UPPSALA
UNIVERSITET

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology

From Physicochemical Features to Interdependency Networks

A Monte Carlo Approach to Modeling HIV-1 Resistome and Post-translational Modifications

MARCIN KIERCZAK

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2009

ISSN 1651-6214
urn:nbn:se:uu:diva-109873

Dissertation presented at Uppsala University to be publicly examined in C8:305, BMC, Husargatan 3, Uppsala, Tuesday, December 15, 2009 at 09:15 for the degree of Doctor of Engineering. The examination will be conducted in English.

Abstract
Kierczak, M. 2009. From Physicochemical Features to Interdependency Networks. A Monte Carlo Approach to Modeling HIV-1 Resistome and Post-translational Modifications. Acta Universitatis Upsaliensis. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 688. 89 pp. Uppsala. ISBN 978-91-554-7650-2.

The availability of new technologies supplied life scientists with large amounts of experimental data. The data sets are large not only in terms of the number of observations, but also in terms of the number of recorded features. One of the aims of modeling is to explain a given phenomenon in possibly the simplest way, hence the need for selection of suitable features.

We extended a Monte Carlo-based approach to selecting statistically significant features with discovery of feature interdependencies and used it in modeling sequence-function relationships in proteins. Our approach led to compact and easy-to-interpret predictive models.

First, we represented protein sequences in terms of their physicochemical properties. This was followed by our feature selection and discovery of feature interdependencies. Finally, predictive models based on e.g., decision trees or rough sets were constructed.

We applied the method to model two important biological problems: 1) HIV-1 resistance to reverse transcriptase-targeted drugs and 2) post-translational modifications of proteins.

In the case of HIV resistance, we were not only able to predict whether the mutated protein is resistant to a drug or not, but we also suggested some new, previously neglected, mutations that possibly contribute to drug resistance. For all these mutations we proposed probable molecular mechanisms of action using literature and 3D structure studies.

In the case of predicting PTMs, we built high accuracy models of modifications. In comparison to other methods, we were able to resolve whether the closest neighborhood of a residue (the nanomer) is sufficient to determine its modification status. Importantly, the application of our method yields networks of interdependent physicochemical properties of amino acids that show how these properties collaborate in establishing a given modification.

We believe that the presented methods will help researchers to analyze a large class of important biological problems and will guide them in their research.

Keywords: bioinformatics, HIV-1, resistome analysis, drug resistance, predicting PTMs, molecular interdependency networks, MCFS-ID, feature selection, interactome, machine-learning, rough sets

Marcin Kierczak, The Linnaeus Centre for Bioinformatics, Box 598, Uppsala University, SE-75124 Uppsala, Sweden

ISSN 1651-6214
ISBN 978-91-554-7650-2
urn:nbn:se:uu:diva-109873 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-109873)

to my Family and my other Teachers

List of Papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I **Kierczak M**, Ginalski K, Dramiński M, Koronacki J, Rudnicki WR, Komorowski J. A rough set-based model of HIV-1 reverse transcriptase resistome. *Bioinformatics and Biology Insights*. 2009; 3: 109-127

II Dramiński M*, **Kierczak M***, Koronacki J, Komorowski J. Monte Carlo feature selection and interdependency discovery in supervised classification. 2009; in: J. Koronacki, Z. Raś, S. Wierzchoń, J. Kacprzyk [eds.], *Recent Advances in Machine Learning*, Springer, Heidelberg. *Accepted*

III **Kierczak M***, Dramiński M*, Koronacki J, Komorowski J. Analysis of local molecular interaction networks underlying HIV-1 resistance to reverse transcriptase inhibitors. *Submitted*

IV **Kierczak M**, Plewczyński D, Dramiński M, Andersson G, Lampa S, Rudnicki W, Koronacki J, Ginalski K, Komorowski J. A Monte Carlo approach to modeling post-translational modification sites using local physicochemical properties. *Manuscript*

* Authors contributed equally.

List of Additional Papers

I Rudnicki WR, **Kierczak M**, Koronacki J, Komorowski J. A statistical method for determining importance of variables in an information system. Lecture Notes in Computer Science: Rough Sets and Current Trends in Computing 2006; 4259: 557-566.

II **Kierczak M**, Rudnicki WR, Komorowski JH. Construction of rough set-based classifiers for predicting HIV resistance to nucleoside reverse transcriptase inhibitors. Studies in Fuzziness and Soft Computing: Granular Computing: At the Junction of Rough Sets and Fuzzy Sets 2008; 224: 249-258.

Contents

Nomenclature

AIDS	acquired immune deficiency syndrome
AUC	area under the ROC curve
DNA	deoxyribonucleic acid
DNAP	DNA-dependent DNA polymeraze
dNTP	deoxyribonucleotide triphosphate
env	envelope proteins
FN	false negative
FP	false positive
FPR	false positive rate
gag	group-specific antigens
HAART	highly-active antiretroviral therapy
HIV	human immunodeficiency virus
HIV-1	human immunodeficiency virus type I
HIV-2	human immunodeficiency virus type II
LTR	long terminal repeat
ML	machine learning
mRNA	messenger RNA
NNRTI	non-NRTI drug
NRTI	nucleoside RT inhibitor
NtRTI	nucleotide RT inhibitor
PBS	primer binding site
PDB	Protein Data Bank
pol	polymerase
PP	polypurine section
PTM	post-translational modification
RNA	ribonucleic acid
ROC	receiver operating characteristics
RT	reverse transcriptase
SIV	simian immune deficiency virus
TN	true negative
TP	true positive
TPR	true positive rate
tRNA	transporter RNA
U3	3’ unique sequence
vDNA	viral DNA

1. Introduction

Frustra fit per plura quod potest fieri per pauciora.[1]

William of Ockham

A complex interplay of numerous molecules and processes constantly takes place in a living organism. The processes that underly life are multi-layered: from nano-scale events through molecules, molecular networks, cells and organs up to inter-organismal and organism-environment interactions. Biology describes and attempts at understanding life at all these levels. Initially observation was the main tool and classification was the main task of a naturalist. Over many years biology underwent significant transformations. Quite recently these changes accelerated in response to the invention of new technologies such as microarray technology or next-generation sequencing. These new technologies allow us to perform high-throughput experiments that supply the scientific community with large amounts of experimental data. The data sets are large not only in terms of the number of observations, but also in terms of the number of recorded features. One of the aims of machine learning-based modeling is to explain a given phenomenon in possibly the simplest way, hence the need for selection of suitable features.

Another aim of modeling is to build models that are not only accurate, but also easy-to-interpret by a domain expert. Such models are especially desirable in modeling the broad range of biological problems where subtle mutation- or modification-induced changes in physicochemical properties affect, sometimes substantially, protein function. This class of problems encompasses phenomena like: modeling drug resistance or post-translational modifications, protein design and engineering or understanding the role of polymorphisms and mutations in health and disease.

We extended a Monte Carlo-based approach to selecting statistically significant features and to make possible the discovery and analysis of feature interdependencies. We used it in modeling sequence-function relationships in proteins. It is often a non-linear interplay of many physicochemical changes rather than singular mutations that leads to a shift of protein function. Our models easily deal with such complex networks and with non-linearity. The

[1] *It is vain to do using many [things] what can be done using less [things].* Occam's formulation of his *razor principle*.

presented approach begins with representing every amino acid in a protein sequence in terms of its physicochemical properties. This is followed by the selection of those molecular features that significantly contribute to the shift of function. At this stage the method allows for the discovery of interdependencies between significant features. Application of a machine-learning technique of choice finalizes the modeling part as it yields a legible and predictive model that can be interpreted and analyzed further.

The approach was applied to model two important biological phenomena: 1) HIV-1 resistance to reverse transcriptase inhibitors and 2) predicting post-translational modifications in proteins. The results are presented in this thesis.

Since the thesis is addressed to researchers representing such different disciplines as biology, medicine and computer science, first I will introduce the reader with a different background to the concepts that are essential for understanding the presented work.

2. Aims

- To develop a general method for the study of sequence-function relationship in proteins and to apply this method to model and increase the understanding of protein-variability space, especially in relation to the performed function.
- To apply the developed method to the HIV-1 resistance problem and build an easy-to-interpret model of viral enzyme, reverse transcriptase resistome.
- To perform an in-depth analysis of the resistome by studying the obtained model and to propose the possible molecular mechanisms of mutation-induced resistance.
- To construct a predictive model allowing for the identification of potential post-translational modification sites solely on the basis of protein sequence.
- To identify the physicochemical properties that determine whether a given sequence fragment will possibly be subjected to a given post-translational modification or not.

3. Biology Background

In this chapter I give a brief introduction necessary to understand the biolmedical aspects of the work. I begin with the central dogma of molecular biology and the structure of nucleic acids, move to the biology of viruses, especially of HIV, briefly talk about AIDS and continue to proteins and the role of post-translational modifications in altering their function. It is an interdisciplinary thesis. The reader with background in biology may proceed directly to Chapter 4 while the reader familiar with computational issues will find biology background in this chapter.

3.1 The flow of genetic information

Organisms are born, they live, reproduce and eventually die. The constant flow of information from an organism to its offspring assures the continuity of life. The full set of instructions necessary to build an organism capable of reproducing itself is stored in the form of *nucleic acids*: long chemical polymers that can be seen as sequences of four letters. There are two major types of nucleic acids: DNA and RNA. Usually the genetic information is stored in DNA which provides a high degree of protection and can be compared to a hard disk. Whenever a piece of information is required, it is *transcribed* into messenger RNA which serves as a kind of temporary storage, quick-access memory. This messenger RNA can be, in turn, *translated* into a sequence of *amino acids*. Twenty different amino acids are the basic building-blocks of proteins. Proteins build cells and perform many important functions from copying DNA to interacting with the outside world. The flow of information from DNA to RNA and to proteins is the *central dogma of molecular biology*.

DNA consists of two strands, long polymers composed of simple building blocks called nucleotides (see Figure 3.1). A DNA-building *nucleotide* is composed of a nucleotide base, a sugar deoxiribose and three phosphate groups. A nucleotide base together with the deoxiribose part of a nucleotide form a *nucleoside* (see Figure 3.1). The backbone of each DNA strand is composed of sugars and phosphate groups joined by phosphodiester bonds (see Figure 3.2). These two strands are anti-parallel (i.e. they run in opposite directions to each other). A *nucleotide base*, adenine (A), cytosine (C), guanine (G) or thymine (T) is attached to each sugar in the backbone and it is these bases that actu-

ally encode genetic information. The two antiparallel strands form the famous double-helix. RNA is a similar polymer, usually consisting of only one strand and containing uracil (U) instead of thymine and sugar ribose instead of deoxyribose.

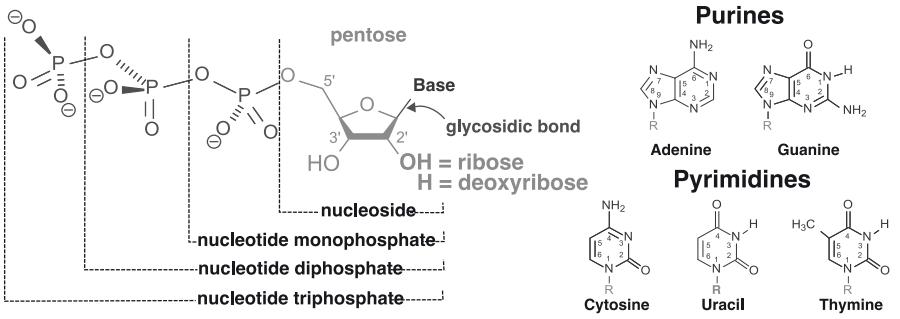

Figure 3.1: The structure elements of nucleosides and nucleotides. Adapted from [2].

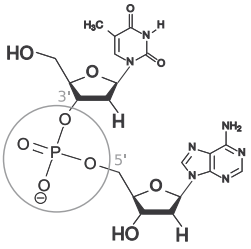

Figure 3.2: A phosphodiester bond (inside the circle) between two nucleotides: thymine (3') and adenine (5').

Proteins are large molecules formed by chains of amino acids. These chains *fold* into characteristic three-dimensional shapes. The vast majority of organisms use the same set of 20 amino acids to construct their proteins. Different combinations of these 20 basic building-blocks create an enormous variability in the shape and function of proteins. The exact amino acid sequence determines the shape and the shape determines the function of a protein. The information about the sequence of amino acids in a protein is stored in the specific part of DNA called a *gene*. Accordingly to the central dogma, each gene can be transcribed into mRNA that serves as a template for protein production. It may happen that information stored in DNA is altered: one or more of the bases can change, be deleted or inserted. This results in the change of an amino acid chain and is called a *mutation*. Not all mutations influence protein function but many do. Sometimes the change of one amino acid to another is sufficient to completely disrupt protein function. Often, however, the effect of a single mutation is subtle or silent and it is rather the complex interplay of

many mutations that leads to the change of protein function. Not all mutations are bad, some allow proteins to better function in a changing environment. From the point of view of a virus, mutations that lead to drug resistance are beneficial as they increase viral *fitness* (i.e., chance of surviving under the drug pressure).

Apart from mutations, protein function can be altered in yet another way: by *post-translational modifications* (PTMs). These are permanent or temporary chemical modifications of the amino acids that build the protein. PTMs are very important in regulating actions performed by proteins. For instance, it is often the case that the addition of a phosphate group to the amino acid serine triggers on or off the entire protein. Post-translational modifications will be discussed in more detail at the end of this chapter.

3.2 Viruses

In 1889 Martin Beijernick identified an infectious particle that was causing mosaic disease in tobacco. The particle transpired to be much smaller than any type of bacteria and this led the researcher to the discovery of a new type of organism that he named a *virus*. Viruses are very simple: in fact, they are one of the simplest organisms found in nature. When outside the host cell, they display no signs of life, being basically nothing more than a piece of genetic information enclosed by a protein capsid. However, once a virus enters its host cell, it releases the genetic content, hijacks the native cellular machinery and harnesses it to produce new copies of itself. The process of producing new viral particles is called *viral replication*. More than 5000 different viruses have been described by now and it is commonly accepted that many still remain undiscovered [20]. They infect all types of living organisms: from plants to animals and from bacteria to archea. Viruses are also very abundant and can be found in nearly every ecosystem on Earth. Not all animal and plant viruses cause harmful diseases but the many of them do. Viral replication makes intensive use of cellular resources and newly-produced particles damage the cell while leaving it. This leads to various pathologies and eventually causes cell death [20, 69].

When it comes to the diversity of genetic cycles, the world of viruses is remarkably rich. A viral genome is characterized by the type of nucleic acid, the shape, the strandedness and the sense. These properties vary from virus to virus and create many different configurations. *Retroviruses*, one of the families of viruses, store their genetic information in the form of RNA and use a viral enzyme, the reverse transcriptase, to transcribe this RNA into viral DNA (vDNA). This is an exception from the central dogma of molecular biology. The vDNA can be incorporated into the host-cell genome and expressed to produce new viral particles. The spectrum of retrovirus-caused diseases is wide and includes tumors, chronic infections (e.g., arthritis), myelopathies

and immune deficiency syndromes. Below I discuss a particular member of this family, the HIV-1 virus which causes AIDS, a severe immune deficiency syndrome.

3.3 Human immunodeficiency virus and AIDS

Human immunodeficiency virus (HIV) is a member of the *Retroviridae* family, *Orthoretrovirinae* subfamily, genus *Lentivirinae*. It is the causative agent of acquired immune deficiency syndrome (AIDS). The virus primarily infects various cells of the human immune system such as T-helper cells, macrophages and dendritic cells. The population of the $CD4^+$T-helper cells is particularly affected.

Typically an acute HIV-1 infection is manifested in the form of a transient symptomatic illness associated with high levels of viral replication (see Figure 3.3). A rather expensive virus-specific immune response usually causes flu-like symptoms including fever, maculopapular rash, weight loss, arthralgia, lymphadenopathy, pharyngitis, myalgia, malaise, oral ulcers and aseptic meningitis. The symptomatic phase lasts approximately from one week to 10 days. Due to the non-specific symptoms, AIDS is difficult to diagnose at this early stage of infection [57, 74].

After the disappearance of the symptomatic phase, the disease progresses while at the same time reducing the effectiveness of the immune response. This, in turn, facilitates the invasion and growth of various pathogens and leads to opportunistic infections. Also the probability of developing tumors increases dramatically. Common opportunistic diseases affecting HIV-positive patients include pulmonary infections (pneumocystis pneumonia, tuberculosis), gastrointestinal infections (esophagitis, candidiasis, diarrhea), neurological and psychiatric conditions (toxoplasmosis, progressive multifocal leukoencephalopathy, AIDS dementia complex, bipolar disorder, manias). The most common tumors result from co-infection with an oncogenic DNA virus such as Epstein-Barr virus, Kaposi's sarcoma-associated herpesvirus or human papilloma virus [57]. While in the areas where anti-viral therapies are in use, the incidence of many AIDS-related conditions has decreased, malignant cancers have become the most common cause of death overall [18].

HIV transmission occurs via the direct contact of a virus-containing bodily fluid with a mucous membrane or the bloodstream. HIV particles are present in blood, semen, vaginal fluid, pre-seminal fluid and breast milk. Infection occurs primarily via anal, vaginal or oral sex, blood transfusion, contaminated needles, mother-foetus transmission, perinatal transmission and breastfeeding [57].

As it is the case in the majority of syndromes, pathophysiology of AIDS is complex. It is, however, the depletion of $CD4^+$T-helper lymphocytes that is

primarily responsible for the development of AIDS [79]. During the state of immune activation, the $CD4^+$ T-cells are more susceptible to apoptosis and this seems to be the main factor leading to their depletion. In the first few weeks post-infection, during the so-called acute phase, HIV replicates intensively in the $CD4^+$ cells [55]. The sub-population of the CCR5 co-receptor-expressing $CD4^+$ lymphocytes is affected first. The CCR5-expressing cells are located mainly in the intestinal mucosa and other mucous membranes while only a small fraction of bloodstream $CD4^+$ cells express the co-receptor [22, 21]. At this stage of infection, viral replication meets a vigorous immune response which is manifested in flu-like symptoms (see Figure 3.3). Eventually the immune system begins to control the viral population, symptoms disappear and the clinically-latent phase begins. At this stage, the population of $CD4^+$ cells is significantly depleted but still sufficient to combat life-threatening invaders [57, 74].

Constant HIV-replication triggers on the generalized immune activation that persists throughout the latent phase. Several viral proteins that are produced induce the release of pro-inflammatory cytokines and other particles that promote and sustain the immune activation. In addition to this, the depletion of mucosal $CD4^+$ lymphocytes results in the breakdown of the immune surveillance system in the mucosal barrier. This facilitates the invasion of various microbes that constitute the normal flora of the gut leading to the first opportunistic infections [21]. The thymus continuously produces new T-cells replacing the lost ones, but these newly produced lymphocytes are quickly infected by HIV particles. Gradually the thymocytes also become infected which results in the production of the already-infected lymphocytes. The immune response breaks down and the organism is no longer able to cope with the invaders [8]. Figure 3.3 presents the relationship between the number of viral particles and the $CD4^+$T-cells -level during an untreated HIV infection.

The origin of the virus has been the subject of debate for a long time. Viral archeology sheds light on the beginnings of the HIV pandemic. Although the first cases of AIDS were reported in 1981, the virus must have existed long beforehand. Currently, it is commonly accepted that sometime between 1902 and 1921 the simian immunodeficiency virus (SIV) acquired mutations that allowed it to infect humans [61]. It was probably through exposure to the blood of chimpanzees butchered for meat that the first humans became infected. Studies summarized by Sharp and Hahn [109] suggest that the pandemic started in Léopoldville (now Kinshasa), Democratic Republic of the Congo and findings reported by Worobey et al. [123] support this suggestion.

Despite numerous efforts undertaken by the scientific community, the HIV pandemic still remains a serious problem that affects both the developed and the developing countries. Worldwide, about 7,000 new infections are reported every day and it is estimated that more than 33.2 million people are infected with HIV-1. In some regions a strong gender-bias can be observed among the infected individuals [116]. For instance, in sub-Saharan Africa 61% of the

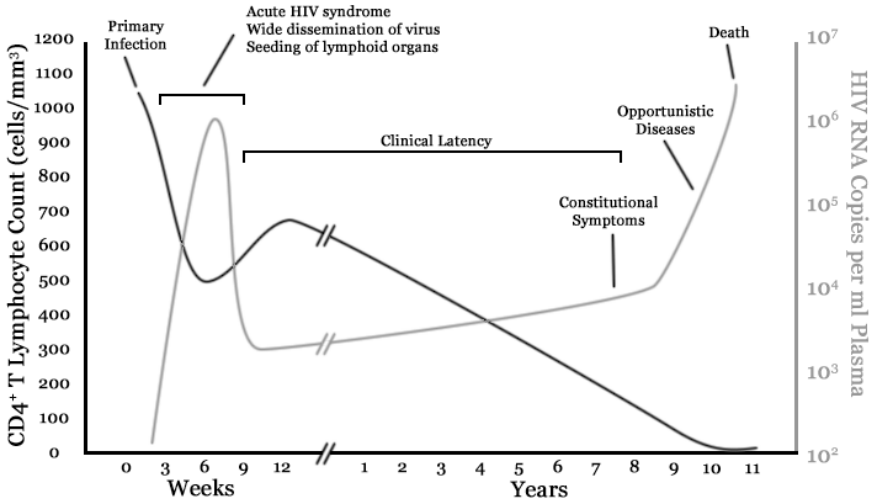

Figure 3.3: A graph showing the relationship between the number of HIV copies and the $CD4^+$T-cell count during the course of an untreated infection. Also, the stages of clinical symptoms are presented. Adapted from [2], modified after [57].

people living with HIV are women. Sexual contact remains the main route of infection and in the aforementioned sub-Saharan Africa every fourth woman is infected by the age of 22 [49, 116].

