interprétables. La construction de cohorte et la définition des expositions permettent de contrôler certains des biais posés par l'utilisation de données SNDS sans pour autant injecter d'hypothèses trop restrictives. L'étude des CRRs fournit de nombreuses informations sur les parcours de soins, permettant une identification rapide de la présence de certains biais. Bien que notre approche soit robuste à certains types de biais et permette la détection d'autres biais, les résultats doivent néanmoins être

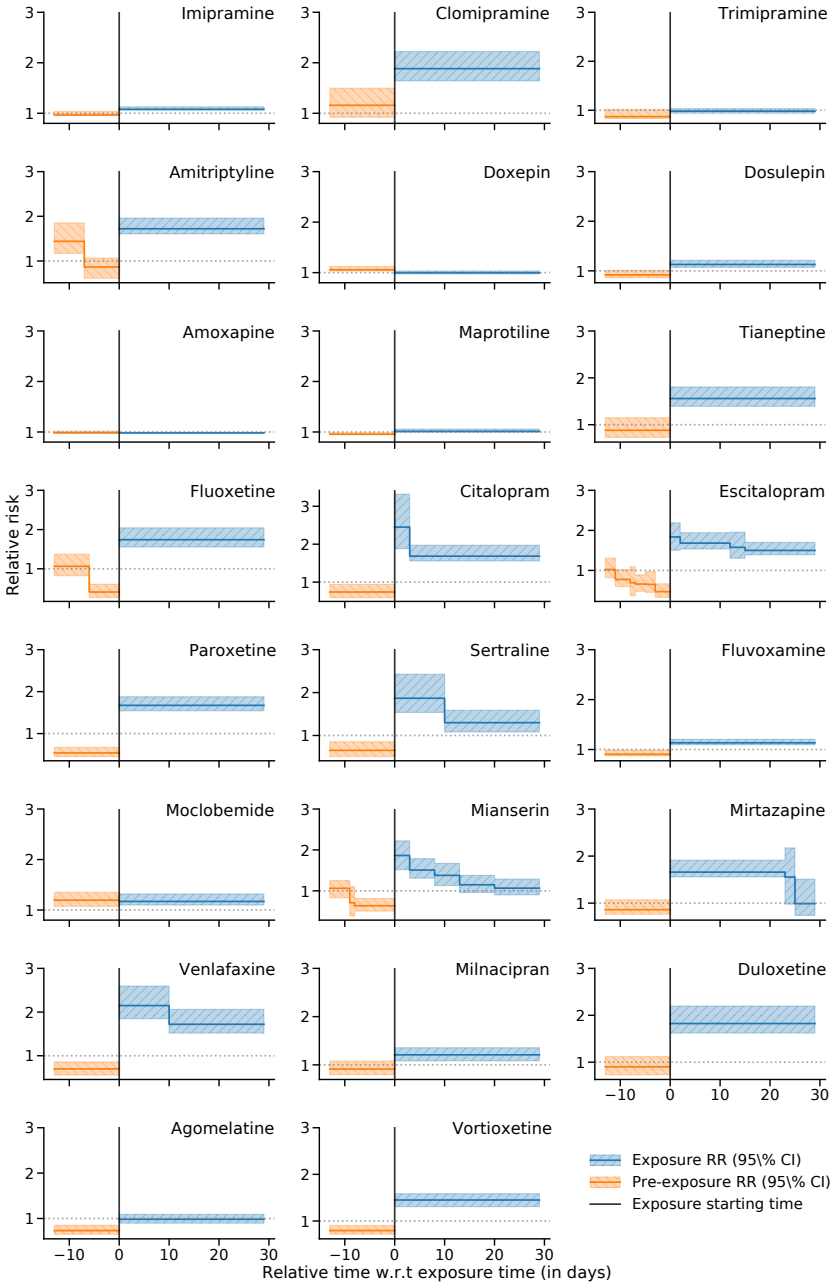


FIG. A.2.5 – Courbes de risques relatifs (CRRs) avant et après une exposition à un antidépresseur. Le temps de début d'exposition est représenté par la barre noire verticale en $x = 0$. Les lignes bleues (resp. orange) représentent les CRRs après (resp. avant) exposition, entourées par des bandes qui représentent des intervalles de confiances à 95 %. Les trois premières lignes de cette figure représentent les CRRs des antidépresseurs tricycliques. Les lignes 4 et 5 représentent les inhibiteurs sélectifs de la recapture de la sérotonine, suivis des inhibiteurs de la recapture de la sérotonine et de la noradrénaline en ligne 7.

interprétés avec précaution. Cette interprétation suppose notamment la coopération entre des experts médicaux et des statisticiens. Une analyse de sensibilité, telle que celle présentée en annexes du Chapitre III, permet notamment de susciter des réflexions intéressantes tant sur les EIMs eux-mêmes que sur leur expression dans les BDOMs. Cette approche permettant de détecter des risques d'EIMs sur des jeux de données de grande taille tout en les contextualisant semble à même de générer des alertes et de faciliter d'éventuelles études confirmatoires.

A.3 Apprentissage de représentations en santé

En santé, les labels sont souvent rares (p. ex. maladies rares, c.f. [MH20]) ou coûteux [Shi+18] à obtenir. Même en utilisant de grandes bases de données telles que le SNDS, la population pertinente pour conduire une analyse peut être très petite. Par exemple, la cohorte de 1,4 million de patients diabétiques utilisée plus haut ne contient que 1699 cas de cancers de la vessie (c.f. Table II.E.1).

La taille des échantillons peut compromettre la qualité des estimations, d'autant plus lorsque la taille des modèles augmente [RJ+91]. L'apprentissage multitâche peut permettre de contourner ce problème lorsque plusieurs tâches peuvent être accomplies à partir d'une représentation partagée. En effet, l'apprentissage de plusieurs tâches permet d'augmenter la quantité de données labellisées disponibles. De plus, l'apprentissage de chaque tâche fournit un point de vue légèrement différent du phénomène étudié, ce qui peut produire des représentations plus robustes [Car97]. Ainsi, un modèle multitâche peut se généraliser plus facilement qu'un modèle équivalent qui apprendrait tâche par tâche.

Toutefois, les modèles tels que ConvSCCS ne peuvent pas être adaptés à ce genre de contexte. En effet, ConvSCCS nécessite des hypothèses qui peuvent entrer en conflit selon les tâches effectuées. Par exemple, il peut être nécessaire de supposer pour une des tâches que les expositions sont finies et peuvent se répéter, alors que pour une autre tâche, elles soient infinies et débutent à la première prise de médicament. De telles hypothèses sont nécessaires pour contrôler certains biais (p. ex. erreurs de codage, bruit dans l'horodatage des évènements) et faciliter l'estimation des paramètres du modèle lorsque les échantillons sont petits. Cependant, ils limitent d'autant l'adoption d'approches telles que l'apprentissage multitâche, qui nécessite des modèles extrêmement flexibles capables d'apprendre des représentations performantes et polyvalentes.

Depuis quelques années, des modèles d'apprentissage profond tels que les réseaux de neurones récurrents (RNNs) ont permis des avancées dans l'accomplissement de certaines tâches en santé. Ces modèles peuvent tirer parti de l'apprentissage multitâche [Har+19]. Contrairement aux approches précédentes, les modèles d'apprentissage profond permettent d'exploiter des données quasiment brutes, et ne

nécessitent peu ou pas d'hypothèses concernant des concepts tels que les expositions.

Ces algorithmes reposent sur l'empilement de petits modèles différentiables (*couches*). Idéalement, chaque couche apprend progressivement à extraire des représentations de plus haut niveau à partir des représentations obtenues par la couche précédente. Bien que ces algorithmes nécessitent moins d'hypothèses et de travail de préparation des données, ils ont besoin de grands jeux de données et sont souvent difficiles à interpréter [Cha+17]. Par ailleurs, l'entraînement d'un modèle d'apprentissage profond se traduit généralement par un problème non convexe. Les méthodes d'optimisation utilisées doivent ainsi être adaptées à ce contexte et ne parviennent qu'à atteindre des optima locaux [LBH15 ; Rud16].

Par ailleurs, si flexible qu'il soit, un modèle multitâche doit être réappris dès lors qu'une nouvelle tâche apparaît. Le préentraînement non supervisé peut être une solution dans ce cas [Rad+19]. En effet, il ne nécessite pas de labellisation manuelle, et permet donc d'utiliser l'ensemble des données présentes dans les bases de données observationnelles. Le préentraînement est une forme d'apprentissage par transfert[6], qui consiste à entraîner un modèle sur une tâche prétexte, conçu pour favoriser l'apprentissage de représentations utiles. Ici, "utile" signifie que cette représentation doit pouvoir être adaptée à la résolution de nombreuses autres tâches inconnues lors du préentraînement. L'utilisation d'un modèle préentraîné sur une autre tâche se fait par exemple en ajoutant une ou plusieurs couches spécifiques à cette tâche. Grâce aux représentations issues du préentraînement, ce modèle apprendra plus vite et nécessitera moins de données labellisées qu'un modèle non-préentraîné.

Le préentraînement non supervisé a constitué une composante essentielle des progrès récents en Traitement Automatique du Langage (TAL) [Dev+18 ; Rad+19], mais aussi en analyse des séries temporelles [FDJ19] et en vision par ordinateur [Che+20 ; DZ17 ; OLV18].

Un parallèle est souvent fait entre les données de parcours de soins et les données textuelles [Aya+20 ; SRB19]. En effet, les deux peuvent être représentés comme une suite de symboles qui correspondent aux mots en TAL et aux codes administratifs médicaux en santé.

L'apprentissage auto supervisé est un type d'apprentissage non supervisé récent, qui consiste à préentraîner un modèle en utilisant une tâche prétexte qui repose sur une labellisation automatisée des données. BERT [Dev+18] (*Bidirectional Encoder Representations from Transformers*) en est un exemple célèbre, qui a donné lieu à de nombreux travaux [Dai+19 ; Lan+19 ; Liu+19 ; Yan+19]. Le travail qui suit essaie d'adapter cette approche aux EHRs.

[6] L'apprentissage par transfert vise à résoudre un problème en utilisant les connaissances acquises lors de la résolution d'un autre problème.

A.3.1 Apprentissage profond en santé.

Tandis que le texte et les parcours de patients sont une suite de codes avec un vocabulaire de grande dimension, certaines caractéristiques des données de santé ne sont pas présentes en TAL :

(i) L'ordre des symboles dans un texte est évident, alors que celui des symboles dans un parcours patient dépend de la pratique médicale qui peut différer selon les établissements. Les relations temporelles entre ces codes sont cruciales en santé, alors que seul compte l'ordre en TAL.

(ii) Comme abordé précédemment, les EHRs ne sont pas des enregistrements directs de la physiologie des patients mais plutôt une compilation de leurs interactions avec le système de soins. Cela peut introduire des boucles de rétroaction et inverser les dynamiques [HA13].

(iii) Les dépendances des évènements contenus dans les EHRs peuvent être beaucoup plus étendues que dans un texte. Par exemple, un diagnostic de diabète est un facteur de risque important qui doit être pris en compte tout au long de la vie d'un patient [Shi+18].

Formalisation. On considère les EHRs comme une suite d'évènements datés $z_i = (x_i, t_i)$, où $x_i \in \mathbb{R}^d$ sont les codes médicaux et t_i les dates. Les évènement qui forment ces séquences sont d'abord vectorisés (*embedded*) afin de réduire leur dimensionnalité. Dans ce travail, on représente les évènements par des vecteurs denses de dimension $D \ll d$ correspondant à un code spécifique. Ces vecteurs sont appris par le modèle. Les dates sont vectorisées à l'aide de plusieurs sinusoïdes dilatées [Vas+17]. La représentation vectorielle de z_i est noté e_i.

Encodage de séquence. Les tâches considérées dans cette section reposent sur l'*encodage* évènement contenu dans un EHR, c.-à-d. la construction progressive de représentations pour chacun de ces évènements. Ces représentations ont pour but de permettre l'accomplissement de plusieurs tâches par la suite. Un élément important du modèle est donc l'*encodeur* qui, à partir d'une suite d'évènements vectorisés $\mathbf{e} = [e_1, \dots, e_n]$ où $e_i \in \mathbb{R}^D$ pour $i = 1, \dots, n$, produit une suite de représentations contextualisées (ou *états cachés*) de même taille. Plusieurs architectures d'apprentissage profond décrites en introduction permettent d'accomplir cette tâche. Elles consistent en plusieurs couches $l = 1, \dots, L$, qui calculent des états cachés $\mathbf{h}^l = [h_1^l, \dots, h_n^l]$ à partir des résultats des couches précédentes, $\mathbf{h}^{l-1}$. La première couche du modèle prend en entrée la suite d'embeddings $\mathbf{h}^0 = \mathbf{e}$.

