# Data integration and trend analysis for surveillance of antimicrobial resistance

Teodoro, Douglas

UNIVERSITÉ DE GENÈVE
Département d'Informatique

FACULTÉ DES SCIENCES
Professeur Ron Appel

Département de Radiologie et Informatique Médicale

FACULTÉ DE MÉDECINE
Docteur Patrick Ruch

---

# Data Integration and Trend Analysis for Surveillance of Antimicrobial Resistance

## THÈSE

présentée à la Faculté des sciences de l'Université de Genève

pour obtenir le grade de Docteur ès sciences, mention informatique

par
Douglas Teodoro

de
Imbé de Minas (Brésil)

Thèse N° 4479

GENÈVE
ReproMail

2012

# UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

***Doctorat ès sciences***
***Mention informatique***

Thèse de *Monsieur Douglas Henrique TEODORO*

intitulée :

## " Data Integration and Trend Analysis for Surveillance of Antimicrobial Resistance "

La Faculté des sciences, sur le préavis de Messieurs P. RUCH, docteur et directeur de thèse (Faculté de médecine, Département de radiologie et informatique médicale), R. D. APPEL, professeur ordinaire et codirecteur de thèse (Département d'informatique), Ch. LOVIS, professeur associé (Hôpital Universitaire de Genève, Département imagerie), M. CUGGIA, docteur (Faculté de médecine, Université de Rennes, France) et Y. TOUSSAINT, docteur (Laboratoire Loran de Recherche en Informatique et ses Applications, Vandoeuvre-lès-Nancy, France), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 22 octobre 2012

**Thèse - 4479 -**

**Le Doyen,** Jean-Marc TRISCONE

N.B. - La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

*To my family*

# Acknowledgements

I would like to thank all my colleagues, in special the folks of the Bibliomics and Text Mining group at the University of Applied Sciences (HEG-SO) – Geneva and of the Division of Medical Information Sciences at the University Hospitals of Geneva (HUG), who have supported me during the years of the PhD. I want also to thank the constant help and support from Louise. She was very important in the realization of this work.

# Abstract

Antimicrobial resistance is a major worldwide public health problem. The misuse of antimicrobial agents and the delay in spotting emerging and outbreak resistances in current biosurveillance and monitoring systems are regarded by health bodies as underlying causes of increasing resistance. In this thesis, we explore novel methods to monitor and analyze antimicrobial resistance trends to improve existing biosurveillance systems. More specifically, we investigate the use of semantic technologies to foster integration and interoperability of interinstitutional and cross-border microbiology laboratory databases. Additionally, we research an original, fully data-driven trend analysis method based on trend extraction and machine learning forecasting to enhance antimicrobial resistance analyses.

In the first part of the thesis, we derive the main requirements for an effective antimicrobial resistance monitoring system, from which we design a decentralized, real-time and source-independent architecture based on the Semantic Web stack. The architecture uses an ontology-driven approach to promote the integration of a network of sentinel hospitals. Then, in the second part, we study a robust model for extraction and forecasting of antibiotic resistance trends. Our method consists of breaking down the resistance time series into different oscillation modes to extract the trends. Furthermore, a learning algorithm based on the k-nearest neighbor framework uses the decomposed series to project mappings from past events into the forecasting dimension.

The results indicate that the Semantic Web-based approach provides an efficient and reliable solution for development of e-health architectures that enable online antimicrobial resistance monitoring from heterogeneous data sources. In addition, our method for trend extraction improves resistance trend analyses by describing short-term trends and their periodicity. Finally, statistically significant performance improvements are found for the machine learning forecasting methods that decompose the resistance time series

and filter out noise components in comparison with baseline approaches. The methods developed here could serve thus to enhance biosurveillance systems by providing complimentary tools for the monitoring and timely analysis of antimicrobial resistance trends.

# Résume

Dans le monde entier, la résistance aux antimicrobiens est un problème de santé publique majeur. La mauvaise utilisation des antimicrobiens, ainsi que le retard dans la détection des épidémies et des émergences des phénotypes résistants dans les systèmes actuels de biosurveillance, sont considérés par les organismes de santé comme les causes sous-jacentes de la croissance de la résistance. Dans cette thèse, nous explorons de nouvelles méthodes pour surveiller et analyser les tendances de la résistance aux antimicrobiens dans le but d'améliorer les systèmes existants de biosurveillance. Plus précisément, nous étudions l'utilisation de technologies sémantiques pour favoriser l'intégration et l'interopérabilité des bases de données des laboratoires de microbiologie interinstitutionnelles. De plus, nous recherchons des méthodes d'analyses originales, entièrement pilotées par les données, qui sont basées sur l'extraction de tendances et sur l'apprentissage automatique appliqué aux prévisions, pour améliorer l'analyse des résistances aux antimicrobiens.

Dans la première partie de la thèse, nous établissons les principaux besoins d'un système efficace de surveillance de la résistance antimicrobienne, à partir desquels nous concevons un système décentralisé, fonctionnant en temps réel, indépendant de la source et basé sur les principes du Web Sémantique. L'architecture utilise une approche axée sur l'ontologie pour promouvoir l'intégration d'un réseau d'hôpitaux sentinelles. Puis, dans la deuxième partie, nous étudions un modèle robuste pour l'extraction et la prévision des tendances de la résistance aux antibiotiques. Notre méthode consiste à décomposer les séries temporelles de résistance dans différents modes d'oscillation pour extraire les tendances. Ensuite, un algorithme d'apprentissage basé sur les k plus proches voisins utilise les séries décomposées pour projeter d'événements passés dans la dimension prospective.

Les résultats indiquent que l'approche basée sur le Web Sémantique fournit une solution efficace et fiable pour le développement d'architectures de cybersanté qui permettent en utilisant des sources de données hétérogènes d'obtenir la surveillance en ligne de l'antibiorésistance. En outre, notre méthode d'extraction de tendance améliore l'analyse des résistances en décrivant des tendances à court terme et leurs périodicité. Enfin, des améliorations des performances statistiquement significatives sont trouvées pour les méthodes de prévision basées sur l'apprentissage automatique qui décomposent les séries chronologiques de résistance et filtrent le bruit, par rapport aux approches de base. Les méthodes développées ici pourraient ainsi servir à améliorer les systèmes de biosurveillance en fournissant des outils complémentaires pour la surveillance et l'analyse en temps réel des tendances de la résistance aux antimicrobiens.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

**ACH** Athens Chest Hospital "Sotiria"

**ANN** Artificial Neural Network

**ARIMA** Autoregressive integrated moving average

**ARTEMIS** Antimicrobial Resistance Trend Monitoring System

**ATC** Anatomical Therapeutic Chemical

**caBIG** cancer Biomedical Informatics Grid

**CEN** European Committee for Standardization

**CI** confidence interval

**COIN** COntext INterchange

**CORBA** Common Object Request Broker Architecture

**CPL** Collection Program Language

**CPOE** Computerized Prescription Order Entry

**DAML** DARPA Agent Markup Language

**DCO** DebugIT Core Ontology

**DDO** data definition ontology

**DebugIT** Detecting and Eliminating Bacteria Using Information Technology

**EARS-Net** European Antimicrobial Resistance Surveillance Network

**ECDC** European Centre for Disease Prevention and Control

**EHR** Electronic Health Record

**EMD** Empirical Mode Decomposition

**GAV** global-as-view

**GRAIL** GALEN Representation and Integration Language

**HEGP** Georges Pompidou European Hospital

**HEMSYS** HEterogeneous Multi-database System

**HL7** Health Level Seven

**HUG** Les Hôpitaux Universitaires de Genève

**ICD** International Classication of Diseases

**IGD** Integrated Genome Database

**IMF** Intrinsic Mode Function

**INDUS** INtelligent Data Understanding System

**IRI** Internationalized Resource Identifier

**ISO** International Organization for Standardization

**IZIP** Internetový Pristup Ke Zdravotním Informacím Pacienta

**k-NN** k-Nearest Neighbor

**KRAFT** Knowledge Reuse And Fusion/Transformation

**LAV** local-as-view

**lCDR** local Clinical Data Repository

**LIS** Laboratory Information System

**LOINC** Logical Observation Identifiers, Names and Codes

**MAD** median absolute deviation

**MAE** mean absolute error

**MIC** minimum inhibitory concentration

**MOMIS** Mediator envirOnment for Multiple Information Sources

**MRSA** methicillin-resistant *Staphylococcus aureus*

**N3** Notation 3

**NHH** National Heart Hospital

**ODBC** Open Database Connectivity

**OIL** Ontology Inference Layer

**OWL** Web Ontology Language

**PAHO** Pan American Health Organization

**RDF** Resource Description Framework

**RDFS** RDF Schema

**RDQL** RDF Data Query Language

**RIM** Reference Information Model

**RMSE** root mean squared error

**SEARCH** Sentinel Surveillance of Antibiotic Resistance in Switzerland

**SeRQL** Sesame RDF Query Language

**SHRINE** Shared Health Research Information Network

**SIMS** Scientic Image Management System

**SIR** Swedish Intensive Care Registry

**SNOMED CT** Systematized Nomenclature of Medicine – Clinical Terms

**SPARQL** SPARQL Protocol and RDF Query Language

**SRS** Sequence Retrieval System

**SVM** Support Vector Machine

**TAMBIS** Transparent Access to Multiple Bioinformatics Information Sources

**TINet** Target Informatics Net

**TOSC** two one-sided convolution test

**TSIMMIS** Stanford-IBM Manager of Multiple Information Sources

**UKLFR** Universitätsklinikum Freiburg

**UniProt** Universal Protein Resource

**USA** United States of America

**VRE** vancomycin-resistant *Enterococcus*

**W3C** World Wide Web Consortium

**WHO** World Health Organization

**XML** eXtensible Markup Language

**XSD** XML Schema Denition Language

# 1

# Introduction

Since the development of the first sulfonamide drugs in early 1930s, antimicrobial agents, which include antibiotics but also other similar drugs such as antimycotics, have been used to treat patients with infectious diseases. They have become an indispensable therapeutic agent in modern medicine to reduce morbidity and mortality caused by infectious agents. Their widespread use in the subsequent decades has led to the natural appearance and selection of resistant microbes. Unfortunately, in the last two decades, the level of antimicrobial resistance to many antimicrobials has been increasing at such a fast rate that it became a worldwide public health concern. Increasing antimicrobial resistance is a natural and longstanding phenomenon [1]. However, due to the frequency in which new emerging resistant strains are occurring among many pathogens it has turned into an ever more alarming situation. Consequently, it has been recognized by many international health institutions, including the European Centre for Disease Prevention and Control (ECDC), the Pan American Health Organization (PAHO) and the World Health Organization (WHO), as one of the major global human health problems [2, 3, 4].

Worldwide, health agencies have identified effective surveillance systems as a key aspect in the fight against resistant pathogens. The implementation of such surveillance system involves the integration and interoperability of distributed healthcare systems but also the development of intelligent tools to support clinicians and infection control officers in decision making. In the first part of this thesis, we investigate the use of semantic technologies to integrate distributed and heterogeneous microbiology data sources to support translational monitoring of antimicrobial resistance. In the second

part, we explore machine learning techniques to model resistance trends and provide short-term resistance forecasting for the analysis of resistance evolution. Despite the different tasks – data management and data analysis – the overall system can be seen as a monolithic information technology framework to help in the battle against ever more resistant bugs.

This chapter provides a brief overview of the thesis content. In Section 1.1, we introduce the reader to antimicrobial resistance surveillance and its application on resistance monitoring and control. Section 1.2 presents the semantic tools used in the integration of heterogeneous microbiology data sources. In Section 1.3, we introduce the machine learning techniques that we will use to model resistance time series. Section 1.4 summarizes the motivation of this work. Finally, Section 1.5 provides the outline of the thesis.

## 1.1 Antimicrobial Resistance Surveillance

Antimicrobial resistance surveillance is the *"systematic, ongoing data collection, analysis and reporting process that quantitatively monitors temporal trends in the occurrence and distribution of susceptibility and resistance to antimicrobial agents, and provides information useful as a guide to medical practice, including therapeutics and disease control activities"* [5]. It is an essential mechanism to provide information on the resistance magnitude and trends, and to monitor the effect of clinical interventions and of public health policies upon resistance evolution.

Antimicrobial resistance has a large impact on public health. Pneumonia, tuberculosis, diarrhoeal diseases, malaria, measles and HIV/AIDS account for about 90% of deaths caused by infection diseases worldwide [6, 4]. Pathogens that cause these diseases are often highly resistant to first-line drugs and, in many cases, treatment with second- and third-line antimicrobials is seriously compromised. Additionally, resistance to antimicrobials is often verified in hospital-acquired and viral infections and is emerging in parasitic diseases such as African trypanosomiasis and leishmaniasis [7]. Furthermore, according to the ECDC, just within the European Union Member States, about 25,000 patients die each year from infections caused by only six antibiotic-resistant bacteria - *Enterococcus spp.*, *Escherichia coli*, *Klebsiella pneumoniae*, *Pseudomonas aeruginosa*, *Staphylococcus aureus* and *Streptococcus pneumoniae* [2]. Whereas in the United States,

for example, methicillin-resistant *Staphylococcus aureus* alone is responsible for more deaths than emphysema, HIV/AIDS, Parkinson's disease and homicide combined [8, 9].

Amongst several interconnected factors that contribute to the emergence of antimicrobial resistance, healthcare agencies claim that current substandard or absent surveillance and monitoring systems is an underlying reason for increasing resistance [5, 7, 10]. Unless antimicrobial resistant pathogens are detected as they are selected and actions are taken quickly to isolate and control their spread, the healthcare system may soon be confronted by (re-)emerging infectious diseases, similar to those faced in the pre-antibiotic era. Integration of data from electronic healthcare systems, such as computerized prescription order entry (CPOE) and laboratory information system (LIS), into monitoring and surveillance frameworks is seen as a key requirement for the success in understanding and controlling resistant agents. The following applications further illustrate the importance of effective biosurveillance systems in resistance management [11].

- **Decision Support for Empirical Treatment.** In infection cases, clinicians may be required, at least in the first therapy, to select an antimicrobial based on guidelines only to abrogate the delay in performing a susceptibility test or perhaps due to the incapability to isolate the pathogen causing the infection. Furthermore, in many countries, access to antibiogram tests is rare and most patients are actually treated empirically throughout the whole treatment course [11]. In these cases, the clinician must estimate based on several variables, such the infection location, the effectiveness of the antiinfective agent and the drug cost, both the pathogen causing the infection and the antimicrobial that will target most efficiently the infection agent. However, empirical therapy comes at some cost. Discordant therapy not only contributes to increased resistance but also to excessive morbidity and mortality [12]. Enhanced biosurveillance systems can provide real-time[1] and on demand statistics, for example, on the current prevalence of a given pathogen in a clinical setting and the pathogen's resistance to the different drugs recommended for treatment. Therefore, the system can offer evidence to the prescriber to support or contradict a particular treatment

[1]Throughout this thesis, the term *real-time* is employed to describe an online process that provides access to antimicrobial resistance data as soon as they are available in the microbiology database, as opposed to *batch-mode* access.

regime during an empirical decision event, eventually improving the accuracy of the treatment.

- **Infection Control.** Pathogens migrate from patient to patient usually through direct contact with fomites or hopping through healthcare workers. The pathogens that carry resistance genes persist and are likely to spread and select hosts receiving antimicrobials[1]. Real-time monitoring systems can help infectiologists and public health officers to track the prevalence of pathogenic species and the evolution of resistance against different antimicrobial classes. Moreover, it can help spot the emergence or outbreak of resistant genes at different locations, either within the hospital or the community, especially if connected through a sentinel network [13]. Hence, monitoring systems working on real-time data of microbiology laboratory can help to track and control virulent infections within the clinical setting and in the community.

- **Guideline Generation.** Health agencies, professional societies and healthcare institutions develop and maintain guidelines for use of antimicrobial agents. These guidelines are generated using evidence on the use of antimicrobials and are derived from several information sources, including surveillance programs based on microbiology laboratory data, the scientific literature and clinical narratives. Up-to-date biosurveillance systems that publicly expose resistance data can foster directly the fast development and updating of clinical guidelines, by providing on demand information on resistance status at the local, and, if connected through a network, at national and international levels. Moreover, they are a valuable source of information for expert systems to create guidelines for bacterial treatment and hospital infection control policies [14]. As demonstrated in [15], the injection of resistance models significantly improves the system's precision in selecting the most appropriate antibiotic for a given treatment.

## 1.2 Transnational Resistance Monitoring

More effective multinational monitoring systems are essential to guide policy makers in public health and hospital infection control. Inappropriate use of antimicrobials is

[1] Antimicrobial consumption reduces the within host competition by killing other non-resistant germs, facilitating then the life of resistant strains.

considered as the main reason for increasing resistance. Improving drug usage, through better prescribing, and infection control and public health policies are critical actions in the fight against antimicrobial resistance. Whether for daily clinical practice or developing infection control policies, access to up-to-date resistance information through monitoring systems is crucial for decision making. Unfortunately, this is often not the case in healthcare. Resistance information provided by monitoring and surveillance systems is usually outdated and many times unavailable, especially in developing countries. Additionally, with the ever seen frequency and widespread of resistant phenotypes among many pathogens, antimicrobial resistance is no longer a localized problem. There is a consensus amongst health agencies that due to the constant movement of people and trading, no country will be able to tackle this issue without international collaboration [16, 17]. Therefore, antimicrobial resistance monitoring should be regarded as a multinational public health issue.

Data sets needed for antimicrobial resistance surveillance are found in clinical and laboratory digital databases worldwide. Several types of database management systems are used to store and control access to these data. In essence, these databases are very heterogeneous not only in the technology used but especially in their information model. Data are stored and organized in various formats, including structured as found in relational databases, but also free text contents, such as clinical notes. In this work, we argue that, in order to monitor resistance coming from such diverse and distributed data sources, semantic-aware information technologies for data integration and analyzes are essential. Berners-Lee [18] has coined the term Semantic Web to define methods and standards that allow machines to comprehend the underlying meaning of the information available in a heterogeneous network. These methods provide formal description of concepts, terms and relationships within a given knowledge domain, and means to store and access remote web resources. Thus, they can be used to interoperate, share and aggregated data unambiguously, fostering the integration of distributed microbiology databases.

Semantic Web technologies include several standards and tools, such as the Resource Description Framework (RDF)[1], a variety of data interchange formats (e.g., RDF/eXtensible Markup Language (XML), Notation 3 (N3), Turtle, N-Triples) and

[1] `http://www.w3.org/RDF/`

notations such as RDF Schema (RDFS)[1], the Web Ontology Language (OWL)[2] and finally query languages (e.g., SPARQL Protocol and RDF Query Language (SPARQL)[3]) for web resources. In his original article, Berners-Lee claimed that the availability of machine-readable metadata would enable automated agents and other software to access the Web more intelligently. The agents would be able to perform tasks automatically and locate related information on behalf of the user.

To make such approaches possible the deployment of ontologies and of terminologies is crucial. Ontologies define models and concepts in a formal language understandable by machines. Many ontological resources are already available in the Web, especially in the bioinformatics field, and many others are being created [19, 20]. Using Semantic Web technologies, these ontologies can be defined (e.g., using the RDF/OWL languages), stored (e.g., in RDF data stores) and accessed (e.g., via the SPARQL query language). In addition to ontologies, terminologies such as the Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT)[4], the International Classification of Diseases (ICD)[5], the Universal Protein Resource (UniProt/NEWT) and the WHO's Anatomical Therapeutic Chemical (ATC)[6], are essential to provide a common way to harmonize the syntax of biomedical concepts across heterogeneous and multinational data sources. Despite not normally being Semantic Web-compatible, particularly because some were developed before the definition of Semantic Web, these terminologies are being incorporated into RDF/OWL ontologies. Additionally, they are being wrapped by some semantic web resources such as DBPedia[7] and Linked Life Data[8] so that they can be accessed seamlessly by SPARQL and other semantic web resources.

## 1.3 Forecasting Antimicrobial Resistance Trends

At the point of care, the resistance information of pathogens causing a given infection tends always to be outdated due to the delay necessary to perform a microbiology

[1] http://www.w3.org/2001/sw/wiki/RDFS
[2] http://www.w3.org/2004/OWL/
[3] http://www.w3.org/TR/rdf-sparql-query/
[4] http://www.ihtsdo.org/snomed-ct/
[5] http://www.who.int/classifications/icd/en/
[6] http://www.whocc.no/atc/structure_and_principles/
[7] http://dbpedia.org/About
[8] http://linkedlifedata.com/

susceptibility test. Even if real-time monitoring systems are in place, the resistance status is available at the best case only after the pathogen has been isolated, cultured and the antibiogram or the genetic screening test has been performed. Although some off-the-shelf tests can provide faster results, such as *strep test*, the standard and mass adopted phenotype-based antibiograms take in average two to three days to provide the answer on the agent's effectivity. Hence, the current resistance information is only accessible some time in the future after the infection symptoms are detected by the physician. Frequently, this delay is not compatible with the infection severity and the physician decision timing, and, as a result, in clinical practice many times physicians are obliged to prescribe without the ultimate evidence on the agent's effectivity. However, given the current alarming resistance rates, physicians should have access to the most up-to-date and accurate possible resistance information in order to make optimal decisions.

Due to the delay observed in obtaining access to susceptibility test results, but also the application of forecasts in outbreak detection models, accurate prediction of antimicrobial resistance trends might have a significant impact on reduction and control of resistance by providing reference levels of the current and future resistance states. While pathogen isolation and antimicrobial testing cannot be performed fast enough to keep up with the progression of infection, short-term resistance forecasting could be used as an alternative to offer prescribers better evidence of the effectiveness of an antimicrobial agent. Moreover, predictive models for resistance trends are key in outbreak detection biosurveillance systems, serving as a reference value between an endemic and pandemic judgment [21].

The literature provides a large number of theoretical and applied works describing methods and applications of time series forecasting tools [22, 23, 24, 25]. Most of them are focused on classical statistical models, such as the autoregressive integrated moving average (Box-Jenkins) [23, 24], exponential smoothing (Holt-Winters) [26, 27] and state space (Kalman) [28, 29] models. In general, these forecasting methods consist in fitting the data sequence into a mathematical function, a process also called *regression*, and predict future values using the fitted function. In the past decade, machine learning models have established themselves as serious contenders to classical statistics in regression problems [30]. In particular, machine learning has been used to forecast financial data [31, 32], trends of physiological data [33], electric load [34], among others.

However, to our knowledge, no original work has been published exploiting the use of machine learning to forecast resistance trends.

Hence, to complement our transnational biosurveillance system, we investigate a robust model to extract and predict antimicrobial resistance rates based on empirical oscillation modes of resistance time series and machine learning. Using antimicrobial susceptibility test data, we extract waveforms that describe different variation modes of resistance, whose dynamics may theoretically represent outbreaks, seasonal effects and periodic antimicrobial resistance trends. The extraction of the different waveforms is performed using the Empirical Mode Decomposition (EMD) algorithm [35], an adaptive and data-driven technique that represents the signal as a set of oscillation functions, called Intrinsic Mode Functions (IMFs), plus a monotonic residual acting as the underlying trend. The decomposed waveforms are further used as the input to the machine learning forecaster. Instead of deploying the full signal spectrum in the learning algorithm, we select only those sequences that contribute significantly to the underlying signal, that is, those that differ from the noise. The learner uses the delay coordinate embedding technique [36, 37] to capture the dynamics of the different time series components and a k-Nearest Neighbor (k-NN) algorithm to project observed resistance events into the future dimension.

## 1.4 Thesis Statement

A system of excellence for antimicrobial resistance surveillance requires building an information technology framework that shares real-time resistance information from distributed and heterogeneous data sources and provides intelligent data analysis tools. This task involves the development of an integration platform for healthcare information systems that captures the semantics of the diverse data sources and that is able to exploit intelligently this data and knowledge intensive environment.

In this thesis, we propose to investigate the use Semantic Web technologies, in particular SPARQL, RDF and OWL together with standard biomedical terminologies, to provide a decentralized monitoring infrastructure where healthcare institutions can share online microbiology information to be used by bodies concerned with antimicrobial resistance surveillance. In the second part of our work, we propose a novel machine learning model for analysis of short-term resistance trends, featuring trend extraction

and forecasting capabilities. We use our data integration framework to fetch resistance data, which will serve as the training examples for our learning algorithms. Precisely, we investigate the use of decomposed time series to improve the learner's accuracy. We hypothesize that the forecasting models we build can provide better evidence to prescribers by predicting accurately short-term resistance evolution.

The main contributions of our work are twofold. First, we introduce an architecture for sharing and monitoring real-time multinational antimicrobial resistance data. Our architecture scales to a large network of heterogeneous sentinel endpoints, while complying with healthcare constraints of data sharing, such as source autonomy and patient privacy. Additionally, we present a novel machine learning model based on empirical mode decomposition and complex system theory for extraction and forecasting of resistance trends. Our model provides further insights on the dynamics of resistance trends and improves the performance of baseline forecasting approaches. To the best of our knowledge, it is the first attempt to use machine learning with decomposed time series to forecast resistance rates.

## 1.5 Thesis Outline

This thesis describes methods for integrating biomedical data sources for biosurveillance and learning-based modeling approaches for antimicrobial trend forecast. To improve the readability, we have organized the thesis in two parts. From Chapter 2 to Chapter 5, we develop and assess our data integration architecture for transnational monitoring. Then, from Chapter 6 to Chapter 8 we present the learning-based time series forecasting model. Finally, Chapter 9 concludes the thesis. The following headings describe in more detail each chapter.

**Chapter 2** provides a review of methods to integrate and interoperate heterogeneous life science databases. We review the existing data integration methodologies, focusing on semantic approaches. These methods are used in the design of our model for integration of distributed microbiology databases.

**Chapter 3** introduces the reader to basic aspects of antimicrobial resistance that are relevant to model the decentralized monitoring architecture. We discuss the rep-

resentation of microbial data and sources, and ways to describe them in a formal and standard model.

**Chapter 4** presents the integration architecture of the interinstitutional antimicrobial resistance trend monitoring system. It details the different levels of the architecture, the ontologies used and how we can align data from heterogeneous systems into a common framework for antimicrobial resistance monitoring.

**Chapter 5** presents the results of the clinical evaluation of the integrative monitoring architecture designed in Chapter 4. First, we assess the system at the technical level, where the performance of the distributed engine and the clinical pertinence of the results are measured. Second, we evaluate the clinical and infection control usefulness of the system from the infectious disease specialists viewpoint.

**Chapter 6** provides a review of existing time series forecasting methods from the machine learning viewpoint. These methods serve as the starting point in the design of our model for prediction of short-term antimicrobial resistance trends.

**Chapter 7** describes our resistance trend extraction and forecasting model. We provide basic descriptive statistics of antimicrobial resistance time series and ways to represent it in a machine learning model. Then, we develop our machine learning algorithm, which employs the k-NN framework to capture the dynamics of short-term antimicrobial resistance time series.

**Chapter 8** presents the results of the trend extraction and forecasting model developed in Chapter 7. The system is evaluated using a large time series data set of real antimicrobial resistance rates.

**Chapter 9** concludes the thesis and presents the future works.

# Part I

# Data Management

# 2

# Biomedical Data Integration and Interoperability Review

## 2.1 Introduction

The expansion of biomedical knowledge, reduction in computing costs and availability of information technology *commodities*, such as network bandwidth, store and processing power, have led to an explosion of the life sciences electronic data stored in databases all around the world. Information ranging from clinical findings of a given patient to the genetic structure of virtually all species are stored in digital media and accessed through remote devices. However, especially in the medical field, there is still modest secondary usage of this information in order to improve further the quality of care, public health and clinical research.

This observation seems particularly verified in the domain of biosurveillance. Very few systems are able to exploit the rich content of local clinical information systems to improve further infection and resistance control. This situation gets worse for multi-institutional and -national monitoring and surveillance. Most of the available systems use data aggregated on a yearly basis, which are no longer suitable to monitor faster ever increasing antimicrobial resistant phenotypes. Current cases of resistance outbreaks [17, 38, 39] are clear evidence that multinational monitoring systems should provide more comprehensive but critically faster answer so that control agencies and local authorities can act accordingly.

## 2.2 Data Integration and Interoperability

In the literature, data integration refers to the process of harmonization of heterogeneous data sources and content to provide the user with a unified and homogenous view of these data [40]. The concept of data integration is normally associated with interoperability, that is, the *"ability of two or more components to exchange information and to use the information that has been exchanged"* [41]. In an integration system, at least three levels of interoperability are required. First, the system has to interoperate at the technical level. The different access and storage protocols have to be in synchrony so that data can be reached and retrieved. Second, the information syntax has to be homogenous across the different data sources. The system's syntax defines the grammar that carries the data semantics and structure. Finally, the integration system has to provide semantic interoperability amongst the different components, that is, the ability of computers to understand and automatically process the exchanged information.

## 2.3 Challenges to Integration of Microbiology Data Sources

Amongst several other issues, to enable secondary usage, like monitoring and knowledge discovery, the information stored in the distributed, heterogeneous and, to some extent, chaotic healthcare storage system needs to be first integrated. However, numerous challenges are faced to develop a system that provides interoperability and homogeneity to biomedical data sources. The following headings list some of the major issues involved in the task of integrating biomedical data.

### 2.3.1 Lack of Technical Interoperability

Integrating proprietary and heterogeneous clinical and non-clinical databases and systems from distributed sources is still a highly challenging technological task. The integration system will have to interoperate with different hardware platforms, operational systems, database management systems, query languages, data models, access protocols, transport formats and programming languages [42]. Despite considerable advances in portability of computer systems in the last decades, with prominence in the JAVA virtual machine and web services, healthcare systems are based usually on non-standard proprietary and technology dependent architectures. For example, WHONET

[43], the system most used internationally to report microbiology results, is dependent on the Microsoft Windows© operational system.

### 2.3.2 Lack of Semantic Interoperability

Semantics is widely the main concern when integrating heterogeneous sources. Various pieces of information and data must have the same meaning in order to be analyzed as part of the same set. However, conceptual ambiguities may occur in the data models of the data sources. For example, attributes of two information models may have different labels but refer to the same concept, a phenomenon referred as *synonymy*. Conversely, attributes may have the same label but refer to different concepts, known in linguistics as *polysemy* [44].

### 2.3.3 Low Data Quality

Low quality is the intrinsic character of real-world clinical data, which are full of missing values, abbreviations and errors. Low quality jeopardizes the integration process since it reduces the confidence on the whole system, invalidating further analysis of the integrated data. Nevertheless, data quality is often not taken into account in limited research studies [45].

### 2.3.4 Missing Transparency about Reliability

Even if data are available and accessible, there is no guarantee that they are valid. Prior to any analysis of the data, one should be able to access information that characterizes the data provenance [46]. However, this information is rarely available from the local sources, if at all.

### 2.3.5 Multimedia Barrier

New knowledge and information often lie in the ability to merge several different sources and modalities of data. One of the complex characteristics of data in life sciences is the multiplicity of media types, ranging from simple number and controlled text fields to images, audio and video, and the ability to analyze these information as time evolves. As an example, the medical literature is rich in findings, such as risks associated with pregnancy and other collateral effects, which many times are not explicitly present

in the structured information of a data warehouse, either permanently or for a given period. However, the correlation of both information (and sources) would be crucial in an effective decision support system for prescription.

### 2.3.6 Security, Privacy and Confidentiality

The power gained in aggregating data from many sources can lead to unacceptable and unforeseen risks for the patient's and the citizen's privacy rights [47, 48, 49]. Differently from publicly available (non-human) biological data, such as some genomic and proteomic banks, medical data are very sensitive concerning sharing of patient identifiers. Protection of patient's identity needs to be taken into account prior to any action towards data integration. Sophisticated mechanisms of data encryption and anonymization are found in the literature. However, storing patient data out of the healthcare intranet would still violate many ethical and legal specifications.

## 2.4 Electronic Health Record Interoperability

Electronic health record (EHR) [50, 51, 47] is the core information of a health information system. According to Iakovidis [50], EHR is a *"digitally stored healthcare information about an individual's lifetime with the purpose of supporting continuity of care, education and research, and ensuring confidentiality at all times"*. In the past decades, the main goal of health informatics has been the development of EHRs that can capture and preserve faithfully the clinical meaning of facts related to healthcare. With the advent of the different systems, several interoperability standards that try to address the requirements of the individual EHR systems were created.

### 2.4.1 HL7

The Health Level Seven (HL7)[1] organization was created in 1987 in the United States of America (USA) from the need to cope with the growing diversity of messaging exchange protocols in the health insurance industry [47]. HL7 is the most widely used healthcare interoperability framework to share EHRs between clinical institutions. HL7 version 3 provides a set of standards for the structuration, markup, exchange, management and integration of medical data. All the protocol specification standards of

[1] `http://www.hl7.org/`

the third version are derived from the Reference Information Model (RIM). The RIM is a formal information model that represents the classes and attributes needed to express healthcare information. It defines three top level classes: entity, such as people, places and devices; role, such as that of patient or university healthcare institution; and acts, such as laboratory observations and procedures. These classes are connected by three other main relationship classes: role relationship, which relates roles; act relationship, which connects acts; and participation, which connects roles to acts. The model is very generic and allows large expressivity of healthcare concepts. However, as pointed by Smith and Ceusters [52], HL7 has still many incoherences, especially concerning the duality of reference ontology and information model, and its application as a fully operational platform for health interoperability is contested.

### 2.4.2 *open*EHR

*open*EHR[1] is an international not-for-profit foundation that works in the development of open specifications, open-source software and knowledge resources for the interoperability of EHRs. Unlike HL7's RIM, the *open*EHR model architecture represents the information and ontological models using two distinct systems: the Reference Model and the Archetype Model. The Reference Model contains basic building blocks that represent the information in the EHR and is responsible for the syntactic interoperability between different EHRs. The Archetype Model formalizes the medical domain knowledge contained in the EHR. It defines constraints from which clinical information can be represented using the Reference Model in consistent and interoperable ways. Thus, the model and specifications can represent independently technical and semantic aspects of the system. Despite having a better design, especially due to its dual-information architecture model, *open*EHR is less adopted by healthcare institutions than HL7.

### 2.4.3 CEN/ISO 13606

The 13606 *Health informatics – Electronic health record communication* standard is a European Committee for Standardization (CEN)[2] and an International Organization

[1] `http://www.openehr.org/`
[2] `http://www.cen.eu/cen/pages/default.aspx`

for Standardization (ISO)[1] standard that defines a protocol for semantic interoperability in EHR data exchanging. It is closely related to *open*EHR, particularly in the dual-model system development methodology, where the information about the EHR and the knowledge expressed in its content are kept into different models; but also in the Reference Model. The main difference is that CEN/ISO 13606 is a specification for exchanging EHRs as opposed to the full EHR management proposed by *open*EHR. Moreover, the development philosophy of CEN/ISO 13606 is guided by the idea of fitting all existing EHRs whereas *open*EHR is designed to find the best representation and highest quality of an EHR. The similarity between the *open*EHR and 13606 reference models allow them to be mapped with some effort [53]. However, this mapping comes at the cost of losing some information due to the incompatibility of the richer *open*EHR concepts and the more generic 13606 model.

## 2.5 Terminologies and Ontologies

Standard terminologies and formal ontologies are largely used in data integration systems to foster the syntactic and semantic interoperability between heterogeneous systems[54, 19, 55, 56]. The term ontology comes originally from philosophy, where it refers to the subject of existence, the nature of being. In computer science, the artificial intelligence community has adopted the term ontology as the logic representation of formal knowledge [57]. According to Gruber, an ontology is "*a specification of a conceptualization*" [58]. That is, an ontology describes formally concepts and their relationships in a certain knowledge domain. The main value added of an ontology in a computer system is the possibility of sharing and reusing knowledge in a formalized way. A terminology, or terminological resource, is simply a standard and usually structured set of terms defined for use in a specific subject field to enable clearer communication between different resources. For example, a drug terminology defines a standard set of antibiotic terms for use in the healthcare domain.

There are many different views concerning the definition and roles of terminologies and ontologies in informatics in general, and more specifically in health informatics [59, 57, 60]. In this thesis, we adopt the idea that the main role of an ontology in an integration system is to assure consistence of the knowledge model, rather than the

[1] `http://www.iso.org/iso/home.htm`

completeness. Complementary, the role of a terminology is to provide a comprehensive set of terms that are able to express the concepts of the domain. That is, ontological resources focus on the relationship of concepts whereas terminologies focus on descriptor terms, which captures the essence of subject.

In the next sections, we list some of the most important clinical terminologies and ontological resources based on their relevance to the infectious disease field and their adoption by healthcare institutions.

### 2.5.1 ATC

ATC is a drug classification system developed and maintained by the WHO[1]. In the healthcare domain, it is the drug classification system most used internationally. ATC classifies drug's active substances into five groups according to the organ or system on which they act and their therapeutic, pharmacological and chemical properties. The classification system favors international non-proprietary drug names and is available only in the English language. Translation to other languages is available from the international non-proprietary names' catalogue.

### 2.5.2 ICD

ICD is an international terminology that organizes and codes diagnostic health information for statistics and epidemiology, healthcare management, monitoring and evaluation, research, primary care, prevention and treatment[2]. Like ATC, ICD is sponsored and managed by the WHO and has been adopted by the WHO Member States since 1967. Currently, it is in the operational version 10 (ICD-10) and is accessible in 42 languages, including Arabic, Chinese, English, French, Russian and Spanish. ICD version 11 is under development in a collaborative and interactive process between health and taxonomy experts, and users.

### 2.5.3 LOINC

The Logical Observation Identifiers, Names and Codes (LOINC) is a universal terminology that provides names and codes for identifying individual laboratory and clinical

[1] http://www.whocc.no/atc/structure_and_principles/
[2] http://www.who.int/classifications/icd/en/

observations[1]. The laboratory part of the LOINC database covers terms from chemistry, hematology, serology, microbiology and toxicology. Moreover, it provides auxiliary concepts such as drugs, cell counts and antibiotic susceptibilities. The clinical part of the terminology includes concepts of vital signs, hemodynamics, obstetric ultrasound, cardiac echo, urologic imaging, survey instruments, among others. LOINC includes over 30,000 observation concepts and the original English terminology has been translated to nine other languages, including Chinese, Spanish, Portuguese and French.

### 2.5.4 SNOMED CT

SNOMED CT is a standard clinical terminology[2] for healthcare. SNOMED CT terms are available in the English language and the terminology is processable by machines. It is the most comprehensive clinical vocabulary available in any language, covering large part of the medical vocabulary such as diseases, symptoms, operations, treatments, devices and drugs. Currently, it contains more than 311,000 active concepts [3]. The intellectual property of SNOMED CT belongs to the International Health Terminology Standards Development Organisation, which owns and administers the terminology.

### 2.5.5 UniProt

UniProt provides a knowledge base, including a set of terminologies and ontologies, of information on protein[4]. The database contains high-quality manually annotated and non-redundant protein sequence records. Moreover, it includes auxiliary resources such as species taxonomy and literature citations. For example, the UniProt/NEWT taxonomy database provides a comprehensive terminology of bacteria and other human pathogens, which can be directly processed by machines, since it uses a computer formalized language to represent concepts. UniProt is extensively used by the biology and bioinformatics community but not as much in health informatics. However, we see it as a relevant resource since, as opposed to other biomedical terminologies and ontologies aforementioned, it allows the expansion of clinical concepts to a more genomic-oriented

[1] `http://loinc.org/`
[2] SNOMED CT is also referred as an ontology.
[3] `http://www.ihtsdo.org/snomed-ct/snomed-ct0/`
[4] `http://www.uniprot.org/help/about`

view. Moreover, its microorganism's taxonomy is very comprehensive and up-to-date, covering recent organisms identified in the literature.

Many, if not all, of these terminologies and ontologies present design problems, such as ambiguity, sparseness, lack of operational meaning and coverage [61, 62]. Nevertheless, if applied properly to problems to which they were specifically designed, they can have positive impact on data integration and interoperability [63].

## 2.6 Semantic Web

The literature has proposed several frameworks to foster the interoperability of heterogeneous data sources. Open Database Connectivity (ODBC) [64] was one of the first standards that provided a common application program interface to access relational databases. Additionally, the Common Object Request Broker Architecture (CORBA) [65] standard was developed to enable the interoperability between different software components. These standards have many successful applications to solve technical interoperability [66, 67]. However, as they were not designed to provide further syntactic and semantic homogeneity to the integrated system, they cannot be used as a full interoperability framework.

More recently, the World Wide Web Consortium (W3C)[1] has lead the development of a set of standards and tools to model, store and access information available in the Web. This framework, called Semantic Web, is designed to manage information in *"a web of data that can be processed directly and indirectly by machines"* [68, 18]. Semantic Web has defined a set of technologies that foster the integration of heterogeneous data sources and data models. It provides methods that contribute to solving problems of lack of technical, syntactic and semantic interoperability between systems [69, 70, 71, 72], bringing formal and meaningful representation to heterogeneous information. First, it presents a standard format to encode information called RDF[2], which models web resources in a graph structure. This generic model, in contrast to the entity-relationship model used in traditional databases, facilitates the representation of clinical facts to an unconstrained dimension [73]. Second, it has defined the SPARQL standard that provides ways to access ubiquitously resources available in the Web[3].

[1] `http://www.w3.org/`

[2] `http://www.w3.org/TR/2004/REC-rdf-primer-20040210`

[3] `http://www.w3.org/TR/rdf-sparql-query`

Finally, computer-interpretable ontologies written in the OWL[1] and other languages bring formal conceptualization to RDF resources, improving the quality of data and fostering interoperability between heterogeneous systems.

### 2.6.1 RDF

RDF is a framework for formal conceptualization, description and modeling of information in the Web. It provides a basic information model in a graph structure that encompasses a data format, and a language to represent the definition of concepts and their primary relationships. The RDF format builds on the document structure syntax of the XML language to provide the main language of the Semantic Web. Information in an RDF data model is organized in the form of subject–predicate–object expressions, where concepts are uniquely identified using an Internationalized Resource Identifier (IRI). Hence, RDF concepts can be distinctively referred in the Web, allowing web resources to be linked, originating the idea of linked-data.

A statement in the $(subject, predicate, object)$ form is called *triple* in the Semantic Web parlance. A set of triples composes an RDF graph. As show in Figure 2.1, an RDF graph is a directed graph where nodes correspond to subjects or objects and edges represent predicates and connect subjects with objects. In this context, an RDF subject is a representation of a web resource whereas predicates assert characteristics or aspects of a resource and define binary relationships between subjects and objects. RDF/XML[2] is the normative syntax of RDF but it can also be represented or serialized in several other formats, including N3[3] and Turtle[4]. Many tools are freely available to parse and read these file formats.

### 2.6.2 Semantic Web Ontologies

The core of the Semantic Web is composed by formal ontologies [74, 75]. They provide the semantics, that is, formal and computer-meaningful knowledge representation, and define the domain of applications available as web resources. RDF itself provides a core ontology language that models basic concepts of an RDF document, such as data type

[1] `http://www.w3.org/TR/owl-features`
[2] `http://www.w3.org/TR/rdf-syntax-grammar/`
[3] `http://www.w3.org/TeamSubmission/n3/`
[4] `http://www.w3.org/TeamSubmission/turtle/`

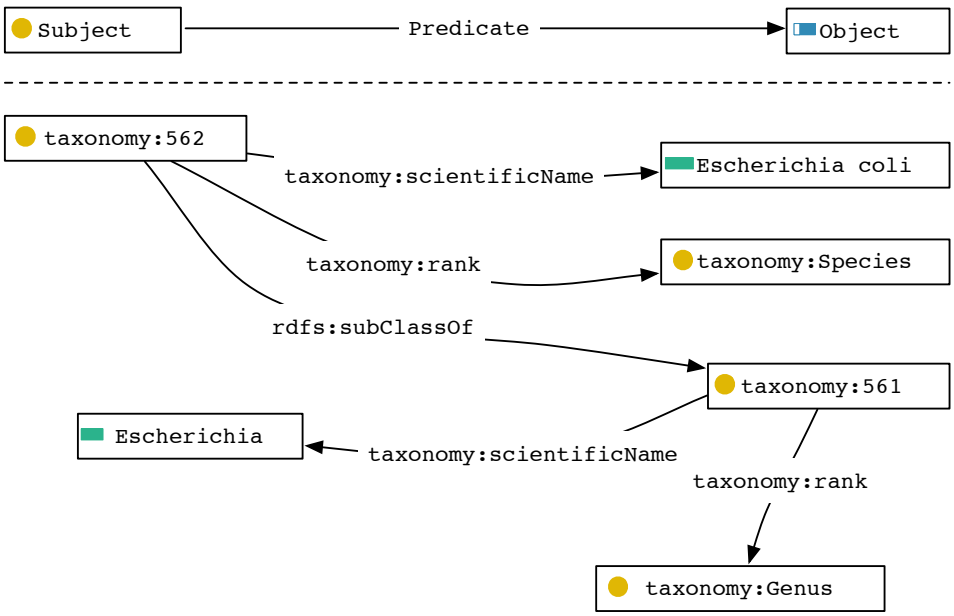


**Figure 2.1: RDF triple and graph example** - Example extracted from UniProt for the representation of *Escherichia coli* taxonomy using a RDF graph.

(*rdf:type*) and group of elements (*rdf:List, rdf:Bag, rdf:Seq*). RDFS extends RDF to provide further constraint and meaning to the RDF concepts. It allows the definition of classes (*rdfs:Class*) and subclasses (*rdfs:subClassOf*), enabling thus the creation of more complex hierarchical structures such as *"is-a"* and *"part-of"* [75].

OWL is the standard semantic web language. It is derived from description logics [76], and extends RDF and RDFS to improve expressiveness and reasoning power of ontologies available in the Semantic Web. The language can be represented using the same syntax of an RDF document and, as RDF, it also provides formal semantics. OWL is available in three versions: OWL Lite for describing a basic system in terms of class, property, subclass relation, and restrictions; OWL DL for system with common formal semantics and inference decidability; and OWL Full for unrestricted OWL vocabulary and syntactic freedom of RDF, but without assuring decidability by the reasoner [77]. Together, the vocabulary and axioms of RDF, RDFS and OWL creates a comprehensive meta-ontology that provides the building blocks for the development of more complex top-level or upper ontologies, and application ontologies.

### 2.6.3 SPARQL

Several query languages, such as the RDF Data Query Language (RDQL) [78] and the Sesame RDF Query Language (SeRQL) [79], have been proposed to access and manage semantic resources in the Web [80, 81]. As of January 15, 2008, the W3C had recommended SPARQL as the standard query language for the Semantic Web and thus, it has become the de facto language for querying RDF data sources. As shown in Figure 2.2, SPARQL is a SQL-like language that uses RDF triples and resources for both the matching part and returning results of the query. The protocol has four querying options: `SELECT`, which is used to select parts of an RDF graph; `CONSTRUCT`, which is used to construct a new RDF graph using an existing RDF graph; `DESCRIBE`, which is used to describe a resource matching a query constraint; and finally `ASK`, which is used to test for the existence of a triple in an RDF graph. The language provides powerful query expressivity for a graph. However, it still presents some limitations, especially for dealing with date/time data types.

```
PREFIX drugbank:  <http://example.com/drugbank/>
SELECT distinct ?drugname ?indication
FROM <http://example.com/graph/>
WHERE {
  ?drug drugbank:genericName ?drugname.
  ?drug drugbank:indication ?indication.
  filter(regex(?indication, 'escherichia coli', 'i'))
}
```

**Figure 2.2: Example of SPARQL query** - This example retrieves the drugs that are indicated to treat *Escherichia coli* infections using an RDF graph such as the one of DrugBank[1]. The `PREFIX` clause associates a label with an resource to short and make cleaner the query statement. The `SELECT` clause specifies the variables that will be returned by the query. The `WHERE` clause asserts the basic graph pattern to match against the RDF data.