The human immunodeficiency virus particle is roughly spherical, with a diameter of about 120 nm. The virus is composed of two copies of positive (+), single-stranded RNA enclosed by a conical capsid. The RNA is bound to p6/p7 protein complexes and contains viral genes that encode 19 viral proteins [57]. Apart from the viral RNA, the capsid contains viral proteins (see Figure 3.5): reverse transcriptase (RT), protease, ribonuclease and integrase. The capsid built primarily of p24 core proteins is surrounded by a matrix composed of p17 proteins. The matrix is, in turn, protected by the viral envelope composed of two layers of phospholipids. The envelope is in fact a part of the cellular membrane taken from the host cell while budding off. About 70 copies of viral Env complexes embedded in the envelope protrude through the surface. Each Env complex is a tetramer that consists of an external glycoprotein gp120 trimer and a trans-membrane glycoprotein gp41 monomer. Both glycoproteins play a crucial role in attaching and fusing the virus with the host cell.

The organization of the HIV-1 genome (Figure 3.4) which is nearly 10kB is similar to that of other members of the *Lentivirinae* genus. The three main genes *gag*, *pol* and *env* encode the main structural and enzymatic proteins. Apart from these, there are an additional six open reading frames between the *pol* and *env* genes: *tat*, *rev*, *nef*, *vif*, *vpr* and *vpu*. The two Tat proteins are transcriptonal transactivators for the LTR promoter, the Rev protein is in-

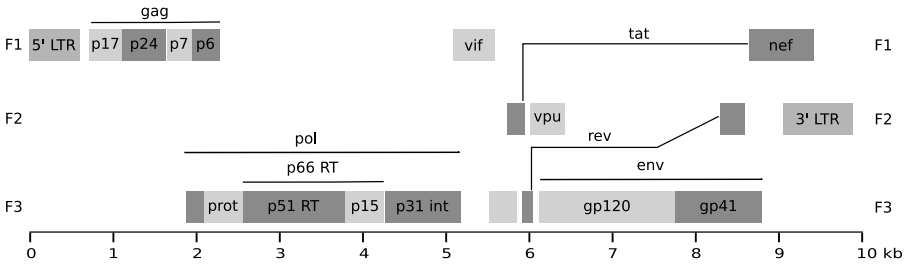

Figure 3.4: The organization of the HIV-1 HXB2 genome (simplified). Open reading frames shown as rectangles. There are three different reading frames available for transcription: F1, F2 and F3. Long Terminal Repeats shown on the sides (grey). Compiled from [57, 93].

volved in shuttling the RNA from the nucleus and the cytoplasm, the Nef protein downregulates the major viral receptor CD4 as well as MHC class I and II molecules. The Vif protein inhibits the APOBEC3G cellular deaminase of DNA-RNA hybrids, which has implications for the emergence of mutants. The Vpr protein arrests cell division at the G_2/M phase and the Vpu protein influences the process of budding off from the host cell [57, 93]. Apart from these functions, the viral genome contains also the Psi element that is involved in packing the viral genome. The whole genome is flanked by two LTR elements that contain switches that control production of new viral particles. These switches are triggered by both the viral and the cellular proteins. The LTR elements also protect viral DNA integrated into the host genome [57, 93].

The HIV life-cycle begins with the adsorption of viral receptors (Env complexes) to the surface of the host cell such as a T-cell or macrophage. This is followed by the fusion of the membranes and the entry of the viral capsid into the cell. Inside the host cell, viral enzymes and single-stranded RNA are released from the capsid. During the microtubule-based transport to the nucleus, the RT enzyme transcribes single-stranded viral RNA into a double-stranded viral DNA. In the nucleus the vDNA is incorporated into the host chromosome with the help of another viral enzyme, HIV integrase. Once integrated, the virus may remain dormant for several years; this is so-called *latent stage of infection*. It is not yet fully understood what triggers the activation of the dormant virus. Shortly after the activation, the integrated vDNA is transcribed into mRNA with the help of native cellular machinery. This mRNA is subsequently spliced and both the viral proteins and the mRNA copies of the whole viral genome are produced. This single-stranded viral RNA binds to a certain viral proteins and is packaged into new capsids. The capsids undergo maturation, then are packaged into the envelope and bud off from the host cell carrying off a small part of its cellular membrane. The newly produced viruses are fully functional and capable of infecting new host cells. Thus, the HIV life-cycle is completed [57, 74, 68].

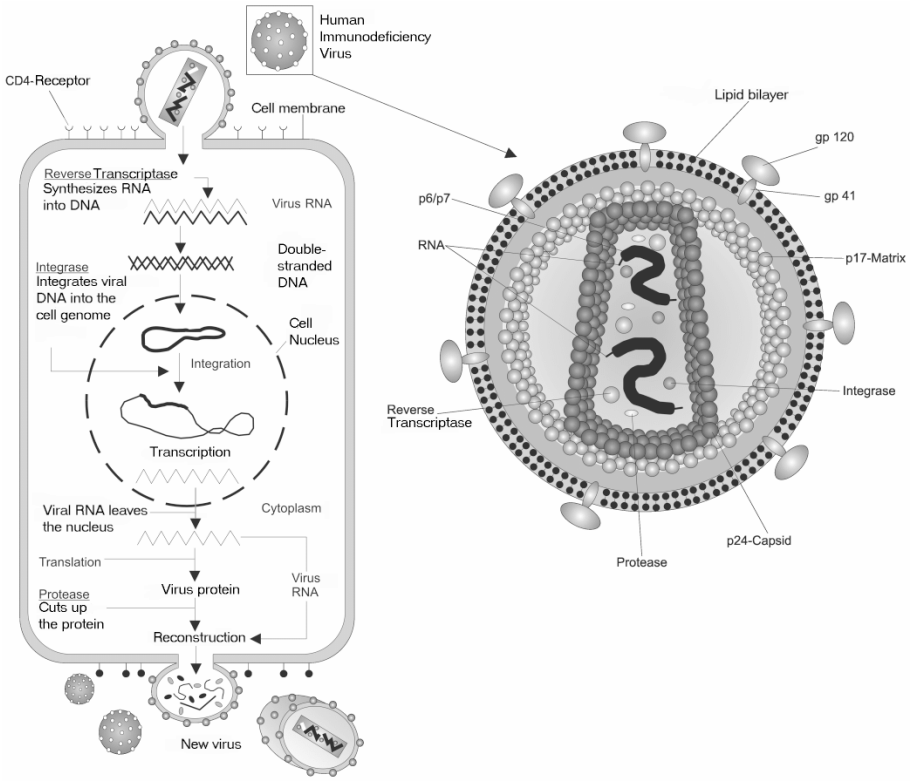

Figure 3.5: Life-cycle and structure of the HIV. Adapted from [2], modified after [57].

Two major types of the virus have been recognized: HIV-1 and HIV-2. The former is more virulent, easier to transmit and responsible for the majority (>80%) of HIV infections globally [26]. The less virulent and less transmittable HIV-2 is present mainly in West Africa [57, 61, 74, 26]. It is responsible for a minority of infections and will not be discussed in this thesis. Three major groups of HIV-1, termed M, N and O, have been recognized [61]. Each of them was independently derived from simian immunodeficiency virus (SIV) which is endemic in chimpanzee (*Pan troglodytes*) populations inhabiting west Central Africa. While groups N and O are relatively rare, the M group is responsible for about 95% of HIV-1 infections globally [109]. Group M can be further divided into subtypes A, B, C, D, F, G, H, J and K. Simultaneous infections with more than one subtype leads to the emergence of hybrids, the so-called "circulating recombinant forms". Throughout the papers included in this thesis we discuss mainly the subtype B virus that is the most common one in Europe and North America [119, 57, 109]. The wild-type strain HXB2 that we use as a reference also belongs to this subtype.

3.4 HIV-1 reverse transcriptase and reverse transcription

The transcription of viral RNA to vDNA is crucial for viral replication. The process is called *reverse transcription* and is mediated by the viral enzyme RT that catalyzes both the RNA-dependent and the DNA-dependent DNA polymerization. In simple words, RT "reads" the viral mRNA template and synthesizes a complementary strand of vDNA by adding one by one the appropriate nucleotides. Once the first strand of vDNA is ready, RT synthesizes the complementary strand to form a proper vDNA double-helix.
The HIV genome consists of two molecules of positive sense single stranded RNA with a 5' cap and 3' polyadenylated tail. During transport to the nucleus, still in the cytoplasm, the vRNA is reverse transcribed into vDNA. The process is illustrated in Figure 3.6 and involves the following main steps [62, 84, 11, 81, 42]:

1. A specific cellular tRNA acts as a primer and hybridizes to a complementary part of the virus genome called the primer binding site (PBS). The tRNA provides a 3' hydroxyl group that is necessary to initiate transcription. Reverse transcription takes place in the $3' \rightarrow 5'$ direction.
2. While the strand of DNA complementary (cDNA) to the vRNA U5 and R region is being synthesized by RT, an RNAseH domain of the enzyme degrades the 5' end of the vRNA. Both the U5 and the R region are removed. The U5 is a 5' unique sequence, the recognition site for viral integrase. The R region is a direct repeat found at both ends of the vRNA molecule.
3. Once the synthesis of the 3' end of the complementary DNA is finished, the primer is translocated to the 3' end of the vRNA where the newly synthesized DNA strand hybridizes to the complementary R region of the vRNA.
4. The first strand of cDNA is extended further and vRNA (except PP site) is subsequently degraded by RNAse H.
5. Once the strand is completed, second strand synthesis is initiated from a new primer, the PP fragment of the viral RNA. The tRNA makes it possible to synthesize the complementary PBS site and is degraded by RNAse H.
6. After another translocation both the DNA strands hybridize at their PBS sites.
7. Both strands are completed by the DNA-dependent DNA polymerase DNAP activity of the RT and are now ready to be incorporated into the host's genome by the enzyme integrase.

The RT enzyme of HIV-1 is a heterodimer consisting of two subunits: p51 (51 kD) and p66 (66 kD). The p51 subunit consists of 440 amino acids and is a product of a proteolytic cleavage of the p66 subunit. The latter consists of 560 amino acids and shares 440 of them with p51. Despite the sequence similarities, these two subunits differ substantially in the performed functions.

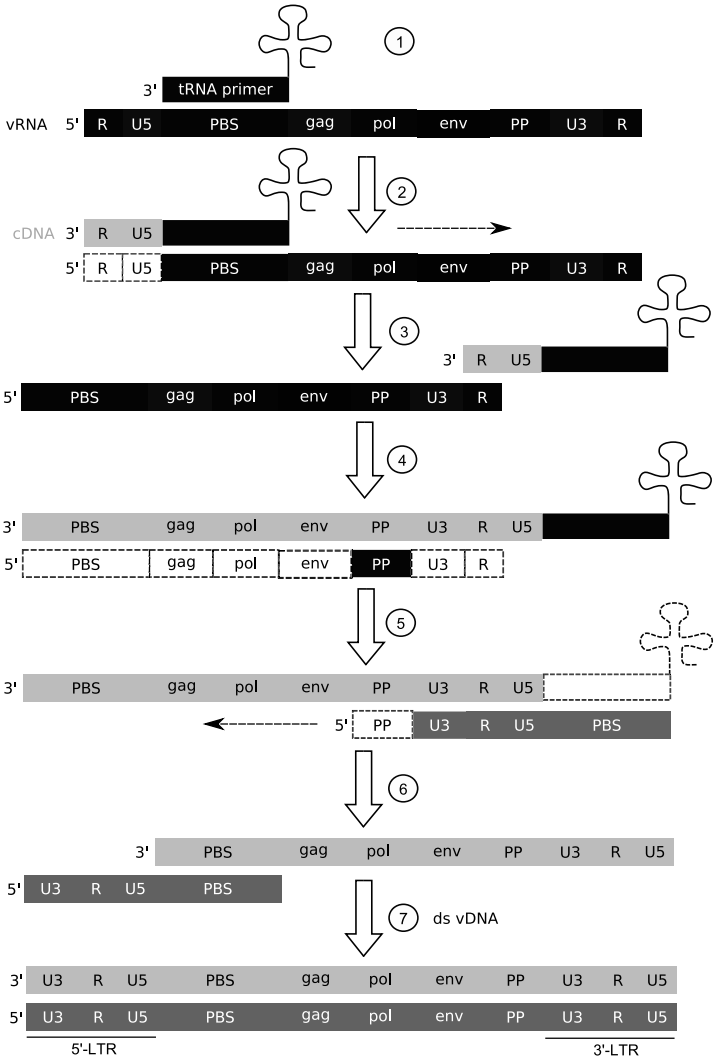

Figure 3.6: Reverse transcription. A simplified schematic representation of the reverse transcription process is shown. U3 – 3' unique sequence; PBS – primer binding site is a site complementary to the incoming tRNA; PP – polypurine section serves as the initiation of the second strand synthesis; R – R region, a direct repeat region; gag, pol and env – HIV genes. Stages discussed in the text, page 3.4.

While p66 contains the DNA-binding groove, the RNaseH domain and is catalytically active, the shorter p51 lacks any enzymatic activity and serves as a scaffold for p66. The ternary structure of the p66 subunit is traditionally compared to a right hand with a thumb, a palm and a fingers domain cf., Figure 3.7 and Figure 3.8 [58, 95, 96, 60].

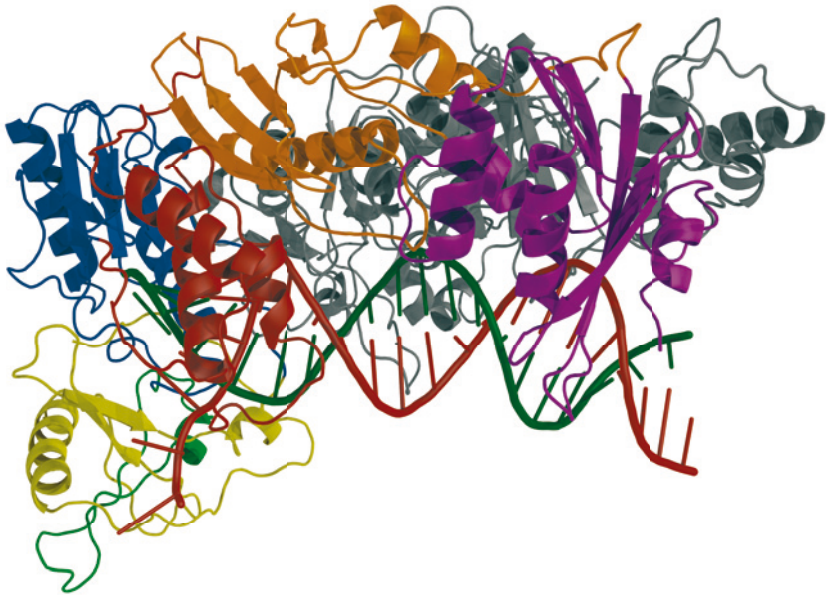

Figure 3.7: General structure of the HIV-1 RT enzyme. p51 subunit shown in grey, p66 subunit consists of fingers domain (yellow), thumb domain (red), palm domain (blue), connection domain (orange) and RNaseH domain (magenta). DNA template shown in red, DNA primer in green. After 1RTD PDB structure [58].

The RNA-dependent DNA polymerization catalyzed by the reverse transcriptase involves the following four steps: 1) template/primer binding to the enzyme; 2) binding of dNTP and bivalent cations (Mg^{2+} or Mn^{2+}) to the RT-template/primer complex; 3) formation of a phosphodiester bond between the 3'-OH primer terminus and the α-phosphate of the dNTP; 4) translocation of the elongated DNA primer from the dNTP binding site to the primer site or release of the template/primer complex [57, 80, 81, 104].

Research reported in [104, 80] suggests that a large conformational shift and rotation of the fingers subdomain towards the thumb subdomain occurs upon the binding of dNTP and bivalent cations. The active centre of the enzyme is located in the palm subdomain which contains three catalytically active residues: Asp 110, Asp 185 and Asp 186. The residues are embedded in a hydrophobic region (cf. e.g., [118]). Similarly to other reverse transcriptases, a highly-conserved Y*X*DD motif formed by the residues Asp 185, Asp 186, Tyr 183 and Met184 is present in the HIV-1 RT [53] (see Figure 3.8).

It has been suggested that during polymerization the fingers subdomain and both alpha-helices of the thumb form a clamp that holds the nucleic acid in the right place over the palm. The template/primer interactions occur between the sugar-phosphate backbone of the DNA/RNA and residues of the p66 subunit [53, 104, 80].

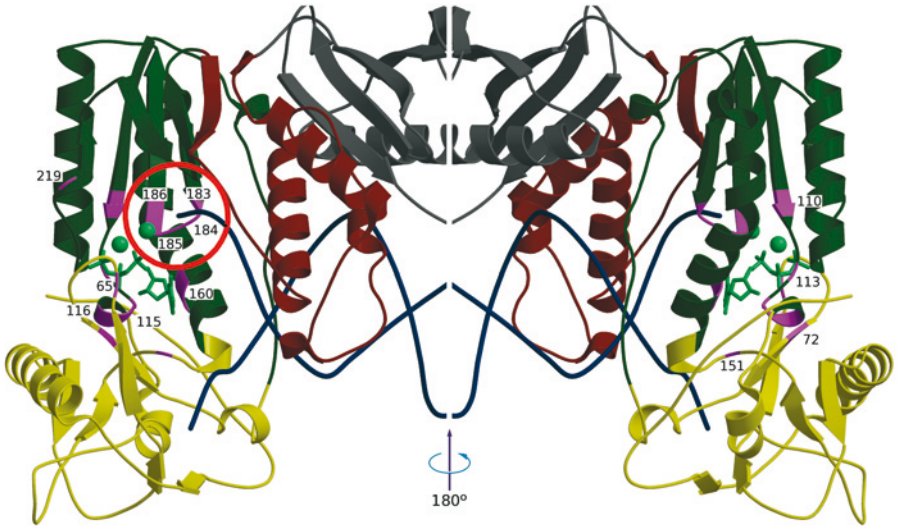

Figure 3.8: Structure of the HIV-1 RT enzyme fragment (PDB structure 1RTD [58]). Thumb domain (red), finger domains (yellow) and palm domain (green) constitute p66 subunit. Residues constituting the dNTP binding pocket are shown in magenta, Y*X*DD motif members are within the red circle. The incoming dNTP and two magnesium ions are shown in light green. (For clarity, the structure to the right is rotated 180°).

3.5 Anti-HIV drugs

Currently, more than 30 drugs based on some 20 different active compounds are used in anti-HIV-1 therapy [117]. Thanks to the existence of anti-viral drugs, AIDS is no longer considered to be a fatal disease but rather a chronic non-lethal condition. The degree of AIDS severity differs from one part of the world to another and greatly depends on the availability of treatment. Historically, the first drugs used in AIDS treatment targeted two important viral enzymes: reverse transcriptase and protease. While there exist chemical agents targeting different molecules and different stages of viral life-cycle, RT inhibitors and protease inhibitors remain the two major classes of drugs used in the highly-active antiretroviral therapy (HAART). Among the compounds approved for AIDS treatment there is also one fusion-inhibitor, one entry inhibitor and one integrase inhibitor [117].

3.5.1 Reverse transcriptase inhibitors

Zidovudine (AZT), the very first drug used in anti-HIV treatment was targeted against the RT enzyme and it gave rise to the whole family of *nucleoside RT inhibitors* (NRTI) [57]. The mode of action is common to all NRTI drugs: they mimic the actual substrates of the enzyme, the dNTPs, but lack the 3'-OH group in the ribose ring and therefore terminate DNA chain elongation

by making the creation of a phosphodiester bond impossible [53, 35, 80] (see Figure 3.9). Currently 13 different RT inhibitor-based drugs are in use and they are based on six active compounds: abacavir, didanosine, emtricitabine, lamivudine, stavudine and zidovudine [116]. In this work, we investigate resistance to all these except emtricitabine, for which no resistance-data was available. All of the previously mentioned drugs mimic nucleosides and in order to become active they need to be phosphorylated by cellular enzymes. There exist also one nucleotide RT inhibitor (NtRTI), Tenofovir, that does not need to be converted to its active form. As the mode of action is very similar for both groups, they are often classified together as NRTIs. Chemical structures of NRTIs and NNRTIs are presented in Figure 3.9.