Modèles d'attention. On s'intéressera ici principalement aux modèles d'attention (définis dans le Chapitre IV). Ces modèles reposent uniquement sur des mécanismes d'attention et ont été créés pour représenter des suites de symboles tout en exploitant au maximum les capacités de calculs des cartes graphiques (GPUs). Ils ont été utilisés avec succès en TAL pour préentraîner de grands modèles performants tels que BERT [Dev+18] ou GPT [Rad+19]. Alors que d'autres types de modèles tels que les réseaux de neurones récurrents (*recurrent neural network*, RNNs) ou les réseaux de neurones convolutif (*convolutional neural network*, CNNs), les modèles d'attention ne déduisent pas l'ordre des éléments d'une suite de leur position réelle, mais grâce à une représentation vectorielle de leur index.

Contrairement aux textes, les données d'EHR sont datées. Par ailleurs, ces suites ne sont pas échantillonnées avec un pas régulier comme c'est souvent le cas en analyse des séries temporelles. La gestion de l'ordre par les modèles d'attention semble donc être un choix naturel pour pouvoir utiliser des données d'EHR sans procéder à des techniques d'imputation qui sont nécessaires avec les RNNs et les CNNs [Har+19 ; Tan+20]. L'utilisation de données imputées peut être très coûteuses en mémoire et en calcul. Cet aspect ne doit pas être négligé pour le préentraînement d'un modèle sur de grands volumes de données.

Plusieurs types de modèles d'attention sont présentés en introduction et dans le Chapitre IV. On s'intéresse notamment au *Transformer* [Vas+17] qui est un modèle d'attention à l'origine de nombreux progrès en TAL, ainsi qu'au *Linear Transformer*, qui est une adaptation moins coûteuse en calcul et en mémoire. Par ailleurs, le modèle d'attention sur des graphes (*Graph Attention Network*, ou GAT) est aussi utilisé dans ce qui suit. Les définitions de ces modèles ne seront pas rappelées ici.

A.3.2 Stratégies de préentraînement.

Masked Language Model. *Masked Language Model* (MLM) a été conçu pour apprendre des représentations en TAL [Dev+18]. Cette stratégie utilise deux tâches prétextes. La première consiste à sélectionner aléatoirement 15 % des symboles de chaque séquence. Ces symboles sont ensuite soumis aux transformations suivantes : 80 % d'entre eux sont remplacés par un symbole `[MASK]`, 10 % d'entre eux sont remplacés par un autre symbole sélectionné aléatoirement, et 10 % ne sont pas modifiés. La tâche prétexte consiste à prédire quel était le code initial (avant transformation) à partir du reste de la séquence de symboles. La seconde tâche consiste à prédire la phrase suivante étant donné la phrase actuelle. Tandis que la première tâche est adaptable aux EHRs, ce n'est pas le cas de la seconde étant donné que le concept de "phrase" ne se transpose pas facilement aux données de soins.

Triplet loss. L'entraînement avec *Triplet loss* consiste à prédire si une sous-séquence appartient ou non à une séquence plus longue. Cette fonction de perte a été utilisée avec succès pour construire des représentations à l'aide d'un modèle causal non supervisé [FDJ19]. Un algorithme d'échantillonnage extrait des sous-séquences x^{ref} (exemple de référence) et x^{pos} (exemple positif) aléatoirement à partir d'une séquence source y_i. Par ailleurs, K séquences x^{neg} (exemples négatifs) sont choisies aléatoirement à partir d'autres séries temporelles y_j où $j \neq i$. On cherche à ce que la représentation de x^{ref} soit proche de celle de x^{pos}, tandis que la représentation de x^{ref} doit être distante ce celle de x^{neg}. Cela se traduit par la minimisation de la fonction de perte suivante (*triplet loss*) :

$$\mathcal{L}_{\text{triplet}} = -\log(\sigma(f(x^{\text{ref}}, \theta)^\intercal f(x^{\text{pos}}, \theta))) - \sum_{k=1}^{K} \log(\sigma(-f(x^{\text{ref}}, \theta)^\intercal f(x_k^{\text{neg}}, \theta))),$$

où σ est une fonction sigmoïde et $f(\cdot, \theta)$ est un encodeur dont les paramètres θ sont à apprendre.

Contrastive Predictive Coding. *Contrastive Predictive Coding* (CPC), apprend des représentations en entraînant un modèle à prédire les représentations d'observations "futures" à partir de représentations "passées" [OLV18]. Cette idée repose sur la maximisation de l'information mutuelle entre les représentations encodées. Les auteurs proposent d'encoder les symboles futurs x et leur contexte c en une représentation compacte à l'aide d'un modèle non linéaire. Ces représentations sont conçues de façon à préserver au mieux l'*information mutuelle* (IM) entre x et c, définie comme

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)}.$$

En maximisant cette information mutuelle, CPC apprend les variables latentes communes à x et c. Cette stratégie d'apprentissage est plus amplement détaillée en introduction de cette thèse et dans le Chapitre IV.

Travaux similaires.

Comme évoqué précédemment, l'adaptation de l'état de l'art en TAL aux données de santé administratives n'est pas triviale. Quelques travaux s'y sont toutefois essayé.

BEHRT [Li+20] développe des modèles préentraînés qui prédisent les diagnostics des visites futures. Cette approche utilise une représentation vectorielle de la position de chaque visite médicale ainsi qu'une couche spécifique à l'âge des patients pour représenter la temporalité. Cependant, BERHT n'utilise que les données de diagnostic et démographiques, ignorant ainsi d'autres informations médicales telles que les

examens ou la consommation de médicaments, ce qui limite sa réutilisation pour d'autres tâches. G-BERT [Sha+19] adapte MLM pour aligner des représentations de diagnostics et de consommation de médicaments au sein d'une même visite. Ces représentations sont utilisées pour prédire les traitements à partir des diagnostics et inversement. Toutefois, ils n'utilisent pas la temporalité, ce qui empêche la réutilisation de ces représentations pour des tâches de prévision. Med-BERT [Ras+20] adapte BERT pour préentraîner des représentations sur des séquences plus longues et des populations plus grandes que ce qui a été fait avec BEHRT et G-BERT. Ce travail introduit une tâche prétexte qui consiste à prédire si les patients ont subi une hospitalisation longue durant leur historique de soins. Cette tâche remplace la prédiction de phrase suivante utilisée dans BERT. Toutefois, Med-BERT n'utilise que les informations de diagnostic et n'exploite pas le temps, mais uniquement l'ordre des visites, ce qui limite son champ d'application. *Graph Convolutional Transformer* décrit dans [Cho+20] représente les visites comme des graphes, dont les arêtes sont estimées à l'aide d'un modèle d'attention similaire à celui défini dans [Vas+17]. Ce modèle d'attention est contraint afin de garantir un ordre entre les relations de plusieurs types d'évènements. Par exemple, les symptômes causent les diagnostics et les diagnostics causent les prescriptions. La représentation des visites est calculée à l'aide de réseaux de neurones convolutifs sur les graphes ainsi estimés. Cette approche suppose un niveau de détail qui est rarement atteint dans les données d'EHR, où des données telles que les symptômes peuvent être absentes.

A.3.3 Contribution : comparaison de modèles d'attention et de méthodes de préentraînement

Ce travail apporte de nouvelles contributions en évaluant plusieurs modèles d'attention combinés à plusieurs stratégies de préentraînement pour l'apprentissage non supervisé de représentations d'EHRs. Les modèles préentraînés sont ensuite adaptés par l'ajout d'une couche spécifique pour accomplir différentes tâches. Les expériences présentées ici ont utilisé les données librement accessibles MIMIC-III [Joh+16], qui ont servi dans de nombreuses publications [Har+19; Shi+18; Son+18]. Ces expériences ont été menées en utilisant les recommandations actuelles en matière de sélection des hyperparamètres (c.f. Section IV.2.4).

Méthodologie

En plus des informations démographiques (p. ex. âge, genre), un EHR structuré consiste, pour chaque patient, en une suite d'évènements médicaux tels que des diagnostics, des délivrances de médicaments, des mesures physiologiques ou des actes médicaux. Chaque évènement est horodaté avec une précision qui dépend de

la base de données utilisée. Selon la taille de l'unité de temps, plusieurs évènements peuvent survenir au même moment et partager le même horodatage.

Représentation chronologique et graphes. La représentation chronologique d'un parcours de patient avec plusieurs types d'évènements est illustré Figure A.3.2A. Alternativement, le même parcours peut être représenté par un graphe dirigé (Figure A.3.2C) où un ensemble de nœuds successifs représente les unités de temps durant lesquelles au moins un évènement a lieu. Un autre ensemble de nœuds représente les évènements qui surviennent durant une unité de temps données. Les arêtes du graphe correspondent aux associations structurelles entre les évènements, telles que "prochaine unité de temps" ou "évènement médical associé à une unité de temps". Les unités de temps sans évènements médicaux associés ne sont pas encodées.

Une représentation similaire a été décrite dans [Cho+18], et a inspiré d'autres travaux tels que [Cho+20] et [Het+19]. La représentation utilisée dans ce travail est similaire à la formulation de [Het+19]. En revanche, on modélise les suites d'évènements à l'aide de modèles d'attention, tandis que [Het+19] utilise des processus ponctuels. Le choix de cette représentation est guidé par le modèle d'attention sur graphes décrit ci-dessous.

Architecture des modèles. Tous les modèles considérés dans cette section partagent la même architecture globale, illustrée Figure 14. Cette architecture est basée sur quatre éléments : un embedding, un encodeur, un *pooler* et une couche dense qui dépend de la tâche à accomplir. L'entraînement du modèle a lieu en deux étapes : l'encodeur est d'abord préentraîné de façon non supervisée en utilisant une tâche prétexte, puis le modèle est ajusté à une tâche clinique de façon supervisée après l'ajout d'une couche dense.

Embedding. Chaque évènement de l'EHR (c.f. Figure A.3.2A) correspond à un code administratif et/ou des valeurs numériques. Les valeurs numériques sont d'abord discrétisées en utilisant les déciles calculés sur l'échantillon d'entraînement. Chaque modalité est ensuite représentée individuellement à l'aide d'un vecteur de faible dimension (*embedding*) entraîné par le modèle. Les positions relatives de chaque évènement par rapport au premier évènement sont encodées à l'aide d'une représentation vectorielle fixe, basée sur un ensemble de sinusoïdes dilatées [Vas+17]. Ces positions sont calculées comme la différence entre la date et l'heure de l'évènement considéré et la date et l'heure du premier évènement de la séquence. Les représentations vectorielles de la modalité et de sa date sont ensuite sommées pour produire la représentation de l'évènement. Enfin, la représentation vectorielle du symbole `[CLS]` est jointe à la fin de la séquence.

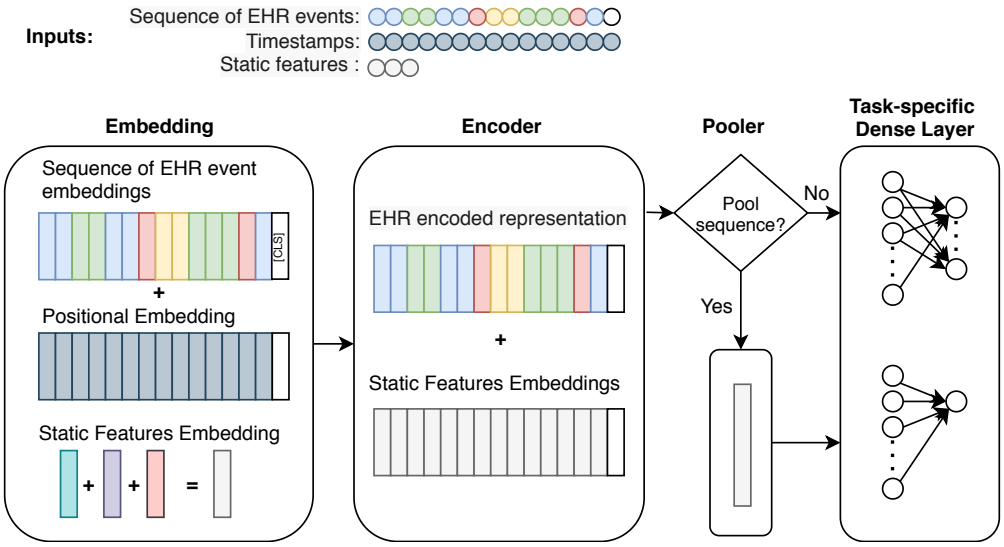


FIG. A.3.1 – Architecture globale du modèle. La représentation de l'EHR (c.f. Figure A.3.2) est utilisée en entrée du modèle. Une couche d'*embedding* construit d'abord la représentation des symboles et des dates associées aux évènements. Un mécanisme d'attention encode ensuite les évènements en fonction de leur contexte. Puis, un *pooler* agrège les représentations obtenues si la tâche nécessite une représentation globale de la séquence d'évènements. Enfin, une couche dense sert à accomplir les prédictions qui correspondent à une tâche donnée.