## 2.7 Data Integration Approaches

There are several methodologies used to integrate distributed heterogeneous data sources in the literature [42, 82, 83, 84, 85, 86, 44, 87, 88, 89, 90, 91, 92]. They usually fall in

one the categories: *link integration, data warehousing, view integration, workflows* or *mashups* [49]. Depending on the physical location of the databases, these approaches can be classified either as centralized, where there is only one persistent storage, or decentralized, where the data are stored throughout several logically and, generally, physically distributed databases. Moreover, integration systems can be characterized as horizontal or vertical based on the data overlapping. In horizontal systems, there are no overlapping of the data sources content. In contrast, in vertical systems, different data sources store information about the same data. For example, the same patient identifier may be distributed amongst many hospitals in universal healthcare system. Finally, they are also theoretically classified as *global-as-view* (GAV), where concepts in the global schema are mapped to views over the sources and as *local-as-view* (LAV), where the sources are mapped into views over the global schema [40].

### 2.7.1 Data Warehousing

Data warehousing is a data integration architecture that aggregates the data of the sharing sites into one central database, so called *data warehouse*, and queries are executed against this central instance (see Figure 2.3). A unified data model is defined in order to accommodate the information contained in the source sites. Data source specific processes, known as *wrappers*, are created for each participant site in order to extract, transform and clean the local data, so that it fits into the new information model, and to load into the data warehouse. Then, queries can be submitted directly to the central database. Data warehousing favors the transformation of the local data to fit the constraints of the central information model. Some projects that follow this approach are the Integrated Genome Database (IGD) [93], Atlas [88], BioWarehouse [94] and BioDWH [90].

High performance is considered the main advantage of this model. In addition, it allows the data obtained from the sources to be modified, annotated and enriched, since this is a replica, not the production database [95]. There are also well-known techniques to build upon data warehouses dynamic and online data navigation [96] that improve user's data analysis capability. However, to keep the information stored in a data warehouse up to date with respect to the source data and schema model, there is a high maintenance cost, especially for dynamic local sources like LIS and CPOE. As a result, many of the systems that follow this approach end up as data

morgues [49]. Additionally, if there is need for moving data out of the intranet, as in multi-institutional data integration, this approach may shown impracticable. This is particularly valid for medical data integration, where storing patient data out the healthcare's intranets can infringe many patient's privacy and confidentiality rights. Even if identifiers are encrypted or blurred, data owners are reluctant to export full data sets persistently.

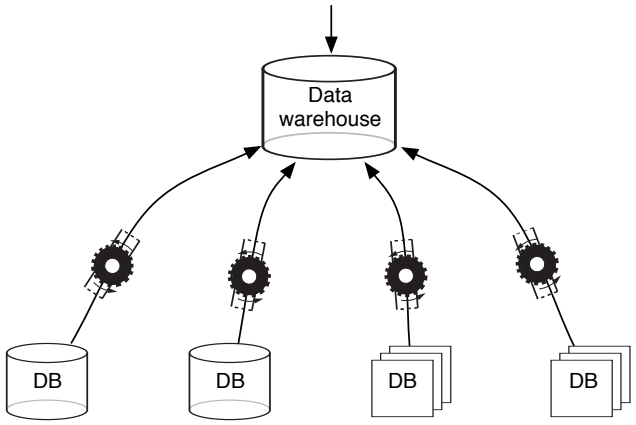


**Figure 2.3: Data warehouse sketch** - Local data are transformed to fit in the central information model. Local wrappers are defined for each data source. They periodically extract, transform and load the data into the central warehouse

### 2.7.2 View Integration

In the view integration approach, the local data are kept only in the source site and a homogenous environment (or view) using all the databases that share information within the system is created at the query time (see Figure 2.4). It offers to the user a unique view of the data even if they are located in different sites with different models. When the query is submitted, it is converted to a common query language that is later handled by a query processor, also called *mediator*. The mediator discovers the source data that needs to be accessed to retrieve the result. Then it splits the query in many subqueries that are passed to different drivers (or wrappers) that will each transform its subquery to the local language using mapping rules metadata and actually access the data. Once the data are fetched in the different sources, the results are globally integrated and returned to the user. Mapping rules are used again to align

the local information into the global information model. The HEterogeneous Multi-database System (HEMSYS) [97], the Stanford-IBM Manager of Multiple Information Sources (TSIMMIS) [83], the Target Informatics Net (TINet) [85], BioKleisli [98] and IBM's Websphere Information Integrator [99] are examples of systems that follow this approach.

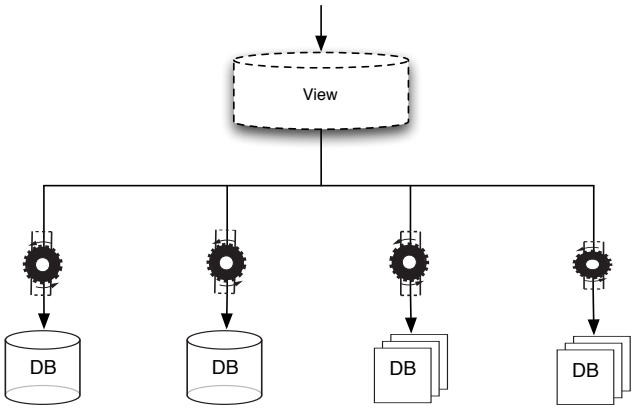


**Figure 2.4: View integration sketch** - Central queries are transformed into local subqueries. Data are fetched locally and aggregated by the central query processor. There is no persistent central storage

Mediated architectures provide two implementation methods that can be used by the mediator to access the local data. First, in the *pull* mode, data are provided to the user at the query time. Local databases are accessed instantaneously after the query is launched. It avoids further delays, but if the system is connected to the production databases, it may disturb normal production operations when there is a high volume of access to the central system, similarly to what happens in a denial of service attack [100]. In the second implementation, *push* mode, once the query is launched the local databases are notified, but they only provide the data after a certain fixed delay delimited by the system. It helps to resolve local performance or/and volume issues in favor of increasing the retrieval time.

Different from the data warehousing approach, view integration is based on the transformation on-the-fly of the central query into local subqueries. Thus, there is no need to centralized the data into a single warehouse, the data are always up to date and there is a (relative) decoupling of the data sources from the integration system. In

the downside, usually the performance is not good, or at least not as good as it could be if the data were centralized in a single data source.

### 2.7.3 Ontology-Driven Data Integration

Ontologies provide common definitions of real-world entities (or concepts) and their relationships using a formal language [19, 77]. As such, they are used as semantic references for the data sources and to make explicit the local and global information models in the data integration systems [101]. Whether applied in centralized or decentralized systems, ontologies may be used to represent the global information model. In this case, they usually serve as terminological or thesaurus services. Applications of lexical ontologies are seen in the Mediator envirOnment for Multiple Information Sources (MOMIS) [86] and the Shared Health Research Information Network (SHRINE) [102] projects. They can also be used to align automatically local and global concepts, having therefore an important role in the data mediation process. In this approach, ontologies are built using a strong formalization language in order to be processed by mediators. Formal ontologies are applied in the COntext INterchange (COIN) [103], PICSEL [84] and Transparent Access to Multiple Bioinformatics Information Sources (TAMBIS) [104] projects.

The literature describes three main methods where ontologies are deployed to support the interoperability of heterogeneous data sources and provide an explicit and machine-understandable conceptualization of a domain (see Figure 2.5): *i*) integration using a single global ontology, *ii*) integration using multiple local ontologies and *iii*) a hybrid approach, with local ontologies and global ontologies [101, 54].

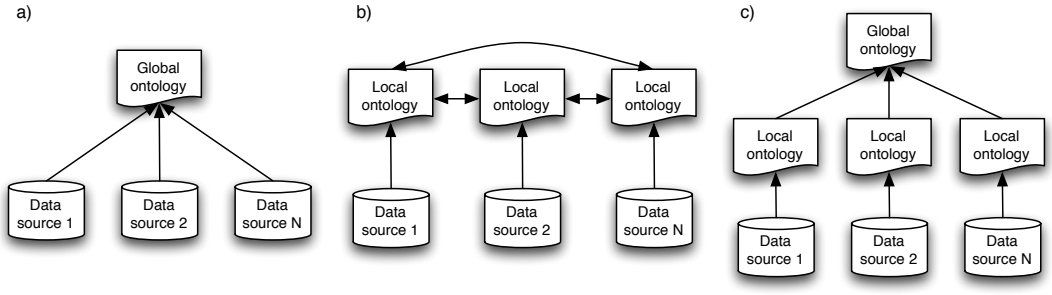


**Figure 2.5: Ontology-driven integration methods** - a) Global (single) ontology, b) Local (multiple) ontology and c) Hybrid (global and local) ontology [101].

In the first approach, a global ontology provides a shared, common vocabulary to define the system's semantics. All information contained in the data sources, which are pertinent to the system's domain knowledge, are represented by this global ontology. The global ontology can be a single monolithic ontology but also the combination of more specific ontologies. Combination is useful especially if several domains need to be represented in the ontology. For example, in drug prescription, the ontology shall model classes of antibiotics but also provide measures to quantify the dose and frequency. Combining smaller ontologies keeps the global ontology modular. In order to represent local information, mappings are created from each local information source model to the global ontology. Thus, local concepts can be unambiguously represented throughout the system. The Scientific Image Management System (SIMS) [105] and TAMBIS [104] are examples of integration systems that follow this model.

In the multiple ontology approach, local information models are formalized by local ontologies. To make local knowledge consistent across the system, mapping amongst the local ontologies are created. For example, in the Knowledge Reuse And Fusion/Transformation (KRAFT) [106] system, the semantics of the information sources are described by local ontologies. Despite of having several ontologies, this model facilitates the development of the ontologies, since the ontologies developed are simpler, fits exactly to the knowledge presented in the local sites and can be developed by local data experts. However, due to the lack of a common model, mapping between local ontologies is very difficult. Thus, in practice, this approach is followed by few integration systems [101].

The third approach uses a combination of a global domain ontology with local information source ontologies [103, 84]. In this hybrid of local and global formalization, local ontologies describe the semantics of each local source and the global ontology describes the basic, common terms of the domain. To make the information consistent across the sources, the local ontologies subscribe to the common, top-level global vocabulary. Thus, there is no need to link the local ontologies. For example, the INtelligent Data Understanding System (INDUS) [107] uses a query-centric approach, where concepts in the global ontology are mapped to concepts in the local ontologies. The global ontology, which is created by the user, defines entities and relationships in the domain of discourse and is used to hide the complexity of the information in the data sources. The user specifies the mappings between the global and the local ontologies and INDUS is

responsible for discovering the data sources, creating the query plan and aggregating the results. The drawback of the INDUS's model is that the user has to know precisely how information in the global model is represented in the local sources, an expertise rarely found, especially in multinational integration systems.

### 2.7.4 Other Integration Methods

*Link integration* is a successful though limited approach to database integration. It is employed by systems like the Sequence Retrieval System (SRS) [82], Entrez [108] and Integr8 [109]. Integration between disparate data sources, usually web pages, is performed using cross references links, which are managed by the integration system (see Figure 2.6). The system provides references to the source data but does not offer any further computing or reasoning power.

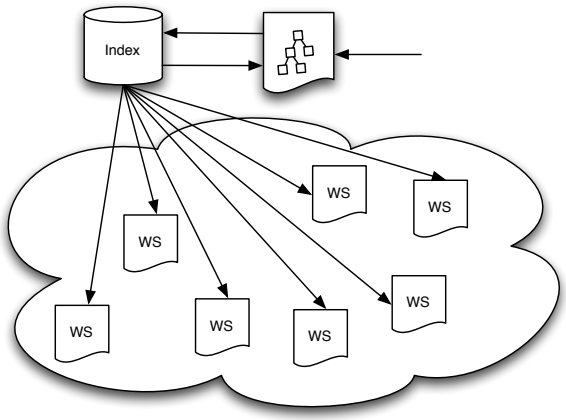


**Figure 2.6: Link integration approach** - The data integration system indexes web resources (*ws*) and provides links (references) as answers to the user queries

*Workflows* are techniques to describe and connect series of related process. The integration system tries to streamline the workflow creation and execution process so that users can design and execute analytical procedures repeatedly with minimal effort. Taverna (academic) [87] and InforSense (commercial) [110] are examples of workflow environments.

Finally, *mashups* provide means to integrate information from multiple web resources into a single new web application [70]. One of the first healthcare and life science mashup examples was the use of Google Earth to track the global spread of

avian flu [89]. Mashups development framework such as Yahoo! Pipes [111] speeds up the creation of new integrated web resources. They make the design of mashups similar to a workflow development but using only web-based resources. Figure 2.7 shows an overview of a basic mashup system.

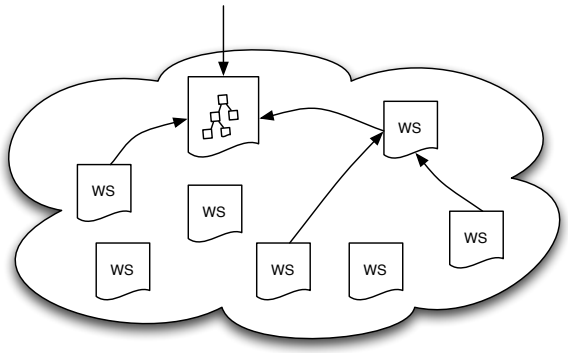


**Figure 2.7: Mashups** - Web resources (*ws*) are integrated on-demand using mashup technologies.

## 2.8 Existing Integration Systems in Life Sciences

In the literature, many systems had been designed to integrate data from heterogeneous life sciences data sources. In this section, we discuss some representative of link integration, data warehousing, (ontology-driven) view integration and service-oriented systems that were introduced in the last decades.

### 2.8.1 SRS

SRS is a link integration system designed to work with molecular biology data banks in the Web. The system uses indexed flat files to store cross-reference links from different data sources. The SRS model is based on a document retrieval engine, working similarly to web-based search tools, where the user can create queries using keywords terms (title, author name, etc.) in Boolean combination [82]. SRS is popular among biologists because it is easy to use and was one of the first data integration systems to provided a specialized search tool for heterogeneous bioinformatics web resources.

SRS model is reasonably effective because there is a strong overlap of terms in indexed the corpora, enabling their linkage. However, the system does not extend readily

to data types other than sequence databanks, for which it was originally designed. Even for sequence data, it neither provides a rich query language nor optimization capabilities. In addition, there is no conceptual model and, like BioKleisli [98], the integration and location details are not transparent. Thus, a large part of the integration onus is on the user side. Furthermore, output results are not machine readable, requiring further parsing if the system is used by other applications.

### 2.8.2 BioDWH

BioDWH is an open source data integration software kit. It intends to increase customization of the data warehouse concept with the advantages of better performance, scalability, synchronization and data quality [90]. The system architecture provides a data warehouse infrastructure that is independent from the underlying relational database management system. It has a monitoring component that tracks changes on the various data sources. When a source is updated, it downloads the data from the original source and extracts the compressed data in a local directory. Then, it is possible to start the synchronization of the data warehouse. BioDWH claims to have advantages when compared to other systems by providing up-to-date information, platform and database independence as well as high usability and customization. However, the system is not suitable to integrate data sources that change frequently, since it will not scale up with data updating.

### 2.8.3 TAMBIS

TAMBIS project was designed to help bioscientists with tasks of choosing, combining and interacting with biological data resources in order to retrieve research information. The system architecture is service oriented and based on an extensive source independent, global and formal ontology, created using the Description Logic GALEN Representation and Integration Language (GRAIL) (currently updated to the DARPA Agent Markup Language+Ontology Inference Layer (DAML+OIL)) [104]. A terminology server is responsible for the ontology management. The ontology is deployed in the user interface for data navigation and to transform queries into an intermediate language, understandable by the Collection Program Language (CPL), TAMBIS' common query language.

In TAMBIS, the query process is realized in three phases. First, there is a query formulation process, where the user creates the request using terms and relationships formalized in the TAMBIS ontology. The result is a source independent conceptual query. Second, this conceptual query is transformed into local queries understandable by the local databases. The system looks for the ontology terms present in the conceptual query and selects the data sources needed to answer the query constraints. Then, a query plan is constructed for each local source. Finally, the tool executes the queries against the selected data sources and returns the results to the user.

### 2.8.4 caGrid

The caGrid system was developed inside the cancer Biomedical Informatics Grid (caBIG) project, supported by the National Cancer Institute (USA). It intended to address the need for standard applications, common data models and software infrastructure to enable efficient access to and sharing of distributed computational resources in cancer research [112]. caGrid is based on a model-driven and service-oriented architecture that provides a framework for the advertising, discovering and invocation of data and analytical resources in a Grid environment. The framework is built using three Grid middleware frameworks: Globus Toolkit [113], OGSA-DAI [114] and Mobius [115].

In the caGrid approach, distributed resources are represented as Grid services and communications between services and clients are done using Grid protocols. The Globus Toolkit is used as the core Grid middleware for creation, deployment and invocation of Grid services. Other Grid services are built on top of toolkit. OGSA-DAI virtualizes the local data sources as Grid data services and manages the queries in the distributed environment. Finally, Mobius is employed to support Grid-wide management of XML schemas representing the structure of common data types in the caBIG domain.

### 2.8.5 OntoFusion

OntoFusion was designed within the INFOGENMED framework to provide unified access to data sources that are publicly available over the internet [116]. The system architecture is based on a multi-agent platform and the integration approach is ontology-driven, where source databases are mapped and unified using ontologies. The

JADE platform[1] is used to enable the execution of different parts of the system on different machines. The user interface allows the user to explore the hierarchy of virtual data repositories, that is, virtual schemas obtained through the mapping or unification process of the heterogeneous data sources, and to issue queries over the unified system.

In OntoFusion, a data mediator module coordinates the access to the virtual repositories. Each virtual repository is assigned to an individual agent, which execute queries issued against its repository and returns the output to the mediator. In addition, an agent can provide its virtual schema to other agents. Virtual schemas are stored using DAML+OIL ontology language and RDQL is used as the query language. The mediator distributes the user queries to the different agents and merges the results. A database access module is responsible for the communication with the physical database systems. It contains wrappers that translate queries from the intermediate query language (RDQL) into the local query languages.

OntoFusion adopts a query translation approach. In contrast to systems like TAMBIS, it does not use a single conceptual schema, avoiding changes on the global conceptualization when adding or removing a database. The vocabulary issue is solved with the multiple virtual schema, where conceptual schemas for all databases are created using terminology from a domain ontology. Hence, objects from one schema and all their semantic counterparts from conceptual schemas of other databases will have a standardized term.

### 2.8.6 SHRINE

In SHRINE [102], the authors developed a prototype federated query tool for clinical data repositories. The system aimed to serve as a framework that would foster scalable collaboration across different healthcare institutions, allowing secure sharing of patient information. The query manager tool, which is based on web services, provides real-time aggregate counts for number of patients having a certain clinical condition. It distributes user queries to the different network peers, which return the number of patients matching the query constraints. Queries are executed locally by local adaptors, which can be created for each source peer to account for local specificity. However, this is not a easy task and requires deep understanding of both the local system and the

[1] http://jade.tilab.com/

query manager architecture. The system was evaluated with only one type of backend – i2b2 [117].

SHRINE is strongly focused on the security and privacy aspects of sharing patient data on real-time. For example, the query engine provides the patient counts added by an small variation error, so that patients can not be tracked by a combination of different queries. The system also blocks user accounts in case they run queries using the same parameters repeatedly. Finally, results matching less than 10 patients will be answered symbolically as "less than 10". All queries are logged so that audit processes can be performed. The system uses X509 certificates to ensure that a peer is truly allowed to retrieve information from the network.

### 2.8.7 The DebugIT Project

The Detecting and Eliminating Bacteria Using Information Technology (DebugIT) project [118] was run by a consortium of 14 industrial, research and clinical institutions from nine countries that collaborated to build a framework for sharing antimicrobial resistance data from clinical information systems in a Europeanwide context. The project aimed to reuse existing clinical data for generating new knowledge to be incorporated in decision support and monitoring engines at the point of care and for developing prevention strategies at policy levels. The project was funded by the European Union Seventh Framework Programme and run from January 2008 to June 2012.

The DebugIT architecture showed in Figure 2.8 is based on distributed services that exchange information using Semantic Web technologies. Each service is represented as a semantic endpoint that communicates in the SPARQL protocol and uses RDF framework to exchange messages. The DebugIT Core Ontology (DCO) [119] describes and formalizes the DebugIT domain. A semantic interoperability platform coordinates the access to the different systems. It uses DCO as the central glue for the semantic endpoints. The architecture and experiments realized in this thesis, in special the transnational antimicrobial resistance monitoring system, were developed under the DebugIT project and follows the principles of data sharing via the Semantic Web.

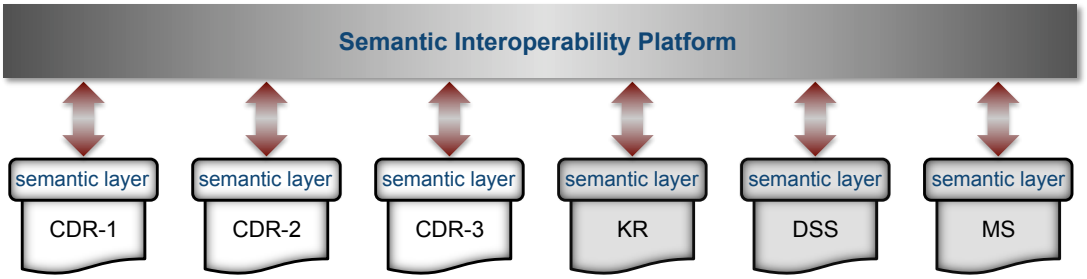


**Figure 2.8: Architecture of the DebugIT framework** - Components of the architecture, such as the clinical data repository (CDR), knowledge repository (KR), decision support system (DSS), and monitoring system (MS), are interconnected using the SPARQL protocol through the Internet bus. Messages are transferred in the RDF format, and ontologies formalize the data model and content.

## 2.9 Summary

A plethora of data integration systems can be found in the literature. Since the popularization of database management systems in the middle of the past century, and, in a ever more larger scale, lately in the past two decades with the mass adoption of the Web in sciences, medicine and engineering, among other areas, data have been stored in digital databases of several types (relational, XML, object oriented, etc.). In the era of knowledge, individuals and organizations have realized the power of integrated systems and several frameworks are being created to unify and provide easier, faster and cleaver methods to access, visualize and analyze information stored in these distributed and heterogeneous resources.

However, despite the large number of integration systems available, most of them are very specific to the task that they were designed. They can work essentially in the context for which they were created, respecting project constraints, such as data source types, information model, etc. As an example, the architectures designed to integrate biological data sources cannot be applied directly in the integration of medical data sources. First, in medical data integration there is a high concern with patient's privacy rights. Information needs to be anonymized and centralization is usually a constraint. While in biological data, such a bacteria and protozoa data banks, this is not a concern. Second, the number of biological data sources that are part of a biological integration system is relatively small. For example, the Linked Life Data[1] project

[1] `http://linkedlifedata.com/sources`

currently integrates 25 data sources. This situation is very different for medical data integration, such as biosurveillance and randomized clinical trials, where the number of sources that could eventually be part of the system is much larger ($\gg 100$).

Semantic Web, on the other hand, rather than being designed as a specific integration system, it provides a standard, generic and widely used set of tools that can be applied to foster the development of integration systems. Amongst other benefits, this increases the portability of the semantically-formalized data sources, allowing them to be (re-)used relatively easier in applications for which the system was not originally projected. Therefore, we select Semantic Web as our framework of choice to support the design of a transnational antimicrobial resistance monitoring system.

# 3

# Modeling and Formalizing Antimicrobial Resistance Data and Sources using Semantic Technologies

## 3.1 Introduction

In this chapter, we first provide a brief overview of antimicrobial susceptibility tests. Then, we propose a method to formalize microbiology data sources using Semantic Web technologies. This is an expansion of the works presented in [120, 121]. Finally, we do a preliminary evaluation of the performance of the formal endpoints. This model will serve as basis for our transnational monitoring system described in the next chapters.

## 3.2 Antimicrobial Susceptibility Tests

Microbe's resistance to antimicrobials is a normal biological phenomenon. In the history of antimicrobial development, the introduction of every agent into clinical practice has triggered the selection of resistance strains. Resistance is characterized by the capability of microorganisms to survive and multiply while exposed to drug concentration levels higher than the tolerated by the human or other living media. Pathogens are naturally resistant to (or not affected by) many antimicrobials. However, resistance can also be developed through evolutionary processes, such as mutation and lateral

gene transfer[1] [122, 123, 124]. In both cases, genes encode different mechanisms that inhibit the action of antimicrobials. These mechanisms may be presented in entire species or exist only in some strains. Moreover, they may be efficient not only with a specific antimicrobial, but sometimes they can neutralize the effect of a whole class of drugs. Further, pathogens can become resistance to multiple drugs used in clinical therapies, a phenomenon called multi-drug resistance. The microbes that carry these multi-resistant genes are informally known as superbugs. Examples of such superbugs, or super bacteria in these specific cases, are the methicillin-resistant *Staphylococcus aureus* (MRSA) and the vancomycin-resistant *Enterococcus* (VRE) bacterium strains.

In order to detect resistant strains, antimicrobial susceptibility tests are performed in microbiology laboratories. Whether based on standard phenotypic antibiograms or more recently on genetic screening tests, these methods follow a basic workflow composed by the steps: *i*) specimen collection, *ii*) microbe culturing, *iii*) microbe isolation and *iv*) susceptibility testing. In the first step, a clinical sample is extracted from the patient presenting the infection symptoms. Then, this material is cultivated in a rich organic media so that microbes that are causing the infection can grow. From the microbes detected in the culturing media, those, which are likely pathogenic[2], are then isolated and subjected to the resistance tests. In case of standard phenotypic antibiograms, these microorganisms will be put in the presence of several antimicrobials, as shown in Figure 3.1, so that their susceptibility can be tested. *Escherichia coli*, for example, will be usually tested against ampicillin, ceftriaxone, gentamicin, trimethoprim-sulfamethoxazole, ciprofloxacin and amoxicillin-clavulanic acid. Depending on how much the isolated pathogen is able to grow in the presence of these antimicrobials or, in other words, on the lowest concentration of the antimicrobial that will be able to inhibit the visible growth of the microbe, that is, the minimum inhibitory concentration (MIC), the result of the test will be defined in terms of the breakpoint values as *sensitive* (S), *intermediate* (I) or *resistant* (R). Unlike phenotype-based antibiograms, in genetic screening tests, the susceptibility test will search for the presence of known resistant genes in the microbe's DNA sequence. In this case, instead of reporting the breakpoint values, the test will reveal the presence or absence of the resistant

[1] Resistant genes are acquired from other microorganisms.

[2] Pathogenicity will depend among other factors on the species and the site.

gene(s). For example, it could reveal the presence of the *mecA* gene [125], which will characterize methicillin resistance.

## 3.3 Modeling and Formalization of Microbiology Databases

Since the first discoveries of resistance, local, national and international antimicrobial resistance data are being produced on routine daily tests in microbiology laboratories. With the increasing availability of information systems in hospitals and laboratories, the content of these tests have been stored in digital databases and integrated into laboratory and clinical information systems all around the world. With the awareness of the potential benefits that the gathered data could have on quality of care, more and more centers have started to use them for secondary purposes, including supporting empirical therapy, infection control and antimicrobial policy decision making, and helping to create prescription guidelines and clinical alerts.

However, as it has happened with other information databases, this electronic data growth was not followed by the standardization and formalization of the information technologies involved in the process. As a result, hospital information systems have become data islands that are difficult to access and integrate [126]. Still nowadays with all the advances in storage technologies, such as cloud and grid computing, in practice, clinical databases even within the same institution does not interoperate well. As an example, in a study about the implementation of CPOE in the United States, Germany, the United Kingdom, France, the Netherlands, Switzerland and Australia [127], Aarts and Koppel found that integration of CPOE with patients electronic records is inexistent in Germany, France, the Netherlands and Australia, and was available in only one Swiss hospital.

In order to integrate these microbiology databases we propose an approach based on formal and standard semantics. More specifically, our hypotheses is that Semantic Web technologies could enable the development of eHealth surveillance networks, providing common meaning to the integrated system while respecting local specificities. Different from existing data integration systems, where a central query engine is designed and the local sources are modified to adjust to the central system, in our approach we start by modeling and formalizing the data sources to create what we call the *local Clinical Data Repository* (lCDR) [120, 121]. Only then that we will define our central engine


**Figure 3.1: Antimicrobial susceptibility test by disk diffusion on Müller-Hinton agar of an enterobacteria** - It shows resistance to CRO (ceftriaxone = Rocephin = 3rd generation cephalosporin), with the synergy in-between AMC (amoxicillin-clavulanic acid) and CRO and other cephalosporins. This is the sign of the production of an ESBL (Extended-spectrum betalactamase). This image has being provided courtesy of Dr. Stéphane Emonet of the Hôpitaux Universitaires de Genève, Geneva, Switzerland.

to coordinate the access to the local endpoints. That is, we perform a bottom-up approach to data integration. We assume in our model that the institutions that we will integrate have already a microbiology database with some content. Further, in order to simplify our design, we consider that these data are stored and managed in a relational database. In the next chapter, we will see that this constraint can be relaxed.

### 3.3.1 Microbiology Databases

Independent of the healthcare setting where an antimicrobial susceptibility test has been performed, the report is composed essentially of four main concepts: the anatomical site where the specimen was collected, the microorganism isolated, the antimicrobial tested, and its respective susceptibility. Additionally, some complementary information is commonly capture by the data model such as the time when the sample was collected and when the test was reported, some demographics data including the patient gender and age (or date of birth), which are associated to a patient identifier, and finally some geographic or organizational data, such as the patient's ward, the clinical setting and the laboratory performing the test. Sometimes, in more complete reports, we can also find the MIC values and the amount of overnight bacteria growth, which is used, among other factors, to differentiate between infection and colonization. Table 3.3 provides an example of a basic microbiology laboratory report.

Figure 3.2 shows an example of a relational databases storing microbiology reports. We have three main elements in these databases – the *data model*, the *data set* and the *data source*. The data source is the database itself. It stores and manages all the information contained in the microbiology reports. The data model defines the content of the data source. It organizes the database information and specifies how they are stored and accessed. A data model is composed of a set of *concepts* and their relationships. Here, we consider that a concept is the representation of an abstract thing and is defined by a set of relationships and, eventually, recursively by other concepts. A data model contains also *data elements*, which are atomic units of information. They are unambiguously defined with precise semantics. Finally, the *data set* is a result of a data model instantiation, for example, as the product of a database query. It contains a set of data elements and their respective values.

Our goal in the modeling of a microbiology database is to formalize these three main elements – data model, data set and data source – so that they can be ubiquitously

| Concept | Data element | Data value | Field size (Byte) |
|---|---|---|---|
| Patient | Patient ID | 20639467 | 10 |
| | Date of birth | 1991-01-01 | 10 |
| | Gender | F | 1 |
| Organization | Organization ID | 105 | 10 |
| | Ward | floor-1-ac | 45 |
| | Service | Cardiology | 45 |
| | Department | Internal medicine | 45 |
| | Organization name | Hospital 1 | 45 |
| Culture | Culture ID | 1910181 | 10 |
| | Patient ID | 20639467 | 10 |
| | Organization ID | 105 | 10 |
| | Collect day | 2012-01-01 | 10 |
| | Specimen | urine | 45 |
| | Organism | E. coli | 45 |
| | Quantity | $> 10^5$ | 10 |
| **Total** | | | **351** |
| Antibiogram | Antibiogram ID | 3180102 | 10 |
| | Culture ID | 1910181 | 10 |
| | Antimicrobial | cefepime | 45 |
| | Susceptibility | S | 1 |
| | MIC | 0.96 | 10 |
| **Total** | | | **76** |

**Table 3.1: Example of microbiology result report** - Example and size of concepts use in a routine antimicrobial susceptibility test report.

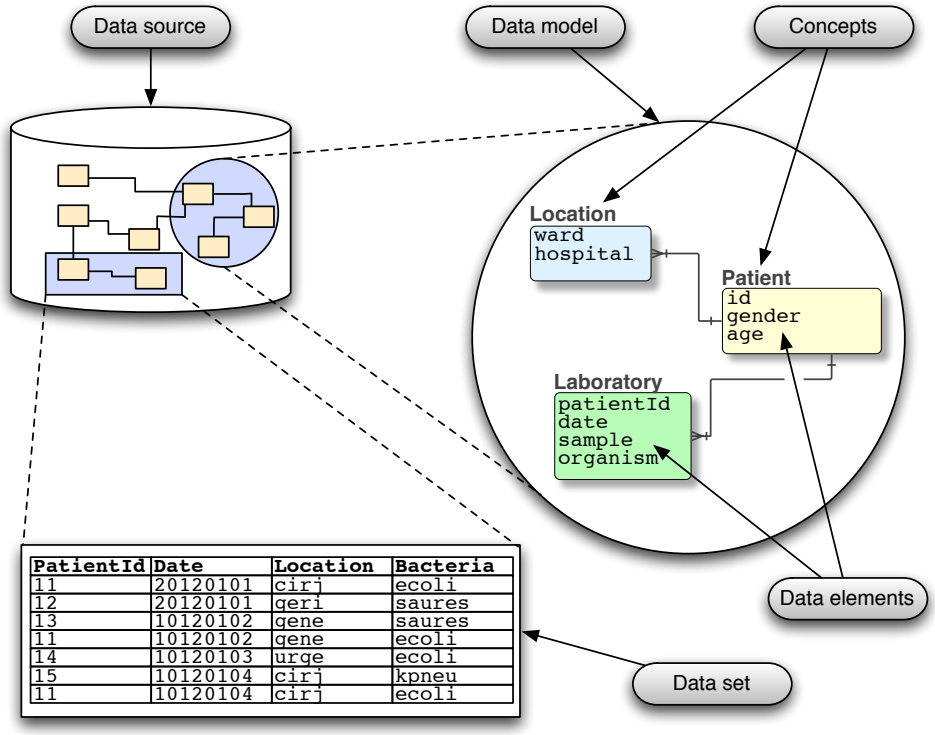


**Figure 3.2: Antimicrobial resistance information dimension** - Resistance information varies in time and location (departments and wards in a single institution, and different organizations in a network monitoring system) [128].

accessed using a formal query language. By formalization, we consider the process of transforming the underlying database content into a formal language representation that computers can understand. That is achieved by defining these elements, or more specifically, their classes, properties, instances and relationships, using a formal ontology language. As we described in the previous chapter, semantic web ontology languages, such as RDF, RDFS and OWL, are all formally defined and provide the means to represent these elements in a computational and semantically formalized way. We will employ these ontologies to formalize the microbiology databases.

### 3.3.2 Standardization in Laboratory Systems

Hitherto, most of the systems storing antimicrobial resistance information use local naming conventions. Despite the availability of standard biomedical terminologies, such as LOINC and SNOMED CT, they are still hardly adopted into local clinical information systems. There are many reasons for that as already pointed by Rector [61]. In the next headings, we highlight four factors we believe have significant impact on terminology adoption into clinical systems:

- Technological momentum [129]. In general, information technologies have a certain momentum to be widely adopted. Existing systems, as long as they work, tend to be kept in place. Since current operation systems were built before or together with the advent of many of these terminologies, they will need some time to be spread and actually deployed into operational databases.

- Language barrier [61]. Standard biomedical terminologies should cover concepts in many languages because healthcare workers need to access information in their own language, amongst other reasons, to avoid translation ambiguities. However, despite the efforts from international health organization, such as the WHO, to have terminologies in several languages, most of the main biomedical terminologies are still very restricted to western European languages, especially English.

- Growing complexity. As standard terminologies, they try to cover as much as possible the knowledge field and many times they become very complex (size wise) to use for both the system developer and the healthcare worker dealing with the information system. For example, the UniProt/NEWT terminology presents a

very comprehensive and even machine readable taxonomy for microbes. However, it includes the whole range of class, genus, species, strains, etc. Just for bacteria, there are currently 238,662 terms. Therefore, it would not be readily suitable for systems that, for example, are interested only in most virulent or resistant pathogens.

- Operational names. Many times standard terminologies use labels that are very distant from healthcare practice. For instance, methicillin, a beta-lactam drug compound of the penicillin class, is an antibiotic no longer manufactured but the term is still used to refer to a class of antibiotics that includes cloxacillin, oxacillin and flucloxacillin, as in methicillin-resistant *Staphylococcus aureus*. However, in some drug terminologies, antibiotics with similar effect are grouped under other labels, such as *Beta-lactamase resistant penicillins* (WHO-ATC - code J01CF), and the methicillin term is inexistent. As an example, a quick search in Pubmed[1] for the term *methicillin* retrieves more than 22 thousand documents while the term *Beta-lactamase resistant penicillins* is found in less than 20 documents. That is, these terminologies neglect some very important operational names while some irrelevant, from the operational view point, are present in their vocabulary.

Comparable to the lack of standardization in the representation of microbiology data sets, data models are usually not derived from standard healthcare information models either. Regardless of standardization and formalization efforts such as *open*EHR and HL7-RIM, microbiology information models are still developed "in-house". The reasons are similar to the lack of standard nomenclature in data sets, in particular, the complexity (or generality) of the standard reference models. In addition, standard information models are sometimes not well designed for relational databases, such as the HL7-RIM. A substantial work is required to adapt them to a relational database management system and the result is not necessary an interoperable system. Furthermore, the lack of a single actual standard model, which would truly allow de facto interoperability, compromises their adoption. Thus, we do not consider that any standardization and formalization are presented in the local microbiology databases, neither at the data set nor at the data model levels.

[1] `http://www.ncbi.nlm.nih.gov/pubmed/`

### 3.3.3 Formal Data Model

A formal data model for a database, which we will call the *data definition ontology* (DDO), defines all the concepts and data elements of the source model using a formal language. The easiest way to formalize a data model is to create a direct map from the database schema to an ontological model. In a direct mapping, the resulting structure of the DDO is a copy of the data model and the DDO vocabulary directly reflects the table and column names of the source model. In the conversion process, the structure and the vocabulary are not modified, or at least, not significantly that it cannot be expressed by trivial mapping rules, such as string conversion. Several authors have used this approach to formalize relational databases [130, 131, 132]. We build on these related works to define our formalization rules.

First, let us assume that we have a normalized relational database schema, at least up to third normal form. In our approach, we define a set of rules that converts the tables and columns of the source model into ontological classes and properties. More specifically, we use the RDF graph format, the RDF(S) and OWL semantic web languages and the XML Schema Definition Language (XSD)[1] to specify the output DDO. In our notation, we employ the Turtle syntax to facilitate reading. Italic terms are placeholder variables for which particular values coming from the data model are supplied. These rules are defined as follow:

**Rule 1** - Tables are mapped to classes in the DDO and the table name defines the class IRI. Hence, in the Turtle syntax, a table is formalized as:
`ddo:`*TableName* `a rdfs:Class .`

**Rule 2** - Columns are mapped to class properties or individuals in the DDO.

**Rule 2.1** - Primary key columns represents an individual, that is, an instantiation of a class. Therefore, they are not mapped in the DDO, which represents only information about the data.

**Rule 2.2** - Foreign key columns represent relations between instances of two classes. They are defined as object properties and the column name defines the property IRI:
`ddo:`*columnName* `a owl:ObjectProperty .`

[1] `http://www.w3.org/TR/xmlschema-2/`

**Rule 2.3** - Data columns, that is, those that are neither primary keys nor foreign keys, represent relations between instances of classes and literal values. They are defined as datatype properties and the column name defines the property IRI:
`ddo:`*columnName* `owl:DatatypeProperty .`

**Rule 3** - Property value ranges are restricted using the column data types or the reference table in case of foreign keys. Restrictions are anonymous or blank node classes.

**Rule 3.1** - Properties derived from foreign key columns range within their reference class type:

```
_:x a owl:Restriction ;
    owl:onProperty ddo:columnName ;
    owl:allValuesFrom ddo:TableName .
```

**Rule 3.2** - Properties derived from data columns range within primitive datatype values:

```
_:x a owl:Restriction ;
    owl:onProperty ddo:columnName ;
    owl:allValuesFrom xsd:dataType .
```

**Rule 4** - In description logic models, classes are represented by a set of subsumption relations derived from their properties. For example, let us suppose we have an antibiogram with some antibiotic tested and some respective susceptibility. In OWL we state this by saying: *"Antibiogram is a subclass of all things that have some tested antibiotics and some respective susceptibility"*. We use the same approach to enrich the class concepts with their restriction properties (columns):

```
ddo:TableName rdfs:subClassOf
    [         a owl:Restriction ;
              owl:onProperty ddo:columnName1 ;
              owl:allValuesFrom ddo:columnDataType1 ] ,
    ...
    [         a owl:Restriction ;
```

owl:onProperty ddo:*columnNameN* ;
owl:allValuesFrom ddo:*columnDataTypeN* ] .

As further demonstrated in [133], these rules can be fully automatized.

By applying the above rules to a database data model, we can formalize the data model structure and have it processable by reasoners, such as the ones implemented by some SPARQL engines[1]. For example, let us consider the data model provided in Figure 3.3 as the representation of a microbiology laboratory report in a relational database. Then, if we apply the aforementioned rules to the table Antibiogram, it will result in the DDO showed in Figure 3.4.

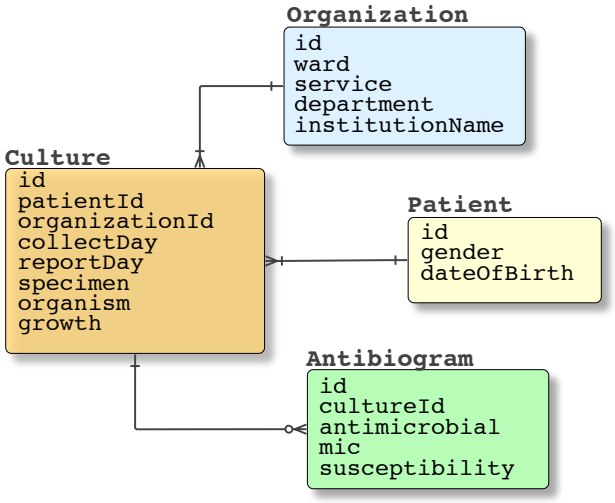


**Figure 3.3: Basic antimicrobial susceptibility test information model** - An organization produces several microbiology (culturing) reports. A cultured microorganism can be tested against several antimicrobials or not. A patient may have several microbiology reports.

### 3.3.4 Formal Data Source

Having formalized the data model, we will now formalize the data source. To be more precise, we will connect the formalized data model to the underlying non-formal database so that it can provide data in the RDF format and be accessed using the

[1] http://jena.apache.org/

```
#namespace declaration
@prefix rdfs : <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl  : <http://www.w3.org/2002/07/owl#> .
@prefix xsd  : <http://www.w3.org/2002/07/owl#> .
@prefix ddo  : <http://localhost:8080/ddo#> .

#definition of the antibiogram class
ddo:Antibiogram   a    rdfs:Class ;
                  rdfs:subClassOf
                  [    a owl:Restriction;
                       owl:onProperty ddo:cultureId;
                       owl:allValuesFrom ddo:Culture ] ,
                  [    a owl:Restriction;
                       owl:onProperty ddo:antimicrobial;
                       owl:allValuesFrom xsd:string ] ,
                  [    a owl:Restriction;
                       owl:onProperty ddo:mic;
                       owl:allValuesFrom xsd:float ] ,
                  [    a owl:Restriction;
                       owl:onProperty ddo:susceptibility;
                       owl:allValuesFrom xsd:string ] .

#definition of the culture properties
ddo:cultureId        a owl:ObjectProperty .
ddo:antimicrobial    a owl:DatatypeProperty .
ddo:mic              a owl:DatatypeProperty .
ddo:susceptibility   a owl:DatatypeProperty .
```

**Figure 3.4: Example of data model formalization** - The DDO is an RDF graph written using the Turtle syntax, where database tables become OWL classes and columns become properties. Data types are derived from foreign key relations and columns types.

SPARQL query protocol. The resulting system, containing a formal data model (DDO) and communicating via SPARQL protocol, will be our lCDR.

So far, the literature has provided two alternatives to formalize existing data sources. First, we can store the non-formal model and data into a formal semantic-complying storage system, such as the RDF stores provided by Sesame[1] and Jena[2]. In this methodology, specific software (wrappers) shall be developed to extract the data from the relational databases, transform them to the DDO model and then load into the RDF store. Second, we can transform on-the-fly the original data source into a semantic-complying storage system using some transformation rules. This so called RDB-to-RDF approach is implemented by several engines, including D2RQ[130] and Triplify [134]. In special, W3C has been developing a standard candidate language to map relational data models to RDF – the R2RML language[3]. R2RML is a powerful transformation language and is starting to be adopted by some RDB-to-RDF transformation engines, such as Virtuoso[4]. Using an RDB-to-RDF approach, the relational database will be regarded as a virtual RDF graph but the data will be only persistently stored in the relational database. The advantage of this methodology is that it does not require to modify the existing relational data sources to represent them as an RDF graph. Moreover, security constraints, which usually lacks in native triple stores, can be applied to the relational databases increasing the trust on the system.

Let us consider the example of D2R. The system offers virtual access through a SPARQL interface that queries the underlying relational database and provides the output data set in the RDF format. The system specifies a map between D2R classes and properties and the source database tables and columns. These mappings are used to convert SPARQL queries into SQL when the queries are issued against the SPARQL endpoint. The results are then converted back to RDF using a reverse engineering process based also on the mapping constraints. For example, in the D2R map of Figure 3.5, the `d2rq:ClassMap` class is used to represent a class of an ontology and to define how instances of the class are identified in the database. The parts between @ identify the database columns that contain the class individuals, that is, the primary key columns. D2R properties are linked to database columns using the `d2rq:PropertyBridge` class.

[1] `http://www.openrdf.org/`
[2] `http://jena.apache.org/`
[3] `http://www.w3.org/TR/r2rml/`
[4] `http://virtuoso.openlinksw.com/`

```
@prefix d2rq:   <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix map:    <http://localhost/mapping#> .

# Table Antibiogram

map:Antibiogram                 a    d2rq:ClassMap ;
    d2rq:dataStorage            map:database ;
    d2rq:uriPattern             'Antibiogram/@@Antibiogram.id@@' ;
    d2rq:class                  ddo:Antibiogram .

map:cultureId                   a    d2rq:PropertyBridge ;
    d2rq:belongsToClassMap      map:Antibiogram ;
    d2rq:property               ddo:cultureId ;
    d2rq:refersToClassMap       map:Culture ;
    d2rq:join                   'Antibiogram.cultureId => Culture.id' .

map:antimicrobial               a    d2rq:PropertyBridge ;
    d2rq:belongsToClassMap      map:Antibiogram ;
    d2rq:property               ddo:antimicrobial ;
    d2rq:column                 'Antibiogram.antimicrobial' ;
    d2rq:datatype               xsd:string .

map:mic                         a    d2rq:PropertyBridge ;
    d2rq:belongsToClassMap      map:Antibiogram ;
    d2rq:property               ddo:mic ;
    d2rq:column                 'Antibiogram.mic' ;
    d2rq:datatype               xsd:float .

map:susceptibility              a    d2rq:PropertyBridge ;
    d2rq:belongsToClassMap      map:Antibiogram ;
    d2rq:property               ddo:susceptibility ;
    d2rq:column                 'Antibiogram.susceptibility' ;
    d2rq:datatype               xsd:string .
```

**Figure 3.5: Data source formalization** - Mapping from the database schema to an ontology. Notice the properties `d2rq:uriPattern`, `d2rq:uriColumn` and `d2rq:join` associating concepts in the database mapping ontology to the actual database tables and columns.

A property bridge class also connects the properties to their respective resources created by a class map. Further, D2R allows classes and properties to be linked to external resources, such as the DDO, using the `d2rq:property` and `d2rq:class` clauses. Then, we can have the data source fully formalized by linking tables and columns from the relational database to the DDO classes and properties.

