

DIPLOMARBEIT

**Prediktive Simulation von Praxis-optimierten  
Alerts für Arzneimittelwechselwirkungen und  
individueller pharmakogenetischer Marker  
mittels retrospektiver Daten im  
österreichischen Gesundheitswesen**

zur Erlangung des akademischen Grades

**Doktor der gesamten Heilkunde  
(Dr. med.univ.)**

an der

**Medizinischen Universität Wien**

ausgeführt am

Institut für Artificial Intelligence and Decision Support (CeMSIIS)

unter der Anleitung von

Ass.-Prof. Mag. Dr. Matthias Samwald

eingereicht von

Wolfgang Kuch

1142370

---

Ort, Datum

---

Unterschrift

## Inhalt

1	Zusammenfassung/Abstract .....	4
2	Einleitung .....	6
2.1	Hintergrund .....	6
2.1.1	Pharmakogenetik.....	6
2.1.2	Arzneimittel-Interaktionen.....	13
2.2	Fragestellung .....	15
2.2.1	Pharmakogenetik.....	15
2.2.2	Arzneimittel-Interaktionen.....	15
2.3	Zielsetzung .....	15
3	Material und Methoden .....	17
3.1	Datenquelle.....	17
3.2	Pharmakogenetik.....	17
3.2.1	Guidelines und Datenaufbereitung.....	17
3.2.2	Datenbanktabellen.....	19
3.3	Arzneimittel-Interaktionslisten.....	19
3.3.1	Datenaufbereitung.....	19
3.3.2	Aufbereitung der Interaktionslisten .....	21
3.3.3	Berechnung der Interaktionen.....	22
3.3.4	Auswertung der Interaktionen.....	23
3.3.5	Statistik.....	24
4	Resultate .....	26
4.1	Pharmakogenetik.....	26
4.1.1	DPWG-Liste.....	26
4.1.2	„Hoch-signifikant“-Liste.....	27
4.1.3	DPWG-Liste vs „Hoch-signifikant“-Liste .....	28
4.1.4	Häufigste drei CYPs einzeln und gemeinsam („Hoch-signifikant“-Liste) .....	30

4.2	Arzneimittel-Interaktionslisten.....	34
4.2.1	Interaktionsberechnung mit Reichweiteninformation.....	34
4.2.2	Interaktionsberechnung ohne Reichweiteninformation .....	43
5	Diskussion .....	55
6	Limitationen .....	59
7	Danksagung.....	60
8	Literaturverzeichnis.....	61
9	Tabellenverzeichnis.....	66
10	Abbildungsverzeichnis .....	69
11	Liste der verwendeten Abkürzungen.....	70
12	Anhang .....	71
12.1	Interaktionsprogramm-Quellcode .....	71
12.1.1	Klasse DDI_High_Memory .....	71
12.1.2	Klasse Interaction_Thread .....	76
12.1.3	Klasse InteractionATC.....	79
12.1.4	Klasse InteractionList .....	81
12.1.5	Klasse Patient.....	82
12.1.6	Klasse Receipt.....	83
12.1.7	Klasse Compara .....	85
12.2	Auswertungsprogramm-Quellcode .....	86

# 1 Zusammenfassung/Abstract

**Einleitung:** The growing knowledge about the impact of Gene-Variants to Pharmacokinetics and -dynamics, as well as lowering costs for genetic tests, enable us to increase the effectiveness and security of the most frequently used drugs. It could support the the selection of optimal medication and dosage, as well as decrease the frequency and severity of adverse drug events.

The rise of Electronic health records (EHR) and e-prescribing allows for the recognition of dangerous drug-combinations at the initiation of the prescription process so that physicians can be warned about potentially dangerous interactions. However, in clinical reality, such systems are not widely used and generated alerts are often ignored. An explanation for such behaviour is alert fatigue, triggered by too many irrelevant alerts.

This diploma thesis discusses the potential effects of two methodologies on decision support: It shows the potential effects of implementation of pharmacogenomic tests throughout Austria and the potential impact of interaction alerts based on different drug-drug-interaction lists.

**Methods:** A research-database containing Austrian prescription data from 2006 to 2007 was used as the data source. Pharmacogenomic guidelines were taken from the DPWG (Dutch Pharmacogenetics Work Group) recommendations were divided into two groups of clinical significance (all recommendations vs. only ‘highly significant’ recommendations). As drug-drug-interaction-lists, the Austria codex and two clinically highly-significant lists were compared.

**Results:** In the age group  $\geq 65$  there is a wide distribution of drugs which are listed in pharmacogenetic Guidelines (PGx-drugs). 47.24% in this age group receive at least one PGx-drug of the ‘highly significant’ recommendations list. Furthermore 72% of the study population received one PGx-drug of the DPWG-List. The two highly-significant interaction lists generate considerably fewer alerts than the Austria codex interaction list alone.

**Discussion:** Especially the age group  $\geq 65$  might profit from pharmacogenomic genotyping. Due to the higher prevalence of chronic disease they are less able to cope with side effects and adverse drug events in comparison to younger patients. Drug-drug interaction lists should support physicians and point out alternative therapy strategies.

**Einleitung:** Das wachsende Wissen über die Auswirkung von bestimmten Genvarianten auf Pharmakokinetik und -dynamik sowie sinkende Kosten genetischer Tests erlauben es mit einem einzigen Test die Effektivität und Sicherheit gängiger Pharmaka zu erhöhen. Die Arzneimittelwahl und -dosierung könnte so optimiert werden und Nebenwirkungen gemindert oder verhindert werden.

Die wachsende Etablierung von elektronischen Gesundheitsakten und e-Prescribing macht es möglich, gefährliche Arzneimittelkombinationen schon während der Verschreibung zu erkennen. In der Praxis werden solche Systeme jedoch schlecht angenommen und generierte Warnungen nur selten Aufmerksamkeit geschenkt. Hauptgrund hierfür ist die „alert fatigue“, welche vorallem durch zu viele irrelevante Warnungen hervorgerufen wird.

Diese Diplomarbeit diskutiert die potentiellen Auswirkungen von zwei Methodologien zur Entscheidungsunterstützung. Sie zeigt die potentiellen Auswirkungen einer österreichweiten Einführung von pharmakogenetischen Tests, sowie den potentiellen Impact von Interaktionswarnungen anhand verschiedener Arzneimittelinteraktionslisten aufzeigen.

**Methoden:** Als Datenquelle diente eine Forschungsdatenbank mit österreichischen Verschreibungsdaten von 2006-2007. Als pharmakogenetische Leitlinien werden die Guidelines der Dutch Pharmacogenetics Working Group (DPWG) verwendet und in zwei Gruppen (alle Recommendations vs. nur Recommendations mit hoher klinischer Signifikanz) eingeteilt. Als Arzneimittelinteraktionslisten werden der Austria Codex und zwei klinisch hochsignifikante Listen miteinander verglichen.

**Resultate:** Besonders in der Altersgruppe  $\geq 65$  sind Medikamente, welche in pharmakogenetischen Guidelines vorkommen (PGx-Medikamente) sehr häufig. 47,24% dieser Altersgruppe erhielten mindestens ein Medikament der „Hoch-signifikant“-Liste. 72% der Studienpopulation erhielten mindestens ein PGx-Medikament der DPWG-Liste. Die Verwendung der hochsignifikanten Interaktionslisten generiert deutlich weniger Alerts als der Austria Codex.

**Diskussion:** Besonders die Altersklasse  $\geq 65$  könnte wegen der starken Verbreitung von PGx-Medikamenten von pharmakogenetischer Genotypisierung profitieren, da aufgrund von häufiger chronischen Krankheiten Nebenwirkungen schlechter verkraftet werden. Dies kann in Hospitalisierungen und weitershohen Kosten für das Gesundheitssystem resultieren. Optimierte Arzneimittelinteraktionslisten können Ärzte und Ärztinnen bei der Vermeidung von Arzneimittelinteraktionen helfen und alternative Therapievorschlüsse unterbreiten.

## 2 Einleitung

Mangelnde Effektivität, Wechselwirkungen und potentiell vermeidbare Nebenwirkungen in der Arzneimitteltherapie, stellen eine signifikante Belastung für Patienten und das moderne Gesundheitswesen dar. In dieser Studie sollen zwei computerbasierte Methodologien zur Entscheidungsunterstützung beleuchtet werden, die das Potential haben, die Sicherheit und Effektivität der Arzneimitteltherapie zu erhöhen, welche aber bisher leider nur unzureichend Eingang in den klinischen Alltag gefunden haben: Einerseits Alerts zur Verhinderung von Wechselwirkungen zweier oder mehrerer Arzneimittel, andererseits Alerts und Empfehlungen zur Optimierung der Wahl und Dosierung von Arzneimitteln, basierend auf individuellen pharmakogenetischen Merkmalen von Patienten.

### 2.1 Hintergrund

Immer öfter werden im klinischen Alltag computergestützte Patientenverwaltungssysteme verwendet. Dabei werden neben der Krankengeschichte auch Allergien, Medikamente, Diagnosen und Therapien gespeichert. Dabei ist es nun möglich, anhand der im System vorhandenen Daten Vorschläge und Hinweise für die Behandlung von Patienten und Patientinnen zu machen.

#### 2.1.1 Pharmakogenetik

Es ist bekannt, dass verschiedene Patienten unterschiedlich auf dasselbe Arzneimittel reagieren. Diese Differenzen sind oft zwischen verschiedenen Populationen größer als zwischen monozygoten Zwillingen oder derselben Person zu verschiedenen Zeiten. (1)

Das Vanderbilt University Medical Center startete das PREDICT Projekt und veröffentlichte eine Studie, welche PGx-Tests (Pharmakogenetische Tests) an 9.589 Patienten, welche eine Pharmakotherapie erhielten, durchführten.(2)(3) Dabei zeigte sich, dass Genotypen, welche klinisch relevant sind, häufiger als allgemein angenommen vorkommen. Es zeigte sich, dass je nach Medikament 9,1% bis 69% der Patienten ein klinisch relevantes genetisches Profil haben (siehe Tabelle 2-1).

**Tabelle 2-1: Arzneimittel und Prozent der Studienpopulation mit relevantem genetischen Profil**

Arzneimittel	% der Studienpopulation mit relevantem genetischen Profil
Clopidogrel	28,5 %
Simvastatin	25,7 %
Warfarin	69 %
Thiopurin	9,1 %
Tacrolimus	23 %

Für die Einbindung in den klinischen Alltag werden zwei Varianten diskutiert:

1. **Pre-emptiver pharmakogenetischer Test:** Jeder Patient wird bei Geburt pharmakogenetisch untersucht und die Ergebnisse in geeigneter Weise gespeichert. Auch könnte erst bei Erreichen eines bestimmten Alters der Test durchgeführt werden.
2. **Reaktiver Pre-emptiver pharmakogenetischer Test:** Jeder Patient wird erst untersucht, wenn das erste Mal ein bekanntes Arzneimittel eingesetzt werden soll, wofür pharmakogenetische Daten verfügbar sind. Vorteil wäre, dass nur Patienten, welche potentiell davon profitieren würden, diesen Test erhalten und deshalb nicht so kostenintensiv ist.

Auch die geeignete Speicherung der Ergebnisse solcher Tests stellt eine Herausforderung dar. Es sind noch keine flächendeckend akzeptierten Speicherungssysteme für pharmakogenetische Gesundheitsdaten vorhanden. Außerdem sind klinische elektronische Entscheidungshilfen (CDS) nötig, um die umfangreichen pharmakogenetischen Daten sinnvoll auszuwerten. Die Benützung von QR-Codes (Quick Response), wie der „Medication Safety Code“ (MSC) könnten helfen, diese Probleme zu überwinden.(4) Mit Hilfe von QR-Readern, welche auf fast jedem Smartphone installiert werden können, ist es möglich, diese Informationen leicht und schnell auszulesen und auszuwerten.

### ***2.1.1.1 Definition von Pharmakogenetik***

Die Pharmakogenetik untersucht die individuellen Unterschiede in der Wirkung und deren Auftreten von Nebenwirkungen aufgrund von genetischen Polymorphismen, welche unter anderem die Pharmakokinetik und Pharmakodynamik von Arzneimitteln beeinflussen können.(5)

## Einleitung

### 2.1.1.1.1 Gene und Allele

Jedes Individuum enthält zwei sogenannte Allele jedes Gens, welches ein Protein beschreibt. Dabei wird eines von der Mutter und eines vom Vater geerbt. Je nachdem, ob ein Allel oder beide Allele in einer nicht-funktionellen Art vorliegt, ist die Metabolisierungsrate des Arzneimittels unterschiedlich. Dabei werden vier unterschiedliche Arzneimittel-Metabolisierungs Kategorien unterschieden:

**Tabelle 2-2: Arzneimittel-Metabolisierungs Kategorien(5)**

<b>Kategorie</b>	<b>Erklärung</b>
Poor Metabolizers (PM-Typ)  Defiziente Metabolisierer	Individuen vom PM-Typ haben zwei nicht-funktionelle Allele und deswegen keine oder nur sehr geringe Metabolisierungsrate (über andere Enzyme)
Intermediate Metabolizers (IM-Typ)  Intermediäre Metabolisierer	Individuen vom IM-Typ haben ein funktionelles und ein nicht-funktionelles Allel und haben deswegen eine eingeschränkte Enzym-Aktivität. Diese ist jedoch höher als beim PM-Typ. Alternativ können Mutationen die Enzymaktivität herabsetzen und so trotz zwei funktionellen Allelen einen IM-Typ hervorrufen.
Extensive Metabolizers (EM-Typ)  Normale Metabolisierer	Individuen vom EM-Typ haben zwei funktionelle Allele und deswegen eine normale Enzym-Aktivität. Dies ist der „normale“-Typ und wird in den Dosierungsrichtlinien der Arzneimittel standardmäßig angenommen.
Ultra-Rapid Metabolizers (UM-Typ)  Extrem schnelle Metabolisierer	Individuen vom UM-Typ haben ein oder zwei mutierte Allele, welches die Enzym-Aktivität verglichen mit dem EM-Typ (stark) erhöht. Alternativ kann eine Vervielfältigung der DNA für das Gen eine absolute Erhöhung der Enzyme hervorrufen.

### 2.1.1.1.2 Aufnahme und Transport

Die Aufnahme von Medikamenten aus dem Darm, der Übertritt vom Blutkreislauf an den Wirkort (zB Gehirn) und die Ausscheidung über die Leber und Niere wird über Transporter bewerkstelligt. Hier kann eine Veränderung der Transportergene eine Änderung der Bioverfügbarkeit, lokaler Anreicherung der Arzneimittel oder verlangsamte oder beschleunigte Ausscheidung zur Folge haben.



## Einleitung

### 2.1.1.1.3 Elimination von Arzneimitteln

Die Elimination von Pharmaka erfolgt meist durch die Biotransformation (metabolische Umwandlung in andere Stoffe) und Ausscheidung. Dies ist besonders bei lipophilen, und daher kaum wasserlöslichen, Stoffen notwendig, da diese Stoffe nicht über die Niere abtransportiert werden können und deshalb im Körper akkumulieren. Dabei wird die Biotransformation in sogenannte Phase I- und Phase II-Reaktionen eingeteilt.

- **Phase I-Reaktion (Funktionalisierungsreaktionen):** Diese fügen funktionelle Gruppen (-OH, -SH) hinzu bzw. legen diese für die Phase II-Reaktionen frei.
- **Phase II-Reaktion (Konjugationsreaktionen):** Diese sorgen dafür, dass über die funktionellen Gruppen am Molekül wasserlösliche Moleküle angebunden werden, damit Stoffe über Niere bzw. Galle ausgeschieden werden kann.

Jeder dieser beiden Schritte kann durch Polymorphismus betroffen sein und dadurch verschiedene Auswirkungen auf die Metabolisierung von Arzneistoffen haben.(5)

### 2.1.1.2 Auswirkungen der Pharmakogenetik

Die Pharmakogenetischen Auswirkungen können u.a. auf folgende Mechanismen zurückgeführt werden:

- Mutationen der Gene, welche einen Funktionsverlust des Enzyms mit einhergehender sehr langsamer Elimination der Pharmaka bewirken. Dabei kommt es zur Akkumulation des Medikaments und dadurch zu stärkeren Nebenwirkungen.
- Eine durch eine Genaplikation (Vermehrung von DNA-Abschnitten -> Vermehrte Synthese von Proteinen und Enzymen) erhöhte Metabolisierungsrate kann zum Wirkverlust der Pharmaka durch nicht Erreichen eines therapeutischen Plasmaspiegels kommen.
- Im Falle jener Arzneimittel, welche die Wirkung erst nach der Metabolisierung (meist durch Phase I-Reaktionen) entwickeln, kann es bei Funktionsverlust des Enzyms zum Wirkverlust der Pharmaka kommen. Diese Arzneimittel werden auch Prodrugs genannt. Als Beispiel ist hier Clopidogrel zu nennen.
- Bei der Aufnahme von Arzneistoffen über den Darm und dem Transport zum Wirkort (zB Gehirn) sind verschiedene Transporterproteine beteiligt. In Abhängigkeit vom Genotyp der Transporter kann der Transport stark variieren.

## Einleitung

- Die Wirkung von Pharmaka wird meist über eine Bindung zu Strukturen von Zellen ausgelöst (zB Enzyme, Rezeptoren). Eine Mutation dieser Strukturen kann einen Wirkverlust oder eine abgeschwächte Wirkung hervorrufen, da der Arzneistoff nicht mehr oder nur schlecht an die Struktur binden kann.

Cytochrom-P450-Enzyme (CYPs) spielen eine große Rolle in der Metabolisierung von verschiedensten Arzneistoffen. Hierbei werden verschiedene Medikamente von verschiedenen CYP-Untergruppen abgebaut. Im Folgenden werden einige Beispiele von Polymorphismen von Genen gezeigt.

Etwa 5-10% der europäischen Bevölkerung exprimiert das Enzym CYP2D6 nicht. Daraus resultiert eine extrem langsame Metabolisierung über andere Enzyme. Außerdem sind Mutationen bekannt, welche in einer herabgesetzten Enzym-Aktivität einhergeht (intermediäre Metabolisierer). 2-3% der Bevölkerung sind extrem schnelle Metabolisierer (UM-Typ). Viele verschiedene Arzneistoffe werden nahezu ausschließlich oder teilweise von CYP2D6 katalysiert. Einige Beispiele dafür sind:

- Antiarrhythmika der Klasse I
- Neuroleptika
- Opioide
- 5-HT<sub>3</sub>-Rezeptor-Antagonisten
- Antidepressiva

Etwa 2-5% der europäischen Bevölkerung exprimieren das Enzym CYP2C19 nicht. Zum Vergleich zeigen sich bei Japaner oder Chinesen mit 15-23% eine deutlich höhere Verbreitung der defizienten Metabolisierer. Der Thrombozytenaggregationshemmer Clopidogrel wird über das CYP2C19-Enzym in einen aktiven Metaboliten metabolisiert. Träger von einem defizientem Allel haben etwa um ein Drittel reduzierte Plasmaspiegel des aktiven Metaboliten.(6) Dies bewirkt einen klinisch signifikanten Effekt. In der TRITON-TIMI-38 Studie zeigt sich eine relative Risikoerhöhung von 53% von Schlaganfall, Myokardinfarkt oder kardiovaskulärem Tod bei Trägern von defizienten Allelen und eine dreifache Risikoerhöhung bei Stent-Thrombosen. (6)

ABC-Transporter (ATP binding Cassette), eine Gruppe von Membranproteinen zum aktiven Transport spezifischer Substrate durch eine Membran, sind die am meisten in Arzneimittelwirkung und -disposition untersuchten Transporter. (7) Diese werden auch oft

## Einleitung

Multidrug-resistance-Transporter (MDR-Transporter) genannt. Eine Untergruppe, das P-Glykoprotein, ist im ABCB1-Gen (MDR1-Gen) kodiert. Dieses Protein ist ein ATP-abhängiger zellulärer Efflux-Transporter, welcher verschiedene Stoffe und Arzneimittel wie Bilirubin, einige Zytostatika, Immunsuppressiva, Herzglykoside, Glucocortikoide und HIV-Protease-Inhibitoren transportiert. (7)(8)(9) P-Glykoproteine spielen bei der Sekretion von Metaboliten in Urin, Gallenflüssigkeit oder Darm und dadurch bei der Ausscheidung eine große Rolle. Auch in der Blut-Hirnschranke sind diese Transporter als Schutz des Gehirns gegen Akkumulation von verschiedenen Medikamenten verantwortlich.(10)

Digitoxin, ein Substrat dieses Transporters zeigt eine signifikant niedrigere Bioverfügbarkeit in Verbindung mit einem single nucleotid Polymorphismus (3435TT-Genotyp).(9)

In einer weiteren Studie wurden 73 Personen mit diagnostizierter Depression und mit einer Standard- oder Hochdosis-Therapie verglichen.(11) Dabei wurden die Patienten auch auf zwei verschiedene SNPs (single nucleotid polymorphisms) genetisch untersucht. Hierbei zeigte sich eine signifikant bessere Symptomreduktion bei Patienten mit Standarddosistherapie und einem Polymorphismus (rs2032583) im Gegensatz zu Patienten ohne diesem SNP. Außerdem zeigt sich bei heterozygoten Trägern dieses Polymorphismus bei Hochdosis-Therapie größere schlafassoziierte Nebenwirkungen als bei homozygoten Trägern.(11)

### *2.1.1.3 Pharmakogenetische Leitlinien*

Unter anderem das Clinical Pharmacogenetics Implementation Consortium (CPIC) und die Dutch Pharmacogenetics Working Group (DPWG) machen es sich zur Aufgabe, pharmakogenetische Guidelines für häufig verwendete Pharmaka zu publizieren.

CPIC wurde im Jahr 2009 als gemeinsames Projekt von The Pharmacogenomics Knowledgebase (PharmGKB) und dem Pharmacogenomics Research Network (PGRN) gegründet. (12) Das Ziel dieses Konsortiums ist, praktisch anwendbare Guidelines zu entwickeln, um Ergebnisse von Gen-Tests in die Verschreibungsentscheidung von Pharmaka einzubinden.

Jede CPIC Guideline folgt einem normierten Format, welches ein Standardsystem zur Beurteilung der Evidenz einschließt, Genotypen und Phenotypen miteinander verbindet und jeder vorgeschlagenen Verschreibung einen Empfehlungsgrad zuweist. Es werden wichtige Informationen vermittelt, die dem Mediziner helfen sollen, patientenspezifische Dipylophen für

## Einleitung

einzelne Gene in klinische Phenotypen oder Dosis-Gruppen einzuteilen, wodurch eine passende und personalisierte Therapie begonnen werden kann.

Die DPWG wurde 2005 von der Royal Dutch Pharmacist's Association (KNMP) ins Leben gerufen. Ihre Ziele sind:

- Dosierungsempfehlungen für die Pharmakotherapie auf Basis pharmakogenetischer Eigenschaften zu entwickeln
- Diese Empfehlungen in computerbasierte Verschreibungs-Systeme (computerized physician order entry, CPOE) zu integrieren

Dabei wird für jedes sogenannte Drug-Gene Paar und dem Phenotyp/Genotyp einer klinischen Signifikanz zugeteilt. Diese wird zwischen AA (niedrige klinische Signifikanz) bis F (sehr hohe klinische Signifikanz) unterteilt.(13,14) Tabelle 2-3 zeigt die Einteilung der klinischen Signifikanz.

**Tabelle 2-3: Klassifikation der klinischen Relevanz(13)**

<b>Klasse</b>	<b>Bedeutung</b>
AA	klinischer Effekt (NS), kinetischer Effekt (NS)
A	geringer klinischer Effekt (S): QT-Verlängerung (< 450ms weiblich, < 470ms männlich), INR Erhöhung < 4.5, Kinetischer Effekt (S)
B	Klinischer Effekt (S): kurze Beeinträchtigung (<48h) ohne permanente Schädigung, zB reduzierte Senkung des Ruhepulses, Reduzierung der Belastungstachykardie, verminderte Schmerzstillung durch Oxycodon und UAW, ausgelöst durch erhöhte Bioverfügbarkeit von Atomoxetin (Appetitlosigkeit, Schlafstörungen, etc)
C	Klinischer Effekt (S): mittellange Beeinträchtigung (48-168h) ohne permanente Schädigung, zB erhöhtes Risiko einer erfolglosen Therapie mit Trizyklischen Antidepressiva oder atypischen antipsychotischen Pharmaka; extrapyramidale Nebenwirkungen, Parkinsonismus; UAW, ausgelöst durch erhöhte Bioverfügbarkeit von trizyklischen Antidepressiva, Metoprolol, Propafenon (zentrale Effekte, zB Verwirrtheit)
D	Klinischer Effekt (S): lange Beeinträchtigung (> 168h), permanentes Symptom oder entkräftende Schädigung, zB Erfolglose Prophylaxe von Vorhofflimmern, tiefe Beinvenenthrombose
E	Klinischer Effekt (S): Erhöhtes Risiko einer erfolglosen lebensrettenden Therapie; erwartete Knochenmarkssuppression
F	Klinischer Effekt (S): Tod, Arrhythmie, unerwartete Knochenmarkssuppression

Abkürzungen: UAW (Unerwünschte Arzneimittelwirkungen), INR (International Normalized Ratio), NS (not statistically significant difference), S (statistically significant difference)

Auch das Canadian Pharmacogenomics Network for Drug Safety (CPNDS) hat schon einige Guidelines publiziert. (15)

### 2.1.2 Arzneimittel-Interaktionen

Im Jahr 2006 gab es 2,2 Mio. Krankenhausaufenthalte in Österreich. Dabei zeigte sich, dass bei 250.000 dieser Aufenthalte Diagnosen, welche auf unerwünschte Medikamentenereignisse aus sieben UAE-Kategorien (Unerwünschte Arzneimittelereignisse) hinweist, dokumentiert wurden. Dabei waren etwa 190.000 Personen (2,3% der österreichischen Bevölkerung) betroffen.