De plus, les variables non longitudinales (telles le genre et l'âge) sont également vectorisées et sommées en un seul vecteur. Ce vecteur est utilisé en entrée de l'encodeur, c.f. Figure A.3.1.

Encodeur. Un encodeur sert à encoder l'ensemble des évènements de l'EHR de façon à produire une nouvelle séquence de même taille, dont les éléments sont représentés en fonction du contexte sélectionné par le mécanisme d'attention. Les variables statiques vectorisées sont ensuite ajoutées à chaque élément de cette nouvelle séquence. Ce travail compare différents encodeurs basés sur des mécanismes d'attention :

(i) Le *Transformer* [Vas+17], qui permet de construire la représentation des éléments d'une séquence grâce à une succession de couches de *self-attention* multi-têtes (*multi-head self-attention* – MSA).

(ii) Le *Transformer* linéaire [Wu+20], qui utilise une quantité de mémoire linéaire ($\mathcal{O}(n)$) en la longueur de la séquence (n). Cela permet de traiter des séquences beaucoup plus longues que *Transformer* classique, dont les besoins en mémoire sont quadratiques ($\mathcal{O}(n^2)$).

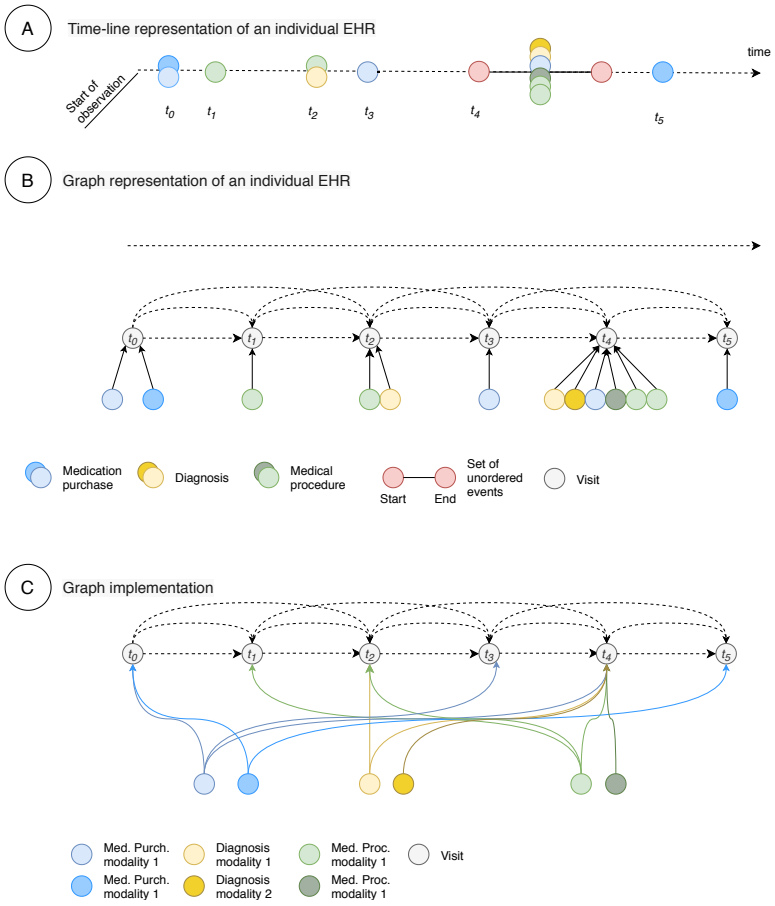


FIG. A.3.2 – Représentation des EHRs. Un EHR correspond à une séquence horodatée d'évènements médicaux **(A)**. Cette séquence peut être représentée par le graphe **(B)**. Un nœud *visite* est créé à chaque unité de temps dans laquelle au moins un évènement médical survient. Des nœuds *évènements* sont créés à chaque fois qu'ils surviennent durant une visite. Les nœuds visite sont initialisées en sommant l'*embedding* du symbole `[visit]` et l'*embedding* de la date de visite. Les nœuds évènements sont initialisés avec l'*embedding* des modalités correspondantes. En pratique, le graphe est implémenté suivant la représentation **(C)** afin d'économiser la mémoire. Pour une séquence donnée, les nœuds évènements ne sont créés qu'une fois par modalité observée (nœuds *évènement-modalité*). Chaque nœud visite dans lequel un évènement s'est produit est lié au nœud évènement-modalité correspondant. Comme aucune arête n'est dirigée vers les nœuds évènement-modalité, leur représentation n'est pas mise à jour par les couches du modèle. Ainsi, cette représentation maintient la causalité temporelle et ne donne pas lieu à une fuite d'information.

(iii) Le *Graph Attention Network* (GAT) [Vel+17 ; Ye+19], dont le mécanisme d'attention n'utilise pas le mécanisme de clé et de requête (*key-query*). Cet encodeur utilise la représentation de graphe décrite dans la Figure A.3.2 C.

Ces encodeurs sont décrits en détail dans les annexes du Chapitre IV.

Pour tous ces encodeurs, les attentions sont contraintes de façon à respecter une causalité temporelle : les représentations d'un évènement ne peuvent être mises à jour qu'à partir des représentations de cet évènement et des évènements antérieurs. Cette contrainte permet d'éviter les fuites de données des évènements futurs vers un évènement présent.

Pooler. Comme expliqué dans la Section A.3.4 ci-dessous, deux types de tâches supervisées sont considérées : (i) les tâches qui nécessitent une prédiction par élément de la séquence encodée (p. ex. prédiction longitudinale du temps d'hospitalisation restant) et (ii) les tâches qui prédisent un label pour l'ensemble de la séquence (p. ex. prédiction du décès éventuel durant le séjour hospitalier). L'utilisation d'un *pooler* n'est nécessaire que pour les tâches (ii). De façon similaire à [Dev+18], le *pooler* utilise la représentation du symbole `[CLS]` pour représenter la séquence.

Préentraînement non supervisé

Comme résumé dans la Figure 15, les stratégies suivantes de préentraînement non supervisé (décrites dans la Section A.3.2) ont été utilisées :

(i) *Masked Language Modeling* (MLM) [Dev+18],

(ii) *Triplet Loss* [FDJ19],

(iii) *Contrastive Predictive Coding* (CPC or InfoNCE) [Che+20 ; Sun+19].

Toutes les architectures ont été entraînées indépendamment pour chacune des stratégies de préentraînement.

Ajustement supervisé, fonctions de perte et métriques

Toutes les combinaisons d'encodeurs et de stratégies de préentraînement ont été évaluées sur leur capacité à accomplir plusieurs tâches cliniques *supervisées* formalisées par [Har+19] et décrites dans la Section A.3.4 ci-dessous. Pour chaque combinaison de modèle et de tâche supervisée, les stratégies d'ajustement décrites ci-dessous ont été comparées :

(i) Ajustement de l'ensemble de l'architecture en utilisant la tâche supervisée, y compris l'*embedding* et l'encodeur qui sont alors initialisés avec les poids appris durant le préentraînement.

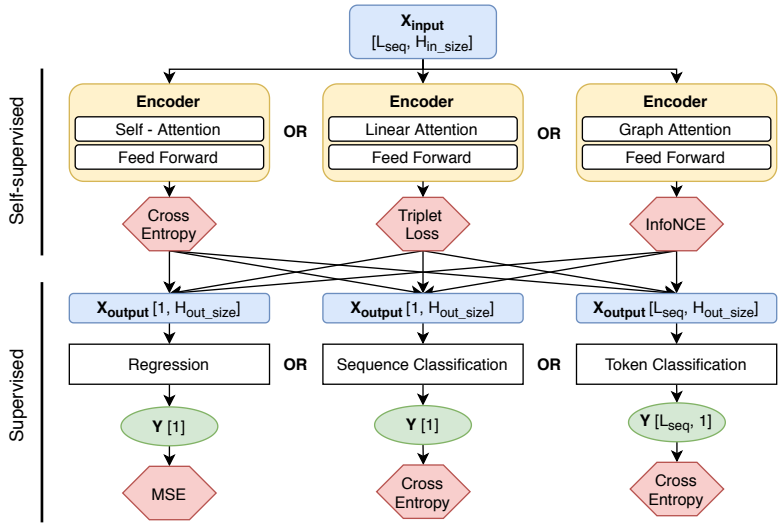


FIG. A.3.3 – Stratégies de préentraînement et procédures d'évaluation. Les trois encodeurs sont d'abord préentraînés indépendamment suivant les trois stratégies de préentraînement. Une couche dense correspondant à la tâche à accomplir est ensuite ajoutée au modèle, ainsi qu'un *pooler* si nécessaire. Ces deux éléments utilisent les représentations produites par l'encodeur pour effectuer les prédictions nécessaires à la tâche. La couche dense peut être entraînée indépendamment de l'encodeur. Il est également possible d'entraîner de façon supervisée l'ensemble du modèle (*embedding* et encodeur compris) pour améliorer sa spécialisation sur une tâche donnée. Cet entraînement ne nécessite en général que quelques itérations sur le jeu de données labellisé.

(ii) Ajustement de la couche dense et du *pooler* uniquement. Les poids de l'encodeur et de l'*embedding* sont fixés et correspondent aux valeurs apprises durant le préentraînement.

(iii) Apprentissage intégralement supervisé, sans préentraînement de l'ensemble de l'architecture, avec initialisation aléatoire des poids.

Selon les tâches supervisées, les fonctions de pertes et les métriques de performance suivantes ont été utilisées : l'entropie croisée a été utilisée pour les problèmes de classification binaire et multiclasse. Les métriques AUROC et AUPRC ont été utilisées pour évaluer les tâches de classification binaire, tandis que le Kappa de Cohen a été utilisé pour la classification multiclasse. Des précisions concernant la sélection des hyperparamètres et l'entraînement des différentes architectures sont données dans le Chapitre IV.

A.3.4 Expériences

L'ensemble des expériences a été accompli à l'aide de la base de données MIMIC-III, qui contient des résumés de séjour en soins intensifs pseudonymisés, collectés ente 2001 et 2012 [Joh+16]. De façon similaire à [Har+19 ; Son+18], les expériences se basent sur une cohorte de 33798 patients, pour un total de 42276 séjours en soins intensifs. La sélection de population, des variables et des labels se base sur le code publié par [Har+19] (c.f. Table IV.3.1 dans le Chapitre IV pour plus de détails). Les ensembles d'entraînement, de validation et de test comptent respectivement 70 %, 15 % et 15 % des séjours en soins intensifs. Le tirage de ces échantillons correspond à ceux utilisés par [Har+19]. Les séjours en soins intensifs comprenant moins de cinq évènements ont été exclus.

Trois tâches de prédiction clinique tirées de [Har+19] et décrites ci-dessous ont été utilisées :

(i) *In-Hospital Mortality* (IHM) : le label est une variable binaire qui indique si un patient décède durant un séjour donné ou non. Cette tâche est traitée comme un problème de classification binaire. Le taux de mortalité dans la cohorte est d'environ 13 %.

(ii) *Length-of-Stay* (LOS) : le label représente la durée restante de séjour en soins intensifs. Cette durée est divisée en dix catégories (≤ 1 jour ; 1; 2; ... ; 7 jours, $[1, 2)$ semaines ; ≥ 2 semaines). Cette tâche est ainsi considérée comme un problème de classification à 10 classes. La prédiction a lieu pour chaque évènement de la séquence.

(iii) *Phenotyping* (PHE) : le label correspond à une des 25 pathologies retenues par [Har+19]. Cette tâche est traitée comme un problème de classification : la pathologie est prédite a posteriori, à partir de l'ensemble des évènements du séjour en soins intensifs. Parmi les 25 pathologies considérées, 12 concernent des insuffisances respiratoires ou cardiaques, 8 concernent des pathologies chroniques telles que le diabète ou l'athérosclérose, et 5 sont des pathologies "mixtes" telles que les infections du foie. Les séjours pouvant avoir plusieurs phénotypes sont exclus [Har+19].

Résultats

Les combinaisons de modèles et de stratégies de préentraînement décrites dans la Section A.3.3 ont été évaluées suivant leurs performances pour les tâches supervisées décrites ci-dessus. Les métriques correspondantes ont été calculées sur l'échantillon de test, et sont reportées dans la Table 1.

On remarque premièrement les performances décevantes de GAT pour chacune des tâches, avec ou sans utilisation de stratégie de préentraînement. Augmenter le

TAB. A.3.1 – Métriques de performance des différents modèles et stratégies de pré-entraînement (lignes) pour les différentes tâches de prédiction (colonnes) sur des données MIMIC-III. Ces métriques ont été calculées sur l'échantillon de test défini dans [Har+19]. En raison de ses mauvaises performances, GAT n'a pas été entraîné pour toutes les tâches et stratégies de préentraînement afin de ne pas consommer des ressources de calcul inutilement. * La tâche *Length Of Stay* (LOS) décrite [Har+19] est légèrement différente de la tâche considérée ici. Alors que [Har+19] prédit la durée de séjour restante (LOS) chaque heure, les modèles d'attention présentés ici font une prédiction pour chaque nouvel évènement médical. Cette différence s'explique par l'inutilité de l'imputation avec l'utilisation des modèles d'attention. Ainsi, les métriques de performance concernant cette tâche ne sont pas tout à fait comparables.