### 3.3.5 Formal Data Set

The formalization of the data set becomes trivial once the data source and data model have been formalized. The DDO provides already all the formalism to express the data elements and values of a data set in the semantic web language. Then, we only need to instantiate the data model to have a formal data set representation. We can do it simply by executing a construct SPARQL query against the formal endpoint. For example, the antibiogram concept provided in Table 3.1 could be instantiated as shown in Figure 3.6. Even using local terms, the data set can be fully formalized using semantic web ontologies and transformed into a RDF linked data. Figure 3.7 shows the equivalent graphical representation of the concept described Figure 3.6.

```
@prefix xsd:   <http://www.w3.org/2002/07/owl#> .
@prefix ddo:   <http://localhost:8080/ddo#> .

ddo:Antibiogram#3180102  a ddo:Antibiogram ;
        ddo:susceptibility     'S'^^xsd:string ;
        ddo:mic                '0.98'^^xsd:float ;
        ddo:antimicrobial      'cefepime'^^xsd:string ;
        ddo:cultureId          ddo:Culture#1910181 .
```

**Figure 3.6: Turtle representation of a data set** - Example of a data set in the RDF format.

If standard terminologies, such as WHO-ATC and SNOMED CT, are employed in the local data sources, terms could be still formalized using the same approach. For instance, if in the antimicrobial definition of Figure 3.6 we use the equivalent WHO-ATC code `J01DE01` instead of the string `cefepime`. Then, the term would be formalized as `'J01DE01'^^xsd:string`. Further, complex data types could be defined by ontologies in order to represent the semantics of the terminology, such as in

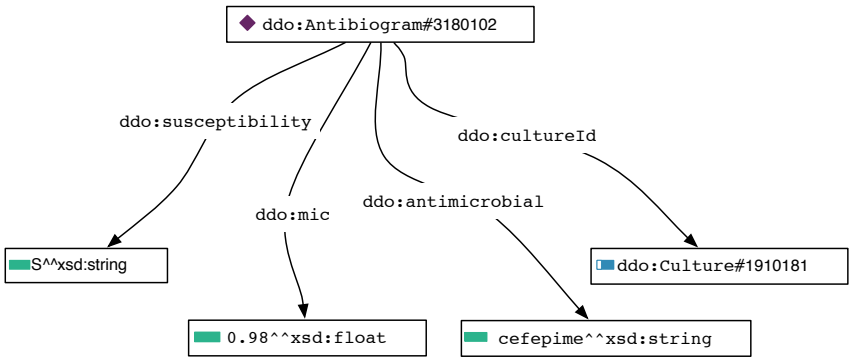


**Figure 3.7: Instantiation of a formal data model** - Visual representation of a RDF graph describing an antibiogram result. The diamond-shape purple node represents an individual, rectangle-shape green nodes represent data type properties, and the rectangle-shape blue node represents an object property.

the *Clinical SKOS Schemes*[1]. Then, the concept *cefepime* could be represented as `'J01DE01'^^clisko:atc20090101`. This representation carries more semantics than the literal `cefepime`, and therefore, leaves less room for ambiguities.

### 3.3.6 Storage Size

Space is a keystone parameter to determine the architecture of a data integration system. The space required by a single database affects especially the scalability and query performance of the distributed system, and therefore, it influences the integration strategy. In order to estimate the size of an antimicrobial resistance surveillance database, we can compute the size of a microbiology report and how many reports are produced in a hospital per year. If we consider the extreme case, where for every patient stay there is a microbiology report, then the incremental yearly size of a database will be the product of the report size $t_{size}$ by the number of stays per year $s$.

The amount of data that are produced by a single plain antimicrobial susceptibility test, that is, without transforming it to one of the normal representations of relational models, can be calculated using the typical report shown in Table 3.1. If we consider an average of 15 antimicrobial tests per culture isolated, a single report would take $t_{size} = 1491$ bytes of space. The number of patient stays per hospital per year $s$ can

[1] `http://www.agfa.com/w3c/2009/clinicalSKOSSchemes#`

be estimated using Equation 3.1

$$s = 365 \times \frac{n \times r}{los}, \tag{3.1}$$

where $n$ is the number of beds, $r$ is the occupancy rate and $los$ is the average length of stay in days.

To estimate the required space of a microbiology database, we can use, for example, the statistics provided by the WHO Regional Office for Europe[1] in 2009. According to the WHO, a European hospital has in average $n = 198$ beds, which are kept occupied $r = 76\%$ during an year, with an average length of stay per patient of $los = 8.17$ days. Substituting these values in Equation 3.1, it results in average $s = 6.7$ thousand patient stays per year per hospital. Multiplying this value by the size of a single report $t_{size}$, we conclude that in a typical healthcare setting only about 10 MB of microbiology data are produced per year. Notice that this value is over-estimated since the normalization of the data model will reduce the space. In addition, it is unlikely that every patient produces a report. Nevertheless, this amount of data (e.g., 100MB in 10 years) does not pose any challenge to current database and storage management systems, especially if we consider the petabytes of data already managed in some scientific experiments [135].

## 3.4 Evaluation

To test the database modeling and formalization approach described in this chapter we perform a preliminary evaluation using some lCDRs. We have two specific objectives. First, we want to test the capability of the lCDR to answer to SPARQL queries and provide RDF results. Second, we want to assess the performance of the remote SPARQL endpoints, using real epidemiological use-cases.

### 3.4.1 Study Context

In collaboration with other partners of the DebugIT project, we have formalized four microbiology databases using data from the following European healthcare institutions – Les Hôpitaux Universitaires de Genève (HUG), Geneva, Switzerland; Georges Pompidou European Hospital (HEGP), Paris, France; Swedish Intensive Care Registry (SIR), Sweden; and Universitätsklinikum Freiburg (UKLFR), Freiburg, Germany. Each data

[1] `http://data.euro.who.int/hfadb/`

source was deployed within its respective institution and provided a SPARQL endpoint, powered by D2R, whose data model was formalized by a DDO (see Table 3.2).

| Site | SPARQL endpoint | DDO |
|---|---|---|
| HEGP | `https://debugit1.spim.jussieu.fr/sparql` | `https://debugit.spim.jussieu.fr/ddo` |
| HUG | `https://babar.unige.ch:8443/cdr/sparql` | `http://babar.unige.ch:8080/vocab/resource/ddo_code` |
| SIR | `https://lincoln.imt.liu.se:8443/d2r-server/sparql` | `https://lincoln.imt.liu.se:8443/vocab/resource/liu_ddo` |
| UKLFR | `https://codeine.averbis.uni-freiburg.de:8443/debugIT/sparql` | `https://codeine.averbis.uni-freiburg.de:8443/vocab/resource/uklfr_ddo` |

**Table 3.2: Formal Microbiology Endpoints** - Formal SPARQL endpoint and respective DDOs.

### 3.4.2 Methods

For each lCDR, we develop a SPARQL query template using the `CONSTRUCT` clause. The queries are designed so that the information about the antimicrobial tested, the antibiogram outcome and the culturing date are present in the result set. Then, real epidemiological use-case questions, such as *"What is the evolution of* :bacteria *resistance to* :antibiotic *during* :period?", can be answered from the resulting graph. Figure 3.8 shows an example of SPARQL query using HUG's DDO. The query matches any *K. pneumoniae* (UniProt/NEWT code 573) antibiogram produced at HUG. If the microbiology database is properly formalized, the query result shall be an RDF graph containing a subset of the endpoint data.

To measure the performance of the lCDRs, we use the response time of queries submitted in serial in a link with bandwidth of 100 Mbps. For each lCDR, we query the endpoints for the antibiograms of the following bacteria: *A. baumannii, E. faecalis, E. faecium, E. coli, K. pneumoniae, N. gonorrhoeae, N. meningitidis, P. aeruginosa, S. aureus* and *S. pneumoniae*. The queries match any antibiotic in order to increase the result set. The results include up to 5 years of data, ranging from 2005-01-01 to 2009-12-31.

```
PREFIX ddo: <http://babar.unige.ch:8080/vocab/resource/ddo_code#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX biosko:
            <http://eulersharp.sourceforge.net/2003/03swap/bioSKOSSchemes#>

CONSTRUCT  {
?antibiogram a ddo:Antibiogram ;
           ddo:hasTestedDrug [ ddo:hasDrugCode ?antimicrobial ] ;
           ddo:hasIdentifiedBacterium [ ddo:hasBacteriumCode ?pathogen ] ;
           ddo:hasOutcome ?outcome ;
           ddo:hasCulture [ ddo:hasResultDate ?date ] .
}
WHERE      {
?antibiogram a ddo:Antibiogram ;
           ddo:hasTestedDrug [ ddo:hasDrugCode ?antimicrobial ] ;
           ddo:hasIdentifiedBacterium [ ddo:hasBacteriumCode ?pathogen ] ;
           ddo:hasOutcome ?outcome ;
           ddo:hasCulture [ ddo:hasResultDate ?date ] .
FILTER     (?date>='2005-01-01T00:00:00'^^xsd:dateTime
           && ?date<='2009-12-31T23:59:59'^^xsd:dateTime)
FILTER     (?pathogen='573'^^biosko:uniProtTaxonomyDT)
}
```