There exist yet another group of RT-targeted drugs that inhibits the enzyme in a completely different way, namely by binding at a specific site on the RT surface and inhibiting the motility of its protein domains. This, in turn, disrupts the enzymatic function. The drugs acting in such a way are called non-nucleoside RT inhibitors (NNRTI) . Their destination-site on the surface of the enzyme is called the *NNRTI-binding pocket* [80, 57].

3.5.2 Other inhibitors

There exist other classes of anti-HIV drugs. *Protease inhibitors* target another important viral enzyme, HIV protease. Protease is also crucial for viral replication as it cleaves long non-functional polypeptides into fully functional viral proteins such as enzymes and capsid proteins. Viral DNA synthesized by the RT is incorporated into host chromosomal DNA with the help of *HIV integrase*, which is also a target for anti-viral drugs [57].

HIV entry into a cell is a complex sequence of events. First, the viral surface protein gp120 binds a CD4 receptor on the surface of the host cell. This is followed by a conformational change of gp120 which increases its affinity to a co-receptor while at the same time exposesing another viral protein, gp41. Now gp120 binds to a co-receptor (CCR5 or CXCR4) and gp41 promotes the fusion of cellular and viral membranes. This enables the entry of the viral core into the cell [37, 108, 43]. Currently one *fusion inhibitor* targeting gp41 and one *entry inhibitor* that blocks CCR5 receptor are available for the treatment of HIV. There is also a number of drugs that combine different active compounds to achieve therapeutical effect [117, 116].

3.5.3 Resistance to drugs

The initial hope following the introduction of Zidovudine, the first anti-HIV drug, was quickly dashed by the emergence of drug-resistant viral strains. It has been estimated that 10^9–10^{10} virions are produced every day in an infected individual [97, 38]. The average half-life of a virion is 2 days. Like other RNA polymerases, the HIV RT lacks a proof-reading activity and about 35 muta-

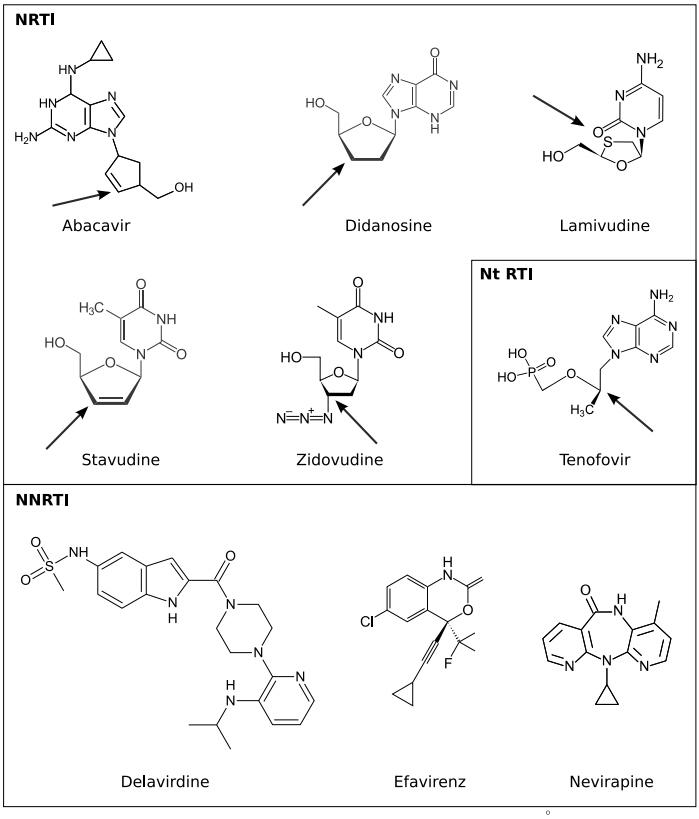

Figure 3.9: Different groups of anti-viral drugs. In the case of NRTI and NtRTI drugs, an arrow points out the main structural difference that is responsible for therapeutical effect. Drug structures after: [1] and manufacturer-issued data.

tions per million nucleotides are introduced during one cycle of replication [97, 87, 62]. In addition to this, the cellular APOBEC3G DNA deaminase is packed into newly synthesized virions where a Vif protein inhibits its function. However, the low activity of APOBEC3G leads to additional G→A mutations [5]. The exceptionally high rate of mutation, together with the high replication rate of the virus and the high rate of RT-mediated recombination, leads to the rapid emergence of various mutations. This results in the very high genetic variability of the virus. These mutations give rise to new viral subtypes and to drug-resistant strains. Potentially, a virus resistant to any single HIV-inhibitor is produced many times a day in an infected organism [90, 57, 68].

The drug resistance mutations are usually deleterious, i.e., they decrease viral *fitness*. Therefore in the absence of drugs, mutant sub-populations are very small compared to the wild-type virus. This situation changes rapidly after administering an anti-viral drug. Now, the resistant viruses have much higher chances to replicate and they quickly dominate in the viral population.

When the drug pressure is removed, many resistance-mutations revert back to the wild type but some may still be advantageous (or not-so-deleterious) and they are observed persisting in the viral population.

It is tempting, though, to administer more than one drug at the same time hoping that the viral population will not be able to survive such a strong pressure. Indeed, in 1999 the so-called highly-active antiretroviral therapy (HAART) was introduced, usually combining three drugs: two RT inhibitors and one protease inhibitor [57]. Unfortunately, it transpired that HIV is able to overcome this type of therapy and that viral strains emerge that are resistant to all the administered drugs. The situation is aggravated even further by the fact that mutations that cause resistance to drug A, also cause resistance to drug B. This phenomenon is called *cross-resistance* and undermines many efforts to replace one drug in the mixture with another (see, e.g., [29]).

Despite these difficulties, AIDS is not regarded a life-threatening condition anymore. Thanks to the existence of efficient therapies, HIV-positive patients are considered to be chronically ill. The high efficiency of the therapy is achieved by constant monitoring of the mutations that emerge in the viral population and responding with the appropriate, optimal combination of drugs. To be this "one step ahead of the virus" is neither easy nor cheap. The most obvious way of keeping track on the emerging drug-resistant strains is to perform direct *in vitro* assays. This approach is called *phenotyping* [57, 13, 36].

In phenotyping, the viral replication rate is measured directly in the cell cultures. A blood sample is taken from a patient, the cell cultures are infected and the replication rate is measured at different drug concentrations. The results are given as values relative to the wild-type virus replication rate. This procedure is time consuming and expensive [57, 13, 36].

Genotyping is another approach to measuring replication rate. A blood sample is taken and viral genetic material is isolated. This is followed by PCR amplification and sequencing. This method gives the exact genotypes of viruses present in the patient's blood. While quick and cheap, genotyping does not give an explicit information on the values of replication rate. The results of genotyping have to be analyzed further and the rate of replication is determined indirectly [57, 13, 36].

Since 1999, sets of rules for resistance prediction from genotyping-results are semiannually issued [25]. The rules are based on the knowledge of many experts around the world. However, even for a group of experts it is difficult if not impossible to perform the detailed in-depth analysis of all the genotyped viruses. It is even difficult for a human to analyze resistance that emerged as a result of a combination of five or more mutations [82, 9, 30]. Since 1999, numerous tools and interpretation systems have been proposed to facilitate genotyping result interpretation. Many of the systems make use of machine-learning algorithms (ML) [105] such as: support vector machines [14], fuzzy-logic [33], decision-trees [13] or neural networks [72, 36, 63]. In Papers I – III we apply novel machine learning algorithms to predict HIV resistance on

the basis of genotyping results. We also perform an in-depth analysis of the observed mutation patterns that lead to resistance.

3.6 Post-translational modifications of proteins

Protein synthesis is a complex, multi-stage process in which cells build proteins. Initially, the term *protein synthesis* was synonymous with *protein translation*, but it turned out that translation is just one step of the whole process. From the point of view of chemistry, a protein is just a chain of amino acids [6, 113]. There are 20 different amino acids that are used as protein-building blocks by the majority of higher organisms. There are organisms (e.g. certain bacteria) that use non-standard amino-acids to build their proteins, but these are a rather exceptional cases [6]. The exact order of amino acids in a protein is termed the *primary structure*. Proteins, however, are not just linear chains of amino acids. They fold into regularly repeating local structures that are stabilized by hydrogen bonds. Alpha helices, beta-sheets and turns are the most common examples of such structures. This level of organization is termed the *secondary structure*. The spatial relations between secondary structures determine an overall shape of a protein, its *tertiary structure*. These relations are usually stabilized by global interactions such as the formation of a hydrophobic core, disulfide bonds, salt bridges and also hydrogen bonds. There is a number of common, stable tertiary structures that appear in a large number of proteins, sometimes independently of their function and evolutionary kinship. Some proteins may form protein complexes and this is the highest level of protein organization known as the *ternary structure* [113].

The primary structure of a protein as obtained from a sequencing project (e.g. human genome project) is not sufficient to provide the full explanation of its various functions and regulatory mechanisms. The vast majority of the proteins encoded in any genome undergo numerous post-translational modifications (PTMs) that alter their function. These modifications constitute an important level in the regulation of protein function [6, 113]. There are many different types of PTMs but they can be classified into four main groups [113, 6, 114, 106, 17, 71]:

1. modifications that involve the addition of a functional group, such as the addition of an acetyl group, methyl group or a phosphate group.
2. changes of the chemical nature of amino acids like citrullination – conversion of arginine to citrulline.
3. modifications involving addition of other proteins or peptides, e.g., covalent linkage to the SUMO protein or ubiquitin.
4. modifications that involve structural changes such as proteolytic cleavage, formation of disulfide bridges between cysteines or racemization of proline.

The first two groups of modifications are the subject of Paper IV and we will discuss it now a bit more. A *functional group* is a term taken from organic chemistry where it denotes a specific group of atoms within a group of molecules that are responsible for the characteristic chemical reactions of those molecules. There exist a large number of PTMs involving addition or removal of many different functional groups. Possibly the most important and best known are [114, 106, 17, 71]:

- *acetylation* and *deacetylation* – the addition or a removal of an acetyl ($-COCH_3$) group either to the N-terminal part of a protein or to lysine residues. This is one of the modifications affecting histones and plays an important role in the regulation of gene expression.
- *methylation/demethylation* is the addition/removal of a methyl ($-CH_3$) group, usually at lysine or arginine residues. This type of modification also plays a crucial role in the regulation of gene expression by modifying histones and regulating the state of chromatin.
- *glycosylation* is the addition of a glycosyl (sugar) group to asparagine, hydroxylysine, serine or threonine. The final product of glycosylation is called a glycoprotein. This is one of the most important modifications. In mammals glycoproteins play a role in white blood cell recognition. Also, the molecules of the major histocompatibility complex responsible for the recognition of pathogens are glycoproteins. Therefore glycosylation plays an important role in many processes, such as the rejection of transplants, autoimmune diseases and the sperm-egg interaction. Also many viral capsid proteins, (e.g. in the influenza virus) are glycosylated.
- *hydroxylation* is the addition of a hydroxyl group (-OH). This modification affects proline and hydroxyproline is the final product. Hydroxyproline is an important building-block of collagen which, in turn, builds the connective tissue.
- addition of a prenyl or an isoprenyl group called *prenylation* and *isoprenylation* respectively. These modifications have been shown to be important for protein-protein interactions.
- *phosphorylation* is the addition of a phosphate group ($-PO_4$). This modification affects histidine, serine, threonine and tyrosine. Phosphorylations are a very important class of modifications and it is estimated that 10-50% of all proteins are phosphorylated. The addition of a phosphate group is catalyzed by *kinases* and the reverse reaction called *dephosphorylation* is catalyzed by another group of enzymes, *phosphatases*. Phosphotylations/dephosphorylations trigger many important enzymes on and off and affect almost all the processes taking place in a living cell or organism: from memory formation to gene expression.

The above list provides just some examples of PTMs and gives an idea of how complex the regulation of protein function is. The majority of PTMs

are catalyzed by enzymes: proteins that recognize specific *motifs* in a protein sequence or in a protein structure and modify the appropriate amino acids [6]. There are many ways to study PTMs, the most common being mass spectrometry and 2D chromatography. Also a number of methods exist that involve transferring cellular proteins into a membrane to search for post-translational modifications using specific probes. These techniques are called Eastern-blotting [113, 17]. Since all these wet-lab procedures involve large amounts of work and a lot of resources, an idea emerged to use *in silico* computational methods to predict modification sites directly from protein sequences. These predictions can be used as a navigational aid for molecular biologists in their research.

4. Computational Background

In this chapter I give a brief introduction necessary to understand the computational aspects of the work. I begin with a definition of a model, proceed to the data representation and discuss the various aspects of model construction. First, I discuss feature selection, then I talk briefly about classification task and model-validation. I conclude with model interpretation and model generalization.

4.1 From data to model

First, let me introduce some basic terminology that computational biologists use when talking about data. Often, it is convenient to represent the outcome of an experiment in tabular form.

Name	Size	Polarity	Aromatic?	Charge	**Efficacy**
C1	28.4	medium	yes	positive	**high**
C2	25.21	low	yes	neutral	**low**
C3	7.14	low	yes	negative	**low**
C4	11.15	medium	yes	neutral	**high**
C5	20.04	medium	yes	neutral	**low**
C6	17.35	high	yes	positive	**high**
C7	29.79	high	no	negative	**high**
C8	3.24	low	no	positive	**low**

Table 4.1: *An example of a data table.*

Table 4.1 summarizes the results of a drug screening experiment. Each row in the table corresponds to an *instance* of a drug candidate. Five *features* of each instance are described by five values: name, size, polarity, aromaticity, and charge. The "Name" attribute is unique for each instance and is used for identification purposes only. In machine learning features are also called *attributes* and in statistics *independent variables* [122, 54]. Such an ensemble

of instances, each characterized by the same set of features, is called an *information system*.

Definition 1. *An information system* $\mathbf{A}$ *is a pair* $\mathbf{A} = (U, A)$*, where* U *is a finite, non-empty set of instances called the universe and* A *is a finite, non-empty set of attributes (features) where an attribute* a *is a function* $a : U \to V_a$ *that associates values from the domain* V_a *with instances from* U *for every* $a \in A$ *[122, 54, 67].*

If the outcome is known for each instance, we can add a *decision attribute* (or more decision attributes) to an information system, and we talk of a *decision system*. Formally,

Definition 2. *A decision system* $\mathbf{A_d}$ *is any information system of the form* $\mathbf{A_d} = (U, A \cup \{d\})$ *where* $d \notin A$ *is the decision attribute. The elements of* A *are called conditional attributes [122, 54, 67].*

The information system presented in Table 4.1 is a decision system. "Efficacy" is the decision attribute while all the remaining attributes are called *condition attributes*.

The discussed attributes are somewhat different in the type of their values. "Name", "Polarity", "Aromatic?", "Charge" and "Efficacy" have qualitative character and are usually referred to as *categorical* or *discrete* attributes. In case of the "Polarity", "Charge" and "Efficacy" attributes the values are ordered, but no metrics can be applied – they are *ordered categorical* attributes. "Size" is a *continuous*, *quantitative* attribute. "Aromatic?" is a *binary* attribute i.e., a value that takes only two values: "True" or "False", here encoded by *Yes* and *No*.

Usually, an experiment is performed in order to increase our understanding of the surrounding world. In other words, we would like to know how condition attributes are related to decision attributes. Ideally, we would like to construct a *model* that will capture all these relations in a simple way and that will let us predict what will happen when we observe a new instance for which we do not know the outcome.

The New Oxford American Dictionary gives the following definition of a model [3]:

> ... a simplified description, esp. a mathematical one, of a system or process, to assist calculations and predictions.

Eykoff [44] defined a mathematical model as

> ... a representation of the essential aspects of an existing system (or a system to be constructed) which presents knowledge of that system in usable form.

A mathematical model usually describes a system by a set of variables and a set of mathematical functions that establish relationships between these variables [70, 54, 122].

Construction of a model usually involves several standard steps. First, data coming from experiments is represented in an appropriate form, annotated, described, normalized etc. Models should be simple, especially if they are to be interpreted by humans. Therefore it is desirable to discard the features that are not necessary to model the phenomenon in question. Various feature selection methods can be applied to achieve this task. This is followed by the discovery of a relationships between condition and decision attributes. The relationships are often complex and can be revealed with the help of computational methods. Once a model is constructed it should be validated analyzed and interpreted. All these steps are described in the following sections.

4.2 Feature extraction and feature selection

In the late 1950s Richard Bellman [15] drew attention to an interesting problem that he named the *curse of dimensionality*. The curse of dimensionality is the problem that arises when an extra dimension is added to a mathematical space resulting in an exponential increase in volume. To illustrate the problem, Bellman considers sampling a unit interval. In order to sample a unit interval with no more than 0.01 distance, the number of 100 evenly-spaced points is sufficient. An equivalent sampling of a unit square requires 100^2 points which is still not too much, but to sample a unit 10-dimensional hypercube one requires as many as 100^{10} evenly-spaced points that constitute a lattice. Thus, in some sense, a 10-dimensional cube can be said to be 10^{18} times "larger" than a unit interval (for sampling density 0.01 as set above). In other words, in a 10-dimensional space it is necessary to cover 80% of the range of each coordinate to capture 10% of the data. The curse of dimensionality problem often arises in the context of machine learning. Many machine learning algorithms can bee seen as some kind of mapping from data space to decision space. Covering data space requires computational resources, and, in the most general case, the amount of resources needed is proportional to the hypervolume of the input space [54].

In many data mining problems, the training set consists of a large number of examples for each decision class, while the number of features is usually limited and much lower than the number of instances. In contrast, an average biological data set contains only a limited number of instances while each instance is represented by a large number of features. It is not uncommon that there are only several dozen examples available versus a thousand or more features describing them. It is aggravated even further by noise inherent to experimental data. These problems turn out to be a serious obstacle when

it comes to the analysis of biological data and the application of machine learning techniques to biological problems.

Feature extraction and *feature selection* are techniques used to alleviate these problems. The goal of feature extraction and selection is to select a subset of relevant features that can subsequently be used to build robust predictive models for classification. By removing irrelevant and redundant features from the data, feature selection notably improves the performance of the majority of predictive models. The application of feature selection often significantly reduces the number of features that are to be considered by a model, thus alleviating the curse of dimensionality problem. This, in turn, speeds up the learning process and enhances the generalization capability of the model eventually leading to the improved interpretability of the results [70, 54, 122].

By feature extraction we understand selection and construction of appropriate features based on the available domain knowledge [54]. For instance, in the HIV-resistance problem (see e.g., Paper I) we did not consider the p51 subunit since it is catalytically inactive. We also created a suitable method of representing protein sequence in terms of its physicochemical properties. This resulted in a number of potentially relevant features. At this point we applied feature selection in order to select the truly relevant ones. Below we discuss feature selection in more detail.

Let us consider a typical machine learning problem. We have a set of N observations (instances), characterized by M features each. Therefore for each instance we have a feature vector: $F_i = (F_1, ..., F_j)$. By performing feature selection, we want to find a minimal subset G of features from F such that the quality of classification remains at a desired level (typically not lower than obtained when training on the original set). Let C be the set of decision classes, $G \subset F$ be the set of features after the feature selection, $P(C|G) = f_G$ be the probability distribution of different decision classes given the feature values in G and $P(C|F) = f$ be the original distribution given the feature values in F. The goal of feature selection is to find an optimal (often also a minimal) subset G such that $P(C|G = f_G)$ is equal or close enough to $P(C|F) = f$.

Let us consider the following example (Table 4.2): here the features

Instance	F_1	F_2	F_3	F_4	F_5	**C**
1	1	1	0	1	0	**0**
2	0	1	0	1	0	**1**
3	1	0	1	1	0	**1**
4	0	0	1	1	0	**0**
5	0	1	0	1	0	**1**

Table 4.2: *An example of a training set.*

$F_1, ..., F_5$ are binary. The target concept is also binary: $C = \{0, 1\}$ and it depends on F_1 and F_2 solely: $C = g(F_1, F_2)$. Additionally, let $F_2 = \overline{F_3}$ and $F_4 = \overline{F_5}$. We should notice that F_1 is indispensable for classification, as well as either F_2 or F_3 is. Both F_4 and F_5 can be discarded. Existence of two optimal subset of features: $\{F_1, F_2\}$ and $\{F_1, F_3\}$ follows immediately.

4.3 Looking for patterns

The purpose of *machine learning* is to use computers in order to learn general concepts from examples provided in a so-called *training set*. A training set is an information system or a decision system. The first informal definition of machine learning was given in 1959 by Arthur [103]:

> Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed.

Samuel wrote a program that was playing checkers with itself. Over time it was learning patterns that lead to wins or to losses. The learned patterns were then used to improve strategy and quickly the program outcompeted an experienced human player. In 1997 Tom Mitchell [83] provided a more formal definition of machine learning:

Definition 3. *A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*

Typically, a training set consists of a number of observations coming from an experiment. There are two main types of machine learning: *supervised learning* and *unsupervised learning*. In the case of unsupervised learning, a training set is an information system, but no decision attribute is given. The task is to discover patterns that naturally occur in data and to group instances according to the similarity of these patterns. Supervised learning deals with situations where the decision class information is available (given by a supervisor) and a training set is a decision system.

When building a model, it is necessary to assume that the instances constituting a training set are representative of the entire universe of possible instances so that they reflect real relationships between attributes and decision classes [86]. Statistics teaches us how to perform sampling and how to assess its quality.