Auch andere Studien zeigen eine hohe Anzahl an Aufnahmen in Krankenhäusern durch unerwünschte Arzneimittelwirkungen (UAW). (16)(17) Diese Behandlungen wären vermeidbar, wenn das medizinische Personal geeignete Hilfsmittel hätte, diese Interaktionen zu vermeiden. Der Austria Codex, ein Nachschlagewerk mit allen in Österreich verfügbaren Medikamenten und bekannten Interaktionen, kann hierfür verwendet werden. Da jedoch alle bekannten Interaktionen, unabhängig von deren klinischen Signifikanz und Evidenz, aufgelistet werden, führt dies zu einer hohen Frequenz an Warnungen und zur sogenannten „alert fatigue“. Dabei werden durch die hohe Anzahl an Alerts ohne hoher klinischer Relevanz die Warnungen ignoriert und die klinisch signifikanten Alerts übersehen.

Um den Problemen der „alert fatigue“ und mangelnden Evidenz zu begegnen, gibt es nun Bemühungen, reduzierte Listen von klinisch hochsignifikanten und mit ausreichender klinischer Evidenz hinterlegten Arzneimittelinteraktionen zu generieren. Die Implementierung solcher Praxis-optimierten Regelwerke könnte das Problem minimieren und das Vertrauen in die Relevanz der Alerts verbessern.

#### 2.1.2.1 Ursachen von Arzneimittel-Interaktionen

Die Ursachen von Arzneimittel-Interaktionen sind meist in der Pharmakodynamik oder -kinetik zu suchen. Außerdem konkurrieren Arzneimittel beim Abbau in der Leber bzw. erhöhen oder vermindern manche Medikamente die Expression von CYPs. Dadurch werden Medikamente, welche durch dieses CYPs metabolisiert werden, entweder zu schnell oder langsam metabolisiert und ihre Wirkung verlieren bzw. die Plasmakonzentration und dadurch die Nebenwirkungen steigern. Manche Interaktionen können auch durch die Wirkmechanismen der einzelnen Pharmaka erklärt werden.

### **2.1.2.2 Interaktionslisten**

Um die vielen bekannten Arzneimittel-Interaktionen zusammenzufassen werden diese in Nachschlagewerken (zB Austria Codex) zusammengefasst. (18) Hier werden alle bisher bekannten Interaktionen zusammengefasst, unabhängig von der klinischen Relevanz. Reduzierte Listen, welche nur hochsignifikante Arzneimittelinteraktionen enthalten, werden von vielen Organisationen, wie zB der französischen Agence nationale de securite du medicament et des produits de sante (ANSM) erstellt.(19) Auch einige Unternehmen haben sich schon auf eine solche Art von Listengenerierung spezialisiert.

Der Austria Codex ist das Standard-Nachschlagewerk von Arzneimitteln in Österreich und wird in Zusammenarbeit vom Apotheker-Verlag und dem Bundesministerium für Gesundheit und Frauen einmal pro Quartal veröffentlicht. Es enthält umfassende Informationen über sämtliche in Österreich zugelassenen Arzneimittel, unter anderem auch bekannte Arzneimittelinteraktionen. Diese sind nach ihrem klinischen Schweregrad eingeteilt (leicht, mittel, schwer).(18)

Die französische Interaktionsliste wurde von der Agence nationale de securite du medicament et des produits de sante (ANSM) erstellt. Bei der Entwicklung dieser Liste wurde besonders auf die klinische Relevanz Wert gelegt.(19)

Die ONC High-priority Liste wurde im Auftrag vom U.S. Office of the National Coordinator for Health Information Technology erstellt.(9) Hierbei wurde besonders Augenmerk auf die klinische Signifikanz der Arzneimittelinteraktionen gelegt. Nur die besonders kritischen Interaktionen wurden in diese Liste aufgenommen.

### **2.1.2.3 E-prescribing und alert fatigue**

In Zeiten der elektronischen Gesundheitsakte und Einführung von Krankenhausinformationssystemen ist auch die elektronische Verschreibung von Medikamenten immer mehr im klinischen Alltag etabliert. Dabei kann die behandelnde Person die gewünschten zu verschreibenden Medikamente direkt aus einer Arzneimittelliste auswählen und direkt auf ein Rezept drucken lassen. Dadurch können Fehler beim Rezeptieren minimiert, kostengünstigere Medikamente des gleichen Wirkstoffs (Generika) verschrieben und Arzneimittel-Interaktionen sofort überprüft werden. Die meisten Systeme verwenden kommerziell verfügbare Datenbanken von Arzneimittelinteraktionen. Aufgrund zu vieler Warnungen dieser Systeme werden diese häufig nicht mehr beachtet, wodurch eine sogenannte

„alert fatigue“ entsteht. Studien zeigen eine Berücksichtigung der Warnungen von Ärzten und Ärztinnen zwischen 33% und 96%. (20,21) Eine Möglichkeit, um dieser „alert fatigue“ entgegenzuwirken wird in der Reduktion an Warnungen gesehen. Die meisten Anbieter von DDI-Listen verwenden den Grad und Schwere der Evidenz als Parameter. DDI-Listen, welche nur klinisch hoch-signifikante Arzneimittelinteraktionen beinhalten, müssen jedoch von Experten gemeinsam entwickelt werden, was sehr ressourcenintensiv und teuer ist. (22) Selbst diese DDI-Listen sind in ihren Inhalten sehr unterschiedlich und haben oft nur wenig Interaktionen gemein. (23,24)

Auch in vielen anderen Bereichen der Medizin ist eine „alert fatigue“ durch zunehmenden Geräteeinsatz zu beobachten (zB Intensivstation, Anästhesie im OP). Die immer genauere Messtechnik der Geräte erkennt immer besser Abweichungen der Bio-Daten. Dabei werden zB bei eine EKG-Ableitung schon kleinste Bewegungen (von Patienten, aber auch eine Manipulation vom Chirurgen) als Abweichung erkannt und deshalb ein Alarm ausgegeben wird. Bei übermäßigem Vorkommen wird deshalb der Alarm deaktiviert und so eventuell ein Herzkammerflimmern nicht sofort erkannt.

## 2.2 Fragestellung

### 2.2.1 Pharmakogenetik

Wie oft wurden Medikamente in Österreich verschrieben, wofür pharmakogenetische Leitlinien der DPWG publiziert wurden?

### 2.2.2 Arzneimittel-Interaktionen

Wie oft wurden Medikamente nach den verschiedenen Interaktionslisten falsch gemeinsam verschrieben? Wie groß ist der relative Unterschied der Alerts zwischen dem Austria Codex und den hoch-signifikanten Listen (französische Interaktionsliste bzw. ONC high priority Liste)?

## 2.3 Zielsetzung

Diese Diplomarbeit soll die potentiellen Auswirkungen von einer österreichweiten Einführung von pre-emptiven pharmakogenetischen Tests darstellen. Desweiteren soll ein kurzer Einblick in die Kosten und eventuelle Kostenvorteile durch vermindern an Spitalsaufenthalten durch unerwünschte Arzneimittelwirkungen dargestellt werden. Außerdem sollen verschiedene

## Einleitung

Arzneimittelinteraktionslisten für E-prescribing-Systeme verglichen und deren potentieller Impact auf Interaktionswarnungen gezeigt werden.



## 3 Material und Methoden

EK-Kennzahl: 1480\_2015

Die selbstentwickelten Programme sowie die kompletten Ergebnisse dieser Diplomarbeit sind auf Github öffentlich zugänglich. Link: [https://github.com/wolkuch/diploma\\_thesis\\_pgx\\_ddi](https://github.com/wolkuch/diploma_thesis_pgx_ddi)

### 3.1 Datenquelle

Als Datenquelle für diese Diplomarbeit wurde die Forschungsdatenbank vom Hauptverband der österreichischen Sozialversicherungsträger GAP-DRG (General Approach for Patient-oriented Outpatient-based Diagnosis Related Groups) verwendet. Diese Datenbank enthält Abrechnungsdaten der einzelnen Sozialversicherungsträger in Österreich von 2006-2007. Der Zugriff auf die Datenbank ist nur mittels Benutzernamen, Passwort und Token über einen verschlüsselten Tunnel (VPN) möglich. Die Daten liegen in einer Datenbank vor und wurden mittels SQL extrahiert. Als ATC-Code-Index wird der amtliche ATC-Index 2007 vom WiDO (Wissenschaftliches Institut der AOK) verwendet. (25)

### 3.2 Pharmakogenetik

Dieser Teil der Arbeit basiert auf einer 2016 publizierten Vorgängerstudie (26), und baut auf in dieser Studie erhobenen Daten auf.

#### 3.2.1 Guidelines und Datenaufbereitung

Für diesen Abschnitt wurden alle Patienten mit Rezepten der Datenbank verwendet, in welchem die Rezepte im Zeitraum 2006-2007 eingelöst wurden. Dabei wurde die Population in vier Altersgruppen eingeteilt:

- < 13 Jahre
- 14 – 39 Jahre
- 40 – 64 Jahre
- $\geq 65$  Jahre

Für die Analyse wurden die Guidelines der DPWG, welche im Jahr 2011 erneuert wurden, verwendet und zwei Listen erstellt. Die erste Liste "Hoch-signifikant" enthielt Drug-Gene Paare, welche der klinischen Signifikanz C-F zugeordnet wurde. Die zweite Liste "DPWG" enthielt alle Drug-Gene Paare, unabhängig ihrer klinischen Signifikanz. Pro Arzneimittel sind

## Material und Methoden

die verschiedene Genotypen/Phänotypen mit unterschiedlicher klinischer Signifikanz. Deshalb wird den Medikamenten immer die jeweils höchste klinische Signifikanz zugewiesen. Die Tabelle 3-1 enthält alle Gene und deren Medikamente, welche in der Liste "Hoch-signifikant" enthalten sind.

**Tabelle 3-1: Alle Drug-Gene Paare in den DPWG Guidelines, welche in der Liste "Hoch-signifikant" aufgenommen wurden**

<b>Gene</b>	<b>Arzneimittel</b>
CYP2D6	Amitriptylin, Aripiprazol, Clomipramin, Codein, Doxepin, Haloperidol, Imipramin, Metoprolol, Nortriptylin, Paroxetin, Propafenon, Risperidon, Tamoxifen, Tramadol, Venlafaxin
CYP2C9	Acenocoumarol, Glimepirid, Phenprocoumon, Phenytoin
CYP2C19	Clopidogrel, Sertralin
UGT1A1	Irinotecan
TPMT	Azathioprin, Mercaptopurin, Thioguanin
HLA-B44	Ribavirin
HLA-B*5701	Abacavir
CYP3A5	Tacrolimus
VKORC1	Phenprocoumon
Faktor V Leiden	Östrogenhaltige Arzneimittel
DPYD	Fluorouracil, Capecitabin

Die meisten Medikamente werden von drei Cytochrom P450 Enzymen (CYPs: CYP2D6, CYP2C9, CYP2C19) metabolisiert. Für diese CYPs wurden eigene Statistiken erstellt.

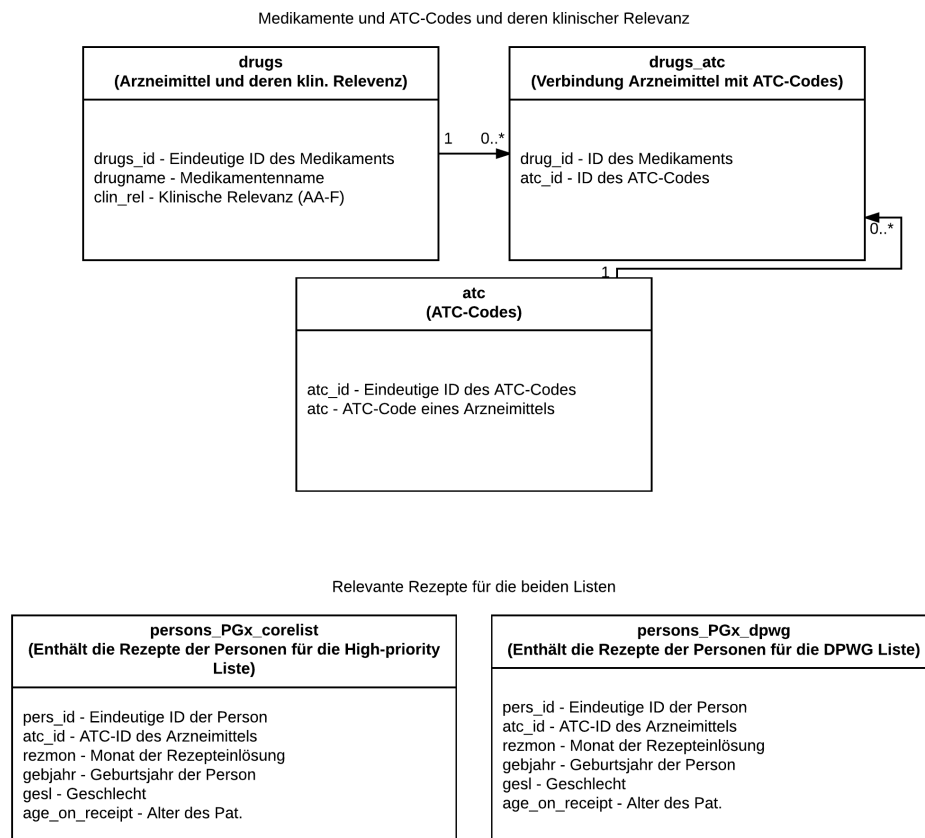
Prodrugs wurden in der Analyse inkludiert, topische Präparate aufgrund mangelnder systemischer Wirkung bei ordnungsgemäßem Gebrauch jedoch ausgeschlossen.

Die Datenbank verwendet ATC-Codes als Kennzeichnung der Arzneimittel. Deshalb wurde ein Mapping von Medikamentennamen und ATC-Codes vorgenommen. (Beispiel: Clopidogrel – B01AC04)

### 3.2.2 Datenbanktabellen

Für die Datenanalyse wurden in der GAP-DRG Datenbank eigene Tabellen erstellt, welche nur die Patientenpopulation mit den notwendigen Informationen für diese Studie enthalten. Abbildung 3.2-1 zeigt die Datenstruktur der Tabellen.

Abbildung 3.2-1: Datenstruktur der Tabellen



Die notwendigen ATC-Codes und Arzneimittel wurden händisch aus dem ATC-Index 2007 in die Datenbanktabelle „atc“ und „drugs“ geschrieben.(25)

## 3.3 Arzneimittel-Interaktionslisten

### 3.3.1 Datenaufbereitung

Die Studienpopulation wurde mit mehreren Kriterien eingeschränkt.

Es wurden nur Personen aus der Forschungspopulation (pop\_forschung = true in der Datenbanktabelle Personen) verwendet.

## Material und Methoden

Für die Überprüfung von Interaktionen zwischen Medikamenten ist ein tagesgenaues Einlösungsdatum der Rezepte notwendig. In der Datenbank GAP-DRG sind jedoch nur von vier Sozialversicherungsträger tagesgenaue Daten vorhanden:

- Gebietskrankenkasse Niederösterreich (NÖGKK)
- Gebietskrankenkasse Kärnten (KGKK)
- Gebietskrankenkasse Salzburg (SGKK)
- Betriebskrankenkasse Neusiedler (BKK Neusiedler)

Um die Bedingung der tagesgenauen Abrechnung sicherzustellen werden Rezepte, in der sich der eingetragene Versicherungsträger mit dem tatsächlich abgerechneten Versicherungsträger unterscheidet, ausgeschlossen. Rezepte, welche keinen ATC-Code, den ungültigen ATC-Code „9999999“ enthalten oder die Länge der ATC-Codes ungleich 7 Zeichen sind, wurden ebenfalls aussortiert.

Für die richtige Funktion des Programmes ist es außerdem wichtig, die Patienten in der richtigen Reihenfolge (aufsteigende pers\_id) zu laden. Die Daten wurden mittels SQL aus der Datenbank extrahiert und als CSV-Datei (comma separated values) gespeichert, um diese in einem eigens entwickeltem Programm auswerten zu können. Abbildung 3.3-1 zeigt das verwendete SQL-Statement:

Abbildung 3.3-1: SQL-Statement zur Extrahierung der notwendigen Daten aus der Datenbank

```
SELECT pers_id, atc, eindat, ddd_float, voanz FROM  
  
(SELECT * FROM personen INNER JOIN rezeptdaten ON personen.pers_id =  
rezeptdaten.id_person_p WHERE personen.pop_forschung = true AND  
rezeptdaten.vtr = rezeptdaten.abrvtr) as a INNER JOIN heilmittel ON a.pharnr =  
heilmittel.pharnr  
  
WHERE vtr IN (12, 16, 17, 24) AND atc IS NOT NULL AND atc != '9999999' AND  
eindat IS NOT NULL AND length(atc) == 7 ORDER BY pers_id ASC
```

Die endgültige Datei enthält insgesamt 39.347.610 Rezepte. Tabelle 3-2 zeigt die Daten, welche in einem Rezept für das Programm enthalten ist.

**Tabelle 3-2: Aufbau eines Rezeptes für die Auswertung**

Attribute	Beschreibung
pers_id	Patienten-ID
atc	ATC-Code des Medikaments, welches verschrieben wurde
eindat	Datum der Einlösung in der Apotheke Format: yyyy-MM-dd
ddd_float	Reichweiteninformation; hypothetische Anzahl der Tage, in der der Arzneistoff eingenommen wird (pro Packung)
voanz	Anzahl der abgegebenen Packungen

Die Reichweiteninformation (ddd\_float) wird aus der Defined Daily Dose (DDD) berechnet. Dieser ist ein standardisierter Wert, welcher von der World Health Organization (WHO) als „angenommene mittlere Erhaltungsdosis pro Tag für die Hauptindikation des Arzneistoffs bei Erwachsenen“ definiert wird. (27) Für etwa ein Viertel der Rezepte in diesem Datensatz sind keine Reichweiteninformation eingetragen. In diesem Fall wird eine Reichweite von 30 Tagen angenommen, welches als durchschnittliche Einnahmedauer der Medikamente in den Daten ermittelt wurde. Bei fehlender Angabe der Packungsanzahl wurde eine Packung angenommen.

### 3.3.2 Aufbereitung der Interaktionslisten

Das Auswertungsprogramm benötigt für die Auswertung eine Liste, welche ein Interaktionspaar pro Zeile enthält und zwei ATC-Codes mittels Doppelpunkt trennt (ATC-Code 1: ATC-Code 2). Da die Interaktionslisten der verschiedenen Institutionen in verschiedenen Formaten aufliegen, müssen die Daten noch ins richtige Format gebracht werden.

#### 3.3.2.1 Französische Interaktionsliste

Die französische Interaktionsliste wurde von NancyVanNess ins Englische übersetzt und als XML-Dateien maschinenlesbar über WorldVista zur Verfügung gestellt.(19)(28) Zwei XML-Dateien sind hierbei relevant. Die Datei *all\_eng\_groups2015.xml* enthält die Arzneimittelgruppen mit ihren dazugehörigen Wirkstoffen und deren ATC-Codes. Die Datei *all\_eng\_tables2015.xml* enthält die eigentlichen Interaktionen. Aus diesen beiden Dateien wurde mittels einem Java-Programms die Interaktionsliste mit den Interaktionspaaren (ATC-Code 1 : ATC-Code 2) erstellt.

### **3.3.2.2 *Austria Codex Interaktionsliste***

Die Interaktionsliste wurde aus einer Datenbank, welche die Interaktionen im richtigen Format (ATC-Code 1 : ATC-Code 2) enthielt, generiert. Diese wurde mit freundlichem Dank an die Autoren einer anderen Studie über Arzneimittelinteraktionen (Estimation of Severe Drug-Drug Interaction Warnings by Medical Specialist Groups for Austrian Nationwide eMedication) für diese Diplomarbeit überlassen. (29)

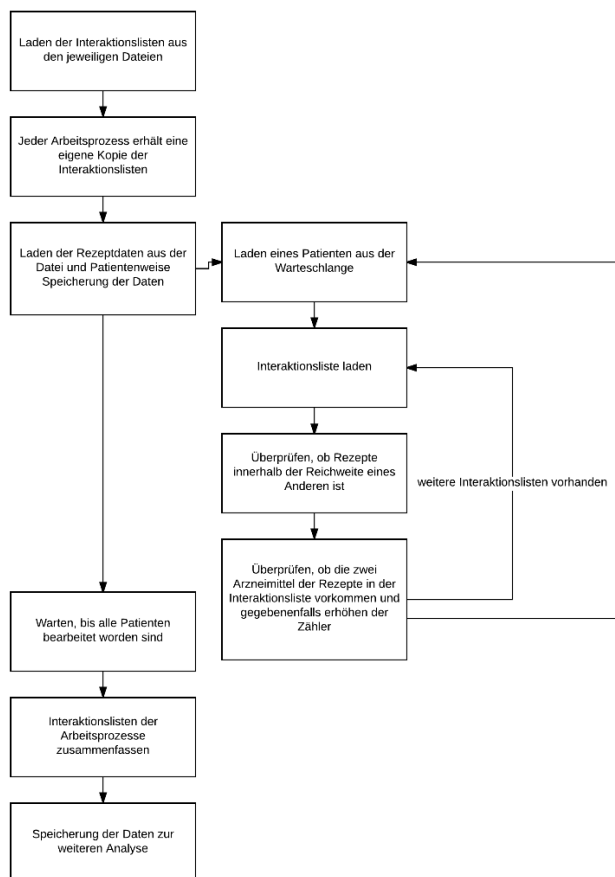
### **3.3.2.3 *ONC high-priority Interaktionsliste***

In Anlehnung an die französische Interaktionsliste wurde selbstständig aus dem Paper die Interaktionen in eine XML-Datei eingetragen. (19) Eine zweite XML-Datei enthielt die passenden Medikamente und deren ATC-Codes. Die Gruppe „QT prolonging drugs“ wurde aus einem Excel-Dokument von CredibleMeds extrahiert. (30) Aus diesen Dateien wurde mittels eines Java-Programms die endgültige Interaktionsliste erzeugt.

## **3.3.3 Berechnung der Interaktionen**

Das Interaktions-Programm wurde mit der Programmiersprache Java entwickelt. Dieses Programm berechnet die Alerts, die ein Interaktions-Warnsystem auslösen würde. Dabei wird auch die Reichweiteninformation der Rezepte verwendet. Kommt es zu einer Überschneidung zweier Rezepte, wird auf eine Interaktion geprüft und gegebenenfalls ein Alertzähler erhöht. Um die anfallenden Daten schneller bearbeiten zu können, wurden mehrere parallel arbeitende Arbeitsprozesse verwendet. Abbildung 3.3-2 zeigt den Workflow des beschriebenen Auswertungsprogramms.

Abbildung 3.3-2: Workflow des Interaktions-Programms



Die Interaktionspaare mit deren jeweiligen Alerts werden als Objekte in eine Datei gespeichert. Diese Datei enthält die Interaktionslisten mit den dazugehörigen Interaktionspaaren und deren Anzahl an Alerts. Im Anhang ist der Quellcode dieses Programms angehängt.

### 3.3.4 Auswertung der Interaktionen

Die Auswertung der Interaktions-Daten erfolgte mittels eines weiteren Java-Programms, welches die geforderten Ergebnisse als CSV-Dateien (Comma Separated Values) speichert. Hier ist es möglich, die Interaktionsdaten mit Angabe von zwei ATC-Codes auf deren Häufigkeit von Interaktionen zu überprüfen. Außerdem wurden die statistischen Auswertungen von einem weiteren Programm erzeugt. Abbildung 3.3-3: Eingabemaske des

## Material und Methoden

Interaktionsprogramms für spezielle Auswertungen. Abbildung 3.3-3 und Abbildung 3.3-4 zeigt ein Beispiel der Programm-Maske und der Ergebnis-Datei.

Abbildung 3.3-3: Eingabemaske des Interaktionsprogramms für spezielle Auswertungen

```
Bitte Quelle angeben : interaktions_paare_objects_new_sameday.data
Daten geladen

Ersten ATC-Code eingeben: C10AA
Zweiten ATC-Code eingeben: J01FA
Ergebnisse in der Datei "output.txt" erfolgreich gespeichert
```

Abbildung 3.3-4: Ausschnitt der Ergebnis-Datei

```
interaction_austria.csv
C10AA01:J01FA09;1447
C10AA03:J01FA09;367
C10AA05:J01FA09;305
C10AA01:J01FA01;15
C10AA02:J01FA09;15
C10AA05:J01FA01;5
C10AA03:J01FA01;2
interaction_onc_high.csv
C10AA01:J01FA09;1447
C10AA01:J01FA01;15
C10AA02:J01FA09;15
C10AA01:J01FA15;2
interaction_french.csv
J01FA09:C10AA01;1447
```

### 3.3.5 Statistik

Für jede Interaktionsliste wurden die Anzahl der Alerts berechnet. Ein Alert wurde dann generiert, wenn ein Patient zwei Medikamente erhält, welche laut Interaktionsliste nicht kombiniert werden sollte. Dabei wurden zwei verschiedene Arten unterschieden:

- Alerts **mit** Mehrfach-Alerts: Alerts, wobei der mehrfache Verstoß gegen ein Interaktionspaar auch mehrfach gerechnet wird
- Alerts **ohne** Mehrfach-Alerts: Alerts, wobei nur der erste Verstoß gegen ein Interaktionspaar auch als Alert gerechnet wird

Für jede Liste wurden verschiedene Gruppen berechnet:

- **häufigste Medikamentengruppen:** die Medikamente werden nach der dritten Unterstufe des ATC-Codes (chemische/therapeutische/pharmakologische Subgruppe zB Antidepressiva N06A als Untergruppe der Psychoanaleptika N06)



## Material und Methoden

- **häufigste Interaktionspaare:** die Alerts der einzelnen Interaktionspaare werden hier berechnet
- **häufigste Medikamente:** Die Gesamt-Alerts der einzelnen Medikamente in den verschiedenen Interaktionspaaren werden hier berechnet.

In diesen Gruppen wurden die Alerts **ohne** Mehrfach-Alerts verwendet.

Für die Austria Codex-Liste wurde außerdem eine Liste erstellt, welche nur Interaktionen mit schwerer klinischen Relevanz enthält. Für die ONC High Priority Liste wurde dieselbe Aufteilung wie in der Studie verwendet und insgesamt 15 Unterlisten erstellt und berechnet.