Encoder	*In-hospital mortality* AUC-PR/AUC-ROC	*Length of Stay* Kappa	*Phenotyping* AUC-ROC
Intégralement supervisé			
LSTM multitâche [Har+19]	**0.533/0.870**	**0.450***	**0.774**
Transformer	0.394/0.809	0.535	0.736
Transformer linéaire	0.355/0.790	**0.584**	0.676
GAT	0.132/0.528	0.218	0.503
Pré-entraînement MLM			
Transformer	**0.409/0.817**	**0.554**	**0.749**
Transformer linéaire	0.344/0.785	0.405	0.708
GAT	0.154/0.572	–	–
Pré-entraînement *Triplet Loss*			
Transformer	**0.357/0.781**	0.451	**0.729**
Transformer linéaire	0.330/0.774	**0.577**	0.686
Pré-entraînement *CPC*			
Transformer	**0.391/0.805**	0.466	**0.741**
Transformer linéaire	0.333/0.770	**0.521**	0.675

nombre de visites passées qui peuvent servir à la mise à jour des représentations de visite n'améliore pas cette performance. Comme l'agrégation d'évènements par visite en utilisant une structure de graphe a produit de bons résultats dans [Cho+20], cette mauvaise performance s'explique probablement par une formulation inadéquate du mécanisme d'attention dans ce contexte. En effet, l'attention dans GAT utilise uniquement l'information du nœud plutôt qu'un mécanisme de requête, clé, valeur utilisé dans les *Transformers*. Au-delà des métriques de performances, l'utilisation d'un graphe a été efficiente en termes de mémoire GPU, permettant de traiter des séquences d'évènements plus longues qu'avec les *Transformers*. Par ailleurs, il est plus naturel d'implémenter une attention causale sur un graphe puisque cela ne nécessite

pas de masquage *ad hoc*. Combiner l'utilisation d'un graphe avec un mécanisme d'attention plus proche de ce qui est utilisé par les *Transformers* constitue une piste de recherche intéressante, encouragée par les résultats prometteurs d'une stratégie analogue utilisée en TAL [Ye+19].

Comme expliqué précédemment, le *Transformer* peine à traiter des séquences longues en raison de sa complexité quadratique en la longueur des séquences. Dans nos expériences, limiter la longueur des séquences pour satisfaire les contraintes de mémoire des GPUs diminue la performance de ce modèle. L'utilisation du *Transformer* linéaire a permis d'utiliser des séquences plus longues, sans que cela se traduise par une augmentation des performances par rapport au *Transformer* classique.

Lors de l'ajustement des modèles préentraînés aux tâches, l'ajustement consistant à entraîner uniquement la couche dense additionnelle en laissant l'encodeur inchangé a produit de moins bons résultats (non reportés) que l'ajustement du modèle entier. L'ajustement du modèle entier initialisé avec des poids préentraînés n'a nécessité qu'entre 5 et 15 itérations sur les données d'entraînement pour atteindre de bonnes performances, selon la tâche considérée. Le temps d'entraînement de chacun des modèles et des stratégies de préentraînement a duré moins de cinq heures, à l'exception de MLM dont l'entraînement a parfois duré jusqu'à deux jours. L'utilisation du préentraînement MLM améliore les scores du *Transformer* par rapport aux scores obtenus par entraînement intégralement supervisé. En revanche, le préentraînement par *triplet loss* et *CPC* n'a produit que des gains de performance marginaux. S'agissant de la *triplet loss*, le tirage des triplets $x^{\text{ref}}, x^{\text{pos}}, x^{\text{neg}}$ peut être la cause de ce constat. En effet, même un modèle très simple peut apprendre rapidement à distinguer x^{pos} de x^{neg} si ces sous-séquences sont choisies totalement aléatoirement. Dans ce cas, la *triplet loss* converge rapidement vers zéro, ralentissant grandement la mise à jour des paramètres [Wu+17]. Adapter la stratégie d'échantillonage pourrait certainement permettre d'améliorer les résultats obtenus par cette méthode.

Le préentraînement contrastif (CPC) n'a probablement pas révélé toutes ses capacités. Des travaux récents en vision par ordinateur ont montré que l'utilisation de techniques d'augmentation des données (*data augmentation*) est cruciale dans la mise en œuvre de ce type de préentraînement [Che+20]. Le développement de techniques d'augmentation des données de type EHR reste à notre connaissance un problème de recherche ouvert qui pourrait déboucher sur des améliorations majeures du préentraînement non-supervisé en santé.

Ce travail a été conçu de façon à pouvoir être mis en œuvre avec des données observationnelles du type SNDS au lieu des EHRs. Tandis que plusieurs publications utilisent des modèles d'apprentissage profond sur des EHRs, à notre connaissance seul [Kab+19] utilise ce type de modèle sur des données de santé administratives. Les BDOMs telles que le SNDS, de par leur taille et leur exhaustivité, pourraient grandement tirer partie des stratégies de préentraînement non-supervisées. La pour-

suite de ce travail sur ce type de données pourrait éventuellement déboucher sur des innovations significatives en santé publique.

LIST OF FIGURES

1 Simplified structure of the SNDS database. 6
2 SCALPEL3 workflow. 9
3 `Extractor` design. 11
4 SCALPEL-Extraction scaling experiments. 13
5 Risk patterns of adverse drug reactions. 15
6 Illustration of Total Variation penalization effect. 31
7 Performance comparison of SCCS models on synthetic data. 33
8 Run times comparison of SCCS models. 33
9 Relative incidence curves of glucose lowering drugs on the risk of bladder cancer. 35
10 Drug exposures computation. 37
11 Fracture relative risk curves estimated before and after antidepressant exposure. 39
12 Three families of deep learning models for sequence representation. 43
13 Graph representation of EHRs. 46
14 Model architecture overview 51
15 Unsupervised pre-training and evaluation procedures. 53

I.1 SCALPEL3 workflow. 63
I.2 `Extractor` design. 65
I.3 SCALPEL-Extraction scaling on several tasks. 70

II.4.1 Performance comparison of SCCS models on synthetic data (set 1). 94
II.4.2 Performance comparison of SCCS models on synthetic data (set 2). 95
II.4.3 Run times comparison of SCCS models. 96
II.4.4 Estimated relative incidences of glucose lowering drugs on the risk of bladder cancer. 99
II.E.1 Example of Hawkes adjacency matrix used to simulate exposures. 104
II.E.2 Set 1 of synthetic relative risk profiles. 106
II.E.3 Set 2 of synthetic relative risk profiles. 106
II.E.4 Sensitivity analysis adding a uniform noise. 108
II.E.5 Sensitivity analysis simulating not-at-random missing data. 108
II.E.6 Sensitivity analysis simulating missing few longitudinal features. . 109
II.E.7 Sensitivity analysis simulating missing many longitudinal features. 109

III.2.1 Illustration of the drug exposures computation. 115
III.2.2 Illustration of Total Variation penalisation effect. 117
III.3.1 Flow chart of the study population. 119
III.5.1 Estimated relative risk of fracture w.r.t. anxiolytics exposure. . . . 131
III.5.2 Estimated relative risk of fracture w.r.t. hypnotics exposure. 132
III.5.3 Estimated relative risk of fracture w.r.t. antidepressants exposure. . 133
III.5.4 Estimated relative risk of fracture w.r.t. neuroleptics exposure. . . 134
III.5.5 Estimated relative risk of hip fracture w.r.t. anxiolytics exposure. . 135
III.5.6 Estimated relative risk of hip fracture w.r.t. hypnotics exposure. . . 136
III.5.7 Estimated relative risk of hip fracture w.r.t. antidepressants exposure. 137
III.5.8 Estimated relative risk of hip fracture w.r.t. neuroleptics exposure. 138
III.5.9 Sensitivity analysis experiments summary. 139
III.B.1 Anxiolytics – Unique fracture. 142
III.B.2 Hypnotics – Unique fracture. 143
III.B.3 Antidepressants – Unique fracture. 144
III.B.4 Neuroleptics – Unique fracture. 145
III.B.5 Anxiolytics – Patients under 85 years old. 146
III.B.6 Hypnotics – Patients under 85 years old. 147
III.B.7 Antidepressants – Patients under 85 years old. 148
III.B.8 Neuroleptics – Patients under 85 years old. 149
III.B.9 Anxiolytics – Non-epileptic patients. 150
III.B.10 Hypnotics – Non-epileptic patients. 151
III.B.11 Antidepressants – Non-epileptic patients. 152
III.B.12 Neuroleptics – Non-epileptic patients. 153
III.B.13 Anxiolytics – Restriction to women. 154
III.B.14 Hypnotics – Restriction to women. 155
III.B.15 Antidepressants – Restriction to women. 156
III.B.16 Neuroleptics – Restriction to women. 157
III.B.17 Anxiolytics – Restriction to men. 158
III.B.18 Hypnotics – Restriction to men. 159
III.B.19 Antidepressants – Restriction to men. 160
III.B.20 Neuroleptics – Restriction to men. 161
III.B.21 Anxiolytics – With control variables. 162
III.B.22 Hypnotics – With control variables. 163
III.B.23 Control drugs – With control variables. 163
III.B.24 Antidepressants – With control variables. 164
III.B.25 Neuroleptics – With control variables. 165
III.B.26 Anxiolytics – Non-hospitalised fractures. 166
III.B.27 Hypnotics – Non-hospitalised fractures. 167
III.B.28 Antidepressants – Non-hospitalised fractures. 168

III.B.29 Neuroleptics – Non-hospitalised fractures. 169
III.B.30 Anxiolytics – Fractures without surgery. 170
III.B.31 Hypnotics – Fractures without surgery. 171
III.B.32 Antidepressants – Fractures without surgery. 172
III.B.33 Neuroleptics – Fractures without surgery. 173
III.B.34 Anxiolytics – Fractures with surgery. 174
III.B.35 Hypnotics – Fractures with surgery. 175
III.B.36 Antidepressants – Fractures with surgery. 176
III.B.37 Neuroleptics – Fractures with surgery. 177
III.B.38 Anxiolytics – Wrist fracture. 178
III.B.39 Hypnotics – Wrist fracture. 179
III.B.40 Antidepressants – Wrist fracture. 180
III.B.41 Neuroleptics – Wrist fracture. 181
III.B.42 Anxiolytics – Spine fracture. 182
III.B.43 Hypnotics – Spine fracture. 183
III.B.44 Antidepressants – Spine fracture. 184
III.B.45 Neuroleptics – Spine fracture. 185
III.C.1 Assessment of event dates and observation dates independance. . 186

IV.2.1 Graph representation of EHRs. 191
IV.2.2 Model architecture. 192
IV.2.3 Unsupervised pre-training and evaluation procedures. 193

A.1.1 Structure simplifiée du SNDS. 210
A.1.2 Architecture de SCALPEL3. 212
A.2.1 Exemples de risques associés aux effets secondaires médicamenteux. 214
A.2.2 Illustration de l'effet de la régularisation variation totale. 220
A.2.3 Courbes de risques relatifs des antidiabétiques pour le cancer de la vessie. 221
A.2.4 Calcul des expositions aux AHANs. 223
A.2.5 Courbes de risques relatifs avant et après une exposition à un antidépresseur. 225
A.3.1 Architecture globale du modèle. 233
A.3.2 Représentation des EHRs. 234
A.3.3 Stratégies de préentraînement et évaluation. 236

List of Tables

1 Architecture and pre-training strategies performance comparison . 55

I.1 Characteristics of the dataset used for SCALPEL3 benchmark. . . 68
I.B.1 List of SNDS databases denormalized by SCALPEL3 79
I.C.1 List of event extractors implemented in SCALPEL3. 80
I.D.1 List of transformers implemented in SCALPEL3. 81

II.4.1 Comparison of SCCS methods with ConvSCCS. 93
II.E.1 Demographics and glucose-lowering drug use of the studies cohort of French diabetic patients. 110

III.3.1 Demographics and Anxiolytics, Hypnotics, Antidepressants or Neuroleptics users. 120
III.3.2 Demographics of fractured patients. 121
III.4.1 Comparison with relevant meta-analyses and reviews. 124
III.A.1 Anxiolytics: Anatomical Therapeutic Chemical (ATC) codes. . . . 140
III.A.2 Hypnotics: Anatomical Therapeutic Chemical (ATC). 140
III.A.3 Antidepressants: Anatomical Therapeutic Chemical (ATC) codes. 141
III.A.4 Neuroleptics: Anatomical Therapeutic Chemical (ATC) codes. . . 141

IV.3.1 MIMIC-III cohorts for each task. 195
IV.4.1 Architecture and pre-training strategies performance comparison . 196

A.3.1 Comparaison de la performance de différents modèles et stratégies de préentraînement 238

Bibliography

[Ach+15] Massil Achab et al. "SGD with Variance Reduction beyond Empirical Risk Minimization." In: *arXiv preprint arXiv:1510.04822* (2015).