4.4 Model validation

Assessment of the predictive quality of a model is a vital part of the modeling process. Having constructed a predictive model (classifier), we would like to

know what the probability is that a new, random observation will be classified erroneously. Similarly, when given many classifiers trained on the same training data, we would like to minimize the risk of misclassifying new objects by choosing the best one. We should notice that the probability of classifying a new random object erroneously is equal to the fraction of objects misclassified by the classifier. Yet the probability itself is not known, but it can be assessed in an experimental way. Let us assume that we have a random sample of objects and that the right decision-class is known for each object. Obviously, this random sample has to be independent of the training set. Such a sample is usually called a *validation sample* or, equivalently, a *validation set*. Now we can use the classifier(s) in question to classify objects from the validation set. The fraction of misclassified ones will be a good estimation of the probability we were looking for [70, 122].

The presented approach gives an accurate estimation of the classification error in the case of a single classifier. It also lets us choose the best one out of a number of classifiers. However, once the best classifier is chosen it would be a mistake to treat the error-estimates obtained on the validation set as unbiased! We should note that all our assessments are random and it is a random factor that led us to select the particular classifier as the best one. Therefore if we want to have an unbiased estimation of the predictive quality of the selected classifier, we should use another random *test set* of objects where we know *a priori* the right decision for each object. If this sample is independent of the training set and the validation set, it can be used to estimate the probability of misclassifying a random object by the best classifier selected in the previous step [70, 54, 122].

It is important to mention that ratios in-between the decision-classes observed in the training sample should be preserved in both the validation and the test set. In other words, we say that the stratification of the decision classes should be preserved. Such an approach minimizes bias in our error estimation [47, 54].

To summarize the above paragraphs, ideally the original data set is randomly divided into three sets:

- A training set used fit the models.
- A validation set used to estimate prediction error for model selection.
- A test set used to assess the generalization error of the final chosen model.

A final question arises regarding how many objects from the initial data set should be used for construction of the training set, the validation set and how many should be left for the test set? There is no ideal answer to this question, but the split-ratio between these three sets is usually close to 50% – 25% – 25% [54, 70].

In machine learning we are usually facing yet another problem, namely very few objects in the initial data set. Therefore, rather than "wasting" examples for constructing separate validation and test sets, we would like to incorporate

all the objects into the training sample. In such situation classification error can be estimated using either a *cross-validation* or a *bootstrap* approach.

Cross-validation is a simple yet powerful method of assessing the quality of a classifier. In K-fold cross-validation, the original training set is split into K parts. Each part contains approximately the same number of objects. In *stratified cross-validation* the ratios between the decision classes should be preserved as well. In the next step, K training pseudo-samples are derived from the original training sample by sequentially removing one of the K parts. Therefore every training pseudo-sample contains $K-1$ parts of the original training set. Subsequently these pseudo-samples are used to train K classifiers. Each classifier is therefore trained on $K-1$ parts of the original training sample and can be validated on the remaining part that is independent of the particular training pseudo-sample. In special case where K is equal to the number of objects in the original training set, we talk of *leave-one-out cross-validation*, also known as *jacknife* [122, 54, 70].

The sum of all the classification errors committed by the K versions of the classifier divided by the total number of objects gives a reliable assessment of the probability that the classifier will misclassify a new random observation. Such an estimation is almost unbiased and can be used to select the best classifier. Obviously, once the classifier is selected it has to be trained on the original training set before being used to classify new objects. In the K-fold cross-validation, each version of the classifier is trained on $n-\frac{n}{K}$ objects and validated using the remaining $\frac{n}{K}$ instances. Therefore the estimate of the misclassification probability is not unbiased for every sample smaller than n. The higher the K, the lower the bias and higher the variance of the estimator. This fact has to be taken into account while deciding upon the number of folds. The 5-fold, 10-fold and leave-one-out and cross-validation schemes are the most frequently used ones [122, 54, 70].

Bootstraping is another approach to the classification-quality assessment. Here a number of training pseudo-samples is constructed by resampling the original training set. In bootstrap, n elements are randomly drawn with replacement from the original training set. It is easy to see that the probability of a particular instance being drawn is $\frac{1}{n}$ and it follows that the probability of the instance not being drawn into the bootstrap sample is $1-\frac{1}{n}$. Since the bootstrap sample has the same size as the original training sample, on average $(1-\frac{1}{n})^n = 0.368$ of the objects will not be drawn. These objects can be used as a validation set while the remaining $0.632n$ objects present in the bootstrap sample will be used for training. Subsequently, N bootstrap samples are derived from the original training set and N classifiers are trained and validated. For each object from the original training sample the number of misclassifications is recorded. Obviously, only these versions of the classifier where the element was not included in the training pseudo-sample are taken into account [122, 70].

It is common to present the predictive quality of a classifier in terms of a *confusion matrix*. A simple confusion matrix for a binary classification problem is shown in Table 4.3.

		Actual		
		Positive	Negative	
Predicted	Positive	**TP**	**FP** (Type I error, p-value)	→ Precision
	Negative	**FN** (Type II error)	**TN**	
		↓ Sensitivity (=Recall)	↓ Specificity ↓ FPR	

Table 4.3: *A confusion matrix. Additionally, the relationships among the terms that are discussed in this section are presented. TP – true positive, TN – true negative, FP – false positive, FN – false negative, FPR – false positive rate. For further explanations see the text.*

In a binary classification task the objects in the test set or the validation set belong to either "positive" or "negative" decision class. Similarly, each object can be classified as either the "positive" or "negative". This gives four different possibilities: *true positive* (TP), *true negative* (TN), *false positive* (FP) and *false negative* (FN). There is a number of predictive quality measures based on these values. The proportion of correctly classified objects is called the *accuracy*:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4.1}$$

Intuitively, the less false positives the classifier yields, the more precise it is. Indeed, the *precision* of a classifier is defined as:

$$Precision = \frac{TP}{TP+FP} \tag{4.2}$$

Another important measure is the *sensitivity*, also known as either the *true positive rate* (TPR) or the *recall rate*. It tells how likely the classifier is not to miss the positive case:

$$Sensitivity = \frac{TP}{TP+FN} \tag{4.3}$$

Even a very sensitive classifier can yield a lot of false positives which is often not desirable. The fraction of false positives is measured by the *specificity* of

a classifier:

$$Specificity = \frac{TN}{FP+TN} \tag{4.4}$$

To better understand the notions of sensitivity and specificity let us consider a simple medical test. The test gives two possible answers: "healthy" and "diseased". The hypothetical confusion matrix for such a test is presented in Table 4.4. The sensitivity of the test is in this case 0.962 which is an estimate of

		Actual	
		Diseased	Healthy
Predicted	Diseased	TP = **180**	FP = **20**
	Healthy	FN = **7**	TN = **93**

Table 4.4: *A hypothetical confusion matrix for a medical test performed on 300 patients.*

the likelihood that a truly diseased patient is classified as such. The specificity, in this example equal to 0.823, estimates the probability that a non-diseased patient will be classified as such. Ideally, one would like to have a classifier that is at the same time sensitive and specific, i.e., that classifies healthy patients as healthy and diseased ones as diseased. In practice this is rarely possible and there is always a trade-off between these two values. Let us notice that the test that classifies every object as "positive" has maximal sensitivity but zero specificity. Eventually, observe that $(1 - specificity)$ is an estimate of the likelihood that a healthy patient will be classified as diseased.

In many classification problems, different types of classification errors have different costs. For instance, it is often more desirable to classify a healthy patient as diseased than the other way round. Therefore sometimes the so-called *cost matrix* is used in order to shift the balance between specificity and sensitivity. It is convenient to visualize sensitivity versus $(1 - specificity)$ for a number of cost matrices where the costs are modified systematically. Such a visualization is called the *receiver operating characteristics curve* (ROC) [52]. Equivalently, the ROC curve can be constructed by plotting the fraction of true positives (TPR) vs. the fraction of false positives (FPR) defined [122] as:

$$FPR = \frac{FP}{FP+TN} \tag{4.5}$$

An example of an ROC curve is given in Figure 4.1. An ROC curve is derived from a number of classifiers and it contains all information provided by a single confusion matrix [115]. The point (0, 1) corresponds to a perfect classifier and the point (1, 0) represents a classifier that is always wrong.

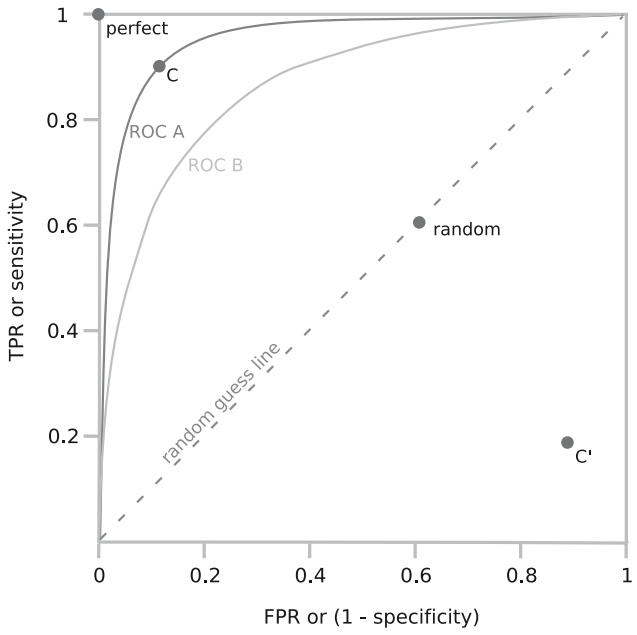

Figure 4.1: An ROC space. Grey points represent classifiers: random, perfect, hypothetical classifier C and its inverse C'. The C' classifier is a result of inverting decisions made by C. Two hypothetical ROC curves are shown. The ROC A curve represents better classificatory quality than the ROC B curve.

The accuracy measure determined by Equation 4.1 can be easily biased by a non-equal distribution of the decision classes in the data. Let us consider a test set that consists of 100 instances and that 95 of them belong to the "negative" class. Naturally, the five remaining instances are all members of the "positive" class. Now, a classifier that always predicts an instance to belong to the "negative" class will have very high accuracy (95%) [77]! Provost [92] prooved that a ROC curve is independent of the class distribution or error cost. Therefore it has been suggested that the *area under an ROC curve* (AUC) can be used as the most accurate measure of the quality of classification [115, 92].

In the papers included in this thesis, yet another method for testing predictive quality of a classifier is used. This method is called a *randomization test*. The randomization test was first introduced by Fisher in 1935 [48] and is also known as *permutation test* or a *rerandomization test* [73]. Once the classifier is chosen and validated, one may wonder what is the probability that the obtained, or even better, results could have occured by chance. The idea of the test is simple: a number of copies of the original training set is made and in each of them the values of the decision attribute are randomly shuffled. Subsequently, the examined classifier is trained on these copies and validated in the same way that it was validated after being trained on the original training set. Let us say that we are measuring AUC: we have one AUC_{orig} value that was

obtained on the original training data and we have a number N_{rand} of AUC values obtained while training the classifier on the randomized training sets. We expect these values to be normally distributed around some value close to 0.5. Our expectation is based on the assumption that when trained on random data, the classifier should be no better than just tossing a coin. Eventually, we can use the Student's t-test to determine the p-value for our AUC_{orig}.

4.5 Interpreting the model

Predictive models constructed using different machine learning algorithms provide different degrees of transparency. While some models resemble black boxes that provided with an instance on one side will return prediction on the other, there are models that give full insight into the process of classification.

The choice of classification method depends on many factors. Quality of predictions and interpretability are among the most important ones. Some methods, e.g., *neural networks* or *support vector machines* while generally accurate, in their basic version do not give much insight into the actual classification process. Decision trees, on the other hand, are easy-to-interpret but they are in general much less stable than neural networks [122]. Random forests attempt at increasing stability of predictions but do this at the price of lower interpretability. Since there are thousands of trees to analyze in a random forest, it is difficult to trace the classification process. The rough set-based approach can produce stable models based on legible rules but rough sets require discrete feature-values. In some cases discretization can be applied to fulfill this requirement, but sometimes it is necessary to work with continuous values.

To increase the interpretability of HIV drug-resistance models, in Paper I we applied our Monte Carlo feature selection method and rough set-based approach to modeling. In Paper II we proposed an enhancement to the MCFS method that allows for analysis of interdependences between significant features. The MCFS-ID method is applied before building classificatory model and is independent of the choice of classification method. By the application of the MCFS-ID method, an interpretational layer is created that can be analyzed by the domain experts in order to reveal interactions leading to a particular outcome.

4.6 Towards generality

When building a model, usually only a limited number of instances is available for training. It is necessary to assume that these instances constitute a good representation of the modeled space, called also a *universe*. This assumption is very important and the *principle of uniformity* lies in its heart

[121, 86]. Statistics provides numerous methods that help creating representative samples of the examined phenomena and provides tools to assess the quality of these samples. Having a limited number of instances, the goal is to build a model that is general enough to predict the outcome for any instance from the examined universe.

The limited number of examples, together with noise inherent to experimental data often result in too specific models that while very accurate when applied on any subset of the training set, perform much worse when presented with instances that were not included in the training sample. This phenomenon is called *over-fitting* and it is illustrated in Figure 4.2 [76, 54, 122].

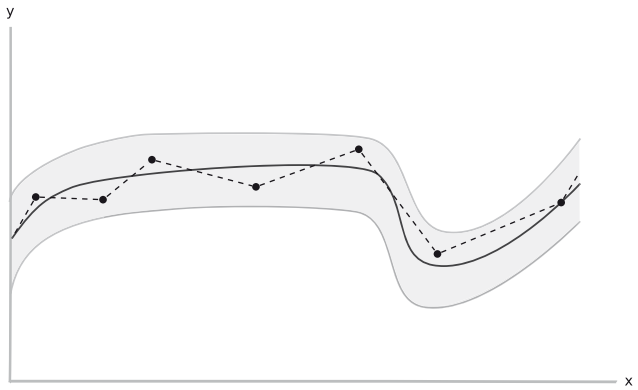

Figure 4.2: An illustration of over-fitting and generalization. The black line represents the true concept to be learned, the target function. Black dots represent the available data points. An over-fitted model derived using the points is a dashed line. Generalization will introduce an "uncertainty region", here represented by a gray shadow. Its shape resembles the shape of the target function more than the shape of the over-fitted model does.

Feature selection eliminates a lot of noise from data and is often a good remedy to over-fitting. Another way of overcoming the problem which is complementary to feature selection is the application of various generalization techniques. By generalization we mean relaxing constraints imposed by the original over-fitted model by, e.g., representing features in a less specific way. In our work (Paper I - Paper IV), we were using physicochemical descriptors of amino acids instead of letter-codes to represent protein sequences. This is a good example of model-generalization by using less specific features. In the case of rough set rule-based models (see Paper I), we create more general models by shortening rules and merging the most similar ones.

5. Methods

In this chapter I present computational methods that are used in the included papers. First, I talk about representing protein sequences in terms of their physicochemical properties. Next, I discuss rough sets and proceed to tree-based methods of classification and feature selection: random forests and our MCFS-ID method.

5.1 Representing sequences

As I mentioned in the chapter devoted to biology, it is convenient to represent protein sequence as a string of letters where each letter denotes an amino acid. Already this simple representation is sufficient to compare sequences, infer phylogenetic trees, search for specific motifs etc. [41, 56] It is, however, physicochemical properties of amino acids that determine protein folding and function. Research presented in this thesis is focused on examining these properties and attempts to understand how they change to alter protein function. Following [101], we selected seven physicochemical properties from the aaIndex [64] database of amino acid descriptors and we use these properties to represent protein sequences. The selected properties are summarized in Table 5.1.

No.	Descriptor	aaIndex code	Authors
1	Transfer free energy from octanol to water	RADA880102	[94]
2	Normalized van der Waals volume	FAUJ880103	[45]
3	Isoelectric point	ZIMJ680104	[127]
4	Polarity	GRAR740102	[51]
5	Normalized frequency of turn	CRAJ730103	[31]
6	Normalized frequency of alpha-helix	BURA740101	[24]
7	Free energy of solution in water	CHAM820102	[27]

Table 5.1: *Physicochemical properties of amino acids selected from the aaIndex database.*

Descriptors 1, 2 and 3 represent three important physicochemical properties: hydrophobicity, size and charge. Interpretation of "Polarity" is straightfor-

ward. Both "Normalized frequency of turn" and "Normalized frequency of alpha-helix" represent the propensity of an amino acid to form a particular secondary structure. "Free energy of solution in water" characterizes amino acid-solvent interactions.

The selected descriptors are biologically meaningful while at the same time being low correlated and thus preserving the ability to discern one amino acid from another. The detailed description of the selection process can be found in [102] and in Paper I.

5.2 Rough sets

The notion of a *rough set* was intorduced by Zdzisław Pawlak in 1982 [88]. The theory of rough sets aims to analyze imprecise, uncertain and often incomplete information that is expressed in terms of data acquired form observation or experiment. This goal is achieved by constructing an approximation of a set that, based on the available data, cannot be expressed in terms of the classical set theory.

Recall the drug screening example presented in the previous chapter, Table 4.1. As the rough set theory requires that all the attributes take discrete values, we *discretized* the "Size" attribute into three classes: small, medium and large. The resulting dataset is presented in Table 5.2.

Compound	Size	Polarity	Aromatic?	Charge	**Efficacy**
C1	large	medium	yes	positive	**high**
C2	large	low	yes	neutral	**low**
C3	small	low	yes	negative	**low**
C4	medium	medium	yes	neutral	**high**
C5	medium	medium	yes	neutral	**low**
C6	medium	high	yes	positive	**high**
C7	large	high	no	negative	**high**
C8	small	low	no	positive	**low**

Table 5.2: *An example of a decision table.*

Note that the first attribute, the name of the compound, is used only for the purpose of identifying instances. Since it does not come from any kind of measurement and is arbitrarily assigned to each instance, it is not a part of the information system and should not be used to classify instances.

5.2.1 Indiscernibility

The notion of *indiscernibility* lies in the very core of the rough set theory. Intuitively, two objects are *indiscernible* if it is not possible to distinguish between them on the basis of a given set of attributes. Indiscernibility is a function of these attributes; an *equivalence relation* defined on a given set of attributes [88, 67]. For each set of attributes a *binary indiscernibility relation* can be defined as a set of pairs of objects that are indiscernible given these attributes. For instance (see Table 5.2) drug candidates C4 and C6 are indiscernible on the basis of $\{size, aromatic?\}$ but once $\{charge\}$ or $\{polarity\}$ is included, they become discernible.

All the objects that are indiscernible on the basis of a given set of attributes are said to belong to the same *equivalence class*. In Figure 5.1 each square represents an equivalence class. The indiscernibility relation partitions a set of objects into a number of equivalence classes.

If $\mathbf{A} = (U, A)$ is an information system then for any $B \subseteq A$ there is an associated equivalence relation $IND_{\mathbf{A}}(B)$ [67, 85]:

$$IND_{\mathbf{A}}(B) = \{(x, x') \in U^2 \mid \forall_{a \in B}\, ax = a(x')\} \tag{5.1}$$

$IND_{\mathbf{A}}(B)$ is called the *B-indiscernibility relation*. If $(x, x') \in IND_{\mathbf{A}}(B)$ then objects x and x' are indiscernible on the basis of the attributes from B. The equivalence class of the B-indiscernibility relation for object x is denoted $[x]_B$. Coming back to our example, we can write: $IND(\{polarity, aromatic?\}) = \{\{C1, C4, C5\}, \{C2, C3\}, \{C6\}, \{C7\}, \{C8\}\}$.

Now, it is convenient to present indiscernibility relations in the form of a *discernibility matrix*. For an information system that contains n instances, a discernibility matrix is a square, symmetric $n \times n$ matrix with entries c_{ij} [67, 85]:

$$c_{ij} = \{a \in A | a(x_i) \neq a(x_j))\} \text{ for } i, j = 1, \ldots, n \tag{5.2}$$

Each entry contains the set of attributes upon values of which instances x_i and x_j differ. The full-discernibility (based on all the condition attributes) matrix for our example is presented in Table 5.2.1.

Given a classification task, it is more interesting to see which attributes discern between the decision classes. This type of information is contained in the so-called *decision-relative discernibility matrix* that can be easily derived from an ordinary discernibility matrix by considering only differences between instances with different decision classes. A decision-relative discernibility matrix for our example is presented in Table 5.2.1.

5.2.2 Rough approximation of a set

A set can be defined in a *crisp* manner when for each equivalence class all the objects that belong to that equivalence class share the same value of the

	[C1]	[C2]	[C3]	[C4]	[C5]	[C6]	[C7]	[C8]
[C1]	∅							
[C2]	p,c	∅						
[C3]	s,p,c	s,c	∅					
[C4]	s,c	s,p	s,p,c	∅				
[C5]	s,c	s,p	s,p,c	∅	∅			
[C6]	s,p	s,p,c	s,p,c	p,c	p,c	∅		
[C7]	p,a,c	p,a,c	s,p,a	s,p,a,c	s,p,a,c	s,a,c	∅	
[C8]	s,p,a	s,a,c	a,c	s,p,a,c	s,p,a,c	s,p,a	s,p,c	∅

Table 5.3: *A full discernibility matrix derived for the discussed information system (Table 5.2). Attribute names are abbreviated to one-letter codes: s – size, p – polarity, a – aromatic?, c – charge., ∅ – indiscernible.*

decision attribute. However, if there are some objects that are indiscernible on the basis of their condition attributes but that have the same value of the decision attribute, the set cannot be defined in a crisp manner. In our example (Table 5.2), objects C4 and C5 belong to the same equivalence class but they have different values of the decision attribute. The equivalence classes let us construct the *rough approximation of a set* [88, 67, 85].