**Figure 3.8: SPARQL Query template** - Example of `CONSTRUCT` query used to test HUG endpoint.

### 3.4.3 Results and Discussion

The four formalized endpoints were able to answer to the SPARQL queries and exchange messages in the RDF protocol. For instance, Figure 3.9 shows an RDF graph as a result of executing the query showed in Figure 3.8 against HUG's endpoint (limited to 1 result). As showed in Table 3.3, in average, the remote query run time was 25.8 seconds. The SPARQL engines of the lCDR were able to execute the queries in $12.6 \pm 23.1s$, responding for 49% of the total run time, while the network was responsible for others 51%.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:ddo="http://babar.unige.ch:8080/vocab/resource/ddo_code#">
  <ddo:Antibiogram rdf:about="https://babar.unige.ch:8443/cdr/resource/
  Antibiogram/726426">
   <ddo:hasOutcome rdf:datatype="http://www.agfa.com/w3c/2009/
   clinicalSKOSSchemes#sct20080731DT">131196009</ddo:hasOutcome>
   <ddo:hasTestedDrug rdf:parseType="Resource">
    <ddo:hasDrugCode rdf:datatype="http://www.agfa.com/w3c/2009/
    clinicalSKOSSchemes#atc20090101DT">J01CR02</ddo:hasDrugCode>
   </ddo:hasTestedDrug>
   <ddo:hasCulture rdf:parseType="Resource">
    <ddo:hasResultDate rdf:datatype="http://www.w3.org/2001/XMLSchema#
    dateTime">2006-08-08T09:07:00</ddo:hasResultDate>
   </ddo:hasCulture>
   <ddo:hasIdentifiedBacterium rdf:parseType="Resource">
    <ddo:hasBacteriumCode rdf:datatype="http://eulersharp.sourceforge.net/
    2003/03swap/bioSKOSSchemes#uniProtTaxonomyDT">573</ddo:hasBacteriumCode>
   </ddo:hasIdentifiedBacterium>
  </ddo:Antibiogram>
</rdf:RDF>
```

**Figure 3.9: SPARQL RDF result** - Result of a SPARQL query against a formalized endpoint. Graph representation in the RDF/XML syntax.

The organization of the data set in the RDF format increases considerably the size of the result set, impacting directly in the network time. The representation of

the concepts as web resources, using fully qualified names, and their formalization, including information such as data type and domain, are responsible for the increase in the size of the data set. For instance, the result set of Figure 3.9 contains only five concepts – antibiogram id: 726426, bacteria: 573, antibiotic tested: J01CR02, susceptibility: 131196009 and culture time: 2006-08-08T09:07:00 – but it sizes to 1047 bytes. On the other hand, the raw five instances (no data type, no domain, etc.) would size to only 48 bytes. The increase in size of the result set is the price paid by the formalization and disambiguation of the data.

The lCDRs provide a common framework to access heterogeneous databases, where data can be fetched using a single query protocol (SPARQL) and results are available in a single and formal data format (RDF). Thus, at this stage, the system provides a common technical platform for querying, and to some extent, a unique syntax to express the results. However, due to the differences in the local data models and data set representation, these endpoints are still not fully semantically interoperable. So far, we do not have either a common model or standard terminologies to represent the local data elements. Nevertheless, this first layer of interoperability is keystone in our integration model. As we will see in the next chapter, we will build upon the lCDRs to developed our transnational monitoring architecture.

| **Endpoint** | $n$ | **Triples/**$s$ | **Run Time** ($s$) | |
|---|---|---|---|---|
| | | | **System** | **Network** |
| HEGP | 10 | $(6.6 \pm 3.5) \times 10^3$ | $31.2 \pm 38.4$ | $29.1 \pm 40.0$ |
| HUG | 10 | $(4.2 \pm 2.8) \times 10^3$ | $2.4 \pm 3.6$ | $1.5 \pm 2.2$ |
| SIR | 10 | $(2.1 \pm 1.5) \times 10^3$ | $3.3 \pm 2.9$ | $0.9 \pm 1.0$ |
| UKLFR | 10 | $(7.6 \pm 3.0) \times 10^3$ | $13.6 \pm 14.5$ | $21.2 \pm 23.3$ |
| All | 40 | $(5.1 \pm 3.4) \times 10^3$ | $12.6 \pm 23.1$ | $13.2 \pm 25.5$ |

**Table 3.3: SPARQL retrieval time** - The total retrieval time is the sum of the system (the lCDR) processing time and the network time. $n$ = number of queries.

## 3.5 Summary

In this chapter, we present issues and solutions involved in modeling and formalizing antimicrobial resistance data and microbiology databases using Semantic Web technologies. We introduced an ontology-driven methodology that formalizes the databases at

three levels – data model, data source and data set. The experiments have shown that the approach is able to homogenize the distinct data sources at the technical level, providing a common query language and a message exchange protocol. Furthermore, the semantic endpoint provided relatively good performance (order of few seconds) using real microbiology databases and clinical questions. However, we observe that about 50% of the query run time is spent in the network. As we will see in the next chapters, the overall query time can be improved by pushing part of the statistical computation to the local data sources to reduce the amount of data retrieved through the network.

# 4

# Online and Transnational Antimicrobial Resistance Monitoring Architecture

## 4.1 Introduction

In this chapter, we introduce an architecture for integrating interinstitutional microbiology databases, featuring real-time access to antimicrobial resistance information and being generic with respect to data sources, in order to support multinational antimicrobial resistance surveillance. In special, we investigate the use of Semantic Web-based architecture in the integration and interoperability of heterogeneous and cross-border databases to support such a framework. The work developed here and in Chapter 5 is an expansion of the paper originally published in [136].

## 4.2 Previous European Antimicrobial Resistance Monitoring and Surveillance Initiatives

Several projects have been implemented to provide monitoring and surveillance of antimicrobial resistance evolution in a European context. WHONET was one of the first initiatives to standardize and aggregate results from laboratories in a cross-country environment [43]. Since 1995, the WHO has been developing the WHONET software, in which participating microbiology laboratories present their tests using a specific

susceptibility testing terminology defined by the WHO.

The most successful European surveillance project is the European Antimicrobial Resistance Surveillance System [137] developed by the European Centre for Disease Prevention and Control. According to the agency, 900 public health laboratories serving over 1400 hospitals in Europe participate in the network, providing results on a yearly basis. To improve data quality, external control is applied to the susceptibility testing methods used by the participating laboratories. The project has recently evolved into the European Antimicrobial Resistance Surveillance Network (EARS-Net) [1].

A few other public initiatives were introduced in parallel. In 1998, the European Society of Biomodulation and Chemotherapy created the European Surveillance of Antibiotic Resistance project [2]. The goal was to establish a representative network of sentinel diagnostic laboratories across Europe to provide antimicrobial resistance monitoring and early detection of new resistant pathogens. In the same year, the Centers for Disease Control and Prevention (USA) launched the International Network for the Study and Prevention of Emerging Antimicrobial Resistance [138] with 79% of participant countries, out of 40, from Europe. The main objective of the project was to serve as an early warning system for emerging resistant pathogens. Finally, in 1999, the Antimicrobial Resistance Information Bank [139] was derived from the WHONET informal network. Results were reported to the WHO and an additional external audit quality control was performed on the data. All of these projects have been discontinued, and some were characterized more as a survey than as a surveillance system.

In contrast to the previous initiatives, The Surveillance Network is a corporate-funded surveillance project [140]. It started in 1992 in the United States and later enrolled European laboratories as well. The data extraction and aggregation processes are done by Focus Technologies Inc. (Herndon, VA, USA), the company responsible for the project. Unfortunately, despite having probably the biggest antimicrobial resistance database worldwide, this network provides no antimicrobial resistance information free to the public.

Over a decade ago, Monnet *et al.* [141] had already described and compared the above European surveillance systems. Since then, no new public transnational surveillance initiatives have been developed [142]. Consequently, most projects in use are

[1] `http://www.ecdc.europa.eu/en/activities/surveillance/EARS-Net/Pages/index.aspx`
[2] `http://www.esbic.de/esbic/ind_esar.htm`

based either on reporting and manual data acquisition or on outdated information technologies, especially concerning data integration and semantics. Furthermore, no cross-country monitoring system that provides online, direct and real-time access to antimicrobial resistance information is available. All the systems implemented so far are dependent on delayed data warehouses, usually compiled yearly, which, among other weaknesses, fail to capture antimicrobial resistance outbreaks [142, 13]. Finally, these systems do not provide easy ways to export data. Participating institutes have to comply with the surveillance system standards, a labor intensive task, especially for newcomer institutions or newly discovered resistance pathogens [13].

In order to improve upon existing monitoring architectures, we have designed the Antimicrobial Resistance Trend Monitoring System (ARTEMIS). ARTEMIS architecture illustrates how Semantic Web technologies can support online monitoring of antimicrobial resistance trends in heterogeneous networks of healthcare institutions. It demonstrates how semantically interoperable endpoints can provide on-demand information on resistance evolution. Furthermore, it describes ways to automate the monitoring process through a state-of-the-art clinical data integration system, which provides mechanisms to adapt to existing electronic health records and laboratory information systems. The architecture was implemented and deployed in a network of healthcare institutions that participated in the DebugIT project.

## 4.3 Methods

### 4.3.1 System Requirements

We design ARTEMIS with the help of other experts from the DebugIT project, who have different backgrounds, including infectiologists, epidemiologists, knowledge engineers and eHealth service providers. Over the course of 2 years, we held weekly meetings with these experts to discuss the status of the tasks involved in the system development [143]. In the process, we reviewed the existing distributed integration and interoperable eHealth systems and European antimicrobial resistance monitoring programs. Thereafter, we elaborated the requirements and designed the system model.

To provide a monitoring system that can be effectively used in the fight against antimicrobial resistance, we derived the following six main requirements based on the published literature and on the expertise of the DebugIT consortium.

- **The System Shall Provide Online Information.** All public European supranational monitoring systems provide resistance information in batch mode–that is, data are collected into batches of laboratory tests and processed periodically, usually on a yearly frequency. While online resistance information is useful on a daily basis at local levels, recent infectious pandemic threats have shown how important this information would be at a multinational level for decision makers. Thus, changing this paradigm to online trends is crucial for antimicrobial resistance surveillance, especially for early warning of emerging resistance trends [142, 13].

- **The system shall provide aggregated information from numerous national sources.** Increasing antibiotic resistance is a worldwide public health concern, and for its effective combat, a successful surveillance system has to offer multinational resistance information [144].

- **The system shall not store data centrally.** Sharing biomedical data raises several ethical concerns [145]. To comply with international standards on sharing biomedical information, increase the trust of data providers and encourage collaboration in the surveillance network, central aggregation must be avoided.

- **The system shall implement a formal and semantic-aware data model.** Most of the available systems do not use formalized biomedical data models, nor computable terminologies and ontologies. As a result, the process of extracting resistance information and data analysis in a heterogeneous environment is done manually or semiautomatically. In addition to the overhead work, the lack of formal conceptualization of the raw laboratory data can have a negative influence on the quality of the data.

- **The system shall be high performing.** To be operatively used by healthcare professionals, whose working environment is recognized to be very time constrained, eHealth systems must provide a fast response time.

- **The system shall provide reliable results.** Automatic extraction of antimicrobial resistance trends from heterogeneous data sources poses several challenges to accurate data analysis, including concept ambiguity and the common denominator, which can degrade the quality of the examination. However, especially if

the system is used by clinicians at the point of care, the accuracy of the results must be equivalent to those obtained by semiautomatic processes, where data cleansing and audit are performed prior to integration and interpretation.

### 4.3.2 System Model

To fulfill the ARTEMIS desiderata, we envisage a system according to the Semantic Web-complying architecture presented in Figure 4.1. The system's semantic interoperability schema (Figure 4.1-a) is based on an ontology-driven data integration approach, where multiple semantically flat local data definition ontologies are mapped to a common domain ontology, the DCO ontology [119]. Semantic mappings at local and global levels align concepts from the local ontologies with the domain knowledge.

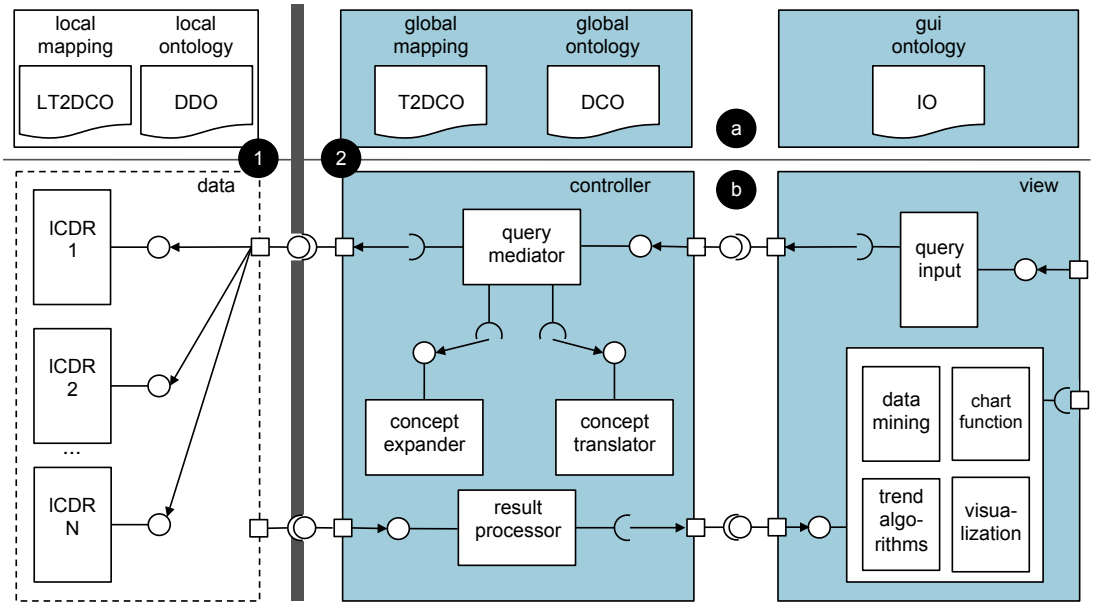


**Figure 4.1: ARTEMIS architecture** - (a) Ontological components. Models: Data Definition Ontology (DDO), DebugIT Core Ontology (DCO) and Interface Ontology (IO). Mappings: local-terminology-to-DCO (LT2DCO) and global-terminology-to-DCO (T2DCO). (b) Run-time business components. (1) Data layer components are deployed within the demilitarized zone of the healthcare institution. (2) Controller and view layers contain central services, which are deployed in the Internet.

In the architecture's data model layer, local laboratory databases are connected online to semantic-aware endpoints, the lCDRs [120, 121], which are further described

in Chapter 3. The lCDRs formalize the local sources and provide a query interface to the controller layer. The semantic mediator, implemented at the controller layer, represents antimicrobial resistance clinical questions as query templates for each endpoint and coordinates the access to the different sites. It performs the query's data aggregation operations locally to improve query performance and the site's data integration on the fly to avoid central storage. Finally, in the view layer, query templates with parameters extracted from the domain ontology are used to represent antimicrobial resistance clinical questions.

The user interface provides methods for users to interact with the system. It implements two main modules: querying input and data visualization. The querying input interface presents a set of clinical question templates with input boxes for the query parameters and the Interface Ontology input menu, which is used to fill in the template parameters. To improve usability and user-friendliness, query templates are expressed in natural language as in the template "What is the prevalence of *:antimicrobial :susceptibility :pathogen* in *:sample* extracted from *:gender* patients at *:clinical_setting* during period *:begin_date* - *:end_date*?". The visualization module provides functions to extract trends, cumulative sum and other statistics from the data retrieved. Ultimately, it implements a set of charts in order to cover comprehensively the interpretation of the data.

### 4.3.3 Participants

To assess ARTEMIS, we connected a network of seven data providers: National Heart Hospital (NHH), Sofia, Bulgaria; HUG; HEGP; Internetový Pristup Ke Zdravotním Informacím Pacienta (IZIP), Prague, Czech Republic; SIR; Athens Chest Hospital "Sotiria" (ACH), Athens, Greece; and UKLFR. Table 4.1 summarizes antimicrobial resistance-related data shared by these institutions.

We obtained permission to use de-identified data from the ethics committees of the respective participant hospitals. Privacy-sensitive information accessible through the local endpoints was pseudoanonymized to conform to the European legal and ethical patient data-sharing framework [146]. Data values such as *date of birth* were truncated to the year, and concepts such as *episode of care* (or *encounter*) and *patient identifiers* were encrypted. Furthermore, query templates are pathogen and population centric– that is, the information collected concerns the resistance and treatment of a pathogen

population for a given antibiotic in a set of microbiology results. It is therefore not related to a specific patient.

| Data | | ACH | HEGP | HUG | IZIP | NHH | SIR | UKLFR |
|---|---|---|---|---|---|---|---|---|
| **Group** | **Element** | | | | | | | |
| Demographics | Age | × | × | × | × | × | × | – |
| | Sex | × | × | × | × | × | × | – |
| Organization | Department | – | × | – | – | – | – | × |
| Laboratory | Bacteria | × | × | × | – | × | × | × |
| | Antibiotic | × | × | × | – | × | × | × |
| | Specimen | × | × | × | – | × | × | × |
| | S.I.R. | × | × | × | – | × | × | × |
| Medication | Drug | × | × | × | × | × | – | – |
| **Triples (M)** | | 0.05 | 25.20 | 19.87 | 2.79 | 0.02 | 3.81 | 19.10 |

**Table 4.1: Data used in ARTEMIS** - × for availability of concepts in the lCDR and – for unavailability. S.I.R. stands for the breakpoint values susceptible (S), intermediate (I) and resistant (R).

### 4.3.4 Outcome Measures

In this chapter, we present the results by describing the implementation of the functional features defined in the first four design requirements introduced in Section 4.3.1 using design patterns [147, 148]. In Chapter 5, we present the evaluation for the last two requirements, which was performed within a larger clinical assessment of the system at HUG.

## 4.4 Results

ARTEMIS was implemented and deployed in a pilot network of seven European healthcare institutions sharing 70+ million triples of antimicrobial resistance information. As shown in Figure 4.2, near real-time resistance trends can be extracted from the distributed network using the system's web interface. The tool can be accessed at `http://babar.unige.ch:8080/artemis`. In the next sections, we present the design patterns describing the main functional features of ARTEMIS.

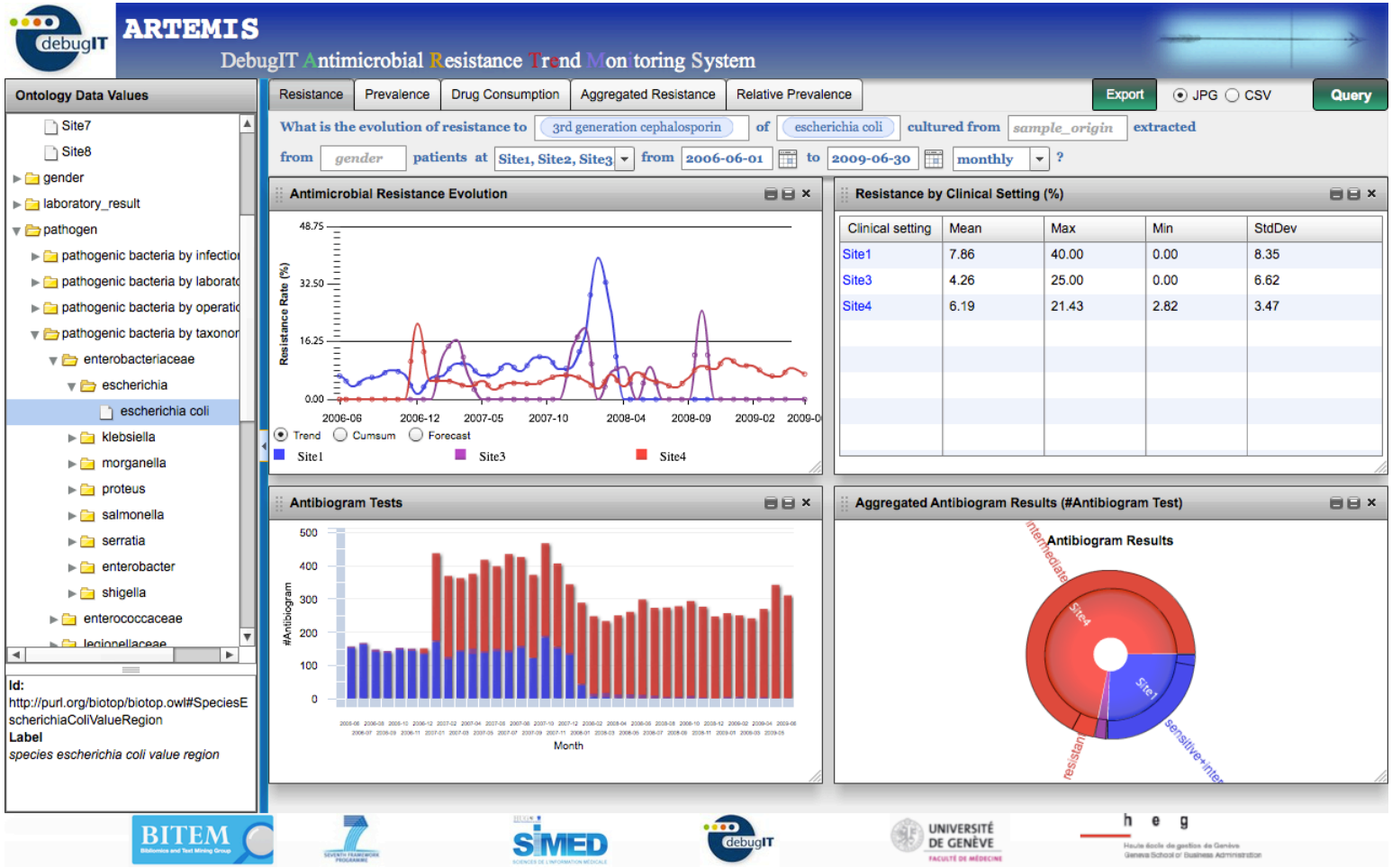


**Figure 4.2: ARTEMIS interface** - The menu on the left displays the interface ontology concepts, which are used to fill in the template parameters. Each of the view tabs represents a different query template. The data visualization interface displays several graphical representations to provide a comprehensive view of the data.

### 4.4.1 Online Information Provider

**Requirement**

The system shall provide online information.

**Design**

In the architecture presented in Figure 4.1, local semantic-aware endpoints, realized by RDF stores, are plugged into the laboratory databases. Thus, microbiology tests are accessible as soon as they are available in the production databases. These endpoints are formalized by local ontologies and exposed in the Web so that data are reachable by other parts of the system. In cases where local laboratory databases communicate in the SPARQL protocol, they can be directly connected to the network. To avoid disclosing patient's sensitive information, data are anonymized either persistently in the local database, as shown in Figure 4.3-a, or on-the-fly, using constraints in the lCDR mapping, as shown in Figure 4.3-b. Hence, no sensitive information is available out of the healthcare intranet.

**Example**

In ARTEMIS, the technical interoperability with the different data sources is provided by D2R [130] engines complemented by site-specific extract, transform and load processes (see Figure 4.3-a), which can exploit autocoding methods [149]. Alternatively, for cases where there is an accessible production laboratory database, D2R can be plugged directly into the existing system to transform the local data source into a semantic endpoint (see Figure 4.3-b). The preference for an RDB-to-RDF engine like D2R instead of a native RDF triple store to formalize local data sources was due to scalability issues. As Schmidt *et al.* [150] noted, native RDF triple stores can hardly be scaled to answer queries when their size is bigger than a few million triples. The anonymization of the data is performed in D2R using database functions, such as string truncation and encryption.

### 4.4.2 Distributed Storage

**Requirement**

The system shall provide aggregated information from numerous international sources.

**Design**

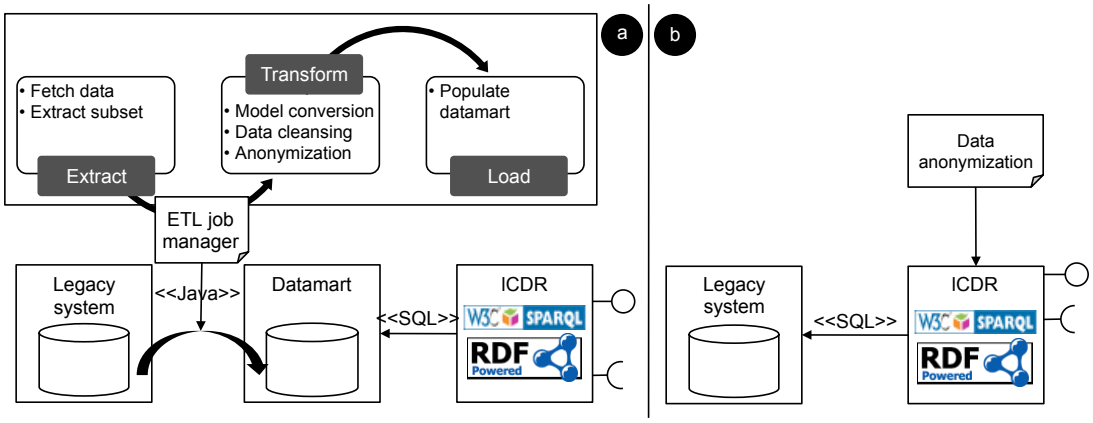


**Figure 4.3: Local CDR deployment and population model** - a) Production data are extracted daily to a local mirror database, which is sparqlized by an RDB-to-RDF engine. b) RDF view is created directly on top of the legacy system. Data anonymization is performed on the fly.

The technical and semantic heterogeneity within models and concepts from different clinical data sources poses an important barrier for data aggregation and analysis. ARTEMIS architecture relies on a layer of semantically formalized endpoints, the lCDRs, to solve part of the integration problem. These endpoints provide a first level of interoperability, modeling the local systems and the data content and providing a common protocol to access data, the SPARQL protocol. The semantic mediator designed in the controller layer builds on top of the lCDR layer and allows the creation of homogeneous aggregated views over the distributed data sources. Thus, the system becomes a grid of semantic-aware sentinels that provide antimicrobial resistance information from heterogeneous supranational data sources.

The query mediator defines, for each lCDR, SPARQL representations of a limited set of antimicrobial resistance clinical questions presented in the view layer. The clinical question SPARQL queries are built as templates, which are parameterized queries using DCO concepts. Assigning values to a clinical question template results in a new SPARQL query. For example, the template "What is the antimicrobial resistance evolution to *:antimicrobial* of *:pathogen* cultured from *:sample_origin* from *:begin_date* to *:end_date*?" might be instantiated as "What is the antimicrobial resistance evolution to *cefepime* of *Escherichia coli* cultured from *blood sample* from *2011-01-01* to *2011-12-31*?". Thus, a template represents an infinite number of queries.

At the query run-time, templates expressed through global concepts are translated into local SPARQL queries with terms from the local ontologies. The query parameters are expanded employing the hierarchical information modeled in the domain ontology and are translated to local terms using the semantic mappings. For example, the DCO concept "3rd generation cephalosporin" shown in Figure 4.4 is expanded to its DCO subclasses, which are further mapped to local DDO terms. In order to optimize network performance and reinforce patient confidentiality, aggregation operations are pushed down to the lCDRs. The SPARQL operators COUNT and GROUP BY are used to perform local result aggregation. Results are fetched respecting the query filter constraints, which perform logical disjunction operations for the expanded parameters. An inverse process is performed on the results retrieved – local terms are translated to global terms, which are aggregated in the root concept, that is,"3rd generation cephalosporin" in the example.

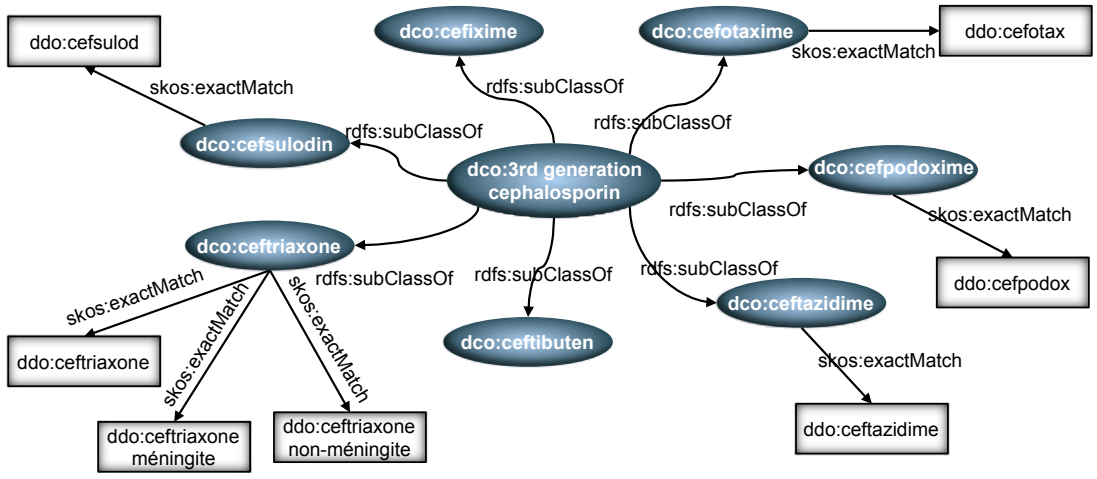


**Figure 4.4: Global-to-local concept translation and query expansion model** - Ontology properties, such as *subClassOf*, and the SKOS semantic mapping *exactMatch* are used for query expansion and translation.

### Example

In ARTEMIS, the lCDRs are provided by RDF-like stores to create the first semantic layer on top of the local databases. The DDOs formalize the local endpoints and expose linkable data in the Web. The JENA Framework[1] is used for querying the remote lCDRs and for reasoning over the RDF models.

[1] `http://jena.apache.org/`

### 4.4.3 Institutional Autonomy

**Requirement**

The system shall not store data centrally.

**Design**

ARTEMIS changes the centralized integration paradigm used in antimicrobial resistance surveillance. Unlike other systems [141, 142, 13], the distributed architecture presented here does not require centralization of microbiology test results. At the query time, a global aggregated view on the local endpoints is created by the semantic mediator, solving the problem of interoperability while avoiding a central repository, which would violates the DebugIT project's legal requirements. Additionally, since there is no need to move data across the healthcare border, this design gives full control to participating sites, allowing them to stop sharing data at any moment. Further, no historical information for the respective site is kept on the system.

**Example**

In the model-view-control pattern [147] presented in Figure 4.1, persistent data stores are deployed only within the demilitarized zone of the data providers. The central mediator processes and aggregates query constraints locally. In this configuration, there is no need to move data sets with information at the patient level out of the institutional borders. Only aggregated population data are retrieved at the query time. Furthermore, institutions can stop sharing data at any moment by shutting down the lCDR server. This change will be automatically reflected in ARTEMIS, which will not be able to retrieve any data from the respective data source – other sources remain seamlessly reachable.

### 4.4.4 Knowledge Representation

**Requirement**

The system shall implement a formal and semantic aware data model.

**Design**

In a multinational environment, content of EHRs and LISs are expressed in several languages and different terminologies. Additionally, spelling mistakes and abbreviations are commonly found in concept definitions. These ambiguities reduce the quality of the statistical analysis. In order to have unified semantics across the different data sources,

in ARTEMIS's knowledge model, antimicrobial resistance concepts are represented using a formal language based on RDF/OWL (see Figure 4.5). Further, they are aligned into common syntaxes defined by biomedical terminologies. Finally, to have a common meaning across the whole system, these formally represented terminologies are mapped to a shared domain ontology.

**Example**

DCO is the core ontology that formalizes the domain knowledge of ARTEMIS. DCO uses the OWL language to represent classes and properties. Currently, it contains 1665 classes that cover the antimicrobial resistance subject. The main clinical areas described by DCO are microbiology laboratories, diagnoses and medication actions. In order to facilitate the interaction of the end-user with the domain ontology, a subset of DCO is used in the ARTEMIS interface. It omits classes that are not relevant to the antimicrobial resistance queries.

Standard terminologies such as SNOMED CT, WHO-ATC and UniProt/NEWT are mapped to DCO using the SKOS ontology and Notation 3 rules (see Figure 4.5-b). If local concepts represented in the DDOs are not already defined using these terminologies, they are normalized against them using automatic classification tools [151, 149]. Alternatively, local concepts represented in the SKOS notation can be directly mapped to DCO. This step is important, as it can easily be adapted to support local needs and evolutions. Finally, DDOs are exposed in the Web so that local concepts can be linked to the domain knowledge.

## 4.5 Discussions

In this chapter, an online and source independent architecture that enables monitoring of multinational microbiology databases was presented. The system was implemented and deployed in a pilot surveillance network distributed across Europe. The architecture is able to interoperate heterogeneous networks via the use of semantic maps that account for local specificity. The data integration is performed on-the-fly using standard endpoints, powered with RDF/SPARQL communication, which are mediated via a central engine. The local endpoints are directly connected to the laboratory databases and as such are able to provide (near) real-time resistance information, while avoiding centralization of the data.

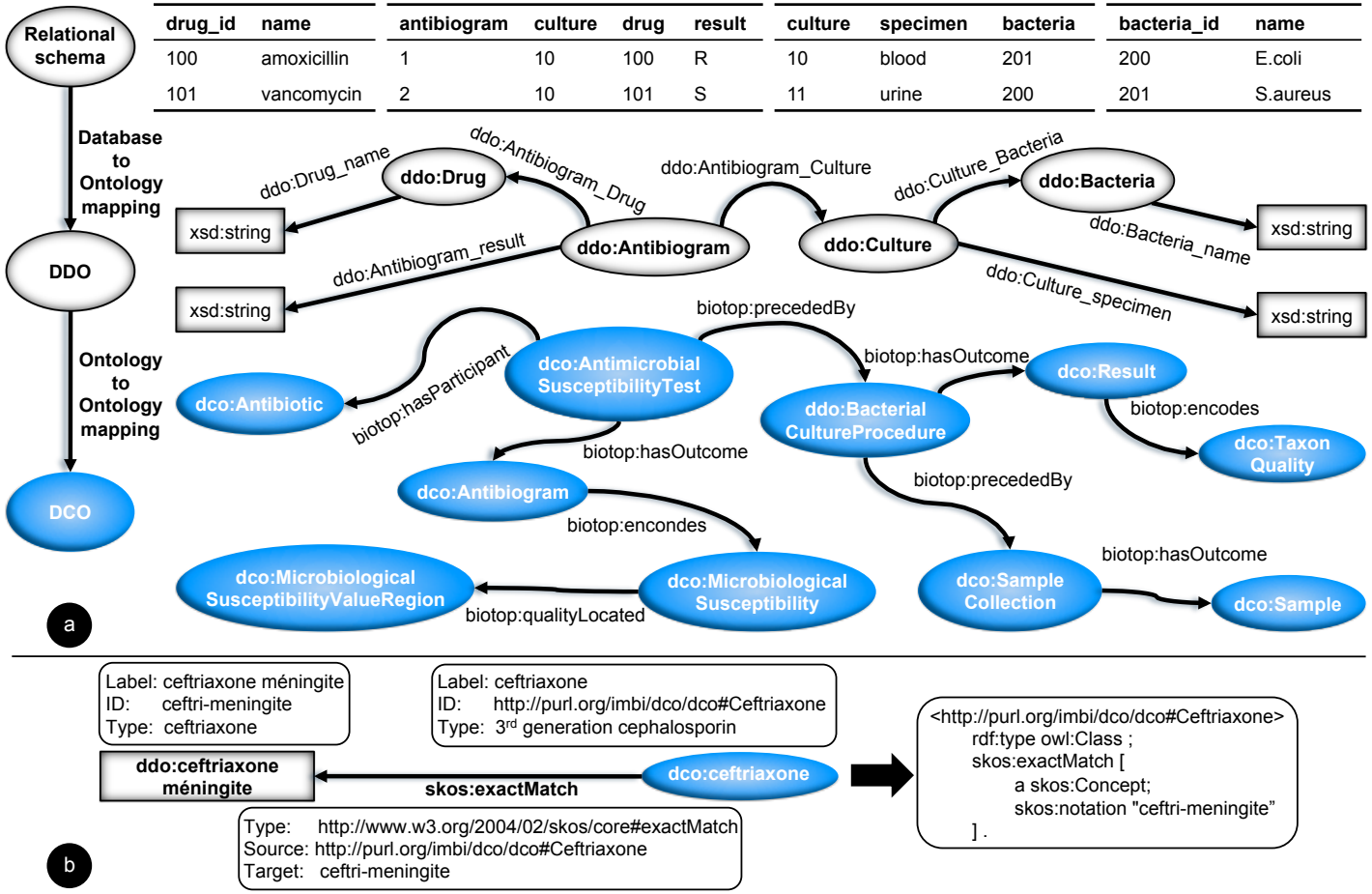


**Figure 4.5: The hybrid ontology-driven interoperability mapping model** - White elements represent local level concepts whereas blue elements represent the shared knowledge. a) Local entity-relationship schemata are formalized by the DDOs. Mappings between DDO data elements and DCO link local concepts to the global knowledge. b) Example of a semantic mapping: concept map diagram (left) and RDF/Notation 3 representation (right).

The data integration architecture proposed in ARTEMIS distinguishes from existing antimicrobial resistance surveillance systems as it implements a loosely coupled data federation design [49], which is realized by the formalization of the data sources and of the data semantics. Thus, the data layer is detached from the central system, which allows the system to operate in a decentralized architecture, guaranteeing then full control over the local information to care providers. Moreover, online semantic data repositories automatize the access to local antimicrobial resistance databases enabling the system to retrieve near real-time antimicrobial resistance trends. Therefore, emerging and outbreak resistances can be easily monitored in a multinational scale. Finally, instead of predetermined and static monitored bacteria/antibiotic pairs, the architecture introduced here facilitates the expansion of the concept coverage, making the process of tracking resistance of new antimicrobials and pathogen trivial. Since concepts are fully formalized by ontologies through the whole architecture, to add a new item to be monitored it is only necessary to create the respective class in the domain ontology and represent it in the semantic mappings (global and/or local). Thus, it is automatically reflected in the user interface, including past occurrences of the given class in microbiology tests.

ARTEMIS uses open Semantic Web technologies to provide technical and semantic interoperability. Semantic data sources create a common technical layer over the local microbiology databases, which can be accessed through a standard query protocol (SPARQL). Since local endpoints are fully formalized and accessible through the Web, they can be linked to external web resources, such as the Linked Life Data [152], or reused in other clinical research projects to leverage knowledge on infectious diseases by combining different sources of information. Another benefit of using ontologies to represent data is the hierarchical structure, which allows higher level representation of concepts. Therefore, the system can handle complex queries expressed at group levels allowing, for example, automatic clustering of antibiotic classes such *3rd generation cephalosporin* or bacteria families such as *Enterobacteriaceae*.

Finally, the powerful query interface allied to the availability of near real-time results make ARTEMIS not only useful to bodies concerned with supranational resistance but also potentially beneficial to local needs, especially if connected to online prescribing systems for empirical treatments. In addition, it might make for the maintenance of the system by healthcare institutions. As it has been discussed in [49], data integration

systems tends to be become "data mortuaries" once the research funds ends. Local appeal can possibly help to change this pattern.

### 4.5.1 Limitations

In an ontology-based integration system, automatic mapping from global to local ontologies using first-order logic reasoners creates logical inconsistencies because knowledge from the various local ontologies cannot be completely reconciled in the global model [153]. For example, if at site 1 vancomycin-resistant *Enterococcus* is prevalent, this fact is not necessary true for all other sites. A solution, as implemented in ARTEMIS, is to create query templates over the local ontologies. However, as the system expands to a large number of clinical providers, this approach may prove difficult to maintain, since query templates must be defined centrally for each new data source. Nevertheless, this limitation could be easily overcome if local sources provide a datamart with a common data model as proposed in Figure 4.3-a.

Aligning multinational microbiology laboratory results presents several issues. For example, it has been shown [138] that, for a given sample test, independent laboratories will present different outcomes. The difference in susceptibility breakpoint across countries is also a complex issue involving standardization of antibiogram methodologies. Additionally, results of second-line antibiotics tend to present bias toward resistance, since they are normally tested when isolates show resistance to first-line drugs [142]. The methodology proposed here cannot solve most of the intrinsic divergence between different laboratory procedures. Regardless, ARTEMIS does not aim to tackle these issues but rather to promote access to distributed antimicrobial resistance information as soon as data are available in a formalized and semantically defined way.

## 4.6 Summary

In this chapter, we present the design of the ARTEMIS architecture. The system was implemented and deployed in a small-scale biosurveillance network of European hospitals, providing real-time access to multinational, heterogeneous microbiology databases. In the next chapter, we provide the results of a larger scale technical and user-based clinical evaluation of the system.

# 5

# Assessment of ARTEMIS

## 5.1 Introduction

In this chapter, we present the results of the clinical evaluation of ARTEMIS. We use two different approaches to assess the system. First, we perform a technical evaluation, where we measure the performance of the semantic mediator and the clinical pertinence of the answers provided by the system. Second, we conduct a user-based evaluation, where we assess the perceived usefulness of the tool to help in the fight against antimicrobial resistance. Our main goal is to find out whether and to what extent our architecture can be used strategically for infection control decisions at organization, country or multinational level and to support decision making at the point of patient care. Our secondary goal is to evaluate the ability of Semantic Web technologies to foster the development of transnational eHealth architectures.

The four specific objectives of the clinical validation of ARTEMIS are the following:

1. To evaluate the performance of the technology in a transnational surveillance network. We want to test whether ARTEMIS can cope with a recognized time-restricted healthcare operational environment.

2. To evaluate the reliability of the tool. We aim to verify whether the framework provide clinical pertinent results and therefore can be trusted for decision support.

3. To evaluate the utility of the technology from the healthcare worker perspective. We want to confirm the perceived value of the tool to support the prevention of antibiotic resistance, if deployed within a healthcare setting.

4. To evaluate the usability of the technology, as implemented in a particular hospital. We aim to check if the implemented version is complete and usable enough to be occasionally productized, promoted, and well adopted.

## 5.2 Methods

### 5.2.1 Theoretical Background

To evaluate ARTEMIS we use the theoretical framework of Nielsen for software engineering [154]. As shown in Figure 5.1, the model presents a hierarchical structure of factors that influence the acceptability of a system by the end user. These factors can be translated as the likelihood that the technology will be adopted by a given institution or type of users. We employ three dimensions of the original model – reliability, utility and usability. Additionally, we extend the model with the dimension responsiveness in order to capture factors related to the response time of the distributed query system.

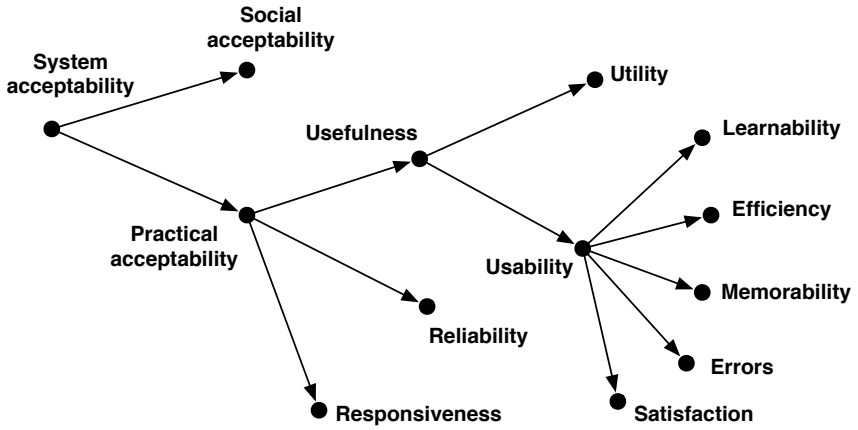


**Figure 5.1: An extended version of Nielsen's hierarchical representation of system acceptability** - We used the original *utility, usability* and *reliability* dimensions to evaluate ARTEMIS. We added the *responsiveness* dimension to the model to represent the performance of the system.

#### 5.2.1.1 Responsiveness

In our model, responsiveness concerns the performance-in-time of the system to execute a given task. We add this dimension to the original Nielsen's model to differentiate be-

tween other time-related dimensions. For example, the original model contains already the dimension efficiency. However, it refers to the capability of a user to accomplish a task efficiently, including then, the time spent in the interface but also the system's processing time. With responsiveness, we are only interested in the system's response time, particularly the performance of the distributed query engine. This dimension is independent of the system's interface, reason why it is not included under efficiency. We measure the responsiveness of the system using the mediator's query retrieval time.

#### 5.2.1.2 Reliability

The dimension reliability defines the level of accuracy, validity and clinical pertinence of the answers provided by ARTEMIS queries. It focuses on the elements that may contribute to the user trust. To assess reliability, resistance rates extracted using the query template "What is the evolution of resistance to *:antimicrobial* of *:pathogen* cultured from *:sample* extracted from *:gender* patients at *:clinical_setting* during period *:begin_date* - *:end_date*?" are compared with data from two publicly available surveillance systems: EARS-Net and the Sentinel Surveillance of Antibiotic Resistance in Switzerland (SEARCH)[1].

#### 5.2.1.3 Utility

In Nielsen's model, utility refers to the capability of a system to provide the functionalities that are potentially useful for the user to accomplish certain tasks. We choose focus groups for the evaluation of the utility dimension of ARTEMIS. The aim of a focus group is to elicit experts' knowledge in a specific domain. This may include experts' factual knowledge and experience, but also ideas, opinions and attitudes. The goal is not consensus building, but to have a wide range of opinions on a given subject. Experts have a broad meta knowledge in their domain and have the capability to foresee the system's impact on the work flow in their medical divisions. In order to gain valid information, questions must be open and leave room for discussion.

[1] `http://www.search.ifik.unibe.ch/en/`

#### 5.2.1.4 Usability

Usability concerns the practical use of the system's features and how users can apply them to accomplish tasks. For instance, a system may provide a specific functionality to support the user in his/her daily work, but this functionality is never used because the user is unaware of its existence. In our case, usability relates to the experience of actual users of the monitoring system and measures how easy and pleasant the functionalities can be used. Together with utility, usability composes one of the underlying dimensions of Nielsen's model, the usefulness of the technology, which refers to the ability of the system to achieve some desired goals. Questionnaires are an established method when it comes to conduct summative usability evaluation of a system. They are standardized and can be applied independent of the system at hand.

### 5.2.2 Participants

We use the seven data sources that joined the ARTEMIS network in the assessment of the system's responsiveness and reliability dimensions. In the user-based evaluation, the system was deployed at HUG. The target group is composed by infectious disease specialists, such as infectiologists, epidemiologists and microbiologists, in charge or not of patients with infectious disease. In the focus group, members of HUG's infectious disease group participated in the discussions. The participant list included experts from different departments of the hospital, so that diverse view points are expressed. The usability evaluation is performed using potential users of the system, including attendants of the focus group. The system was available for the participants of the utility and usability tests during two months and they have used ARTEMIS at least 6 times and for 3 days.

### 5.2.3 Study Flow and Evaluation Criteria

In this section, we detail how the studies are designed and performed for each evaluation dimension. In addition, we describe the outcome measures that are used in the dimensions.

#### 5.2.3.1 Response Time Assessment

We use the three query templates:

**Template 1** What is the evolution of resistance to *:antimicrobial* of *:pathogen* cultured from *:sample* extracted from *:gender* patients at *:clinical_setting* during period *:begin_date* - *:end_date*?

**Template 2** What is the prevalence of *:antimicrobial :susceptibility :pathogen* in *:sample* extracted from *:gender* patients at *:clinical_setting* during period *:begin_date* - *:end_date*?

**Template 3** What is the rate of *:gender* patients that get *:antimicrobial* to treat *:pathogen* infection found in *:sample* at *:clinical_setting* during period *:begin_date* - *:end_date*?

to measure the system's response time. A query mix composed of 225 unique queries, spanning four years in daily, monthly and yearly periods were created. Combinations of pathogens, antibiotics and sample types were employed to vary the queries and thus avoid database caching effects. Each query mix was submitted 10 times against the seven endpoints and the average response time was measured. Results of the local aggregation mode employed in the ARTEMIS' query mediator are compared with a central aggregation strategy (baseline).

#### 5.2.3.2 Comparison with Existing Systems

We assess the clinical pertinence of the antimicrobial resistance rates extracted by ARTEMIS using EARS-Net and SEARCH as reference systems. ARTEMIS data sources that do not contain either more than 1 million triples or data elements to answer the queries (see Table 4.1) were excluded from the analysis, resulting then in four sites: Georges Pompidou European Hospital, Hôpitaux Universitaires de Genève, Swedish Intensive Care Registry and Universitätsklinikum Freiburg. We compare results from Georges Pompidou European Hospital, Swedish Intensive Care Registry and Universitätsklinikum Freiburg with the resistance rates of their respective EARS-Net countries – France, Sweden and Germany – and results from Hôpitaux Universitaires de Genève with SEARCH.

Yearly resistance trends of seven key pathogenic bacteria – *Enterococcus faecalis*, *Enterococcus faecium*, *Escherichia coli*, *Klebsiella pneumoniae*, *Pseudomonas aeruginosa*, *Staphylococcus aureus* and *Streptococcus pneumoniae* – extracted based on their

presence in ARTEMIS, EARS-Net and SEARCH are used in our comparison. Antibiotics are selected if they are present on both ARTE-MIS and the reference system. Resistance rates of the last 4 years (2006 to 2009) available in EARS-Net are used, whereas all years (2008 to 2010) available in SEARCH are taken into account. We report correlation and equivalence results using the Spearman rank correlation and the two one-sided convolution [155, 156] tests, respectively.

#### 5.2.3.3 Focus Group

At the beginning of the focus group, participants are introduced to each other and to the focus group moderator. The discussion is done in an informal atmosphere, in order that the participants discuss freely on the topics. They are informed that the focus group aim to evaluate the utility of ARTEMIS and that the discussion is voice recorded, but their identity are not revealed in the analysis and reports. Then, the following introductory questions are asked in order to evaluate their experience with information technologies to help with antimicrobial resistance control and ARTEMIS:

- *"What computer programs do you currently use to track resistance patterns?"*
- *"Have you all used ARTEMIS?"*

Subsequently, a demo of the main functionalities of the tool is provided using some basic use cases of antimicrobial resistance queries. Then, the moderator introduces the following questions to actually open the discussion on ARTEMIS' utility and hands it over to the participants:

- *"What would the deployment of ARTEMIS change in your service?"*
- *"What kind of problems would you expect when ARTEMIS is used for studies on a population level?"*

When all utility-related questions are discussed, the moderator asks two final questions on the aspects of trust and social acceptability and closes the focus group:

- *"What makes you to trust in the information provided by an antimicrobial resistance monitoring system?"*
- *"Are there any medico-legal concerns for such type of population monitoring system?"*

We assess qualitatively the utility of ARTEMIS at the functional and technical levels using content analysis. At the functional level, we are interested in high level factors that influence the usefulness of the tool, that is, its actual epidemiological and clinical relevance. At the technical level, we are particularly interested in factors of the interface design that might influence on the utility of ARTEMIS. We are guided by the following key questions:

- *"Is there a positive influence of up-to-date knowledge of resistance patterns on infection control?"*
- *"Is there a positive influence of prescribers' access to up-to-date resistance information on appropriate antimicrobial prescription?"*
- *"Does the ontology menu add value to the query construction?"*
- *"Do the charts provide useful information on the subject?"*

Notice that these key questions are not presented directly to the focus group participants.

#### 5.2.3.4 Usability Questionnaire

To assess of the usability dimension, we provide paper-based questionnaires to users during the focus group. We also set up a web form[1] for those who are not present or able to provide the answers during the focus group. The questionnaires contain five items according to the model of Nielsen (see Figure 5.1):

**Learnability** - *ARTEMIS is easy to learn when starting to use it.* This dimension measures how easy it is to accomplish basic tasks when the user interacts with ARTEMIS for the first time.

**Efficiency** - *ARTEMIS is efficient to use when functions are known.* This dimension measures how quickly the user can perform tasks in the interface once he/she has learned the design of ARTEMIS.

**Memorability** - *ARTEMIS' functions are easy to find again.* This dimension measures how easily the user can reestablish proficiency in performing tasks when he/she returns to the interface after a period of not using it.

[1] `https://docs.google.com/spreadsheet/viewform?formkey=dHFMTkdmcEgyamE5U0hGVHdBVFVpLWc6MQ`

**Errors** - *ARTEMIS enables you to make queries with few errors.* This dimension measures how many errors the user makes when using the interface, how severe they are and how easily he/she can recover from these errors.

**Satisfaction** - *The design of ARTEMIS is pleasant.* This dimension measures how the user is satisfied with the interface.

Each item can be ranked according to the Likert 1-7 scale: 1 = *Strongly disagree*, 2 = *Disagree*, 3 = *Somewhat disagree*, 4 = *Neither agree nor disagree*, 5 = *Somewhat agree*, 6 = *Agree* and 7 = *Strongly agree*. The answers to the five questions provide quasi-quantitative measures on the users' perception of ease of use.

### 5.2.4 Methods for Data Acquisition and Analysis

In this section, we provide all the relevant aspects of the data acquisition and analysis methods so that our evaluation experiments can be replicated.

#### 5.2.4.1 System Log

ARTEMIS logs most of the user interaction with the web interface. In special, it registers the views accessed, the queries launched, including their parameters and the response time. We use the query logs and basic statistics to analyze the responsiveness of the system.