Aufgrund der Limitation der Daten, dass die Reichweiteninformation nicht mit der tatsächlichen Einnahmedauer übereinstimmen muss, werden in einer eigenen Auswertung die Reichweiteninformationen der Rezepte ignoriert. Dabei werden nur Rezepte, welche am selben Tag eingelöst worden sind, auf ihre Interaktionen überprüft.

## 4 Resultate

### 4.1 Pharmakogenetik

Als Studienpopulation wurden alle Personen inkludiert, welche innerhalb der Jahre 2006 und 2007 mindestens ein Rezept eingelöst hatten. Die Datenbank enthält 6.831.128 Personen, wobei Patienten mit unbekanntem Alter ausgeschlossen wurden. Insgesamt 6.761.034 Personen (52% weiblich) wurden in die Studienpopulation eingeschlossen. Dabei zeigte sich bei Personen  $\geq 65$  Jahre eine hohe Verbreitung von mindestens einem PGx-Medikaments von 72% in der DPWG-Liste und 47,2% in der „Hoch-signifikant“-Liste. Tabelle 4-1 zeigt weitere Details.

**Tabelle 4-1: Anzahl der Patienten (absolut und relativ zur Altersgruppen) mit mindestens einem PGx Medikament von den Listen DPWG und "Hoch-signifikant"; gruppiert nach den Altersgruppen**

	<b>0 – 13 Jahre</b>	<b>14 – 39 Jahre</b>	<b>40 – 64 Jahre</b>	<b><math>\geq 65</math> Jahre</b>
<b>DPWG-Liste</b>	39.557 (4.6%)	547.441 (25.7%)	1.190.242 (51.1%)	1.037.127 (72.0%)
<b>„Hoch-signifikant“- Liste</b>	33.483 (3.9%)	178.809 (8.4%)	599.811 (25.7%)	680.052 (47.2%)

#### 4.1.1 DPWG-Liste

41,63% der Studienpopulation (n = 2.814.367 Personen) erhielten mindestens ein Medikament der DPWG-Liste. Dabei erhielten insgesamt 1.451.616 Personen bzw. 41,63% der Studienpopulation mindestens ein und 20,16% mindestens zwei PGx-Medikamente. Außerdem erhielten 48,4% der Personen, welche mindestens ein PGx-Medikament erhalten, ein zweites zusätzlich (reaktiv-pre-emptiv, siehe S.8.). Tabelle 4-2 zeigt weitere Details.

**Tabelle 4-2: Anzahl der PGx-Medikamente der DPWG-Liste mit der Anzahl der Personen mit der relativen Anzahl von Personen, welche die Anzahl der Medikamente erhalten**

<b>Anzahl der PGx-Medikamente</b>	<b>Anzahl der Personen</b>	<b>% von n (reaktiv-pre-emptiv)</b>	<b>% der Studienpopulation (pre-emptiv)</b>
1	1.451.616	100,0%	41,6%
2	690.193	48,4%	20,2%
3	355.045	23,9%	10,0%
4	174.540	11,3%	4,7%

## Resultate

Besonders in der Altersgruppe  $\geq 65$  zeigt sich eine hohe Verwendung von PGx-Medikamenten. Insgesamt 1.037.127 (72,04% der Personen  $\geq 65$ ) erhielten mindestens ein PGx-Medikament. 62,39% derjenigen Personen, welche mindestens ein PGx-Medikament erhielten, erhielten auch ein Zweites. Die häufigsten verschriebenen Medikamente hierbei sind Protonenpumpenhemmer wie Pantoprazol, Lansoprazol oder Omeprazol, sowie das schwache Opioid Tramadol. Tabelle 4-3 zeigt die 5 häufigsten verschriebenen Medikamente.

**Tabelle 4-3: Häufigste verschriebene Medikamente der Altersgruppe  $\geq 65$  der DPWG-Liste und relative Anzahl dieser Altersgruppenpopulation. In Klammer die relative Anzahl der Personen, die mind. ein DPWG-Medikament erhalten.**

	Arzneimittel	Personen	% von dieser Altersgruppenpopulation (% der DPWG-Personen)
<b>Rang 1</b>	Pantoprazol	405.553	28,2% (39,1%)
<b>Rang 2</b>	Tramadol	218.593	15,2% (21,1%)
<b>Rang 3</b>	Lansoprazol	213.741	14,9% (20,6%)
<b>Rang 4</b>	Omeprazol	167.148	11,6% (16,1%)
<b>Rang 5</b>	Metoprolol	152.000	10,6% (14,7%)

### 4.1.2 „Hoch-signifikant“-Liste

22,07% der Studienpopulation (n = 1.492.154 Personen) erhielten mindestens ein Medikament der „Hoch-signifikant“-Liste. Dabei erhielten insgesamt 1.060.860 Personen bzw. 22,07% der Studienpopulation mindestens ein und 6,38% mindestens zwei PGx-Medikamente. Tabelle 4-4 zeigt weitere Details.

**Tabelle 4-4: Anzahl der PGx-Medikamente der „Hoch-signifikant“-Liste mit der Anzahl der Personen mit der relativen Anzahl von Personen, welche die Anzahl der Medikamente erhalten**

Anzahl der PGx-Medikamente	Anzahl der Personen	% von n (reaktiv-pre-emptiv)	% der Studienpopulation (pre-emptiv)
1	1.060.860	100,0%	22,1%
2	313.744	28,9%	6,4%
3	88.731	7,9%	1,7%
4	22.272	1,9%	0,4%

## Resultate

Insgesamt 680.052 Personen (47,24% der Alterspopulation) in der Altersgruppe  $\geq 65$  erhielten mindestens ein PGx-Medikament der „Hoch-signifikant“-Gruppe. Tabelle 4-5 zeigt die fünf häufigsten Medikamente und Anzahl der Personen.

**Tabelle 4-5: Häufigste verschriebene Medikamente der Altersgruppe  $\geq 65$  der „Hoch-signifikant“-Liste und relative Anzahl dieser Altersgruppenpopulation. In Klammer die relative Anzahl der Personen, die mind. ein „Hoch-signifikant“-Medikament erhalten.**

	Arzneimittel	Personen	% von dieser Altersgruppenpopulation (% von den „Hoch-sig.“-Personen mit min 1 PGx)
<b>Rang 1</b>	Tramadol	218.593	15,2% (32,1%)
<b>Rang 2</b>	Metoprolol	152.000	10,6% (22,4%)
<b>Rang 3</b>	Östrogenhaltige Arzneimittel	134.388	9,3% (19,8%)
<b>Rang 4</b>	Clopidogrel	72.884	5,1% (10,7%)
<b>Rang 5</b>	Sertralin	53.831	3,7% (7,9%)

Insgesamt knapp 600.000 Personen (25,7% der Alterspopulation) in der Altersgruppe 40-64 erhielten mindestens ein PGx-Medikament. Tabelle 4-6 zeigt ebenfalls die fünf häufigsten Medikamente und die Anzahl der Personen.

**Tabelle 4-6: Häufigste verschriebene Medikamente der Altersgruppe 40-64 der „Hoch-signifikant“-Liste und relative Anzahl dieser Altersgruppenpopulation. In Klammer die relative Anzahl der Personen, die mind. ein „Hoch-signifikant“-Medikament erhalten.**

	Arzneimittel	Personen	% von dieser Altersgruppenpopulation (% von den „Hoch-sig.“-Personen mit min 1 PGx)
<b>Rang 1</b>	Östrogenhaltige Medikamente	201.999	8,7% (33,7%)
<b>Rang 2</b>	Tramadol	154.298	6,6% (25,7%)
<b>Rang 3</b>	Metoprolol	109.205	4,7% (18,2%)
<b>Rang 4</b>	Sertralin	58.314	2,5% (9,72%)
<b>Rang 5</b>	Paroxetin	42.162	1,8% (7,0%)

### 4.1.3 DPWG-Liste vs „Hoch-signifikant“-Liste

Ungefähr ein Viertel (22,07%) bzw. 41,63% der Studienpopulation erhielt mindestens ein Arzneimittel von der „Hoch-signifikant“- bzw. „DPWG“-Liste. Die häufigsten drei Medikamente waren hierbei die Protonenpumpeninhibitoren Pantoprazol, Lansoprazol und

## Resultate

Omeprazol in der DPWG-Liste. In der „Hoch-signifikant“-Liste waren Tramadol, Östrogenhaltige Arzneistoffe und Metoprolol am häufigsten. Tabelle 4-7 zeigt den Vergleich der häufigsten Medikamente der beiden Listen.

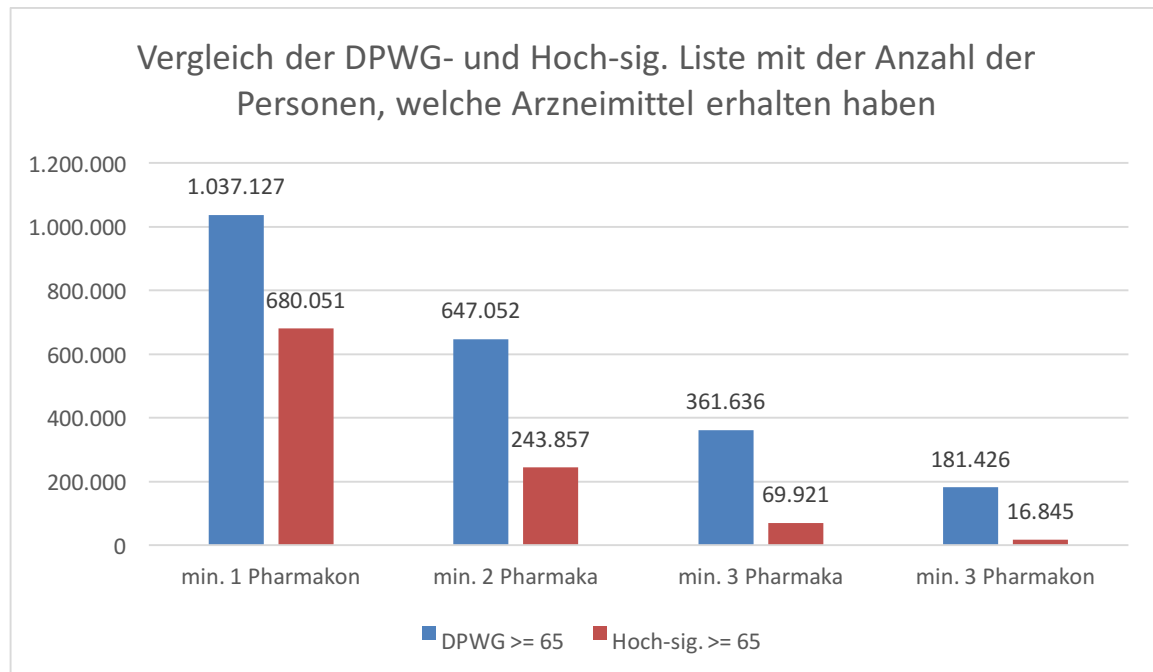
**Tabelle 4-7 : Die zehn häufigsten verschriebenen Arzneimittel von der DPWG- und "Hoch-signifikant"-Liste und die % der inkludierten Patienten, welche das Arzneimittel in dieser Studienperiode erhielten**

	<b>DPWG</b>	<b>Hoch-signifikant</b>
<b>Rang 1</b>	Pantoprazol (15,1%)	Tramadol (6,1%)
<b>Rang 2</b>	Lansoprazol (8,6%)	Östrogenhaltige Arzneistoffe (5,9%)
<b>Rang 3</b>	Omeprazol (6,8%)	Metoprolol (4,1%)
<b>Rang 4</b>	Tramadol (6,1%)	Sertralin (2,1%)
<b>Rang 5</b>	Östrogenhaltige Arzneistoffe (5,9%)	Phenprocoumon (1,8%)
<b>Rang 6</b>	Metoprolol (4,1%)	Clopidogrel (1,5%)
<b>Rang 7</b>	Esomeprazol (3,9%)	Paroxetin (1,4%)
<b>Rang 8</b>	Citalopram (3,3%)	Glimepirid (1,1%)
<b>Rang 9</b>	Rabeprazol (2,7%)	Amitriptylin (1,1%)
<b>Rang 10</b>	Escitalopram (2,6%)	Risperidon (1,1%)

In der „Hoch-signifikant“-Liste erhalten etwa die Hälfte der Personen über 65 Jahre (47,2%) mindestens ein PGx-Medikament. Im Vergleich dazu erhalten etwa Dreiviertel der Personen (72,0%) mindestens ein PGx-Medikament der DPWG-Liste. Abbildung 4.1-1 zeigt den Vergleich der beiden Listen der Personen älter als 65 Jahre, geordnet nach der Anzahl der erhaltenen verschiedenen PGx-Medikamente.

## Resultate

**Abbildung 4.1-1: Vergleich der DPWG- und "Hoch-signifikant"-Liste mit der Anzahl der Personen älter als  $\geq 65$ , welche Arzneimittel erhalten haben.**



### 4.1.4 Häufigste drei CYPs einzeln und gemeinsam („Hoch-signifikant“-Liste)

In der „Hoch-signifikant“-Liste verwenden 17,58% der Gesamtpopulation Arzneimittel, welche vom einem der drei CYPs CYP2D6, CYP2C9 oder CYP2C19 metabolisiert werden. In der Altersgruppe  $\geq 65$  steigt dieser Anteil auf 41,54% dieser Altersgruppe. Tabelle 4-8 zeigt weitere Details.

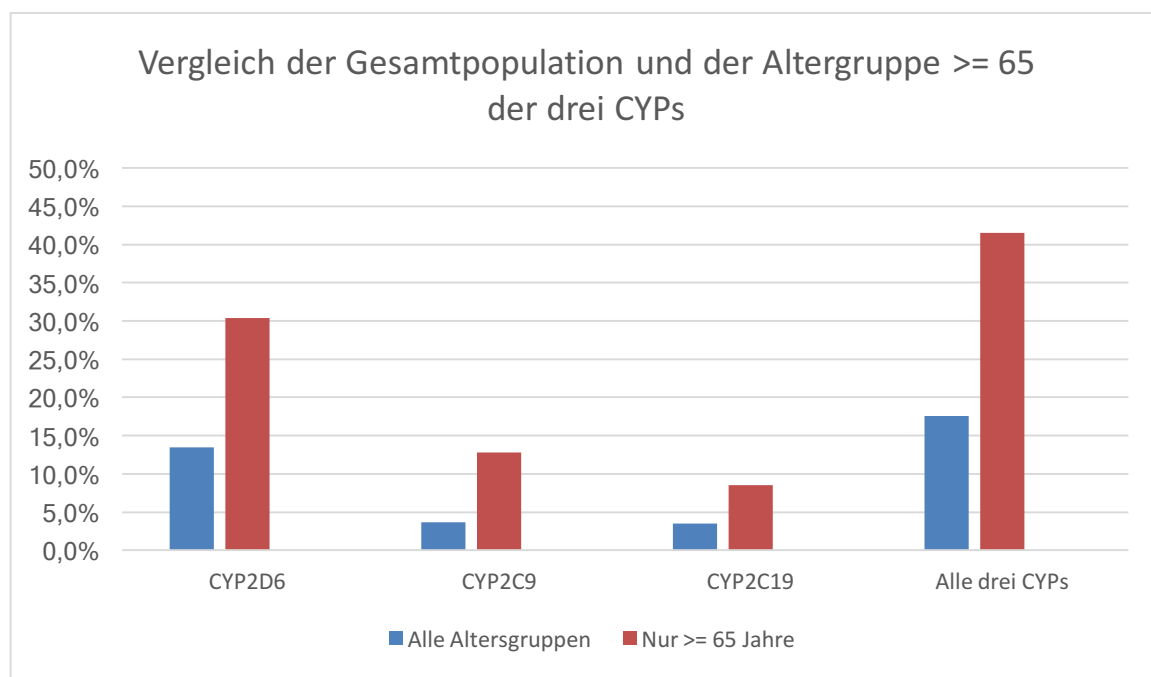
**Tabelle 4-8: Prozent der Gesamtpopulation in der "Hoch-signifikant"-Gruppe von allen Altersgruppen und der Altersgruppe  $\geq 65$**

CYP-Enzym	% der Gesamtpopulation in der „Hoch-signifikant“-Liste von allen Altersgruppen	% der Gesamtpopulation in der „Hoch-signifikant“-Liste von der Altersgruppe $\geq 65$
CYP2D6	13,5%	30,4%
CYP2C9	3,7%	12,8%
CYP2C19	3,5%	8,5%
Alle drei CYPs	17,6%	41,5%

Abbildung 4.1-2 vergleicht die relative Anzahl der Personen, welche mindestens ein die verschiedenen CYPs der Altersgruppe  $\geq 65$  mit allen vier Altersgruppen zusammen.

## Resultate

**Abbildung 4.1-2: Vergleich der Gesamtpopulation und der Altersgruppe  $\geq 65$  mit der "Hoch-signifikant"-Liste und drei CYPs**



### 4.1.4.1 Alle Altersgruppen

In der „Hoch-signifikant“-Gruppe erhielten 13,48% der Gesamtpopulation mindestens ein Medikament, welches von CYP2D6 metabolisiert wird. Medikamente der beiden CYPs CYP2C9 und CYP2C19 wurden von 3,67% bzw. 3,52% verwendet. Bei 61,1% der Personen, welche min. ein PGx-Medikament der „Hoch-signifikant“-Gruppe erhalten haben, erhielten eines von der CYP2D6-Gruppe, 79,67% erhielten eines von den drei CYPs-Medikamentengruppen. Tabelle 4-9 zeigt die Medikamente der einzelnen CYPs und deren Anzahl an Personen, welche dieses Medikament erhalten haben.

**Tabelle 4-9: die drei CYPs mit deren Medikamenten und Anzahl der Personen, welche es min. einmal erhalten haben**

	Medikament	Personen	% der Gesamt-population	% der Pers. in dieser Gruppe, welche min. 1 PGx erhielten
<b>CYP2D6</b>				
	Tramadol	414.846	6,1%	27,8%
	Metoprolol	273.673	4,1%	18,3%
	Paroxetin	91.403	1,4%	6,1%
	Amitriptylin	74.721	1,1%	5,0%
	Risperidon	70.767	1,1%	4,7%
	Venlafaxin	62.444	0,9%	4,2%

## Resultate

	Codein	37.404	0,6%	2,5%
	Tamoxifen	14.255	0,2%	1,0%
	Haloperidol	13.639	0,2%	0,9%
	Aripiprazol	9.397	0,1%	0,6%
	Propafenon	8.222	0,1%	0,5%
	Clomipramin	6.610	0,1%	0,4%
	Doxepin	4.356	0,1%	0,3%
	Nortriptylin	328	0,0%	0,0%
	<b>Gesamt</b>	911.223	13,48%	61,1%
<b>CYP2C9</b>				
	Phenprocoumon	123.979	1,8%	8,3%
	Glimepirid	76.605	1,1%	5,1%
	Acenocoumarol	56.266	0,8%	3,8%
	Phenytoin	1.614	0,1%	0,1%
	<b>Gesamt</b>	248.396	3,5%	16,7%
<b>CYP2C19</b>				
	Sertralin	142.522	2,1%	9,6%
	Clopidogrel	102.054	1,5%	6,9%
	<b>Gesamt</b>	238.104	3,5%	16,0%

Im Vergleich erhielten 93,9% der Personen, welche mindestens ein PGx-Medikament der DPWG-Liste erhielten, ein diesen drei CYPs zugeordnetes Medikament.

### 4.1.4.2 Altersgruppe $\geq 65$

In der „Hoch-signifikant“-Gruppe erhielten 30,38% der Gesamtpopulation mindestens ein Medikament, welches von CYP2D6 metabolisiert wird. Medikamente der beiden CYPs CYP2C9 und CYP2C19 wurden von 12,75% bzw. 8,46% verwendet. Bei 64,31% der Personen, welche min. ein PGx-Medikament der „Hoch-signifikant“-Gruppe erhalten haben, erhielten eines von der CYP2D6-Gruppe, 87,92% erhielten mindestens eines von den drei CYP-Medikamentengruppen. Tabelle 4-10 zeigt die Medikamente der einzelnen CYPs und deren Anzahl an Personen, welche dieses Medikament erhalten haben.



## Resultate

**Tabelle 4-10: Die drei CYPs mit deren zugeordneten Medikamenten und Anzahl der Personen  $\geq 65$  Jahre, welche das jeweilige Medikament min. einmal erhalten haben**

	<b>Medikament</b>	<b>Personen</b>	<b>% der Altersgruppenpopulation</b>	<b>% der Pers. in dieser Gruppe, welche min. 1 PGx erhielten</b>
<b>CYP2D6</b>				
	Tramadol	218.593	15,2%	32,1%
	Metoprolol	152.000	10,6%	22,4%
	Risperidon	46.881	3,3%	6,9%
	Amitriptylin	31.646	2,2%	4,7%
	Paroxetin	31.326	2,2%	4,6%
	Venlafaxin	18.989	1,3%	2,8%
	Haloperidol	7.624	0,5%	1,1%
	Tamoxifen	7.426	0,5%	1,1%
	Codein	6.116	0,4%	0,9%
	Propafenon	5.521	0,4%	0,8%
	Clomipramin	2.605	0,2%	0,4%
	Doxepin	1.601	0,1%	0,2%
	Aripiprazol	989	0,1%	0,2%
	Nortriptylin	92	0,0%	0,0%
	<b>Gesamt</b>	437.310	30,38%	64,3%
<b>CYP2C9</b>				
	Phenprocoumon	95.782	6,7%	14,1%
	Glimepirid	52.146	3,6%	7,7%
	Acenocoumarol	43.306	3,0%	6,4%
	Phenytoin	779	0,1%	0,1%
	<b>Gesamt</b>	183.500	12,8%	27,0%
<b>CYP2C19</b>				
	Clopidogrel	72.884	5,1%	10,7%
	Sertralin	53.831	3,7%	8,0%
	<b>Gesamt</b>	121.832	8,5%	17,9%

## 4.2 Arzneimittel-Interaktionslisten

Insgesamt 1.587.829 Patienten wurden in die Studienpopulation eingeschlossen. Diese hatten insgesamt 39.347.610 Rezepte. Dabei waren jedoch 8.725.198 Rezepte ohne Reichweiteninformation. Tabelle 4-11 zeigt die Gesamtübersicht der Interaktionslisten und den Alerts.

**Tabelle 4-11: Gesamtübersicht der Interaktionslisten (Alerts ohne Mehrfachalerts + Alerts mit Mehrfachalerts)**

	<b>Alerts mit Mehrfachalerts</b>	<b>Alerts ohne Mehrfachalerts</b>
franz. Interaktionsliste	9.694.700	1.079.614
Austria Codex Interaktionsliste	19.262.610	2.959.883
ONC High-Priority Interaktionsliste	189.924	53.336

### 4.2.1 Interaktionsberechnung mit Reichweiteninformation

#### 4.2.1.1 Französische Interaktionsliste

Die französische Interaktionsliste enthält 46.179 verschiedene Interaktionspaare, welche von 876 verschiedenen ATC-Codes gebildet werden. Tabelle 4-12 zeigt die zehn häufigsten Interaktionspaare und die Anzahl der Warnungen.

**Tabelle 4-12: häufigste Interaktionspaare der franz. Interaktionsliste**

<b>Häufigste Interaktionspaare</b>	<b>ATC-Codes</b>	<b>Alerts</b>
Amlodipin : Bisoprolol	C08CA01:C07AB07	9.963
Estriol : Levothyroxin-Natrium	G03CA04:H03AA01	9.224
Mefenaminsäure : Bisoprolol	M01AG01:C07AB07	8.796
Amlodipin : Carvedilol	C08CA01:C07AG02	8.405
Amlodipin : Metoprolol	C08CA01:C07AB02	7.470
Citalopram : Mefenaminsäure	N06AB04:M01AG01	6.550
Furosemid : Lisinopril	C03CA01:C09AA03	6.531
Sprionolacton : Furosemid	C03DA01:C03CA01	6.176
Tetrazepam : Tramadol	M03BX07:N02AX02	6.163
Enoxaparin : Mefenaminsäure	B01AB05:M01AG01	6.064

## Resultate

Die häufigsten Medikamente, welche einen Alert ausgelöst haben sind in Tabelle 4-13 zusammengefasst. Furosemid und Mefenaminsäure waren oft in Interaktionen mit über 100.000 Warnungen beteiligt.

**Tabelle 4-13: häufigste Medikamente in der franz. Interaktionsliste, welche Alerts ausgelöst hat**

Häufigste Medikamente	ATC-Code	Alerts
Furosemid	C03CA01	102.495
Mefenaminsäure	M01AG01	101.892
Bisoprolol	C07AB07	83.599
Citalopram	N06AB04	61.886
Amlodipin	C08CA01	61.856
Tramadol	N02AX02	61.055
Lisinopril	C09AA03	55.688
Dexibuprofen	M01AE14	50.443
Escitalopram	N06AB10	50.384
Metoprolol	C07AB02	49.610

Dabei zeigte sich, dass Furosemid, besonders mit Medikamenten der ATC-Gruppe *ACE-Hemmer, rein*, Mefenaminsäure besonders mit Medikamenten der ATC-Gruppe *Antidepressiva* oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten in Alerts involvierte Medikamentengruppen sind:

- Antidepressiva (N06A): 262.856 Alerts
- Beta-Adrenorezeptor-Antagonisten (C07A): 226.476 Alerts
- Nichtsteroidale Antiphlogistika und Antirheumatika (M01A): 202.749 Alerts

### 4.2.1.2 Austria Codex Interaktionsliste

Die Austria Codex Interaktionsliste enthält 38.092 verschiedene Interaktionspaare, welche von 1.220 verschiedenen ATC-Codes gebildet werden. Tabelle 4-14 zeigt die zehn häufigsten Interaktionspaare und die Anzahl der Warnungen.