A collection of all the equivalence classes where all the members share a given value of the decision attribute is called the *lower approximation* of the set. In our example the lower approximation of the "high efficacy" concept is:

$$\underline{Efficacy : high} = \{C1\} \cup \{C6\} \cup \{C7\} \tag{5.3}$$

Similarly, a collection of all the equivalence classes where at least one member has a given value of the decision attribute is called the *upper approximation* of the set. In our example the upper approximation of the "high efficacy" concept is:

$$\overline{Efficacy : high} = \{C1\} \cup \{C4, C5\} \cup \{C6\} \cup \{C7\} \tag{5.4}$$

Now we will define the above introduced concepts in a formal way. Throughout Definition 4-Definition 8 we will be considering information system $\mathbf{A} = (U, A)$ where $B \subseteq A$ and $X \subseteq U$ (see also [67, 85].

Definition 4. *The lower approximation $\underline{B}(X)$ of a set is a union of all equivalence classes which are fully included in that of X.*

$$\underline{B}(X) = \{x \mid [X]_B \subseteq X\} \tag{5.5}$$

	[C1]	[C2]	[C3]	[C4]	[C5]	[C6]	[C7]	[C8]
[C1]	∅							
[C2]	p,c	∅						
[C3]	s,p,c	∅	∅					
[C4]	∅	s,p	s,p,c	∅				
[C5]	s,c	∅	∅	∅	∅			
[C6]	∅	s,p,c	s,p,c	∅	p,c	∅		
[C7]	∅	p,a,c	s,p,a	∅	s,p,a,c	∅	∅	
[C8]	s,p,a	∅	∅	s,p,a,c	∅	s,p,a	s,p,c	∅

Table 5.4: *A decision-relative discernibility matrix derived for the discussed information system (Table 5.2). Attribute names are abbreviated to one-letter codes: s – size, p – polarity, a – aromatic?, c – charge., ∅ – indiscernible.*

Definition 5. *The upper approximation $\overline{B}(X)$ of a set is a union of all equivalence classes which have a non-empty intersection with that of X.*

$$\overline{B}(X) = \{x \mid [X]_B \cap X \neq 0\} \tag{5.6}$$

It can be shown that the lower approximation will be always fully contained within the upper approximation [88, 67, 85]. Equivalence classes that belong to the upper approximation but do not belong to the lower approximation are constituting the *boundary region*. In our example, objects C4 and C5 are in the boundary region. Such objects cannot be classified as belonging to or not belonging to X. Formally the boundary region is [88, 67, 85]:

Definition 6. *The boundary region of a set is a union of equivalence classes that belong to the upper approximation of the set but do not belong to the lower approximation:*

$$BN_B(X) = \overline{B}(X) - \underline{B}(X) \tag{5.7}$$

Finally, the *B-outside region* is constituted by all the equivalence classes where all the drugs have "low effiacy". In our case drugs {C2, C3, C8} belong to the B-outside region. Formally [67],

Definition 7. *The B-outside-region of a set is a union of all equivalence classes that do not belong to the upper approximation of the set.*

$$B - outside\ region\ of\ X = U - \overline{B}(X) \tag{5.8}$$

Having defined set approximations, we can proceed further towards defining a set in a rough manner [67]:

Definition 8. *A rough set is a tuple:* $\{\underline{B}(X), \overline{B}(X)\}$. *When the lower and the upper approximations are equal, i.e., when* $\overline{B}(X) - \underline{B}(X) = \emptyset$ *the set is crisp (or standard).*

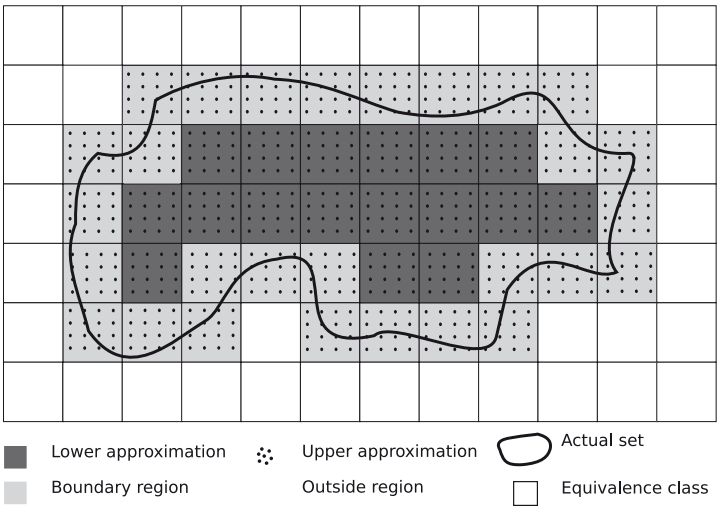

Figure 5.1: Rough approximation of a set.

Rough set can be also characterized numerically by the *accuracy of approximation* coefficient α_B.

Definition 9. *The accuracy of approximation is:*

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|} \tag{5.9}$$

where $|X|$ *denotes the cardinality of* $X \neq \emptyset$.

We should notice that $0 \leq \alpha_B(X) \leq 1$. If X is crisp with respect to B, $\alpha_B(X) = 1$ and we can say that X is precise with respect to B. Otherwise $\alpha_B(X) < 1$ and X is rough (or vague) with respect to B [67].

In the classical set theory, an element either belongs to a set or not and the corresponding membership function takes only two values: one and zero (true and false). In rough set theory the *rough membership function* quantifies the degree of relative overlap between the set X and the equivalence class $[x]$ to which element x belongs (see Figure 5.2) [67, 85].

Definition 10. *The rough membership function* $\mu_X^B : U \rightarrow [0,1]$ *is:*

$$\mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \tag{5.10}$$

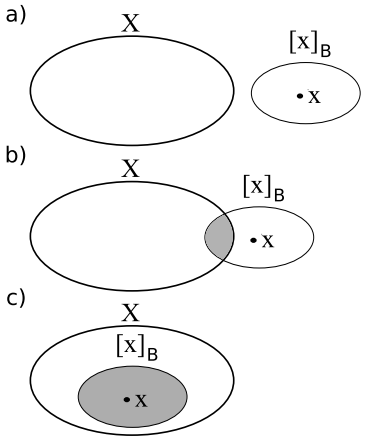

Figure 5.2: Meaning of the rough membership function. a) $\mu_X^B(x) = 0$; b) $\mu_X^B(x) < 1$; c) $\mu_X^B(x) = 1$.

The rough membership function can be used to define set approximations and the boundary region of a set [67]:

$$\underline{B}X = \{x \in U : \mu_X^B(x) = 1\} \tag{5.11}$$

$$\overline{B}X = \{x \in U : \mu_X^B(x) > 0\} \tag{5.12}$$

$$BN_B(X) = \{x \in U : 0 < \mu_X^B(x) < 1\} \tag{5.13}$$

The rough membership function can be interpreted as a frequency-based estimate of the conditional probability $Pr(x \in X | x \in [x]_B)$ that element x belongs to set X given the information provided by attributes B [89].

Rough sets are often compared to *fuzzy sets* introduced by Zadeh [125]. One of the differences between these approaches is the character of the set membership function. The classical *fuzzy membership function* does not have probabilistic interpretation and is defined in an arbitrary way. While fuzzy set theory expresses vagueness by means of the degree of set membership, rough set expresses vagueness by employing a boundary region of a set. Rough set theory clearly distinguishes two important concepts: *vagueness* and *uncertainty*. Vagueness is the property of sets can be expressed by approximations; uncertainty is the property of elements of a set and can be expressed by the rough membership function [89].

5.2.3 Reducts

A discernibility matrix can be presented in the form of a corresponding Boolean *discernibility function*. The discernibility function $f_d(B)$ computes the minimal sets of attributes from B that are required to discern a given equivalence class from all the others [67, 85, 111]. For the discussed decision-relative discernibility matrix, Table 5.2.1, the discernibility function takes the following form:

$$\begin{aligned} f_d(s,p,a,c) = {} & (p \vee c) \wedge \\ & (s \vee p \vee c) \wedge \\ & (s \vee p) \wedge (s \vee p \vee c) \wedge \\ & (s \vee c) \wedge \\ & (s \vee p \vee c) \wedge (s \vee p \vee c) \wedge (p \vee c) \wedge \\ & (p \vee a \vee c) \wedge (s \vee p \vee a) \wedge (s \vee p \vee a \vee c) \wedge \\ & (s \vee p \vee a) \wedge (s \vee p \vee a \vee c) \wedge (s \vee p \vee a) \wedge (s \vee p \vee c) \end{aligned}$$

and after removing redundant terms and simplification it becomes:

$$\begin{aligned} f_d(s,p,a,c) = {} & (p \vee c) \wedge (s \vee p \vee c) \wedge (s \vee p) \wedge \\ & (s \vee c) \wedge (p \vee a \vee c) \wedge (s \vee p \vee a) \wedge (s \vee p \vee a \vee c) \\ = {} & (p \wedge s) \vee (p \wedge c) \end{aligned}$$

Formally [67, 111],

Definition 11. *For an information system* **A** *the discernibility function is [67, 111]:*

$$f_{\mathbf{A}}(a_1^*, \ldots, a_m^*) = \forall\{\exists c_{ij}^* | 1 \leq j \leq i \leq n, c_{ij} \neq \emptyset\}, \tag{5.14}$$

where $a_1^*, \ldots, a_m^*$ *are* m *Boolean variables corresponding to attributes* $a_1, \ldots, a_m$ *and* $c_{ij}^* = \{a^* | a \in c_{ij}\}$.

A *prime implicant* of a function is an implicant that cannot be covered by a more general (i.e., more reduced, with fewer literals) implicant. In our example, each of the $(p \wedge s), (p \wedge c)$ prime implicants determines a *reduct*, i.e., the minimal set of attributes that is preserving the original indiscernibility relation based on all the attributes [67, 88, 89].

Definition 12. *The reduct is the minimal set of attributes that preserve indiscernibility relation between objects. In other words,* $P \subseteq Q$ *is a reduct of* Q *if* P *is minimal among all subsets of* Q *which yield the same classification as* Q*:* $IND_B(P) = IND_B(Q)$.

The attributes within a reduct are independent and none of them can be omitted for the description of Q.

5.2.4 Rules

Reducts are very useful when constructing a classifier. IF-THEN classification rules are constructed by reading off the values for each attribute in the reduct and associating them with one or more decision classes [67, 76, 4, 89, 88]. The IF-part of a rule is called the *antescendent* or *premise* and the THEN-part is called the *consequent*. Given the previously found $\{p,s\}$ reduct we can read the rule off the C4 drug candidate:

IF Size is medium AND Polarity is medium THEN Efficacy is high

and off the C5 drug candidate:

IF Size is medium AND Polarity is medium THEN Efficacy is low

Since the decision class is rough with respect to both "Polarity" and "Size", the THEN part of the rule derived from both the C4 and C5 is

IF Size is medium AND Polarity is medium THEN Efficacy is high OR Efficacy is low

The fraction of instances from the decision class in the THEN-part that also match the IF-part is called *coverage*. Coverage measures the generality of a rule. The fraction of instances that match the IF-part and are from the decision class of the THEN-part is called *accuracy*. Accuracy tells how specific a rule is. An ensemble of rules read off all the objects in the training set constitutes a classifier. A new instance is classified by first identifying all the rules where the IF-part matches attribute values of this instance. Once the matching rules are identified, they cast votes to the decision classes accordingly to the THEN-parts. The number of instances that match both the IF-part and the THEN-part of a rule is called *support*. Each rule casts a number of votes that is proportional to the support of the rule and to its generality (more general rules cast more votes). Finally, the decision classes that received the fraction of votes that is higher than a given threshold value (determined by ROC analysis) are considered predictions. If cost of making false negative predictions and false positive predictions is equal, usually the ROC point closest to the (0, 1) point is used [92, 122, 54, 75].

5.2.5 Rule shortening and generalization

Noise is inherent to any type of experimental data. Using these data for training may result in very specific rules that do not describe true dependencies in the universe [76, 4]. In such case, rules are often long and have low support [76]. One solution to this problem is rule filtering, e.g., discarding all the rules that have support below certain threshold. Another way to alleviate this problem is rule shortening. We use rule shortening and generalization method proposed by M akosa [76]. Short rules are obtained by removing some descriptors from the IF-part of a rule. The choice of the right subset of descriptors that will

constitute the IF-part of the shortened rule is the crucial step in the shortening process. The goal is to obtain a rule which is better at classifying instances from the part of the universe not included in the training data. Shortening a rule may result in the drop of its accuracy and it is up to the operator to set the maximal acceptable value α of the drop. Let us consider a possible scenario of rule shortening and generalization. Here, instead of descriptive labels such as "low", "medium", "high", value intervals are considered. The following rules:

IF $f_1 = (-\infty, 25.0]$ *AND* $f_2 = (0.45, 2.0]$ *AND* $f_3 = (0.2, 0.7]$ *THEN* d_1
IF $f_1 = (1.5, 22.0]$ *AND* $f_2 = (2.0, 5.0]$ *AND* $f_3 = (0.2, 0.7]$ *THEN* d_1
IF $f_1 = (-\infty, 25.0]$ *AND* $f_2 = (2.0, 5.0]$ *AND* $f_3 = (0.2, 0.7]$ *THEN* d_1

First step is rule shortening which involves removing these descriptors that do not result in the drop of accuracy larger than the preset value α. Let us assume that in the case of all the above rules, the removal of the f_3 descriptor transpired to have very little influence on accuracy. The shortened rules will be:

IF $f_1 = (-\infty, 25.0]$ *AND* $f_2 = (0.45, 2.0]$ *THEN* d_1
IF $f_1 = (1.5, 22.0]$ *AND* $f_2 = (2.0, 5.0]$ *THEN* d_1
IF $f_1 = (-\infty, 25.0]$ *AND* $f_2 = (2.0, 5.0]$ *THEN* d_1

Now, the feature values (here intervals) can be merged if this do not cause the drop of rule accuracy exceeding α:

IF $f_1 = (-\infty, 25.0]$ *AND* $f_2 = (0.45, 5.0]$ *THEN* d_1

The resulting ensemble of general rules (in our example just one rule) is expected to be better in classifying instances not included in the training set and, thus, minimizes the risk of over-fitting.

5.3 Tree-based methods

In the following sections I discuss a family of methods that are based on or make use of decision-trees. Among others, the family encompasses: random forests, Monte Carlo feature selection and Monte Carlo feature selection and interdependency discovery. Decision trees that can be defined in terms of graph theory:

Definition 13. *A graph G consists of a finite set V of vertices an a set E of two-subsets of V, members of which are called edges. It is denoted as*

$$G = (V, E) \tag{5.15}$$

where V is the vertex set and E is the edge set [16].

The usual way to visualize a graph is to draw a dot for each vertex and to join two of these dots with a line if the corresponding vertices constitute an edge [34, 16].

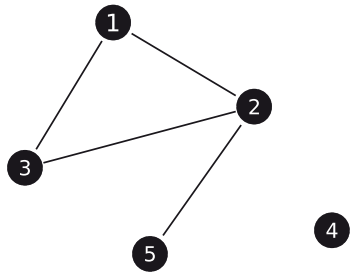

Figure 5.3: Visualization of a graph $G = \{V,E\}$, where $V = \{1,\ldots,5\}$, $E = \{\{1,2\},\{1,3\},\{2,3\},\{2,5\}\}$.

An edge is a *directed edge* if it is a one-way route between the nodes it connects, which is graphically represented as an arrow. A graph is *directed* when its edges are directed. A graph is *acyclic* if for every vertex there exist no series of edges connecting the vertex with itself. Finally, a graph is *connected* if for every pair of vertices there exist a series of edges that connects the members of that pair [16]. Graphical interpretation of the introduced terms is provided in Figure 5.4.

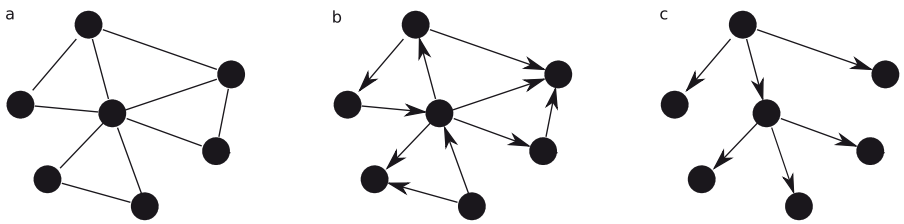

Figure 5.4: Various graph types: an undirected cyclic graph (a); a directed cyclic graph (b) and a directed acyclic graph (c).

A *tree* is defined as an acyclic and connected graph. It can be either directed or undirected. A directed tree is presented in Figure 5.5. The top-level vertex is called the *root*. The down-level vertices are called *children*. In machine learning, vertices are also called *nodes*. Every node in a tree, except the *leaf-nodes*, has *children*, i.e., lower-level nodes that are connected to this *parent node*. A tree where every node has at most two children is called a *binary tree* [16].

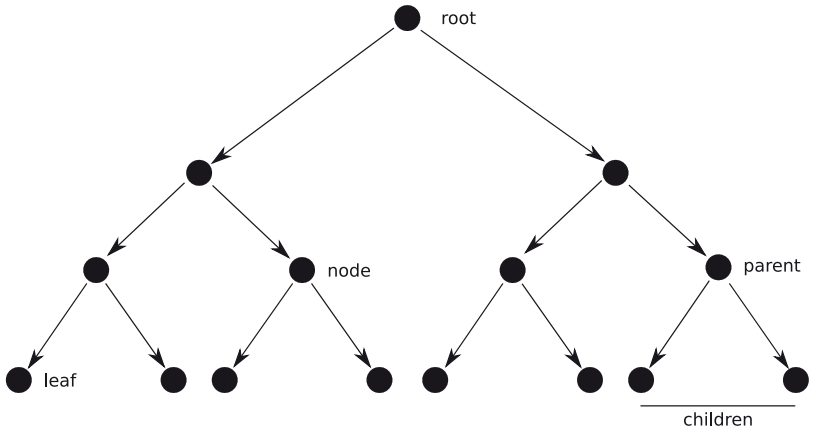

Figure 5.5: An example of a directed binary tree.

5.3.1 Decision trees

A *decision tree* (or a *classification tree*) defined as a directed, acyclic and connected graph is a mathematical object frequently used to represent a classification process [122]. First, classification trees were introduced in social sciences, in the early 1960s and by the end of the 1970s they had become widely used in the domain of machine learning [99, 70]. In machine learning, a specific terminology is used to describe classification trees. Vertices are called *nodes* and series of edges are called *paths*. The top-level vertex is called *the root* it is connected to its *child* nodes that in turn have their own children and so forth. The vertices that do not have any child-nodes are called the *leaves*. The terminology is explained in Figure 5.5.

Decision trees are a natural way of representing a classification process [70]. Nodes of a decision tree involve testing a particular attribute and leaf-nodes provide the classification that applies to all the instances that reached that particular leaf-node. Typically, the test at a node compares an attribute value with a constant, but sometimes two attributes are compared or a more complex function of one or more attributes is used to direct an instance to the next node. Leaf nodes give a classification, a set of classifications or a probability distribution over all possible classifications that apply to all instances that reach the leaf. Initially, an instance to be classified is placed at the root of a decision tree. In the root, the first test is performed and the instance is directed to an appropriate child-node where yet another test is performed. The instance is routed further down the tree reaching subsequent test-nodes to finally reach a leaf where it is classified [122, 70].