[ACH08] David M Adamson, Stella Chang, and Leigh G Hansen. "Health research data for the real world: the MarketScan databases." In: *New York: Thompson Healthcare* (2008), b28.

[AJ17] N Agarwal and M Joshi. "Effectiveness of amitriptyline and lamotrigine in traumatic spinal cord injury-induced neuropathic pain: a randomized longitudinal comparative study." In: *Spinal Cord* 55.2 (2017), pp. 126–130.

[AGG15] M. Z. Alaya, S. Gaiffas, and A. Guilloux. "Learning the intensity of time events with change-points." In: *IEEE Transactions on Information Theory* 61.9 (2015), pp. 5148–5171.

[Alb+18] D.J. Albers et al. "Estimating summary statistics for electronic health record laboratory data for use in high-throughput phenotyping algorithms." In: *Journal of Biomedical Informatics* 78 (Feb. 2018), pp. 87–101.

[All+05] Hervé Allain et al. "Postural instability and consequent falls and hip fractures associated with use of hypnotics in the elderly." In: *Drugs & aging* 22.9 (2005), pp. 749–765.

[Alv+98] A. Alvarez-Requejo et al. "Under-reporting of adverse drug reactions: Estimate based on a spontaneous reporting scheme and a sentinel system." In: *European Journal of Clinical Pharmacology* 54.6 (1998), pp. 483–488.

[Apa15] ORC Apache. *Apache ORC: High-Performance Columnar Storage for Hadoop*. 2015.

[Arm+15] Michael Armbrust et al. "Spark SQL: Relational Data Processing in Spark." In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. SIGMOD '15. Melbourne, Victoria, Australia: ACM, 2015, pp. 1383–1394. DOI: 10.1145/2723372.2742797.

[AGT14] Ben G Armstrong, Antonio Gasparrini, and Aurelio Tobias. "Conditional Poisson models: a flexible alternative to conditional logistic case cross-over analysis." In: *BMC medical research methodology* 14.1 (2014), p. 122.

[AF03] J. K. Aronson and R. E. Ferner. "Joining the DoTS: new approach to classifying adverse drug reactions." In: *BMJ* 327.7425 (2003), pp. 1222–1225.

[Aro+18] JK Aronson et al. "Confounding by indication." In: *Catalogue of bias* (2018).

[ATI] ATIH. *Website of the Technical Hospitalization Information Agency (ATIH), http://www.atih.sante.fr*.

[Aya+20] Jose Roberto Ayala Solares et al. "Deep learning for electronic health records: A comparative review of multiple deep neural architectures." In: *Journal of Biomedical Informatics* 101 (Jan. 2020), p. 103337. DOI: 10.1016/j.jbi.2019.103337.

[Azo+12] Laurent Azoulay et al. "The use of pioglitazone and the risk of bladder cancer in people with type 2 diabetes: nested case-control study." In: *Bmj* 344 (2012), e3645.

[Bac+12] Francis Bach et al. "Optimization with sparsity-inducing penalties." In: *Foundations and Trends® in Machine Learning* 4.1 (2012), pp. 1–106.

[Bac+17a] E. Bacry et al. "tick: a Python library for statistical learning, with a particular emphasis on time-dependent modeling." In: (July 2017). arXiv: 1707.03003.

[Bac+17b] Emmanuel Bacry et al. "Tick: a Python library for statistical learning, with an emphasis on hawkes processes and time-dependent models." In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 7937–7941.

[Bac+20] Emmanuel Bacry et al. "SCALPEL3: a scalable open-source library for healthcare claims databases." In: *International Journal of Medical Informatics* (2020), p. 104203.

[BW17] Gregori Baetschmann and Rainer Winkelmann. "A dynamic hurdle model for zero-inflated count data." In: *Communications in Statistics-Theory and Methods* 46.14 (2017), pp. 7174–7187.

[Ban+18] Juan M Banda et al. "Advances in electronic phenotyping: from rule-based definitions to machine learning models." In: *Annual Review of Biomedical Data Science* 1 (2018), pp. 53–68.

[BT09] Amir Beck and Marc Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.

[Bee66] Henry K Beecher. “Ethics and clinical research.” In: *Biomedical ethics and the law*. Springer, 1966, pp. 215–227.

[Ben+15] Eric I Benchimol et al. “The REporting of studies Conducted using Observational Routinely-collected health Data (RECORD) statement.” In: *PLoS medicine* 12.10 (2015), e1001885.

[BS13] Duane Bender and Kamran Sartipi. “HL7 FHIR: An agile and RESTful approach to healthcare information exchange.” In: *Proceedings of CBMS 2013 - 26th IEEE International Symposium on Computer-Based Medical Systems*. IEEE, June 2013, pp. 326–331.

[Ber99] DP Bertsekas. “Nonlinear Programming 2nd edn (Belmont, MA: Athena Scientific).” In: (1999).

[Bez+17] Julien Bezin et al. “The national healthcare system claims databases in France, SNIIRAM and EGB: Powerful tools for pharmacoepidemiology.” In: *Pharmacoepidemiology and Drug Safety* 26.8 (Aug. 2017), pp. 954–962. DOI: 10.1002/pds.4233.

[Blo+13] Frederic Bloch et al. “Estimation of the risk factors for falls in the elderly: Can meta-analysis provide a valid answer?” In: *Geriatrics & Gerontology International* 13.2 (Apr. 2013), pp. 250–263.

[Bon+17] Stephen Bonner et al. “Exploring the Evolution of Big Data Technologies.” In: *Software Architecture for Big Data and the Cloud*. Elsevier, 2017, pp. 253–283.

[Bou+20] Benjamin Bouyer et al. “Burden of fractures in France: incidence and severity by age, gender, and site in 2016.” en. In: *International Orthopaedics* 44.5 (Feb. 2020), pp. 947–955. DOI: 10.1007/s00264-020-04492-2.

[Bro+10] M Alan Brookhart et al. “Confounding control in healthcare database research: challenges and potential approaches.” In: *Medical care* 48.6 0 (2010), S114.

[Car97] Rich Caruana. “Multitask learning.” In: *Machine learning* 28.1 (1997), pp. 41–75.

[Cha+17] Supriyo Chakraborty et al. "Interpretability of deep learning models: a survey of results." In: *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE. 2017, pp. 1–6.

[Che+20] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG].

[Chi+19] Rewon Child et al. *Generating Long Sequences with Sparse Transformers*. 2019. arXiv: 1904.10509 [cs.LG].

[Cho+18] Edward Choi et al. "Mime: Multilevel medical embedding of electronic health records for predictive healthcare." In: *Advances in neural information processing systems*. 2018, pp. 4547–4557.

[Cho+20] Edward Choi et al. "Learning the Graphical Structure of Electronic Health Records with Graph Convolutional Transformer." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (Apr. 2020), pp. 606–613. DOI: 10.1609/aaai.v34i01.5400.

[Cho18] Francois Chollet. *Deep Learning with Python*. Manning Publications Co., 2018.

[Con13] L. Condat. "A Direct Algorithm for 1-D Total Variation Denoising." In: *IEEE Signal Processing Letters* 20.11 (Nov. 2013), pp. 1054–1057. DOI: 10.1109/LSP.2013.2278339.

[Cor17] Core Team, R. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2017.

[CL07] Ricardo Cortez and Allan D Levi. "Acute spinal cord injury." In: *Current treatment options in neurology* 9.2 (2007), pp. 115–125.

[Cox72] David R Cox. "Regression models and life-tables." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2 (1972), pp. 187–202.

[Cug+18] Marc Cuggia et al. *Health Data Hub: mission de préfiguration*. Tech. rep. In French. Ministère des Solidarités et de la Santé, Oct. 2018.

[Dai+19] Zihang Dai et al. "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019). DOI: 10.18653/v1/p19-1285.

[DV03] Daryl J Daley and David Vere-Jones. “An introduction to the theory of point processes, volume 1: Elementary theory and methods.” In: *Verlag New York Berlin Heidelberg: Springer* (2003).

[Dal03] Vere-Jones D. Daley D.J. *An introduction to the theory of Point Processes - Vol. 1: Elementary theory and methods*. 2003. DOI: 10.1007/b97277.

[Dat19] Hard Drive Data. “Stats/Backblaze.” In: *URL: https://www. backblaze. com/b2/hard-drive-test-data. html. Checked 24.03* (2019).

[Dea+10] Silvia Deandrea et al. “Review Article: Risk Factors for Falls in Community-dwelling Older People: ”A Systematic Review and Meta-analysis”.” In: *Epidemiology* 21 (2010), pp. 658–668.

[Deh+15] Khaled Dehdouh et al. “Using the column oriented NoSQL model for implementing big data warehouses.” In: *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering, & Applied Computing. 2015, p. 469.

[Den+13] Joshua C Denny et al. “Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data.” In: *Nature biotechnology* 31.12 (2013), p. 1102.

[Dev+18] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805.

[DZ17] Carl Doersch and Andrew Zisserman. “Multi-task Self-Supervised Visual Learning.” In: *CoRR* abs/1708.07860 (2017). arXiv: 1708.07860.

[Dou+20] M Doutreligne et al. “Alignement à grande échelle du Système des données de santé vers le modèle commun de données OMOP.” In: *Revue d’Épidémiologie et de Santé Publique* 68 (2020), S37.

[Dow+17] N. S. Downing et al. “Postmarket Safety Events Among Novel Therapeutics Approved by the US Food and Drug Administration Between 2001 and 2010.” In: *JAMA* 317.18 (May 2017), p. 1854.

[Dro04] David R Drover. “Comparative pharmacokinetics and pharmacodynamics of short-acting hypnosedatives.” In: *Clinical pharmacokinetics* 43.4 (2004), pp. 227–238.

[Erh+10] Dumitru Erhan et al. “Why Does Unsupervised Pre-Training Help Deep Learning?” In: *J. Mach. Learn. Res.* 11 (Mar. 2010), pp. 625–660.

[Fai15] Jean-Luc Faillie. “Indication bias or protopathic bias?” In: *British journal of clinical pharmacology* 80.4 (2015), p. 779.

[Far95] C. P. Farrington. "Relative Incidence Estimation from Case Series for Vaccine Safety Evaluation." In: *Biometrics* 51.1 (1995), pp. 228–235.

[FW06] C. P. Farrington and H. J. Whitaker. "Semiparametric analysis of case series data." In: *Journal of the Royal Statistical Society. Series C: Applied Statistics* 55.5 (Nov. 2006), pp. 553–594.

[Far+11] C. P. Farrington et al. "Self-Controlled Case Series Analysis With Event-Dependent Observation Periods." In: *Journal of the American Statistical Association* 106.494 (June 2011), pp. 417–426.

[FDJ19] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. *Unsupervised Scalable Representation Learning for Multivariate Time Series*. 2019. arXiv: 1901.10738 [cs.LG].

[GAB15] Rolina D. van Gaalen, Michal Abrahamowicz, and David L. Buckeridge. "The impact of exposure model misspecification on signal detection in prospective pharmacovigilance." In: *Pharmacoepidemiology and Drug Safety* 24.5 (May 2015), pp. 456–467.

[Gau+17] N. Gault et al. "Self-controlled designs in pharmacoepidemiology involving electronic healthcare databases: a systematic review." In: *BMC medical research methodology* 17.1 (2017), p. 25.

[Geh+17] Jonas Gehring et al. "Convolutional sequence to sequence learning." In: *arXiv preprint arXiv:1705.03122* (2017).

[GWF16] Y. Ghebremichael-Weldeselassie, H. J. Whitaker, and C. P. Farrington. "Flexible modelling of vaccine effect in self-controlled case series models." In: *Biometrical Journal* 58.3 (2016), pp. 607–622.

[GWF17] Y. Ghebremichael-Weldeselassie, H. J. Whitaker, and C. P. Farrington. "Spline-based self-controlled case series method." In: *Statistics in Medicine* 36.19 (2017), pp. 3022–3038.

[Gib+19] Robert Gibbons et al. "Medications and Suicide: High Dimensional Empirical Bayes Screening (iDEAS)." In: *Harvard Data Science Review* 1.2 (2019).

[Gib+09] Jack E Gibson et al. "Use of self-controlled analytical techniques to assess the association between use of prescription medications and the risk of motor vehicle crashes." In: *American journal of epidemiology* 169.6 (2009), pp. 761–768.