**Tabelle 4-14: häufigste Interaktionspaare der Austria Codex Interaktionsliste**

Häufigste Interaktionspaare	ATC-Codes	Alerts
Betamethason : Diclofenac	H02AB01:M01AB05	14.370
Bisoprolol : Diclofenac	C07AB07:M01AB05	13.638

## Resultate

Metformin : Diclofenac	A10BA02:M01AB05	12.461
Furosemid : Diclofenac	C03CA01:M01AB05	11.542
Lisinopril : Lisinopril und Diuretika	C09AA03:C09BA03	11.384
Enalapril : Enalapril und Diuretika	C09AA02:C09BA02	10.947
Lisinopril und Diuretika : Diclofenac	C09BA03:M01AB05	10.560
Bisoprolol : Amlodipin	C07AB07:C08CA01	9.963
Enalapril und Diuretika : Diclofenac	C09BA02:M01AB05	9.623
Lisinopril : Diclofenac	C09AA03:M01AB05	9.335

Die häufigsten Medikamente, welche einen Alert ausgelöst haben, sind in Tabelle 4-15 zusammengefasst. Diclofenac und Ranitidin sind hier oft in Interaktionswarnungen vorgekommen.

**Tabelle 4-15: häufigste Medikamente der Austria Codex Interaktionsliste**

Häufigste Medikamente	ATC-Codes	Alerts
Diclofenac	M01AB05	287.677
Ranitidin	A10BA02	149.339
Mefenaminsäure	M01AG01	136.696
Furosemid	C03CA01	136.243
Diclofenac, Kombinationen	M01AB55	126.497
Acetylsalicylsäure	B01AC06	123.279
Lisinopril und Diuretika	C09BA03	120.266
Bisoprolol	C07AB07	115.952
Phenprocoumon	B01AA04	111.710
Enalapril und Diuretika	C09BA02	104.373

Dabei zeigte sich, dass Diclofenac, besonders mit Medikamenten der ATC-Gruppe *Beta-Adrenorezeptor-Antagonist*, Ranitidin besonders mit Medikamenten der ATC-Gruppe *Nicht steroidale Antiphlogistika und Antirheumatika* oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten Medikamentengruppen sind:

Diplomarbeit

## Resultate

- Nichtsteroidale Antiphlogistika und Antirheumatika (M01A): 872.739 Alerts
- Antidepressiva (N06A): 450.752 Alerts
- Antithrombotische Mittel (B01A): 361.115 Alerts

Bei Verwendung von nur **schweren** Interaktionen der Austria Codex Liste enthält diese Liste 3.996 verschiedene Interaktionspaare, welche von 752 verschiedenen ATC-Codes gebildet werden. Insgesamt wurden 131.994 Alerts ausgelöst. Tabelle 4-16 zeigt die zehn häufigsten Interaktionspaare und die Anzahl der Warnungen.

**Tabelle 4-16: häufigste Interaktionspaare der Austria Codex Interaktionsliste (nur schwere Interaktionen)**

Häufigste Interaktionspaare	ATC-Codes	Alerts
Simvastatin : Clarithromycin	C10AA01:J01FA09	5.483
Desloratadin : Clarithromycin	R06AX27:J01FA09	2.551
Trazodon : Prothipendyl	N06AX05:N05AX07	2.278
Mirtazapin : Prothipendyl	N06AX11:N05AX07	2.081
Quetiapin : Trazodon	N05AH04:N06AX05	1.810
Quetiapin : Mirtazapin	N05AH04:N06AX11	1.755
Gliclazid : Phenprocoumon	A10BB09:B01AA04	1.701
Ketoconazol : Acetyldigitoxin	D01AC08:C10AA01	1.634
Trazodon : Melitracen und Psycholeptika	N06AX05:N06CA02	1.547
Metformin : Nifedipin	A10BA02:C08CA05	1.434

Die häufigsten Medikamente, welche einen Alert ausgelöst haben, sind in Tabelle 4-17 zusammengefasst. Clarithromycin und Simvastatin sind hier oft in Interaktionen vorgekommen.

**Tabelle 4-17: häufigste Medikamente der Austria Codex Interaktionsliste (nur schwere Interaktionen)**

Häufigste Medikamente	ATC-Codes	Alerts
Clarithromycin	J01FA09	17.962
Simvastatin	C10AA01	11.057
Mirtazapin	N06AX11	9.809
Prothipendyl	N05AX07	9.790
Quetiapin	N05AH04	9.446
Trazodon	N06AX05	9.159
Melitracen und Psycholeptika	N06CA02	8.891
Ranitidin	A02BA02	6.878

## Resultate

Metformin	A10BA02	6.789
Olanzapin	N05AH03	6.017

Dabei zeigte sich, dass Clarithromycin, besonders mit Medikamenten der ATC-Gruppe *Mittel, die den Lipidstoffwechsel beeinflussen, rein*, Simvastatin besonders mit Medikamenten der ATC-Gruppe *Makrolide, Lincosamide und Streptogramine* oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten Medikamentengruppen sind:

- Antipsychotika (N05A): 43.387 Alerts
- Antidepressiva (N06A): 34.297 Alerts
- Makrolide, Lincosamide und Streptogramine (J01F): 23.656 Alerts

### 4.2.1.3 *ONC high priority Interaktionsliste*

Die ONC high priority Interaktionsliste enthält 2.947 verschiedene Interaktionspaare, welche von 171 verschiedenen ATC-Codes gebildet werden. Tabelle 4-18 zeigt die zehn häufigsten Interaktionspaare und die Anzahl der Warnungen.

**Tabelle 4-18: häufigste Interaktionspaare der ONC high priority Interaktionsliste**

Häufigste Interaktionspaare	ATC-Codes	Alerts
Simvastatin : Clarithromycin	C10AA01:J01FA09	5.483
Citalopram : Clarithromycin	N06AB04:J01FA09	3.374
Ciprofloxacin : Citalopram	J01MA02:N06AB04	2.679
Clarithromycin : Escitalopram	J01FA09:N06AB10	2.657
Ciprofloxacin : Escitalopram	J01MA02:N06AB10	2.135
Simvastatin : Amiodaron	C10AA01:C01BD01	1.909
Clarithromycin : Moxifloxacin	J01FA09:J01MA14	1.792
Citalopram : Escitalopram	N06AB04:N06AB10	1.784
Ciprofloxacin : Clarithromycin	J01MA02:J01FA09	1.428
Azithromycin : Citalopram	J01FA10:N06AB04	1.345

Die häufigsten Medikamente, welche einen Alert ausgelöst haben, sind in Tabelle 4-19 zusammengefasst. Clarithromycin und Citalopram sind hier oft in Interaktionen vorgekommen.

## Resultate

Tabelle 4-19: Häufigste Medikamente, welche einen Alert ausgelöst haben, in der ONC high priority Liste

Häufigste Medikamente	ATC-Codes	Alerts
Clarithromycin	J01FA09	19.474
Citalopram	N06AB04	13.978
Escitalopram	N06AB10	11.866
Demeclocyclin	J01AA01	10.982
Simvastatin	C10AA01	10.761
Azithromycin	J01FA10	6.013
Amiodaron	C01BD01	5.534
Moxifloxacin	J01MA14	5.343
Domperidon	A03FA03	3.031
Donepezil	N06DA02	2.632

Dabei zeigte sich, dass Clarithromycin, besonders mit Medikamenten der ATC-Gruppe *Antidepressiva*, Citalopram besonders mit Medikamenten der ATC-Gruppe *Makrolide*, *Lincosamide* und *Streptogramine* oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten Medikamentengruppen sind:

- Makrolide, Lincosmaide und Streptogramine (J01F): 26.070 Alerts
- Antidepressiva (N06A): 25.984 Alerts
- Chinolone (J01M): 18.385 Alerts

Die ONC high priority Interaktionsliste wird von insgesamt 16 Interaktionspaar-Gruppen gebildet. Die Gruppe Febuxostate - Azathioprine/Mercaptopurine wurde in dieser Diplomarbeit jedoch nicht ausgewertet, da das Arzneimittel Febuxostat in Österreich erst seit 2011 zugelassen ist und deswegen kein Eintrag im ATC-Index 2007 vorhanden ist. In Tabelle 4-20 sind die restlichen 14 Interaktionspaar-Gruppen mit der Anzahl der verschiedenen ATC-Codes und Interaktionspaaren und deren Alerts aufgeführt.

Tabelle 4-20: Die Interaktionspaar-Gruppen mit deren Anzahl an Interaktionspaaren und verschiedenen ATC-Codes, sortiert nach der Anzahl der Alerts

Interaktionspaar-Gruppe	verschiedene ATC-Code	Anzahl Interaktionspaare	Alerts
QT prolonging agents - QT prolonging agents	68	2.278	40.710

## Resultate

HMG Co-A reductase inhibitors - CYP4A4 inhibitors + Protease inhibitors/Macrolides/Azoles	32	135	11.240
Tizanidine - CYP1A2 inhibitors	4	3	990
SSRIs - MAO inhibitors	18	72	200
Narcotic analgesics - MAO-inhibitors	19	78	98
CYP3A4 inhibitors + Protease inhibitors/Macrolides/Azoles - Ergot alkaloids and derivatives	25	100	55
Tricyclic antidepressants - MAO inhibitors	29	138	27
Atazanavir - PPI	6	5	9
Amphetamine derivatives - MAO inhibitors	14	48	3
Strong CYP3A4 inducers - Protease inhibitors	14	45	2
Triptans - MAO inhibitors	6	8	2
Febuxostat - Azathioprine/Mercaptopurine	3	2	0
Irinotecan - CYP3A4 inhibitors + protease inhibitors/Macrolides/Azoles	34	33	0
Tranylecypromine - Procarbazine	3	2	0

### 4.2.1.4 In allen drei Listen vorkommende Interaktionen

Insgesamt 72 Interaktionspaare kommen in allen drei Interaktionslisten ident vor. Dabei wurden 12.449 Alerts ausgelöst. Tabelle 4-21 zeigt die zehn häufigsten Interaktionspaare mit deren ATC-Codes und den ausgelösten Alerts.

**Tabelle 4-21: Zehn häufigste Interaktionspaare, welche in allen 3 Interaktionslisten vorkommen, geordnet nach Alerts**

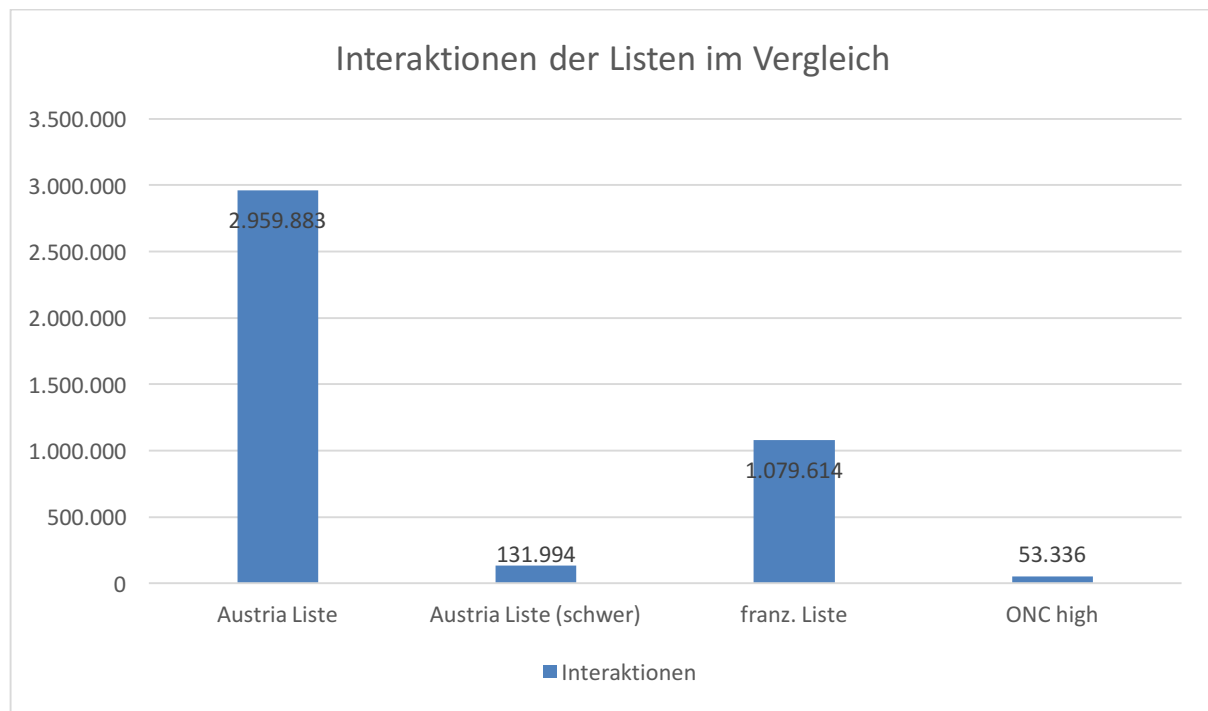
Häufigste Interaktionspaare	ATC-Codes	Alerts
Simvastatin : Clarithromycin	C10AA01:J01FA09	5.483
Simvastatin : Amiodaron	C10AA01:C01BD01	1.909
Verapamil : Simvastatin	C08DA01:C10AA01	1.277
Diltiazem : Simvastatin	C08DB01:C10AA01	1.198
Amiodaron : Clarithromycin	C01BD01:J01FA09	542
Amiodaron : Moxifloxacin	C01BD01:J01MA14	326
Sotalol : Clarithromycin	C07AA07:J01FA09	282
Simvastatin : Itraconazol	C10AA01:J02AC02	256
Sotalol : Amiodaron	C07AA07:C01BD01	142
Tramadol : Selegelin	N02AX02:N04BD01	98



### 4.2.1.5 Vergleich der Listen

Die Austria CodexListe erzeugt 2.959.883 Alerts (1,8 Alerts pro Patient), bei Inkludierung von nur schweren Interaktionspaaren der Austria-Liste werden 131.994 Alerts erzeugt. Die französische Interaktionsliste zeigt etwa ein Drittel der Alerts der Austria Codex Liste (1.079.614), die ONC high priority Liste deutlich weniger mit nur 53.336 Alerts. Mit Ausnahme der Austria-Codex Liste sind die häufigsten Medikamente, welche bei Interaktionen beteiligt sind, in der Neurologie und Psychiatrie verwendete Arzneimittel wie Antidepressiva, Antidementiva und Antipsychotika. Abbildung 4.2-1 zeigt ein Vergleich der drei Hauptlisten (+1 nur schwere Interaktionen des Austria Codex) Listen und der Anzahl der erzeugten Alerts.

**Abbildung 4.2-1: Interaktionen im Vergleich von den drei Listen inkl. der Austria-Liste mit ausschließlich schweren Interaktionen**



### 4.2.1.6 Nur Antidepressiva

Antidepressiva (ATC-Code: N06A) lösen in allen vier Interaktionslisten viele Alerts aus. Deshalb wurde diese Gruppe näher analysiert und verglichen. Insgesamt 193.568 Patienten erhielten Antidepressiva, welches als Interaktion in eines der Listen beteiligt ist.

Die Tabelle 4-22 zeigt die Anzahl der Alerts der Listen und deren Alerts bei Antidepressiva.

## Resultate

**Tabelle 4-22: Anzahl der Alerts pro Interaktionsliste von Antidepressiva**

<b>Interaktionsliste</b>	<b>Alerts</b>
französische Interaktionsliste	236.447
Austria Codex Interaktionsliste	385.439
Austria Codex Interaktionsliste (schwer)	34.297
ONC high priority Interaktionsliste	24.200

Citalopram und Escitalopram war häufig in Interaktionswarnungen beteiligt, jedoch je nach Liste unterschiedlich häufig. Auffällig ist, dass in der schweren Austria Codex-Interaktionsliste die meisten Medikamente aus der Gruppe der Anderen Antidepressiva (N06AX) kommen. Die häufigsten drei Medikamente der Listen sind in Tabelle 4-23 zusammengefasst.

**Tabelle 4-23: Häufigste drei Alert-auslösende Medikamente für die untersuchten Listen**

	<b>Häufigste Medikamente</b>	<b>ATC-Codes</b>	<b>Alerts</b>
<b>franz. Liste</b>			
	Citalopram	N06AB04	61.886
	Escitalopram	N06AB10	50.384
	Amitriptylin	N06AA09	37.141
<b>Austria Codex Liste</b>			
	Citalopram	N06AB04	73.888
	Trazodon	N06AX05	61.080
	Escitalopram	N06AB10	60.456
<b>Austria Codex Liste (schwer)</b>			
	Mirtazapin	N06AX11	9.809
	Trazodon	N06AX05	9.159
	Venlafaxin	N06AX16	5.020
<b>ONC high priority Liste</b>			
	Citalopram	N06AB04	13.978
	Trazodon	N06AB10	11.866
	Escitalopram	N06AB06	33

In der französischen Interaktionsliste sowie der Austria Codex sind häufig Interaktionen zwischen Antidepressiva und NSAR zu beobachten. Die ONC high priority Liste zeigt hier

jedoch die Prioritäten auf Interaktionen zwischen Antidepressiva und Antibiotika zu haben. Die häufigsten drei Interaktions-Paare sind in Tabelle 4-24 zusammengefasst.

**Tabelle 4-24: Häufigste drei Interaktions-Paare in den 3 (+1) Listen und deren Alerts**

	<b>Häufigste Medikamente</b>	<b>ATC-Codes</b>	<b>Alerts</b>
<b>franz. Liste</b>			
	Citalopram : Mefenaminsäure	N06AB04:M01AG01	6.550
	Escitalopram : Mefenaminsäure	N06AB10:M01AG01	5.149
	Citalopram : Tramadol	N06AB04:N02AX02	5.111
<b>Austria Codex Liste</b>			
	Diclofenac : Citalopram	M01AB05:N06AB04	8.469
	Escitalopram : Trazodon	N06AB10:N06AX05	6.813
	Mefenaminsäure : Citalopram	M01AG01:N06AB04	6.550
<b>Austria Codex Liste (schwer)</b>			
	Trazodon : Prothipendyl	N06AX05:N05AX07	2.278
	Mirtazapin : Prothipendyl	N06AX11:N05AX07	2.081
	Quetiapin : Trazodon	N05AH04:N06AX05	1.810
<b>ONC high priority Liste</b>			
	Citalopram : Clarithromycin	N06AB04:J01FA09	3.374
	Ciprofloxacin : Citalopram	J01MA02:N06AB04	2.679
	Clarithromycin : Escitalopram	J01FA09:N06AB10	2.657

### 4.2.2 Interaktionsberechnung ohne Reichweiteninformation

Um die Einschränkungen und Ungenauigkeiten der Reichweiteninformation auszuschalten wurde auch eine Interaktionsberechnung ohne diese durchgeführt. Dabei werden nur Rezepte auf Interaktionen verglichen, die am selben Tag eingelöst wurden.

**Tabelle 4-25: Gesamtübersicht der Interaktionslisten (Alerts ohne Mehrfachwarnungen + Alerts mit Mehrfachalerts)**

	<b>Alerts mit Mehrfachwarnungen</b>	<b>Alerts ohne Mehrfachwarnungen</b>
franz. Interaktionsliste	2.580.114	630.926
Austria Codex Interaktionsliste	5.017.482	1.507.300
ONC High-Priority Interaktionsliste	53.312	20.504

### 4.2.2.1 Französische Interaktionsliste

Insgesamt 630.926 Alerts wurden mit der französischen Interaktionsliste ohne Reichweiteninformation ausgelöst. Im Vergleich dazu stehen 1.079.614 Alerts bei Beachtung der Reichweiteninformation. Die häufigsten Interaktionspaare sind in Tabelle 4-26 aufgelistet.

**Tabelle 4-26: häufigste Interaktionspaare der franz. Interaktionsliste (ohne Reichweiteninf.)**

<b>Häufigste Interaktionspaare</b>	<b>ATC-Codes</b>	<b>Alerts</b>
Amlodipin : Bisoprolol	C08CA01:C07AB07	7.859
Amlodipin : Carvedilol	C08CA01:C07AG02	7.071
Amlodipin : Metoprolol	C08CA01:C07AB02	6.201
Furosemid : Lisinopril	C03CA01:C09AA03	4.804
Mefenaminsäure : Bisoprolol	M01AG01:C07AB07	4.801
Nicorandil : Bisoprolol	C01DX16:C07AB07	4.548
Spironolacton : Furosemid	C03DA01:C03CA01	4.288
Nicorandil : Furosemid	C01DX16:C03CA01	3.904
Furosemid : Enalapril	C03CA01:C09AA02	3.756
Tetrazepam : Tramadol	M03BX07:N02AX02	3.736

Die häufigsten Medikamente, welche einen Alert ausgelöst haben, sind in Tabelle 4-27 zusammengefasst. Furosemid und Bisoprolol waren oft in Interaktionen beteiligt.

**Tabelle 4-27: häufigste Medikamente der französischen Interaktionsliste**

<b>Häufigste Medikamente</b>	<b>ATC-Code</b>	<b>Alerts</b>
Furosemid	C03CA01	64.144
Bisoprolol	C07AB07	56.709
Mefenaminsäure	M01AG01	50.339
Amlodipin	C08CA01	44.307

## Resultate

Carvedilol	C07AG02	34.426
Tramadol	N02AX02	34.327
Lisinopril	C09AA03	34.313
Bisoprolol	C07AB02	34.264
Citalopram	N06AB04	34.172
Doxazosin	C02CA04	31.704

Dabei zeigte sich, dass Furosemid besonders mit Medikamenten der ATC-Gruppe *ACE-Hemmer, rein*, Bisoprolol besonders mit Medikamenten der ATC-Gruppe *selektive Calciumkanalblocker mit vorwiegender Gefäßwirkung* oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten Medikamentengruppen sind:

- Beta-Adrenorezeptor-Antagonisten, nichtselektiv (C07A): 153.585 Alerts
- Antidepressiva (N06A): 131.469 Alerts
- ACE-Hemmer, rein (C09A): 98.336 Alerts

### 4.2.2.2 Austria Codex Interaktionsliste

Insgesamt 1.507.300 Alerts wurden mit der Austria Codex Interaktionsliste ohne Reichweiteninformation ausgelöst. Im Vergleich dazu stehen 2.959.883 Alerts bei Beachtung der Reichweiteninformation. Die häufigsten Interaktionspaare sind in Tabelle 4-28 aufgelistet.

Tabelle 4-28: häufigste Interaktionspaare der Austria Codex Interaktionsliste (ohne Reichweiteninf.)

Häufigste Interaktionspaare	ATC-Codes	Alerts
Bisoprolol : Amlodipin	C07AB07:C08CA01	7.859
Lisinopril : Lisinopril und Diuretika	C09AA03:C09BA03	7.761
Enalapril : Enalapril und Diuretika	C09AA02:C09BA02	7.605
Carvedilol : Amlodipin	C07AG02:C08CA01	7.071
Bisoprolol : Diclofenac	C07AB07:M01AB05	7.062
Metformin : Diclofenac	A10BA02:M01AB05	6.298
Metoprolol : Amlodipin	C07AB02:C08CA01	6.201
Lisinopril und Diuretika : Allopurinol	C09BA03:M04AA01	6.001

## Resultate

Lisinopril und Diuretika : Diclofenac	C09BA03:M01AB05	5.964
Metformin : Lisinopril und Diuretika	A10BA02:C09BA03	5.769

Die häufigsten Medikamente, welche einen Alert ausgelöst haben, sind in Tabelle 4-29 zusammengefasst. Diclofenac und Metformin sind hier oft in Interaktionswarnungen vorgekommen.

**Tabelle 4-29: häufigste Medikamente der Austria Codex Interaktionsliste (ohne Reichweiteninf.)**

Häufigste Medikamente	ATC-Codes	Alerts
Diclofenac	M01AB05	134.473
Metformin	A10BA02	87.981
Lisinopril und Diuretika	C09BA03	69.749
Mefenaminsäure	M01AG01	69.329
Furosemid	C03CA01	69.203
Acetylsalicylsäure	B01AC06	69.173
Bisoprolol	C07AB07	68.948
Enalapril und Diuretika	C09BA02	61.577
Lisinopril	C09AA03	60.743
Enalapril	C09AA02	55.316

Dabei zeigte sich, dass Diclofenac besonders mit Medikamenten der ATC-Gruppe *Beta-Adrenorezeptor-Antagonisten*, Metformin besonders mit Medikamenten der ATC-Gruppe *Nicht steroidale Antiphlogistika und Antirheumatika* oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten Medikamentengruppen sind:

- Nichtsteroidale Antiphlogistika und Antirheumatika (M01A): 394.566 Alerts
- Antidepressiva (N06A): 222.195 Alerts
- Beta-Adrenorezeptorenantagonisten (C07A): 203.598 Alerts

Bei Verwendung der nur schweren Austria Codex Interaktionsliste ohne Reichweiteninformation wurden insgesamt 66.352 Alerts ausgelöst. Im Vergleich dazu stehen

## Resultate

131.994 Alerts bei Beachtung der Reichweiteninformation. Die häufigsten Interaktionspaare sind in Tabelle 4-30 aufgelistet.

**Tabelle 4-30: häufigste Interaktionspaare der Austria Codex Interaktionsliste (nur schwere Interaktionen, ohne Reichweiteninf.)**

Häufigste Interaktionspaare	ATC-Codes	Alerts
Prothipendyl : Trazodon	N05AX07:N06AX05	1.731
Prothipendyl : Mirtazapin	N05AX07:N06AX11	1.592
Simvastatin : Clarithromycin	C10AA01:J01FA09	1.447
Trazodon : Quetiapin	N06AX05:N05AH04	1.416
Quetiapin : Mirtazapin	N05AH04:N06AX11	1.408
Trazodon : Melitracen und Psycholeptika	N06AX05:N06CA02	1.062
Carvedilol : Theophyllin	C07AG02:R03DA04	1.002
Desloratadin : Clarithromycin	R06AX27:J01FA09	983
Olanzapin : Mirtazapin	N05AH03:N06AX11	976
Venlafaxin : Prothipendyl	N06AX16:N05AX07	940

Die häufigsten Medikamente, welche einen Alert ausgelöst haben, sind in Tabelle 4-31 zusammengefasst. Mirtazapin und Trazodon sind hier oft in Interaktionswarnungen vorgekommen.