The process of constructing a decision tree can be expressed in a recursive way. Let us consider construction of a binary decision tree. At the beginning, all training instances are placed at the root node. First, an attribute that will

be used for testing at the root node has to be selected. This splits the set of training instances into two subsets corresponding to the possible outcomes of the test. This process can be repeated recursively for each branch using only the instances that reached that branch. Once all instances that reached a branch belong to the same class, the development of that part of the tree is terminated. Let us consider the following data set:

Size	Charge	Polarity	Is aromatic?	**Is metabolized?**
small	neutral	high	FALSE	**non-metabolized**
large	neutral	high	TRUE	**non-metabolized**
small	positive	high	FALSE	**non-metabolized**
small	positive	high	TRUE	**non-metabolized**
large	negative	low	TRUE	**non-metabolized**
medium	negative	low	TRUE	**metabolized**
small	neutral	low	TRUE	**metabolized**
medium	neutral	high	TRUE	**metabolized**
medium	positive	high	FALSE	**metabolized**
medium	positive	low	FALSE	**metabolized**
small	negative	low	FALSE	**metabolized**
large	neutral	high	FALSE	**metabolized**
large	neutral	low	FALSE	**metabolized**
large	negative	low	FALSE	**metabolized**

Table 5.5: *Simple training set.*

When constructing a decision tree, one would like to minimize the number of nodes and maximize classification accuracy at the same time. A model should be at the same time simple and useful for classification. Let us consider a top-down approach to the construction of a decision tree. First, we have to select the best attribute to split. But what does "the best" mean? We will use the *information gain ratio* measure. Initially, there are 14 instances in our data set (Table 5.5): 9 labelled as "metabolized" and 5 labelled as "non-metabolized". Before making any splits, we can compute the *information value* for the entire dataset:

$$\text{info}([\,9,\,5\,]) = -\frac{9}{14}\cdot\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\cdot\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits} \qquad (5.16)$$

Now, let us consider the "size" attribute. One can split the data set on "size" as illustrated in Figure 5.6:

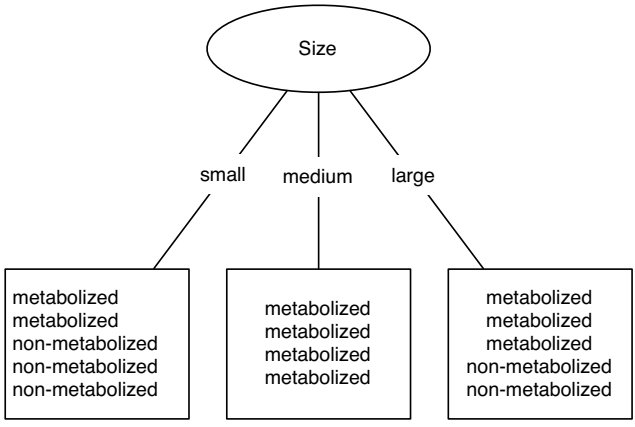

Figure 5.6: An example split. Here on "Size".

Such a split results in somehow more ordered data – we have one pure "metabolized" class. The number of "metabolized" and "non-metabolized" classes at the leaf nodes are: [2, 3], [4, 0] and [3, 2] respectively. The information values at these nodes are:

$$\text{info}([2,\ 3]) = -\frac{2}{5}\cdot\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\cdot\log_2\left(\frac{3}{5}\right) = 0.971 \text{ bits} \tag{5.17}$$

$$\text{info}([4,\ 0]) = -\frac{4}{4}\cdot\log_2\left(\frac{4}{4}\right) - \frac{0}{4}\cdot\log_2\left(\frac{0}{4}\right) = 0 \text{ bits} \tag{5.18}$$

$$\text{info}([3,\ 2]) = -\frac{3}{5}\cdot\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\cdot\log_2\left(\frac{2}{5}\right) = 0.971 \text{ bits} \tag{5.19}$$

An average information value of these can be easily calculated by taking into account the number of instances that go down each branch, in our case: 5 instances down the"small" branch, 4 instances down the "medium" branch and 5 down the "large" branch. Now we can compute:

$$\text{info}([2,\ 3],\ [4,0],\ [3,2]) = \frac{5}{14}\cdot 0.971 + \frac{4}{14}\cdot 0 + \frac{5}{14}\cdot 0.971 = 0.693 \text{ bits} \tag{5.20}$$

This average represents the amount of information that is expected to be necessary to specify the class of a new instance, given the tree structure. Now we shall recall the (Equation 5.16) where the amount of information before performing any splits was 0.940 bits. Finally we are ready to calculate how much we gained by splitting the data on "size":

$$\begin{aligned} \text{gain}(\text{'size'}) &= \text{info}([9,\ 5]) - \text{info}([3,\ 2],\ [4,\ 0],\ [2,\ 3]) \\ &= 0.940 - 0.693 = 0.247 \text{ bits} \end{aligned} \tag{5.21}$$

However, information gain is not the perfect measure to decide on the best split. Why? It will favor the attributes that have many possible values. In the extreme case, if each instance has a "unique id", this attribute will be used to construct the best and the only split. To correct for this, we need to compute the *split information value* of the attribute. We do it by ignoring the decision classes and taking into account only the fact that splitting on the attribute "size" will result in 3 branches, containing 5, 4 and 5 instances respectively:

$$\text{split info}(\text{`size`}) = \text{info}([5,\ 4,\ 5]) = \text{info}([5,\ 9]) + \frac{9}{14} \cdot \text{info}([4,\ 5]) = 1.577 \text{ bits} \tag{5.22}$$

Now we are ready to compute the final measure, the *information gain ratio*:

$$\text{gain ratio}(\text{`size`}) = \frac{\text{gain}(\text{`size`})}{\text{split info}(\text{`size`})} = \frac{0.247}{1.577} = 0.157 \text{ bits} \tag{5.23}$$

An attribute with the highest gain ratio will be used to construct the first node. By applying this procedure recursively, one will end up with the complete decision tree which, for this particular classification problem, is presented in Figure 5.7.

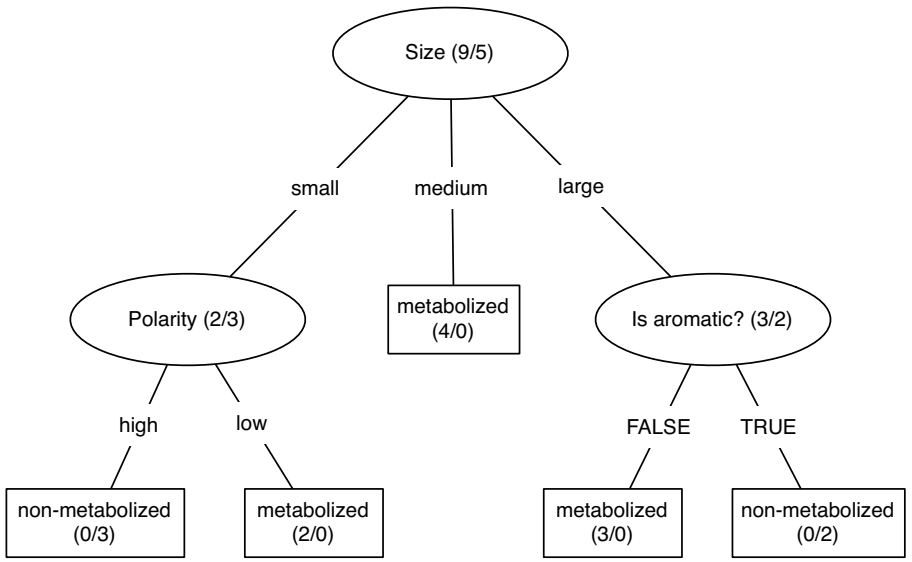

Figure 5.7: An example of a decision tree. A decision tree for the data from Table 5.5 has been constructed with the J48 algorithm as implemented in WEKA suite [122]. The actual number of objects belonging to a given class that reached the given node or leaf is given in parentheses: (metabolized/non-metabolized).

5.3.2 Combining classifiers - bagging

Unfortunately, decision tree-based classifiers are quite sensitive to slight changes in the training data and, when limited training data is available, their predictions can be unstable [122]. *Bagging* (bootstrap aggregating) is an algorithm that relies on using random subsets of the original training data to construct a large number of (often not very accurate) classifiers that all together will yield accurate and stable predictions. In bagging, each random subset is prepared by randomly drawing (with replacement) N objects from the original training set that also contains N objects [122, 70]. We can consider a classification problem with two possible decisions and assume that we have a number, N_c, of independent classifiers and that the probability of assigning the wrong class-label to an instance is the same for each classifier and equals 0.4. Thus, each classifier gives right answers with the probability of 0.6. Naturally, among all the decisions taken by the N_c classifiers, we expect $0.6 \cdot N_c$ to be correct ($variance = 0.4 \cdot 0.6 \cdot N_c$ and $standard\ deviation = 0.4899\sqrt{N_c}$). It follows that the probability that the majority of the classifiers gives correct answer increases with the number of the classifiers. Therefore the decision returned by the majority of a sufficiently large number of medium-quality, independent classifiers can be accurate. This observation led to the introduction of bagging. In 1996 Leo Breiman applied bagging to create an ensemble of tree-based classifiers where a random selection (without replacement) of instances from the training set was made to grow each tree an the final decision was a result of voting [19, 70].

5.3.3 Random forests

Interest in bagging and other methods of combining multiple classifiers into one led Breiman to the invention of Random Forest in 2001. The name *random forest* pertains also to a general concept denoting the whole family of classifiers that combine decision trees. A random forest is an ensemble of classification trees where each tree is constructed using a random vector that governs the growth of the tree. Let k decision trees constitute a random forest. For each tree a random vector Θ_k is generated that is independent of the past random vectors $\Theta_1, \ldots, \Theta_{k-1}$ but has the same distribution. Now, a tree is grown using the training set and Θ_k resulting in a classifier $h(\mathbf{x}, \Theta_k)$ where $\mathbf{x}$ is an input vector.

Definition 14. *A random forest is a classifier consisting of a collection of tree-structured classifiers* $\{h(\mathbf{x}, \Theta_k), k = 1, \ldots\}$ *where the* $\{\Theta_k\}$ *are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input* $\mathbf{x}$ *[19].*

For instance, in bagging the random vector Θ is generated as the number of units in N boxes resulting from N darts thrown at random at the boxes, where N is the number of examples in the training set [19] .

In Breiman's approach to random forest construction, yet another source of randomness is used: in each tree of the forest a randomly selected subset of original attributes is used at each node [19, 28]:

1. From the original training set containing N objects and M attributes draw randomly with replacement $n = N$ observation vectors.
2. For each node of the tree select m out of the original M attributes ($m << M$) and determine the best split at the node using only the selected attributes. As a rule of thumb, Breiman suggests to set $m = \sqrt{M}$.
3. Each tree is grown to the largest extent and not pruned.

This algorithm is used to construct a number, τ, of decision trees that constitute a random forest. Every new instance with unknown decision class is classified by all the τ trees and each tree casts a vote to one of the decision classes. The class that received the majority of votes is the final decision for the instance.

I have already mentioned (page 39) that in bootstrap approximately $\frac{1}{3}$ of the instances from the original training set are not present in a given bootstrap sample. In random forests-related terminology the instances not used for tree construction are called *out-of-bag* instances. The out-of-bag instances are used to assess the performance of the random forest classifier, but they can be also used to build a ranking of feature importance. Suppose that there are M features and each time a new tree is constructed, values of the m-th feature are permuted in all the out-of-bag instances and the instances are run down the tree. The classification that is assigned to each $\mathbf{x}_n$ that is out-of-bag is saved. This is repeated for $m = 1, 2, \ldots, M$. At the end the number of out-of-bag class-votes for $\mathbf{x}_n$ with the m-th feature reshuffled is compared with the true class label of $\mathbf{x}_n$. This gives a relative importance of the m-th feature [19, 70].

Random forests are considered to be one of the best off-the-shelf classifiers currently available and many authors recommend them as the first choice classifiers when working on a new data set. However, when the training data contains a lot of noise, random forests are prone to overfitting [107]. Also, when the training data contains a lot of irrelevant features and correlated features, random forests do not perform well [50]. These problems can be alleviated by the application of feature selection prior to constructing the classifier. While the built-in feature ranking procedure is a good way to get an idea of the importance of features, it can fail to produce reliable rankings in some cases. As it relies on reshuffling values of a single attribute at a time, it may give false results when two or more attributes co-operate in determining the outcome [40]. Also, the more values an featuree has, the less noise is introduced by simple reshuffling. Similarly, an uneven distribution of feature-values can lead to erroneous assessment of the importance of that feature. While these

problems are not pronounced in the case of many data sets, experimental data coming from biological studies often contains noise, and it is not uncommon that the outcome is determined by an interplay of a number of features that, when considered separately, do not appear to be relevant.

5.3.4 Monte Carlo feature selection and interdependency discovery

Also our current implementation of Monte Carlo feature selection (MCFS) method uses ensembles of decision tree-based classifiers. Let us assume that the original data set contains N instances and M features; each object is annotated with a decision class label. MCFS procedure involves the following steps (see also Figure 5.8):

1. Set the parameter $m << M$. Select s subsets, each consisting of m different features. The m features are selected in a random fashion without replacement.
2. For each subset s, construct t trees, each trained using a different, randomly selected subset of N objects. Each time the original set of N objects is randomly split into a training set $(0.66 \cdot N)$ and a test set $(0.33 \cdot N)$ preserving the original decision-class proportions.
3. For each tree constructed using a given training set, use the corresponding test set to evaluate the tree.

Eventually $s \cdot t$ trees are constructed and evaluated. Provided sufficiently large s and t, each feature has the chance of appearing in many different subsets s of features and the randomness that is inherent to the natural variability of data is properly accounted for.

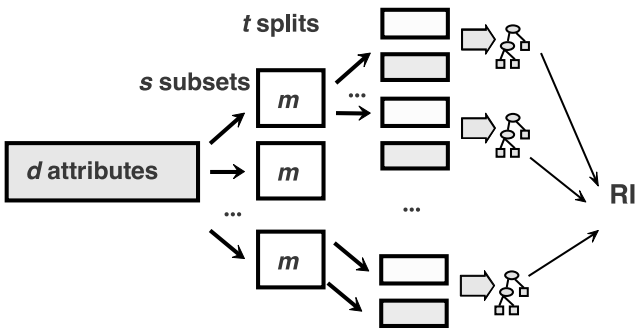

Figure 5.8: Schematic representation of the main steps of the MCFS procedure. After [40].

The simplest approach to determining the importance of a feature would be to examine how many times a split was made on the feature in all the trees. The most important features are expected to be used more frequently than the

less important ones. However, such a measure does not take into account the information gain achieved at the split, the number of instances in the split node and the accuracy of the entire tree. In the MCFS method, the final importance measure is therefore weighted. First, let us define *weighted accuracy*.

Definition 15. *Let n_{ij} denote the number of instances from class i classified as belonging to class j. For a classification problem with c classes weighted accuracy is the mean of c true positive rates:*

$$wAcc = \frac{1}{c}\sum_{i=1}^{c}\frac{n_{ii}}{n_{i1}+n_{i2}+\ldots+n_{ic}} \tag{5.24}$$

Please note that if a split is made on a particular feature g_k, the more informative the feature is, the higher *wAcc* of the whole tree, information gain and the number of instances at the node are. Information gain can be measured by gain ratio as discussed in the "Decision trees" section (page 56). Finally, the *relative importance of a feature* is defined as:

Definition 16. *Let there be $s \cdot t$ trees, τ denotes the τ-th tree in a sequence, n_{g_k} is the number of nodes in the τ-th tree where the split is made on feature g_k, IG stands for any measure of information gain and n_{inst} stands for the number of instances. The number of instances in the root is denoted $n_{inst}(\tau)$. Both u and v are parameters to be set by the experimenter. The relative importance of the g_k feature is defined as:*

$$RI_{g_k} = \sum_{\tau=1}^{st}(wAcc)^u \sum_{n_{g_k}(\tau)} IG(n_{g_k}(\tau))\left(\frac{n_{inst}(n_{g_k}(\tau))}{n_{inst}(\tau)}\right)^v \tag{5.25}$$

The two parameters u and v allow for penalizing the trees with low values of *wAcc* and the nodes with a small fraction of instances respectively.

Now, for fixed m, t, u and v the MCFS procedure is run iteratively, with the increasing number of subsets s, e.g., $s = s_1, s_1 + 10, s_1 + 20, \ldots$ and for each run a ranking of feature importance is recorded. The procedure is terminated when the two subsequent rankings of features are sufficiently similar. To measure the similarity, the distance between two rankings is defined (for details see Paper II). Once the final ranking of features is available, an additional permutation test is made (a number of MCFS rankings produced using permuted decision attribute) and the cut-off value is determined using Student's t-test. The features that in the final ranking scored below the determined cut-off are considered unimportant. Once the list of significant features is obtained, another question arises: How do the features co-operate in determining the outcome?

In order to be able to address this question we proposed (Paper II) an extension to the MCFS procedure. We call our approach MCFS-ID. The interdependency discovery (ID) part of the MCFS-ID procedure allows for pairwise

analysis of interacting features. In each MCFS-constructed tree a node represents a feature on which a split was made. Clearly, the distance between any two nodes in a tree averaged over all the $s \cdot t$ trees reflects the degree of interdependency between these nodes. The distance between two nodes is simply defined as the number of edges in the path connecting the nodes. Now an interdependency measure can be introduced.

Definition 17. *Let ξ_τ denote a path in the τ-th tree, dist denote distance and the remaining symbols are the same as used in Definition 16. The dependency between two features g_i and g_j is:*

$$Dep(g_i, g_j) = \sum_{\tau=1}^{st} \sum_{\xi_\tau} \sum_{n_{g_i}(\xi_\tau), n_{g_j}(\xi_\tau)} \frac{1}{dist\left(n_{g_i}(\xi_\tau), n_{g_j}(\xi_\tau)\right)} \tag{5.26}$$

$Dep(g_i, g_j)$ calculated on the basis of thousands of trees provides stable and reliable information about the strength of interdependency between the two features. Figure 5.9 shows an example of an inderdependency graph generated by our method.

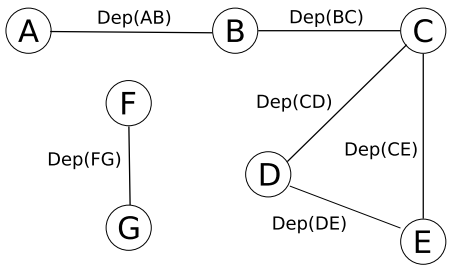

Figure 5.9: An example of an interdependency graph. The interdependency relation is not transitive: the fact that A and B as well as B and C are pairwise interdependent does not imply that A and C are pairwise interdependent.

In an interdependency graph a pair of interdependent features is represented by a pair of nodes connected by an edge. We should note that the interdependency relation is not transitive, i.e., the fact that A and B as well as B and C are interdependent does not imply that A and C are also interdependent. In order to get a deeper insight into the character of an interdependency, further analysis is required. In the case of HIV-1 resistance to drugs we mapped the interacting features onto the 3D structure of the RT (Paper III). The mapping let us interpret the observed interdependencies in terms of molecular interactions leading to resistance. Yet another approach to the interpretation of interdependency graphs is to build a rule-based model using interdependent features and mapping the obtained rules onto the corresponding graph. We took such an approach in Paper IV where we analyze interdependency networks that characterize post-translational modification sites.

While relying on the construction of a large number of decision trees, each trained on a randomly chosen subset of the original data, the MCFS-ID method is substantially different from Breiman's random forests. In MCFS-ID, a random subset of features is selected before constructing the tree. In random forests this selection is made at each node of the tree in order to maximize the degree of independence between the single decision trees. This, in turn, results in the minimization of classification variance. In contrast to RF, MCFS-ID aims at selecting significant features rather than at classifying instances and hence the different approach to attribute selection. In MCFS-ID the final importance of an attribute is measured by taking into account its importances in all the $s \cdot t$ trees.

These two methods differ also in their approach to feature selection. In random forests, feature selection involves introducing noise to the values of one feature at a time and measuring its effect on classification. As discussed in the subsection devoted to random forests (page 5.3.3), this may cause problems when two or more features are strongly correlated, when the distribution of feature values is uneven or when there are many insignificant features in the training data. The MCFS-ID method uses different measure of feature importance. Instead of introducing random noise to one feature at a time and measuring its importance in terms of the change of the classification quality, MCFS-ID uses an importance measure based on the importance of a feature in all the $s \cdot t$ trees. Thanks to this approach MCFS-ID can asses the importance of even highly correlated features.

6. Results and Discussion

In this section a brief summary and discussion of results presented in the included papers is provided.

6.1 Paper I – rough set-based model of HIV-1 resistome

High mutability of the viral enzyme reverse transcriptase results in the rapid emergence of HIV mutants. The vast majority of the acquired mutations has a negative effect on viral replication rate and it is the wild-type strain that dominates in the viral population. However, once an anti-viral treatment is applied, the mutations that give resistance to the applied drug are favored and drug-resistant strains quickly dominate in the population. The drug-resistant viruses are responsible for the majority of treatment failures, efficiently undermining the efforts to cure AIDS. Reverse transcriptase, along with another viral enzyme, called HIV protease, is the main target for anti-viral therapies. A number of mutations leading to drug resistance have been observed in RT but the sequence-resistance relationship remains only partially understood.

In Paper I we present a rough-set based approach to modeling HIV-1 resistance to reverse transcriptase inhibitors. We analyze a number of RT sequences coming from over 15 years of HIV proteome research. Each such sequence is annotated with a drug-resistance level relative to the wild-type strain. Using Monte Carlo feature selection followed by the application of rough set theory we constructed rule-based models of HIV resistance to eight reverse transcriptase inhibitors. The models consider mutation-induced changes in the physicochemical properties of the enzyme and relate these changes to drug-resistance levels. Thanks to the application of the MCFS method, only the properties that significantly contribute to the emergence of resistance are considered by the models.

The obtained results show that drug-resistance is determined in a more complex way than previously assumed. We confirmed the importance of several known drug-resistance mutations, revised the view on the importance of some other mutations and – more importantly – identified a number of previously overlooked mutations that are potentially relevant. By mapping the newly discovered mutations on the 3D structure of the enzyme, we were able to propose possible molecular mechanisms of mutation-induced drug-resistance. We have also validated our findings in literature studies. The obtained mod-

els are general and can be applied to predict resistance of previously unseen cases.