[Gra+11] Simon Matthew Graham et al. "Risk of osteoporosis and fracture incidence in patients on antipsychotic medication." In: *Expert opinion on drug safety* 10.4 (2011), pp. 575–602.

[Gro+04] GRADE Working Group et al. "Grading quality of evidence and strength of recommendations." In: *BMJ: British Medical Journal* 328.7454 (2004), p. 1490.

[Hag+14] Yolanda Hagar et al. "Survival analysis with electronic health record data: Experiments with chronic kidney disease: Survival Analysis of EHR CKD Data." In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 7.5 (Oct. 2014), pp. 385–403.

[Han+13] Richard A. Hansen et al. "How well do various health outcome definitions identify appropriate cases in observational studies." In: *Drug Safety* 36.SUPPL.1 (Oct. 2013), pp. 27–32. DOI: 10.1007/s40264-013-0104-0.

[Har+20] Charles R Harris et al. "Array programming with NumPy." In: *Nature* 585.7825 (2020), pp. 357–362.

[Har+18] Steve Harris et al. "Critical Care Health Informatics Collaborative (CCHIC): Data, tools and methods for reproducible research: A multicentre UK intensive care database." In: *International Journal of Medical Informatics* 112 (Apr. 2018), pp. 82–89.

[Har+19] Hrayr Harutyunyan et al. "Multitask learning and benchmarking with clinical time series data." In: *Scientific Data* 6.1 (June 2019). DOI: 10.1038/s41597-019-0103-9.

[HAF16] Manfred Hauben, Jeffrey K. Aronson, and Robin E. Ferner. "Evidence of Misclassification of Drug-Event Associations Classified as Gold Standard 'Negative Controls' by the Observational Medical Outcomes Partnership (OMOP)." In: *Drug Safety* 39.5 (May 2016), pp. 421–432.

[HO74] A. G Hawkes and D. Oakes. "A cluster process representation of a self-exciting process." In: *Journal of Applied Probability* 11.3 (1974), pp. 493–503.

[HS+79] US Department of Health, Human Services, et al. "The Belmont report: Ethical principles and guidelines for the protection of human subjects of research." In: *US Department of Health and Human Services* (1979).

[Hén+19] Olivier J. Hénaff et al. *Data-Efficient Image Recognition with Contrastive Predictive Coding*. 2019. arXiv: 1905.09272 [cs.CV].

[HG16] Dan Hendrycks and Kevin Gimpel. "Gaussian error linear units (gelus)." In: *arXiv preprint arXiv:1606.08415* (2016).

[Het+19] Bhagya Hettige et al. "MedGraph: Structural and Temporal Representation Learning of Electronic Medical Records." In: *arXiv preprint arXiv:1912.03703* (2019).

[HBB20] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. "Recurrent neural networks for time series forecasting: Current status and future directions." In: *International Journal of Forecasting* (2020).

[Hin+11] Benjamin Hindman et al. "Mesos: A platform for fine-grained resource sharing in the data center." In: *NSDI*. Vol. 11. 2011. 2011, pp. 22–22.

[Hon+18] Na Hong et al. "Preliminary exploration of survival analysis using the OHDSI common data model: a case study of intrahepatic cholangiocarcinoma." In: *BMC Medical Informatics and Decision Making* 18.S5 (Dec. 2018), p. 116. DOI: 10.1186/s12911-018-0686-7.

[Hon+07] Adriaan Honig et al. "Treatment of post-myocardial infarction depressive disorder: a randomized, placebo-controlled trial with mirtazapine." In: *Psychosomatic medicine* 69.7 (2007), pp. 606–613.

[HR18] Jeremy Howard and Sebastian Ruder. *Universal Language Model Fine-tuning for Text Classification*. 2018. arXiv: 1801.06146 [cs.CL].

[HA13] G. Hripcsak and D. J. Albers. "Next-generation phenotyping of electronic health records." In: *Journal of the American Medical Informatics Association* 20.1 (Jan. 2013), pp. 117–121.

[HAP11] George Hripcsak, David J Albers, and Adler Perotte. "Exploiting time in electronic health record correlations." en. In: *Journal of the American Medical Informatics Association* 18.Supplement_1 (Dec. 2011), pp. i109–i115.

[Hri+15] George Hripcsak et al. "Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers." In: *Studies in health technology and informatics* 216 (2015), p. 574.

[Hri+16] George Hripcsak et al. "Characterizing treatment pathways at scale using the OHDSI network." In: *Proceedings of the National Academy of Sciences* 113.27 (July 2016), pp. 7329–7336.

[Hub+03] Richard Hubbard et al. "Exposure to tricyclic and selective serotonin reuptake inhibitor antidepressants and the risk of hip fracture." In: *American journal of epidemiology* 158.1 (2003), pp. 77–84.

[Hus+16] Vojtech Huser et al. "Multisite evaluation of a data quality tool for patient-level clinical data sets." In: *eGEMs* 4.1 (2016).

[HS15] Kyuyeon Hwang and Wonyong Sung. "Single stream parallelization of generalized LSTM-like RNNs on a GPU." In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 1047–1051.

[Jan+17] Anne-Sophie Jannot et al. "The georges pompidou university hospital clinical data warehouse: a 8-years follow-up experience." In: *International journal of medical informatics* 102 (2017), pp. 21–28.

[Joh+16] Alistair EW Johnson et al. "MIMIC-III, a freely accessible critical care database." In: *Scientific data* 3 (2016), p. 160035.

[JZ13] Rie Johnson and Tong Zhang. "Accelerating stochastic gradient descent using predictive variance reduction." In: *Advances in neural information processing systems*. 2013, pp. 315–323.

[Kab+19] Anastasiia Kabeshova et al. *ZiMM: a deep learning model for long term and blurry relapses with non-clinical claims data*. 2019. arXiv: 1911.05346 [cs.LG].

[Kat+20] Angelos Katharopoulos et al. *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention*. 2020. arXiv: 2006.16236 [cs.LG].

[KKL20] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. *Reformer: The Efficient Transformer*. 2020. arXiv: 2001.04451 [cs.LG].

[Klu+16] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows." In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[Kum+19] Sathiya P. Kumar et al. *SCALPEL-Flattening*. 2019.

[LTD09] Malcolm Lader, Andre Tylee, and John Donoghue. "Withdrawing benzodiazepines in primary care." In: *CNS drugs* 23.1 (2009), pp. 19–34.

[Lan+19] Zhenzhong Lan et al. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. 2019. arXiv: 1909.11942 [cs.CL].

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." In: *nature* 521.7553 (2015), pp. 436–444.

[Lew+15] J. D Lewis et al. "Pioglitazone use and risk of bladder cancer and other common cancers in persons with diabetes." In: *Jama* 314.3 (2015), pp. 265–277.

[Li+20] Yikuan Li et al. "BEHRT: Transformer for Electronic Health Records." In: *Scientific Reports* 10.1 (Apr. 2020). DOI: 10.1038/s41598-020-62922-y.

[LP14] Yinan Li and Jignesh M Patel. "Widetable: An accelerator for analytical data processing." In: *Proceedings of the VLDB Endowment* 7.10 (2014), pp. 907–918.

[LN89] D. C. Liu and J. Nocedal. “On the Limited Memory BFGS Method for Large Scale Optimization.” In: *Math. Program.* 45.3 (Dec. 1989), pp. 503–528. DOI: 10.1007/BF01589116.

[Liu+20] Liyuan Liu et al. “On the Variance of the Adaptive Learning Rate and Beyond.” In: *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*. Apr. 2020.

[Liu+19] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[Loo19] Vincent Looten. “Are studies of claims databases reproducible? The hypothesis of an instituted ethical misconduct in public health.” In: *Medecine sciences* 35.8-9 (2019), pp. 689–692.

[MSR13] David Madigan, Martijn J. Schuemie, and Patrick B. Ryan. “Empirical performance of the case-control method: Lessons for developing a risk identification and analysis system.” In: *Drug Safety* 36.SUPPL.1 (Oct. 2013), pp. 59–72.

[Mad+14] David Madigan et al. “A Systematic Statistical Approach to Evaluating Evidence from Observational Studies.” In: *Annual Review of Statistics and Its Application* 1.1 (2014), pp. 11–39. DOI: 10.1146/annurev-statistics-022513-115645.

[Mar+12] J. Marc Overhage et al. “Validation of a common data model for active safety surveillance research.” In: *Journal of the American Medical Informatics Association* 19.1 (Jan. 2012), pp. 54–60. DOI: 10.1136/amiajnl-2011-000376.

[Mar+15] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[Mel+10] Sergey Melnik et al. “Dremel: Interactive Analysis of Web-scale Datasets.” In: *Proc. VLDB Endow.* 3.1-2 (Sept. 2010), pp. 330–339. DOI: 10.14778/1920841.1920886.

[Mey+18] Sebastian Meyer et al. “Self-Exciting Point Processes: Infections and Implementations.” In: *Statistical Science* 33.3 (2018), pp. 327–329.

[MTC91] JM Meythaler, SM Tuel, and LL Cross. “Spinal cord seizures: a possible cause of isolated myoclonic activity in traumatic spinal cord injury.” In: *Spinal Cord* 29.8 (1991), pp. 557–560.

[MFL16] Joëlle Micallef, Elisabeth Frauger, and Maryse Lapeyre-Mestre. “Misuse of benzodiazepines in France.” In: *Neuropathology of Drug Addictions and Substance Misuse*. Elsevier, 2016, pp. 1101–1111.

[Mil11] Rupert G Miller Jr. *Survival analysis*. Vol. 66. John Wiley & Sons, 2011.

[MH20] Aya A Mitani and Sebastien Haneuse. “Small Data Challenges of Studying Rare Diseases.” In: *JAMA Network Open* 3.3 (2020), e201965–e201965.

[MRM16] R. Moghaddass, C. Rudin, and D. Madigan. “The Factorized Self-Controlled Case Series Method: An Approach for Estimating the Effects of Many Drugs on Many Outcomes.” In: *Journal of Machine Learning Research* 17 (2016), pp. 1–24.

[Mon+11] Jean-Louis Montastruc et al. “Benefits and strengths of the disproportionality analysis for identification of adverse drug reactions in a pharmacovigilance database.” In: *British journal of clinical pharmacology* 72.6 (2011), pp. 905–908.

[Mor+20] Maryan Morel et al. “ConvSCCS: convolutional self-controlled case series model for lagged adverse event detection.” In: *Biostatistics* 21.4 (2020), pp. 758–774.

[Mor+13] Janne Kaergaard Mortensen et al. “Post stroke use of selective serotonin reuptake inhibitors and clinical outcome among patients with ischemic stroke: a nationwide propensity score–matched follow-up study.” In: *Stroke* 44.2 (2013), pp. 420–426.

[Mul17] Marianne Muller. *728 000 résidents en établissements d’hébergement pour personnes âgées en 2015,* in French. Tech. rep. 1015. Études et Résultats, Drees, 2017.

[Mur+10] Shawn N Murphy et al. “Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2).” In: *Journal of the American Medical Informatics Association* 17.2 (2010), pp. 124–130.

[NS19] Greta Nemergut and Jennifer Sandra. “Neuroleptics (Typical/Atypical Antipsychotics).” In: *Pain*. Springer, 2019, pp. 255–260.

[Nes83] Yu Nesterov. “A method of solving a convex programming problem with convergence rate $O(1/k^2)$.” In: *Sov. Math. Dokl*. Vol. 27. 2. 1983.

[Neu+12] A Neumann et al. “Pioglitazone and risk of bladder cancer among diabetic patients in France: a population-based cohort study.” In: *Diabetologia* 55.7 (2012), pp. 1953–1962.

[NN19] Peter Nordström and Anna Nordström. “Use of short-acting and long-acting hypnotics and the risk of fracture: a critical analysis of associations in a nationwide cohort.” In: *Osteoporosis International* 30.10 (2019), pp. 1983–1993.

[Nor+13] G Niklas Norén et al. “Empirical performance of the calibrated self-controlled cohort analysis within temporal pattern discovery: lessons for developing a risk identification and analysis system.” In: *Drug safety* 36.1 (2013), pp. 107–121.

[Nor+10] G. N. Norén et al. “Temporal pattern discovery in longitudinal electronic patient records.” In: *Data Mining and Knowledge Discovery* 20.3 (2010), pp. 361–387.

[Ode+04] Martin Odersky et al. *An overview of the Scala programming language*. Tech. rep. École Polytechnique Fédérale de Lausanne, 2004.

[Oga81] Y. Ogata. “On Lewis’ simulation method for point processes.” In: *IEEE Transactions on Information Theory* 27.1 (1981), pp. 23–31.

[Oga99] Yosihiko Ogata. “Seismicity analysis through point-process modeling: A review.” In: *Seismicity patterns, their statistical significance and physical meaning*. Springer, 1999, pp. 471–507.

[Ong+17] Toan C Ong et al. “Dynamic-ETL: a hybrid approach for health data extraction, transformation and loading.” In: *BMC medical informatics and decision making* 17.1 (2017), p. 134.