**Tabelle 4-31: häufigste Medikamente der Austria Codex Interaktionsliste (nur schwere Interaktionen, ohne Reichweiteninf.)**

häufigste Medikamente	ATC-Code	Alerts
Mirtazapin	N06AX11	7.309
Trazodon	N06AX05	6.763
Prothipendyl	N05AX07	6.735
Quetiapin	N05AH04	6.527
Clarithromycin	J01FA09	5.432
Melitracen und Psycholeptika	N06CA02	4.844
Olanzapin	N05AH03	4.182
Ranitidin	A02BA02	3.856
Venlafaxin	N06AX16	3.839
Simvastatin	C10AA01	3.563

## Resultate

Dabei zeigte sich, dass sowohl Mirtazapin, als auch Trazodon besonders mit Medikamenten der ATC-Gruppe *Antipsychotika*, oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten Medikamentengruppen sind

- Antipsychotika (N05A): 28.2105 Alerts
- Antidepressiva (N06A): 25.099 Alerts
- Mittel bei peptischem Ulkus und gastroösophagealer Refluxkrankheit (A02B): 7.267 Alerts

### 4.2.2.3 *ONC high priority Interaktionsliste*

Insgesamt 20.504 Alerts wurden mit der ONC high priority Interaktionsliste ohne Reichweiteninformation ausgelöst. Im Vergleich dazu stehen 54.336 Alerts bei Beachtung der Reichweiteninformation. Die häufigsten Interaktionspaare sind in Tabelle 4-32 aufgelistet.

Tabelle 4-32: häufigste Interaktionspaare der ONC high priority Interaktionsliste (ohne Reichweiteninf.)

Häufigste Interaktionspaare	ATC-Codes	Alerts
Simvastatin : Amiodaron	C10AA01:C01BD01	1.467
Simvastatin : Clarithromycin	C10AA01:J01FA09	1.447
Citalopram : Clarithromycin	N06AB04:J01FA09	1.052
Simvastatin : Verapamil	C10AA01:C08DA01	1.017
Simvastatin : Diltiazem	C10AA01:C08DB01	1.016
Clarithromycin : Escitalopram	J01FA09:N06AB10	851
Ciprofloxacin : Citalopram	J01MA02:N06AB04	812
Ciprofloxacin : Escitalopram	J01MA02:N06AB10	635
Citalopram : Donezepil	N06AB04:N06DA02	555
Donezepil : Escitalopram	N06DA02:N06AB10	525

Die häufigsten Medikamente, welche einen Alert ausgelöst haben, sind in Tabelle 4-33 zusammengefasst. Clarithromycin und Simvastatin sind hier oft in Interaktionen vorgekommen.

Tabelle 4-33: häufigste Medikamente der ONC high priority Interaktionsliste (ohne Reichweiteninf.)

Häufigste Medikamente	ATC-Code	Alerts
-----------------------	----------	--------



## Resultate

Clarithromycin	J01FA09	5.279
Simvastatin	C10AA01	5.205
Citalopram	N06AB04	4.991
Escitalopram	N06AB10	4.379
Ciprofloxacin	J01MA02	3.630
Amiodaron	C01BD01	2.891
Azithromycin	J01FA10	1.731
Donezepil	N06DA02	1.513
Domperidon	A03FA03	1.504
Moxifloxacin	J01MA14	1.159

Dabei zeigte sich, dass Clarithromycin besonders mit Medikamenten der ATC-Gruppe *Antidepressiva*, Simvastatin besonders mit Medikamenten der ATC-Gruppe *selektive Calcium-Kanalblocker mit vorwiegender Gefäßwirkung* oft gemeinsam verschrieben wurden und Interaktionswarnungen ausgelöst haben.

Die häufigsten Medikamentengruppen sind:

- Antidepressiva (N06A): 9.162 Alerts
- Makrolide, Lincosamide und Streptogramine (J01F): 7.110 Alerts
- Chinolone (J01M): 5.323 Alerts

In Tabelle 4-34 sind die 14 Interaktionspaar-Gruppen der ONC high priority Liste aufgeführt. Hier zeigt sich, dass auch ohne Reichweiteninformation eine Kombination zweier QT-verlängernder Medikamente die häufigste Interaktionswarnungs-Ursache war.

**Tabelle 4-34: Die Interaktionspaar-Gruppen mit deren Anzahl an Interaktionspaaren und verschiedenen ATC-Codes, sortiert nach der Anzahl der Alerts (ohne Reichweiteninf.)**

<b>Interaktionspaar-Gruppen</b>	<b>verschiedene ATC-Codes</b>	<b>Anzahl Interaktionspaare</b>	<b>Alerts</b>
QT prolonging agents - QT prolonging agents	68	2.278	14.461
HMG Co-A reductase inhibitors - CYP4A4 inhibitors + Protease inhibitors/Macrolides/Azoles	32	135	5.446
SSRIs - MAO inhibitors	4	3	138
Tizanidine - CYP1A2 inhibitors	18	72	359
Narcotic analgesics - MAO-inhibitors	19	78	49

## Resultate

Tricyclic antidepressants - MAO inhibitors	25	100	15
CYP3A4 inhibitors + Protease inhibitors/Macrolides/Azoles - Ergot alkaloids and derivatives	29	138	27
Atazanavir - PPI	6	5	4
Strong CYP3A4 inducers - Protease inhibitors	14	48	2
Amphetamine derivatives - MAO inhibitors	14	45	2
Triptans - MAO inhibitors	6	8	1
Febuxostat - Azathioprine/Mercaptopurine	3	2	0
Irinotecan - CYP3A4 inhibitors + protease inhibitors/Macrolides/Azoles	34	33	0
Tranlycypromine - Procarbazine	3	2	0

### 4.2.2.4 In allen drei Listen vorkommende Interaktionen

Insgesamt 72 Interaktionspaare kommen in allen drei Interaktionslisten vor. Dabei wurden 5.803 Alerts ausgelöst. Tabelle 4-35 zeigt die zehn häufigsten Interaktionspaare mit deren ATC-Codes und den ausgelösten Alerts. Hier zeigt sich ein ähnliches Bild wie bei den häufigsten Interaktionspaaren mit Reichweiteninformation.

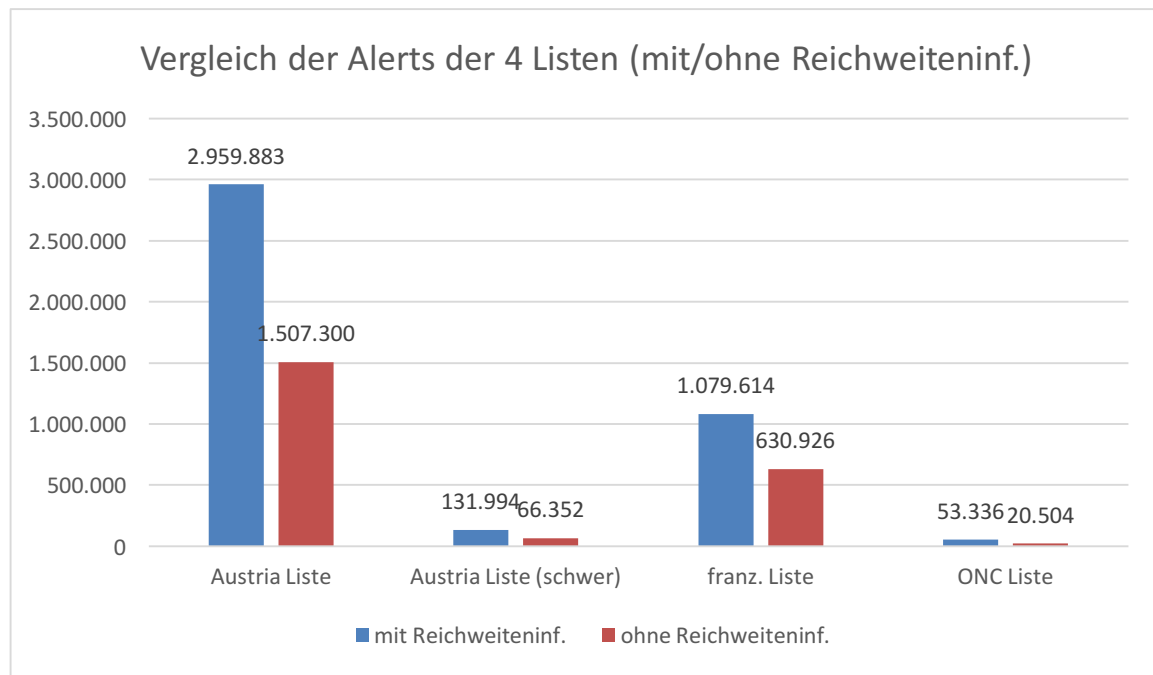
**Tabelle 4-35: häufigste Interaktionspaare, welche in allen 3 Interaktionslisten vorkommen, geordnet nach Alerts**

Häufigste Interaktionspaare	ATC-Codes	Alerts
Amiodaron : Simvastatin	C01BD01:C10AA01	1.467
Simvastatin : Clarithromycin	C10AA01:J01FA09	1.447
Verapamil : Simvastatin	C08DA01:C10AA01	1.017
Diltiazem : Simvastatin	C08DB01:C10AA01	1.016
Amiodaron : Clarithromycin	C01BD01:J01FA09	97
Amiodaron : Moxifloxacin	C01BD01:J01MA14	80
Sotalol : Clarithromycin	C07AA07:J01FA09	79
Simvastatin : Itraconazol	C10AA01:J02AC02	59
Verapamil, Kombinationen : Simvastatin	C08DA51:C10AA01	59
Tramadol : Selegilin	N02AX02:N04BD01	49

### 4.2.2.5 Vergleich der Listen

Auch ohne Reichweiteninformation zeigt sich ein ähnliches Bild wie mit Berücksichtigung dieser Information. Die Austria Codex Interaktionsliste erzeugt deutlich die meisten Interaktionen. Am wenigsten werden von der ONC high priority Liste erzeugt. Abbildung 4.2-2 zeigt einen Vergleich der 4 Listen mit und ohne Reichweiteninformation.

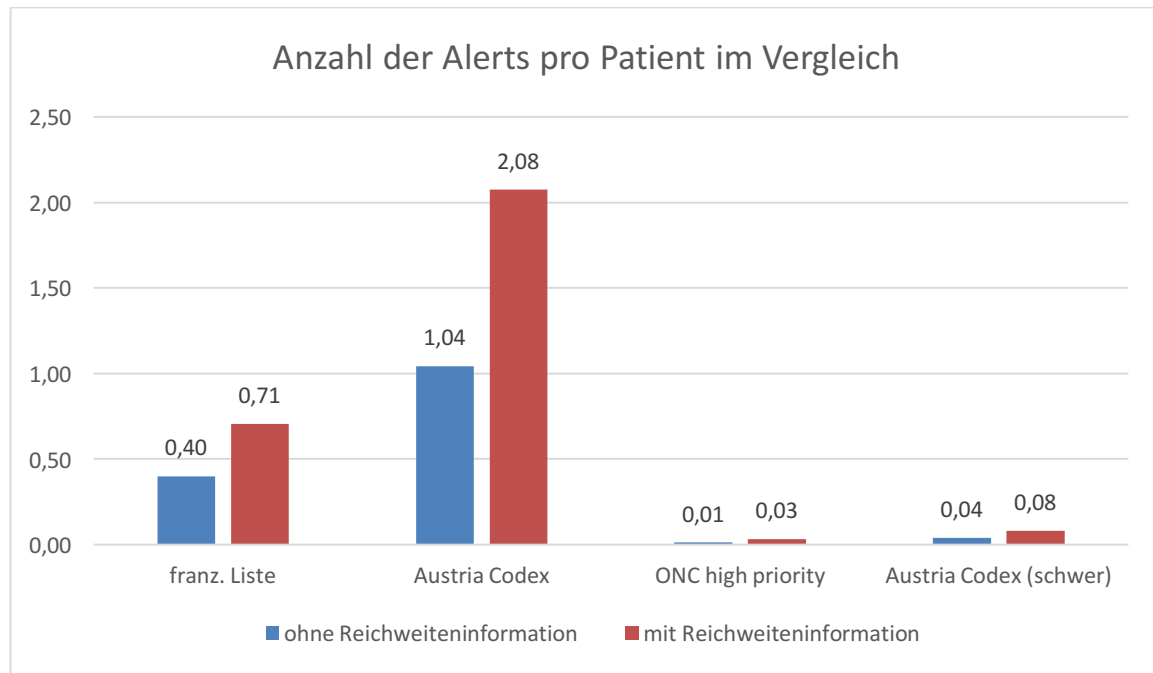
**Abbildung 4.2-2: Vergleich der Alerts der 4 Listen mit und ohne Reichweiteninformation**



Im Vergleich zur Auswertung mit Reichweiteninformation zeigt sich hier etwa eine Halbierung der Anzahl der Alerts bei einer Interaktionsberechnung mit Rezepten, welche am selben Tag eingelöst wurden. Abbildung 4.2-3 zeigt den Vergleich.

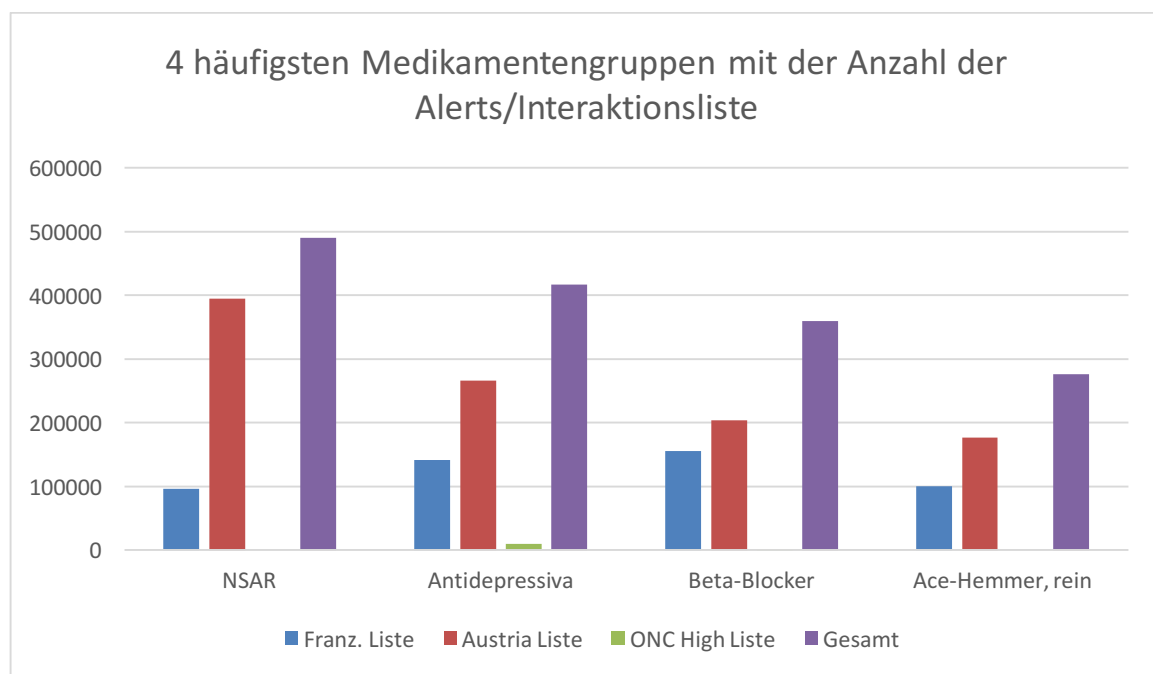
## Resultate

Abbildung 4.2-3: Anzahl der Alerts pro Patient je Liste im Vergleich mit/ohne Reichweiteninformation



Die häufigsten Medikamentengruppen der drei Listen (ohne schwere Austria Codex-Liste) sind NSAR, Antidepressiva, Beta-Blocker und ACE-Hemmer. Abbildung 4.2-4 zeigt den Vergleich der drei Listen.

Abbildung 4.2-4: Vergleich der 4 häufigsten Medikamentengruppen mit der Anzahl der Alerts pro Interaktionsliste



### 4.2.2.6 Nur Antidepressiva

Auch ohne Reichweiteninformation sind Antidepressiva eine häufige Medikamentengruppe von Interaktionswarnungen. Die Tabelle 4-36 zeigt die Anzahl der Alerts bei Antidepressiva.

## Resultate

**Tabelle 4-36: Anzahl der Alerts pro Interaktionsliste von Antidepressiva (ohne Reichweiteninf.)**

<b>Interaktionsliste</b>	<b>Alerts</b>
französische Interaktionsliste	131.469
Austria Codex Interaktionsliste	222.195
Austria Codex Interaktionsliste (schwer)	25.099
ONC high priority Interaktionsliste	9.162

Auch ohne Reichweiteninformation waren Citalopram und Escitalopram häufig bei Interaktionswarnungen beteiligt. Auch hier ist auffällig, dass die schwere Austria Codex Interaktionsliste die meisten Medikamente aus der Gruppe der Anderen Antidepressiva (N06AX) beinhaltet. Die häufigsten drei Medikamente der drei (+1) Listen sind in Tabelle 4-37 zusammengefasst.

**Tabelle 4-37: häufigste drei Medikamente in den 3 (+1) Listen und deren Alerts (ohne Reichweiteninf.)**

	<b>häufigste Medikamente</b>	<b>ATC-Code</b>	<b>Alerts</b>
<b>franz. Liste</b>			
	Citalopram	N06AB04	34.172
	Escitalopram	N06AB10	27.723
	Amitriptylin	N06AA09	22.717
<b>Austria Codex Liste</b>			
	Citalopram	N06AB04	41.726
	Trazodol	N06AX05	39.915
	Escitalopram	N06AB10	35.499
<b>Austria Codex Liste (schwer)</b>			
	Mirtazapin	N06AX11	7.309
	Trazodon	N06AX05	6.763
	Venlafaxin	N06AX16	3.839
<b>ONC high priority Liste</b>			
	Citalopram	N06AB04	4.991
	Escitalopram	N06AB10	4.379
	Sertralin	N06AB06	23

## Resultate

Es zeigt sich im Gegensatz zu den Daten mit Reichweiteninformation keine klare Richtung, außer bei der ONC high priority Liste, welche häufig vor Interaktionen zwischen Antibiotika und Antidepressiva warnen. Tabelle 4-38 zeigt die häufigsten drei Interaktions-Paare der 3(+1) Listen.

**Tabelle 4-38: häufigste drei Interaktions-Paare der 3 (+1) Listen und deren Alerts (ohne Reichweiteninf.)**

	<b>Häufigste Interaktionspaare</b>	<b>ATC-Codes</b>	<b>Alerts</b>
<b>franz. Liste</b>			
	Citalopram : Mefenaminsäure	N06AB04:M01AG01	3.461
	Furosemid : Citalopram	C03CA01:N06AB04	2.879
	Citalopram : Tramadol	N06AB04:N02AX02	2.877
<b>Austria Codex Liste</b>			
	Escitalopram : Trazodol	N06AB10:N06AX05	5.609
	Trazodol : Citalopram	N06AX05:N06AB04	4.430
	Diclofenac : Citalopram	M01AB05:N06AB04	4.412
<b>Austria Codex Liste (schwer)</b>			
	Prothipendyl : Trazodon	N05AX07:N06AX05	1.731
	Prothipendyl : Mirtazapin	N05AX07:N06AX11	1.592
	Trazodon : Quetiapin	N06AX05:N05AH04	1.416
<b>ONC high priority Liste</b>			
	Citalopram : Clarithromycin	N06AB04:J01FA09	1.052
	Clarithromycin : Escitalopram	J01FA09:N06AB10	851
	Ciprofloxacin : Citalopram	J01MA02:N06AB04	812

## 5 Diskussion

Pharmakogenetische Informationen können Therapieentscheidungen stark beeinflussen. In dieser Diplomarbeit zeigt sich, dass eine große Anzahl an Personen PGx-Medikamente erhielt, was ein großes Potential für die Verbesserung der Therapie mittels pharmakogenetischen Daten bietet. In den zwei analysierten Jahren hat etwa ein Viertel der Population mindestens ein PGx-Medikament erhalten. In der Altersgruppe  $\geq 65$  erhielten etwa 50% zumindest ein PGx-Medikament der „Hoch-signifikant“-Gruppe. Diese Altersgruppe ist durch Polymedikation und chronischen Krankheiten nicht mehr so gut in der Lage, mit etwaigen Arzneimittelnebenwirkungen umzugehen, wodurch häufige Hospitalisierungen und hohe Kosten für das Gesundheitssystem resultieren.

In der Altersgruppe 40-64 zeigt sich eine starke Verbreitung von Östrogenhaltigen Medikamenten (8,7% der Altersgruppe). Dabei ist zu berücksichtigen, dass die Verbreitung von diesen Medikamenten deutlich höher ist als hier dargestellt. Das resultiert daraus, dass die Pille als Verhütungsmittel keine Kassenleistung ist und daher nicht in diesen Daten enthalten. In einer Umfrage im Verhütungsreport 2015 von GynMed zeigt sich, dass 38% der Frauen die Pille als Verhütungsmethode verwenden. (31) Bei einer Annahme von 4.367.382 Frauen, wie im Gender Index 2015 beschrieben, verwenden also 1.659.605 Frauen die Pille. (32)

Die drei CYPs CYP2C9, CYP2C19 und CYP2D6 sind in den Abbau von vielen verschiedenen Arzneistoffen involviert. Dabei zeigt sich, dass etwa 80% der Personen, welche ein PGx-Medikament erhielten, ein Medikament mit einem der drei CYP-Gruppen assoziiert war. Besonders in der Altersgruppe  $\geq 65$  steigt dieser Anteil auf etwa 88% an. Allein die CYP2D6-assoziierten Medikamente werden bei 30,38% der Altersgruppe  $\geq 65$  eingesetzt. Hier zeigt sich ein großes Potential von reaktiver pre-emptiver pharmakogenetischer Genotypisierung, da einzelne Gene relativ kostengünstig typisiert werden können. Im Folgenden ein Kostenrechnungsbeispiel mit Clopidogrel und CYP2C19:

In einem Labor in Graz wurde 2015 für die Genotypisierung von drei Genen (CYP2C9, CYP2C19, CYP2D6) etwa 300 Euro verlangt. Dabei kostet die CYP2C19 Bestimmung, welches für den Abbau von Clopidogrel verantwortlich ist, 89 Euro. In Österreich erhielten in den Jahren 2006-2007 102.054 Personen mindestens einmal Clopidogrel. Bei der Annahme, dass auch 28,5% der Patienten ein relevantes genetisches Profil, wie in der Studie der Vanderbilt University haben, würden 29.085 Personen von einem preemptiven

## Diskussion

pharmakogenetischen Test profitieren. (2,3) Bei der willkürlichen Annahme, dass 10% dieser Personen eine behandlungsbedürftige unerwünschte Arzneimittelwirkung entwickeln, ist ab den durchschnittlichen Behandlungskosten von mindestens 3122,8 Euro die Genotypisierung kostengünstiger. Tabelle 5-1. zeigt die Zahlen in Übersicht.

**Tabelle 5-1: Annahme des Break-Even Points der Genotypisierung von CYP2C19**

Gesamtkosten der CYP2C19 Genotypisierung (102.054 Personen)	9.082.806 €
Anzahl an Personen, welche ein genetisch relevantes Profil haben	29.085 Personen
Break-Even Point der Kosten/Behandlung der Personen mit genetisch relevantem Profil	312,28 €
Break-Even Point der Kosten/Behandlung bei Annahme, dass 10% der Personen benötigen eine Behandlung benötigen	<b>3122,8 €</b>

In dieser Diplomarbeit zeigt sich, dass die reaktive pre-emptive pharmakogenetische Genotypisierung ein deutlich zielgerichteteres Tool ist, diese Informationen einzusetzen. Das European PGx Implementation Project „Ubiquitous pharmacogenomics“ (U-PGx) hat es sich zum Ziel gesetzt, gemeinsame Europäische Guidelines und Technologien zu entwickeln, um die Verbreitung des Einsatzes von pharmakogenetischen Informationen im klinischen Routinebetrieb voranzutreiben. (33) Eine multizentrische Studie von 4.000 Personen mit pre-emptiver Genotypisierung für Gene, welche in den DPWG Guidelines enthalten sind, wird 2017 durchgeführt. (34)(35)

Weitere Studien zur Evaluation der Kosten/Nutzen-Rechnung der Pharmakogenetischen Genotypisierung sind nötig. Insbesondere ist die Frage, wie hoch die Wahrscheinlichkeit ist, dass ein Mensch in seinem gesamten Leben mindestens ein PGx-Medikament erhält *und* durch pharmakogenetische Genotypisierung eine unerwünschte Arzneimittelwirkung verhindert werden kann.

Wie erwartet zeigen die Arzneimittel-Interaktionssimulationen mittels Austria Codex eine hohe Anzahl an Warnungen. Ohne Berücksichtigung der Reichweiteninformation wird statistisch bei jedem einzelnen Patienten eine Warnung ausgegeben. Im Vergleich zeigt sich bei der hochsignifikanten ONC high priority Liste nur eine einzige Warnung bei jedem 100. Patient. Bei Verwendung der Austria Codex-Liste schwerwiegender Interaktionen zeigt sich eine Warnung bei etwa jedem 25. Patient. Bei der französischen Interaktionsliste zeigt sich eine



## Diskussion

Warnung bei etwa jeden 2-3. Patient. Hier zeigt sich also das Potential einer deutlichen Reduktion der Alerts unter Verwendung von selektiven Interaktionslisten im Vergleich zum Austria Codex-Liste.

Statine, Makrolide sowie Antidepressiva sind in allen drei Listen häufig in Interaktionen vertreten und lösen auch häufig Alerts aus. Die häufigste Interaktion der ONC high priority Liste ist *Simvastatin* : *Amiodaron*. In mehreren Studien zeigt sich, dass Simvastatin in Kombination von Amiodaron eine Rhabdomyolyse auslösen kann, was eine typische Nebenwirkung von Statinen bei zu hohem Plasmaspiegel ist. (36–38) Da Simvastatin über CYP3A4 metabolisiert wird, zeigt sich bei genetischer Veränderung dieses Enzyms oder bei Einnahme von sogenannten CYP3A4-Inhibitoren, wie zB Amiodaron, dass der Abbau von Simvastatin verringert wird. Dies resultiert in einem erhöhten Plasmaspiegel im Blut und somit einer erhöhten Gefahr von Nebenwirkungen.