6.2 Paper II – towards understanding feature interdependencies

Rapid development in the life sciences makes it necessary to go beyond traditional machine learning techniques. A life scientist is often interested in knowing which features contribute to classifying observations, how significant these features are and what are the interdependencies between them. Even the most accurate classifier will not increase our knowledge if it lacks an interpretational layer. In Paper I we used the Monte Carlo approach to feature selection [40]. The application of MCFS resulted in the discovery of several mutations that lead to drug resistance. Many of the discovered mutations have been reported in the literature. We also found some mutations that possibly contribute to resistance but have been overlooked. Using the knowledge of significant mutations, we constructed a number of accurate and easy-to-interpret rule-based classifiers. However, drug resistance is a complex problem and it is rather an interplay of several mutations than a single mutation that leads to drug resistance. Therefore the remaining issue was to determine how the MCFS-selected mutations interdepend on each other when creating resistance.

To this end, we significantly extended the MCFS method so that it does not only detect and rank significant features, but also finds significant interdependencies between them. First, we introduced a method for finding a cut-off between significant and non-significant features. Having an MCFS ranking of feature importances, a number of randomization tests was performed to assess the probability of a particular feature receiving its rank by chance. Once the cut-off was determined, we proceeded to finding significant interdependencies between the selected features. The reliability of the approach rests on the multiple construction of tree classifiers. Each classifier was trained on a randomly chosen subset of the original data using only a subset of features. For each pair of features, we recorded the distance between them in the tree, their importance for classification and the accuracy of the tree. These values were used to compute the strength of interdependency between the features. By averaging interdependency strengths over all the trees, we obtained the final measure of interdependency between the features.

We illustrated the MCFS-ID method on a task of modeling HIV-1 resistance to an anti-viral drug, Didanosine.

The approach proposed in Paper II makes analysis of feature interactions independent of the choice of classification method. The MCFS-ID method gives the researcher more flexibility and the ability to get insight into the actual process of classification.

6.3 Paper III – molecular interaction networks underlying HIV-1 resistance

In Paper III we applied MCFS-ID in order to understand interdependences between physicochemical features at the sites forming the HIV-1 RT resistome. We examined resistance to six drugs used in anti-HIV therapy: four nucleoside RT inhibitors, one nucleotide RT inhibitor and one non-nucleoside RT inhibitor. For each drug we obtained a network of 20% of the most significant interdependent physicochemical features of mutating amino acids. The obtained results show the complexity of the HIV-1 resistome. Particularly resistance to abacavir and resistance to nevirapine are described by complex networks. To validate our findings, we mapped the selected significant sites onto the 3D structure of reverse transcriptase. All interdependent pairs are located around the active site of the enzyme. We showed that the members of an interacting pair are often located either on the opposite sides of the active site or within the same alpha-helix. We also found pairs that describe fingers-palm interactions.

While some of the observed interdependences have been described in the literature, some novel, previously neglected or overlooked interdependent sites were observed. We hypothesize (see also Paper I) that the novel interdependencies may play compensatory role and increase viral fitness in the presence of drug pressure. Some of the sites might have emerged as a result of previous therapies. To our best knowledge it is the first study that attempts at explaining HIV resistome in terms of interdependencies between the physicochemical properties of mutating amino acids.

6.4 Paper IV – towards predicting post-translational modifications

In Paper IV we applied an MCFS-ID approach to post-translational modifications. We extracted a number of short (9 aa-long) protein sequence fragments labelled with their modification status from the UniProtKB/SwissProt [10] database. For each of the 76 examined types of modifications we created a separate training set: an ensemble of modified and non-modified fragments. Since recognition of a modification site by an enzyme is a molecular process dependent on physicochemical properties of the sequence and the enzyme, we represented each sequence fragment in terms of its physicochemical properties. The next step was the application of the MCFS-ID method to find the physicochemical properties that determine modification sites.

For each type of modification, we constructed two random forest [19] classifiers: one using all 63 features, the other one using only the significant features. For 59 types of modifications, MCFS-ID selected at least one significant feature and for the remaining 17 modifications no significant feature was

found within 9aa sequence fragment. Curiously enough, in many cases classifiers using all the features performed well, even if no significant feature was found. This interesting finding shows the need for the application of feature selection prior to model construction. Additional investigation is required to determine what the influence of non-significant features is on the quality of classification.

High-quality random forest-based classifiers that take into account only the significant features can be used to predict modification sites in previously unseen protein sequences. We are planning to release our models as a publicly available web-based server.

We also applied clustering to group types of modifications where the modification site is determined by similar patterns of physicochemical properties. This led to several groups of modifications. Interestingly, while some of these groups are homogenous, for example, modifications catalyzed by the same enzyme, the other groups consist of apparently distantly-related types of PTMs. These groups deserve further attention.

Finally, we selected the modification of lysine to allysine catalyzed by lysyl oxidase (LOX) as an example how MCFS-ID results can be complemented by an ensemble of decision rules. Since abnormal LOX activity plays crucial role in lathyrism and in cancer metastasis, understanding LOX substrate specificity is an interesting biomedical task. Using rough sets theory we inferred 38 general rules describing recognition site. The obtained ensemble of rules led to a high-quality (AUC = 0.93) classifier for predicting modification to allysine.

6.5 Summary

A large number of important problems in molecular biology can be thought of as a problem of finding a sequence-function relationship. More precisely, it is often interesting to find a relationship between the change in protein sequence and the change in protein function. In the papers included in this thesis, we focused on modeling sequence-function relationship in proteins. The problem is illustrated in Figure 6.1.

Figure 6.1: An illustration of a sequence change-induced shift in protein function. Δseq denotes a change in protein sequence, Δfun denotes the corresponding change in protein function. Often, but not always, Δseq is synonymous to a mutation.

In this thesis we developed a universal approach to modeling the Δseq → Δfun relationship. We model change in function using significant physicochemical

features and their values. To construct such models, we significantly extended the Monte Carlo feature selection method [40] so that it can be used to analyze interdependencies between features. Thanks to using a large number of classifiers, each constructed on randomly selected subset of the original data, the MCFS-ID method produces statistically sound results. The application of our method to a decision system results in a ranking of significant features ordered with respect to their relative importance and in a graph showing interdependencies between the features. The selected set of significant features can be used to build predictive models using a machine learning method of choice. In our research, we used rough set-based and random forest-based approaches to modeling. Rough set models are based on a number of easy-to-understand rules which are of particular advantage when a deeper insight into the modeled problem is desirable. Random forests are considered to be one of the "best off-the-shelf" classifiers and can easily deal with large data sets.

The length of a protein sequence varies from 40 to several thousand residues with an average of about 300 amino acid residues [23]. Three hundred amino acids, each described by just three basic physicochemical properties (hydrophobicity, polarity and size) give 900 features. At the same time there is only a limited, often rather small, number of sequence-variants available for investigating the sequence-function relationship. This results in the so-called *ill-defined* problems where the number of features exceeds the number of instances. Many machine learning techniques do not perform well on such data sets. The MCFS method can be used to reduce the dimensionality of such problem by removing non-significant features.

To our knowledge, the MCFS-ID method presents a completely novel approach to feature-interdependency analysis. In Paper II we show on the problem of modeling HIV-1 resistome that networks of interdependent features often reflect molecular interactions underlying the modeled phenomenon.

Using our approach, we modeled the sequence-function relationship in two biological problems: 1) HIV resistance to drugs and 2) determination of post-translational modification sites in proteins. In the case of HIV we were not only able to re-discover several mutation sites previously associated with resistance, but we also found some new (sometimes previously neglected) sites that possibly contribute to drug resistance. By analyzing interdependency networks mapped onto the 3D structure of HIV-1 RT, we were able to propose possible molecular mechanisms of the discovered mutations. We validated our findings by literature studies and by careful analysis of the available 3D structures of the enzyme.

In the case of post-translational modifications we built a number of models that can be used to predict several modification types. We showed which types of modifications cannot be predicted using short sequence fragments. Finally, we provided an example of combining interdependency graphs and rough set-derived rules to get insight into the actual classification process.

6.6 Future research

Our future research will be focused on developing the MCFS-ID method. We would like to increase interpretability of interdependency graphs by, for instance, mapping classification rules onto them. We would also like to examine the possibility of using our approach in protein engineering, for example, to increase thermostability of enzymes.

We would like to build more complete models of the HIV resistome by including information on treatment history, phylogeny and on viral fitness.

Results obtained in Paper IV showed the need of considering longer sequence fragments when modeling certain types of modifications. We would also like to examine 3D structures of the modification sites and include this information in our models.

7. Sammanfattning på Svenska

Ett stort antal viktiga problem inom molekylärbiologin kan betraktas som ett problem att finna en relation mellan sekvens och funktion. Mer exakt, det är ofta intressant att hitta ett samband mellan förändringen av proteinsekvens (t.ex. en mutation) och förändringen av proteiners funktion. I artiklarna som ingår i denna avhandling har vi fokuserat på modellering av relation mellan sekvens och funktion på protein-nivå. Problemet illustreras i Figur 7.1.

Figure 7.1: En illustration av en sekvensförändring-inducerad shift av proteins funktion. Δseq betecknar en förändring av proteinens sekvens, Δfun betecknar motsvarande förändring av proteinens funktion. Ofta, men inte alltid är Δseq synonymt med en mutation.

I denna avhandling har vi utvecklat en universell modell för modellering av Δseq → Δfun relation. För att uppnå detta har vi gjort omfattande tillägg till Monte Carlo metoden för egenskap urval [40] så att den kan användas för att analysera samband mellan olika egenskaper. Metoden används av ett stort antal klassificerare. Varje klassificerare är uppbyggd av en slumpvis utvald delmängd av de ursprungliga uppgifterna. Tack vare detta ger MCFS-ID metoden statistiskt korrekta resultat.

Tillämpningen av vår metod på ett beslutssystem ger en rangordning av viktiga egenskaper och ett diagram som visar sambanden mellan olika egenskap. De utvalda viktiga egenskapet kan användas för att bygga prediktiva modeller med hjälp av en valfri "maskinlärande" metod.

Vi använde "rough set"-baserade och "random skogs"-baserade metoder för modelleringen. "Rough set"-modeller bygger på ett antal regler som är lätta att förstå och av särskild fördel när en djupare insikt i modellens problem är önskvärt. Random-skogar anses vara en av de bästa “off-the-shelf” klassificerare och kan enkelt hantera stora datamängder.

Längden på en proteinsekvens varierar från fyrtio till flera tusen restprodukter med ett genomsnitt på cirka 300 aminosyror [23]. Tre hundra aminosyror, vilka beskrivs av blott tre grundläggande fysikalisk-kemiska egenskaper: hydrofobicitet, polaritet och storlek ger 900 funktioner. Samtidigt finns det bara

ett begränsad antal, ofta ganska små, sekvens-varianter som man kan använda för att undersöka sekvens-fungerande relation. Detta resulterar i så kallade "illa-definierade"-problem där antalet funktioner överstiger antalet instanser. Många maskinlärande tekniker utnyttjar inte sådana datamängder tillfredställande. MCFS-metoden kan då användas för att minska eller elliminera icke väsentliga egenskaper.

Såvitt vi vet representerar MCFS-ID metoden en helt ny syn på analys av funktionsrika ömsesidiga beroenden. I Paper II visar vi på problemet med modellering av HIV-1 resistome, det att nätverk av ömsesidigt beroende funktioner ofta speglar molekylära interaktioner baserade på fenomen från modellen.

Vi användar vår strategi för att modellera en sekvens-funktion relation i två biologiskt viktiga problem: 1) hiv-resistens mot läkemedel, 2) bestämning av posttranslationella-modifiering platser i proteiner.

När det gäller HIV har vi inte bara kunna upptäcka flera nya muterade platser som tidigare var förknippade med motstånd, men vi har också hittat några nya (ibland tidigare försummade) platser som kan bidra till läkemedelsresistens. Genom att analysera ömsesidigt beroende nätverk kartlägger vi 3D-strukturen av HIV-1 RT, och vi kunde föreslå de möjliga molekylära mekanismer som upptäcker mutationer. Vi har validerat vårt resultat genom litteraturstudier och genom noggrann analys av tillgängliga 3D-modeller av enzymet. När det gäller post-translationella modifieringar har vi byggt ett antal modeller som kan användas för att förutsäga flera ändringstyper. Vi visade också att övervägande korta sekvensfragment inte räcker för att förutsäga flera typer av ändringar. Vi gav ett exempel på en kombination av ömsesidigt beroende grafer och (summariska-derived) summariskt härledda regler för att få insikt i klassificeringsprocessen.

8. Acknowledgements

Many people travelled by my side on this magical, rewarding journey and I would like to express my gratitude to them.

My supervisor, prof. Jan Komorowski was a great captain: patient, wise, and believing in his crew. He taught me how to avoid rocks and safely navigate to the destination port. I would like to thank him for rising my interest in computational biology and, for showing how to approach difficult problems in an elegant and efficient way.

My co-supervisor, dr. Gunnar Andersson was accompanying me on my way from virology to bioinformatics. Thank you for our long, lively discussions on science, curse of dimensionality, Sweden and life. Thank you for telling "Don't panic!" when I needed these words.

I would like to thank my first project-supervisor in Sweden, prof. Göran Akusjärvi who shared with me his passion and knowledge of virology and encouraged me to continue studies. Ett stort Tack till alla från B11 korridoren!

I would also like to thank all my co-authors: prof. Jacek Koronacki and dr. Michał Dramiński, both from Institute of Computer Science, Polish Academy of Sciences for their enthusiasm, for answering all my questions and for supporting me. Working with you was a great pleasure and a real advanture. Also our collaboration with dr. Witold Rudnicki, dr. Krzysztof Ginalski and dr. Dariusz Plewczyński from the Centre of Mathematical and Computational Modeling, University of Warsaw was very fruitful. Thank you for sharing your knowledge on statistics, protein structures and post-translational modifications. I really enjoyed my stay in Warsaw at Witold's lab.

My colleagues from The Linnaeus Centre of Bioinformatics created a really great working environment. I would like to thank Aleksejs Kontijevskis for discussions on HIV resistance and for always being my friend. I have learnt a lot from other members of Jan Komorowski's group: Adam Ameur, Álvaro Rada Iglesias, Jakub Orzechowski-Westholm, Robin Andersson, Stefan Enroth and Torgeir Hvidsten. Thank you guys! Vladimir Yankovskiy from LCB helped me many times with computational issues and was a great companion. Volodya, I miss our Friday pool evenings. I enjoyed numerous discussions with doc. Erik Bongcam-Rudloff who so passionately talks about computational biology, photography and Open Source ideas.

Helena Strömbergsson shared with me her knowledge on rough sets and drug-design. At some point we also shared office and I enjoyed really a lot!

Thank you for your enthusiasm when we were organizing Swedish Bioinformatics Workshop and for hosting me so many times with Andreas.

Nils-Einar Eriksson, Emil Lundberg, Anders Lövgren and Gustavo Gonzáles-Wall taught me a lot about system administration and were always willing to help.

For many years I have been sharing office with Álvaro Martinez Barrio. He showed me how to design databases and we wrote a bit of code together... You are such a great companion, thank you for helping me so many times and for your optimism!

During my PhD time, many people were members of LCB. I would like to thank all of them, especially: Arnaud Le Rouzic, José Álvarez Castro, François Besnier, Gabriela Aguilleta, Mikhail Velikanov, Hedi Peterson, Ewa Szczurek, Marta Łuksza and Jakub Jurkiewicz.

Jakub Gałkowski and Jakub Jurkiewicz patiently answered all my Java-related questions. Łukasz Ligowski, Ewa Szczurek and Torgeir Hvidsten helped me compiling and running ROSETTA [66].

Gunilla Nyberg and Anders Sjöberg from LCB and Sigrid Pettersson from ICM always had time and a handful of good advices! Thank you for your friendliness and efficiency!

Allison Perrigo and Jakub Orzechowski-Westholm carefully read this thesis and gave a lot of valuable comments. Allison also helped me editing the text. Hans-Henrik Fuxelius corrected my Sammanfattning på Svenska.

Without friends around me the journey wouldn't be possible. Thank you all: José for climbing Kebnekaise twice; François for sharing knowledge about wines, being my personal tennis trainer and, well, for being François!; María for the constant desire to take part in a beaver safari, all our trips and her wisdom; Arnaud for a nice co-variate trip to Oslo; Luisa for her wise advices and sense of humor; Lorenzo for replying to all my jokes and discussing fine arts. Anneleen for climbing Jebel Toubkal, quest for Fiat 126p and for all the pizza-and-gin-tonic evenings Lost somewhere in sauna. Hugo for our past, present and future adventures, in particular for saving my reputation in 1542, rescuing me in 1729 and fixing my car since 2007. Sofie for encouraging me to try African dances; Wijnand for letting me pull his volvo for soooo many miles; Hedi for being my passenger despite our ditch adventures; Volodya and Ludmila for nice dinners at their place. Misha for our future trip to Kodar; Aleksejs for his everlasting contagious optimism. Gabriela and Pablo for all our trips to the lake. Hans-Henrik for our trips and for skating. Rocío for golf lesson, dragón rojo and support. My friends and colleagues: Alex, Gosia, Patrik, Weronica, Angela, Julia, Agnes, Sonia, Karin, Mikael, Anders, Kristoffer, Elin, Samuel, Monika, Antonio, Patrik, Katia and Kasia were giving me support and sharing their knowledge. Thank you!

My Polish friends: Agnieszka and Michał, Przemek, Marta, Paweł, Łukasz and Janusz were always very supportive and encouraged me to keep going.

Special thanks to Teresa Szczenińska and Magda Gierszewska for discussions about HIV and bioinformatics, mountain trips and tons of beautiful pictures.

I am grateful to my Swedish Family: Angelina, Folker, Hartmut and Madeleine who received me so warmly. Thanks to them I have never felt alone in Sweden.

Special thanks to Martyna for being my best friend, sharing with me joys, sorrows and the passion for mountains, jazz and 1930-ties.

My Parents were always standing by my side. Thank you for your love, support and optimism. I also received a lot of support from my Family, especially from my Grandparents. Thank you for everything!

Bibliography

[1] Drug Bank, http://www.drugbank.ca.

[2] MediaWiki Commons, http://www.mediawiki.org.

[3] *The New Oxford American Dictionary*. Oxford University Press, 2005.

[4] T Ågontes, J Komorowski, and T Löken. Taming large rule models in rough set approaches. In J Zytkow and J Rauch, editors, *Principles of Data Mining and Knowledge Discovery*, volume 1704 of *Lecture Notes in Artificial Intelligence*, pages 193–203. Springer, 1999.

[5] R Aguair and BM Peterlin. APOBEC3 proteins and reverse transcription. *Virus Research*, 134:74–85, 2008.

[6] B Alberts and A Johnson. *Molecular Biology of the Cell*. Taylor and Francis INC, 2008.

[7] DG Altman and JM Bland. Diagnostic tests 2: Predictive values. *BMJ*, 309:102, 1994.

[8] V Appay and D Sauce. Immune activation and inflammation in HIV-1 infection: causes and consequences. *Journal of Pathology*, 214(2):231–241, 2008.

[9] A Baddeley. The Magical Number of Seven: Still Magic After All These Years. *Psychological Review*, 1994.

[10] A Bairoch and R Apweiler. The Swiss-Prot protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Research*, 27:49–54, 1999.

[11] VP Basu, M Song, L Gao, ST Rigby, MN Hanson, and RA Bambara. Strand transfer events during HIV-1 reverse transcription. *Virus Research*, 134:19–38, 2008.

[12] JD Bauman, K Das, WC Ho, M Baweja, DM Himmel, AD Clark Jr, DA Oren, PL Boyer, SH Hughes, AJ Shatkin, and E Arnold. Crystal engineering of HIV-1 reverse transcriptase for structure-based drug design. *Nucleic Acid Research*, 36:5083–5092, 2008.

[13] N Beerenwinkel, B Schmidt, H Walter, R Kaiser, T Lengauer, D Hoffmann, K Korn, and J Selbig. Geno2pheno: Interpreting genotypic HIV drug resistance tests. *IEEE Intellig Syst*, 16:35–41, 2001.

[14] N Beerenwinkel, B Schmidt, H Walter, R Kaiser, T Lengauer, D Hoffmann, K Korn, and J Selbig. Diversity and complexity of HIV-1 drug resistance: A bioinformatics approach to predicting phenotype from genotype. *Proc Natl Acad Sci USA*, 99(12):8271–8276, 2002.

[15] RE Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[16] N Biggs. *Discrete Mathematics*. Oxford University Press, 2003.

[17] N Blom, S Gammeltoft, and S Brunak. Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. *J Mol Biol*, 294(5):1351–1362, 1999.

[18] F Bonnet, C Lewden, and T May. Malignancy-related causes of death in human immunodeficiency virus-infected patients in the era of highly active antiretroviral therapy. *Cancer*, 101(2):317–324, 2004.

[19] L Breiman. Random Forest. *Machine Learning*, 45:5–32, 2001.