[OLV18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation Learning with Contrastive Predictive Coding.” In: *CoRR* abs/1807.03748 (2018). arXiv: 1807.03748.

[Ora08] Oracle. *Exadata Database Machine*. 2008.

[Pas+17] Adam Paszke et al. *Automatic differentiation in PyTorch*. 2017.

[Pau+19] Daniel Paula e Silva et al. *SCALPEL-Extraction*. 2019.

[PDZ06] Roger D Peng, Francesca Dominici, and Scott L Zeger. “Reproducible epidemiologic research.” In: *American journal of epidemiology* 163.9 (2006), pp. 783–789.

[Pie+01] Corinne Pierfitte et al. “Benzodiazepines and hip fractures in elderly people: case-control study.” In: *Bmj* 322.7288 (2001), pp. 704–708.

[Piv+14] Rimma Pivovarov et al. “Identifying and mitigating biases in EHR laboratory tests.” In: *Journal of Biomedical Informatics* 51 (Oct. 2014), pp. 24–34.

[Pos96] Behandelt PostgreSQL. “PostgreSQL.” In: *Web resource: http://www. PostgreSQL. org/about* (1996).

[Pow16] Joshua Powers. “Apache Spark Performance Compared to a Traditional Relational Database using Open Source Big Data Health Software.” In: (2016).

[Pra+11] Nicole Pratt et al. “Risk of hospitalization for hip fracture and pneumonia associated with antipsychotic prescribing in the elderly.” In: *Drug safety* 34.7 (2011), pp. 567–575.

[Rad+19] Alec Radford et al. “Language Models are Unsupervised Multitask Learners.” In: (2019).

[Ras+20] Laila Rasmy et al. *Med-BERT: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction*. 2020. arXiv: 2005.12833 [cs.CL].

[RJ+91] Sarunas J Raudys, Anil K Jain, et al. “Small sample size effects in statistical pattern recognition: Recommendations for practitioners.” In: *IEEE Transactions on pattern analysis and machine intelligence* 13.3 (1991), pp. 252–264.

[Rei+18] Alex Reinhart et al. “A review of self-exciting spatio-temporal point processes and their applications.” In: *Statistical Science* 33.3 (2018), pp. 299–318.

[Rei+10] Stephanie J Reisinger et al. “Development and evaluation of a common data model enabling active drug safety surveillance using disparate healthcare databases.” In: *Journal of the American Medical Informatics Association* 17.6 (2010), pp. 652–662.

[Req+20] Gema Requena et al. “Impact of pre-exposure time bias in self-controlled case series when the event conditions the exposure: Hip/femur fracture and use of benzodiazepines as a case study.” In: *Pharmacoepidemiology and Drug Safety* (2020).

[RM51] Herbert Robbins and Sutton Monro. “A stochastic approximation method.” In: *The annals of mathematical statistics* (1951), pp. 400–407.

[Ros+10] Paul R Rosenbaum et al. *Design of observational studies*. Vol. 10. Springer, 2010.

[Rud16] Sebastian Ruder. “An overview of gradient descent optimization algorithms.” In: *arXiv preprint arXiv:1609.04747* (2016).

[ROF92] Leonid I Rudin, Stanley Osher, and Emad Fatemi. “Nonlinear total variation based noise removal algorithms.” In: *Physica D: nonlinear phenomena* 60.1-4 (1992), pp. 259–268.

[RSM13] Patrick B Ryan, Martijn J Schuemie, and David Madigan. “Empirical performance of a self-controlled cohort method: lessons for developing a risk identification and analysis system.” In: *Drug safety* 36.1 (2013), pp. 95–106.

[Rya+13a] Patrick B Ryan et al. “Defining a reference set to support methodological research in drug safety.” In: *Drug safety* 36.1 (2013), pp. 33–47.

[Rya+12] Patrick B. Ryan et al. “Empirical assessment of methods for risk identification in healthcare data: results from the experiments of the Observational Medical Outcomes Partnership.” In: *Statistics in Medicine* 31.30 (Dec. 2012), pp. 4401–4415.

[Rya+13b] Patrick B. Ryan et al. “A comparison of the empirical performance of methods for a risk identification system.” In: *Drug Safety* 36.SUPPL.1 (Oct. 2013), pp. 143–158.

[San07] Haute Autorité de Santé. “Améliorer la prescription des psychotropes chez le sujet âgé.” In: *Propositions d’actions concertées, version courte. Saint-Denis* (2007).

[Sup76] SAS Support. *SAS Enterprise Guide*. 1976.

[SM14] M. J. Schuemie and M. Moinat. *WhiteRabbit*. 2014.

[Sch+16] M. J. Schuemie et al. “Detecting adverse drug reactions following long-term exposure in longitudinal observational data: The exposure-adjusted self-controlled case series.” In: *Statistical methods in medical research* 25.6 (2016), pp. 2577–2592.

[Sch11] Martijn J Schuemie. “Methods for drug safety signal detection in longitudinal observational databases: LGPS and LEOPARD.” In: *Pharmacoepidemiology and drug safety* 20.3 (2011), pp. 292–299.

[Sch+04] S.C.E Schuit et al. “Fracture incidence and association with bone mineral density in elderly men and women: the Rotterdam Study.” In: *Bone* 34.1 (2004), pp. 195–202.

[Seb+19] Youcef Sebiat et al. *SCALPEL-Analysis*. 2019.

[Sep+18a] Lotta J. Seppala et al. “Fall-Risk-Increasing Drugs: A Systematic Review and Meta-Analysis: II. Psychotropics.” In: *Journal of the American Medical Directors Association* 19.4 (Apr. 2018), 371.e11–371.e17.

[Sep+18b] Lotta J. Seppala et al. “Fall-Risk-Increasing Drugs: A Systematic Review and Meta-analysis: III. Others.” In: *Journal of the American Medical Directors Association* 19.4 (Apr. 2018), 372.e1–372.e8.

[SRB19] Nida Shahid, Tim Rappon, and Whitney Berta. “Applications of artificial neural networks in health care organizational decision-making: A scoping review.” In: *PloS one* 14.2 (2019), e0212356.

[Sha+19] Junyuan Shang et al. “Pre-training of Graph Augmented Transformers for Medication Recommendation.” In: *CoRR* abs/1906.00346 (2019). arXiv: 1906.00346.

[Shi+18] Benjamin Shickel et al. "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis." In: *IEEE Journal of Biomedical and Health Informatics* 22.5 (Sept. 2018), pp. 1589–1604. DOI: 10.1109/jbhi.2017.2767063.

[Shv+10] Konstantin Shvachko et al. "The Hadoop Distributed File System." In: *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. MSST '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1–10. DOI: 10.1109/MSST.2010.5496972.

[Sim+13] S. E. Simpson et al. "Multiple self-controlled case series for large-scale longitudinal observational databases." In: *Biometrics* 69.4 (2013), pp. 893–902.

[Son+18] Huan Song et al. "Attend and diagnose: Clinical time series analysis using attention models." English (US). In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018 ; Conference date: 02-02-2018 Through 07-02-2018. AAAI press, Jan. 2018, pp. 4091–4098.

[IBM68] SPSS IBM. *Statistical Package for the Social Sciences*. 1968.

[Sta+10] Paul E Stang et al. "Advancing the science for active surveillance: rationale and design for the Observational Medical Outcomes Partnership." In: *Annals of internal medicine* 153.9 (2010), pp. 600–606.

[Sun+19] Chen Sun et al. *Learning Video Representations using Contrastive Bidirectional Transformer*. 2019. arXiv: 1906.05743 [cs.LG].

[Tan+20] Shengpu Tang et al. "Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data." In: *Journal of the American Medical Informatics Association* (2020).

[Thu+20] Nicolas H. Thurin et al. "Empirical assessment of case-based methods for drug safety alert identification in the French National Healthcare System database (SNDS): Methodology of the ALCAPONE project." In: *Pharmacoepidemiology and Drug Safety* (2020). DOI: 10.1002/pds.4983.

[Tol+17] Anna-Maija Tolppanen et al. "Screening approach for identifying candidate drugs and drug-drug interactions related to hip fracture risk in persons with Alzheimer disease." In: *Pharmacoepidemiology and drug safety* 26.8 (2017), pp. 875–889.

[Tre+17] Nir Treves et al. "Z-drugs and risk for falls and fractures in older adults – a systematic review and meta-analysis." In: *Age and Ageing* 47.2 (Oct. 2017), pp. 201–208.

[Tri+09] G. Trifiro et al. “The EU-ADR project: preliminary results and perspective.” In: *Studies in health technology and informatics* 148 (2009), pp. 43–9.

[Tsa+17] Avraam Tsantekidis et al. “Forecasting stock prices from the limit order book using convolutional neural networks.” In: *2017 IEEE 19th Conference on Business Informatics (CBI)*. Vol. 1. IEEE. 2017, pp. 7–12.

[Tup+10a] P. Tuppin et al. “French national health insurance information system and the permanent beneficiaries sample.” In: *Revue d’Épidémiologie et de Santé Publique* 58.4 (2010), pp. 286–290.

[Tup+10b] P. Tuppin et al. “French national health insurance information system and the permanent beneficiaries sample.” In: *Revue d’Épidémiologie et de Santé Publique* 58.4 (2010), pp. 286–290.

[Tup+17a] P. Tuppin et al. “Value of a national administrative database to guide public decisions: From the système national d’information interrégimes de l’Assurance Maladie (SNIIRAM) to the système national des données de santé (SNDS) in France.” In: *Revue d’Épidémiologie et de Santé Publique* 65 (Oct. 2017), S149–S167. DOI: 10.1016/j.respe.2017.05.004.

[Tup+17b] P. Tuppin et al. “Value of a national administrative database to guide public decisions: From the système national d’information interrégimes de l’Assurance Maladie (SNIIRAM) to the système national des données de santé (SNDS) in France.” In: *Revue d’Épidémiologie et de Santé Publique* 65 (2017). Réseau REDSIAM, S149–S167.

[TVW05] Berwin A Turlach, William N Venables, and Stephen J Wright. “Simultaneous variable selection.” In: *Technometrics* 47.3 (2005), pp. 349–363.

[Typ16] Typesafe. *HOCON (Human-Optimized Config Object Notation)*. 2016.

[Vas+17] Ashish Vaswani et al. “Attention is all you need.” In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[Vel+17] Petar Velivckovic et al. “Graph attention networks.” In: *arXiv preprint arXiv:1710.10903* (2017).

[Ver04] Annemiek Vermeeren. “Residual effects of hypnotics.” In: *CNS drugs* 18.5 (2004), pp. 297–328.

[Ves09] Peter Vestergaard. “Fracture risks of antidepressants.” In: *Expert review of neurotherapeutics* 9.1 (2009), pp. 137–141.

[Voh16] Deepak Vohra. “Apache parquet.” In: *Practical Hadoop Ecosystem*. Springer, 2016, pp. 325–335.

[Vri+18] Max de Vries et al. “Fall-Risk-Increasing Drugs: A Systematic Review and Meta-Analysis: I. Cardiovascular Drugs.” In: *Journal of the American Medical Directors Association* 19.4 (Apr. 2018), 371.e1–371.e9.

[Wan+16] S. V. Wang et al. “Transparency and reproducibility of observational cohort studies using large healthcare databases.” In: *Clinical Pharmacology and Therapeutics* 99.3 (Mar. 2016), pp. 325–332. DOI: 10.1002/cpt.329.

[Wan+19] Yue Wang et al. “Dynamic graph cnn for learning on point clouds.” In: *Acm Transactions On Graphics (tog)* 38.5 (2019), pp. 1–12.

[Was13] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[Wei+08] Zhou Wei et al. “Service-oriented data denormalization for scalable web applications.” In: *Proceedings of the 17th international conference on World Wide Web*. 2008, pp. 267–276.

[Whi+18] Heather J Whitaker et al. “Investigating the assumptions of the self-controlled case series method.” In: *Statistics in medicine* 37.4 (2018), pp. 643–658.

[Wis+17] David S Wishart et al. “DrugBank 5.0: a major update to the DrugBank database for 2018.” In: *Nucleic Acids Research* 46.D1 (Nov. 2017), pp. D1074–D1082.

[Woo+09] John C. Woolcott et al. “Meta-analysis of the Impact of 9 Medication Classes on Falls in Elderly Persons.” In: *Archives of internal medicine* 169.21 (Nov. 2009), p. 1952.

[Wu+17] Chao-Yuan Wu et al. “Sampling matters in deep embedding learning.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2840–2848.

[Wu+20] Zhanghao Wu et al. “Lite Transformer with Long-Short Range Attention.” In: *arXiv preprint arXiv:2004.11886* (2020).

[XZ14] L. Xiao and T. Zhang. “A Proximal Stochastic Gradient Method with Progressive Variance Reduction.” In: *SIAM Journal on Optimization* 24 (2014), pp. 2057–2075.