Unerwünschte Arzneimittelwirkungen sind ein relativ häufiger Grund für stationäre Aufnahmen. Eine Studie zeigte eine deutliche Zunahme an stationären Aufnahmen durch die Kombination von Statinen und Amiodaron. Das relative Risiko einer Hospitalisierung bei Kombination eines Statins mit Amiodaron steigt um 187% im Vergleich zu einer reinen Statin-Gabe. (39)

Für die Arzneimittelwarnungen sind nicht nur die richtigen Interaktionspaare für die Akzeptanz von Ärztinnen und Ärzten wichtig. Auch sollten bei Warnungen Alternativen vorgeschlagen werden. Als Beispiel einer Interaktion bei Simvastatin und Amiodaron, wobei ein anderes Statin wie zB Pravastatin vorgeschlagen werden soll, da dieses größtenteils ohne Umwandlung ausgeschieden wird und daher CYP unabhängig ist, angeboten wird. (40)

Eine grundlegende Frage muss bei der Erstellung hoch-signifikanter Arzneimittelinteraktionslisten gestellt werden: Soll die Interaktionsliste der verschreibenden Person helfen, nur vor den stärksten und schwerwiegendsten Arzneimittelinteraktionen zu bewahren, oder auch vor eher unwahrscheinlichen Interaktionen? Zu viele Interaktionspaare in der Liste führen zu häufigeren Alerts und dadurch womöglich zu „alert fatigue“. Bei zu wenigen Interaktionspaaren könnte der Arzt eine Interaktion zwischen Medikamenten übersehen und so im schlimmsten Fall seinen Patienten schädigen.

Ein weiteres Gebiet für die Verbesserung von klinischen Entscheidungssystemen ist die Phenoconversion. Dies beschreibt jenes Phänomen, welches genetisch schnelle Metabolisierer (EM-Typ) phenotypisch als langsame Metabolisierer (PM-Typ) eingestuft werden. (41) Ein

## Diskussion

Mismatch zwischen Genotyp und klinischer Präsentation bzw. Phenotyp liegt hier vor. Ein Plasmaspiegelverhältnis der O-desmethylvenlafaxin/Venlafaxin Konzentration  $< 1$  zeigt eine „Schwäche“ des CYP2D6-Enzyms, wie bei Patienten mit genetischem PM-Status, oder als Phenoconversion nach Einnahme von starken CYP2D6-Inhibitoren. (42) Venlafaxin zeigt leichte CYP2D6-hemmende Eigenschaften. So zeigen 21% der Patienten mit genetischem EM-Status nach Einnahme von Venlafaxin einen PM-Status aufgrund von Phenoconversion. (42) Deshalb muss bei den Dosierungsempfehlungen der pharmakogenetischen Daten auch auf eventuelle CYP-hemmende oder CYP-induzierende Medikamente Rücksicht genommen werden.

## 6 Limitationen

Diese Studie hat einige Limitationen, die genannt werden müssen. Da in der GAP-DRG Datenbank nur Rezepte, welche von der Krankenkasse übernommen wurden, eingetragen sind, werden viele OTC-Produkte (rezeptfreie Medikamente) nicht erfasst (zB viele Schmerzmittel). Außerdem werden in dieser Datenbank keine Medikamente erfasst, welche in Krankenhäusern verabreicht wurden. Besonders Östrogenhaltige Arzneimittel werden in den Daten deutlich in ihrer Verbreitung unterschätzt, da die Pille als Verhütungsmittel keine Kassenleistung und deshalb auch nicht im Datensatz enthalten ist.

Das Datum der Rezepte wird bei Einlösung bei der Apotheke eingetragen und muss nicht mit dem ersten Einnahme-Datum übereinstimmen. Außerdem könnte der Arzt bei der Verschreibung eines zweiten Medikaments, welches nicht zusammen verschrieben werden sollte, das erste Medikament absetzen, bevor die Packung zu Ende ist. Die Reichweiteninformation über die DDD ist ein Durchschnittswert und deshalb nicht sehr genau. Außerdem enthalten ein Viertel der Rezepte keine Reichweiteninformation, wodurch der durchschnittliche Wert (30 Tage) angenommen wurde.

## 7 Danksagung

Wir danken dem Hauptverband und speziell Gottfried Endel für die Zugriffserlaubnis auf ihre Forschungsdatenbank. Außerdem danke ich DI Dr. Christoph Rinner und Ao.Univ.-Prof., Mag. Dr. Walter Gall für die großartige Unterstützung. Insbesondere beim Thema Arzneimittelinteraktionen konnten sie mir mit ihrer Expertise helfen, das komplexe Thema zu bearbeiten. Auch ein großes Dankeschön an Dr. Kathrin Blagec, welche insbesondere mit ihrem Know-how zu Pharmakogenetik sehr weiterhelfen konnte. Außerdem danke ich meiner Freundin Sophie, welche mir während meiner Arbeit an der Diplomarbeit moralisch beigestanden ist. Ebenfalls einen großen Dank an meine Eltern, die mich bei meinen Werdegang immer unterstützt haben.

Ein großes Dankeschön auch meinem Betreuer Ass.-Prof. Mag. Dr. Matthias Samwald bei der Themenvergabe und dem schnellen Beantworten meiner Fragen während des Entwicklungsprozesses der Diplomarbeit.

## 8 Literaturverzeichnis

1. Vesell ES. Pharmacogenetic perspectives gained from twin and family studies. *Pharmacol Ther.* 1989;41(3):535–52.
2. Van Driest SL, Shi Y, Bowton EA, Schildcrout JS, Peterson JF, Pulley J, u. a. Clinically actionable genotypes among 10,000 patients with preemptive pharmacogenomic testing. *Clin Pharmacol Ther.* April 2014;95(4):423–31.
3. Pulley JM, Denny JC, Peterson JF, Bernard GR, Vnencak-Jones CL, Ramirez AH, u. a. Operational implementation of prospective genotyping for personalized medicine: The design of the Vanderbilt PREDICT project. *Clin Pharmacol Ther.* Juli 2012;92(1):87–95.
4. Samwald M, Adlassnig K-P. Pharmacogenomics in the pocket of every patient? A prototype based on quick response codes. *J Am Med Inform Assoc JAMIA.* Mai 2013;20(3):409–412.
5. Aktories K, Förstermann U, Hofmann F, Starke K. Allgemeine und spezielle Pharmakologie und Toxikologie. 10., übera. München: Elsevier, Urban & Fischer; 2009. 32 ff.
6. Mega JL, Close SL, Wiviott SD, Shen L, Hockett RD, Brandt JT, u. a. Cytochrome P-450 Polymorphisms and Response to Clopidogrel. *N Engl J Med.* 22. Jänner 2009;360(4):354–62.
7. Borst P, Evers R, Kool M, Wijnholds J. A family of drug transporters: the multidrug resistance-associated proteins. *J Natl Cancer Inst.* 16. August 2000;92(16):1295–302.
8. Choo EF, Leake B, Wandel C, Imamura H, Wood AJ, Wilkinson GR, u. a. Pharmacological inhibition of P-glycoprotein transport enhances the distribution of HIV-1 protease inhibitors into brain and testes. *Drug Metab Dispos Biol Fate Chem.* Juni 2000;28(6):655–60.
9. Brinkmann U, Roots I, Eichelbaum M. Pharmacogenetics of the human drug-transporter gene MDR1: impact of polymorphisms on pharmacotherapy. *Drug Discov Today.* 15. August 2001;6(16):835–9.
10. Thiebaut F, Tsuruo T, Hamada H, Gottesman MM, Pastan I, Willingham MC. Cellular localization of the multidrug-resistance gene product P-glycoprotein in normal human tissues.

Proc Natl Acad Sci U S A. November 1987;84(21):7735–8.

11. Breitenstein B, Scheuer S, Brückl TM, Meyer J, Ising M, Uhr M, u. a. Association of ABCB1 gene variants, plasma antidepressant concentration, and treatment response: Results from a randomized clinical study. J Psychiatr Res. Februar 2016;73:86–95.
12. Whirl-Carrillo M, McDonagh E, Hebert J, Gong L, Sangkuhl K, Thorn C, u. a. Pharmacogenomics Knowledge for Personalized Medicine. Clin Pharmacol Ther. Oktober 2012;92(4):414–7.
13. Swen JJ, Wilting I, de Goede AL, Grandia L, Mulder H, Touw DJ, u. a. Pharmacogenetics: from bench to byte. Clin Pharmacol Ther. Mai 2008;83(5):781–7.
14. Swen JJ, Nijenhuis M, de Boer A, Grandia L, Maitland-van der Zee AH, Mulder H, u. a. Pharmacogenetics: from bench to byte--an update of guidelines. Clin Pharmacol Ther. Mai 2011;89(5):662–73.
15. Ross CJD, Visscher H, Sistonen J, Brunham LR, Pussegoda K, Loo TT, u. a. The Canadian Pharmacogenomics Network for Drug Safety: a model for safety pharmacology. Thyroid Off J Am Thyroid Assoc. Juli 2010;20(7):681–7.
16. Pirmohamed M, James S, Meakin S, Green C, Scott AK, Walley TJ, u. a. Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients. BMJ. 3. Juli 2004;329(7456):15–9.
17. McLachlan CYL, Yi M, Ling A, Jardine DL. Adverse drug events are a major cause of acute medical admission. Intern Med J. Juli 2014;44(7):633–8.
18. Austria Codex Apotheker-Verlag [Internet]. Apo-direkt. 2016 [zitiert 14. Jänner 2016]. Verfügbar unter: <http://www3.apoverlag.at/>
19. ANSM. französische Interaktionsliste - Agence nationale de securite du medicament et des produits de sante (ANSM) [Internet]. 2016 [zitiert 20. April 2016]. Verfügbar unter: [http://ansm.sante.fr/Dossiers/Interactions-medicamenteuses/Interactions-medicamenteuses/\(offset\)/0](http://ansm.sante.fr/Dossiers/Interactions-medicamenteuses/Interactions-medicamenteuses/(offset)/0)
20. Shah NR, Seger AC, Seger DL, Fiskio JM, Kuperman GJ, Blumenfeld B, u. a. Improving Acceptance of Computerized Prescribing Alerts in Ambulatory Care. J Am Med Inform Assoc JAMIA. 2006;13(1):5–11.

21. Taylor LK, Tamblyn R. Reasons for physician non-adherence to electronic drug alerts. *Stud Health Technol Inform.* 2004;107(Pt 2):1101–5.
22. Sittig DF, Wright A, Simonaitis L, Carpenter JD, Allen GO, Doebbeling BN, u. a. The state of the art in clinical knowledge management: An inventory of tools and techniques. *Int J Med Inf.* Jänner 2010;79(1):44–57.
23. Vitry AI. Comparative assessment of four drug interaction compendia. *Br J Clin Pharmacol.* Juni 2007;63(6):709–14.
24. Abarca J, Malone DC, Armstrong EP, Grizzle AJ, Hansten PD, Van Bergen RC, u. a. Concordance of severity ratings provided in four drug interaction compendia. *J Am Pharm Assoc JAPhA.* April 2004;44(2):136–41.
25. Wido. ATC-Index 2007 [Internet]. 2016 [zitiert 2. Februar 2016]. Verfügbar unter: [http://www.wido.de/fileadmin/wido/downloads/pdf\\_arzneimittel/wido\\_arz\\_amtl\\_atc-index\\_0107.pdf](http://www.wido.de/fileadmin/wido/downloads/pdf_arzneimittel/wido_arz_amtl_atc-index_0107.pdf)
26. Kuch W, Rinner C, Gall W, Samwald M. How Many Patients Could Benefit From Pre-emptive Pharmacogenomic Testing and Decision Support? A Retrospective Study Based on Nationwide Austrian Claims Data. *Stud Health Technol Inform.* 2016;223:253–8.
27. WHO - ATC/DDD Index [Internet]. ATC/DDD Index. 2016 [zitiert 14. Jänner 2016]. Verfügbar unter: [https://www.whocc.no/atc\\_ddd\\_index/](https://www.whocc.no/atc_ddd_index/)
28. Nancy Van Ness. französische Interaktionsliste [Internet]. 2016 [zitiert 20. April 2016]. Verfügbar unter: <http://opensourcevista.net/NancysVistAServer/FrenchDrugInteractionsTranslatedByNancyVanNess/>
29. Rinner C, Sauter SK, Neuhofer LM, Edlinger D, Grossmann W, Wolzt M, u. a. Estimation of Severe Drug-Drug Interaction Warnings by Medical Specialist Groups for Austrian Nationwide eMedication. *Appl Clin Inform.* 2. Juli 2014;5(3):603–11.
30. Credible Meds [Internet]. Credible Meds. 2016 [zitiert 14. Jänner 2016]. Verfügbar unter: <https://crediblemeds.org>
31. Gynmed. Verhütungsreport 2015 Gynmed [Internet]. Verhütungsreport. 2017 [zitiert

14. Jänner 2016]. Verfügbar unter: <http://verhuetungsreport.at/sites/verhuetungsreport.at/files/2015/gynmed.pdf>
32. Bundesministerium für Bildung und Frauen. Gender Index 2015 [Internet]. Gender Index 2015. 2017 [zitiert 14. Jänner 2016]. Verfügbar unter: [https://www.bmb.gv.at/frauen/gender/gender\\_index\\_2015.pdf?5oc6ap](https://www.bmb.gv.at/frauen/gender/gender_index_2015.pdf?5oc6ap)
33. Ubiquitous Pharmacogenomics (P-PGx) [Internet]. U-PGx. 2017 [zitiert 14. Jänner 2016]. Verfügbar unter: <http://upgx.eu>
34. Cecchin E, Roncato R, Guchelaar HJ, Toffoli G, and for the Ubiquitous Pharmacogenomics Consortium null. Ubiquitous Pharmacogenomics (U-PGx): The time for implementation is now. An Horizon2020 program to drive pharmacogenomics into clinical practice. *Curr Pharm Biotechnol*. 02 2017;
35. van der Wouden CH, Cambon-Thomsen A, Cecchin E, Cheung K-C, Lucía Dávila-Fajardo C, Deneer VH, u. a. Implementing Pharmacogenomics in Europe: Design and Implementation Strategy of the Ubiquitous Pharmacogenomics Consortium. *Clin Pharmacol Ther*. 27. Dezember 2016;
36. Borders-Hemphill V. Concurrent use of statins and amiodarone. *Consult Pharm J Am Soc Consult Pharm*. Mai 2009;24(5):372–9.
37. Marot A, Morelle J, Chouinard VA, Jadoul M, Lambert M, Demoulin N. Concomitant use of simvastatin and amiodarone resulting in severe rhabdomyolysis: a case report and review of the literature. *Acta Clin Belg*. April 2011;66(2):134–6.
38. Ricaurte B, Guirguis A, Taylor HC, Zabriskie D. Simvastatin-amiodarone interaction resulting in rhabdomyolysis, azotemia, and possible hepatotoxicity. *Ann Pharmacother*. April 2006;40(4):753–7.
39. Endel G.; Neumann K.; Boubela R.; Kalcher K.; Nell G. Clinical Relevance of pharmacological interactions [Internet]. 2016 [zitiert 14. Jänner 2017]. Verfügbar unter: <http://www.hauptverband.at/cdscontent/load?contentid=10008.564517&version=1391184568>
40. Neuvonen PJ, Backman JT, Niemi M. Pharmacokinetic comparison of the potential over-the-counter statins simvastatin, lovastatin, fluvastatin and pravastatin. *Clin Pharmacokinet*. 2008;47(7):463–74.



## Literaturverzeichnis

41. Shah RR, Smith RL. Addressing phenoconversion: the Achilles' heel of personalized medicine. *Br J Clin Pharmacol*. Februar 2015;79(2):222–40.
42. de Leon J. Phenoconversion and therapeutic drug monitoring. *Br J Clin Pharmacol*. Oktober 2015;80(4):777–8.

## 9 Tabellenverzeichnis

Tabelle 2-1: Arzneimittel und Prozent der Studienpopulation mit relevantem genetischen Profil .....	7
Tabelle 2-2: Arzneimittel-Metabolisierungs Kategorien .....	8
Tabelle 2-3: Klassifikation der klinischen Relevanz(13).....	12
Tabelle 3-1: Alle Drug-Gene Paare in den DPWG Guidelines, welche in der Liste "Hoch-signifikant" aufgenommen wurden.....	18
Tabelle 3-2: Aufbau eines Rezeptes für die Auswertung .....	21
Tabelle 4-1: Anzahl der Patienten (absolut und relativ zur Altersgruppen) mit mindestens einem PGx Medikament von den Listen DPWG und "Hoch-signifikant"; gruppiert nach den Altersgruppen .....	26
Tabelle 4-2: Anzahl der PGx-Medikamente der DPWG-Liste mit der Anzahl der Personen mit der relativen Anzahl von Personen, welche die Anzahl der Medikamente erhalten .....	26
Tabelle 4-3: Häufigste verschriebene Medikamente der Altergruppe $\geq 65$ der DPWG-Liste und relative Anzahl dieser Altersgruppenpopulation. In Klammer die relative Anzahl der Personen, die mind. ein DPWG-Medikament erhalten.....	27
Tabelle 4-4: Anzahl der PGx-Medikamente der „Hoch-signifikant“-Liste mit der Anzahl der Personen mit der relativen Anzahl von Personen, welche die Anzahl der Medikamente erhalten .....	27
Tabelle 4-5: Häufigste verschriebene Medikamente der Altergruppe $\geq 65$ der „Hoch-signifikant“-Liste und relative Anzahl dieser Altersgruppenpopulation. In Klammer die relative Anzahl der Personen, die mind. ein „Hoch-signifikant“-Medikament erhalten. ....	28
Tabelle 4-6: Häufigste verschriebene Medikamente der Altergruppe 40-64 der „Hoch-signifikant“-Liste und relative Anzahl dieser Altersgruppenpopulation. In Klammer die relative Anzahl der Personen, die mind. ein „Hoch-signifikant“-Medikament erhalten. ....	28
Tabelle 4-7 : Die zehn häufigsten verschriebenen Arzneimittel von der DPWG- und "Hoch-signifikant"-Liste und die % der inkludierten Patienten, welche das Arzneimittel in dieser Studienperiode erhielten .....	29
Tabelle 4-8: Prozent der Gesamtpopulation in der "Hoch-signifikant"-Gruppe von allen Altersgruppen und der Altersgruppe $\geq 65$ .....	30
Diplomarbeit .....	66

## Tabellenverzeichnis

Tabelle 4-9: die drei CYPs mit deren Medikamenten und Anzahl der Personen, welche es min. einmal erhalten haben .....	31
Tabelle 4-10: Die drei CYPs mit deren zugeordneten Medikamenten und Anzahl der Personen >= 65 Jahre, welche das jeweilige Medikament min. einmal erhalten haben .....	33
Tabelle 4-11: Gesamtübersicht der Interaktionslisten (Alerts ohne Mehrfachalerts + Alerts mit Mehrfachalerts).....	34
Tabelle 4-12: häufigste Interaktionspaare der franz. Interaktionsliste.....	34
Tabelle 4-13: häufigste Medikamente in der franz. Interaktionsliste, welche Alerts ausgelöst hat .....	35
Tabelle 4-14: häufigste Interaktionspaare der Austria Codex Interaktionsliste.....	35
Tabelle 4-15: häufigste Medikamente der Austria Codex Interaktionsliste .....	36
Tabelle 4-16: häufigste Interaktionspaare der Austria Codex Interaktionsliste (nur schwere Interaktionen) .....	37
Tabelle 4-17: häufigste Medikamente der Austria Codex Interaktionsliste (nur schwere Interaktionen) .....	37
Tabelle 4-18: häufigste Interaktionsspaare der ONC high priority Interaktionsliste.....	38
Tabelle 4-19: Häufigste Medikamente, welche einen Alert ausgelöst haben, in der ONC high priority Liste .....	39
Tabelle 4-20: Die Interaktionspaar-Gruppen mit deren Anzahl an Interaktionspaaren und verschiedenen ATC-Codes, sortiert nach der Anzahl der Alerts.....	39
Tabelle 4-21: Zehn häufigste Interaktionspaare, welche in allen 3 Interaktionslisten vorkommen, geordnet nach Alerts .....	40
Tabelle 4-22: Anzahl der Alerts pro Interaktionsliste von Antidepressiva.....	42
Tabelle 4-23: Häufigste drei Alert-auslösende Medikamente für die untersuchten Listen .....	42
Tabelle 4-24: Häufigste drei Interaktions-Paare in den 3 (+1) Listen und deren Alerts.....	43
Tabelle 4-25: Gesamtübersicht der Interaktionslisten (Alerts ohne Mehrfachwarnungen + Alerts mit Mehrfachalerts) .....	44
Tabelle 4-26: häufigste Interaktionspaare der franz. Interaktionsliste (ohne Reichweiteninf.)	44
Tabelle 4-27: häufigste Medikamente der französischen Interaktionsliste.....	44
Tabelle 4-28: häufigste Interaktionspaare der Austria Codex Interaktionsliste (ohne Reichweiteninf.) .....	45
Tabelle 4-29: häufigste Medikamente der Austria Codex Interaktionsliste (ohne Reichweiteninf.) .....	46

## Tabellenverzeichnis

Tabelle 4-30: häufigste Interaktionspaare der Austria Codex Interaktionsliste (nur schwere Interaktionen, ohne Reichweiteninf.) .....	47
Tabelle 4-31: häufigste Medikamente der Austria Codex Interaktionsliste (nur schwere Interaktionen, ohne Reichweiteninf.) .....	47
Tabelle 4-32: häufigste Interaktionspaare der ONC high priority Interaktionsliste (ohne Reichweiteninf.) .....	48
Tabelle 4-33: häufigste Medikamente der ONC high priority Interaktionsliste (ohne Reichweiteninf.) .....	48
Tabelle 4-34: Die Interaktionspaar-Gruppen mit deren Anzahl an Interaktionspaaren und verschiedenen ATC-Codes, sortiert nach der Anzahl der Alerts (ohne Reichweiteninf.) .....	49
Tabelle 4-35: häufigste Interaktionspaare, welche in allen 3 Interaktionslisten vorkommen, geordnet nach Alerts .....	50
Tabelle 4-36: Anzahl der Alerts pro Interaktionsliste von Antidepressiva (ohne Reichweiteninf.) .....	53
Tabelle 4-37: häufigste drei Medikamente in den 3 (+1) Listen und deren Alerts (ohne Reichweiteninf.) .....	53
Tabelle 4-38: häufigste drei Interaktions-Paare der 3 (+1) Listen und deren Alerts (ohne Reichweiteninf.) .....	54
Tabelle 5-1: Annahme des Break-Even Points der Genotypisierung von CYP2C19 .....	56

## 10 Abbildungsverzeichnis

Abbildung 3.2-1: Datenstruktur der Tabellen.....	19
Abbildung 3.3-1: SQL-Statement zur Extrahierung der notwendigen Daten aus der Datenbank .....	20
Abbildung 3.3-2: Workflow des Interaktions-Programms .....	23
Abbildung 3.3-3: Eingabemaske des Interaktionsprogramms für spezielle Auswertungen .....	24
Abbildung 3.3-4: Ausschnitt der Ergebnis-Datei.....	24
Abbildung 4.1-1: Vergleich der DPWG- und "Hoch-signifikant"-Liste mit der Anzahl der Personen älter als $\geq 65$ , welche Arzneimittel erhalten haben. ....	30
Abbildung 4.1-2: Vergleich der Gesamtpopulation und der Altersgruppe $\geq 65$ mit der "Hoch- sigifikant"-Liste und drei CYPs.....	31
Abbildung 4.2-1: Interaktionen im Vergleich von den drei Listen inkl. der Austria-Liste mit ausschließlich schweren Interaktionen .....	41
Abbildung 4.2-2: Vergleich der Alerts der 4 Listen mit und ohne Reichweiteninformation ...	51
Abbildung 4.2-3: Anzahl der Alerts pro Patient je Liste im Vergleich mit/ohne Reichweiteninformation .....	52
Abbildung 4.2-4: Vergleich der 4 häufigsten Medikamentengruppen mit der Anzahl der Alerts pro Interaktionsliste .....	52

# 11 Liste der verwendeten Abkürzungen

- **GAP-DRG:** General Approach for Patient-oriented Outpatient-based Diagnosis Related Groups
- **ATC:** Anatomisch-Therapeutisch-Chemisches Klassifikationssystem für Arzneistoffe
- **ONC:** U.S. Office of National Coordinator
- **DDI:** Arzneimittelinteraktion (Drug-Drug-Interaction)
- **PharmGKB:** Pharmacogenomics Knowledgebase
- **PGRN:** Pharmacogenomics Research Network
- **CPIC:** Clinical Pharmacogenetics Implementation Consortium
- **DPWG:** Dutch Pharmacogenetics Working Group
- **KNMP:** Royal Dutch Pharmacist's Association
- **ANSM:** Agence nationale de sécurité du médicament et des produits de santé
- **CPNDS:** Canadian Pharmacogenomics Network for Drug Safety
- **CPOE:** Computerized Physician Order Entry
- **PGx-Drugs:** Medikamente, welche in pharmakogenetischen Guidelines vorkommen
- **OTC-Produkte:** Over the Counter. Arzneimittel, welche rezeptfrei erhältlich sind
- **EHR:** Electronic Health Record (elektronische Krankenakte)
- **CDS:** Clinical decision support. Ein System, welches Mediziner in klinischen Entscheidungen helfen soll.
- **QR-Code:** Quick response-Code. Ein grafischer Code zur Verpackung von Informationen in (meist) ein Rechteck.
- **CYP:** Cytochrome P450. Hämproteine mit enzymatischer Aktivität.
- **UAE:** Unerwünschtes Arzneimittelereignis
- **UAW:** Unerwünschte Arzneimittelwirkung
- **XML:** Extensible Markup Language. Eine Auszeichnungssprache, die zur Darstellung von Daten in Form von Textdateien verwendet wird. Wird häufig zum maschinenlesbaren Austausch von Daten verwendet.
- **CSV:** Comma-separated values. Datenspeicherung, in welche Diese mit einem Komma getrennt gespeichert werden.