[20] M Breitbart and F Rohwer. Here a virus, there a virus, everywhere the same virus? *Trends Microbiol*, 13:278–284, 2005.

[21] JM Brenchley, DA Price, TW Schacker, TE Asher, G Silvestri, S Rao, Z Kazzaz, E Bornstein, O Lambotte, D Altmann, BR Blazar, B Rodriguez, L Teixeira-Johnson, A Landay, J N Martin, F M Hecht, LJ Picker, MM Lederman, SG Deeks, and DC Douek. Microbial translocation is a cause of systemic immune activation in chronic HIV infection. *Nature Medicine*, 12(12):1365–1371, 2006.

[22] JM Brenchley, TW Schacker, LE Ruff, DA Price, JH Taylor, GJ Beilman, PL Nguyen, A Khoruts, M Larson, AT Haase, and DC Douek. $CD4^+$ T cell depletion during all stages of HIV disease occurs predominantly in the gastrointestinal tract. *Journal of Experimental Medicine*, 200(6):749–759, 2004.

[23] L Brocchieri and S Karlin. Protein length in eukaryotic and prokaryotic proteomes. *Nucleic Acids Research*, 33:3390–3400, 2005.

[24] AW Burgess, PK Ponnuswamy, and HA Scheraga. Analysis of conformations of amino acid residues and prediction of backbone topography in proteins. *Isr J Chem*, 12:239–286, 1974.

[25] V Calvez and B Masquelier (coords.). Hiv-1 genotypic drug resistance interpretetion's algorithms.

[26] W Cardona-Maya, P Velilla, CJ Montoya, A Cadavid, and MT Rugeles. Presence of HIV-1 DNA in Spermatozoa from HIV-Positive Patients: Changes in the Semen Parameters. *Current HIV Research*, 7:418–424, 2009.

[27] M Charton and BI Charton. The structural dependence of amino acid hydrophobicity parameters. *J Theor Biol*, 99:629–644, 1982.

[28] C Chen, A Liaw, and L Breiman. Using Random Forest to Learn Imbalanced Data. Technical report, Department of Statistics, University of California, Berkeley, USA, 2004.

[29] JH Condra, WA Schleif, OM Blahy, LJ Gabryelski, DJ Graham, J Quintero, A Rhodes, HL Robbins, E Roth, M Shivaprakash, D Titus, T Yang, H Tepplert, KE Squires, PJ Deutsch, and EA Emini. In vivo emergence of HIV-1 variants resistant to multiple protease inhibitors. *Nature*, 374:569–571, 2002.

[30] N Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Beh Brain Sci*, 24:87–185, 2001.

[31] JL Crawford, WN Lipscomb, and CG Schellman. The reverse turn as a polypeptide conformation in globular proteins. *Proc Natl Acad Sci USA*, 70:538–542, 1973.

[32] P Dalgaard. *Introductory Statistics with R*. Statistics and Computing. Springer, 2002.

[33] A De Luca, M Vendittelli, and F Boldini. Construction, training and clinical validation of an interpretation system for genotypic HIV-1 drug resistance based on fuzzy rules revised by virological outcomes. *Antivir Ther*, 99(9):583–593, 2002.

[34] R Diestel. *Graph Theory*. Springer-Verlag, 2005.

[35] J Ding, K Das, Y Hsiou, SG Sarafianos, AD Clark Jr, A Jacobo-Molina, C Tantillo, SH Hughes, and E Arnold. Structure and Functional Implications of the Polymerase Active Site Region in a Complex of HIV-1 RT with a Double-stranded DNA Template-primer and an Antibody Fab Fragment at 2.8Å Resolution. *J Mol Biol*, 284:1095–1111, 1998.

[36] S Draghici and RB Potter. Predicting HIV drug resistance with neural networks. *Bioinformatics*, 19:98–107, 2003.

[37] T Dragic, A Trkola, DA Thompson, EG Cormier, FA Kajumo, E Maxwell, SW Lin, W Ying, SO Smith, TP Sakmar, and JP Moore. A binding pocket for a small molecule inhibitor of HIV-1 entry within the transmembrane helices of CCR5. A binding pocket for a small

molecule inhibitor of HIV-1 entry within the transmembrane helices of CCR5. *Proc Natl Acad Sci USA*, 97:5639–5644, 2000.

[38] JW Drake. Rates of spontaneous mutation among RNA viruses. *Proc Natl Acad Sci USA*, 90:4171–4175, 1990.

[39] M Dramiński, M Kierczak, J Koronacki, and J Komorowski. Monte Carlo feature selection and interdependency discovery in supervised classification. *in press*, 2009.

[40] M Dramiński, A Rada Iglesias, S Enroth, C Wadelius, J Koronacki, and J Komorowski. Monte Carlo feature selection for supervised classification. *Bioinformatics*, 24:110–117, 2008.

[41] R Durbin, SR Eddy, A Krogh, and G Mitchison. *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, 1998.

[42] TH Eickbush and VK Jamburuthugoda. The diversity of retrotransposons and the properties of their reverse transcriptases. *Virus Research*, 134:221–234, 2008.

[43] P Emau, B Tian, BR O'keefe, T Mori, JB McMahon, KE Palmer, Y Jiang, G Bekele, and CC Tsai. Griffithsin, a potent HIV entry inhibitor, is an excellent candidate for anti-HIV microbicide. *J Med Primatol*, 36:244–253, 2007.

[44] P Eykoff. *System Identification: Parameter and State Estimation*. Wiley and Sons, 1974.

[45] JL Fauchere, M Charton, LB Kier, A Verloop, and V Pliska. Amino acid side chain parameters for correlation studies in biology and pharmacology. *Int J Pept Prot Res*, 32:269–278, 1988.

[46] AS Fauci. 25 years of HIV. *Nature*, 453:289–290, 2008.

[47] T Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874, 2005.

[48] RA Fisher. *The design of experiments*. Oliver and Boyd, Edingurgh, 1935.

[49] P Fletcher, S Harman, H Azijn, N Armanasco, P Manlow, D Perumal, Marie-Pierre de Bethune, J Nuttall, J Romano, and J Shattock. Inhibition of human immunodeficiency virus type 1 infection by the candidate microbicide dapivirine, a nonnucleoside reverse transcriptase inhibitor. *Antimicrobial Agents and Chemotherapy*, 53:487–495, 2009.

[50] M Gashler, C Giraud-Carrier, and T Martinez. Decision Tree Ensemble: Small Heterogenous is Better Than Large Homogenous. In *Seventh International Conference on Machine Learning and Applications (ICMLA 08)*, pages 900–905, 2008.

[51] R Grantham. Amino acid difference formula to help explain protein evolution. *Science*, 185:862–864, 1974.

[52] JA Hanley and BJ McNeil. The meaning and use of the area under a receiver operating characteristics (ROC) curve. *Radiology*, 143:29–36, 1982.

[53] D Harris, N Kaushik, PK Pandey, and VN Pandey PNS Yadav. Functional analysis of amino acid residues constituting the dNTP binding pocket of HIV-1 reverse transcriptase. *Journal of Biological Chemistry*, 273:33624–33634, 1998.

[54] T Hastie, R Tibshirani, and J Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

[55] Z Hel, J R McGhee, and J Mestecky. HIV infection: first battle decides the war. *Trends in Immunology*, 27(6):274–281, 2006.

[56] PG Higgs and TK Atwood. *Bioinformatics and Molecular Evolution*. Blackwell Publishing, 2005.

[57] C Hoffmann, JK Rockstroh, and BS Kamps, editors. *HIV Medicine 2007*. Flying Publisher, Paris, Cagliari, Wuppertal, 2007.

[58] H Huang, R Chopra, GL Verdine, and SC Harrison. Structure of a covalently trapped catalytic complex of HIV-1 reverse transcriptase: implications for drug resistance. *Science*, 282:1669–1675, 1998.

[59] TR Hvidsten. *Predicting Function of Genes and Proteins from Sequence, Structure and Expression Data*. PhD thesis, The Linnaeus Centre for Bioinformatics, Uppsala University, Uppsala, Sweden, 2004.

[60] A Jacobo-Molina, J Ding, RG Nanni, AD Clark Jr., X Lu, C Tantillo, RL Williams, G Kamer, AL Ferris, P Clark, A Hizi, SH Hughes, and E Arnold. Crystal structure of human immunodeficiency virus type 1 reverse transcriptase complexed with double-stranded DNA at 3.0 Å resolution shows bent DNA. *Proc Natl Acad Sci USA*, 90:6320–6324, 1993.

[61] MA Jobling, ME Hurles, and C Tyler-Smith. *Evolutionary Genetics*. Garland Science, 2004.

[62] H Jonckheere, J Anne, and E De Clerq. The HIV-1 reverse transcription (RT) process as target for RT inhibitors. *Medicinal Research Reviews*, 20:129–154, 2000.

[63] J Kær, L Høj, Z Fox, and JD Lundgren. Prediction of phenotypic susceptibility to antiretroviral drugs using physicochemical properties of the primary enzymatic structure combined with artificial neural networks. *HIV Medicine*, 9:642–652, 2008.

[64] S Kawashima, H Ogata, and M Kanehisa. AAindex: amino acid index database. *Nucleic Acids Research*, 27:368–369, 1999.

[65] M Kierczak, WR Rudnicki, and J Komorowski. Construction of Rough Set-Based Classifiers for Predicting HIV Resistance to Nucleoside Reverse Transcriptase Inhibitors. In R Bello, Rafael Falcón, Witold Pedrycz, and J Kacprzyk, editors, *Granular Computing: At the Junction of Rough Sets and Fuzzy Sets*, volume 224 of *Studies in Fuzziness and Soft Computing*, pages 249–258, Berlin/Heidelberg, 2008. Springer.

[66] J Komorowski, A Øhrn, and A Skowron. The ROSETTA rough set software system. In W Klösgen and J Zytkow, editors, *Handbook of Data Mining and Knowledge Discovery*, pages 554–559. Oxford University Press, 2002.

[67] J Komorowski, Z Pawlak, L Polkowski, and A Skowron. Rough sets: A tutorial. In SK Pal and A Skowron, editors, *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, pages 3–98. Springer, 1999.

[68] A Kontijevskis. *Modelling the Interaction Space of Biological Macromolecules: A Proteochemometric Approach. Applications for Drug Discovery and Development*. PhD thesis, The Linnaeus Centre for Bioinformatics, Uppsala University, Uppsala, Sweden, 2008.

[69] EV Koonin, TG Senkevich, and VV Dolja. The ancient Virus World and evolution of cells. *Biol Direct*, 1(29), 2006.

[70] J Koronacki and J Ćwik. *Statystyczne Systemy Uczące się*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, secnd edition, 2008.

[71] A Kreegipuu, N Blom, and S Brunak. PhosphoBase, a database of phosphorylation sites: release 2.0. *Nucl Acid Res*, 27(1):237–239, 1999.

[72] B Larder, D Wang, A Revell, J Montaner, R Harrigan, F De Wolf, J Lange, S Wegner, L Ruiz, MJ Pérez-Elías, S Emery, J Gatell, A D'Arminio Monforte, C Torti, M Zazzi, and C Lane. The development of artificial neural networks to predict virological response to combination HIV therapy. *Antivir Ther*, 12:15–24, 2007.

[73] F Lindgren, B Hansen, W Karcher, M Sjöström, and L Eriksson. Model validation by permutation tests: applications to variable selection. *J Chemom*, 10:521–532, 1996.

[74] A Lindström. *Resistance to antiviral drugs in HIV and HBV*. PhD thesis, Microbiology and Tumor Biology Center, Karolinska Institutet and the Swedish Institute for Infectious Disease Control, Stockholm, Sweden, 2005.

[75] DJ MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003-2004.

[76] E Mąkosa. Rule Tuning. Master's thesis, The Linnaeus Centre for Bioinformatics, Uppsala University, Uppsala, Sweden, 2005.

[77] M Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *CML-2003 workshop on learning from imbalanced data sets II*, 2003.

[78] ER Mardis. Next-Generation DNA Sequencing Methods. *Annual Review of Genomics and Human Genetics*, 9:387–402, 2008.

[79] S Mehandru, MA Poles, K Tenner-Racz, A Horowitz, A Hurley, C Hogan, D Boden, P Racz, and M Markowitz. Primary HIV-1 infection is associated with preferential depletion of $CD4^+$ T lymphocytes from effector sites in the gastrointestinal track. *Journal of Experimental Medicine*, 200(6):761–770, 2004.

[80] L Menédez-Arias. Mechanisms of resistance to nucleoside analogue inhibitors of HIV-1 reverse transcriptase. *Virus Research*, 134:124–146, 2008.

[81] L Menédez-Arias and B Berkhout. Retroviral reverse transcription. *Virus Research*, 134:1–3, 2008.

[82] GA Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psych Rev*, 63:81–97, 1956.

[83] T Mitchell. *Machine Learning*. McGraw Hill, 1997.

[84] M Nassal. Hepatitis B viruses: reverse transcription a different way. *Virus Research*, 134:235–249, 2008.

[85] A Øhrn. *Discernibility and Rough Sets in Medicine: Tools and Applications*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 1999.

[86] S Okasha. *Philosophy of science*. A very short introduction. Oxford University Press, 2002.

[87] BB Oude Essink and B Berkhout. The fidelity of reverse transcription differs in reactions primed with RNA versus DNA primers. *J Biomed Sci*, 6:121–132, 1999.

[88] Z Pawlak. Rough sets. *Int J Inform Comp Science*, 11:341–356, 1982.

[89] Z Pawlak. Rough sets: theoretical aspects of reasoning about data. In *Theory and decision library.*, System theory, knowledge engineering and problem solving, page 229. Kluwer Academic Publishers, 1991.

[90] AS Perelson, AU Neumann, and M Markowitz. HIV-1 dynamics in vivo: Virion clearance rate, infected cell life-span, and viral generation time. *Science*, 271:1582–1586, 1996.

[91] K Popper. *Conjectures and Refutations*. Routledge, 1963.

[92] F Provost, T Fawcett, and R Kohavi. The case against accuracy estimation for comparing classifiers. In *Proceedings of the Fifteenth International Conference on Machine Learning*, San Francisco, 1998. Morgan Kaufmann.

[93] C Quiken, B Foley, P Marx, S Wolinsky, T Leitner, B Hahn, F McCutchan, and B Corber. HIV Sequence Compendium. Technical report, Los Alamos National Laboratory, Theoretical Biology and Biophysics Group, 2008.

[94] A Radzicka and R Wolfenden. Comparing the polarities of the amino acides. Side-chain distribution coefficients between the vapor phase, cyclohexane, 1-octanol and neutral aqueous solution. *Biochemistry*, 27:1664–1670, 1988.

[95] J Ren, R Esnouf, E Garman, D Somers, C Ross, I Kirby, J Keeling, G Darby, Y Jones, and D Stuart. High resolution structures of HIV-1 RT from four RT-inhibitor complexes. *Nat Struct Biol*, 2(293-302), 1995.

[96] J Ren and DK Stammers. Structural basis for drug resistance mechanisms for non-nucleoside inhibitors of HIV reverse transcriptase. *Virus Research*, 134:157–170, 2008.

[97] LF Rezende and VR Prasad. Nucleoside-analog resistance mutations in HIV-1 reverse transcriptase and their influence on polymerase fidelity and viral mutation rates. *Int J Biochem Cell Biol*, 36:1716–1734, 2004.

[98] SY Rhee, J Taylor, G Wadhera, A Ben-Hur, DL Brutlag, and RW Schafer. Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences of the United States of America*, 103:17355–17360, 2006.

[99] BD Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[100] JL Rodgers. The Bootstrap, the Jackknife, and the Randomization Test: A Sampling Taxonomy. *Multivariate Behav Res*, 34(4):441–456, 1999.

[101] WR Rudnicki, M Kierczak, J Koronacki, and J Komorowski. A Statistical Method for Determining Importance of Variables in an Information System. In S Greco, Y Hata, S Hirano, M Inuiguchi, S Miyamoto, HS Nguyen, and R Słowiński, editors, *Rough Sets and Current Trends in Computing*, volume 4259 of *Lecture Notes in Computer Science*, pages 557–566, Berlin/Heidelberg, November 2006. Springer.

[102] WR Rudnicki and J Komorowski. Feature synthesis and extraction for the construction of generalized properties of amino acids. In S Tsumoto, R Słowiński, and J Komorowski, editors, *Rough Sets and Current Trends in Computing*, volume 3066 of *Lecture Notes in Computer Science*, pages 786–791, Berlin/Heidelberg, June 2004. Springer.

[103] AL Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal*, 3(3):210–229, 1959.

[104] SG Sarafianos, K Das, J Ding, PL Boyer, SH Hughes, and E Arnold. Touching the heart of HIV-1 drug resistance: the fingers close down on the dNTP at the polymerase active site. *Chemistry and Biology*, 6:R137–R146, 1999.

[105] B Schmidt, H Walter, N Zeitler, and K Korn. Genotypic Drug Resistance Interpretation Systems – The Cutting Edge of Antiretroviral Therapy. *AIDS Rev*, 4:148–156, 2002.

[106] D Schwartz, MF Chou, and GM Church. Predicting protein post-translational modifications using meta-analysis of proteome scale data sets. *Mol Cell Proteomics*, 8(2):365–379, 2009.

[107] MR Segal. Machine Learning Benchmarks and Random Forest Regression. Technical report, Center for Bioinformatics and Molecular Biostatistics, University of California, http://repositories.cdlib.org/cgi/viewcontent.cgi?article=1012context=cbmb, 2004.

[108] C Seibert, W Ying, S Gavrilov, F Tsamis, SE Kuhmann, A Palani, JR Tagat, JW Clader, SW McCombie, BM Baroudy, SO Smith, T Dragic, JP Moore, and TP Sakmar. Interaction of small molecule inhibitors of HIV-1 entry with CCR5. *Virology*, 349:41–54, 2006.

[109] PM Sharp and BH Hahn. AIDS: Prehistory of HIV-1. *Nature*, 455(605-606), 2008.

[110] J Shendure and H Ji. Next-generation DNA sequencing. *Nature Biotechnology*, 26(10):1135–1145, 2008.

[111] A Skowron and HS Nguyen. Boolean reasoning scheme with some applications in data mining. In JM Zytkow and J Rauch, editors, *Third European Symposium on Principles and Practice of Knowledge Discovery in Databases*, Lecture Notes in Artificial Intelligence, pages 107–115. Springer-Verlag, 1999.

[112] A Skowron and C Rauszer. The discernibility matrices and functions in information systems. In R Sński, editor, *Intelligent Decision Support: Handbook of Applications and Advances in Rough Sets Theory*, pages 331–362. Kluwer Academic Publishers, 1992.

[113] L Stryer, JM Berg, and JL Tymoczko. *Biochemistry*. WH Freeman, international edition edition, 2006.

[114] B Sunyer, W Diao, and G Lubec. The role of post-translational modifications for learning and memory formation. *Electrophoresis*, 29(12):2593–2602, 2008.

[115] J Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

[116] UNAIDS. 2008 report on the global aids epidemic.

[117] U.S. Food and Drug Administration. Antiretroviral drugs used in the treatment of HIV infection.

[118] V Valverde-Garduño, P Gariglio, and L Gutiérrez. Functional analysis of HIV-1 reverse transcriptase motif C: site-directed mutagenesis and metal cation interaction. *Journal of Molecular Evolution*, 47:73–80, 1998.

[119] Mark A Wainberg. HIV-1 subtype distribution and the problem of drug resistance. *AIDS*, 18:S63–S68, 2004.

[120] K Wang, E Jenwitheesuk, R Samudrala, and JE Mittler. Simple linear model provides highly accurate genotypic predictions of HIV-1 drug resistance. *Antivir Ther*, 9:343–352, 2004.

[121] Marx W Wartofsky. *Conceptual Foundations of Scientific Thought: An Introduction to the Philosophy of Science*. Collier-Macmillan, 1968.

[122] IH Witten and E Frank. *Data Mining. Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[123] M Worobey, M Gemmel, DE Teuwen, T Haselkorn, K Kunstman, M Bunce, J-J Muyembe, J-MM Kabongo, RM Kalengayi, E VanMarck, MTP Gilbert, and SM Wolinsky. Direct evidence of extensive diversity of HIV-1 in Kinshasa by 1960. *Nature*, 455, 2008.

[124] W Yong-Xiang, Z Hao-Jie, X Jing, Z Bo-Jian, and W Yu-Mei. Mutational analysis of the 'turn' of helix clamp motif of HIV-1 reverse transcriptase. *Biochemical Biophysical Research Communications*, 377:915–920, 2008.

[125] LA Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[126] J Zhang, SY Rhee, J Taylor, and RW Shafer. Comparison of the precision and sensitivity of the Antivirogram and PhenoSense HIV drug susceptibility assays. *Journal of Acquired Immune Deficiency Syndromes*, 38:439–444, 2005.

[127] JM Zimmerman, N Eliezer, and R Simha. The characterization of amino acid sequences in proteins by statistical methods. *J Theor Biol*, 21:170–201, 1968.

Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. (Prior to January, 2005, the series was published under the title "Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology".)

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2009

Distribution: publications.uu.se
urn:nbn:se:uu:diva-109873