[Xu+11] S. Xu et al. “Identifying optimal risk windows for self-controlled case series studies of vaccine safety.” In: *Statistics in Medicine* 30.7 (2011), pp. 742–752.

[Yan+19] Zhilin Yang et al. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2019. arXiv: 1906.08237 [cs.CL].

[Ye+19] Zihao Ye et al. “BP-Transformer: Modelling Long-Range Context via Binary Partitioning.” In: *arXiv preprint arXiv:1911.04070* (2019).

[You+18] T. Young et al. “Recent Trends in Deep Learning Based Natural Language Processing [Review Article].” In: *IEEE Computational Intelligence Magazine* 13.3 (2018), pp. 55–75.

[YL06] M. Yuan and Y. Lin. “Model selection and estimation in regression with grouped variables.” In: *Journal of the Royal Statistical Society, series B* 68 (2006), pp. 49–67.

[Zah+16] Matei Zaharia et al. “Apache Spark: A Unified Engine for Big Data Processing.” In: *Commun. ACM* 59.11 (Oct. 2016), pp. 56–65. DOI: 10.1145/2934664.

[Zha+19] Michael R. Zhang et al. *Lookahead Optimizer: k steps forward, 1 step back*. 2019. arXiv: 1907.08610 [cs.LG].

[Zho+12] J. Zhou et al. “Modeling disease progression via fused sparse group lasso.” In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2012, pp. 1095–1103.

[Zuu+09] Alain Zuur et al. *Mixed Effects Models and Extensions in Ecology with R*. ISBN 978-0-387-87457-9. Springer, 2009.

List of acronyms

ADR	Adverse Drug Reaction
AHAN	Anxiolytics, Hypnotics, Antidepressants, Neuroleptics
API	Application Programming Interface
ATC	Anatomical Therapeutic Chemical classification system
AUPRC	Area Under Precision-Recall Curve
AUROC	Area Under Receiver Operating Characteristic Curve
BERT	Bidirectional Encoder Representations from Transformers
BDOM	*Bases de Données Observationnelles Massive*
CCAM	*Classification commune des actes médicaux* – French medical procedures classification
OMOP CDM	Observational Medical Outcomes Partnership Common Data Model
CI	Confidence interval
CIM	*Classification Internationale des Maladies*
CNAM	*Caisse Nationale de l'Assurance Maladie* – French national health insurance agency
CNIL	*Commission Nationale de l'Informatique et des Libertés* – French National Commission on Informatics and Liberty
CNN	Convolutional Neural Network
CPC	Contrastive Predictive Coding
CPU	Central Processing Unit
CRR	*Courbe de Risque Relatif*
CSV	Comma-Separated Values
DCIR	*Données de Consommation Inter-Régimes* – Inter-scheme consumption data
EHR	Electronic Health Records
EIM	*Effet Indésirable Médicamenteux*
ERC	*Essai Randomisé Contrôlé*
FFN	Feed-Forward Network
GAT	Graph ATtention network

GPU	Graphical Processing Unit
PMSI-HAD	*Hospitalisation À Domicile* – Home to home care
HDFS	Hadoop Distributed File System
HOCON	Human-Optimized Config Object Notation
ICD	International Classification of Diseases
ICTPD	Information Component Temporal Pattern Discovery
ICU	Intensive Care Unit
IHM	In-Hospital Mortality
IRR	Incidence Rate Ratio
LOD	Large Observational Database
LOS	Length Of Stay
LPP	*Liste des produits et prestations* – List of product and services
LR	Logistic Regression
LSTM	Long Short-Term Memory
ReLU	Rectified Linear Unit
MAE	Mean Absolute Error
MI	Mutual Information
MIMIC-III	Medical Information Mart for Intensive Care III
MLM	Masked Language Model
MSA	Multi-head Self-Attention
NABM	*Nomenclature des Actes de Biologie Médicale* – Classification of clinical pathology procedures
NCE	Noise Contrastive Estimation
NLP	Natural Language Processing
OHDSI	Observational Health Data Sciences and Informatics
OMOP	Observational Medical Outcomes Partnership
PHE	Phenotyping
PMSI	*Programme de Médicalisation des Systèmes d'Information* – Medical information system programme
PMSI-MCO	*Médecine, Chirurgie, Obstétrique et Odontologie* – Acute care ward
PMSI-PSY	Psychiatric care
RCT	Randomized controlled trial
RNN	Recurrent Neural Network
RR	Relative Risk
RRC	Relative Risk Curve
SCALPEL	SCAlable Pipeline for hEaLth data

SCCS	Self-Controlled Case Series
SGD	Stochastic Gradient Descent
SNDS	*Système National des Données de Santé* – National Health Data System
SNIIR-AM	*Système National d'Information Inter-Régimes* – National Health Insurance Information System
SNRI	Serotonin-Norepinephrine Reuptake Inhibitors
SQL	Structured Query Language
SSR	*Soins de Suite et Réadaptation* – Rehabilitation care
SSRI	Selective Serotonin Reuptake Inhibitors
SVRG	Stochastic Variance Reduced Gradient
TAL	*Traitement Automatique du Language*
TCA	Tricyclic Antidepressants
TTCA	Tetracyclic Antidepressants
TTE	Time To Event
TV	Total-Variation

INDEX

$L-$smooth function, 18
`Extractor`, 11
`CohortCollection`, 66
`CohortFlow`, 66
`Cohort`, 10, 66
`Event`, 9, 64
`Extractor`, 10, 64
`FeatureDriver`, 67
`Patient`, 64
`Transformer` (ETL), 64

adverse drug reaction, 14, 84, 112
Apache Spark, 8
attention model, 43, 44
    causal attention, 52, 192
    graph attention network, GAT, 45, 52, 199
    linear transformer, 45, 52, 199
    multi-head self-attention, 44, 52, 198

batch gradient descent, 20

case, 24
case-control study, 24
case-crossover, 25
cohort study, 24
columnar storage, 8, 62
conditional intensity, 21
conditional Poisson regression, *see* self-controlled case series
confounding by indication, 123
contrastive predictive coding, 48, 201
control, 24
convolutional neural network, 42, 43
    kernel, 42
convolutional SCCS, 29, 89
ConvSCCS, *see* convolutional SCCS
counting process, 21
Cox model, 22

data model, 7
    denormalized, 8, 60, 62
    i2b2, 7
    normalized, 7, 59
    OMOP CDM, 7
deep learning, 40
    layer, 40
disproportionality analysis, 26
distributed computing, 60
    shuffle, 60

embedding, 42, 190
    positional, 44
encoder, 42, 190
exposure
    antidepressants, 115
    antidiabetics, 34, 97
    anxiolytics, 115
    hypnotics, 115
    neuroleptics, 115
    pre-exposure, 37, 115
Extract-Transform-Load, ETL, 58

feed forward network, 45

goodness-of-fit, 17

Hadoop File System, 8

hazard function, 22
    cumulative, 22
hazard ratio, 23
Health Data Hub, 4

in-hospital mortality, 54, 195
indication bias, 16
information component temporal pattern discovery, 25
inpatient, 2

large observational database, 2
    claims data, 2
    electronic health records, 2
length-of-stay, 54, 195
Lipschitz constant, 18
longitudinal gamma Poisson shrinker, 26

masked language model, 48, 201
mini-batch, 20
multitask learning, 40
mutual information, 49

new-users, 24
noise contrastive estimation, 49

observational study, 2
OHDSI, 7
OMOP, 7
optimization, 19
outcome, 86
    bladder cancer, 34, 97
    fracture, 37, 114
outpatient, 3
overfitting, 17

penalty, 17
    group-Lasso, 18, 91
    Lasso, 18
    total variation, 19, 31, 91
phenotyping, 5, 59
    acute care, 54, 195
    automatic, 5
    manual, 5
    task, 54, 195
point process, 21
Poisson process, 21
    inhomogeneous, 21
pooler, 52, 192
prediction function, 17
protopathic bias, 26
proximal operator, 18

randomized control trial, 1
recurrent neural network, 43
relative incidence curve, *see* relative risk curve
relative risk curve, 32, 34
right-censored, 22
risk period, 24
    exposure period, 25

SCCS, *see* self-controlled case series
self-controlled case series, 25, 86
    discrete-time, 29, 90, 102
    likelihood, 102
self-controlled cohort, 25
self-controlled designs, 24
self-supervised learning, 41
    contrastive predictive coding, *see* contrastive predictive coding
    masked language model, *see* masked language model
    triplet loss, *see* triplet loss
SNDS, 3, 6, 61
    DCIR, 3, 4, 6
    PMSI, 3, 4, 6
SNIIR-AM, 3
step size, 20
stochastic gradient descent, 20
    stochastic variance reduced gradient, 21
supervised learning, 17

survival analysis, 22
survival function, 22

time-to-event, 22
transfer learning, 41
    downstream task, 41
    pretext task, 41
triplet loss, 48, 230

unsupervised pre-training, 41, 193

zero-inflation, 23

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre: Apprentissage automatique pour les bases de données de santé massives.

Mots clés: Bases de données observationelles massives, ETL, analyse longitudinale, effet indésirable médicamenteux, pré-entrainement non-supervisé.

Résumé: Cette thèse développe des méthodes innovantes exploitant des bases de données observationelles massives (BDOM) en santé, et plus particulièrement le Système National des Données de Santé (SNDS). Ces bases de données, à visée comptable et non épidémiologique, enregistrent des informations administratives qui accompagnent les soins et leur facturation. L'identification et l'extraction des historiques de soins nécessite ainsi des transformations coûteuses.
Le premier chapitre introduit SCALPEL3, une suite logicielle open-source qui facilite l'extraction de concepts médicaux et la manipulation de données de cohorte. Ce logiciel tire partie du calcul distribué, de la dénormalisation des données, et du stockage orienté colonne des données. SCALPEL3 est maintenant utilisée au sein de la Caisse Nationale de l'Assurance Maladie, à la Direction de la Recherche, des Études, de l'Évaluation et des Statistiques, et bientôt au sein du Health Data Hub.
Les deux chapitres suivants se concentrent sur la détection d'effets indésirable médicamenteux à partir de données du SNDS. Le chapitre 2 élabore ConvSCCS, un modèle basé sur des processus de Poisson et des techniques de régularisation. Une convolution entre des fonctions étagées et des évènements ponctuels permet l'estimation de courbes de risque longitudinales facilement interprétables. Ce modèle ré-identifie correctement une association connue entre un anti-diabétique et le cancer de la vessie à partir d'évènements de remboursement de médicaments et de diagnostiques. ConvSCCS est ensuite appliqué à la détection d'association entre l'utilisation d'anxiolytiques, d'hypnotiques, d'antidépresseurs et de neuroleptiques et le risque de fractures chez les personnes âgées (Chapitre 3). Cette étude révèle des structures temporelles inédites ainsi que des biais spécifiques au SNDS.
Enfin, le chapitre 4 s'intéresse à la construction de représentations génériques de parcours de soins. De nombreuses expériences y évaluent plusieurs types de modèles d'attention et de stratégies de pré-entrainement. Bien que les résultats ne soient pas encore satisfaisants, ce travail ouvre des pistes de recherche intéressantes.

Title: Machine learning for large observational datasets in healthcare.

Keywords: Large observational databases, ETL, longitudinal analysis, adverse drug reaction, unsupervised pre-training.

Abstract: This thesis develops innovative tools and methods to leverage large observational databases (LODs) in healthcare, with a focus on the *Système National des Données de Santé* (SNDS), one of the largest healthcare claims database. These databases record administrative information supporting the care and its billing. As SNDS data was not initially designed for research but for accounting purposes, identifying patients' healthcare history requires costly transformations.
The first chapter introduces SCALPEL3, an open-source framework easing medical concept extraction and cohort data manipulation, focusing on scalability and reproducibility. SCALPEL3 relies on distributed computing, data denormalization, and columnar storage. It is now used at the agency collecting SNDS data, at the French Ministry of Health, and soon at the national Health Data Hub in France.
The following two chapters focus on adverse drug reactions detection using SNDS data. Chapter 2 introduces ConvSCCS, a new model based on conditional Poisson processes and penalization techniques. Using a convolution between step functions and temporal events, this model estimates readily interpretable longitudinal risks. Applied to a cohort of diabetic patients, it recovers a known association between a molecule and bladder cancer from timestamped sequences of drug reimbursements and diagnoses. In Chapter 3, the same model is used to screen anxiolytic, hypnotic, antidepressant, and neuroleptic molecules for bone fracture risk among the elderly. This study reveals original patterns and SNDS-specific biases.
Finally, Chapter 4 focuses on building reusable representation for health data. Extensive experiments evaluate several deep attention models and pre-training strategies. While the results are not yet satisfying, this work opens exciting tracks for future research.

Institut Polytechnique de Paris
91120 Palaiseau, France