#### 5.2.4.2 Equivalence Test

Two one-sided t-test is an equivalence test widely applied in bioequivalence studies [157] but also in model comparison [158]. We use its variation for non-normal distribution, the two one-sided convolution test (TOSC), in the susceptibility equivalence test as part of the reliability assessment. The TOSC test is based on a Wilcoxon test to derive the confidence interval (CI) [159]. The null hypothesis is that resistance rates of ARTEMIS and of the reference surveillance system differs by at least an interval $\Delta$. ARTEMIS results are deemed equivalent to the reference trends at the $\alpha = 0.05$ level if the CI for the difference in resistance rates is completely contained within a region of similarity, delimited by the endpoints $-\Delta$ and $+\Delta$. The susceptibility results' standard deviation of different countries in EARS-Net is used to estimate the region of similarity. Since ARTEMIS rates come from different data samples, this interval is extrapolated as a

level of acceptance of the results. A similar procedure is applied for SEARCH but instead of different countries, the standard deviation among the different regions of Switzerland (East, Mid and West) is used.

#### 5.2.4.3 Content Analysis

Focus groups are relatively easy to conduct but difficult to evaluate. We recur to a method from social sciences called content analysis that belongs to the family of qualitative research methods [160] to analyze the arguments discussed during the focus group . In our experiment, the content analysis is performed according to the flow diagram presented in Figure 5.2. The preliminary classification used in step 3 to classify the text passages is derived from the four main topics of the focus group questions: *Functional, Medico-legal, Technical* and *Trust*.

#### 5.2.4.4 Questionnaires

We collect the anonymous responses to the paper-based and web form questionnaires and apply basic descriptive statistics, such as median and median absolute deviation (MAD), to evaluate the dimension usability and its subdimensions learnability, efficiency, memorability, errors and satisfaction.

## 5.3 Results

### 5.3.1 Responsiveness

Table 5.1 resumes the results of the responsiveness test for the distributed query engine. The mean query response time was $\mu = 4.3s$ and the standard deviation $\sigma = 0.1 \times 10^2 s$. Comparing the local reasoning approach used in ARTEMIS with a different aggregation strategy, based on central reasoning, the average retrieval time increases almost 30 fold ($\mu = 130.5 \pm 0.1 \times 10^3 s$).

Figure 5.3 shows how the response time of ARTEMIS queries varies with the number of rows retrieved for different query templates and aggregation periods. As we can see, the response time is highly correlated with the number of rows retrieved ($\rho = 0.81, P < .001$).

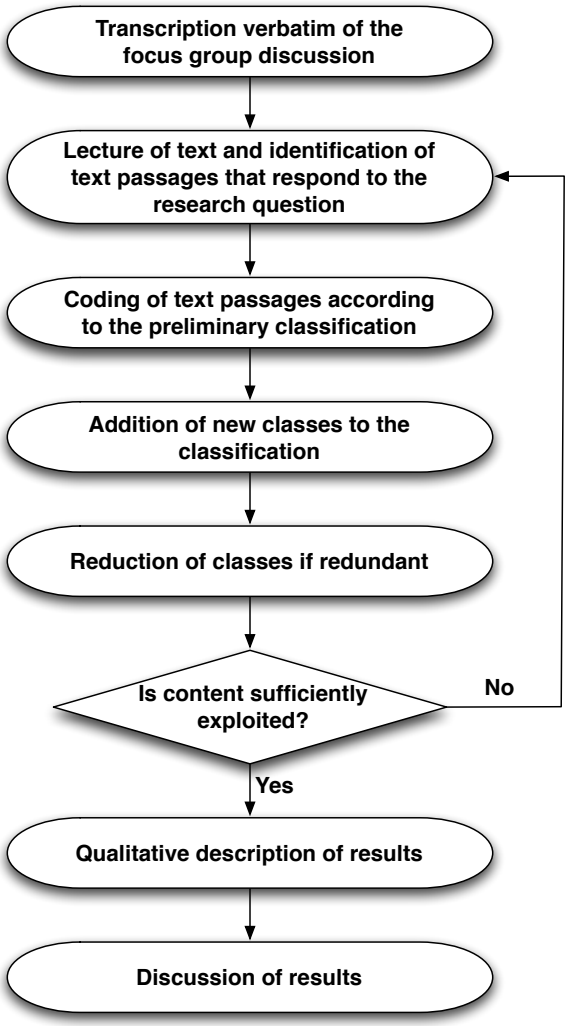


**Figure 5.2: Content analysis flow diagram** - Methodology used to analyze the content of the focus group discussion.

| **Template** | $n$ | **Baseline** | | **ARTEMIS** | |
|---|---|---|---|---|---|
| | | $t_a$ $(s)$ | $t_g$ $(s)$ | $t_a$ $(s)$ | $t_g$ $(s)$ |
| T1 | 75 | $311.0 \pm 0.9 \times 10^3$ | $308.3 \pm 0.1$ | $8.4 \pm 0.1 \times 10^2$ | $4.2 \pm 0.1$ |
| T2 | 75 | $74.7 \pm 0.6 \times 10^2$ | $72.1 \pm 0.1$ | $2.3 \pm 0.6 \times 10$ | $1.3 \pm 0.1$ |
| T3 | 75 | $5.9 \pm 0.8 \times 10$ | $2.7 \pm 0.1$ | $2.0 \pm 0.2 \times 10$ | $1.7 \pm 0.1$ |
| All | 225 | $130.5 \pm 0.1 \times 10^3$ | $39.2 \pm 0.1$ | $4.3 \pm 0.1 \times 10^2$ | $2.1 \pm 0.1$ |

**Table 5.1: Mediator performance -** Arithmetic ($t_a$) and geometric ($t_g$) mean execution times for the two different query mediation strategies: local (ARTEMIS) vs. central (Baseline) reasoning. $n$: number of distinct queries.

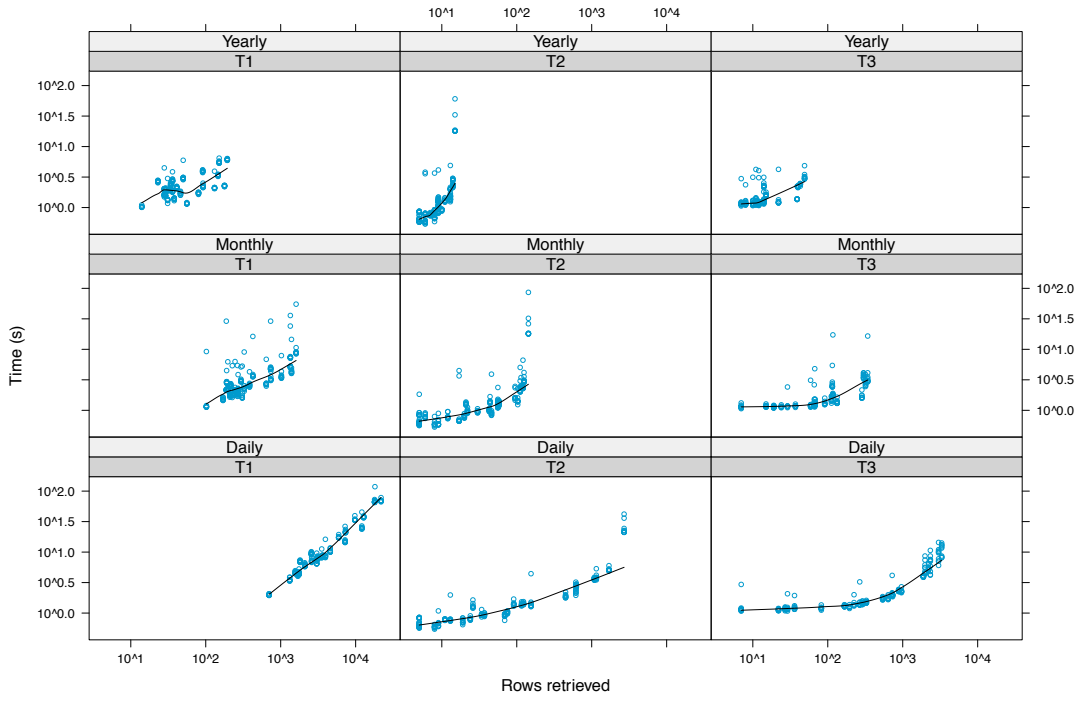


**Figure 5.3: Query performance** - Response time and rows retrieved by template and aggregation period. As expected, the response time tends to increase with the number of rows retrieved.

### 5.3.2 Reliability

Following the data selection criterion, 221 queries for EARS-Net and 153 for SEARCH were created based on template T1. The geometric mean resistance rates extracted from the three systems are displayed in Table 5.2. The results yielded a strong positive correlation coefficient between ARTEMIS and both EARS-Net ($\rho = 0.86, P < .001$) and SEARCH ($\rho = 0.84, P < .001$) reference systems.

| **n** | **Resistance rate** | | | $\rho$ | ***P*-value** |
|---|---|---|---|---|---|
| | **EARS-Net** | **SEARCH** | **ARTEMIS** | | |
| 221 | $0.032 \pm 0.002 \times 10^2$ | NA | $0.038 \pm 0.002 \times 10^2$ | 0.86 | $< 0.001$ |
| 153 | NA | $0.042 \pm 0.001 \times 10^2$ | $0.053 \pm 0.002 \times 10^2$ | 0.84 | $< 0.001$ |

**Table 5.2: Resistance rate geometric mean and correlation results** - $n$: number of queries. $\rho$: Spearman rank correlation coefficient.

The within countries geometric standard deviation of EARS-Net resulted in $\sigma_{ears} = 0.130$. This value was extrapolated to the TOSC similarity region $\Delta$ ($|\Delta| = \sigma_{ears}$). Figure 5.4-a (all results) and 5.4-b (without outliers) present the correlation between the two systems and Figure 5.4-c shows the regions of similarity. As one can see, the confidence interval lies in the region of similarity (95% CI 0 to 0.030; $P < .001$), confirming the equivalence between ARTEMIS and EARS-Net resistance rates. Similarly, for SEARCH, the Swiss region's geometric standard deviation was $\sigma_{search} = 0.042$, indicating a small susceptibility rate variation in the different regions. In this scenario, as Figure 5.5 shows, the results of ARTEMIS cannot be considered equivalent to SEARCH (95% CI 0 to 0.052; $P = 0.18$). However, removing outliers (10 out of 153 data points), that is, those results that fall within a difference in resistance rate bigger than $3\sigma_{search}$, leads also to an equivalent outcome (95% CI -0.004 to 0.028; $P = 0.004$).

### 5.3.3 Utility

The focus group was composed of seven participants (P1 to P7) with mixed backgrounds, involved in epidemiological surveillance but also having a role in clinical care. The participants have already used other software to track resistance patterns, including local applications but also at the national (SEARCH) and European (EARS-Net) levels. The discussion was guided by four main subjects: the application of the tool in

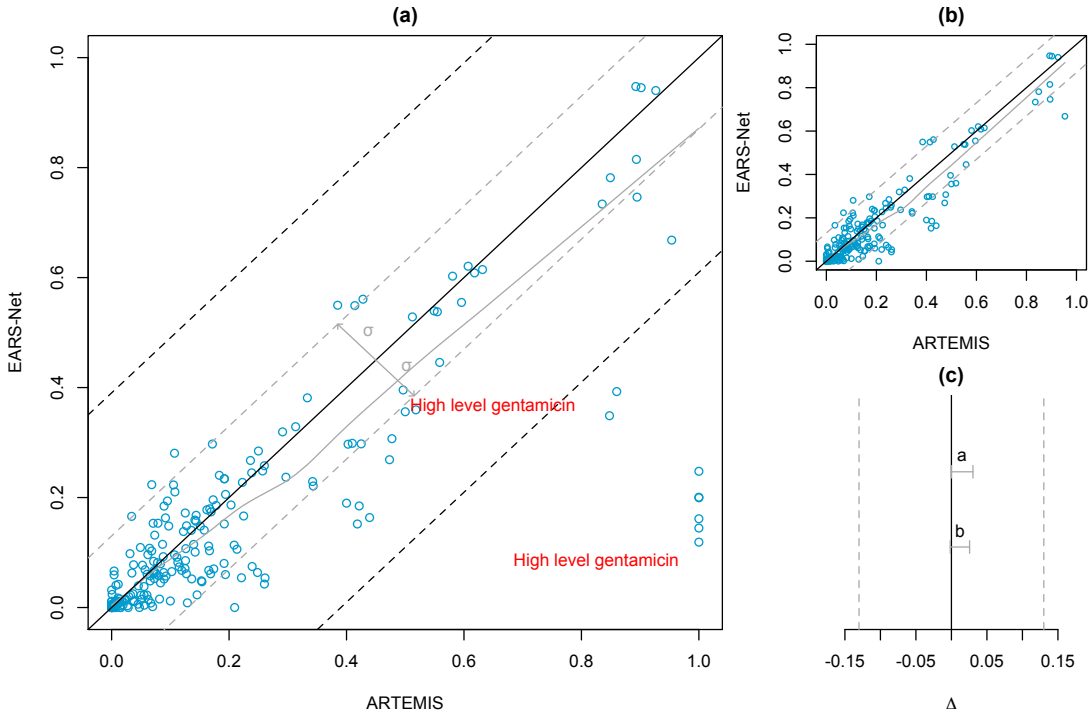


**Figure 5.4: ARTEMIS *vs.* EARS-Net** - a) Resistance rates ($n = 221$). Black line: exact match (100% equivalence). Grey line: best fit. Grey dashed lines: $\Delta = \pm 0.130$. b) Resistance rates without outliers ($n = 213$). c) Grey vertical dashed lines: similarity region $\Delta$. Grey horizontal bars: TOSC confidence interval. 95% $CI_a$ 0 to 0.030 ($P < .001$); 95% $CI_b$ -0.002 to 0.026 ($P < .001$).

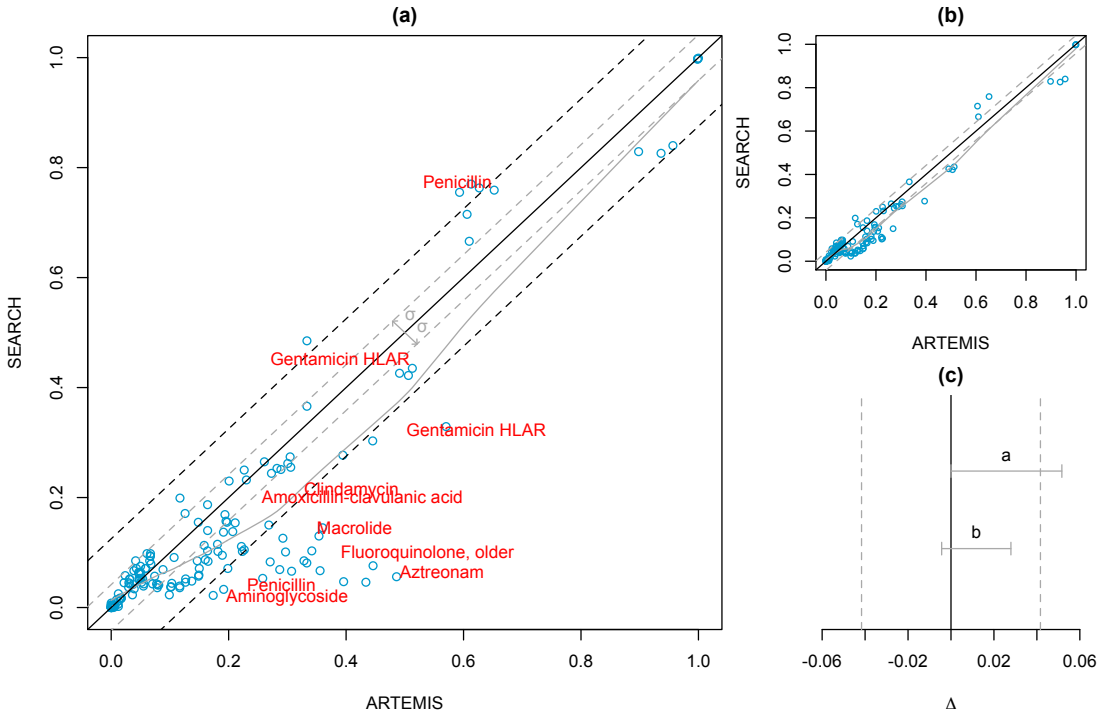


**Figure 5.5: ARTEMIS *vs.* SEARCH** - a) Resistance rates ($n = 153$). Black line: exact match (100% equivalence). Grey line: best fit. Grey dashed lines: $\Delta = \pm 0.042$. b) Resistance rates without outliers ($n = 143$). c) Grey vertical dashed lines: similarity region $\Delta$. Grey horizontal bars: TOSC confidence interval. 95% $\text{CI}_a$ 0 to 0.052 ($P = .17$); 95% $\text{CI}_b$ -0.004 to 0.028 ($P = .004$).

healthcare, the technical components of ARTEMIS, including the quality of the data and the interface design, the factors that influence trust in a decision supporting tool and medico-legal issues involved in data sharing. After the transcription verbatim of the focus group discussion and the identification of text passages that were associated to the research questions, the original classification was further refined as shown in Figure 5.6.

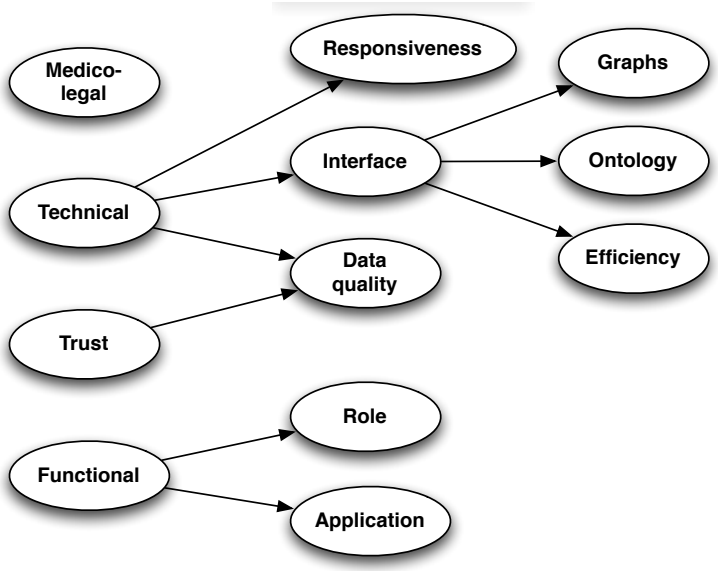


**Figure 5.6: Categories for the content analysis classification** - The four main initial categories – *Functional, Medico-legal, Technical* and *Trust* – were further refined to reflect the focus group discussion.

Overall, participants were positive about the utility of the tool. 6 out of 7 participants mentioned explicitly that the tool is useful or that they see a clear application to healthcare. In their view, provided that the content is reliable, ARTEMIS would serve as a complementary tool to existing frameworks used in epidemiological surveillance. It would be useful to provide a broader view of resistance patterns and evolution. However, more detailed tools that allow looking at the individual patient level would be still required in the investigation and analysis of antimicrobial resistance.

#### 5.3.3.1 Functional Dimension

While answering to the question "What would the deployment of ARTEMIS change in your service?", the participants have identified several application roles for the tool. The roles varied from a day-to-day working tool for epidemiological studies, to a decision support system for empirical treatment and generation of guidelines, and finally, to a research tool, where correlation with other data could be tested. For example, in the passage:

(P1) *For us in infection control, [ARTEMIS would be useful] to follow the trends of resistance, to follow the emergence of new pathogens, new resistance mechanisms. It can be even a sort of an early warning system that we can use. But that's mostly on ecologic basis. Then, this can help us to link it to other data we already have, either antibiotic usage data or data on infection control, like hand hygiene. So, it could be also a sort of research tool where we can do some ecologic correlation analysis, time series analysis, all kind of different things.*

However, despite believing it could be used for empirical treatment, some participants have argued that most of clinicians would not change their treatment based on the up-to-date resistance patterns provided by ARTEMIS. For example, in the following two passages:

(P1) *I don't think that at this point it'll have a direct impact on patient care for many colleagues. It may have a impact first on guidelines. And second, for those who are knowledgeable, specially, for instance, advanced clinicians like in the ICU, or specific units, that they will adjust their empirical treatment pattern.*

(P6) *Let's say we do an aspiration on the respiratory tract and we see there is an enterobacteria among different types of germs and they don't go to the antibiograms. We still want to treat that patient and would like to know what is the profile of resistance of enterobacteria on our wards and that might be helpful [for empirical treatment], for example. That's the only utility [in clinical practice] I find by looking at the patterns. But, in any case, we'll use large spectrum antibiotics for this type of patients. So, I don't think it will change our practice.*

#### 5.3.3.2 Technical Dimension

Surprisingly, the interface ontology has received very negative comments by several participants. In general, the users found it too complicated and not useful. They preferred to try to find the terms directly in the query input box than to search in the ontology tree. For example, in the passages:
(P5) *I don't know the classification used for it. Why don't you specify the most frequent fungi which are responsible in the human infection?*
(P3) *Even the microbiologists don't [know the classification]. But you can go directly in the [query input] field and look for it. Because the left side [of the interface] there is no use. No one knows this. Even the microbiologist don't know it. Even for bacteria because they are really the old family names of bacteria.*
To be useful, the interface ontology should be simplified and adapted to a more operational-oriented nomenclature.

The second component of the user interface discussed was the visualization module. According to the users, most of the views (resistance trends, number of antibiograms, statistics, etc. - see Figure 4.2) were relevant, utile and added value. However, few plots were regarded as presenting redundant information and some modifications were suggested. For instance, to stratify aggregated resistance results into age groups:
(P3) *There is no added information in this chart. If you replace it for example by just separating less than 16 years old and more than 65, and then the third part is between the two.*

#### 5.3.3.3 Trust Dimension

One of the main points raised by the participants during the focus group was the quality of the data. They have identified several inconsistencies in the results and were in general negative about the quality of ARTEMIS content. For example:
(P2) *That's the problem with some of the queries. When you ask hospital wide you can have the real information. But when it comes for special wards you'll not have all the information.*
(P3) *We've always been under 8% [of resistance of pneumococcus to penicillin]. I did a detailed statistics for pneumococcus in 2008 and I'll do it again this year but had not*

*time to bring for this section. But you can see here that if I trust ARTEMIS I have to worry a lot because since 2008 to 2011 it's reaching over 11%.*
This was considered as a dangerous situation, where in the worst case scenario, treatment guidelines would be based on wrong evidence, impacting directly on the quality of care.

#### 5.3.3.4 Medico-legal Dimension

The last topic of the focus group was related to medico-legal issues involved in sharing microbiology data within the ARTEMIS network. In general, the participants did not find medico-legal problems in ARTEMIS data content. However, they were not in favor of a fully open system. As shown in the next two passages, they have identified positive points on sharing data with other hospitals to create benchmarks, but also negative points (in a fully open system), claiming that the pharmaceutical industry could direct their marketing using local information:
(P6) *But, maybe it should be shared with the different hospitals, or university best hospitals, to have a sense of benchmarking. To see how people... the evolution... We know where our resistance's come from. And that might be one way to improve our practice by acknowledging the fact that there might be better for such a level of resistance in Zurich, or we might be better than Lausanne, or another one.*
(P1) *It could also be misused by industry. Because pharmaceutical industry is very interested in this kind of data. Because then, they can tailor their propaganda to our local situation. And they can also say ok, the other competitor drug is now losing ground, they're getting more resistance, so please use the new drug.*

### 5.3.4 Usability

Eight participating specialists in infectious diseases completed the usability questionnaire. As shown in Table 5.3, the median usability score was 6.0 (MAD=1.0), indicating that overall the participants have perceived ARTEMIS, including all its features, as an easy to use and pleasant system. The dimension that received more negative comments was *error* (median 4, MAD=1.0). Indeed, there were some mistakes while executing the queries in the interface or interpreting the results. Based on the comments in the focus group and the query logs, we grouped the errors in the following categories:

- Global semantic errors – The semantics of the interface ontology were not clear and sometimes misleading. For example, users used the class *penicillin*, which includes all penicillin-like antimicrobial agents in our ontology (and in the WHO-ATC terminology), in place of the specific *benzylpenicillin* agent (penicillin G), which is normally referred to simply as penicillin by practitioners.

- Local semantic errors – For some query parameters, due to inadequate maps in the local ontology mapping files, ARTEMIS was not able to capture fully the local semantics. For example, at HUG, most of the ward names have changed over time (at least once in the last decade) and it was not reflected properly in the mapping files. Only the snapshot of the current ward names was included.

- Constraint errors – Users were not fully aware of the contents of the local database, as it is expected in a distributed system, but the interface was not able to constraint the queries based on existing data. For example, the antibiotic reported in HUG tests as representative of the *methicillin* class is *flucloxacillin*. However, several queries were issued using *oxacillin* and the interface would just show empty plots, which confused the users. Indeed, looking at the query logs, 10% of the queries issued at HUG did not return any result.

The dimension satisfaction has also received some critical comments. For example, to improve some plots (or remove the redundancy of information), but also to simplify the interface ontology to fit more with the operational needs. Nevertheless, this dimension was still well evaluated (median 6, MAD=0.0).

| **Dimension** | **Median** | **Mode** | **Range** | **Inter-quartile Range** | **Median Absolute Deviation** |
|---|---|---|---|---|---|
| Learnability | 6.0 | 6 | 2 | 0.50 | 0.5 |
| Efficiency | 6.0 | 6 | 2 | 0.25 | 0.0 |
| Memorability | 5.5 | 5 | 2 | 1.25 | 0.5 |
| Errors | 4.0 | 3 | 3 | 2.25 | 1.0 |
| Satisfaction | 6.0 | 6 | 3 | 0.25 | 0.0 |
| All | 6.0 | 6 | 4 | 1.00 | 1.0 |

**Table 5.3: Usability descriptive statistics** - *All* is the combination of the five usability dimensions assessed.

Figure 5.7 shows the distribution of answers per usability dimension and the overall score of the interface. As we can see, the overall perception of a system easy to use and pleasant is clearly positive.

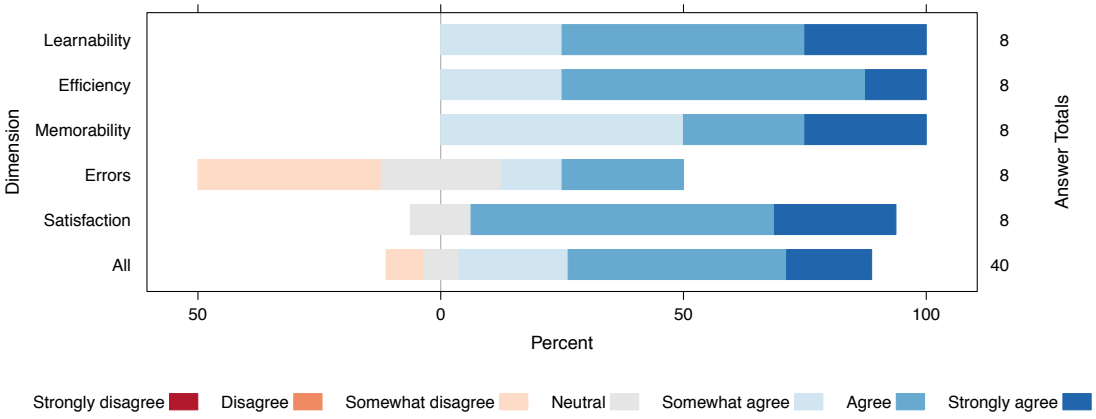


**Figure 5.7: Perceived usability** - Perceived usability results. *All* is the combination of the five usability dimensions assessed.

## 5.4 Discussion

### 5.4.1 Responsiveness

All SPARQL performance benchmarks presented in the literature are focused on local single-source servers [161]. Thus, they are not adequate to assess the performance of data integration systems. Hence, the ARTEMIS semantic mediator was compared with a standard approach of retrieving and aggregating centrally. As Table 5.1 shows, the push-down procedure has reduced the retrieval time by 30-fold (19-fold considering the geometric mean). Indeed, as shown in Figure 5.3, response time is nearly linearly correlated ($\rho = .81, P < .001$) with the amount of data retrieved in a distributed system. Thus, local reasoning is crucial for systems that require fast response time.

At the mediation level, the use of a push-down approach while performing aggregation has proved its efficiency. The average query response was in the order of a few seconds ($\mu = 4.3$, $\sigma = 0.1 \times 10^2$ seconds), which could contribute to the adoption of the system by healthcare workers, who consider a good response time as an important requirement in the system design [162]. This result was confirmed by the usability test,

where users were satisfied with the efficiency of the querying process as a whole (median 6 and MAD=0.5). In addition, during the focus group they have explicitly mentioned positively the performance of the system:
(P3) *Also, the time where it's extracting the data is much faster [than the other tool currently used in the hospital].*

### 5.4.2 Reliability

Existing surveillance systems normally use semiautomatic methods to extract antimicrobial resistance rates [163]. Validation and cleansing steps are taken by experts before statistical analysis. In ARTEMIS, this process is fully automated and, as such, errors can be introduced. To validate ARTEMIS content, we compared antimicrobial the resistance rates with European and national reference systems. The results indicated a strong positive correlation between the antimicrobial susceptibility test outcomes. We carried out a second evaluation based on equivalence tests to confirm the trustworthiness of the results. The tests showed that at the limit of $3\sigma$ ARTEMIS trends are deemed equivalent to both EARS-Net and SEARCH.

Some differences in concept definition in ARTEMIS and in the reference systems affected negatively the results. The majority of outliers (18 out of 33) presented in Figure 5.4-a and Figure 5.5-a were caused by semantic ambiguities between concepts. For example, the gentamicin definition, which is not related to concentration in ARTEMIS, is defined as *Gentamicin HLAR* in SEARCH and *High level gentamicin* in EARS-Net. This issue was not accentuated in the comparison with EARS-Net because, as expected, the region of similarity was wider than that of SEARCH, which considers only within-country variations. Adoption of standard and formalized terminologies in the eHealth care field and a more dynamic evolution of terminological resources so that they can cover operational needs are part of the semantic solution.

These findings were confirmed during the utility assessment. In the focus group, the users have mentioned that the resistance rates of the major pathogens hospital wide, that is, what was actually assessed in the multinational reliability test, were in general convergent with local gold standard data. However, for some specific cases of sample types and local wards, the system provided inconsistent results. In order to fix these local inconsistencies, the participants recommended to cross-check with existing local studies in order to gain trust in the system. We have proceed accordingly in an

interactive validation phase were the ontology was modified, local mappings validated and results re-checked.

### 5.4.3 Utility

In general, the infectious disease specialists that participated in the focus group found ARTEMIS utile as a complementary tool for epidemiological surveillance, providing easy and broad view on resistance patterns. On the other hand, despite the potential of ARTEMIS to support appropriate antimicrobial therapy during prescription, in practice, they believe it would not be effective as a decision support tool because clinicians would stick to standard procedures. It would be more utile in providing evidence to guidelines generation and infection control benchmarks of intra- or inter-institutional services and departments. Resistance patterns could be applied, for instance, as further evidence of hand hygiene compliances within the different groups of the institution. Hence, the main role of the tool would be on epidemiological surveillance analysis rather than directly on clinical care.

Further, the ARTEMIS ontology was designed as a general knowledge model to align heterogeneous data sources and to attend the needs of infectious disease and public health specialists in different countries and institutions. As such, it was based on international terminological resources, like the WHO-ATC, UniProt/NEWT and SNOMED CT. This was important to align successfully local terms to a global information model, against which queries could be issued. However, when we assessed the utility of a subset of this ontology using infectious diseases specialists from a specific healthcare institution, it was not seen as importan. We identify two situations there. First, a global ontology is essential to align information from multinational microbiology databases, harmonizing the semantics across the system and providing access to the distributed data. However, this core ontology should not be used as the system information interface. Local users are likely to stick to their original nomenclature and ways of organizing the data. These results were confirmed in a recent work by Li *et al.* [164]. Therefore, we believe that our model should be changed to provide local flexibility also at the user side. That is, the interface ontology should be completely independent of the domain ontology, instead of a subset, as in our case.

### 5.4.4 Usability

The usability evaluation showed that overall the users perceived the ARTEMIS interface as easy to use and pleasant. To some extent, they were able to perform real antimicrobial resistance queries and interpret the results without exhaustive system training. Additionally to the suggestions provided by the users during the focus group, such as simplification of the ontology and improvement of some plots to reduce redundant information, we believe that the system could profit greatly from a concise help, explaining some fundaments of the query interface, ontology and charts.

### 5.4.5 Batch *vs.* Real-Time Antimicrobial Resistance Monitoring

The monitoring system introduced here advances the state-of-the-art in surveilling evolution of antimicrobial resistance by providing an architecture that aggregates and delivers resistance information as soon as it is available in the local databases of the surveillance network data sources. Currently, even the most advanced antimicrobial resistance biosurveillance systems, like the European-wide EARS-Net or the national-wide SEARCH, are based on delayed, yearly-processed antibiograms. Participants of their surveillance network provide data that are aggregated and processed in yearly-batches before being available to bodies concerned with antimicrobial resistance surveillance. This operation mode is no longer suitable giving the alarming and fast increasing resistance rates for most of the pathogens. Conversely, our system provides real-time access to resistance data by accessing directly and in real-time the local databases where the data are originally generated. As a consequence, it allows outbreak and emerging resistant phenotypes to be spot readily as they are selected, being thus more effective for antimicrobial resistance surveillance.

### 5.4.6 Limitations

This study has a few potential limitations. The sample size of the focus group and of the usability evaluation was relatively small ($n = 7$ and $n = 8$ respectively) for a transnational surveillance system. Further, all participants came from a single healthcare institution. Nevertheless, the samples were composed by a mix of infectious disease specialists, including infectiologists, microbiologists and experienced clinicians, with wide experience in antimicrobial resistance surveillance and in public health. Moreover, the

findings across the different evaluation methods were overlapping and complementary, indicating that the main issues were identified, that is, data saturation among these user-based and the technical evaluation was achieved. Finally, our strict adherence to established methods of qualitative research increases potential confidence in considering the utility results.

## 5.5 Conclusion

We have performed a comprehensive evaluation of ARTEMIS, where four dimensions of the tool were assessed – responsiveness, reliability, utility and usability. Results indicate that the distributed monitoring architecture introduced in Chapter 3 and Chapter 4 can potentially be used to build transnational antimicrobial resistance surveillance networks. The architecture showed efficient and reliable, while complying with local legal and regulatory frameworks. The Semantic Web-based approach of ARTEMIS proved to be an effective solution for development of eHealth architectures that enable online antimicrobial resistance monitoring from heterogeneous data sources. In the future, we plan to investigate local mediation models, paving the way to a more easily maintainable system.

# Part II

# Data Analysis

# 6

# Review on Machine Learning Forecasting

## 6.1 Introduction

Artificial intelligence is the branch of computer science that studies and designs computer systems that present some intelligent behavior [165]. According to John McCarthy, a pioneer in the field, it is *"the science and engineering of making intelligent machines, especially intelligent computer programs"* [1]. Artificial intelligence studies the human cognitive process and ways to simulate and improve it using machine's larger computing and storage powers. A basic element of artificial intelligence is learning. In intelligent systems, learning algorithms provide methods for storing, updating and inferring knowledge from data examples.

Machine learning [166] is the subfield of artificial intelligence that designs algorithms that allow computers to learn behaviors or patterns from large, complex and noisy example data sets. In this context, learning is regarded as inductive inference, by which machines process and memorize examples that describe a particular phenomenon in order to execute a certain task. Machine learning algorithms differ from standard algorithms in their capacity to improve performance and effectiveness depending on the learning method (the student), the quality of the example data (quality of the teacher) and the amount of data (amount of teaching time) available. Learning algorithms are based on several knowledge fields such as logics, statistics, cognitive sci-

[1] http://www-formal.stanford.edu/jmc/whatisai/whatisai.html

ences (psychology, neurology, etc.), and human and animal biological process modeling. While cognitive and biological models help to understand the process of learning and executing intelligent tasks, logics, probability and statistics provide the mathematical foundations so that computers can process information and infer knowledge from the model learnt. Thus, despite relying on statistics, machine learning algorithms are theoretically more powerful since they can employ also logics, conditionality and other process optimization strategies to improve modeling.

In 1958, when the first ideas of artificial intelligence were created, A. Turing had already identified learning as a requirement for intelligent systems [167]. Since then, machine learning has been deployed successfully in many fields to help in decision making and knowledge discovering. Applications include, for example, weather and climate analyzes [168, 169], drug discovery [170], gene selection and cancer classification [171], brain computer interfacing [172, 173], elementary particle searching in high-energy physics [174], stock market forecast [175], fraud detection [176], text categorization [177], handwriting recognition [178], image recognition [179] and many others in science, engineering and medicine, where intricate problems need to be solved and there exists sufficient volume of example data.

## 6.2 Machine Learning in Healthcare

Machine learning has been often applied to solve medical problems [180, 181, 182, 183, 184, 185]. Hospitals have been collecting and storing large amount of data with the increasingly deployment of information systems, such as electronic health records and computerized physician order entry, as part of their normal operational workflow. Add to that the daily production of genomic, proteomic, and diagnostic and imaging data in an scale never seen before in research and clinical environments. Paradoxically, while for humans this huge amount of information gathered makes it each time more difficult to analyze and visualize the big picture, for computers, due to their much larger storage capacity and processing power, it actually improves their capability of learning and providing correct answers the intricate problems. Thus, as in other fields, in healthcare more and more computer systems are being used to assist humans in complex and specific tasks that need to deal with data intensive environments, containing sizable and diverse data sets.

In the literature, there are substantial machine learning works applied to the analysis of clinical conditions and their influence in small specialized problems [186, 187]. In a review, Cruz and Wishart [184] found that from 1994 to 2005 the number of published papers applying machine learning algorithms for cancer prediction and prognosis grew exponentially. These studies showed that machine learning methods could improve risk assessment and outcome prediction up to 25% when compared to classical statistical methods. Syeda-Mahmood *et al.* [188] applied a method for non-rigid alignment of electrocardiogram shapes in the diagnosis of heart diseases. Their image learning algorithm identified the similarity between shapes in different electrocardiogram readouts and helped in recognizing some types of heart disease by the characteristic shape of waves produced by an electrocardiogram. In [189], Visweswaran *et al.* developed an algorithm based on the Markov blanket and Bayesian models for learning patient-specific outcome applied to the classification of sepsis and hearth failures outcomes. For small samples, the patient-specific learning method outperformed population-centric models. Finally, Ramoni *et al.* [190] used a robust version of the naive Bayes classifier to predict mortality in intensive care. Their robust algorithm improved the classification accuracy when the training sample contained missing data, however at the cost of coverage.

In the infectious diseases field, machine learning algorithms have been applied in predictive scenarios to assist healthcare workers and officers with outbreak alerts, generation of guidelines and antimicrobial prescription [191, 192, 14]. In [193], a hidden Markov model was deployed in the characterization of outbreak of resistant pathogens. The authors developed an epidemic model using 157 weeks of vancomycin-resistant enterococci prevalence data to quantify cross-transmission and sporadic colonization of the strain. The system successfully estimated the transmission rate of 89% while genotyping methods varied between 84% and 90%. Gierl *et al.* [194] developed a case-based reasoning using a hierarchical categorization tree for antibiotics prescription decision support based on previously documented clinical cases. Despite having a high approval rate among the physicians, due to divergent opinions amongst the experts evaluating the system, it failed to develop a gold standard advisor. Leibovici *et al.* [195] developed TREAT, a decision support system for aiding antibiotic prescription for the treatment of inpatients with bacteria infections. TREAT used Bayesian networks (causal probabilistic networks) fed by up-to-date clinical and laboratory data, and antibiotic cost-benefit models to create an intelligent antibiotic advisor. The system was evaluated in

a randomized controlled trial in three hospitals from different countries. Treatments supported by the system's recommendations improved the percentage of appropriate antibiotic usage and reduced the mean duration of hospital stay.

## 6.3 Machine Learning Design

The goal of a learning algorithm is to create a model that, when receiving new unknown inputs, will be able to associate them to example data, whether the example data are kept in their original form or transformed into more abstracts formalisms. The design of a machine learning system can be divided into a two-fold process. First, a pre-processing step is required to prepare the data set to fit the machine learning's input format. It involves the collection, formatting and selection of the most appropriate features that will be used to model the system. The pre-processing phase is estimated to account for around 80% of the designing work [196]. It is a data dependent task and despite of not being an end per se, it is an indispensable task in the modeling of a learning algorithm. In our work, this task is partially performed by the data integration system described in the first part of the thesis. The second step is the actual design of the learning algorithm. The most appropriate algorithm will depend on the system that is being modeled. The algorithm is conditioned on the type of data available (symbolic, numeric, continuous, discrete, labeled, unlabeled, etc.), the type of task (classification or regression), the performance and accuracy required for the machine, and other constraints. Theoretically, it is very difficult to define the best algorithm to use in a given problem. It is often determined experimentally using validation data sets [197]. This phase is developed in the second part of the thesis. As shown in Figure 6.1, in practice, in a machine learning system there is also a third phase, which is the post-processing of the information inferred by the machine for decision support, knowledge discovery or to feed other intelligent systems.

Independent of the task in which the machine learning system is involved, the algorithm needs a data set to be used as *training* examples. In a typical problem, this training data have a set of features, such as weight, height and time, and an outcome measurement associated to them. The features as well as the outcome can assume numeric or symbolic values. Symbolic and discrete outcomes are found in classification tasks while continuous numeric values are the output of regression tasks.

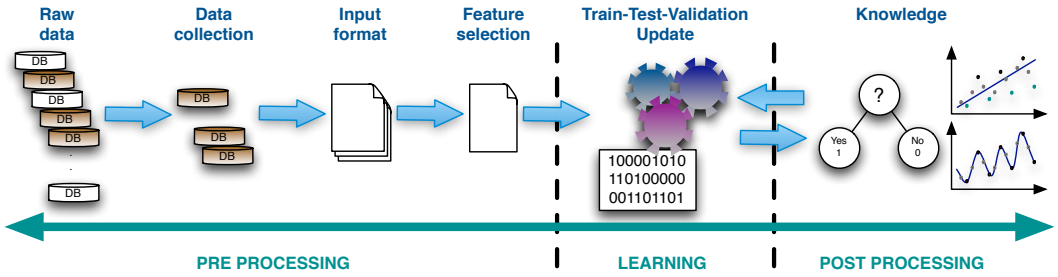


**Figure 6.1: Machine learning workflow** - Preprocessing. Learning-Inference. Knowledge

### 6.3.1 Supervised and Unsupervised Learning

Supervised learning is the most used approach in machine learning and has been applied successfully in many real world applications [198]. Supervised algorithms are trained using examples of existing classified data. A supervised learning problem consists in learning a model $P(X)$ based on a set of examples or instances $D = \{x_i, f(x_i)\}_{i=1}^{N}$, where $x_i$ is the input and $f(x_i)$ is the respective output training examples. The ensemble of examples $D$ is called the *training* set. A learning algorithm, also called *inducer*, will induce a function $f$ using the training set as basis for the model. The function $f$ should generalize the model since the training set $D$ represents only an observable part of the real model $P(X)$. Supervised learning normally requires large amount of labeled data for the system to learn the model $P(X)$ and many times these data are not available or are very expense to obtain. In contrast, unsupervised learning uses existing data without any classification. It tries to create generic classes in the sample data using algorithms that determine how existing data should be labeled. This model is applied when the classes are not known in advance. A classical example of unsupervised learning is clustering, where data are grouped based on a distance function implemented by the clustering algorithm. In this thesis, we deal specifically with supervised learning, since we have the labels for our training set, the outcome of the resistance rates themselves.

### 6.3.2 Classification and Regression Algorithms

Machine learning tasks can be broadly divided into two categories: classification and regression (see Figure 6.2). Algorithms that classify the inputs into a finite set of discrete classes are called *classifiers* and the respective learning task *classification* [199,

200]. For example, patent offices, such as the World Intellectual Property Organization (WIPO), classify medical, biotech and other inventions according to the International Patent Classification (IPC) system. Patent officers manually assign IPC classes to each patent document. Using the classified corpus as training data, this task can be automatized to a certain level via automatic text classifiers. The aim is to design a classification system that, based on existing classified documents, will assign the most appropriate IPC class to a new unseen document. This is also a supervised learning process, since there exists examples of the correct answer. Classifiers are usually evaluated in terms of precision, that is the number of correct answers amongst the set answers provided, and recall, which is the number of correct answers retrieved amongst the set of correct answers. Naïve Bayes [201] is an example of a baseline classification algorithm.

In cases where the output values of the learning algorithm are continuous, the task is called regression [30]. The aim of the learning algorithm is to find the best model that will fit to the data, taking into account current observations but also unobserved data. For example, in stock markets, financial analysts want to model share price time series to forecast how a given price will behave in the future. Regression is usually evaluated in terms of the error (absolute, mean squared, etc.) between the observed output and the forecasted value. Depending on the algorithm and the system modeled, the input data can take both continuous and discrete values, whereas the output is a continuous value. Sometimes the output might need to be restricted (or normalized) to a given range, for example between 0 and 1, in order to comply with the regression algorithm. Random walk [202] is the standard machine learning baseline algorithm for the evaluation of forecasting problems. In this thesis we focus on regression algorithms since they are the only ones that can be used in time series forecasting.

### 6.3.3 Generalization and Specification

One of the challenges of machine learning systems is to find the right balance between generalization and specification. Generality is a property of the model that measures how well the learning algorithm will classify instances that are not part of the training set. Conversely, specificity refers to the property of finding the most specific function that will include all positive and none of the negative training examples [203]. The failure to find the best fit between these two properties can lead to two known problems in

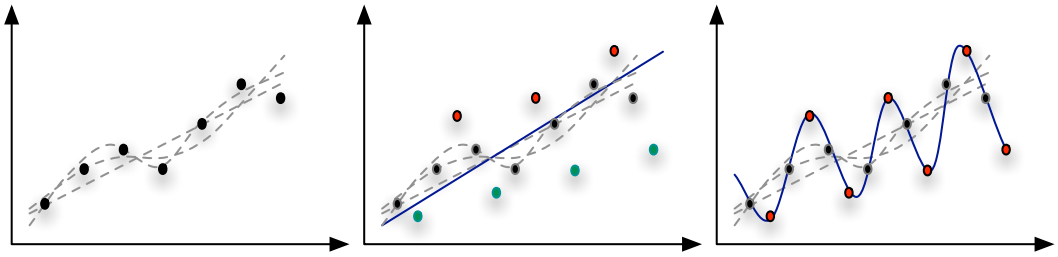

**Figure 6.2: Classification *vs.* regression** - *Left* - black points represent the observed data and the dashed grey lines show the estimated models; *Center* - classification task finds the best curve (or surface) that splits the different classes; *Right* - regression task finds the best curve that fits the data, taking into account unobserved values. Red and green points: unobserved data.

machine learning – underfitting and overfitting. As shown in Figure 6.3, in underfitted systems the accuracy of an algorithm is below its learning capability. It is usually a result of lack of training and poorly estimated parameters. Differently, in overfitted systems the model is too specific to the training data that when new unseen examples are tested, it will no longer be able to generalize and recognize the example as part of the learnt classes.

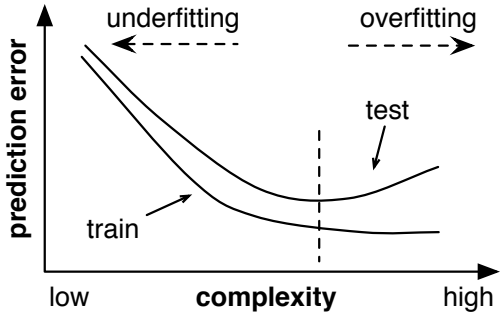


**Figure 6.3: Generalization and specialization** - As the model complexity increases, after a certain turning point, the actual performance on unseen data starts to decrease.

## 6.4 Time Series Forecasting

Time series analysis has been extensively exploited in the literature [22, 25]. Accurate time series forecasting has always been the ultimate goal in the study of time varying

processes. Given a time series, the aim of a forecasting algorithm is to predict the outcome of the system using observed patterns embedded into the time series so that the predicted value is as close as possible of the real future value.

More recently, with the increase in memory and computing power, several data driven forecasting methods based on machine learning have been proposed in the literature. Machine learning algorithms have been successfully applied in the analysis of time series data, including trend detection, outliers and forecasting problems [204, 30]. Without doubts, forecasting is the most popular application used extensively in the financial sector, economics but also in medicine and biology.

In time series forecasting, the data set is represented by chronologically ordered events data defines the state of a given variable. In machine learning regression, time series are decomposed into small sequences, or chunks, which are associated to an outcome. Figure 6.4 shows an example of basic a representation, where a chunk is associated with a 1-step ahead outcome. These sequences compose the supervised training set and are used by the learning method to learn the behavior of the system. Depending of the algorithm more complex representations may be used [205, 206, 207]. It is essential though that they are designed so that the properties of the dynamic system represented by the time series are captured.

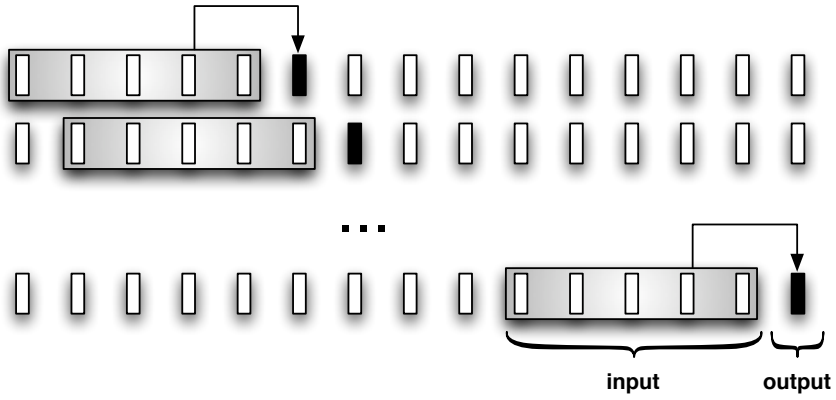


**Figure 6.4: Example of time series representation in machine learning** - The data sequence is decomposed into smaller chunks, which are associated to an output to compose the supervised training set.

The machine learning field provides several algorithms for time series regression and forecasting. In the following sections, we provide a brief overview of classical

statistical approaches and list the machine learning methods that we will investigate on our antimicrobial resistance forecasting algorithm.

### 6.4.1 Classical Statistics

Autoregressive integrated moving average (ARIMA) [23] is the classical model for time series analysis in statistics. ARIMA models assume that the system can be modeled using a linear combination of parameters, that is, $x(t) = f(a_0, a_1, \ldots, a_n, t) + \varepsilon_t$, where $a_i$ is the vector of parameters to be discovered and $\varepsilon_t$ is the error associated [23, 24]. They regard the time series as the realization of a stochastic process [208, 25], which can be represented using a linear combination of autoregressive, integrative and moving average terms. Mathematically, it can be written as

$$\phi(B)(1-B)^d x_t = \theta(B)e_t, \tag{6.1}$$

where $B$ is the backward shift operator, such that $Bx_t = x_{t-1}$, $e_t$ is a purely random process with zero mean and variance $\sigma^2$, $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$ is a polynomial in $B$ of order $p$, $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q$ is a polynomial in $B$ of order $q$, and $\phi(B)(1-B)^d$ is the combined derivative and autoregressive operator [24].

In [209, 210], the authors explored the relationship between antimicrobial consumption and resistance using microbiology susceptibility testing together with pharmacy data. ARIMA models were deployed to create antimicrobial resistance forecasting models, which were employed to quantify empirically the impact of resistance prevalence and antimicrobial use on future resistance rates. Abeku *et al.* [211] applied an ARIMA model to forecast the monthly incidence of malaria using historical morbidity patterns. Despite the very good fit produced in the training data, the model was outperformed by a simple seasonal adjustment model in the out-of-sample forecast.

Exponential smoothing is another branch of linear models that use simple recursive updating formulae to produce time series forecasts [24]. Regardless of the model simplicity, it is able to capture the dynamics of processes with changes in the local level, trend and seasonality. The simplest exponential model can be represented in the form

$$\hat{x}_{N+1} = \alpha x_N + \alpha(1-\alpha)x_{N-1} + \alpha(1-\alpha)^2 x_{N-2} + \ldots, \tag{6.2}$$

or recurrently as

$$\hat{x}_{N+1} = \alpha x_N + (1-\alpha)\hat{x}_N, \tag{6.3}$$

where $\alpha$ is a smoothing parameter in the range (0,1), $\hat{x}_{N+1}$ is the forecast value at time point $t = N + 1$ and $\hat{x}_N$ is the previous forecast.

Exponential smoothing techniques have also been applied to forecast biosurveillance data. Ngo *et al.* used exponential smoothing forecasts to detect outbreaks of gentamicin resistant *Pseudomonas aeruginosa*. The 95% confidence interval upper envelop of the forecasts was used to define the limit between an outbreak and an endemic prevalence. If resistance rates were not higher than the upper limit, the hypothesis of an epidemic was rejected. In [212], the authors introduced a robust prediction method based on Holt-Winters [27, 26] exponential smoothing technique to forecast biosurveillance data and used the predictive results in a control-chart alerting algorithm to detect outbreaks. In their study, the Holt-Winters-based algorithm outperformed the traditional adaptive regression models used for syndromic surveillance.

Theses models are overall effective to fit and forecast several time series processes. However, they assume a linear dependency between the model's parameters, which is not necessarily true for many systems. As we will see, the dynamics of short-term resistance trends are not linear and these approaches are a priori not optimal to model such processes. Conversely, most of the machine learning methods do not assume any dependency of the systems parameters. Most important, the model is defined intrinsically by the data, or more precisely, by the training set. Hence, machine learning models are able to generalize to wider range of systems independently of their dynamics.

### 6.4.2 Least Squares Regression

The least squares estimator [213, 214] is a linear mathematical optimization procedure used in statistics and machine learning. It searches for the best data set fitting by minimizing the residual error obtained from the sum of the squares of the difference between the estimated values and actual observed data. It is the most used approach to approximate solutions in overdetermined equation systems, that is, systems where there are more equations than unknowns. The least squares estimator minimizes the sum of the squares of the regression residuals in order to maximize the fitting to the model.

The least squares algorithm used in time series learning can be defined as follows. Consider a data stream $\mathbf{x}$ being updated at every time-tick. Supposing that $\mathbf{x}$ can be