# 12 Anhang

## 12.1 Interaktionsprogramm-Quellcode

### 12.1.1 Klasse DDI\_High\_Memory

```
package ddi_high_memory;

import ddi_high_memory.interactions.InteractionATC;
import ddi_high_memory.interactions.InteractionList;
import ddi_high_memory.interactions.Receipt;
import ddi_high_memory.interactions.Patient;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.lang.Thread.sleep;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collection;
import java.util.HashSet;
import java.util.Iterator;
import java.util.concurrent.ConcurrentLinkedQueue;

/**
 *
 * @author Wolfgang Kuch
 */
public class DDI_High_Memory {

    public final static boolean LOW_MEMORY = false;
    public static ArrayList<InteractionList> interactlist = new ArrayList<InteractionList>();
    public static ArrayList<ArrayList<InteractionList>> interact = new ArrayList<ArrayList<InteractionList>>();

    /**
     * @param args 0: Liste mit den Rezepten, 1-n: Interaktionslisten, auf
     * welche geprüft werden soll
     * @throws java.lang.ClassNotFoundException
     * @throws java.sql.SQLException
     * @throws java.io.FileNotFoundException
     * @throws java.text.ParseException
     * @throws java.lang.InterruptedException
     */
    public static void main(String[] args) throws ClassNotFoundException, SQLException, FileNotFoundException, ParseException, IOException,
    InterruptedException {
        int cores = (int) (Runtime.getRuntime().availableProcessors());
        //cores = 2;
        System.out.println("Anzahl Threads: " + cores);

        System.out.println("Patientenliste: " + args[0]);
        System.out.println("Interaktionsliste(n): ");
        for (int i = 1; i < args.length; i++) {
            System.out.println(" " + args[i]);
        }

        File f = new File(args[0]);
        BufferedReader br = new BufferedReader(new FileReader(f));

        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

        String line = br.readLine(); //Erste Zeile werfen (Tabellenüberschriften)
        String[] splitline = null;

        long pat_id = -1;

        Patient p = null;
        Receipt r = null;

        /**
         * Interaktionslisten aus den Dateien laden
         */
        for (int i = 1; i < args.length; i++) {
```

## Anhang

```
ArrayList<InteractionATC> a = getInteractionATCfromFile(new File(args[i]));
interactlist.add(new InteractionList(args[i], a));
}

/**
 * Jeder Thread erhält eigene Interaktionslisten
 */
for (int i = 0; i < cores; i++) {
    ArrayList<InteractionList> temp = copyArrayList(interactlist);
    interact.add(temp);
}
System.out.println("Interaktionslisten geladen");

long anzahlzeilen = f.length() / 35;
f = null;

System.out.println("Geschätzte Zeilen in der Patienten-Datei: " + anzahlzeilen); //Schätzung der Anzahl der Zeilen, die im File enthalten sind

/**
 * Erzeugung und Initialisierung der Interaktions-Threads
 */
Interaction_Thread[] workingthreads = new Interaction_Thread[cores];
System.out.println("Initialisiere Auswertungs-Threads");

ConcurrentLinkedQueue<Patient> patientqueue = new ConcurrentLinkedQueue<Patient>();

for (int i = 0; i < cores; i++) {
    workingthreads[i] = new Interaction_Thread(patientqueue, interact.get(i));
    workingthreads[i].setDaemon(true);
    workingthreads[i].setPriority(Thread.MAX_PRIORITY - 1);
    workingthreads[i].start();
}

Runtime.getRuntime().gc();

/**
 * Auswertungs-Schleife
 */
System.out.println("Auswertung wird begonnen");

long patientenanzahl = 0;
while ((line = br.readLine()) != null) {

    //Patient laden
    splitline = line.split(";");
    Long id = Long.parseLong(splitline[0]);
    //Wenn ein neuer Patient geladen wurde, wird der Alte zur Auswertung in die Queue gegeben
    if (pat_id != id) {
        if (pat_id != -1) {
            p.sortlist();
            patientqueue.add(p);
            patientenanzahl++;
            if (LOW_MEMORY) {
                if (Runtime.getRuntime().freeMemory() < 500000000) {
                    Thread.sleep(10000);
                }
            }
        }
        pat_id = id;
        p = new Patient(pat_id);
    }

    //Rezepte laden
    r = new Receipt();

    //ATC
    r.setAte(splitline[1].trim());
    if (splitline[1].length() != 7) {
        System.err.println(splitline[1]);
    }

    //DATUM
    r.setEindat(format.parse(splitline[2]).getTime());

    //DDD UND TED
    try {
        r.setDdd(Double.parseDouble(splitline[3]));
    } catch (NumberFormatException e) {
        r.setDdd(30); //Bei leerem TED wird standardmäßig 30 Tage angenommen
    } catch (ArrayIndexOutOfBoundsException e) {
        r.setDdd(30);
    }
    try {
        r.setVoanz(Long.parseLong(splitline[4]));
    } catch (NumberFormatException e) {
        r.setVoanz(1); //Standard bei leerer Packungsanzahl = 1
    } catch (ArrayIndexOutOfBoundsException e) {

```



## Anhang

```
r.setVoanz(1);
}
p.addReceipt(r);
}
//Letzten Patienten auch in Queue hinzufügen
p.sortlist();
patientqueue.add(p);
patientenanzahl++;

br.close();

/**
 * Warten, dass alle Patienten ausgewertet werden
 */
System.out.println("Alle Daten in Arbeitsspeicher geladen. Bitte warten");
while (true) {
    System.out.println("Noch " + patientqueue.size() + " Patienten übrig");
    if (patientqueue.isEmpty()) {
        System.out.println("Auswertung beendet. Bitte warten");
        sleep(60000);
        for (Interaction_Thread t : workingthreads) {
            t.interrupt();
        }

        break;
    }
    sleep(60000);
}

System.out.println("Interaktions-Daten werden erstellt");

try {
    //interactlist wird einfach bearbeitet
    for (int i = 0; i < interact.size(); i++) { //i = Nummer der Threads
        for (int j = 0; j < interact.get(i).size(); j++) { //j = Welche InteraktionsListe wird gezählt?
            ArrayList<InteractionATC> l = interact.get(i).get(j).getInteractions();
            for (int k = 0; k < l.size(); k++) { //k = Einzelnen Interaktionen in der Liste durchgehen
                InteractionATC iatc = interactlist.get(j).getInteractions().get(k); //Diese Interaktion wird jetzt bearbeitet
                iatc.addCountDoubles(l.get(k).getCount_without_doubles());
                iatc.addCount(l.get(k).getCount());

                if (iatc.atcs == null && l.get(k).atcs != null) {
                    System.err.println(iatc.toString());
                    System.err.println(l.get(k).toString());
                    System.err.println(l.get(k).atcs.toString());
                }

                if (l.get(k).atcs != null) {
                    if (!l.get(k).atcs.isEmpty()) {
                        //Unterliste der InteractionATCs, wenn ATC-Code kleiner 7 zusammenzählen
                        for (Iterator it = l.get(k).atcs.iterator(); it.hasNext(); ) {
                            InteractionATC atc = (InteractionATC) it.next(); //ATC-Code von der Unterliste
                            //Kann man es hinzufügen, oder ist es schon in der neuen Liste vorhanden (dann addieren)
                            if (!iatc.atcs.add(atc)) {
                                //Nach dem schon vorhandenen suchen
                                for (Iterator it2 = iatc.atcs.iterator(); it2.hasNext(); ) {
                                    InteractionATC atc2 = (InteractionATC) it2.next();
                                    if (atc2.equals(atc)) {
                                        atc2.addCount(atc.count);
                                        atc2.addCountDoubles(atc.count_without_doubles);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
} catch (NullPointerException ex) {
    System.err.println(ex.toString());
    System.err.println();
    System.err.println(ex.getMessage());
    throw new NullPointerException();
}

System.out.println("Interaktionspaar-Objekte werden gespeichert");
File file = new File("interaktions_paare_objects_french_sameday.data");
ObjectOutputStream os = new ObjectOutputStream(new FileOutputStream(file));

os.writeLong(patientenanzahl);

os.writeInt(interactlist.size());
for (int i = 0; i < interactlist.size(); i++) {
    os.writeObject(interactlist.get(i));
}
```

## Anhang

```
os.close();

System.out.println("Interaktionspaar-Objekte gespeichert");
System.out.println("Programm beendet");
}

/**
 * Läd die Interaktionen, welcher sich in dem File befinden und gibt eine
 * ArrayList mit InteractionATC-Objekten zurück. Doppelte Interaktionen
 * werden ignoriert.
 *
 * @param file Datei, in welcher sich die Interaktionen befinden
 * @return ArrayList mit InteractionATC-Objekten
 * @throws FileNotFoundException
 * @throws IOException
 */
private static ArrayList<InteractionATC> getInteractionATCfromFile(File file) throws FileNotFoundException, IOException {
    Collection<InteractionATC> interactions = new HashSet<InteractionATC>();

    FileReader fr = new FileReader(file);
    BufferedReader br = new BufferedReader(fr);

    String str;
    String[] atcs;
    while ((str = br.readLine()) != null) {
        atcs = str.split(",");

        //Interaktionen mit sich selber ausschließen
        if (atcs[0].equals(atcs[1])) {
            continue;
        }

        //Wenn ATC Code ungültig (die franz. Interaktionsliste enthält dies)
        if (atcs[0].equals("as") || atcs[1].equals("as")) {
            continue;
        }

        /**
         * Aussortieren und Zusammenfassen der Interaktionen
         */
        for (InteractionATC atc : interactions) {
            if (atcs[0].length() == 7 && atcs[1].length() == 7) {
                break;
            }

            if (atcs[0].length() < 7 && atcs[1].length() == 7) {
                if (atc.atc1.contains(atcs[0])) {
                    if (atc.atc2.equals(atcs[1])) {
                        interactions.remove(atc);
                        break;
                    }
                } else if (atc.atc2.contains(atcs[0])) {
                    if (atc.atc1.equals(atcs[1])) {
                        interactions.remove(atc);
                        break;
                    }
                }
            } else if (atcs[0].length() == 7 && atcs[1].length() < 7) {
                if (atc.atc1.contains(atcs[1])) {
                    if (atc.atc2.equals(atcs[0])) {
                        interactions.remove(atc);
                        break;
                    }
                } else if (atc.atc2.contains(atcs[1])) {
                    if (atc.atc1.equals(atcs[0])) {
                        interactions.remove(atc);
                        break;
                    }
                }
            } else if (atcs[0].length() < 7 && atcs[1].length() < 7) {
                if (atc.atc1.contains(atcs[0])) {
                    if (atc.atc2.contains(atcs[1])) {
                        interactions.remove(atc);
                        break;
                    }
                } else if (atc.atc2.contains(atcs[0])) {
                    if (atc.atc1.contains(atcs[1])) {
                        interactions.remove(atc);
                        break;
                    }
                }
            }
        }
    }

    if (atcs[0].length() == 7 || atcs[1].length() == 7) {
        boolean ausstieg = false;
        for (InteractionATC a : interactions) {

```

## Anhang

```
        if ((atcs[0].contains(a.atc1))) {
            if (atcs[1].contains(a.atc2)) {
                ausstieg = true;
                break;
            }
        } else if (atcs[1].contains(a.atc1)) {
            if (atcs[0].contains(a.atc2)) {
                ausstieg = true;
                break;
            }
        }
    }

    if (ausstieg) {
        continue;
    }
}

InteractionATC a = new InteractionATC(atcs[0], atcs[1]);
if (atcs[0].length() < 7 || atcs[1].length() < 7) {
    a.setAtcgeneral(true);
}
interactions.add(a);

}

br.close();
ArrayList<InteractionATC> a = new ArrayList<InteractionATC>(interactions);
a.trimToSize();

return a;
}

public static ArrayList<InteractionList> copyArrayList(ArrayList<InteractionList> list) {
    ArrayList<InteractionList> newlist = new ArrayList<InteractionList>(list.size());

    for (InteractionList l : list) {
        newlist.add(new InteractionList(l));
    }

    return newlist;
}
}
```

## 12.1.2 Klasse Interaction\_Thread

```

package ddi_high_memory;

import ddi_high_memory.interactions.InteractionATC;
import ddi_high_memory.interactions.InteractionList;
import ddi_high_memory.interactions.Receipt;
import ddi_high_memory.interactions.Patient;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import java.util.concurrent.ConcurrentLinkedQueue;

/**
 *
 * @author Wolfgang Kuch
 */
public class Interaction_Thread extends Thread {

    private final ArrayList<InteractionList> interact;
    private final ConcurrentLinkedQueue<Patient> patients;

    public Interaction_Thread(ConcurrentLinkedQueue<Patient> patients, ArrayList<InteractionList> interact) {
        this.patients = patients;
        this.interact = interact;
    }

    @Override
    public void run() {
        Thread.currentThread().setName("Interaction_Thread");

        while (!this.isInterrupted()) {
            //Patient aus Queue laden
            Patient p = patients.poll();

            /**
             * Wenn keine vorhanden, nochmal versuchen und dann 1 sek. warten
             */
            while (p == null) {
                p = patients.poll();

                try {
                    Thread.sleep(1000);
                } catch (InterruptedException ex) {
                }
            }

            ArrayList<Receipt> list = p.receipts;

            for (InteractionList inter : this.interact) {
                HashSet<InteractionATC> doubles = new HashSet<InteractionATC>(30); //mehrfache Interaktionen werden aussortiert

                for (int i = 0; i < list.size(); i++) {
                    Receipt r1 = list.get(i);
                    if (i < (list.size() - 1)) {
                        for (int j = i + 1; j < list.size(); j++) {

                            Receipt r2 = list.get(j);
                            if (r1.atc.equals(r2.atc)) {
                                continue;
                            }

                            //Wenn das Datum vom ersten Rezept + die Reichweite größer ist, als das Datum des zweiten Rezeptes
                            //if ((r1.eindat + r1.ted) >= r2.eindat) {
                            if (r1.eindat == r2.eindat) {
                                if (!doubles.add(new InteractionATC(r1.atc, r2.atc))) { //Wenn diese Interaktion schon überprüft worden ist
                                    checkInteraction(r1, r2, inter.interactions);
                                } else {
                                    checkInteraction(r1, r2, inter.interactions, true);
                                }
                            }
                        }
                    }
                }
            }
        }

        System.err.println(Thread.currentThread().getName() + " beendet");
    }

    /**
     *
     * @param r1
     * @param r2

```

## Anhang

```
* @param interact
* @return
*/
private boolean checkInteraction(Receipt r1, Receipt r2, ArrayList<InteractionATC> interact) {
    for (InteractionATC interaction : interact) {
        if (r1.atc.startsWith(interaction.atc1)) {
            if (r2.atc.startsWith(interaction.atc2)) {
                if (interaction.atcgeneral) {
                    InteractionATC a = new InteractionATC(r1.atc, r2.atc);
                    a.count++;
                    boolean yes = interaction.atcs.add(a);
                    if (!yes) {
                        for (Iterator<InteractionATC> it = interaction.atcs.iterator(); it.hasNext(); ) {
                            InteractionATC at = it.next();
                            if (at.equals(a)) {
                                at.addCount(a.count);
                            }
                        }
                    }
                }
                interaction.count++;
                return true;
            }
        } else if (r1.atc.startsWith(interaction.atc2)) {
            if (r2.atc.startsWith(interaction.atc1)) {
                if (interaction.atcgeneral) {
                    InteractionATC a = new InteractionATC(r1.atc, r2.atc);
                    a.count++;
                    boolean yes = interaction.atcs.add(a);
                    if (!yes) {
                        for (Iterator<InteractionATC> it = interaction.atcs.iterator(); it.hasNext(); ) {
                            InteractionATC at = it.next();
                            if (at.equals(a)) {
                                at.addCount(a.count);
                            }
                        }
                    }
                }
                interaction.count++;
                return true;
            }
        }
    }
    return false;
}

/**
 *
 * @param r1 Rezept 1
 * @param r2 Rezept 2
 * @param interact Interaktionsliste
 * @param dummy
 * @return true, wenn eine Interaktion gefunden wurde
 */
private boolean checkInteraction(Receipt r1, Receipt r2, ArrayList<InteractionATC> interact, boolean dummy) {
    for (InteractionATC interaction : interact) {
        if (r1.atc.startsWith(interaction.atc1)) {
            if (r2.atc.startsWith(interaction.atc2)) {
                if (interaction.atcgeneral) {
                    InteractionATC a = new InteractionATC(r1.atc, r2.atc);
                    a.count++;
                    a.count_without_doubles++;
                    boolean yes = interaction.atcs.add(a);
                    if (!yes) {
                        for (Iterator<InteractionATC> it = interaction.atcs.iterator(); it.hasNext(); ) {
                            InteractionATC at = it.next();
                            if (at.equals(a)) {
                                at.addCount(a.count);
                                at.addCountDoubles(a.count_without_doubles);
                            }
                        }
                    }
                }
                interaction.count++;
                interaction.count_without_doubles++;
                return true;
            }
        } else if (r1.atc.startsWith(interaction.atc2)) {
            if (r2.atc.startsWith(interaction.atc1)) {
                if (interaction.atcgeneral) {
                    InteractionATC a = new InteractionATC(r1.atc, r2.atc);
                    a.count++;
                    a.count_without_doubles++;
                    boolean yes = interaction.atcs.add(a);
                }
            }
        }
    }
}
```

## Anhang

```
        if (!yes) {
            for (Iterator<InteractionATC> it = interaction.atcs.iterator(); it.hasNext(); ) {
                InteractionATC at = it.next();
                if (at.equals(a)) {
                    at.addCount(a.count);
                    at.addCountDoubles(a.count_without_doubles);
                }
            }
        }
        interaction.count++;
        interaction.count_without_doubles++;
        return true;
    }
}

return false;
}
```

## 12.1.3 Klasse InteractionATC

```

package ddi_high_memory.interactions;

import java.io.IOException;
import java.io.Serializable;
import java.util.HashSet;
import java.util.Iterator;

/**
 *
 * @author Wolfi
 */
public class InteractionATC implements Comparable, Serializable {

    public String atc1;
    public String atc2;
    public long count;
    public long count_without_doubles;
    public HashSet<InteractionATC> atcs = null;
    public boolean atcgeneral = false;

    public InteractionATC(String atc1, String atc2) {
        this.atc1 = atc1;
        this.atc2 = atc2;
        this.count = 0;
        this.count_without_doubles = 0;
    }

    public InteractionATC() {
    }

    public InteractionATC(InteractionATC atc) {
        this.atc1 = atc.atc1;
        this.atc2 = atc.atc2;
        this.atcgeneral = atc.atcgeneral;
        this.count = atc.count;
        this.count_without_doubles = atc.count_without_doubles;
        if (atc.atcs != null) {
            this.atcs = new HashSet<InteractionATC>();
            if (!atc.atcs.isEmpty()) {
                for (Iterator it = atc.atcs.iterator(); it.hasNext();) {
                    InteractionATC a = (InteractionATC) it.next();
                    this.atcs.add(new InteractionATC(a));
                }
            }
        }
    }

    public String getAtc1() {
        return atc1;
    }

    public void setAtc1(String atc1) {
        this.atc1 = atc1;
    }

    public String getAtc2() {
        return atc2;
    }

    public void setAtc2(String atc2) {
        this.atc2 = atc2;
    }

    public long getCount() {
        return count;
    }

    public void addCount() {
        this.count++;
    }

    public void addCount(long value) {
        this.count += value;
    }

    public void addCountDoubles() {
        this.count_without_doubles++;
    }

    public void addCountDoubles(long value) {
        this.count_without_doubles += value;
    }
}

```

## Anhang

```
public long getCount_without_doubles() {
    return count_without_doubles;
}

public void setAtcgeneral(boolean atcgeneral) {
    this.atcgeneral = atcgeneral;
    this.atcs = new HashSet<InteractionATC>(10);
}

@Override
public int compareTo(Object o) {
    InteractionATC atc = (InteractionATC) o;

    return Long.compare(this.count, atc.count);
    //return new Integer(count).compareTo(atc.count);
}

@Override
public int hashCode() {
    int hash = 3;
    hash = 97 * hash + (this.atc1 != null ? this.atc1.hashCode() : 0) + (this.atc2 != null ? this.atc2.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final InteractionATC other = (InteractionATC) obj;

    if (!this.atc1.equals(other.atc1)) {
        if (!this.atc1.equals(other.atc2)) {
            return false;
        }
    }
    if (!this.atc2.equals(other.atc2)) {
        if (!this.atc2.equals(other.atc1)) {
            return false;
        }
    }

    return true;
}

@Override
public String toString() {
    return "InteractionATC{" + "atc1=" + atc1 + ", atc2=" + atc2 + ", count=" + count + ", count_without_doubles=" + count_without_doubles + '}';
}

private void writeObject(java.io.ObjectOutputStream out) throws IOException {
    out.writeUTF(this.atc1);
    out.writeUTF(this.atc2);
    out.writeLong(this.count);
    out.writeLong(this.count_without_doubles);
    if (this.atcs == null) {
        out.writeInt(0);
    } else if (this.atcs.isEmpty()) {
        out.writeInt(0);
    } else {
        out.writeInt(this.atcs.size());
        for (InteractionATC atc : this.atcs) {
            out.writeObject(atc);
        }
    }
    out.writeBoolean(this.atcgeneral);
}

private void readObject(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException {
    this.atc1 = in.readUTF();
    this.atc2 = in.readUTF();
    this.count = in.readLong();
    this.count_without_doubles = in.readLong();
    int anz = in.readInt();
    this.atcs = new HashSet<InteractionATC>(anz);
    for (int i = 0; i < anz; i++) {
        this.atcs.add((InteractionATC) in.readObject());
    }
    this.atcgeneral = in.readBoolean();
}
}
```



## 12.1.4 Klasse InteractionList

```

package ddi_high_memory.interactions;

import java.io.IOException;
import java.io.Serializable;
import java.util.ArrayList;

/**
 *
 * @author Wolfi
 */
public class InteractionList implements Serializable {

    private String name;
    public ArrayList<InteractionATC> interactions = null;

    public InteractionList(String name, ArrayList<InteractionATC> list) {
        this.name = name;
        this.interactions = list;
    }

    public InteractionList(InteractionList list) {
        this.name = list.name;
        this.interactions = new ArrayList<InteractionATC>(list.interactions.size());
        if (this.interactions != null) {
            for (InteractionATC atc : list.interactions) {
                this.interactions.add(new InteractionATC(atc));
            }
        }
    }

    public String getName() {
        return name;
    }

    public ArrayList<InteractionATC> getInteractions() {
        return interactions;
    }

    public void setInteractions(ArrayList<InteractionATC> interactions) {
        this.interactions = interactions;
    }

    private void writeObject(java.io.ObjectOutputStream out) throws IOException {
        out.writeUTF(name);
        out.writeLong(interactions.size());
        for (InteractionATC atc : interactions) {
            out.writeObject(atc);
        }
    }

    private void readObject(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException {
        this.name = in.readUTF();
        long anz = in.readLong();
        interactions = new ArrayList<InteractionATC>();

        for (long i = 0; i < anz; i++) {
            interactions.add((InteractionATC) in.readObject());
        }
    }
}

```

## 12.1.5 Klasse Patient

```

package ddi_high_memory.interactions;

import java.io.IOException;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Collections;

/**
 *
 * @author Wolfi
 */
public class Patient implements Serializable {
    private long pers_id;
    public ArrayList<Receipt> receipts = new ArrayList<Receipt>();

    public Patient(long pers_id) {
        this.pers_id = pers_id;
    }

    public Patient() {
    }

    public long getPers_id() {
        return pers_id;
    }

    public ArrayList<Receipt> getReceipts() {
        return receipts;
    }

    public void addReceipt(Receipt r) {
        this.receipts.add(r);
    }

    public void sortlist() {
        Collections.sort(receipts, new Compara());
    }

    private void writeObject(java.io.ObjectOutputStream out) throws IOException {
        out.writeInt((int) this.pers_id);
        out.writeInt(this.receipts.size());
        for(Receipt r : this.receipts) {
            out.writeUTF(r.getAtc());
            out.writeLong(r.getEindat());
            out.writeDouble(r.getDdd());
            out.writeLong(r.getVoanz());
        }
    }

    private void readObject(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException {
        this.pers_id = in.readInt();
        int anz_r = in.readInt();
        this.receipts = new ArrayList<Receipt>();
        for(int i = 0; i < anz_r; i++) {
            Receipt receipt = new Receipt();
            receipt.setAtc(in.readUTF());
            receipt.setEindat(in.readLong());
            receipt.setDdd(in.readDouble());
            receipt.setVoanz(in.readLong());

            this.receipts.add(receipt);
        }
    }

    @Override
    public String toString() {
        return "Patienten-ID: " + this.pers_id + " mit " + this.receipts.size() + " Medikamente";
    }
}

```

## 12.1.6 Klasse Receipt

```

package ddi_high_memory.interactions;

import java.util.Objects;

/**
 *
 * @author Wolfi
 */
public class Receipt implements Comparable<Receipt> {

    public String atc;
    public long eindat;
    public double ddd;
    public long voanz;
    public long ted;

    public String getAtc() {
        return atc;
    }

    public void setAtc(String atc) {
        this.atc = atc;
    }

    public long getEindat() {
        return eindat;
    }

    public void setEindat(long eindat) {
        this.eindat = eindat / 86400000;
    }

    public double getDdd() {
        return ddd;
    }

    public void setDdd(double ddd) {
        this.ddd = ddd;
    }

    public long getVoanz() {
        return voanz;
    }

    public void setVoanz(long voanz) {
        this.voanz = voanz;
        this.ted = ((long) Math.ceil(this.ddd * this.voanz));
    }

    public long getTed() {
        return ted;
    }

    public void setTed(long ted) {
        this.ted = ted;
    }

    @Override
    public String toString() {
        return "Receipt: " + atc + " Datum: " + eindat + " : " + ddd + " : " + voanz;
    }

    @Override
    public int hashCode() {
        int hash = 3;
        hash = 31 * hash + Objects.hashCode(this.atc);
        hash = 31 * hash + (int) (this.eindat ^ (this.eindat >>> 32));
        hash = 31 * hash + (int) (Double.doubleToLongBits(this.ddd) ^ (Double.doubleToLongBits(this.ddd) >>> 32));
        hash = 31 * hash + (int) (this.voanz ^ (this.voanz >>> 32));
        hash = 31 * hash + (int) (this.ted ^ (this.ted >>> 32));
        return hash;
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
    }