estimated at instant $t = N + 1$ as a linear combination of past values of $\mathbf{x}$ within a window of size $w$, that is

$$x_{N+1} = \varphi_0 + x_N\varphi_1 + x_{N-1}\varphi_2 + \ldots + x_{N-w}\varphi_w + \varepsilon_N, \tag{6.4}$$

where $\varepsilon_N \sim N(0; \sigma^2)$ is an error term. This model can be conveniently represented in the matrix form as

$$\mathbf{y} = \mathbf{X\Phi} + \epsilon, \tag{6.5}$$

where $\mathbf{y} = (x_{N+1}, \ldots, x_{w+1})^T$, $\mathbf{X} = (\mathbf{1}, \mathbf{x_N}, \ldots, \mathbf{x_{N-w}})$, $\mathbf{\Phi} = (\varphi_0, \varphi_1, \ldots, \varphi_w)^T$ and $\epsilon = (\varepsilon_N, \ldots, \varepsilon_w)^T$. Then, the *least squares* method is based on finding an approximate solution for the coefficients $\mathbf{\Phi}$ in Equation 6.5 that minimizes the sum of squared errors between the real value of $\mathbf{y}$ and the estimate $\hat{\mathbf{y}}$. Formally, it is represented as

$$\min ||\epsilon||_2^2 = \min_{\Phi \in \Re^n} ||\mathbf{y} - \mathbf{X\Phi}||_2^2. \tag{6.6}$$

Solving Equation 6.6 with respect to $\mathbf{\Phi}$, that is, $\partial\epsilon/\partial\Phi = 0$ results in

$$\mathbf{\Phi} = (\mathbf{X^T X})^{-1}\mathbf{X^T y}. \tag{6.7}$$

Then, the machine learning algorithm consists in finding the weights $\mathbf{\Phi}$ that solves Equation 6.7 but also the window size $w$ that optimizes the forecasting.

### 6.4.3 k-Nearest Neighbors

k-NN is one the simplest, most intuitive but also most effective machine learning algorithms [215, 216]. The algorithm implements a instance-based learning, which delays the induction process until the test phase. The term $k$ refers to the number of neighbors that are needed to describe a class in the training set. The algorithm is based on the assumption that, when represented in a $d$-dimensional space, instances from the same class tend to be naturally aggregated. For example, all fishes within a shoal are likely to be from the same species or people from the same city district have higher probability to be from the same social class than people coming from another district.

The distance between instances can be calculated using different metrics, including Euclidian, Manhattan or any other algorithm that is able to quantify differences between instances in the training and testing sets. The $k$ value calculated in the learning phase is used in the test phase to determine the class (or value) of a new unknown

instance. The algorithm performs a search for the $k$ nearest neighbors in the training data and infers the class of the test instance using the set of neighbors, as shown in Figure 6.5. The standard k-NN regression algorithm gives equal weights to all neighbors. In this case, the outcome can be estimated as

$$\hat{y} = \frac{1}{k} \sum_{y_i \in \mathcal{U}(x)} y_i \tag{6.8}$$

where $y_i$ is the target output of training data point $x_i$ and $\mathcal{U}(x)$ is the neighborhood of the test point $x$. Training the number of neighbors optimally is crucial in this model. Too large values for $k$ results in underrepresented classes to be considered as part of larger neighborhoods whereas too small values leads to noise instances to be regarded as actual classes [217].

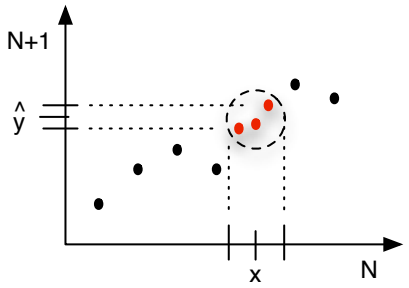


**Figure 6.5: k-NN regression** - The test point $x$ is projected into the result (future) axis using the nearest neighbors.

### 6.4.4 Decision Trees

Decision trees are widely used in applications that requires learning features. It is especially applied in the medical context due to its ability of generating human readable rules [186, 218]. Decision tree is an intuitive class of regression algorithm. As shown in Figure 6.6, it consists in a tree form graph with a set of internal decision nodes and terminal leaves. Each internal node represents a decision space whereas the leaves correspond to the output of the system. The algorithm solves the regression problem by recursively partitioning the input space using a set of decision questions or rules (`if then else`), which splits the learning sample into smaller homogenous parts at each node. In binary trees, a question with binary answer (0/1, yes/no, left/right, etc.) is

asked in each node of the tree. For example, a tree that models resistance rate would ask "Is rate at time $t$ greater than 50%?" and a respective rule would be created to provide the answer, such as `if` $x_t > 50$ `then` *node_left* `else` *node_right*.

In time series forecasting, the decision algorithm will walk through the tree created during the training phase to produce the prediction. Given a new test point $x$, the algorithm will perform a test along the decision nodes starting from the root node until it reaches a leaf node, which will correspond to the prediction. Decision trees are easy to create and interpret, and still produce effective results. As with other algorithms, they computational complexity increase with the number of classes in the training set. However, even for large data sets they tend to perform relatively fast. One of the major drawbacks of the method is its variability with the training space. A small modification in the training set can have significant impact on the tree rules. This high variability phenomenon affects essentially the capability of finding the optimal data fitting. If not tuned properly decision trees are likely to overfit [219].

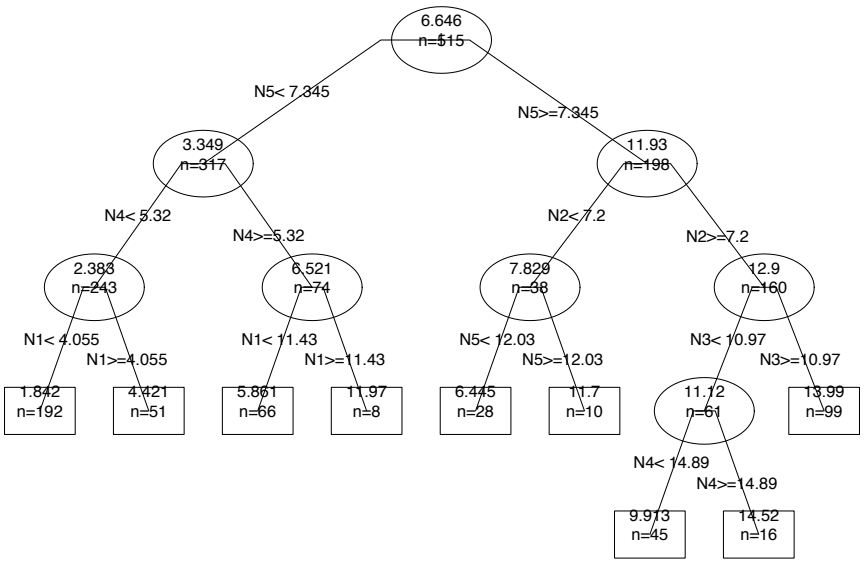


**Figure 6.6: Decision tree regression** - Example of a tree trained using the space formed by observed values at time $t = \{t_{N-1}(N1), t_{N-2}(N2), \ldots, t_{N-5}(N5)\}$. The algorithm decides which leaf will match a given input $x$ following the tree decision nodes.

### 6.4.5 Artificial Neural Networks

Artificial neural networks (ANN), or simply neural networks, are probably the most popular machine learning algorithms in the literature [220, 221]. They have been applied to several regression and classification problems and are one of the responsible for the popularization of machine learning techniques as problem-solvers. Neural networks comprise a class of connectionist algorithms inspired by the behavior of the human brain. As shown in Figure 6.7, a neural network consists of a set of interconnected units, so called *neurons*, that executes small processing tasks based on pre-determined transfer function. The connection between the units are weighted to reflect the influence of one unit upon the other. In a neural network, the units can be grouped into three classes: *i*) input, where the information is fed into the system and the processing starts; *ii*) hidden, which provides further optional processing power to the network; and *iii*) output, where the final results are produced.

The multilayer perceptron is a classical neural network architecture, which contains at least one hidden layer. The output of each unit in the architecture is defined by

$$y_j = g\left(\sum_{i=1}^{N}(w_{ij}x_i + b_j)\right) \tag{6.9}$$

where $x_i$ is the $i$th input vector, $w_{ij}$ denotes the weight of the $i$th input connection of the $j$th node, $b_j$ is the bias and $y_j$ is the $j$th network output node. The function $g$ represents the node's activation function. A classical approach is to use the logistic sigmoid function $g(u) = 1/(1 + \exp(-u))$ to activate the nodes.

While setting the number of hidden layers in a neural network architecture has proved to be trivial, the optimal size of the hidden layer still remains challenging. It is known that a three-layer network with sigmoidal units in the hidden layer suffice to approximate large class mappings. However, setting the number of neurons in the hidden layer too low will lead to poor specification of the classes whereas too many neurons will result in an very complex (computationally and logically) system, which tends to overfit in an out-of-sample test but also to increase considerably the training time [222].

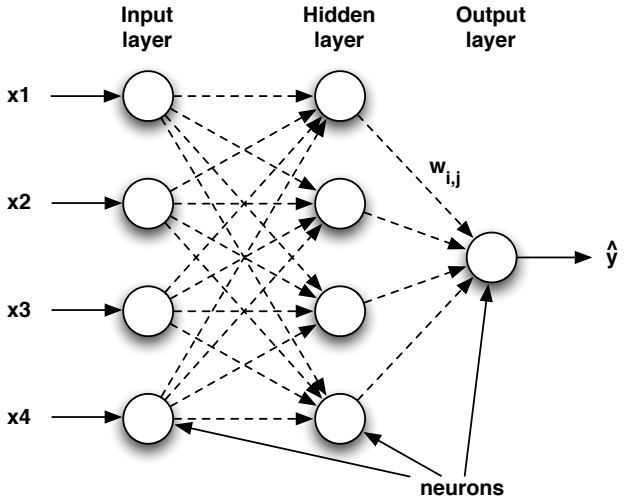


**Figure 6.7: Multilayer perceptron example** - Structure of a feed-forward neural network with four input neurons, one hidden layer and one output.

### 6.4.6 Support Vector Machines

Support vector machine (SVM) algorithms have been originally designed to solve classification tasks applied to optical character recognition [223]. In support vector machine classification [224], hyperplanes in a $d$-dimensional space are constructed to split optimally elements of different classes. The algorithm maximizes the distance of the hyperplanes to the class data points in order to minimize the generalization error. The algorithm works by mapping the original input space $\mathcal{X}$ into a higher dimensional space $\mathcal{F}$, using a function $\Phi : \mathcal{X} \rightarrow \mathcal{F}$, where $\Phi$ is called the *kernel* function.

Later on, based on the same principles, support vector machines have been applied to regression problems [225], what is usually referred as support vector regression in the literature. In this case, given a set of points $\{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathcal{X} \times \Re$, the goal is to find a function $f(x)$ that has at most $\varepsilon$ deviation from the actual target $\mathcal{F}$ for all the training data, and is as flat[1] as possible. That is, the algorithm ignores data points that deviates more than $\varepsilon$ from the hyperplane created by the kernel function.

Let us consider the linear case, as shown in Figure 6.8. In linear kernels, the goal

[1]Flatness is a concept associated to curvature and smoothness of the surface or curve defined by $f(x)$.

is to find a function

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b, \tag{6.10}$$

where $w \in \Re^d$ is the weight vector, $b \in \Re$ is the bias and $\mathbf{x}$ is the input vector. In the case of Equation 6.10, *flatness* refers to small value for $w$. Let $x_i$ and $y_i$ denote respectively the $i$th training input vector and target output, with $i = 1, \ldots, N$. It can be shown that the error function that ensures flatness and that the deviation of the target $\mathbf{y}$ is of less than $\varepsilon$ is given by

$$\min \left( \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \right),$$

$$\text{subject to} \begin{cases} y_i - \mathbf{w} \cdot \mathbf{x_i} - b \leq \varepsilon + \xi_i, \\ \mathbf{w} \cdot \mathbf{x_i} + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0. \end{cases} \tag{6.11}$$

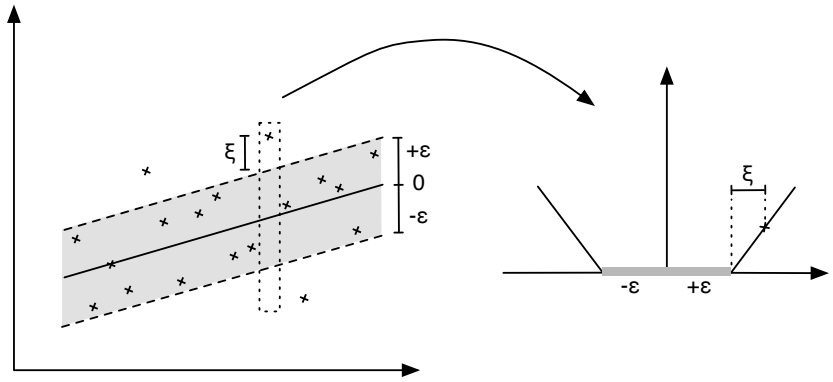


**Figure 6.8: Support vector regression example** - Setting the soft margin loss for a linear support vector machine. Only the points outside the shaded region contribute to the loss function [225].

The first term of Equation 6.11 penalizes model complexity whereas the second term penalizes errors above $\varepsilon$, allowing some room for the parameters to move to reduce model complexity. The constant $C > 0$ determines the trade-off between the flatness of $f(x)$ and the amount up to which deviations larger than $\varepsilon$ are accepted. This corresponds to dealing with a so called $\varepsilon$-insensitive loss function $|\xi|_\varepsilon := \max\{0, |\xi| - \varepsilon\}$ [225, 30]. It can be shown that the solution that minimizes the error function in

Equation 6.11 is given by

$$f(x) = \sum_{i=1}^{N} (\alpha + \alpha^*) \mathbf{x_i} \cdot \mathbf{x} + b, \tag{6.12}$$

where $\alpha$ and $\alpha^*$ are the non-negative Lagrange multipliers associated with the constraints of the problem. Training vectors giving non-zero Lagrange multipliers are called *support vectors* and are the only ones that contribute directly to the solution of the support vector machine algorithm. Equation 6.12 can be easily extended to a nonlinear problem simple by applying a nonlinear kernel $\mathcal{K}$ to the linear term $\mathbf{x_i} \cdot \mathbf{x}$, that is,

$$f(x) = \sum_{i=1}^{N} (\alpha + \alpha^*) \mathcal{K}(\mathbf{x_i} \cdot \mathbf{x}) + b. \tag{6.13}$$

## 6.5 Model Comparison

In the previous section, we have presented some of the main methods used for time series forecasting. As we are interested to investigate the use of machine learning approaches to forecast resistance rates, we have focused more on learning oriented methods than on classical statistics. The literature provides still many other machine learning approaches, such as the Gaussian Processes [226] method, and hundreds of variations within the methods presented. However, we believe that the review presented here is enough to investigate the problems involved in our task.

In Table 6.1, we summarize some features of the methods aforementioned. We consider the basic version of these methodologies to rate their characteristics. The least squares regression model is a priori not suitable to our problem due to its inability to model nonlinear process. Moreover, ANN and SVM, despite of their higher prediction accuracy in large data sets, they do not perform well in data sets with small number of data points. As data of resistance rate time series are unlikely to be available in long periods (several decades), these models are probably not going to be as accurate as they would normally be in other large data sets. Finally, decision tree and k-NN have similar characteristics, with the difference that the latter tends to have a higher predictive power in regression tasks. k-NN is also suitable to forecast resistance rates due to the easiness of building models using this algorithm. One needs to determine

only the parameter $k$ to fully specify a k-NN model. This is of particular importance in antimicrobial resistance analysis since different pathogen/antimicrobial time series will require different models. Despite the theoretical differences amongst the machine learning algorithms presented here, as we have mentioned previously, in machine learning modeling it is extremely difficult to decide a priori which method will perform best in a given task. Thus, in Chapter 7 we evaluate these algorithms to determine which is more appropriate for our learning-based antimicrobial resistance forecasting task.

| **Characteristic** | **Least squares** | **k-NN** | **Decision Tree** | **ANN** | **SVM** |
|---|---|---|---|---|---|
| Model construction | +++ | +++ | +++ | + | ++ |
| Parameter tuning | +++ | +++ | +++ | + | ++ |
| Predictive power | + | ++ | + | +++ | +++ |
| Interpretability | +++ | ++ | +++ | + | + |
| Computation in prediction | +++ | + | +++ | + | + |
| Handling of missing values | + | +++ | + | + | + |
| Robustness to outliers | + | +++ | +++ | + | + |
| Ability to extract linear combinations | +++ | ++ | + | +++ | +++ |
| Ability to extract nonlinear combinations | + | ++ | ++ | +++ | +++ |
| Ability to work with small samples | +++ | +++ | +++ | + | + |

**Table 6.1: Model comparison** - Comparative features of the popular models provided in the previous section. Symbols: + → poor, ++ → fair, +++ → good.

## 6.6 Evaluation of Time Series Forecasting

The classification and regression results of a machine learning method need to be evaluated before we can have any confidence in their predictions. As shown in Figure 6.3, performance on the training set is a biased indication of the performance on an independent data set. To determine more accurately the error of a learning algorithm on an unseen data set, we need to assess its error rate on a sample that was not used in the training phase. This independent data set is called the test set [215]. In practice, the

machine learning algorithms are usually trained with two-thirds of the data available and tested with the one-third remaining.

In both the training and the test phases, the samples shall be representative of the system modeled otherwise they will not be able to describe the system when the algorithm is actually deployed in a unknown set. Particularly, in time series forecasting the training and test sets have a chronological dependence. They cannot be picked randomly as in a typical classification problems. To represent the system during its real application, the training set must contain data with some time dependence and it is paramount to be prior to the test set.

### 6.6.1 Cross Validation

If the amount of data available is large enough to be partitioned into training and test sets and still produce statistic significant results, a unique partition between training and test sets is the most appropriated to evaluate the forecaster. However, it is common that the quality data points available are not sufficient. For example, monthly data in a ten years data set will produce only 120 data points for training and testing. In this case, the cross validation is a preferred methodology to assess the system [227]. In cross validation, the data are divided into a certain number $n$ of partitions or "folds". To assess the system, each fold is held out as the test data and the remaining $n-1$ folds are used as the training set. The performance of the algorithm achieved in each test fold is averaged to estimated the overall system accuracy. In time series analysis, the leave-one-out cross validation is a standard method to evaluated the forecaster. The training set is created with a minimum of observations required to train the system and the independent test set is defined with the size of the forecasting horizon. At each evaluation iteration, the training and test sets are shifted to the future. This process continues until the test set reaches the last data point. We deploy the leave-one-out to train and test our forecaster.

### 6.6.2 Loss Function

A loss function need to be defined for the assessment of a learning algorithm. It determines the measure that will evaluate the algorithm. In time series forecasting, the error can be computed as a function of the observed values and the forecast, that is, $e = f(observed, predicted)$. Then, the loss function can be specified as a function of

the error $L(e)$. A common approach is to define the error as $e = observed - predicted$. Thus, to calculate the absolute error the loss function would be $L(e) = |e|$. Sometimes, the forecast is also evaluated in terms of the squared error, that is, $L(e) = e^2$. The final error is commonly taken as the average of the predictions. Several other loss functions to assess forecast are presented in the literature [25]. In our evaluations, we will use the two methods aforementioned.

## 6.7 Summary

In this review, we discussed the machine learning methods applied to the problem of time series forecasting. Many machine learning algorithms have been presented in the literature to forecast time series data. They are very well exploited in finance, economy, etc., and with less frequency in biomedical sciences. The biomedical community is still using classical models for time series forecasting of resistance data. To the best of our knowledge, there is no published approach that deploys data intensive machine learning methods to the problem of antimicrobial resistance forecasting. In the next chapters, we will investigate the use of such approach for modeling and forecasting short-term resistance rates.

# 7

# Data Driven Antibiotic Resistance Trend Extraction and Forecasting

## 7.1 Introduction

At the point of care, up-to-date antimicrobial resistance information is important for empirical therapies because it reproduces faithfully the current resistance dynamics within the clinical setting. However, high frequency resistance time series, that is, those containing daily and weekly aggregated information from antimicrobial susceptibility tests, are challenging to analyze. In this thesis, we investigate a method that aims to improve analysis algorithms for antibiotic resistance data by building a novel, fully data-driven trend extraction and machine learning forecasting model for resistance trends. Trend extraction and forecasting tools model real-value functions using multi-dimensional vectorial space of events. Our method consists in breaking down the resistance time series into different oscillation modes using the EMD technique and use the vectorial space generated to represent the system's function. The resulting waveforms, which describe intrinsic resistance trends, are then used as the input for a machine learning algorithm based on the k-NN framework for projecting mappings from past events into the future dimension, that is, the forecast.

This chapter introduces our approach and provides a basic evaluation of our methodology for both trend extraction and forecasting. It starts by describing the main features

of resistance time series and the challenges of modeling short-term (days, weeks) resistance trends. Section 7.3 presents some existing methods to decompose time varying signals and provides some examples of their application to extract trends. Section 7.4 introduces our antimicrobial resistance trend model and describes in detail the forecasting algorithm. Finally, in Section 7.5 we compare different machine learning algorithms against the k-NN algorithm by showing the empirical results of resistance forecasting.

## 7.2 Analysis of Resistance Data on the Time Domain

Antimicrobial susceptibility tests are performed routinely in microbiology laboratories. Clinicians use the results primarily to determine the optimal agent for the antibiotherapy. Secondarily, microbiologists, infectiologists and public health specialists exploit the reports to analyze how resistance rates vary over time and plan interventions. The most common methodology applies time series of long-term (yearly aggregated) resistance rates in order to detect trends [228, 229], where large historic data sets are compared with recent rates to spot any significant variation in the mean resistance rates. Some works model the evolution of resistance using differential equations that incorporate variables associated with resistance, such as consumption of antibiotics [230]. This approach can be used to forecast resistance, but also to infer the effects of different factors upon resistance. In all of these analyzes, the time dimension is a keystone to understand the process of antimicrobial resistance and evaluate its impact on healthcare.

Data acquisition and sampling rates are two important variables to consider when analyzing data in the time domain. The prevalence of infections in the community and of nosocomial infections within a clinical setting are factors that influence the amount of antimicrobial susceptibility tests performed and consequently the acquisition rate of the microbiology database. Antimicrobial tests are performed and reported obeying many factors, varying from societal, such as weekends and holidays, to clinical, such as resistance outbreaks. Hence, one should expect an irregular data acquisition rate. However, for time series analysis, the sampling rate needs to be as constant as possible. In other to achieve such a fixed rate, higher frequency data need to be aggregated to reduce all data to the same sampling period. For resistance analysis, five sampling periods are prominent: annual, quarterly, monthly, weekly and daily resistance rates. Since we are interest in providing methods for antimicrobial resistance analysis using

up-to-date data, our model focuses on short-term resistance trends. Then, we naturally exclude the first three sampling rates. On the other hand, often daily resistance rates do not have clinical meaning, especially because tests are not performed daily for most of the pathogens. Technically, it is means that the sampling rate is higher than the acquisition rate. Therefore, we center our attention to weekly aggregated antimicrobial resistance rates.

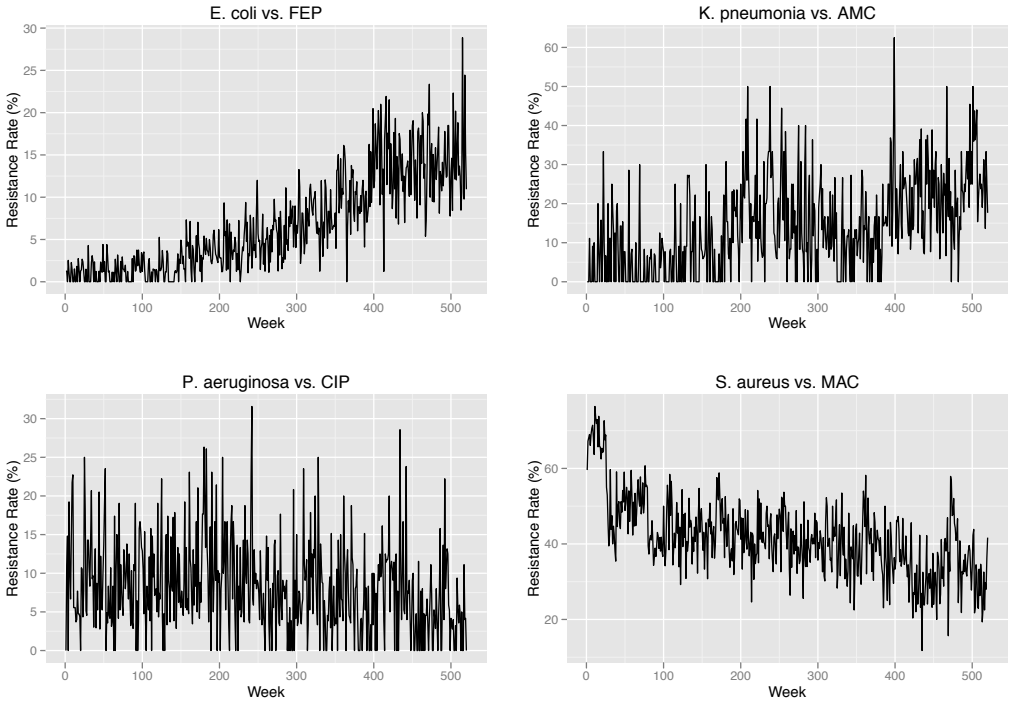


**Figure 7.1: Resistance time plots** - Example of weekly resistance time series for four pathogens: *Escherichia coli*, *Klebsiella pneumonia*, *Pseudomonas aeruginosa* and *Staphylococcus aureus*. FEP = cefepime; AMC = amoxicillin-clavulanic acid; CIP = ciprofloxacin; MAC = macrolide.

The first step in time series modeling is to plot the observations against time, to produce the so called time plot of the data [24]. Time plots may reveal at a glance important features of a time series, such as trends, seasonality and outliers. In Figure 7.1, we show some time plots of antimicrobial resistance rate using a decade of weekly aggregated antibiogram tests for four pathogen/antibiogram pairs. A priori, we can say that there exist trends for the time series at the top left (upward) and bottom right (downward). The presence of outliers is verified in all the series but no apparent

seasonal events. The time series of different pathogen/antimicrobial pairs are very distinct and, at the first sight, are contaminated by random signals. Indeed, modeling antimicrobial resistance time series poses many challenges. First, the dynamics of the time series depends on the resistance stage. The function can present bursts, as usually seen at initial resistance states or in resistance outbreaks, but also a slow varying underlying trend, which is more common once the pathogen has acquired some level of resistance. In any stage though, the signal-to-noise ratio of the time series is relatively low, making the system's modeling task more difficult whatever methodology is adopted. Second, the sampled data within a clinical setting are only a local fraction of the data. Antimicrobial resistance is a much larger problem affecting human but also animal pathogens, the latter not being obviously captured in clinical databases. Hence, data from clinical microbiology databases are not representative of all the information related to the resistance dynamics. Therefore, the antimicrobial resistance algorithm or model has to account for unseen information to avoid the overfitting phenomena. Finally, the dynamics of antimicrobial resistance are very distinct from antimicrobial to antimicrobial and from pathogen to pathogen. Thus, the model has to be general enough to adapt to different time series dynamics, but also has to account for the specificity of each case of interest in order to provide good fit and forecasting accuracy.

Another important tool in time series analysis is the autocorrelation function. In particular, autocorrelation plays an important role in forecasting. Probabilistically, autocorrelated time series are to some extent predictable because future values have dependency on the present and past data and then forecasting models can learn about the behavior of the system using observed values. Figure 7.2 shows the autocorrelation plots for the time series displayed in Figure 7.1. The plot at the top left shows a strong correlation from one observation to the next, maintaining the autocorrelation after lag 0 almost constant as we walk to the past. In the two plots in the right, autocorrelation decreases roughly linearly as the lag increases. Finally, in the plot at the bottom left, the time series presents almost no autocorrelation. Antimicrobial resistance time series are often autocorrelated because of the slow dynamics and carryover of the evolutionary processes. As can be seen from Figure 7.2, the time series usually show at least some degree of positive autocorrelation, characterizing certain tendency for the resistance to remain in the same state or trend from a past observation to the future. To illustrate that, imagine that for a given pathogen/antibiotic time series, resistance has been

increasing for the past months. Thus, the likelihood of resistance next month being higher than today is greater than if the time series had a negative slope. However, due to the variations in several factors associated to resistance, such as antimicrobial consumption and infection control measures, and how these factors are reported in the database (technical and societal factors), resistance time series may sometimes present no autocorrelation, as in the example shown at the bottom left of Figure 7.2. Hence, we can expect that some antimicrobial resistance time series with higher (negative or positive) autocorrelation will result in smaller forecasting errors, while others it will be more challenging to forecast.

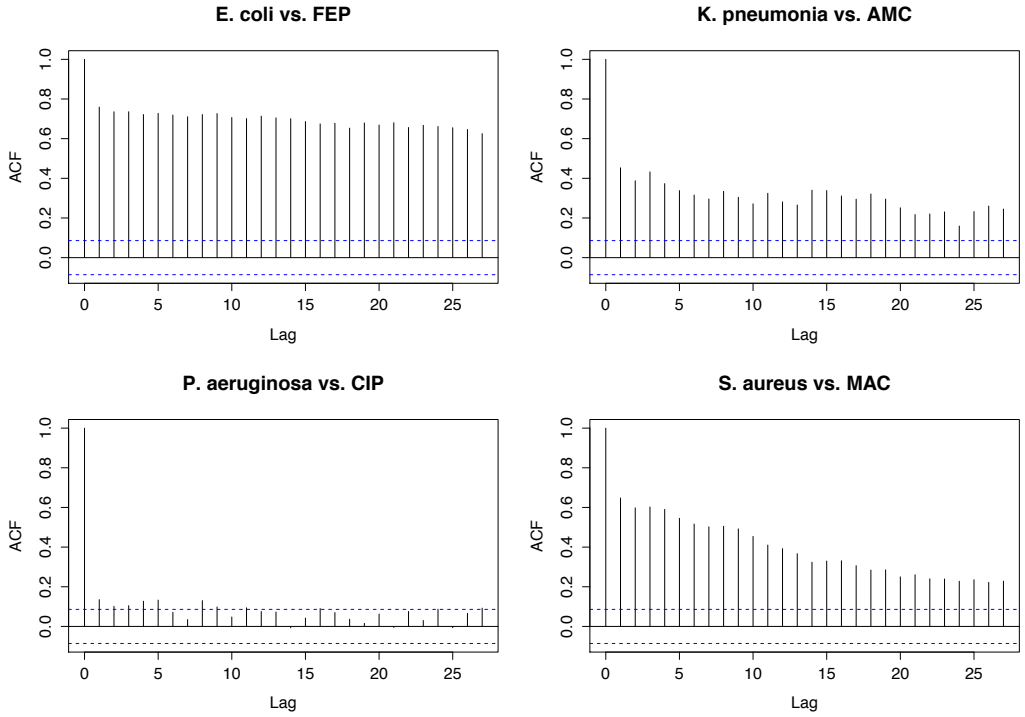


**Figure 7.2: Resistance autocorrelation** - Autocorrelation plots for time series of Figure 7.1

## 7.3 Antimicrobial Resistance Trend Extraction

As shown in Figure 7.1, weekly resistance signals are complex, having a large part of the signal contaminated by noise. This increases the challenge for computer algorithms to

learn existing patterns from the real signal. We propose a methodology to analyze and learn the resistance trends by decomposing the signal in different modes of oscillation. Instead of looking at the time series as a single signal, first we decompose it in a set of simpler and easier to learn data sequences. Depending on the characteristics of the new decomposed signal, different machine learning techniques can be applied taking into account the features of the processed signal and of the learning algorithm. Furthermore, through transforming the signal, new information or features can be obtained from the signal that is not readily available in the raw signal. In the next sections, we give an overview of some existing algorithms that can be employed in signal decomposition.

### 7.3.1 Hodrick–Prescott Filter

The Hodrick–Prescott filter [231] is a model-free mathematical algorithm that is applied particularly in macroeconomics to obtain long-term trends from time series. The algorithm assumes that a given time series $x_t$ can be decomposed into a slowly evolving trend component $\tau_t$ and a cyclical component $c_t$, such that

$$x_t = \tau_t + c_t, \tag{7.1}$$

where the cyclical components $c_t$ average tends to zero over long periods. Figure 7.3 shows an example of applying the Hodrick–Prescott filter to decompose a time series into the cyclical $c_t$ (C1) and trend $\tau_t$ (C2) components.

The Hodrick–Prescott filter estimates the trend $\tau_t$ using cubic smoothing spline method, where the following equation is minimized for $\tau_t$:

$$\min_{\{\tau_t\}} \left\{ \sum_{t=1}^{T} (x_t - \tau_t)^2 + \lambda \sum_{t=2}^{T-1} [(\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})]^2 \right\}, \tag{7.2}$$

with $\lambda \geq 0$. The first term of Equation 7.2 is the sum of the squared deviations of $x_t$ from the trend and penalizes the cyclical component. The second term, which performs a sum over the squared second differences of the trend, is a penalty for variability in the trend. To obtain the cyclical component, we can just substitute the resulting trend $\tau_t$ of Equation 7.2 into Equation 7.1.

The adjustment of the sensitivity of the trend to high frequency components is achieved by modifying the smoothing factor $\lambda$. As $\lambda$ decreases towards 0, the trend approximates the raw signal $x_t$, whereas the greater the penalty imposed by $\lambda$, the

smoother the resulting trend will be, with $\tau_t$ becoming linear for $\lambda \to \infty$. In practice, $\lambda$ is assigned empirically. For example, for quarterly data Hodrick and Prescott suggest a smoothness parameter of 1600. The Hodrick–Prescott filter was not designed to be optimal for specific time series and, apart from the choice of $\lambda$, the same filter can be used in processes with difference dynamics [232]. However, under certain conditions, namely the normal and independent distribution of the $c_t$ component and of the second difference of $\tau_t$, $\Delta^2\tau = (\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})$, the Hodrick–Prescott filter is an "optimal filter", that is, it optimizes the mean squared error, and $\lambda$ can be computed as the ratio of the cyclical and trend variances, $\sigma_c^2/\sigma_{\Delta^2\tau}^2$, where $\Delta^2$ is the second difference operator [233].

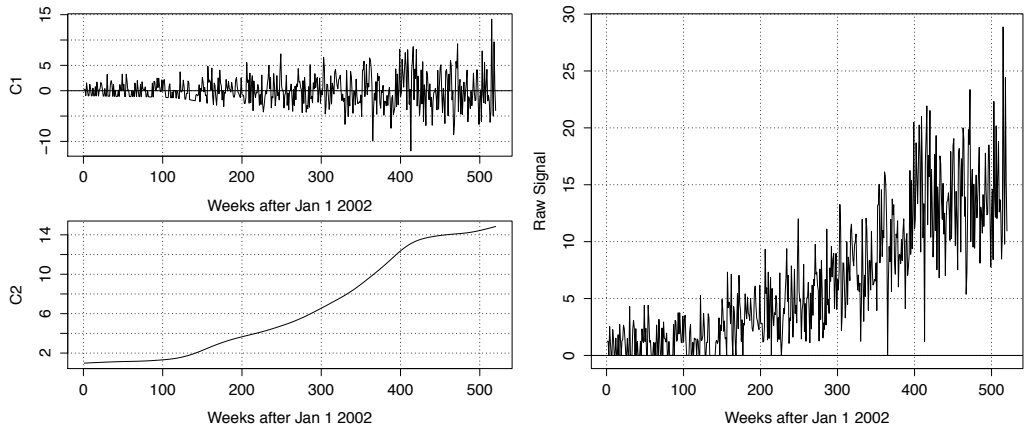


**Figure 7.3: Hodrick-Prescott trend extraction** - Component *C1* provides the cyclical component $c_t$ whereas component *C2* provides the trend $\tau_t$ for the raw signal displayed on the right.

### 7.3.2 Wavelets

Wavelet theory provides a formal mathematical framework for decomposing a signal as a weighted sum of basis functions with different scales [234]. The method consists in adopting a wavelet prototype function, called an analyzing wavelet or mother wavelet, and decomposing the signal using such function as basis (for some examples of wavelet basis function, please refer to [234]). Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet. In this sense, a wavelet transform is similar to the Fourier transformation, where a given signal is decomposed into a

series of sinusoids with different frequencies. However, when compared to Fourier transforms, it presents a key advantage, its temporal resolution, which allows the wavelets to capture both frequency and location in time information. Hence, they perform better in analyzing physical situations where the signal contains discontinuities and sharp spikes.

Wavelet methods decompose the signal into a set of orthogonal signals, containing a coarser signal approximation $A$ at large scale (low frequency) and additional signal details $D$ at different resolutions, of decreasing scales. The approximation $A_j$ at level $j$ roughly represents the local mean signal on intervals of length $2^j$ while the detail $D_j$ at level $j$ contains fluctuations around this local mean on the same corresponding intervals. Let us consider a time series $x$ with $N$ observations. Then, for a given orthogonal wavelet basis function $y$, the time series can be decomposed as [235]:

$$x = \sum_{j=1}^{n} D_j + A_n, \tag{7.3}$$

where $n$ is an integer in the range $1 \leq n \leq \log_2(N)$ and denotes the decomposition level, $A_n$ is the approximation level and $D_j$ is the detail at level $j$ of the signal.

Figure 7.4 shows an example of time series decomposition using the wavelet method with the least asymmetric filter of length 8 as the wavelet basis function at the decomposition level $n = 4$. The first four components (C1-4) show the details from the finer (D1) to the coarser (D4), whereas the component C5 displays the candidate trend (A4) for the respective raw signal at the bottom right.

### 7.3.3 Empirical Mode Decomposition

EMD is an empirical, adaptive and fully data-driven method for signal decomposition suitable for nonlinear and non-stationary processes [35, 236, 237]. It works by breaking down the signal as superpositions of intrinsic local functions with different modes of oscillation called IMFs. According to Huang [35], each IMF satisfies two particular conditions: $(i)$ in the whole dataset, the number of extrema, that is, the local minima or maxima, and the number of zero crossings must either equal or differ at most by one; and $(ii)$ at any point, the mean value of the envelopes defined by the local maxima (upper) and the local minima (lower) is zero. The result of the EMD process is a set of

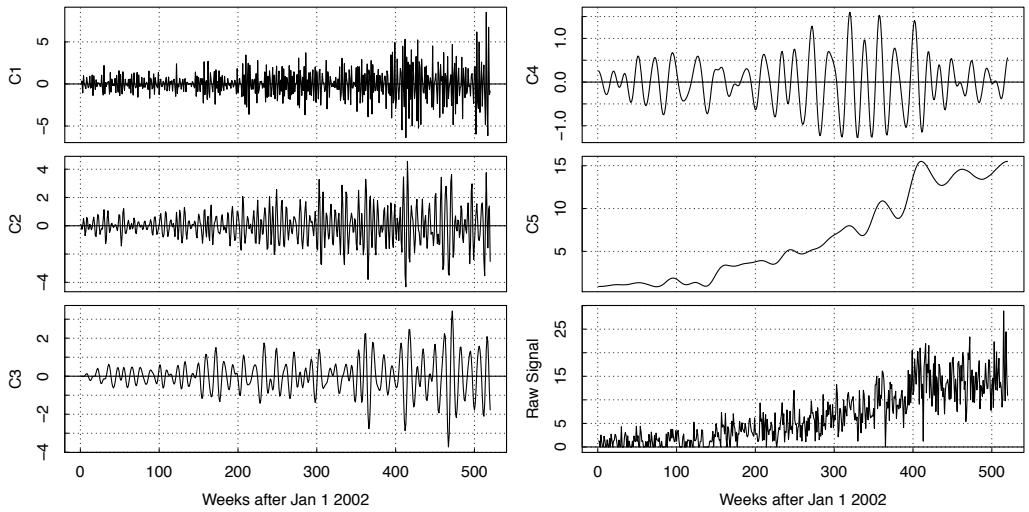


**Figure 7.4: Wavelet signal decomposition** - C5: signal approximation $A_n$; C1-4: details $D_j$. The raw signal is equivalent to the sum of the five components C1-5.

IMF components, with zero mean and unrestricted amplitude and frequency along the time axis, and a residual component, which accounts for the mean underlying trend.

IMFs are extracted from the signal through a sifting process, which can be implemented according to the following algorithm (see Figure 7.5):

1. Identify the local maxima and minima extrema of a signal $x(t)$.

2. Connect the local maxima with a cubic spline as the upper envelope $e_{max}(t)$. Repeat the process for the local minima to create the lower envelope $e_{min}(t)$.

3. At every time point $t$, calculate the local mean $m(t)$ of the two envelopes given by the average of the upper and lower envelopes:

$$m(t) = \frac{e_{max}(t) + e_{min}(t)}{2}. \tag{7.4}$$

4. Obtain the first oscillation component $h(t)$ by taking the difference between the data signal $x(t)$ and the local mean $m(t)$, that is, $h(t) = x(t) - m(t)$.

5. If the first component $h(t)$ is not an IMF, it is taken as the new signal $x(t)$ and steps 1-4 are repeated until the first component is an IMF. The final $h(t)$ is designated as $c_j(t)$, the $j$th IMF component.

6. Once the first IMF component $c_j(t)$ has been identified, it is subtracted from the original signal, leaving a residual $r(t) = x(t) - c_j(t)$. Steps 1-5 are repeated with $r(t)$ taking the place of $x(t)$ until $r(t)$ becomes a monotonic function from which no more IMFs may be extracted.

After the data signal $x(t)$ has passed through the IMF sifting process, it can be represented in terms of the IMFs $c_j(t)$ and the monotone residual component $r(t)$ as

$$x(t) = \sum_{j=1}^{n} c_j(t) + r(t), \tag{7.5}$$

where $n$ is the number of IMFs obtained in the sifting process.

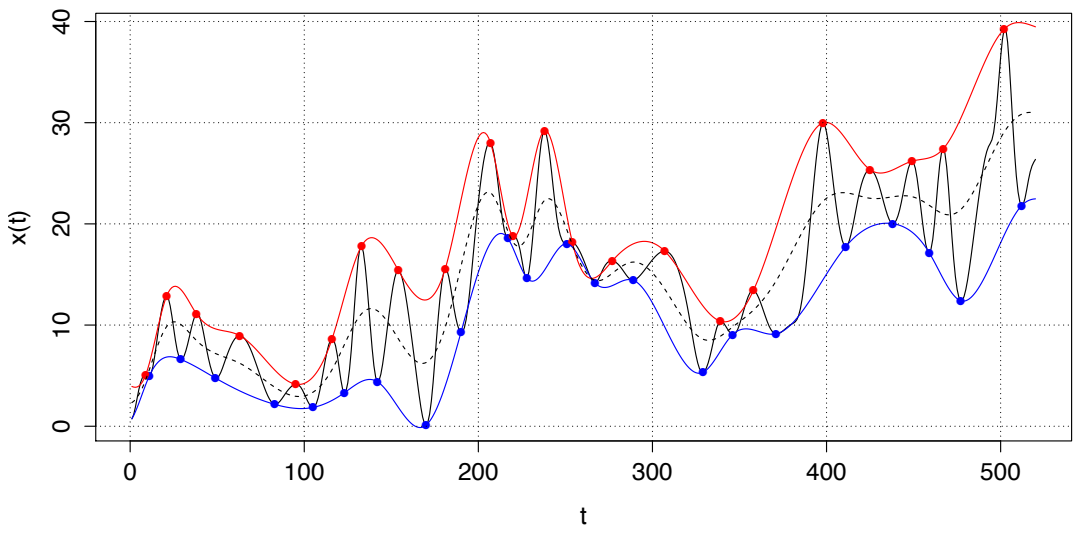


**Figure 7.5: EMD – IMF sifting process** - Signal $x(t)$ – black continuous line; upper envelope $e_{max}(t)$ – red line; lower envelop $e_{min}(t)$ – blue line; local mean $m(t)$ – black dashed line.

The residual $r(t)$ in Equation 7.5 provides the mean trend of the signal. The oscillating components $c_j(t)$ are usually physically meaningful [236] and represent short, medium- and long-term trends (see Figure 7.6). They might be associated for example with seasonal trends, or cyclical components in the econometrics parlance. The first component $c_1(t)$ has the smallest time scale and thus corresponds to the highest frequency component. As such, it is associated to noise. Notice that the components are extracted using a fully data-driven process, where it is not required to predetermine

any basis functions. Therefore, this methodology is adaptive to any time varying signal, which makes it suitable to extract trends from the different pathogen-antimicrobial resistance time series.

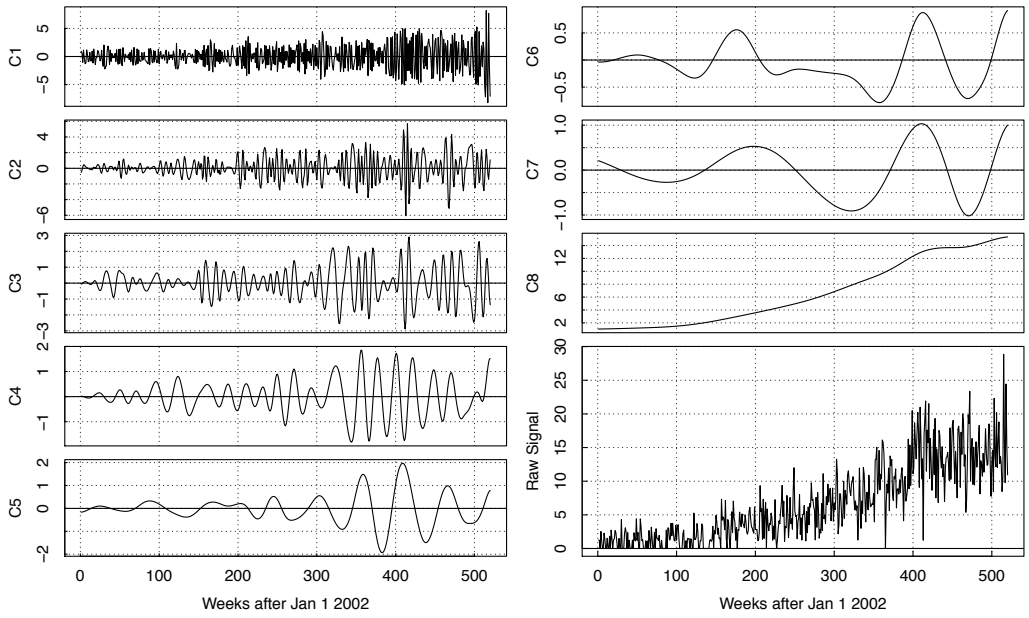


**Figure 7.6: EMD decomposition** - C8: residual trend $r(t)$; C1-7: IMFs $c_j(t)$. The raw signal is equivalent to the sum of the eight components C1-8.

### 7.3.4 Comparison

Figure 7.7 shows a comparison of the different methods to extract trends applied to the resistance time series of Figure 7.1. The three signal decomposition methods described previously are able to generalize and adapt to time series with different dynamics. However, compared to the Hodrick–Prescott and wavelet methods, EMD provides some advantages. First, it provides a good estimation of the slow varying mean trend as the Hodrick–Prescott method. Notice that the wavelet method cannot describe the mean trend but rather an oscillatory curve. Second, it is also able to decompose the signal into several intrinsic oscillatory components of different frequencies as in wavelets, not being restricted to a single oscillatory component as in Hodrick–Prescott. Furthermore, the algorithm is simple to implement and there is no parameter to determine, as the smoothing parameter $\lambda$ in the Hodrick–Prescott filter or the decomposition level $n$ in

wavelets. Finally, there is no assumption of any basis function. EMD is strictly data-driven, differently from wavelets, for example, where a mother wavelet basis function is used to fit the model. Therefore, we employ the EMD algorithm as our tool to extract intrinsic trends of the resistance rate time series.

## 7.4 Machine Learning Forecasting for Antimicrobial Resistance Time Series

There are two main approaches to model and forecast antimicrobial resistance rates. The first is based on pure time series analysis, where only observed resistance rates are used to model and predict future resistance values [209, 238]. The other approach uses differential models that incorporate several biological and clinical variables that are associated to resistance, such as the basic reproduction number of the pathogen, number of patients in the clinical setting, admission and discharge rates, hand-washing compliance and resistance prevalence on admission, to describe resistance evolution [230, 239, 240].

Whereas differential models are useful for some special cases, particularly for assessing the impact of interventions on resistance, they have some weaknesses as a forecasting model. First, they are theoretically valid only under certain a priori conditions, such as constant antibiotic pressure, which are often violated. Second, usually these models contain many parameters that are hard to estimate. For example, the definition of the transmission rate variable, used in some equations to extrapolate future resistance rates is in itself a complex task since, in general, there is no evidence in the data that can be used to confirm the optimal value. At best, the forecast is as accurate as the estimation of the several model parameters. Finally, some models, like those based on logistic regression, may fail for the trivial cases in which resistance starts to decrease over time after some initial increasing – a phenomenon verified often from actual microbiology data [230, 241].

On the other hand, data-driven analysis as in machine learning is more general and can be applied mostly out-of-the-box to different antimicrobial resistance time series.

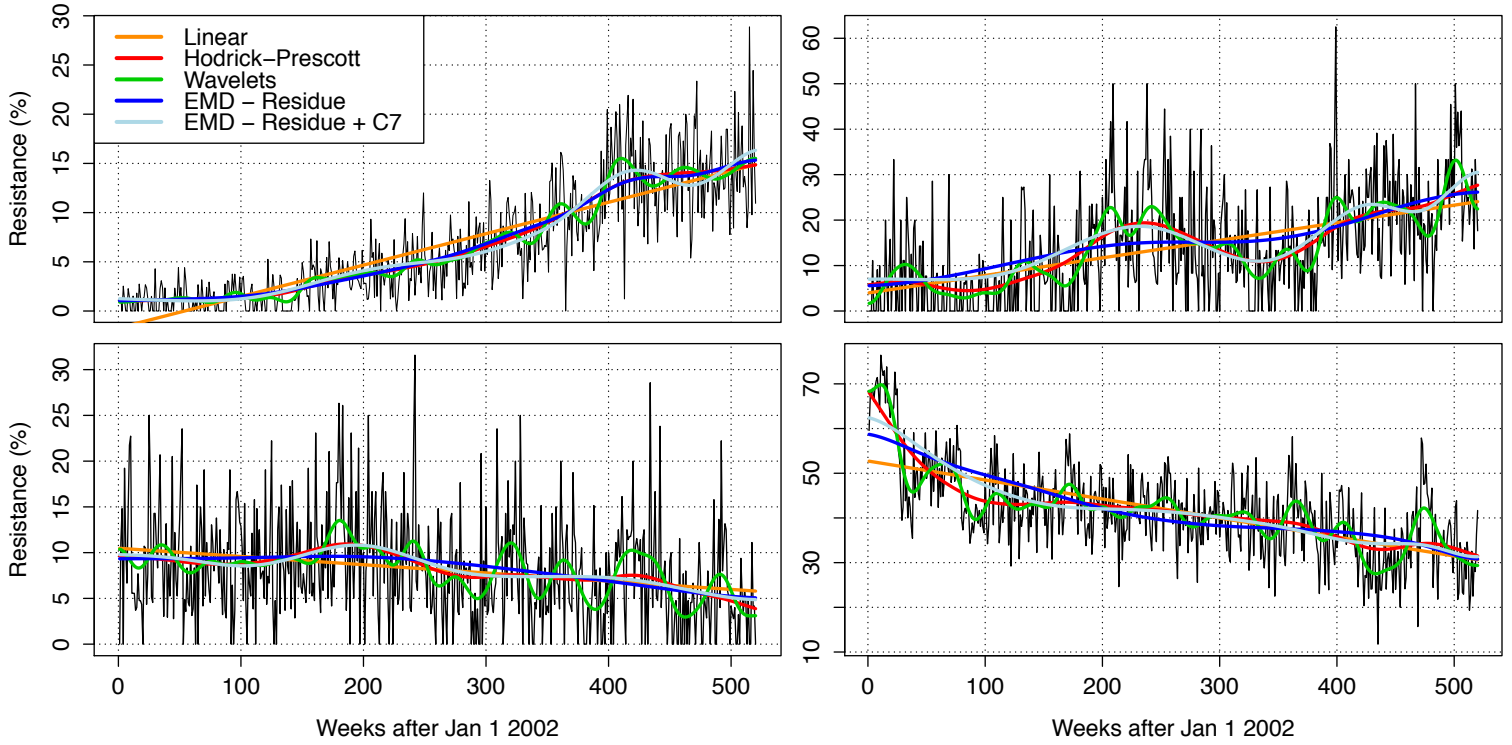


**Figure 7.7: Trends extraction methods** - Comparison between different methods to extract resistance trends applied to the time series of Figure 7.1. EMD provides a good estimation of the mean trend (*EMD - Residue*) but it also describes other components, as in the example of *EMD - Residue + C7*.

While modeling antimicrobial resistance using only chronologic ordered resistance rates might provide less insight into the resistance process itself when compared to multivariate models, the algorithm has no dependence on the underlying resistance model. As long as there are enough example data, the system shall be able to model and forecast resistance time series independent of the actual resistance dynamics. This is particularly important in resistance forecasting since, as we have seen, the dynamics of the different pathogen/antimicrobial time series are very diverse. Furthermore, the same algorithm can be potentially deployed into different environments, as in hospitals, where with some data mining efforts, drug consumption could be eventually associated to resistance, but also in laboratories, where access to antimicrobial prescription information is not available, or even in nontherapeutic areas, as in animal husbandry, which is believed to account for a large part of antibiotic consumption and so resistance [242].

### 7.4.1 Modeling Resistance Rate Time Series

As we can see from Figure 7.1, short-term resistance trends are generated by a nonlinear process. To be able to model such processes, our forecasting algorithm applies the delay coordinate embedding theorem to derive the training set [36, 37]. This theorem describes a phase space reconstruction technique that provides the conditions for nonlinear dynamical systems to be reconstructed from a finite sequence of observations of the system's state. Let us consider a time series $x$, that is, a set of chronologically ordered events $x_1, x_2, \ldots, x_N$ generated from a nonlinear system $f(\cdot)$. In delay coordinate embedding, vectors in the new phase space, the embedding space, are defined by

$$x'_n = \{x_{n-(m-1)\tau}, x_{n-(m-2)\tau}, \ldots, x_n\}, \tag{7.6}$$

where $n$ is the current state in the embedding space, $m$ is the embedding dimension and $\tau$ is the delay time or lag relative to the sampling rate. Equation 7.6 provides a multidimensional representation of a unidimensional nonlinear time series, which according to Takens [36] and Sauer *et al.* [37] can be used to reconstruct the observations made with a generic unknown function $f(\cdot)$ of a nonlinear dynamical system. The dimension $m$ can be considered as the minimum number of state variables required to describe the system. For the sake of exposition, in the remainder of this thesis we take by convention $\tau = 1$.

Now, considering the same one-dimensional time series $x$ generated by a system $f(\cdot)$. Our goal is to forecast $x_{N+h}$ at the time point $N$, where $h$ is called the forecast horizon and represents how far in the future, $h = 1, 2, \ldots$, with respect to $x_N$ is the predicted point. Then, the forecasting algorithm is a method for projecting future values, $\hat{x}_{N+h}$, relying only on observed values of the given time series $x$ [24]. Formally, we can write it as

$$\hat{x}_{N+h} = \hat{f}(x), \tag{7.7}$$

where $\hat{f}(\cdot)$ is the estimated function of the actual system $f(\cdot)$ and $\hat{x}_{N+h}$ is the estimate of the $N + h$ system state. Similarly, we can project the future system state using the delay vectors of Equation 7.6 such that

$$\hat{x}_{N+h} = \hat{f}(x'). \tag{7.8}$$

Under suitable hypotheses on the dynamics, the correspondence presented in Equation 7.8 is one-to-one, which means that the behavior of the nonlinear system is accounted for in the behavior of the delay coordinate embedding defined by the mapping $\hat{f}(\cdot)$ [36].

### 7.4.2 Estimating the Function $f$

To estimate the function $f(\cdot)$ that generates the resistance rates we employ a machine learning approach using the *k-nearest embedding vectors*. Machine learning forecasting algorithms use the observed data points $x$ as examples to learn the unknown model. To have enough learning examples, the time series are divided into smaller data sequences, $D = \{s_1, s_2, \ldots, s_v\}$, where $s_i$ represents a training example $\{s_{input} \rightarrow s_{output}\}$ and $v$ is the number of examples in the training set $D$. These sequences are then fed to the learning algorithm, which creates a decision function $\hat{f}(\cdot)$ that models the behavior of the system. To predict a point in the future, the algorithm compares a given test input $t_h$ using the decision function $\hat{f}(\cdot)$ and estimates the output value $\hat{x}_{N+h}$.

There are several ways to estimate the function $f(\cdot)$ to obtain the point forecast $\hat{x}_{N+h}$ [243]. Our model employs the k-NN framework as a piecewise estimator of $f(\cdot)$. The k-NN algorithm implements a function approximator that stores a set of mappings $x'_i \rightarrow x_{i+h}$. The delay coordinate vector $x'_i$ acts then as a surrogate for $x_i$. When the query point $x'_{i+\delta}$ of a future state $i + \delta$ is performed, the $k$ delay vectors most similar (in a Euclidian space) to the query state are extracted. In the ideal case, we find an

exact match $x'_i$ and use $x_{i+h}$ as our prediction for $x_{i+\delta}$. If the neighborhood contains more than one delay vector, the value of the query state is computed as the average of the $k$ extracted samples, that is,

$$\hat{x}_{N+h} = \frac{1}{|\mathcal{U}_k|} \sum_{x'_i \in \mathcal{U}_k} x'_i, \tag{7.9}$$

where $\mathcal{U}_k$ is a neighborhood of size $\epsilon$ in the space defined by the embedding vectors $x'_i$ and $|\mathcal{U}_k| \doteq k$ is the number of neighbors.

### 7.4.3 k-Nearest Embedding Vectors Forecasting Algorithm

Our forecasting system is depicted in Figure 7.8. To simplify discussion, we refer from now on to all the IMF components $c_j(t)$ and the residue $r(t)$ obtained from the decomposition process simply as *components*, and denote them by the vector $C = \{c_1, \ldots, c_n, r\}$ of length $n + 1$, where $C_1$ corresponds to the first IMF and $C_{n+1}$ to the residue $r(t)$. The system breaks down the input resistance rate time series $x$ into several oscillating components using the EMD algorithm. Then, components that do not contribute to the signal are removed and those remaining are embedded into delay vectors of dimension $m$. Further, the algorithm is trained to compute the size of the delay vector neighborhood $k$, which will provide the best estimate of the forecast $\hat{y}$. Thus, from a machine learning viewpoint, the task of the learning algorithm resumes to find the EMD components $C_i$ that represent best the system being modeled, the dimension $m$ of the embedding sequences and the number $k$ of nearest neighbors that encompass the dynamics of the system.

In the following steps, we summarize the computationally efficient leave-one-out cross-validation algorithm that we used to train and test our machine learning forecasting system:

1. Divide the input time series of size $N$ into two independent parts: a training set $x_i = \{x_1, \ldots, x_{N'}\}$ and a testing set $y_i = \{x_{N'+1}, \ldots, x_{N'+h}\}$, where $N' < N$ is the minimum number of observations necessary to fit the model and $h$ is the forecasting horizon.

2. Decompose the training time series $x_i$ into components $C_i$ using the EMD algorithm.

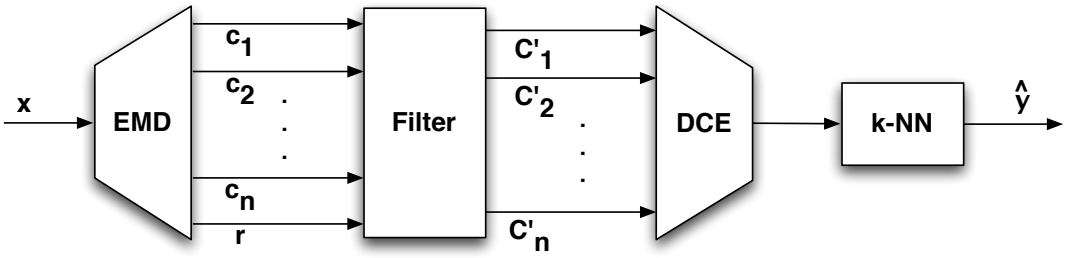


**Figure 7.8: High-level block diagram of the k-nearest embedding vectors forecaster.** - The *EMD* block decomposes the input $x$. Then, the *filter* block selects the time series functions that are relevant to the signal. Further, the *delay coordinate embedding* (*DCE*) block determines the embedding dimension $m$ and embeds the signal $C'$ into a multidimensional space. Finally, the *k-NN* block calculates the distance between the input query and the training points in the embedded space and projects them in the future dimension to obtain the forecasting $\hat{y}$

3. Select the subset of components $C_i' \subseteq C_i$ that are relevant to the learning model.

4. Compute the optimal embedding dimension $m$ for the space created by $C_i'$.

5. Embed the components $C_i'$ into a space of dimension $m$ and together with the respective one-step-ahead output create the training set

$$D_{ij} = \{s_{ij-(m-1)}, s_{ij-(m-2)}, \ldots, s_{ij} \rightarrow x_{ij+1}\}, \tag{7.10}$$

   where $m \leq j \leq N' - 1$ and $s_{ij}$ is a vector containing the $j$th elements of the components $C_i'$. Then, compute the optimal number of nearest neighbors $k$ for the training set $D_{ij}$ using cross validation.

6. For $h' = 1, \ldots, h$:

   (a) Create the test input using the latest embedded vector

$$t_{ih'} = \{s_{iN'-(m-1)}, \ldots, s_{iN'}\} \tag{7.11}$$

   from the components $C_i'$. Then, using the training model $\{D_{ij}, k\}$ created in step 5, find the $k$ nearest neighbors of the test input $t_{ih'}$. Finally, project the $k$ embedding vectors into the dimension $N' + h'$ using Equation 7.9 to estimate the $h'$-step-ahead forecast $\hat{x}_{N'+h'}$.

(b) While $h' < h$, concatenate the forecast outcome $\hat{x}_{N'+h'}$ into the time series $x$ and repeat steps 2 and 3 to update the components $C'_i$.

7. Compute the residual error using the forecasted values $\hat{y}_i = \{\hat{x}_{N'+1}, \ldots, \hat{x}_{N'+h}\}$: $E_{iN'} = y_i - \hat{y}_i$.

8. Increase $N'$ and go to step 1 while $N' < N$.

9. Compute the overall cross-validation error using the mean absolute error

$$MAE = \frac{1}{h}\sum_{j=1}^{h}\left(\frac{1}{N - N'_o}\sum_{i=N'_o+1}^{N}|E_{ij}|\right), \tag{7.12}$$

and root mean squared error

$$RMSE = \frac{1}{h}\sum_{j=1}^{h}\sqrt{\frac{1}{N - N'_o}\sum_{i=N'_o+1}^{N}E_{ij}^2}, \tag{7.13}$$

cost functions, where $N'_o$ is the initial minimum number of observations.

In our algorithm, the prediction is made using the latest sequence chunk. Further steps-ahead are computed using the one-step-ahead forecasting as the latest arrival chunk. An optional algorithm would create a training set for each h-step-ahead forecast of interest and, at step 5, the one-step-ahead mapping would be replaced by a h-step-ahead. Then, the loop in step 6 would be avoided. However, it would require one training model for each step-ahead forecasting $1, \ldots, h$, which is computationally more expensive.

### 7.4.4 Selecting the Components $C'$

We envisage three models to select the relevant EMD components $C'$. The first model, *DECA*, does not actually filter any component and thus the system is trained with the full signal spectrum. For the other two models, we make a fair assumption that the machine learning algorithm cannot learn the noisy components and hence they shall be excluded from the signal to avoid a negative impact on the learning model. The remaining components, which correspond to the physically meaningful signals, are then used to train the system. Based on this assumption, the second model (Figure 7.9 - left), *DECF*, filters out noisy components using a frequency threshold. High frequency

components are naïvely associated with noise. We consider a period of 10 weeks as the minimum necessary to learn the signal. Components with lower periods are filtered out. The last model (Figure 7.9 - right) uses a statistical significance test derived by Wu and Huang [35] to distinguish between noise and signal in the IMF components. The test assumes that the first IMF is a random noise. Then, other components are compared with this IMF using a distance metric based on the logarithm of the component's variance and period [236]. The components, whose variance and period exceed the noise bounds, are considered to contain statistically significant information for the signal. In our experiments, we use a $2\sigma$ distance for the noise boundaries. This model is further referred as *DECS*.

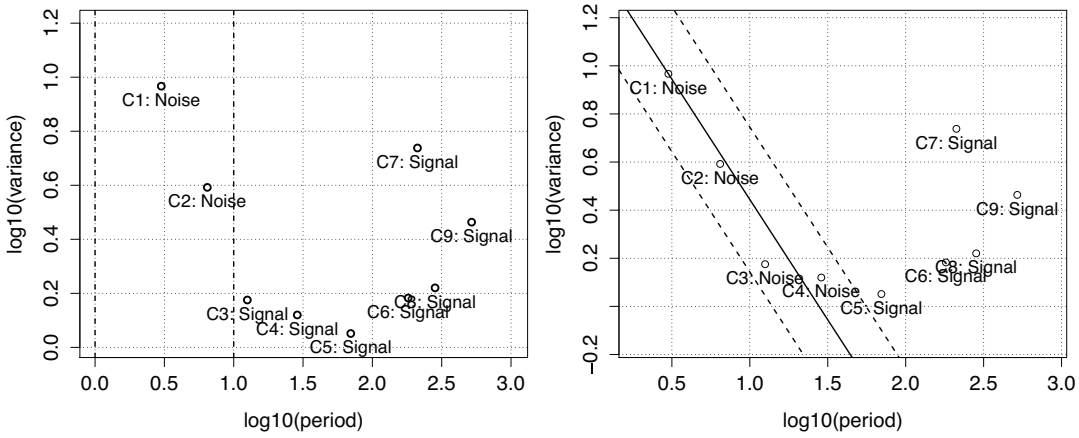


**Figure 7.9: Component selection models** - A component labeled as *noise* implies that it is not distinguishable from a pure white noise series. Therefore, it cannot be learned by the learning algorithm. *Left* - example of the naïve selection model using a threshold filter (DECF), where components with period smaller than 10 weeks (or $\log_{10} period = 1$) are excluded from the learning algorithm. *Right* - example of the component selection model using the Wu and Huang [236] expectation of variance approach to define statistical significant components (DECS).

### 7.4.5 Determining the Embedding Dimension $m$

We propose two methods to determine the dimension $m$ of the delay vectors. In the first approach, we naïvely set the embedding dimension to a fixed size. In the second approach, we use a modified version of a methodology derived in [244] that applies

fractal dimensions to specify dynamically the optimal length of the delay vectors. According to the authors, the fractal dimension $f_L$ of the time series, which gives the intrinsic dimensionality of the embedding vectors in the embedding space, can be used to determine the optimal value of the embedding dimension $m$. In their algorithm, $m$ is incremented between a range $1 \leq m \leq m_{max}$ and $f_L$ is calculated for each space $d$ created. After some value of $m \geq 1$, increasing the embedding dimension does not add any relevant information regarding the state space, which is verified by a flattening in the slope of $f_L$. The turning point, which lies within 95% of the maximum $f_L$, is taken as the optimal $m$. Further details of the algorithm can be found in [244]. Since we have several components, we calculate $m_i$ for each space defined by the components $C_i'$ and the final $m$ is defined as the mean of $m_i$.

## 7.5 Empirical Comparison of Machine Learning Regression Algorithms

In this section, we present some experimental results of applying the least squares, k-NN, decision tree, neural network and support vector machine learning algorithms introduced in Section 6.4 to estimate the function that generates the antimicrobial resistance trends $f(\cdot)$. As mentioned in the previous chapter, we consider only the basic versions of these algorithms. Hundreds of variations are found in the literature but it is not our goal here to validate all of them.

### 7.5.1 Methods

Apart from the least squares method, whose optimal parameters can be computed deterministically, the parameters for all the other methods are tuned using 5-fold cross-validation. The k-NN method is trained for the neighborhood size varying in the range $\{1, 5, 10, 25, 50\}$. Then, the decision tree is pruned with a complexity parameter varying in the range $\{0.0001, 0.001, 0.01, 0.1\}$. Nodes with complexity inferior to the complexity parameter are trimmed. Further, a single-hidden-layer neural network architecture applying the classic logistic sigmoid function $g(u) = 1/(1 + \exp(-u))$ as the node's activation function is trained for the size of the hidden layer $\{1, 3, 10\}$ and for the weight decay parameter $\{0, 0.0001, 0.001, 0.01\}$. Finally, a support vector machine using a linear kernel is tuned for the $C$ parameter of Equation 6.11 for the range

$\{0.001, 0.01, 0.1, 1\}$. We use the $DECF$ model with embedding dimension $m = 6$ applied to the time series presented in Figure 7.1 as our prediction model. Results are reported using the MAE measure for 1 week ahead horizon.

### 7.5.2 Results

Table 7.1 shows the forecasting results of the different machine learning algorithms. Overall, the k-NN method has the smallest prediction error, followed by the support vector machine, least squares, neural network and decision tree methods. Figure 7.10 shows the frequencies that a method outperforms (wins) or is outperformed by other methods (losses). Distinctly, the k-NN algorithm has be best performance with 12 wins and only 4 losses. Therefore, we use the k-NN algorithm to estimate the decision function and provide the forecasting of the antimicrobial resistance rates.

| **Time series** | **LM** | **KNN** | **RT** | **ANN** | **SVM** |
|---|---|---|---|---|---|
| *E. coli*/FEP | 3.49 | 3.42 | **3.28** | 3.47 | 3.48 |
| *K. pneumonia*/AMC | 9.50 | **8.17** | 8.84 | 8.80 | 8.63 |
| *P. aeruginosa*/CIP | **4.31** | 4.44 | 4.82 | 4.50 | 4.40 |
| *S. aureus*/MAC | **5.89** | 6.04 | 6.71 | 6.62 | 6.20 |
| Mean | 5.80 | **5.52** | 5.91 | 5.85 | 5.68 |

**Table 7.1: Forecasting results for the different machine learning methods** - MAE of the 1 week ahead forecasting for the different machine learning algorithms. Results in bold show the best performance. LM: least squares; RT: decision tree.

The power of the k-NN algorithm may be explained mainly by two factors. First, the size of the training data set, despite being large from the epidemiological view point (a decade), it is rather small as machine learning is concerned (between 350 and 519 data points in the leave-one-out cross-validation). Thus, normally best performing algorithms like neural networks and support vector machines fail to efficiently learn the system due to the lack of training examples. On the other hand, simpler algorithms like the least squares and decision trees, that do not require a large training set, have their performance degraded by the lack of linearity and high complexity of the time series models. Therefore, the k-NN algorithm, by providing a compromise between accuracy with complex systems and learning set size, outperforms the other algorithms for the weekly antimicrobial resistance forecasting.

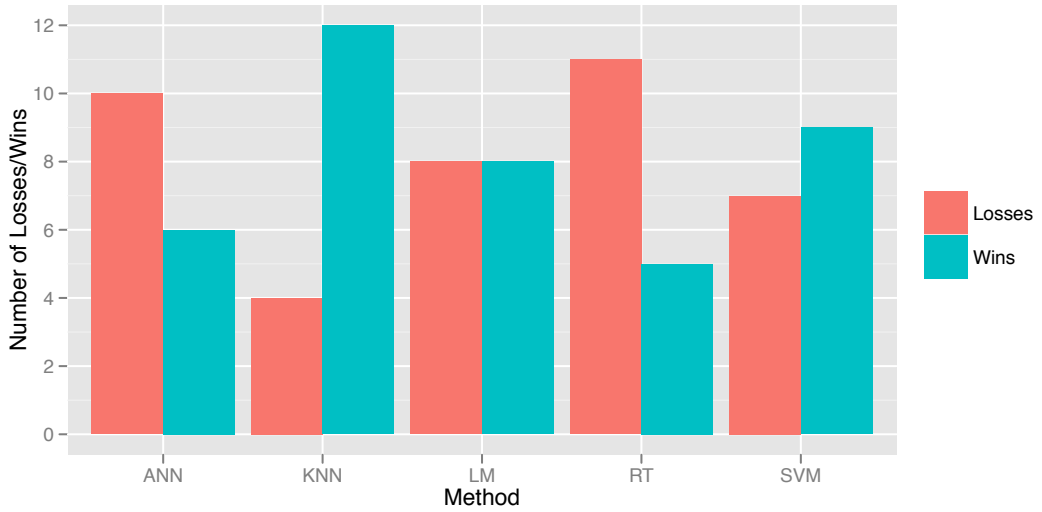


**Figure 7.10: Machine learning algorithm comparison** - Wins: frequency that a method outperforms another method. Losses: frequency that a method is outperformed by another method.

## 7.6 Summary

In this chapter, we present a novel two-stage methodology for the analysis of antimicrobial resistance trends that features trend extraction and forecasting. Both the trend extraction and forecasting methods are fully data-driven, which makes them suitable to work with any data set type. This is important to adapt to the different antimicrobial resistance time series dynamics. We provide some empirical studies to justify our choice for the EMD algorithm for trend extraction and for the k-NN based forecasting algorithm. The EMD algorithm is able to provide good estimation of the intrinsic antimicrobial resistance trends independent of the underlying model and does not require any parameter tuning. Further, the experiments show that the k-NN algorithm provides the best forecasting accuracy compared with the other four machine learning algorithms – least squares, decision tree, neural network and support vector machine – for short-term antimicrobial resistance rates. In the next chapter, we present the results of a larger scale assessment of the methodology introduced here using microbiology data sets.

# 8

# Experimental Results

## 8.1 Introduction

In this chapter, we present the evaluation of the methodology for extraction and forecasting of antimicrobial resistance trends described Chapter 7. We use real weekly antibiotic resistance rates to assess our methods. Qualitative results are presented for the trend extraction method whereas our forecasting methods are compared to baseline machine learning models.

## 8.2 Methods

In this study, we use retrospective antibiotic resistance time series of anonymized and weekly aggregated antibiograms provided by HUG's microbiology laboratory to test our model for extraction and prediction of antibiotic resistance trends applied to short-term variations. Permission to use anonymized population aggregated information was granted through the DebugIT project [118], within which HUG collaborated as a data provider. The data were extracted using our antimicrobial resistance monitoring engine described in Chapter 3 to Chapter 5. Particularly, we have deployed a dedicated version within the HUG intranet (`http://vmmedsup.hcuge.ch:8080/artemis`).

The statistics of the dataset used to train and assess the system are presented in Table 8.1. The training set contains twenty six resistance rate time series of four key pathogens – *Escherichia coli, Klebsiella pneumonia, Pseudomonas aeruginosa* and *Staphylococcus aureus* – tested against a set of antibiotics, selected based on their relevance in susceptibility tests and antibiotherapies. Each time series comprised a

decade of resistance information, containing weekly data from January 1, 2002 through December 31, 2011 in 520 data points. The forecasting algorithm was trained using leave-one-out cross validation with minimum observation set to 350 weeks, resulting in a test set of 170 data points.

### 8.2.1 Performance Measures

The results of the trend extraction method are provided using two use-cases of resistance trend analysis. Due to the lack of a standard and formal definition for trend, there is no benchmark for trend extraction and therefore it is difficult to quantitatively measure trend extraction methods. Then, to demonstrate qualitatively the power of the EMD algorithm, we first correlate components from the resistance time series with components of time series that may be associated with resistance. We take a temperature time series for the Geneva region as an example. Second, we present the statistics on the period of the different components for the time series of the study.

For the machine learning forecaster, we first use time series *EC 4*, *KP 2*, *PA 5* and *SA 5* and the *DECF* model to select the best estimation method of the embedding dimension $m$ and consequently of the dimension $m$. The naïve approach is trained for $m = \{3, 6, 10\}$ and $m_{max}$ is set to 10 in the fractal dimension method. Then, we provide the results for 1, 3 and 12 week-ahead forecasting horizons using the MAE and RMSE cost functions. Since MAE and RMSE measure the deviation between actual and predicted values, the smaller the values of MAE and RMSE the closer the predicted time series is to the true time series. Results of the models *DECA*, *DECF* and *DECS* are compared to a baseline approach based on the random walk method, which is the standard benchmark in machine learning forecasting [245], and to a k-NN regression applied to the raw signal with $m = 6$.

### 8.2.2 Statistical Analysis

We use R version 2.15.0 to decompose the resistance trends, implement the machine learning models and perform the statistical analyses. We apply a two-sided Student's t-test to compare the error of the forecasting models. $P$-values lower than 0.05 are considered significant. Correlation statistics are reported using the Pearson's coefficient of correlation.

**Table 8.1: Weekly resistance rate time series – means and standard deviations (SD) -** Time series of weekly resistance rates defined as the percentage (%) of resistant tests from the total of antibiograms (including intermediate results) for four groups of pathogens – *Escherichia coli*, *Klebsiella pneumonia*, *Pseudomonas aeruginosa* and *Staphylococcus aureus*.

| Id | Organism | Antibiotic | Mean (%) | SD |
|---|---|---|---|---|
| EC 1 | *E. coli* | aminoglycoside | 6.42 | 2.91 |
| EC 2 | *E. coli* | aminopenicillin | 47.22 | 7.39 |
| EC 3 | *E. coli* | amoxicillin-clavulanic acid | 12.96 | 6.24 |
| EC 4 | *E. coli* | cefepime | 6.59 | 5.73 |
| EC 5 | *E. coli* | 3rd generation cephalosporin | 6.92 | 5.75 |
| EC 6 | *E. coli* | fluoroquinolone | 16.76 | 6.25 |
| EC 7 | *E. coli* | trimethoprim-sulfamethoxazole | 27.47 | 6.07 |
| KP 1 | *K. pneumonia* | aminoglycoside | 7.32 | 7.42 |
| KP 2 | *K. pneumonia* | amoxicillin-clavulanic acid | 14.03 | 11.82 |
| KP 3 | *K. pneumonia* | cefepime | 10.80 | 10.59 |
| KP 4 | *K. pneumonia* | 3rd generation cephalosporin | 10.91 | 10.63 |
| KP 5 | *K. pneumonia* | fluoroquinolone | 9.07 | 9.07 |
| KP 6 | *K. pneumonia* | piperacillin-tazobactam | 3.95 | 5.90 |
| KP 7 | *K. pneumonia* | trimethoprim-sulfamethoxazole | 17.26 | 12.18 |
| PA 1 | *P. aeruginosa* | aminoglycoside | 7.11 | 4.99 |
| PA 2 | *P. aeruginosa* | carbapanem | 11.34 | 6.54 |
| PA 3 | *P. aeruginosa* | cefepime | 3.94 | 4.05 |
| PA 4 | *P. aeruginosa* | ceftazidime | 6.52 | 4.97 |
| PA 5 | *P. aeruginosa* | ciprofloxacin | 8.14 | 5.96 |
| PA 6 | *P. aeruginosa* | piperacillin-tazobactam | 6.79 | 9.36 |
| SA 1 | *S. aureus* | aminoglycoside | 33.40 | 13.06 |
| SA 2 | *S. aureus* | benzylpenicillin | 92.20 | 5.32 |
| SA 3 | *S. aureus* | clindamycin | 38.28 | 10.08 |
| SA 4 | *S. aureus* | fluoroquinolone | 39.10 | 12.30 |
| SA 5 | *S. aureus* | macrolide | 41.66 | 10.29 |
| SA 6 | *S. aureus* | trimethoprim-sulfamethoxazole | 1.56 | 1.83 |

## 8.3 Results

In the following sections, we present the results of the trend extraction and forecasting methods for the analysis of timely antibiotic resistance trends. Because this study yielded several hundred results, we provide aggregated statistics and some prominent examples for each of the evaluation dimensions. We start by showing some qualitative analyses using the trend extraction methods, where we apply the EMD technique to extract periodicity of the time series and to correlate resistance trends with external factors likely to be associated with changes in resistance. Then, we present the results of the machine learning forecasting, where we show the performance in terms of MAE and RMSE for the models described previously.

### 8.3.1 Trend Extraction

Figure 8.1 shows the result of the EMD algorithm applied to time series *EC 6* for extracting antibiotic resistance trends. The components C1-7 correspond to the IMFs and describe short-, medium- and long-term periodic trends. The component C8 is the residue of the sifting process and represents the slowly varying mean resistance trend. The raw weekly resistance signal displayed at the bottom right is equivalent to the sum of the 8 components. The first component presents the highest frequency and as the component index increases the frequency also increases. The same pattern is verified for all the other time series and it is inherent to the EMD algorithm. The mean resistance trend is determined empirically and, for *EC 6*, it approximates a sigmoid shape, which has also been verified in other studies of resistance evolution [230]. If the dynamics of the resistance model of the time series *EC 6* is indeed sigmoidal, the resistance has reached its equilibrium point. Then, using component C8 it becomes trivial to detect the point of stabilization, which happens to be around week 380 in the example. Considering that resistance has started to increase around week 80, it took 5.8 years to reach the stabilization maximum, which is similar to the raise of penicillin-resistant pneumococcal verified in other studies [230]. The same cannot be directly inferred from the raw signal.

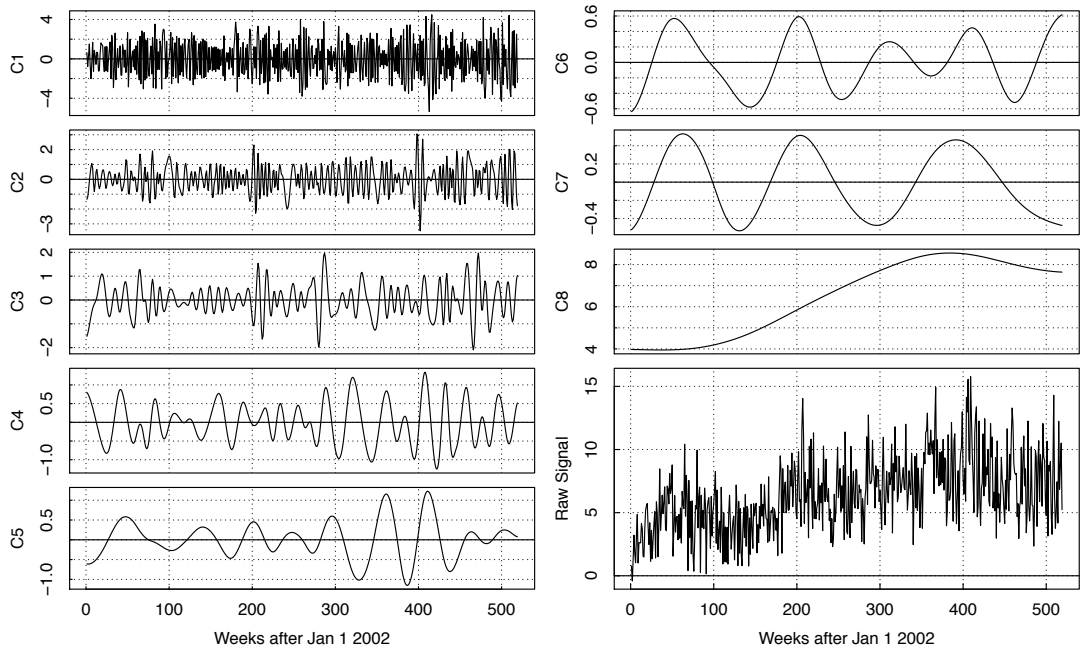


**Figure 8.1: Result of the EMD technique applied to trend extraction** - In the example, EMD is used to decompose the *E. coli* vs. aminoglycoside resistance time series. C1 and C2 components describe short-term resistance trends; C3 and C4 components describe medium-term trends; and C5 to C8 components describe long-term trends. The last component (C8) provides the underlying mean trend of the resistance signal. The raw signal displayed at the bottom right is equivalent to the sum of the eight components.

#### 8.3.1.1 Association with Factors that Influence the Development of Resistance

Figure 8.1 presents very distinct patterns for some of the periodic components, especially C4-7. To illustrate their correlation with external factors that may be associated with resistance evolution, we employ a monthly temperature time series of the Geneva region for the same study period. Then, we associate its components with monthly components of the time series *EC 2* and *SA 5*. Applying the EMD algorithm to the monthly temperature and to the resistance time series yields 4 and 5 IMFs respectively, from which components C3 of the temperature and C4 of resistance are displayed in Figure 8.2. The components C4 of the resistance time series show high negative correlation with C3 component of the temperature time series ($\rho = -0.73$ and $\rho = -0.71$ for EC 2 and SA 5 respectively). In this case, temperature might not to be the cause of changes in resistance, that is, there is no causality implied in the correlation. Nevertheless, both might be influenced by a common denominator, the weather. In the case of resistance, the different seasons affect the incidence of infections, which changes the dynamics of antibiotic consumption [246]. It is also fair to assume from Figure 8.2 that a variation in the component C3 of the temperature time series is likely to be followed by a proportional change in the resistance of *EC 2* and *SA 5* time series some time in the future. These data could be used, for instance, to enhance multivariate resistance analysis models.

#### 8.3.1.2 Period

Figure 8.3 shows for the 4 groups of time series in Table 8.1 the central period of oscillation in weeks of the first 6 components found by counting the zero-crosses. Components C1 and C2 have the smallest periods and provide information on short-term trends (period of 3.0, SD 0.2 weeks and 6.6, SD 0.7 weeks respectively) whereas C3 and C4 represent medium-term variations in the resistance trend. The period of component C3 is around 3 months (13.7, SD 1.3 weeks), which could be related, for instance, to antibiotic cycling, as in the 3 months cycle experiment done in [247] to decrease bacterial antibiotic resistance. C4 has a period slightly longer than 6 months (29.2, SD 4.9 weeks), which could be related to seasonal changes due to weather factors, such as temperature, precipitation, etc. Component C5 has period around 1 year, which could

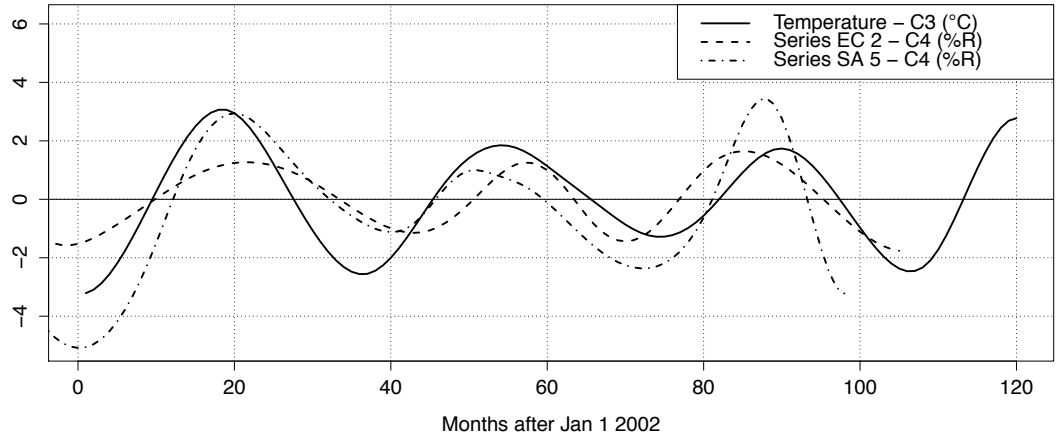


**Figure 8.2: Correlation between temperature and resistance IMF components** - Component C3 of temperature (period of 34.7 months) compared with component C4 of *E. coli* vs. aminopenicillin and *S. aureus* vs. macrolide resistance time series (period of 34.7 and 35.0 months respectively). Resistance component is back-shifted half a period to account for the negative correlation. Temperature source: Federal Office of Meteorology and Climatology MeteoSwiss (`http://www.meteosuisse.admin.ch/web/en.html`).

be for example related to warm and cold seasons, or more precisely, to high winter peaks of antibiotic use as verified in the study presented in [246]. Finally, component C6 (period between 2 and 4 years) represent long-term trends, which might be associated to interventions within the healthcare institutions and in the community, or to the actual evolutionary process of bacteria, such as the natural competition between resistant and susceptible strains [248].

### 8.3.2 Forecasting

#### 8.3.2.1 Embedding Dimension

There were no statistical differences between the forecasts using any of the naïve methods and the method based on the fractal dimension $f_L$ in the experiments with time series *EC 4*, *KP 2*, *PA 5* and *SA 5* for 1, 3 and 12 forecasting horizons. Nevertheless, similarly to the results obtained in [244], the dynamic method was able to adapt to the different time series and compute a well performing $m$, while keeping it small enough so as not to degrade the training and testing time. It resulted in an overall MAE of 5.57%, being the lowest MAE in 4 out of 12 tests – a result equivalent to the best naïve

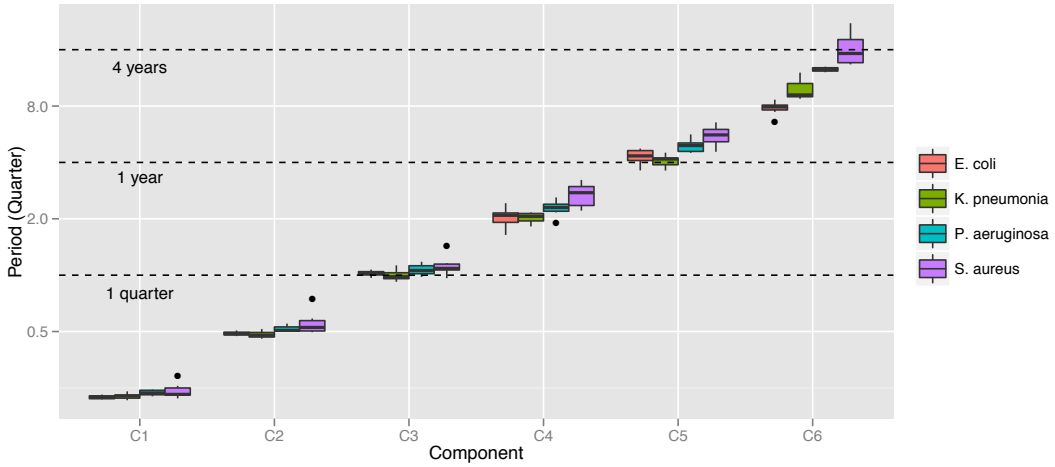


**Figure 8.3: Oscillating period in quarters of the decomposed time series components stratified into pathogen groups** - C1 and C2 components describe short-term variations in the resistance rates (median period $< 7$ weeks); C3 and C4 components describe medium-term variations (median period between 3 and 12 months); and C5 and C6 components describe long-term changes (median period $> 1$ year). Notice how the period of the resistance trends are tightly associated to meaningful calendar cycles.

method ($m = 6$). Thus, we employed the fractal dimension method in the *DECA*, *DECF* and *DECS* models to determine the size of the optimal embedding dimension $m$.

#### 8.3.2.2 Forecasting Models

Table 8.2 provides the MAE and RMSE measures of the one week-ahead forecasts generated by the various models for the datasets not used to train the embedding dimension $m$. Overall, the models that employ decomposition of the time series and filter out noisy components, that is, *DECF* and *DECS*, improve significantly the forecast over the simpler models for both error measures ($P \leq .001$). They have the smallest prediction errors, outperforming the other methods for all but time series *EC 7* and *KP 1*. The *DECF* model performs slightly better than the *DECS* model when we consider the MAE values. Conversely, the *DECS* model outperforms the *DECF* model if we consider the RMSE values. However, their difference in forecasting accuracy is not statistically significant ($P = .78$).

**Table 8.2: Performance of the forecasting methods -** Error for one week-ahead forecasting – mean absolute (MAE) and root mean squared (RMSE) errors. Results for the best forecasting performance are displayed in bold.

| | | **Time series** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Error** | **Method** | EC 1 | EC 2 | EC 3 | EC 5 | EC 6 | EC 7 | PA 1 | PA 2 | PA 3 | PA 4 | PA 6 |
| | RW | 3.15 | 5.78 | 4.42 | 4.56 | 5.19 | 5.64 | 4.62 | 6.13 | 3.46 | 4.10 | 5.46 |
| | KNN | 2.52 | 4.74 | 4.00 | 3.72 | 4.06 | 4.66 | 3.95 | 4.86 | 3.01 | 3.70 | 4.59 |
| MAE | DECA | 2.55 | 4.74 | 3.65 | 3.56 | 4.09 | 4.46 | 3.98 | 4.87 | 3.05 | 3.81 | 4.67 |
| | DECF | 2.51 | **4.50** | 3.64 | 3.40 | **3.93** | **4.36** | 3.86 | 4.79 | **2.95** | **3.38** | **4.27** |
| | DECS | **2.47** | **4.50** | **3.60** | **3.35** | **3.93** | 4.37 | **3.83** | **4.69** | 3.03 | 3.40 | 4.30 |
| | RW | 3.79 | 7.35 | 5.64 | 5.76 | 6.62 | 7.07 | 6.02 | 7.71 | 4.67 | 5.49 | 7.07 |
| | KNN | 3.07 | 5.85 | 4.93 | 4.79 | 5.36 | 5.70 | 4.85 | 6.24 | 3.60 | 4.47 | 5.76 |
| RMSE | DECA | 3.03 | 5.81 | 4.71 | 4.68 | 5.40 | 5.41 | 5.01 | 6.30 | 3.80 | 4.69 | 6.00 |
| | DECF | 2.95 | **5.47** | 4.59 | 4.38 | **5.19** | **5.29** | 4.83 | 6.16 | **3.52** | **4.24** | **5.48** |
| | DECS | **2.92** | 5.48 | **4.55** | **4.34** | **5.19** | **5.29** | **4.80** | **6.08** | 3.57 | **4.24** | **5.48** |

| | | **Time series** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Error** | **Method** | KP 1 | KP 3 | KP 4 | KP 5 | KP 6 | KP 7 | SA 1 | SA 2 | SA 3 | SA 4 | SA 6 |
| | RW | 6.08 | 8.83 | 8.85 | 8.18 | 5.69 | 10.97 | 6.51 | 4.38 | 7.28 | 7.44 | 1.40 |
| | KNN | 5.14 | 7.41 | 7.46 | 6.69 | **5.35** | 8.89 | 5.82 | 3.88 | 6.50 | 6.86 | 1.27 |
| MAE | DECA | **5.13** | 8.01 | 7.94 | 6.93 | 5.92 | 9.11 | 5.66 | 3.97 | 6.49 | 6.70 | **1.20** |
| | DECF | 5.52 | **7.26** | **7.31** | **6.33** | 5.58 | **8.26** | 5.19 | **3.87** | 6.09 | **6.11** | 1.22 |
| | DECS | 5.50 | 7.37 | 7.40 | 6.36 | 5.49 | 8.31 | **5.10** | 3.89 | **6.05** | 6.24 | 1.23 |
| | RW | 7.89 | 11.66 | 11.70 | 9.98 | 7.82 | 13.85 | 8.28 | 5.73 | 9.22 | 9.53 | 2.08 |
| | KNN | **6.26** | 9.74 | 9.78 | 8.30 | **7.11** | 11.05 | 7.31 | 4.94 | 8.24 | 8.55 | 1.53 |
| RMSE | DECA | 6.54 | 10.50 | 10.43 | 8.48 | 7.76 | 11.24 | 7.25 | 5.15 | 8.24 | 8.51 | 1.49 |
| | DECF | 6.70 | **9.50** | **9.60** | 7.99 | 7.26 | 10.30 | 6.57 | **4.86** | 7.73 | **7.71** | 1.49 |
| | DECS | 6.71 | 9.70 | 9.64 | **7.96** | 7.22 | **10.26** | **6.53** | 4.89 | **7.69** | 7.80 | **1.48** |

Figure 8.4 displays the result of a statistical significance comparison amongst the five forecasting models for the three forecasting horizons using the MAE values. The frequency that a given model significantly outperforms other models (wins) is shown in green whereas the frequency that the same model is outperformed by other models (losses) is shown in red. As we can see, the *DECS* and *DECF* models have the best performance overall, that is, they have the highest number of wins (*DECS*: 77 wins, *DECF*: 76 wins) and smallest number of losses (4 for both). As the horizon increases it becomes more evident the superiority of the models that use decomposition. Particularly, at the 12 week-ahead horizon all of them outperform the *RW* and *KNN* models. All the k-NN based models improve upon the *RW* baseline model.

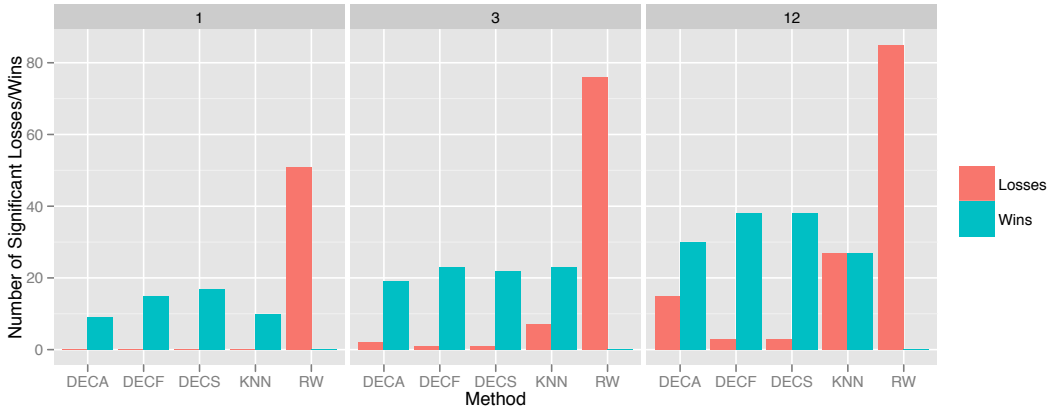


**Figure 8.4: Statistical comparison of the different forecasting methods using the mean absolute error (MAE) for the 1, 3 and 12 step-ahead horizons.** - *Wins*: frequency that a method significantly outperforms other methods. *Losses*: frequency that a method is significantly outperformed by other methods.

Figure 8.5 shows a representative result of the machine learning forecaster and its respective residual errors for the *EC 7* time series in the test phase of the leave-one-out cross-validation. In the top panel, the actual resistance rate is displayed in black whereas the one week-ahead forecasts for the *RW*, *KNN*, *DECA*, *DECF* and *DECS* models are shown in red, green, dark blue, light blue and purple, respectively. The forecasts of the baseline model follow the signal but are always lagged by one week. Thus, this model presents the largest absolute residuals, caused especially by the zigzag variations in the resistance time series. The *KNN* and *DECA* models, despite using the full signal spectrum, are not able to capture high frequency changes either. They

forecast medium-term trends but without accuracy. Finally, the *DECF* and *DECS* models learn essentially the underlying mean trend.

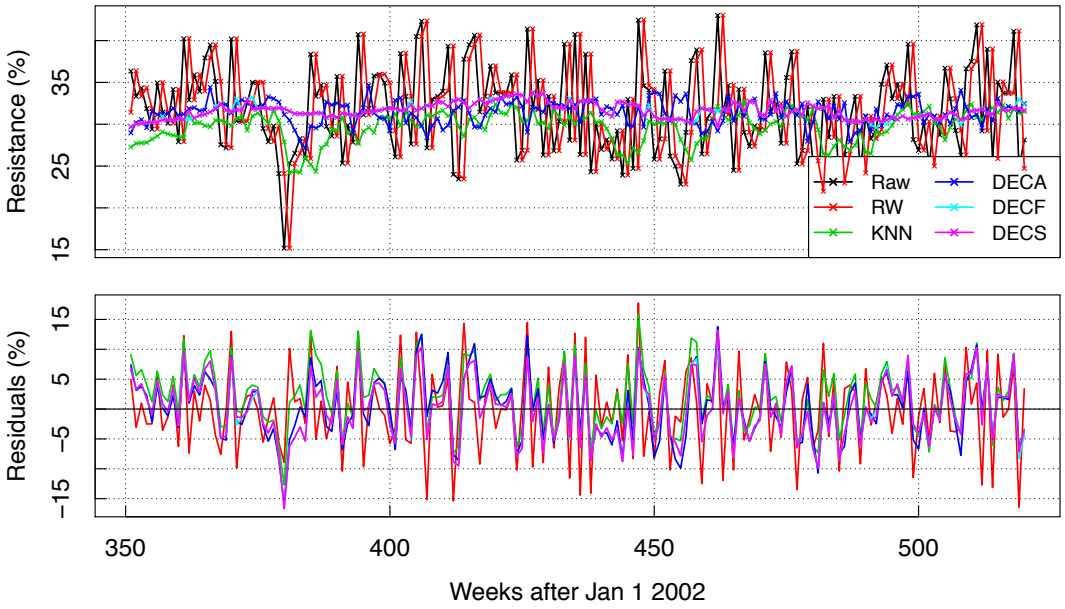


**Figure 8.5: Forecasting and residuals.** - *Top:* one week-ahead forecasting results of the times series *EC 7 – E. coli* vs. trimethoprim-sulfamethoxazole. *Bottom:* Respective forecasting residuals.

Finally, Figure 8.6 presents the residual errors of the *EC 3*, *KP 7*, *PA 6* and *SA 1* time series for 1, 3 and 12 forecasting horizons using the same color schema of Figure 8.5. The row panels correspond to forecasting horizons and the column panels correspond to the time series. Similarly to Figure 8.5, given that each of the models uses different forecasting algorithms, none of them is able to forecast the high-frequency trend components. The short-term spikes are present in all the results as we walk along the columns. The residual errors do not increase significantly with the forecasting horizon, being of the same order of magnitude for the three forecasting horizons. Further, the column residuals are highly correlated amongst the different forecasting horizons ($\rho \geq 0.90$ for the models that use decomposition).

## 8.4 Discussion

In this chapter, we present the results of our two-stage model for analyzes of antibiotic resistance data applied to up-to-date and short-term trends. Our model was validated in a large scale dataset spanning a decade of weekly aggregated resistance time series. The use of the EMD algorithm for trend extraction provided effective results not only to obtain the resistance trends, but also to provide insight into the periodicity of the resistance trends and into the level of correlation with external variables. The machine learning forecasting model based on the k-nearest embedding vectors produced results with good accuracy, statistically outperforming the random walk baseline approach. The decomposition of the raw signal and exclusion of the noisy components were effective in reducing the forecasting error. Finally, both trend extraction and forecasting methods proved to be robust, adapting to time series with different resistance dynamics.

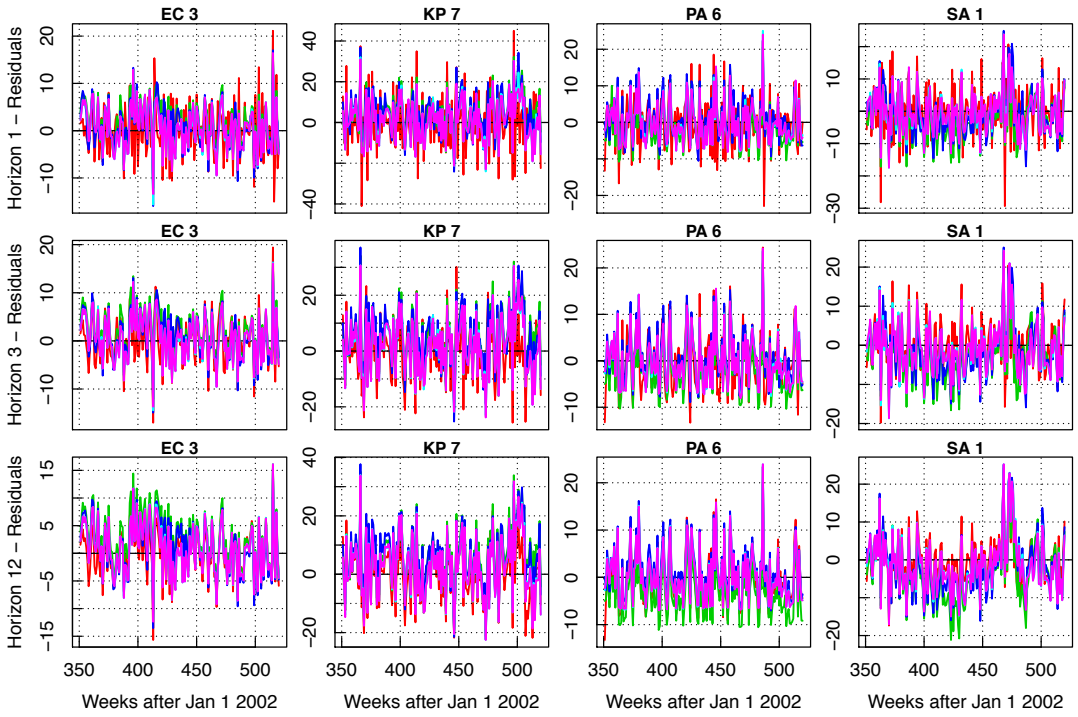


**Figure 8.6: Forecasting residuals for 1, 3 and 12 week-ahead horizons.** - Residuals of four representative time series of each pathogen group – *EC 3*: *E. coli* vs. amoxicillin-clavulanic acid; *KP 7*: *K. pneumonia* vs. trimethoprim-sulfamethoxazole; *PA 6*: *P. aeruginosa* vs. piperacillin-tazobactam; *SA 1*: *S. aureus* vs. aminoglycoside. *RW*: red; *KNN*: green; *DECA*: dark blue; *DECF*: light blue; *DECS*: purple.

We focused on short-term trends because they are able to capture more efficiently the resistance dynamics within a given clinical setting. Especially in cases of resistance outbreaks, monthly and yearly resistance trends cannot spot readily changes in the mean rates. For example, in the vancomycin-resistance *Enterococcus* outbreak experienced at Princess Alexandra Hospital, Brisbane, Australia in 1999, in approximately 10 weeks the number of prevalence cases increased 14 fold, from 1 case per week to 14 cases, even if an abnormal prevalence rate had already been detected in the first week of the outbreak [193]. Thus, effective biosurveillance systems should be based on up-to-date trend analysis methods to avoid further spreading of resistance strains.

### 8.4.1 Trend Extraction

We have explored the EMD algorithm to extract antibiotic resistance trends from weekly aggregated time series. Traditionally, epidemiologists and infectiologists use monthly and yearly resistance data and statistical tests to assess resistance trends [229]. Compared to that approach, the methodology introduced here provides more insight into the dynamics of resistance than the simple detection of upward/downward trends. It is able to extract medium- and long-term but also short-term variations in the resistance rate through the IMF components, which are neglected in the trend detection analysis. The lower frequency components could be used, for instance, in biosurveillance systems as an early warning of emerging resistance. Moreover, infectious disease specialists may be able to determine periodicity and cycles within resistance trends using the decomposed components, adopting infection control interventions accordingly. Thus, the EMD methodology may serve as a complementary tool for the analysis of short-term antibiotic resistance trends.

The components extracted using EMD technique could be also used to correlate resistance evolution to variations in other clinical, societal and environmental factors associated with antibiotic resistance, such as duration of treatment, infection control measures, antibiotic consumption and weather [241]. The EMD technique does not assume any a priori model for the data and thus is suitable for extracting trends from any time varying system, independent of the system's underlying model. Moreover, since the method is fully data-driven, the components are likely to represent physically meaningful events in the resistance process [249, 236]. Sometimes, these events might not be obvious when considering only the raw signal, especially in the case of factors

influencing resistance rates in opposite directions, as verified for example around week 300 of Figure 8.1, where for C4 and C7 components the wave form is negative and for C5 and C6 it is positive. To further illustrate that, imagine that in a given clinical setting resistance has increased by 1% due to antibiotic misuse and decreased by 1% due to better hand hygiene practices, resulting in no change in the raw signal. Hence, the EMD technique could be applied to spot such events, which may appear in details in the signal components, following similar patterns as observed in the external factors.

### 8.4.2 Resistance Forecasting

We have developed a novel machine learning method to forecast antibiotic resistance trends based on the k-nearest embedding vectors. Our method showed good forecasting accuracy for short-term trends, outperforming baseline machine learning benchmarks but also other enhanced methods, such as the k-NN applied to the raw signal. Our method is supported by the delay coordinate embedding theorem, a technique derived from the studies of chaos to model deterministic nonlinear time series, and by the k-NN framework to project observed resistance events in an embedding space into the future dimension. From our experiments, we notice that decomposing the raw signal to enhance the features of the training data and excluding high frequency components from the learning set improves the performance of the forecaster. It reinforces our hypothesis that some components of the resistance signal are derived from a pure random process. Hence, they cannot be learned by and degrade the quality of the learning algorithm. Therefore, they should be filtered out from the antibiotic resistance model.

The machine learning model based on the k-nearest embedding vectors could be used to improve clinical decision support systems for antibiotic prescription, giving more accurate information on the current resistance dynamics than the latest resistance statistics when there are delays of a week or more in the resistance numbers. As shown in Figure 8.5, the forecasts provided by the naïve method, which was used as the baseline benchmark, are delayed by one week (notice the one-step forward shift between the red and black lines). As such, they are equivalent to the latest resistance data points, or $\hat{x}_{n+1} = x_n$, which are obtained in phenotype-based antibiograms, in the best case, from samples extracted two or three days in the past. Since the models that use decomposition significantly improves upon the naïve method, by consequence,

they also provide better evidence to empirical therapy than methods based at the latest resistance rate information when actual results are delayed of at least one data point.

### 8.4.3 Limitations

This study was limited to time series of pathogens that present some level of resistance to the respective antibiotics. Sequences showing bursting patterns, as verified at the beginning of the resistance development process, were not tested and from the forecasting results it is unlikely that our model will be able to forecast bursts. Moreover, we have not investigated the effect of irregular time series, that is, those that contain null values, in our model. The time series of the study are of bacteria with high prevalence rate, having least one positive culture followed by an antibiogram per week. Finally, the use of statistical tests to compare the classifiers has a limited value because the results were derived from the same (overlapping) training data and therefore they were not independent. Nevertheless, the two classifiers based on signal decomposition systematically outperformed the naïve and pure k-NN methods, providing enough evidence on their superiority.

## 8.5 Conclusion

In this chapter, we present the results of our methodology for analysis of antimicrobial resistance trends using a large time series data set. The results show that the decomposition of the raw signal not only helped to improve the baseline forecasting method, but also added valuable insight into the dynamics of the resistance time series. Especially if ward specific data are employed, our fully automated method could be potentially applied in outbreak detection and biosurveillance models. Furthermore, it could be integrated into clinical decision support systems dedicated to improve the accuracy of empirical antibiotic therapies. Moreover, since the methodology does not assume any underlying model for the data set, it could be generalized to other time varying clinical events. Future research may aim at investigating the correlation of other clinical factors, such as antibiotic consumption and hand hygiene compliance, with the decomposed resistance trends to confirm the physical meaning of the signal components. Further, the methodology presented here could be combined with burst detection models to improve the forecasting accuracy.

# Part III

# Conclusion

# 9

# Conclusions and Future Work

In recent decades, surveillance of antimicrobial resistance has been performed using reporting and manual or semi-automatic procedures with yearly compiled data sets. However, with the rapid increase of resistance amongst many pathogens, this paradigm needs to be revised if we do not want to go back to the pre-antibiotic era. Recent outbreaks of resistant bacteria, such as *Escherichia coli*, *Enterococcus spp.* and *Staphylococcus aureus* [38, 193, 8], provide examples of why resistance rates must be monitored closely and in a larger scale to avoid further spread of resistant strains. In this scenario, there is an urgent need for better tools to access, interoperate, aggregate and analyze resistance trends to be used in biosurveillance systems.

In this thesis, we propose novel data- and knowledge-driven methods to monitor and analyze antimicrobial resistance evolution using up-to-date microbiology data from inter-institutional databases. The main contributions of the thesis include (*i*) a knowledge-aware framework for online large-scale data sharing and monitoring of antimicrobial resistance, and (*ii*) a data-driven method to extract and forecast resistance trends. In particular, we studied the use of Semantic Web technologies to enable real-time integration and interoperability of heterogeneous and transnational microbiology data sources. Our experiments resulted in a novel architecture that can be used in the development of eHealth networks to share real-time resistance data in a cross-institutional environment. Moreover, we researched new models to analyze up-to-date antimicrobial resistance trends using empirical mode decomposition and machine learning. To the best of our knowledge, our work is the first to apply trend decomposition and learning algorithms to forecast antimicrobial resistance rates. The proposed

model provides further insight into resistance trend analyses and enhances the prediction accuracy when compared to baseline machine learning approaches. Particularly, our forecasting models outperformed the random walk and pure k-NN models.

The overall work developed in this thesis can be seen through two different perspectives. First, from the healthcare viewpoint, it can be regarded as a transnational platform for biosurveillance, providing new knowledge-aware tools for gathering real-time data from distributed clinical systems and to foster intelligent analyses of the material. Then, from the informatics viewpoint, our work can be thought of as a data mining framework, where we developed advanced data-intensive methods for accessing, collecting and processing information to finally produce new knowledge from large-scale distributed and semantically-rich data sets.

## 9.1 Management of Distributed Microbiology Data and Sources

In the first part of this thesis, we explored and developed the use of Semantic Web technologies to integrate and interoperate heterogeneous microbiology databases applied to the development of a transnational antimicrobial resistance monitoring system. In contrast to existing methods, our approach focuses on providing automatic real-time cross-institutional access to microbiology data to improve tracking variations and evolution of pathogen resistance rates.

Our method for data management deals with heterogeneous and distributed data and sources using semantic technologies at the technical, syntactic and semantic levels. In Chapter 3, we developed a model to formalize microbiology databases and provide a common communication protocol (SPARQL) and message exchanging format (RDF), creating thus a single formal technical layer to access local microbiology data sources. Then, in Chapter 4 we presented our methodology to integrate the distributed semantic endpoints, built on the work developed in Chapter 3. At the knowledge level, our solution to the problem of heterogeneous semantics was a hybrid-ontology approach, where local ontologies that define and formalize the microbiology data sources, subscribed to a global common ontology via ontology mappings. Standard terminologies, such as SNOMED CT and UniProt/NEWT, were used to define the basic syntax of the domain. They served as proxy between the local syntax and the global concepts

defined in the domain ontology. At the query engine level, we developed a template-based mediator that represented the clinical queries of the user-interface for each local endpoint.

Our transnational monitoring architecture was clinically evaluated in Chapter 5. Results showed that the use of a push-down approach in the distributed query engine considerably reduced the querying time when compared to central reasoning, making our integration model suitable for time-constraint operational environments, such as those experienced by physicians. Furthermore, our real-time monitoring approach produced equivalent resistance results to existing yearly batch-based biosurveillance systems, such as EARS-Net and SEARCH, proving that our model could be used for online transnational resistance monitoring, a step-ahead of existing antimicrobial resistance monitoring systems. Finally, the user evaluation showed that the monitoring system developed in Chapter 4 has practical applications, being useful, for example, to infection control specialists for tracking resistance trends and the emergence of new resistant pathogens. Therefore, the novel methodology proposed in the thesis based on knowledge-intensive technologies advances the state-of-the-art by enabling the integration and interoperability of existing heterogeneous microbiology databases, fostering the development of more effective transnational antimicrobial resistance monitoring systems with enhanced time-constraint capabilities.

## 9.2 Analysis of Antimicrobial Resistance Data

In the second part of the thesis, we researched new data-driven methods for analyses of antimicrobial resistance time series. Our work resulted in a novel model based on decomposition of the resistance signals and a learning algorithm to extract and forecast resistance trends. Compared to existing methodology of antimicrobial resistance analysis, such as trend detection, our model works more effectively with short-term resistance trends, identifying different oscillation modes of resistance evolution. Moreover, our forecasting algorithm outperforms baseline machine learning approaches, providing improved prediction accuracy and thus better evidence of future resistance rates.

The methods for antimicrobial resistance trend analysis were developed in Chapter 7. Then, they were evaluated in Chapter 8 on a large time series data set extracted using the monitoring system developed in the first part of the thesis (Chapter 3 to Chapter

5). Our analysis model uses the empirical mode decomposition algorithm to decompose resistance signals and extract periodic trends. It improves upon existing methods for analysis of resistance trends, such as long-term trend extraction and detection, by exposing peculiarities not easily verified from the raw signal, such as short-term trends and periodicity. These processed signals are then used to feed our machine learning algorithm based on the k-NN to forecast resistance rates. The model applies the delay coordinate embedding theorem to reconstruct the state-space and, together with k-NN, it projects past similar events in the future dimension, creating thus the resistance forecasts. We evaluated several machine learning methods and, from our experiments, the k-NN algorithm showed the best forecasting performance. The models that use decomposition of the resistance signal and filtering out of noisy components showed better forecasting accuracy when compared to models that use the full signal spectrum, improving the baseline forecasting method but also the k-NN algorithm applied to the raw signal. Finally, our model is suitable for different pathogen/antimicrobial time series but also to other time-varying clinical processes because both the trend extraction and the forecasting methodologies are model free. Therefore, they can adapt to time series data sets representing systems of different dynamics.

The models for trend extraction and forecasting of resistance trends developed in this thesis have several potential practical applications. First, they can be used in the analysis of resistance trends to identify association with factors that influence resistance, such as antimicrobial consumption. Further, they can be applied in clinical decision support systems to guide empirical treatment. As shown in Chapter 8, our model is able to depict more faithfully the actual resistance status in a given clinical setting than resistance rate statistics delayed by one week or more, serving thus as a better source of evidence for antibiotherapy advising. Finally, due to their ability to work with short-term trends and the enhanced forecasting accuracy, these models can be used in resistance outbreak detection systems to provide reference levels to distinguish between endemic and pandemic resistance.

## 9.3 Future Work

We see several opportunities for further research built on the work developed here. First, the distributed query engine could be improved to reduce global maintenance.