```

## Anhang

```
    }  
    final Receipt other = (Receipt) obj;  
    if (((this.atc == null) ? (other.atc != null) : !this.atc.equals(other.atc)) {  
        return false;  
    }  
    if (this.eindat != other.eindat) {  
        return false;  
    }  
    if (Double.doubleToLongBits(this.ddd) != Double.doubleToLongBits(other.ddd)) {  
        return false;  
    }  
    if (this.voanz != other.voanz) {  
        return false;  
    }  
    if (this.ted != other.ted) {  
        return false;  
    }  
    return true;  
}  
  
@Override  
public int compareTo(Receipt o) {  
    return Long.compare(this.eindat, o.getEindat());  
}  
  
}
```

## 12.1.7 Klasse Compara

```
package ddi_high_memory.interactions;

import java.util.Comparator;

/**
 * @author Wolfi
 */
public class Compara implements Comparator<Receipt> {

    @Override
    public int compare(Receipt o1, Receipt o2) {
        if (o1.eindat < o2.eindat) {
            return -1;
        } else if (o1.eindat == o2.eindat) {
            return 0;
        } else {
            return 1;
        }
    }
}
```

## 12.2 Auswertungsprogramm-Quellcode

```

package interactiondata_auswertung;

import ddi_high_memory.interactions.ComparaATC;
import ddi_high_memory.interactions.InteractionATC;
import ddi_high_memory.interactions.InteractionList;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.PrintStream;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.HashSet;
import java.util.LinkedHashMap;
import java.util.LinkedHashSet;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Set;

/**
 *
 * @author Wolfi
 */
public class Interactiondata_Auswertung {

    private static final String FILE_WITHOUT_DOUBLES = "interactionlist_without_doubles.txt";
    private static final String FILE_WITH_DOUBLES = "interactionlist_with_doubles.txt";

    public static void main(String[] args) throws IOException, ClassNotFoundException {
        ArrayList<InteractionList> list = loadInteractionLists("interaktions_paare_objects_new.data");

        System.setOut(new PrintStream(new File("out.txt")));

        ArrayList<InteractionList> l1 = new ArrayList<InteractionList>(list.size() + 100);
        for (InteractionList l : list) {
            l1.add(new InteractionList(l));
        }

        for (InteractionList l : l1) {
            ArrayList<InteractionATC> deletelist = new ArrayList<InteractionATC>();
            ArrayList<InteractionATC> addlist = new ArrayList<InteractionATC>();
            for (InteractionATC atc : l.getInteractions()) {
                int i = 0;
                int j = 0;
                if (!atc.atcs == null) {
                    if (!atc.atcs.isEmpty()) {
                        for (InteractionATC a : atc.atcs) {
                            i += a.count;
                            j += a.count_without_doubles;
                            addlist.add(a);
                        }
                    }
                    if (atc.count != i) {
                        if (atc.count_without_doubles != j) {
                            System.err.println("Nicht gleich!");
                        }
                    }
                    deletelist.add(atc);
                }
            }
            l.getList().removeAll(deletelist);
            l.interactions.addAll(addlist);
        }

        for (InteractionList li : l1) {
            HashSet<String> a = new HashSet<String>(1000);
            int i = 0; //Anzahl der Interaktionspaare
            for (InteractionATC atc : li.getInteractions()) {
                i++;
                a.add(atc.atc1);
                a.add(atc.atc2);
            }

            System.err.println(li.getName());
            System.err.println(" Anzahl der verschiedenen ATC-Codes: " + a.size());
        }
    }
}

```

```

        System.err.println(" Anzahl der verschiedenen Interaktionspaare: " + i);
    }

    getInteractions_without_doubles(list);
    getInteractions_with_doubles(list);
    //getInteractions_without_ted(list);

    getSummary(list);

    mostInteractionATCs_withoutdoubles(list);
    mostInteractionATCs_withdoubles(list);
    //mostInteractionATCs_withoutted(list);

    mostfalsedescribedmedis_without_doubles(list);
    mostfalsedescribedmedis_with_doubles(list);
    //mostfalsedescribedmedis_without_ted(list);

    mostfalsedescribedmedis_without_doubles_third_stage(list);
    mostfalsedescribedmedis_with_doubles_third_stage(list);
    //mostfalsedescribedmedis_without_ted_third_stage(list);

    all_mostfalsedescribedmedis_without_doubles_third_stage(list);
    all_mostInteractionATCs_withoutdoubles(list);
    all_mostfalsedescribedmedis_without_doubles(list);

    all_three_getInteractions_without_doubles(list);

    onlyantidepressiva(list);
    onlyantidepressivamedis(list);

    //savefordatabase(list);
}

private static ArrayList<InteractionList> loadInteractionLists(String datafile) throws IOException, ClassNotFoundException {
    File f = new File(datafile);
    ObjectInputStream os = new ObjectInputStream(new FileInputStream(f));

    long patientenanzahl = os.readLong();
    System.err.println("Patientenanzahl: " + patientenanzahl);
    int len = os.readInt();

    ArrayList<InteractionList> list = new ArrayList<InteractionList>(len);
    for (int i = 0; i < len; i++) {
        InteractionList l = (InteractionList) os.readObject();
        l.getInteractions().sort(new ComparaATC());
        list.add(l);
    }

    os.close();

    return list;
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void getInteractions_without_doubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File(FILE_WITHOUT_DOUBLES);
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));

    for (InteractionList l : list) {
        bw.write(l.getName());
        bw.newLine();
        for (InteractionATC atc : l.getInteractions()) {
            bw.write(atc.getAtc1() + ":" + atc.getAtc2() + ";" + atc.getCount_without_doubles());
            bw.newLine();
        }
    }

    bw.close();
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void getInteractions_with_doubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File(FILE_WITH_DOUBLES);
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));

    for (InteractionList l : list) {
        bw.write(l.getName());
        bw.newLine();
        for (InteractionATC atc : l.getInteractions()) {

```

## Anhang

```
        bw.write(atc.getAtc1() + ":" + atc.getAtc2() + ";" + atc.getCount());
        bw.newLine();
    }
}

bw.close();
}

/**
 *
 * @param list
 */
private static void getSummary(ArrayList<InteractionList> list) {
    long alerts = 0;

    System.out.println("Interactionlist without doubles");
    for (InteractionList l : list) {
        alerts = 0;
        System.out.print(l.getName() + ":");
        for (InteractionATC atc : l.getInteractions()) {
            alerts += atc.getCount_without_doubles();
        }
        System.out.println(alerts);
    }

    System.out.println("Interactionlist with doubles");
    for (InteractionList l : list) {
        alerts = 0;
        System.out.print(l.getName() + ":");
        for (InteractionATC atc : l.getInteractions()) {
            alerts += atc.getCount();
        }
        System.out.println(alerts);
    }
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void mostInteractionATCs_withoutdoubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File("mostInteractionATCs_without_doubles.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    for (InteractionList l : list) {
        bw.write(l.getName());
        bw.newLine();

        newInteractionList();
        ArrayList<InteractionATC> alist = l.getInteractions();
        alist.sort(new ComparaATC());
        for (int i = 0; i < alist.size(); i++) {

            bw.write(alist.get(i).getAtc1() + ":" + alist.get(i).getAtc2() + ";" + alist.get(i).getCount_without_doubles());
            bw.newLine();

        }
    }
    bw.close();
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void mostInteractionATCs_withdoubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File("mostInteractionATCs_with_doubles.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    for (InteractionList l : list) {
        bw.write(l.getName());
        bw.newLine();
        newInteractionList();
        ArrayList<InteractionATC> alist = l.getInteractions();
        alist.sort(new ComparaATC());
        for (int i = 0; i < alist.size(); i++) {
            bw.write(alist.get(i).getAtc1() + ":" + alist.get(i).getAtc2() + ";" + alist.get(i).getCount());
            bw.newLine();
        }
    }
    bw.close();
}

/**
 *
 * @param list
 * @throws IOException
 */
```



## Anhang

```
*/
private static void mostfalsedescribedmedis_without_doubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File("mostfalsedescribedmedis_without_doubles.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    ArrayList<InteractionList> l1 = new ArrayList<InteractionList>({list.size()});
    for (InteractionList l : list) {
        l1.add(new InteractionList(l));
    }

    for (InteractionList l : l1) {
        bw.write(l.getName());
        bw.newLine();

        HashMap<String, Long> map = new HashMap<String, Long>();

        newInteractionList(l);

        for (InteractionATC atc : l.getInteractions()) {
            if (map.containsKey(atc.getAtc1())) {
                map.put(atc.getAtc1(), map.get(atc.getAtc1()) + atc.getCount_without_doubles());
            } else {
                map.put(atc.getAtc1(), atc.getCount_without_doubles());
            }

            if (map.containsKey(atc.getAtc2())) {
                map.put(atc.getAtc2(), map.get(atc.getAtc2()) + atc.getCount_without_doubles());
            } else {
                map.put(atc.getAtc2(), atc.getCount_without_doubles());
            }
        }

        Map<String, Long> a = sortByValue(map);

        for (String str : a.keySet()) {
            bw.write(str + ";" + a.get(str));
            bw.newLine();
        }
    }

    bw.close();
}

/**
 * @param list
 * @throws IOException
 */
private static void mostfalsedescribedmedis_with_doubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File("mostfalsedescribedmedis_with_doubles.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));

    ArrayList<InteractionList> l1 = new ArrayList<InteractionList>({list.size() + 100});
    l1.addAll(list);

    for (InteractionList l : l1) {
        bw.write(l.getName());
        bw.newLine();

        HashMap<String, Long> map = new HashMap<String, Long>();

        newInteractionList(l);

        for (InteractionATC atc : l.getInteractions()) {
            if (map.containsKey(atc.getAtc1())) {
                map.put(atc.getAtc1(), map.get(atc.getAtc1()) + atc.getCount());
            } else {
                map.put(atc.getAtc1(), atc.getCount());
            }

            if (map.containsKey(atc.getAtc2())) {
                map.put(atc.getAtc2(), map.get(atc.getAtc2()) + atc.getCount());
            } else {
                map.put(atc.getAtc2(), atc.getCount());
            }
        }

        Map<String, Long> a = sortByValue(map);

        for (String str : a.keySet()) {
            bw.write(str + ";" + a.get(str));
            bw.newLine();
        }
    }

    bw.close();
}
```

```

}

/**
 *
 * @param list
 * @throws IOException
 */
private static void mostfalsedescribedmedis_without_doubles_third_stage(ArrayList<InteractionList> list) throws IOException {
    File f = new File("mostfalsedescribedmedis_without_doubles_third_stage.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));

    for (InteractionList l : list) {
        newInteractionList(l);
        bw.write(l.getName());
        bw.newLine();

        HashMap<String, Long> map = new HashMap<String, Long>();

        for (InteractionATC atc : l.getInteractions()) {
            try {
                if (map.containsKey(atc.getAtc1().substring(0, 4))) {
                    map.put(atc.getAtc1().substring(0, 4), map.get(atc.getAtc1().substring(0, 4)) + atc.getCount_without_doubles());
                } else {
                    map.put(atc.getAtc1().substring(0, 4), atc.getCount_without_doubles());
                }
                /**
                 * Wenn beide ATC-Codes der selben Gruppe angehören, wird
                 * nicht doppelt gezählt.
                 */
                if (atc.atc1.substring(0, 4).equals(atc.atc2.substring(0, 4))) {
                    continue;
                }

                if (map.containsKey(atc.getAtc2().substring(0, 4))) {
                    map.put(atc.getAtc2().substring(0, 4), map.get(atc.getAtc2().substring(0, 4)) + atc.getCount_without_doubles());
                } else {
                    map.put(atc.getAtc2().substring(0, 4), atc.getCount_without_doubles());
                }
            } catch (StringIndexOutOfBoundsException ex) {
                System.err.println("Fehler");
            }
        }

        Map<String, Long> a = sortByValue(map);

        for (String str : a.keySet()) {
            bw.write(str + ";" + a.get(str));
            bw.newLine();
        }

        bw.close();
    }
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void mostfalsedescribedmedis_with_doubles_third_stage(ArrayList<InteractionList> list) throws IOException {
    File f = new File("mostfalsedescribedmedis_with_doubles_third_stage.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));

    for (InteractionList l : list) {
        bw.write(l.getName());
        bw.newLine();

        HashMap<String, Long> map = new HashMap<String, Long>();

        for (InteractionATC atc : l.getInteractions()) {
            try {
                if (map.containsKey(atc.getAtc1().substring(0, 4))) {
                    map.put(atc.getAtc1().substring(0, 4), map.get(atc.getAtc1().substring(0, 4)) + atc.getCount_without_doubles());
                } else {
                    map.put(atc.getAtc1().substring(0, 4), atc.getCount());
                }

                if (map.containsKey(atc.getAtc2().substring(0, 4))) {
                    map.put(atc.getAtc2().substring(0, 4), map.get(atc.getAtc2().substring(0, 4)) + atc.getCount_without_doubles());
                } else {
                    map.put(atc.getAtc2().substring(0, 4), atc.getCount());
                }
            } catch (StringIndexOutOfBoundsException ex) {
            }
        }
    }
}

```

## Anhang

```
Map<String, Long> a = sortByValue(map);

for (String str : a.keySet()) {
    bw.write(str + ";" + a.get(str));
    bw.newLine();
}

bw.close();
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void all_mostfalsedescribedmedis_without_doubles_third_stage(ArrayList<InteractionList> list) throws IOException {
    File f = new File("all_mostfalsedescribedmedis_without_doubles_third_stage.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    HashMap<String, Long> map = new HashMap<String, Long>();

    for (InteractionList l : list) {

        for (InteractionATC atc : l.getInteractions()) {
            try {

                if (map.containsKey(atc.getAtc1().substring(0, 4))) {
                    map.put(atc.getAtc1().substring(0, 4), map.get(atc.getAtc1().substring(0, 4)) + atc.getCount_without_doubles());
                } else {
                    map.put(atc.getAtc1().substring(0, 4), atc.getCount_without_doubles());
                }
                /**
                 * Wenn beide ATC-Codes der selben Gruppe angehören, wird
                 * nicht doppelt gezählt.
                 */
                if (atc.atc1.substring(0, 4).equals(atc.atc2.substring(0, 4))) {
                    continue;
                }
                if (map.containsKey(atc.getAtc2().substring(0, 4))) {
                    map.put(atc.getAtc2().substring(0, 4), map.get(atc.getAtc2().substring(0, 4)) + atc.getCount_without_doubles());
                } else {
                    map.put(atc.getAtc2().substring(0, 4), atc.getCount_without_doubles());
                }
            } catch (StringIndexOutOfBoundsException ex) {
            }
        }
    }

    Map<String, Long> a = sortByValue(map);

    for (String str : a.keySet()) {
        bw.write(str + ";" + a.get(str));
        bw.newLine();
    }
    bw.close();
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void all_mostInteractionATCs_withoutdoubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File("all_mostInteractionATCs_without_doubles.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));

    Set<InteractionATC> aset = new HashSet<InteractionATC>();
    ArrayList<InteractionATC> alist = new ArrayList<InteractionATC>();
    for (InteractionList l : list) {
        aset.addAll(l.getInteractions()); //doppelte werden aussortiert
    }
    alist.addAll(aset);
    alist.sort(new ComparaATC());
    InteractionList ilist = new InteractionList("all", alist);
    newInteractionList(ilist);

    for (int i = 0; i < ilist.getInteractions().size(); i++) {

        bw.write(alist.get(i).getAtc1() + ";" + alist.get(i).getAtc2() + ";" + alist.get(i).getCount_without_doubles());
        bw.newLine();
    }

    bw.close();
}

/**
```

## Anhang

```
* Sortiert eine Map von groß nach klein
*
* @param <K>
* @param <V>
* @param map
* @return
*/
public static <K, V extends Comparable<? super V>> Map<K, V> sortByValue(Map<K, V> map) {
    List<Map.Entry<K, V>> list
        = new LinkedList<Map.Entry<K, V>>(map.entrySet());
    Collections.sort(list, new Comparator<Map.Entry<K, V>>() {
        @Override
        public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) {
            return (o2.getValue()).compareTo(o1.getValue());
        }
    });

    Map<K, V> result = new LinkedHashMap<K, V>();
    for (Map.Entry<K, V> entry : list) {
        result.put(entry.getKey(), entry.getValue());
    }
    return result;
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void all_mostfalsedescribedmedis_without_doubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File("all_mostfalsedescribedmedis_without_doubles.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    HashMap<String, Long> map = new HashMap<String, Long>();

    ArrayList<InteractionList> l1 = new ArrayList<InteractionList>(list.size() + 1000);
    l1.addAll(list);

    for (InteractionList l : l1) {
        newInteractionList(l);

        for (InteractionATC atc : l.getInteractions()) {
            if (map.containsKey(atc.getAtc1())) {
                map.put(atc.getAtc1(), map.get(atc.getAtc1()) + atc.getCount_without_doubles());
            } else {
                map.put(atc.getAtc1(), atc.getCount_without_doubles());
            }

            if (map.containsKey(atc.getAtc2())) {
                map.put(atc.getAtc2(), map.get(atc.getAtc2()) + atc.getCount_without_doubles());
            } else {
                map.put(atc.getAtc2(), atc.getCount_without_doubles());
            }
        }
    }

    Map<String, Long> a = sortByValue(map);

    for (String str : a.keySet()) {
        bw.write(str + ";" + a.get(str));
        bw.newLine();
    }
    bw.close();
}

/**
 *
 * @param list
 * @throws IOException
 */
private static void all_three_getInteractions_without_doubles(ArrayList<InteractionList> list) throws IOException {
    File f = new File("all_three_interactions_without_doubles.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    Set<InteractionATC> set1 = new LinkedHashSet<InteractionATC>();
    Set<InteractionATC> set2 = new LinkedHashSet<InteractionATC>();
    Set<InteractionATC> set3 = new LinkedHashSet<InteractionATC>();

    ArrayList<InteractionATC> alist = new ArrayList<InteractionATC>();
    int a = 0;
    for (InteractionList l : list) {
        if (!l.getName().equals("interaction_french.csv") && !l.getName().equals("interaction_Austria.csv") &&
            !l.getName().equals("interaction_one_high.csv")) {
            continue;
        }
        switch (a) {
            case 0:
                set1.addAll(l.getInteractions());
                a++;
                break;
            case 1:
                set2.addAll(l.getInteractions());
                a++;
                break;
            case 2:
                set3.addAll(l.getInteractions());
                a++;
                break;
        }
    }
    if (a % 3 == 0) {
        ArrayList<InteractionATC> all = new ArrayList<InteractionATC>(set1.size() + set2.size() + set3.size());
        all.addAll(set1);
        all.addAll(set2);
        all.addAll(set3);
        Map<String, Long> map = sortByValue(new HashMap<String, Long>());
        for (InteractionATC atc : all) {
            map.put(atc.getAtc1(), map.get(atc.getAtc1()) + atc.getCount_without_doubles());
        }
        for (String str : map.keySet()) {
            bw.write(str + ";" + map.get(str));
            bw.newLine();
        }
    }
}
```

```

        a++;
        break;
    case 1:
        set2.addAll(l.getInteractions());
        a++;
        break;
    case 2:
        set3.addAll(l.getInteractions());
        a++;
        break;
    default:
        break;
    }
}

}

for (InteractionATC atc : set1) {
    if (set2.contains(atc) && set3.contains(atc)) {
        alist.add(atc);
    }
}

alist.sort(new ComparaATC());
for (int i = 0; i < alist.size(); i++) {
    bw.write(alist.get(i).getAtc1() + ":" + alist.get(i).getAtc2() + ";" + alist.get(i).getCount_without_doubles());
    bw.newLine();
}

bw.close();
}

private static void onlyantidepressiva(ArrayList<InteractionList> list) throws IOException {
    File f = new File("onlyantidepressiva.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    Set<InteractionATC> set1 = new LinkedHashSet<InteractionATC>();
    Set<InteractionATC> set2 = new LinkedHashSet<InteractionATC>();
    Set<InteractionATC> set3 = new LinkedHashSet<InteractionATC>();
    Set<InteractionATC> set4 = new LinkedHashSet<InteractionATC>();

    for (InteractionList l : list) {
        newInteractionList(l);
        if (!l.getName().equals("interaction_french.csv") && !l.getName().equals("interaction_Austria.csv") &&
            !l.getName().equals("interaction_onc_high.csv") && !l.getName().equals("interaction_Austria_heavy.csv")) {
            continue;
        }
        if (l.getName().equals("interaction_french.csv")) {
            for (InteractionATC atc : l.getInteractions()) {
                if (atc.atc1.contains("N06A") || atc.atc2.contains("N06A")) {
                    set1.add(atc);
                }
            }
        }
        } else if (l.getName().equals("interaction_Austria.csv")) {
            for (InteractionATC atc : l.getInteractions()) {
                if (atc.atc1.contains("N06A") || atc.atc2.contains("N06A")) {
                    set2.add(atc);
                }
            }
        }
        } else if (l.getName().equals("interaction_onc_high.csv")) {
            for (InteractionATC atc : l.getInteractions()) {
                if (atc.atc1.contains("N06A") || atc.atc2.contains("N06A")) {
                    set3.add(atc);
                }
            }
        }
        } else if (l.getName().equals("interaction_Austria_heavy.csv")) {
            for (InteractionATC atc : l.getInteractions()) {
                if (atc.atc1.contains("N06A") || atc.atc2.contains("N06A")) {
                    set4.add(atc);
                }
            }
        }
    }
}

if (set1.isEmpty() && set2.isEmpty() && set3.isEmpty() && set4.isEmpty()) {
    return;
}

bw.write("interaction_french.csv");
bw.newLine();
for (InteractionATC atc : set1) {

    bw.write(atc.atc1 + ":" + atc.atc2 + ";" + atc.count_without_doubles);
    bw.newLine();
}
bw.write("interaction_Austria.csv");
bw.newLine();
for (InteractionATC atc : set2) {

```

## Anhang

```
        bw.write(atc.atc1 + ":" + atc.atc2 + ";" + atc.count_without_doubles);
        bw.newLine();
    }
    bw.write("interaction_onc_high.csv");
    bw.newLine();
    for (InteractionATC atc : set3) {

        bw.write(atc.atc1 + ":" + atc.atc2 + ";" + atc.count_without_doubles);
        bw.newLine();
    }
    bw.write("interaction_Austria_heavy.csv");
    bw.newLine();
    for (InteractionATC atc : set4) {

        bw.write(atc.atc1 + ":" + atc.atc2 + ";" + atc.count_without_doubles);
        bw.newLine();
    }

    bw.close();
}

public static void onlyantidepressivamedis(ArrayList<InteractionList> list) throws IOException {
    File f = new File("onlyantidepressivamedis.txt");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    HashMap<String, Long> map = new HashMap<String, Long>();

    ArrayList<InteractionList> l1 = new ArrayList<InteractionList>(list.size() + 100);
    l1.addAll(list);

    for (InteractionList l : l1) {

        newInteractionList(l);

        map.clear();
        if (!l.getName().equals("interaction_french.csv") && !l.getName().equals("interaction_Austria.csv") &&
            !l.getName().equals("interaction_onc_high.csv") && !l.getName().equals("interaction_Austria_heavy.csv")) {
            continue;
        }

        for (InteractionATC atc : l.getInteractions()) {
            if (!atc.atc1.contains("N06A") && !atc.atc2.contains("N06A")) {
                continue;
            }
            if (map.containsKey(atc.getAtc1())) {
                map.put(atc.getAtc1(), map.get(atc.getAtc1()) + atc.getCount_without_doubles());
            } else {
                map.put(atc.getAtc1(), atc.getCount_without_doubles());
            }

            if (map.containsKey(atc.getAtc2())) {
                map.put(atc.getAtc2(), map.get(atc.getAtc2()) + atc.getCount_without_doubles());
            } else {
                map.put(atc.getAtc2(), atc.getCount_without_doubles());
            }
        }
        Map<String, Long> a = sortByValue(map);

        if (l.getName().equals("interaction_french.csv")) {
            bw.write("interaction_french.csv");
            bw.newLine();
        } else if (l.getName().equals("interaction_Austria.csv")) {
            bw.write("interaction_Austria.csv");
            bw.newLine();
        } else if (l.getName().equals("interaction_onc_high.csv")) {
            bw.write("interaction_onc_high.csv");
            bw.newLine();
        } else if (l.getName().equals("interaction_Austria_heavy.csv")) {
            bw.write("interaction_Austria_heavy.csv");
            bw.newLine();
        }

        for (String str : a.keySet()) {
            if (str.contains("N06A")) {
                bw.write(str + ";" + a.get(str));
                bw.newLine();
            }
        }

    }
    bw.close();
}

/**
 * neue Liste, welche die ATCs kürzer als 7 in die Einzelnen ATC-Codes
 * aufteilt
 *
 * @param l
```

## Anhang

```
*/
public static void newInteractionList(InteractionList l) {
    ArrayList<InteractionATC> deletelist = new ArrayList<InteractionATC>();
    ArrayList<InteractionATC> addlist = new ArrayList<InteractionATC>();

    for (InteractionATC atc : l.getInteractions()) {
        int i = 0;
        int j = 0;
        if (atc.atcgeneral) {
            if (!atc.atcs == null) {
                if (!atc.atcs.isEmpty()) {
                    for (InteractionATC a : atc.atcs) {
                        i += a.count;
                        j += a.count_without_doubles;
                        addlist.add(a);
                    }
                    if (atc.count != i) {
                        if (atc.count_without_doubles != j) {
                            System.err.println("Nicht gleich!");
                            throw new RuntimeException();
                        }
                    }
                }
                deletelist.add(atc);
            } else {
                deletelist.add(atc);
            }
        }
    }
    l.getInteractions().removeAll(deletelist);
    l.interactions.addAll(addlist);
}
}
```