The query engine works with query templates, where local source queries are represented centrally in the query mediator. This reduces the overhead on local sources at the expense of a more complex central engine. An optimal engine would push all local representations to the local sources, improving the maintenance of the global system. Recent work of Hoehndorf *et al.* [250] provides some research directions to solve this issue, pointing to biomedical ontologies to represent local data sources and combining classes from multiple ontologies with upper-level ontologies and expressive relations. Second, the monitoring interface could be more flexible to improve query power. Currently, the users are presented with pre-defined classes, which restrict the query expressivity. The system could allow users to group information according to their need by enabling, for example, classes to be defined interactively at the query time. A similar approach could be studied for logical operations within the query templates as recently investigated in the works of Shaw *et al.* [251]. Finally, the association of factors that influence resistance could be further exploited using, for example, operational data of antibiotic consumption. The perspective of looking at the resistance signal through the decomposed waves opens more subjects to research. Several works have attempted to associate factors that are believed to have influence on the mutation and selection of resistant pathogens. However, sometimes these works are inconclusive. For instance, in [252] the authors found that prescription rate of some antibiotics presented correlation with bacterial resistance while for other antibiotics, such as cephalosporin use, there were no correlations. A fair assumption is that many other factors affect resistance but they are not all captured within the resistance rate time series, the so called unseen variables. However, it might be also that, due to the concurrent effects over the time series, the variations are hidden. By breaking down the signal and making explicit the periodicity of the different signal components, the model investigated in Chapter 8 might help to investigate such cases.

# References


[1] D'Costa, V. M., King, C. E., Kalan, L., Morar, M., Sung, W. W. L., Schwarz, C., Froese, D., Zazula, G., Calmels, F., Debruyne, R., Golding, G. B., Poinar, H. N., and Wright, G. D.: Antibiotic resistance is ancient. *Nature* 477, 457–61 (2011) 1

[2] Anonymous: The bacterial challenge: time to react. Tech. rep., European Center for Disease Prevention and Control (2009) 1, 2

[3] Anonymous: Guía para el Tratamiento de las Enfermedades Infecciosas. Tech. Rep. OPS/DPC/CD/296/2004, Organización Panamericana de la Salud, Washington DC (2004) 1

[4] Anonymous: Report on infectious diseases: Removing obstacles to Healthy Development. Tech. rep., World Health Organization (1999) 1, 2

[5] Cornaglia, G., Hryniewicz, W., Jarlier, V., Kahlmeter, G., Mittermayer, H., Stratchounski, L., Baquero, F., and ESCMID Study Group for Antimicrobial Resistance Surveillance: European recommendations for antimicrobial resistance surveillance. *Clin Microbiol Infect* 10, 349–83 (2004) 2, 3

[6] Anonymous: *WHO global strategy for containment of antimicrobial resistance.* World Health Organization, Department of Communicable Disease Surveillance and Response (2001) 2

[7] Leung, E., Weil, D. E., Raviglione, M., Nakatani, H., and World Health Organization World Health Day Antimicrobial Resistance Technical Working Group: The WHO policy package to combat antimicrobial resistance. *Bull World Health Organ* 89, 390–2 (2011) 2, 3

[8] Klevens, R. M., Morrison, M. A., Nadle, J., Petit, S., Gershman, K., Ray, S., Harrison, L. H., Lynfield, R., Dumyati, G., Townes, J. M., Craig, A. S., Zell, E. R., Fosheim, G. E., McDougal, L. K., Carey, R. B., Fridkin, S. K., and Active Bacterial Core surveillance (ABCs) MRSA Investigators: Invasive methicillin-resistant Staphylococcus aureus infections in the United States. *JAMA* 298, 1763–71 (2007) 3, 165

[9] Infectious Diseases Society of America (IDSA), Spellberg, B., Blaser, M., Guidos, R. J., Boucher, H. W., Bradley, J. S., Eisenstein, B. I., Gerding, D., Lynfield, R., Reller, L. B., Rex, J., Schwartz, D., Septimus, E., Tenover, F. C., and Gilbert, D. N.: Combating antimicrobial resistance: policy recommendations to save lives. *Clin Infect Dis* 52 Suppl 5, S397–428 (2011) 3

[10] Anonymous: A Public Health Action Plan to Combat Antimicrobial Resistance. Tech. rep., The Centers for Disease Control and Prevention (CDC), the Food and Drug Administration (FDA) and the National Institutes of Health (NIH) (2011) 3

[11] O'Brien, T. and Stelling, J.: *Antimicrobial Resistance in Bacteria*. Taylor & Francis (2007) 3

[12] Daneman, N., Low, D. E., McGeer, A., Green, K. A., and Fisman, D. N.: At the threshold: defining clinically meaningful resistance thresholds for antibiotic choice in community-acquired pneumonia. *Clin Infect Dis* 46, 1131–8 (2008) 3

[13] O'Brien, T. F. and Stelling, J.: Integrated Multilevel Surveillance of the World's Infecting Microbes and Their Resistance to Antimicrobial Agents. *Clin Microbiol Rev* 24, 281–95 (2011) 4, 65, 66, 74

[14] Pasche, E., Teodoro, D., Gobeill, J., Ruch, P., and Lovis, C.: QA-driven guidelines generation for bacteriotherapy. *AMIA Annu Symp Proc* 2009, 509–13 (2009) 4, 107

[15] Pasche, E., Gobeill, J., Teodoro, D., Vishnyakova, D., Gaudinat, A., Ruch, P., and Lovis, C.: Using multimodal mining to drive clinical guidelines development. *Stud Health Technol Inform* 169, 477–81 (2011) 4

[16] Spellberg, B., Guidos, R., Gilbert, D., Bradley, J., Boucher, H. W., Scheld, W. M., Bartlett, J. G., Edwards, J., Jr, and Infectious Diseases Society of America: The epidemic of antibiotic-resistant infections: a call to action for the medical community from the Infectious Diseases Society of America. *Clin Infect Dis* 46, 155–64 (2008) 5

[17] Wernli, D., Haustein, T., Conly, J., Carmeli, Y., Kickbusch, I., and Harbarth, S.: A call for action: the application of The International Health Regulations to the global threat of antimicrobial resistance. *PLoS Med* 8, e1001022 (2011) 5, 13

[18] Lee, T., Hendler, J., and Lassila, O.: The Semantic Web. *Scientific American* 284, 34–43 (2001) 5, 21

[19] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., OBI Consortium, Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R. H., Shah, N., Whetzel, P. L., and Lewis, S.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol* 25, 1251–5 (2007) 6, 18, 28

[20] Rubin, D. L., Shah, N. H., and Noy, N. F.: Biomedical ontologies: a functional perspective. *Brief Bioinform* 9, 75–90 (2008) 6

[21] Murphy, S. P. and Burkom, H.: Recombinant temporal aberration detection algorithms for enhanced biosurveillance. *J Am Med Inform Assoc* 15, 77–86 (2008) 7

[22] Weigend, A. and Gershenfeld, N.: The future of time series. In *Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis*, 1–70 (1992) 7, 111

[23] Box, G. E. P. and Jenkins, G. M.: *Time series analysis: forecasting and control*. Holden-Day series in time series analysis and digital processing. Holden-Day, San Francisco, rev. ed edn. (1976). ISBN 0816211043 7, 113

[24] Chatfield, C.: *Time-series forecasting*. Chapman & Hall/CRC, Boca Raton (2001). ISBN 1584880635 (alk. paper) 7, 113, 127, 139

[25] De Gooijer, J. and Hyndman, R.: 25 years of time series forecasting. *International journal of forecasting* 22, 443–473 (2006) 7, 111, 113, 124

[26] Winters, P.: Forecasting sales by exponentially weighted moving averages. *Management Science* 324–342 (1960) 7, 114

[27] Holt, C.: Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting* 20, 5–10 (2004) 7, 114

[28] Kalman, R.: A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82, 35–45 (1960) 7

[29] Meinhold, R. and Singpurwalla, N.: Understanding the Kalman filter. *American Statistician* 123–127 (1983) 7

[30] Ahmed, N., Atiya, A., El Gayar, N., and El-Shishiny, H.: An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews* 29, 594–621 (2010) 7, 110, 112, 120

[31] Cao, L. and Tay, F.: Support vector machine with adaptive parameters in financial time series forecasting. *Ieee Transactions On Neural Networks* 14, 1506–1518 (2003) 7

[32] Tang, Z., de Almeida, C., and Fishwick, P.: Time series forecasting using neural networks vs. Box-Jenkins methodology. *Simulation* 57, 303–310 (1991) 7

[33] Verplancke, T., Van Looy, S., Steurbaut, K., Benoit, D., De Turck, F., De Moor, G., and Decruyenaere, J.: A novel time series analysis approach for prediction of dialysis in critically ill patients using echo-state networks. *BMC Med Inform Decis Mak* 10, 4 (2010) 7

[34] Tzafestas, S. and Tzafestas, E.: Computational intelligence techniques for short-term electric load forecasting. *Journal of Intelligent & Robotic Systems* 31, 7–68 (2001) 7

[35] Huang, N. E. and Shen, S. S.: *Hilbert-Huang transform and its applications*, vol. v. 5. World Scientific, Singapore (2005). ISBN 9812563768 (alk. paper) 8, 132, 143

[36] Takens, F.: Detecting strange attractors in turbulence. In Rand, D. and Young, B., eds., *Dynamical Systems and Turbulence*, vol. 898 of *Lecture Notes in Mathematics*, 366–381. Springer-Verlag, Berlin (1981) 8, 138, 139

[37] Sauer, T., Yorke, J., and Casdagli, M.: Embedology. *Journal of Statistical Physics* 65, 579–616 (1991) 8, 138

[38] Bielaszewska, M., Mellmann, A., Zhang, W., Köck, R., Fruth, A., Bauwens, A., Peters, G., and Karch, H.: Characterisation of the Escherichia coli strain associated with an outbreak of haemolytic uraemic syndrome in Germany, 2011: a microbiological study. *Lancet Infect Dis* 11, 671–6 (2011) 13, 165

[39] Walsh, T. R., Weeks, J., Livermore, D. M., and Toleman, M. A.: Dissemination of NDM-1 positive bacteria in the New Delhi environment and its implications for human health: an environmental point prevalence study. *Lancet Infect Dis* 11, 355–62 (2011) 13

[40] Lenzerini, M.: Data integration: a theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '02, 233–246. ACM, New York, NY, USA (2002). ISBN 1-58113-507-6 14, 25

[41] *IEEE standard computer dictionary: a compilation of IEEE standard computer glossaries, 610*. Institute of Electrical and Electronics Engineers, New York, NY, USA (1990). ISBN 1559370793 14

[42] Sheth, A. P. and Larson, J. A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput Surv* 22, 183–236 (1990) 14, 24

[43] Stelling, J. M. and O'Brien, T. F.: Surveillance of antimicrobial resistance: the WHONET program. *Clin Infect Dis* 24 Suppl 1, S157–68 (1997) 15, 63

[44] Karasavvas, K. A., Baldock, R., and Burger, A.: Bioinformatics integration and agent technology. *J Biomed Inform* 37, 205–19 (2004) 15, 24

[45] Mattes, W. B., Pettit, S. D., Sansone, S.-A., Bushel, P. R., and Waters, M. D.: Database development in toxicogenomics: issues and efforts. *Environ Health Perspect* 112, 495–505 (2004) 15

[46] Philippi, S. and Köhler, J.: Addressing the problems with life-science databases for traditional uses and systems biology. *Nat Rev Genet* 7, 482–8 (2006) 15

[47] Kalra, D.: Electronic health record standards. *Yearb Med Inform* 136–44 (2006) 16

[48] Sinnott, R. O., Stell, A. J., and Ajayi, O.: Supporting grid-based clinical trials in Scotland. *Health Informatics J* 14, 79–93 (2008) 16

[49] Goble, C. and Stevens, R.: State of the nation in data integration for bioinformatics. *J Biomed Inform* 41, 687–93 (2008) 16, 25, 26, 77

[50] Iakovidis, I.: Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare record in Europe. *Int J Med Inform* 52, 105–15 (1998) 16

[51] Eichelberg, M., Aden, T., Riesmeier, J., Dogac, A., and Laleci, G.: A survey and analysis of Electronic Healthcare Record standards. *ACM Computing Surveys (CSUR)* 37, 277–315 (2005) 16

[52] Smith, B. and Ceusters, W.: HL7 RIM: an incoherent standard. *Stud Health Technol Inform* 124, 133–8 (2006) 17

[53] Martínez-Costa, C., Menárguez-Tortosa, M., and Fernández-Breis, J. T.: Towards ISO 13606 and openEHR archetype-based semantic interoperability. *Stud Health Technol Inform* 150, 260–4 (2009) 18

[54] Cruz, I. and Xiao, H.: The role of ontologies in data integration. *Engineering Intelligent Systems For Electrical Engineering and Communications* 13, 245–252 (2005) 18, 28

[55] Bodenreider, O.: Biomedical ontologies in action: role in knowledge management, data integration and decision support. *Yearb Med Inform* 67, 79 (2008) 18

[56] Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., and Tanabe, M.: KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res* 40, D109–14 (2012) 18

[57] Yu, A. C.: Methods in biomedical ontology. *J Biomed Inform* 39, 252–66 (2006) 18

[58] Gruber, T.: A translation approach to portable ontology specifications. *Knowledge acquisition* 5, 199–220 (1993) 18

[59] Cimino, J. J.: In defense of the Desiderata. *J Biomed Inform* 39, 299–306 (2006) 18

[60] Baud, R. H., Ceusters, W., Ruch, P., Rassinoux, A.-M., Lovis, C., and Geissbühler, A.: Reconciliation of ontology and terminology to cope with linguistics. *Stud Health Technol Inform* 129, 796–801 (2007) 18

[61] Rector, A. L.: Clinical terminology: why is it so hard? *Methods Inf Med* 38, 239–52 (1999) 21, 46

[62] Ceusters, W. and Smith, B.: A unified framework for biomedical terminologies and ontologies. *Stud Health Technol Inform* 160, 1050–4 (2010) 21

[63] Bakken, S., Warren, J. J., Lundberg, C., Casey, A., Correia, C., Konicek, D., and Zingo, C.: An evaluation of the usefulness of two terminology models for integrating nursing diagnosis concepts into SNOMED Clinical Terms. *Int J Med Inform* 68, 71–7 (2002) 21

[64] Geiger, K.: *Inside ODBC.* Microsoft Press, Redmond, Wash. (1995). ISBN 1556158157 21

[65] Vinoski, S.: CORBA - Integrating diverse applications within distributed heterogeneous environments. *Ieee Communications Magazine* 35, 46–55 (1997) 21

[66] Zahavi, R. and David, S.: *Enterprise application integration with CORBA: component and Web-based solutions.* John Wiley & Sons, Inc. (1999) 21

[67] Linthicum, D.: *Enterprise application integration.* Addison-Wesley Longman Ltd. (2000) 21

[68] Berners-Lee, T. and Fischetti, M.: *Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor*. HarperCollins Publishers, New York, 1st pbk. ed edn. (2000). ISBN 006251587X (pbk.) 21

[69] Sahoo, S. S., Bodenreider, O., Rutter, J. L., Skinner, K. J., and Sheth, A. P.: An ontology-driven semantic mashup of gene and biological pathway information: application to the domain of nicotine dependence. *J Biomed Inform* 41, 752–65 (2008) 21

[70] Belleau, F., Nolin, M.-A., Tourigny, N., Rigault, P., and Morissette, J.: Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J Biomed Inform* 41, 706–16 (2008) 21, 30

[71] Chen, B., Dong, X., Jiao, D., Wang, H., Zhu, Q., Ding, Y., and Wild, D. J.: Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. *BMC Bioinformatics* 11, 255 (2010) 21

[72] Miñarro-Gimenez, J. A., Egaña Aranguren, M., Martínez Béjar, R., Fernández-Breis, J. T., and Madrid, M.: Semantic integration of information about orthologs and diseases: the OGO system. *J Biomed Inform* 44, 1020–31 (2011) 21

[73] Corwin, J., Silberschatz, A., Miller, P. L., and Marenco, L.: Dynamic tables: an architecture for managing evolving, heterogeneous biomedical data in relational database management systems. *J Am Med Inform Assoc* 14, 86–93 (2007) 21

[74] Noy, N.: Semantic integration: A survey of ontology-based approaches. *Sigmod Record* 33, 65–70 (2004) 22

[75] Chen, H., Yu, T., and Chen, J. Y.: Semantic Web meets Integrative Biology: a survey. *Brief Bioinform* (2012) 22, 23

[76] Donini, F., Lenzerini, M., Nardi, D., and Schaerf, A.: Reasoning in description logics. *Principles of knowledge representation* 191–236 (1996) 23

[77] Ding, L., Kolari, P., Ding, Z., and Avancha, S.: Using ontologies in the semantic web: A survey. *Ontologies* 79–113 (2007) 23, 28

[78] Seaborne, A.: RDQL - A Query Language for RDF. Tech. rep., W3C Member Submission (2004) 24

[79] Wielemaker, J.: An optimised Semantic Web query language implementation in prolog. *Logic Programming, Proceedings* 3668, 128–142 (2005) 24

[80] Haase, P., Broekstra, J., Eberhart, A., and Volz, R.: A comparison of RDF query languages. *The Semantic Web–ISWC 2004* 502–517 (2004) 24

[81] Bailey, J., Bry, F., Furche, T., and Schaffert, S.: Web and Semantic Web query languages: A survey. *Reasoning Web* 3564, 35–133 (2005) 24

[82] Etzold, T. and Argos, P.: SRS - an indexing and retrieval tool for flat file data libraries. *Computer Applications in the Biosciences* 9, 49–57 (1993) 24, 30, 31

[83] Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom, J.: The TSIMMIS project: Integration of heterogeneous information sources. In *Proceedings of IPSJ Conference*, 7–18. Citeseer (1994) 24, 27

[84] Goasdoue, F., Lattes, V., and Rousset, M.: The use of Carin language and algorithms for information integration: The Picsel system. *International Journal of Cooperative Information Systems* 9, 383–401 (2000) 24, 28, 29

[85] Eckman, B. A., Kosky, A. S., and Laroco, L. A., Jr: Extending traditional query-based integration approaches for functional characterization of post-genomic data. *Bioinformatics* 17, 587–601 (2001) 24, 27

[86] Bergamaschi, S., Castano, S., Vincini, M., and Beneventano, D.: Semantic integration of heterogeneous information sources. *Data & Knowledge Engineering* 36, 215–249 (2001) 24, 28

[87] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., and Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20, 3045–54 (2004) 24, 30

[88] Shah, S. P., Huang, Y., Xu, T., Yuen, M. M. S., Ling, J., and Ouellette, B. F. F.: Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics* 6, 34 (2005) 24, 25

[89] Butler, D.: Mashups mix data into global service. *Nature* 439, 6–7 (2006) 24, 31

[90] Töpel, T., Kormeier, B., Klassen, A., and Hofestädt, R.: BioDWH: a data warehouse kit for life science data integration. *J Integr Bioinform* 5 (2008) 24, 25, 32

[91] Shironoshita, E. P., Jean-Mary, Y. R., Bradley, R. M., and Kabuka, M. R.: semCDI: a query formulation for semantic data integration in caBIG. *J Am Med Inform Assoc* 15, 559–68 (2008) 24

[92] Cruz, I. F. and Xiao, H.: Ontology Driven Data Integration in Heterogeneous Networks. In *Complex Systems in Knowledge-based Environments*, 75–98. Springer (2009) 24

[93] Ritter, O., Kocab, P., Senger, M., Wolf, D., and Suhai, S.: Prototype implementation of the integrated genomic database. *Comput Biomed Res* 27, 97–115 (1994) 25

[94] Lee, T. J., Pouliot, Y., Wagner, V., Gupta, P., Stringer-Calvert, D. W. J., Tenenbaum, J. D., and Karp, P. D.: BioWarehouse: a bioinformatics database warehouse toolkit. *BMC Bioinformatics* 7, 170 (2006) 25

[95] Hernandez, T. and Kambhampati, S.: Integration of biological sources: Current systems and challenges ahead. *Sigmod Record* 33, 51–60 (2004) 25

[96] Wang, L., Zhang, A., and Ramanathan, M.: BioStar models of clinical and genomic data for biomedical data warehouse design. *Int J Bioinform Res Appl* 1, 63–80 (2005) 25

[97] Pillai, S. V., Gudipati, R., and Lilien, L.: Design issues and an architecture for a heterogenous multidatabase system. In *Proceedings of the 15th annual conference on Computer Science*, CSC '87, 74–79. ACM, New York, NY, USA (1987). ISBN 0-89791-218-7 27

[98] Davidson, S. B., Overton, G. C., Tannen, V., and Wong, L.: BioKleisli: A Digital Library for Biomedical Researchers. *Int J on Digital Libraries* 1, 36–53 (1997) 27, 32

[99] Haas, L., Schwarz, P., Kodali, P., Kotlar, E., Rice, J., and Swope, W.: DiscoveryLink: A system for integrated access to life sciences data sources. *Ibm Systems Journal* 40, 489–511 (2001) 27

[100] Kargl, F., Lawrence, E., Fischer, M., and Lim, Y.: Security, privacy and legal issues in pervasive ehealth monitoring systems. In *Mobile Business, 2008. ICMB'08. 7th International Conference on*, 296–304. Ieee (2008) 27

[101] Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S.: Ontology-based integration of information-a survey of existing approaches. In *IJCAI-01 workshop: ontologies and information sharing*, vol. 2001, 108–117. Citeseer (2001) 28, 29

[102] Weber, G. M., Murphy, S. N., McMurry, A. J., Macfadden, D., Nigrin, D. J., Churchill, S., and Kohane, I. S.: The Shared Health Research Information Network (SHRINE): a prototype federated query tool for clinical data repositories. *J Am Med Inform Assoc* 16, 624–30 (2009) 28, 34

[103] Goh, C., Bressan, S., Madnick, S., and Siegel, M.: Context interchange: New features and formalisms for the intelligent integration of information. *Acm Transactions On Information Systems* 17, 270–293 (1999) 28, 29

[104] Stevens, R., Baker, P., Bechhofer, S., Ng, G., Jacoby, A., Paton, N. W., Goble, C. A., and Brass, A.: TAMBIS: transparent access to multiple bioinformatics information sources. *Bioinformatics* 16, 184–5 (2000) 28, 29, 32

[105] Arens, Y., Hsu, C., and Knoblock, C.: Query processing in the SIMS information mediator. *Advanced Planning Technology* 32, 78–93 (1996) 29

[106] Preece, A., Hui, K., Gray, A., Marti, P., Bench-Capon, T., Jones, D., and Cui, Z.: The KRAFT architecture for knowledge fusion and transformation. *Knowledge-Based Systems* 13, 113–120 (2000) 29

[107] Castillo, J., Silvescu, A., Caragea, D., Pathak, J., and Honavar, V.: Information extraction and integration from heterogeneous, distributed, autonomous information sources-a federated ontology-driven query-centric approach. In *Information Reuse and Integration, 2003. IRI 2003. IEEE International Conference on*, 183–191. IEEE (2003) 29

[108] Schuler, G. D., Epstein, J. A., Ohkawa, H., and Kans, J. A.: Entrez: molecular biology database and retrieval system. *Methods Enzymol* 266, 141–62 (1996) 30

[109] Kersey, P. J., Morris, L., Hermjakob, H., and Apweiler, R.: Integr8: enhanced inter-operability of European molecular biology databases. *Methods Inf Med* 42, 154–60 (2003) 30

[110] InforSense. `http://www.inforsense.com` (2012). Last accessed: 2012-05-03 30

[111] About Pipes. `http://pipes.yahoo.com/pipes` (2012). Last accessed: 2012-05-03 31

[112] Saltz, J., Oster, S., Hastings, S., Langella, S., Kurc, T., Sanchez, W., Kher, M., Manisundaram, A., Shanbhag, K., and Covitz, P.: caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid. *Bioinformatics* 22, 1910–6 (2006) 33

[113] Foster, I. and Kesselman, C.: Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications and High Performance Computing* 11, 115–128 (1997) 33

[114] Antonioletti, M., Atkinson, M., Baxter, R., Borley, A., Hong, N., Collins, B., Hardman, N., Hume, A., Knox, A., Jackson, M., Krause, A., Laws, S., Magowan, J., Paton, N., Pearson, D., Sugden, T., Watson, P., and Westhead, M.: The design and implementation of Grid database services in OGSA-DAI. *Concurrency and Computation-Practice & Experience* 17, 357–376 (2005) 33

[115] Hastings, S., Langella, S., Oster, S., and Saltz, J.: Distributed data management and integration: The mobius project. In *GGF Semantic Grid Workshop*, vol. 2004, 20–38 (2004) 33

[116] Alonso-Calvo, R., Maojo, V., Billhardt, H., Martin-Sanchez, F., García-Remesal, M., and Pérez-Rey, D.: An agent- and ontology-based system for integrating public gene, protein, and disease databases. *J Biomed Inform* 40, 17–29 (2007) 33

[117] Murphy, S. N., Weber, G., Mendis, M., Gainer, V., Chueh, H. C., Churchill, S., and Kohane, I.: Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *J Am Med Inform Assoc* 17, 124–30 (2010) 35

[118] Lovis, C., Colaert, D., and Stroetmann, V. N.: DebugIT for patient safety - improving the treatment with antibiotics through multimedia data mining of heterogeneous clinical data. *Stud Health Technol Inform* 136, 641–6 (2008) 35, 147

[119] Schober, D., Boeker, M., Bullenkamp, J., Huszka, C., Depraetere, K., Teodoro, D., Nadah, N., Choquet, R., Daniel, C., and Schulz, S.: The DebugIT core ontology: semantic integration of antibiotics resistance patterns. *Stud Health Technol Inform* 160, 1060–4 (2010) 35, 67

[120] Teodoro, D., Choquet, R., Pasche, E., Gobeill, J., Daniel, C., Ruch, P., and Lovis, C.: Biomedical data management: a proposal framework. *Stud Health Technol Inform* 150, 175–9 (2009) 39, 41, 67

[121] Teodoro, D., Choquet, R., Schober, D., Mels, G., Pasche, E., Ruch, P., and Lovis, C.: Interoperability driven integration of biomedical data sources. *Stud Health Technol Inform* 169, 185–9 (2011) 39, 41, 67

[122] Levin, B. R., Perrot, V., and Walker, N.: Compensatory mutations, antibiotic resistance and the population genetics of adaptive evolution in bacteria. *Genetics* 154, 985–97 (2000) 40

[123] Ochman, H., Lawrence, J. G., and Groisman, E. A.: Lateral gene transfer and the nature of bacterial innovation. *Nature* 405, 299–304 (2000) 40

[124] Koonin, E. V., Makarova, K. S., and Aravind, L.: Horizontal gene transfer in prokaryotes: quantification and classification. *Annu Rev Microbiol* 55, 709–42 (2001) 40

[125] Hiramatsu, K., Cui, L., Kuroda, M., and Ito, T.: The emergence and evolution of methicillin-resistant Staphylococcus aureus. *Trends Microbiol* 9, 486–93 (2001) 41

[126] Sujansky, W.: Heterogeneous database integration in biomedicine. *J Biomed Inform* 34, 285–98 (2001) 41

[127] Aarts, J. and Koppel, R.: Implementation of computerized physician order entry in seven countries. *Health Aff (Millwood)* 28, 404–14 (2009) 41

[128] Brazhnik, O. and Jones, J.: Anatomy of data integration. *Journal of biomedical informatics* 40, 252–269 (2007) 45

[129] Hughes, T.: *Does technology drive history?* MIT Press (1994) 46

[130] Bizer, C.: D2R MAP–A DB to RDF Mapping Language. In *12th International World Wide Web Conference, Budapest* (2003) 48, 52, 71

[131] Tirmizi, S., Sequeda, J., and Miranker, D.: Translating sql applications to the semantic web. In *Database and Expert Systems Applications*, 450–464. Springer (2008) 48

[132] Das, S., Sundara, S., and Cyganiak, R.: R2RML: RDB to RDF mapping language. Tech. rep., W3C RDB2RDF Working Group, Available at http://www.w3. org/TR/r2rml (2010) 48

[133] Assélé Kama, A., Primadhanty, A., Choquet, R., Teodoro, D., Enders, F., Duclos, C., and Jaulent, M.-C.: Data Definition Ontology for clinical data integration and querying. *Stud Health Technol Inform* 180, 38–42 (2012) 50

[134] Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumueller, D.: Triplify: light-weight linked data publication from relational databases. In *Proceedings of the 18th international conference on World wide web*, WWW '09, 621–630. ACM, New York, NY, USA (2009). ISBN 978-1-60558-487-4 52

[135] G., B., Bauerdick, L., Belforte, S., Bloom, K., Bockelman, B., Bonacorsi, D., Brew, C., D'Hondt, J., Egeland, R., Elgammal, S., Fassi, F., Fisk, I., Flix, J., Hernandez, J. M., Kadastik, M., Klem, J., Kodolova, O., Kuo, C.-M., Letts,

J., Maes, J., Magini, N., Metson, S., Piedra, J., Pukhaeva, N., Qin, G., Rossman, P., Sartirana, A., Shih, J., Sonajalg, S., Teodoro, D., Trunov, A., Tuura, L., Van Mulders, P., Wildish, T., Wu, Y., and Wurthwein, F.: The CMS data transfer test environment in preparation for LHC data taking. In *Nuclear Science Symposium Conference Record, 2008. NSS'08. IEEE*, 3475–3482. IEEE (2008) 56

[136] Teodoro, D., Pasche, E., Gobeill, J., Emonet, S., Ruch, P., and Lovis, C.: Building a transnational biosurveillance network using semantic web technologies: requirements, design, and preliminary evaluation. *J Med Internet Res* 14, e73 (2012) 63

[137] Bronzwaer, S., Goettsch, W., Olsson-Liljequist, B., Wale, M., Vatopoulos, A., and Sprenger, M.: European Antimicrobial Resistance Surveillance System (EARSS): objectives and organisation. *Euro Surveill* 4, 41–44 (1999) 64

[138] Richet, H. M., Mohammed, J., McDonald, L. C., and Jarvis, W. R.: Building communication networks: international network for the study and prevention of emerging antimicrobial resistance. *Emerg Infect Dis* 7, 319–22 (2001) 64, 78

[139] Anonymous: The WHO Antimicrobial Resistance Information Bank. *WHO Drug Information* 13 (1999) 64

[140] Karlowsky, J. A., Kelly, L. J., Thornsberry, C., Jones, M. E., Evangelista, A. T., Critchley, I. A., and Sahm, D. F.: Susceptibility to fluoroquinolones among commonly isolated Gram-negative bacilli in 2000: TRUST and TSN data for the United States. Tracking Resistance in the United States Today. The Surveillance Network. *Int J Antimicrob Agents* 19, 21–31 (2002) 64

[141] Monnet, D. L.: Toward multinational antimicrobial resistance surveillance systems in Europe. *Int J Antimicrob Agents* 15, 91–101 (2000) 64, 74

[142] Giske, C. G., Cornaglia, G., and ESCMID Study Group on Antimicrobial Resistance Surveillance (ESGARS): Supranational surveillance of antimicrobial resistance: The legacy of the last decade and proposals for the future. *Drug Resist Updat* 13, 93–8 (2010) 64, 65, 66, 74, 78

[143] Schwaber, K. and Beedle, M.: *Agile software development with Scrum*, vol. 18. Prentice Hall (2001) 65

[144] So, A. D., Gupta, N., and Cars, O.: Tackling antibiotic resistance. *BMJ* 340, c2071 (2010) 66

[145] Piwowar, H. A., Becich, M. J., Bilofsky, H., Crowley, R. S., and caBIG Data Sharing and Intellectual Capital Workspace: Towards a data sharing culture: recommendations for leadership from academic health centers. *PLoS Med* 5, e183 (2008) 66

[146] Anonymous: *A comprehensive approach on personal data protection in the European Union*. European Commission (2010) 68

[147] Gamma, E.: *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Professional (1995) 69, 74

[148] Timpka, T., Eriksson, H., Gursky, E. A., Strömgren, M., Holm, E., Ekberg, J., Eriksson, O., Grimvall, A., Valter, L., and Nyce, J. M.: Requirements and design of the PROSPER protocol for implementation of information infrastructures supporting pandemic response: a Nominal Group study. *PLoS One* 6, e17941 (2011) 69

[149] Ruch, P., Gobeill, J., Lovis, C., and Geissbühler, A.: Automatic medical encoding with SNOMED categories. *BMC Med Inform Decis Mak* 8 Suppl 1, S6 (2008) 71, 75

[150] Schmidt, M., Hornung, T., Lausen, G., and Pinkel, C.: SP2Bench: A SPARQL Performance Benchmark. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, 222–233. Ieee (2009) 71

[151] Ruch, P.: Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics* 22, 658–664 (2006) 75

[152] Momtchev, V., Peychev, D., Primov, T., and Georgiev, G.: Expanding the pathway and interaction knowledge in linked life data. *Proc of International Semantic Web Challenge* (2009) 77

[153] Calvanese, D., Giacomo, G. D., and Lenzerini, M.: Ontology of Integration and Integration of Ontologies. In *Description Logics* (2001) 78

[154] Nielsen, J.: *Usability engineering.* Academic Press, Boston (1993). ISBN 0125184050 (acid-free paper) 80

[155] Lung, K. R., Gorko, M. A., Llewelyn, J., and Wiggins, N.: Statistical method for the determination of equivalence of automated test procedures. *J Autom Methods Manag Chem* 25, 123–7 (2003) 84

[156] Johnston, R. and Duke, J.: Benefit transfer equivalence tests with non-normal distributions. *Environmental and Resource Economics* 41, 1–23 (2008) 84

[157] Pigeot, I., Hauschke, D., and Shao, J.: The bootstrap in bioequivalence studies. *J Biopharm Stat* 21, 1126–39 (2011) 86

[158] Cribbie, R. A., Gruman, J. A., and Arpin-Cribbie, C. A.: Recommendations for applying tests of equivalence. *J Clin Psychol* 60, 1–10 (2004) 86

[159] Hothorn, L. A. and Hasler, M.: Proof of hazard and proof of safety in toxicological studies using simultaneous confidence intervals for differences and ratios to control. *J Biopharm Stat* 18, 915–33 (2008) 86

[160] Silverman, D.: *Qualitative research: theory, method and practice.* Sage Publications, London, 2nd ed edn. (2004). ISBN 0761949348 (pbk.) 87

[161] Morsey, M., Lehmann, J., Auer, S., and Ngomo, A.-C. N.: DBpedia SPARQL benchmark: performance assessment with real queries on real data. In *Proceedings of the 10th international conference on The semantic web - Volume Part I*, ISWC'11, 454–469. Springer-Verlag, Berlin, Heidelberg (2011). ISBN 978-3-642-25072-9 98

[162] Lee, F., Teich, J. M., Spurr, C. D., and Bates, D. W.: Implementation of physician order entry: user satisfaction and self-reported usage patterns. *J Am Med Inform Assoc* 3, 42–55 (1996) 98

[163] Reynolds, R., Hope, R., and Williams, L.: Survey, laboratory and statistical methods for the BSAC Resistance Surveillance Programmes. *Journal of antimicrobial chemotherapy* 62, ii15–ii28 (2008) 99

[164] Li, N., Raskin, R., Goodchild, M., and Janowicz, K.: An Ontology-Driven Framework and Web Portal for Spatial Decision Support. *Transactions in GIS* 16, 313–329 (2012) 100

[165] Russell, S. and Norvig, P.: *Artificial intelligence: a modern approach*. Prentice hall (2010) 105

[166] Hastie, T., Tibshirani, R., and Friedman, J. H.: *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, New York, NY, 2nd ed edn. (2009). ISBN 9780387848570 (hardcover : alk. paper) 105

[167] Muggleton, S.: Logic and learning: Turings legacy. *Muggleton, SH and Michie, D Furukaw, K, editors, Machine Intelligence* 13 (1993) 106

[168] Anctil, F. and Rat, A.: Evaluation of neural network streamflow forecasting on 47 watersheds. *Journal of Hydrologic Engineering* 10, 85–88 (2005) 106

[169] Krasnopolsky, V. M. and Fox-Rabinovitz, M. S.: Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks* 19, 122–134 (2006) 106

[170] Burbidge, R., Trotter, M., Buxton, B., and Holden, S.: Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput Chem* 26, 5–14 (2001) 106

[171] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422 (2002) 106

[172] Blankertz, B., Curio, G., and Müller, K.-R.: Classifying Single Trial EEG: Towards Brain Computer Interfacing. In *NIPS*, 157–164 (2001) 106

[173] Müller, K.-R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., and Blankertz, B.: Machine learning for real-time single-trial EEG-analysis: from brain-computer interfacing to mental state monitoring. *J Neurosci Methods* 167, 82–90 (2008) 106

[174] Aguilar-Arevalo, A. A., Bazarko, A. O., Brice, S. J., Brown, B. C., Bugel, L., Cao, J., Coney, L., Conrad, J. M., Cox, D. C., Curioni, A., Djurcic, Z., Finley, D. A., Fleming, B. T., Ford, R., Garcia, F. G., Garvey, G. T., Green, C., Green, J. A., Hart, T. L., Hawker, E., Imlay, R., Johnson, R. A., Kasper, P., Katori, T., Kobilarcik, T., Kourbanis, I., Koutsoliotas, S., Laird, E. M., Link, J. M., Liu, Y., Liu, Y., Louis, W. C., Mahn, K. B. M., Marsh, W., Martin, P. S., McGregor, G., Metcalf, W., Meyers, P. D., Mills, F., Mills, G. B., Monroe, J., Moore, C. D., Nelson, R. H., Nienaber, P., Ouedraogo, S., Patterson, R. B., Perevalov, D., Polly, C. C., Prebys, E., Raaf, J. L., Ray, H., Roe, B. P., Russell, A. D., Sandberg, V., Schirato, R., Schmitz, D., Shaevitz, M. H., Shoemaker, F. C., Smith, D., Sorel, M., Spentzouris, P., Stancu, I., Stefanski, R. J., Sung, M., Tanaka, H. A., Tayloe, R., Tzanov, M., Van de Water, R., Wascko, M. O., White, D. H., Wilking, M. J., Yang, H. J., Zeller, G. P., and Zimmerman, E. D.: Search for electron neutrino appearance at the Delta m(2)similar to 1 eV(2) scale. *Physical Review Letters* 98, 231801 (2007) 106

[175] Huang, W., Nakamori, Y., and Wang, S.: Forecasting stock market movement direction with support vector machine. *Computers & Operations Research* 32, 2513–2522 (2005) 106

[176] Chan, P. and Stolfo, S.: Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proceedings of the fourth international conference on knowledge discovery and data mining*, vol. 164, 168 (1998) 106

[177] Sebastiani, F.: Machine learning in automated text categorization. *Acm Computing Surveys* 34, 1–47 (2002) 106

[178] Graves, A. and Schmidhuber, J.: Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. In *NIPS*, 545–552 (2008) 106

[179] Tong, S. and Chang, E.: Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, MULTIMEDIA '01, 107–118. ACM, New York, NY, USA (2001). ISBN 1-58113-394-4 106

[180] Ledley, R. S. and Lusted, L. B.: Probability, Logic and Medical Diagnosis. *Science* 130, 892–930 (1959) 106

[181] Magoulas, G. and Prentza, A.: Machine learning in medical applications. *Machine Learning and Its Applications* 300–307 (2001) 106

[182] Weston, A. D. and Hood, L.: Systems biology, proteomics, and the future of health care: toward predictive, preventative, and personalized medicine. *J Proteome Res* 3, 179–96 (2004) 106

[183] Stach, W., Kurgan, L., Pedrycz, W., and Reformat, M.: Genetic learning of fuzzy cognitive maps. *Fuzzy Sets and Systems* 153, 371–401 (2005) 106

[184] Cruz, J. A. and Wishart, D. S.: Applications of machine learning in cancer prediction and prognosis. *Cancer Inform* 2, 59–77 (2006) 106, 107

[185] Peng, Y., Li, W., and Liu, Y.: A hybrid approach for biomarker discovery from microarray gene expression data for cancer classification. *Cancer Inform* 2, 301–11 (2006) 106

[186] Kononenko, I.: Machine learning for medical diagnosis: history, state of the art and perspective. *Artif Intell Med* 23, 89–109 (2001) 107, 116

[187] Savage, N.: Better medicine through machine learning. *Communications of the ACM* 55, 17–19 (2012) 107

[188] Syeda-Mahmood, T., Beymer, D., and Wang, F.: Shape-based matching of ECG recordings. *Conf Proc IEEE Eng Med Biol Soc* 2007, 2012–8 (2007) 107

[189] Visweswaran, S., Angus, D. C., Hsieh, M., Weissfeld, L., Yealy, D., and Cooper, G. F.: Learning patient-specific predictive models from clinical data. *J Biomed Inform* 43, 669–85 (2010) 107

[190] Ramoni, M., Sebastiani, P., and Dybowski, R.: Robust outcome prediction for intensive-care patients. *Methods of Information in Medicine-Methodik der Information in der Medizin* 40, 39–45 (2001) 107

[191] Schurink, C. A. M., Lucas, P. J. F., Hoepelman, I. M., and Bonten, M. J. M.: Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care units. *Lancet Infect Dis* 5, 305–12 (2005) 107

[192] Sintchenko, V., Coiera, E., and Gilbert, G. L.: Decision support systems for antibiotic prescribing. *Curr Opin Infect Dis* 21, 573–9 (2008) 107

[193] McBryde, E. S., Pettitt, A. N., Cooper, B. S., and McElwain, D. L. S.: Characterizing an outbreak of vancomycin-resistant enterococci using hidden Markov models. *J R Soc Interface* 4, 745–54 (2007) 107, 159, 165

[194] Gierl, L., Steffen, D., Ihracky, D., and Schmidt, R.: Methods, architecture, evaluation and usability of a case-based antibiotics advisor. *Comput Methods Programs Biomed* 72, 139–54 (2003) 107

[195] Leibovici, L., Paul, M., Nielsen, A. D., Tacconelli, E., and Andreassen, S.: The TREAT project: decision support and prediction using causal probabilistic networks. *Int J Antimicrob Agents* 30 Suppl 1, S93–102 (2007) 107

[196] Zhang, S., Zhang, C., and Yang, Q.: Data preparation for data mining. *Applied Artificial Intelligence* 17, 375–381 (2003) 108

[197] Wolpert, D.: The lack of A priori distinctions between learning algorithms. *Neural Computation* 8, 1341–1390 (1996) 108

[198] Dougherty, J., Kohavi, R., and Sahami, M.: Supervised and unsupervised discretization of continuous features. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, 194–202. Morgan Kaufmann Publishers, Inc. (1995) 109

[199] Teodoro, D., Pasche, E., Vishnyakova, D., Lovis, C., Gobeill, J., and Ruch, P.: Automatic IPC encoding and novelty tracking for effective patent mining. In *Proceedings of NTCIR-8 Workshop Meeting* (2010) 110

[200] Teodoro, D., Gobeill, J., Pasche, E., Vishnyakova, D., Ruch, P., and Lovis, C.: Automatic Prior Art Searching and Patent Encoding at CLEF-IP 2010. In *Workshop of the Cross-Language Evaluation Forum, LABs andWorkshops, Notebook Papers* (2010) 110

[201] Duda, R., Hart, P., and Stork, D.: Pattern classification. *New York: John Wiley, Section* 10, l (2001) 110

[202] MacKay, D. J. C.: *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge, UK (2003). ISBN 0521642981 110

[203] Alpaydin, E.: *Introduction to machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 2nd ed edn. (2010). ISBN 9780262012430 (hardcover : alk. paper) 110

[204] Lane, T. and Brodley, C. E.: Temporal sequence learning and data reduction for anomaly detection. *ACM Trans Inf Syst Secur* 2, 295–331 (1999) 112

[205] Keogh, E. and Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*, 239–241 (1998) 112

[206] Sakurai, Y., Yoshikawa, M., and Faloutsos, C.: FTW: fast similarity search under the time warping distance. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '05, 326–337. ACM, New York, NY, USA (2005). ISBN 1-59593-062-0 112

[207] Ramirez-Amaro, K. and Chimal-Eguia, J.: Machine Learning Tools to Time Series Forecasting. In *Artificial Intelligence-Special Session, 2007. MICAI 2007. Sixth Mexican International Conference on*, 91–101. IEEE (2007) 112

[208] Yule, G.: On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London Series A, Containing Papers of a Mathematical or Physical Character* 226, 267–298 (1927) 113

[209] López-Lozano, J. M., Monnet, D. L., Yagüe, A., Burgos, A., Gonzalo, N., Campillos, P., and Saez, M.: Modelling and forecasting antimicrobial resistance and its dynamic relationship to antimicrobial use: a time series analysis. *Int J Antimicrob Agents* 14, 21–31 (2000) 113, 136

[210] Kritsotakis, E. I., Christidou, A., Roumbelaki, M., Tselentis, Y., and Gikas, A.: The dynamic relationship between antibiotic use and the incidence of vancomycin-resistant Enterococcus: time-series modelling of 7-year surveillance data in a tertiary-care hospital. *Clin Microbiol Infect* 14, 747–54 (2008) 113

[211] Abeku, T. A., de Vlas, S. J., Borsboom, G., Teklehaimanot, A., Kebede, A., Olana, D., van Oortmarssen, G. J., and Habbema, J. D. F.: Forecasting malaria incidence from historical morbidity patterns in epidemic-prone areas of Ethiopia: a simple seasonal adjustment method performs best. *Trop Med Int Health* 7, 851–7 (2002) 113

[212] Elbert, Y. and Burkom, H. S.: Development and evaluation of a data-adaptive alerting algorithm for univariate temporal biosurveillance data. *Stat Med* 28, 3226–48 (2009) 114

[213] Najmi, A.-H. and Magruder, S. F.: An adaptive prediction and detection algorithm for multistream syndromic surveillance. *BMC Med Inform Decis Mak* 5, 33 (2005) 114

[214] Najmi, A.-H. and Burkom, H.: Recursive least squares background prediction of univariate syndromic surveillance data. *BMC Med Inform Decis Mak* 9, 4 (2009) 114

[215] Witten, I., Frank, E., and Hall, M.: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann (2011) 115, 122

[216] McNally, R. J. Q., James, P. W., Picton, S. V., McKinney, P. A., van Laar, M., and Feltbower, R. G.: Space-time clustering of childhood central nervous system tumours in Yorkshire, UK. *BMC Cancer* 12, 13 (2012) 115

[217] Kotsiantis, S., Zaharakis, I., and Pintelas, P.: Supervised machine learning: A review of classification techniques. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS* 160, 3 (2007) 116

[218] Bellazzi, R. and Zupan, B.: Predictive data mining in clinical medicine: current issues and guidelines. *Int J Med Inform* 77, 81–97 (2008) 116

[219] Dasgupta, A., Sun, Y. V., König, I. R., Bailey-Wilson, J. E., and Malley, J. D.: Brief review of regression-based and machine learning methods in genetic epidemiology: the Genetic Analysis Workshop 17 experience. *Genet Epidemiol* 35 Suppl 1, S5–11 (2011) 117

[220] Bishop, C. M.: *Neural networks for pattern recognition*. Clarendon Press, Oxford (1995). ISBN 0198538499 (hbk) 118

[221] Dorffner, G.: Neural networks for time series processing. In *Neural Network World*. Citeseer (1996) 118

[222] Camargo, L. and Yoneyama, T.: Specification of training sets and the number of hidden neurons for multilayer perceptrons. *Neural Computation* 13, 2673–2680 (2001) 118

[223] Cortes, C.: *Prediction of generalization ability in learning machines*. Ph.D. thesis, Department of Computer Science, University of Rochester (1995) 119

[224] Burges, C.: A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2, 121–167 (1998) 119

[225] Smola, A. and Schölkopf, B.: A tutorial on support vector regression. *Statistics and computing* 14, 199–222 (2004) 119, 120

[226] Rasmussen, C.: Gaussian processes in machine learning. *Advanced Lectures On Machine Learning* 3176, 63–71 (2004) 121

[227] Arlot, S. and Celisse, A.: A survey of cross-validation procedures for model selection. *Statistics Surveys* 4, 40–79 (2010) 123

[228] Fridkin, S. K., Edwards, J. R., Tenover, F. C., Gaynes, R. P., McGowan, J. E., Jr, Intensive Care Antimicrobial Resistance Epidemiology (ICARE) Project, and National Nosocomial Infections Surveillance (NNIS) System Hospitals: Antimicrobial resistance prevalence rates in hospital antibiograms reflect prevalence rates among pathogens associated with hospital-acquired infections. *Clin Infect Dis* 33, 324–30 (2001) 126

[229] Jaecklin, T., Rohner, P., Jacomo, V., Schmidheiny, K., and Gervaix, A.: Trends in antibiotic resistance of respiratory tract pathogens in children in Geneva, Switzerland. *Eur J Pediatr* 165, 3–8 (2006) 126, 159

[230] Austin, D. J., Kristinsson, K. G., and Anderson, R. M.: The relationship between the volume of antimicrobial consumption in human communities and the frequency of resistance. *Proc Natl Acad Sci U S A* 96, 1152–6 (1999) 126, 136, 150

[231] F. Araujo, M. B. M. A. and Neto, J. A. R.: r-filters: a Hodrick-Prescott Filter Generalization. In *Working Paper Series*, 69. Banco Central do Brasil (2003) 130

[232] Alexandrov, T., Bianconcini, S., Dagum, E., Maass, P., and McElroy, T.: A review of some modern approaches to the problem of trend extraction. Tech. Rep. RRS2008/03, US Census Bureau (2008) 131

[233] French, M.: Estimating changes in trend growth of total factor productivity: Kalman and HP filters versus a Markov-switching framework. In *FEDS Working Paper No. 2001-44*. Board of Governors of the Federal Reserve System (US), Available at SSRN: http://ssrn.com/abstract=293105 or http://dx.doi.org/10.2139/ssrn.293105 (2001) 131

[234] Graps, A.: An introduction to wavelets. *Computational Science & Engineering, IEEE* 2, 50–61 (1995) 131

[235] Mhamdi, F., Jaıdane-Saıdane, M., and Poggi, J.: Empirical mode decomposition for trend extraction: application to electrical data. In *Proceedings of COMPSTAT*, 22–27 (2010) 132

[236] Wu, Z., Huang, N. E., Long, S. R., and Peng, C.-K.: On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proc Natl Acad Sci U S A* 104, 14889–94 (2007) 132, 134, 143, 159

[237] Li, K., Hogrel, J.-Y., Duchêne, J., and Hewson, D. J.: Analysis of fatigue and tremor during sustained maximal grip contractions using Hilbert-Huang Transformation. *Med Eng Phys* 34, 832–40 (2012) 132

[238] Vernaz, N., Huttner, B., Muscionico, D., Salomon, J.-L., Bonnabry, P., López-Lozano, J. M., Beyaert, A., Schrenzel, J., and Harbarth, S.: Modelling the impact of antibiotic use on antibiotic-resistant Escherichia coli using population-based data from a large hospital and its surrounding community. *J Antimicrob Chemother* 66, 928–35 (2011) 136

[239] Cooper, B. S., Medley, G. F., and Scott, G. M.: Preliminary analysis of the transmission dynamics of nosocomial infections: stochastic and management effects. *J Hosp Infect* 43, 131–47 (1999) 136

[240] Haber, M., Levin, B. R., and Kramarz, P.: Antibiotic control of antibiotic resistance in hospitals: a simulation study. *BMC Infect Dis* 10, 254 (2010) 136

[241] Barbosa, T. M. and Levy, S. B.: The impact of antibiotic use on resistance development and persistence. *Drug Resist Updat* 3, 303–311 (2000) 136, 159

[242] Kuehn, B. M.: FDA aims to curb farm use of antibiotics. *JAMA* 307, 2244–5 (2012) 138

[243] Hegger, R., Kantz, H., and Matassini, L.: Denoising human speech signals using chaoslike features. *Physical Review Letters* 84, 3197–3200 (2000) 139

[244] Chakrabarti, D. and Faloutsos, C.: F4: large-scale automated forecasting using fractals. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, 2–9. ACM, New York, NY, USA (2002). ISBN 1-58113-492-4 143, 144, 153

[245] Crone, S., Hibon, M., and Nikolopoulos, K.: Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting* 27, 635–660 (2011) 148

[246] Goossens, H., Ferech, M., Vander Stichele, R., Elseviers, M., and ESAC Project Group: Outpatient antibiotic use in Europe and association with resistance: a cross-national database study. *Lancet* 365, 579–87 (2005) 152, 153

[247] Cadena, J., Taboada, C. A., Burgess, D. S., Ma, J. Z., Lewis, J. S., 2nd, Freytes, C. O., and Patterson, J. E.: Antibiotic cycling to decrease bacterial antibiotic

resistance: a 5-year experience on a bone marrow transplant unit. *Bone Marrow Transplant* 40, 151–5 (2007) 152

[248] Chait, R., Craney, A., and Kishony, R.: Antibiotic interactions that select against resistance. *Nature* 446, 668–71 (2007) 153

[249] Xie, H. and Wang, Z.: Mean frequency derived via Hilbert-Huang transform with application to fatigue EMG signal analysis. *Comput Methods Programs Biomed* 82, 114–20 (2006) 159

[250] Hoehndorf, R., Dumontier, M., Oellrich, A., Rebholz-Schuhmann, D., Schofield, P. N., and Gkoutos, G. V.: Interoperability between biomedical ontologies through relation expansion, upper-level ontologies and automatic reasoning. *PLoS One* 6, e22006 (2011) 169

[251] Shaw, M., Detwiler, L., Brinkley, J., and Suciu, D.: A dataflow graph transformation language and query rewriting system for RDF ontologies. In *Scientific and Statistical Database Management*, 544–561. Springer (2012) 169

[252] Hsu, L.-Y., Tan, T.-Y., Tam, V. H., Kwa, A., Fisher, D. A., Koh, T.-H., and Network for Antimicrobial Resistance Surveillance (Singapore): Surveillance and correlation of antibiotic prescription and resistance of Gram-negative bacteria in Singaporean hospitals. *Antimicrob Agents Chemother* 54, 1173–8 (2010) 169