

## Deliverable 4.1 Virtualisation and Automation

<b>Project start:</b>	1 March 2006
<b>Project Duration:</b>	36 months
<b>Priority area:</b>	2.4.11
<b>Contract No.:</b>	INFSO-IST-027446
<b>Website:</b>	<a href="http://www.virolab.org">http://www.virolab.org</a>

<b>Due-Date:</b>	Month 12
<b>Delivery:</b>	Resubmitted month 16
<b>Lead Partner:</b>	University of Amsterdam
<b>Project Leader</b>	Prof. P. M. A. Sloot
<b>Dissemination Level:</b>	public
<b>Status:</b>	final
<b>Approved:</b>	Steering Committee, Quality Board
<b>Version:</b>	2.0

# Log of Document

Version	Date	Changes Summary	Authors
1.0	11-04-2007	First version in Word format	Breannán Ó Nualláin
1.1	12-04-2007	Included suggestions by P. Sloot: Future work, abbrevs, publishing	Breannán Ó Nualláin
1.2	13-04-2007	List of Publications included	Gohar Sargsyan
1.3	13-04-2007	Corrected Lead Partner	Breannán Ó Nualláin
1.4	13-04-2007	Updated References and table of content	Gohar Sargsyan
2.0	29-06-2007	Rewritten to reflect reviewers' comments	Breannán Ó Nualláin

# Table of Contents

<b>1</b>	<b>Executive summary</b>	<b>5</b>
<b>2</b>	<b>Description</b>	<b>6</b>
2.1	Clinical guidelines and protocols . . . . .	6
2.2	HIV drug resistance interpretation systems . . . . .	6
2.3	State of the Art . . . . .	8
2.4	Design decisions . . . . .	9
2.5	Shortcomings of Retrogram . . . . .	9
2.6	Shortcomings of ASI . . . . .	10
2.7	Virtualisation . . . . .	12
<b>3</b>	<b>Status</b>	<b>12</b>
<b>4</b>	<b>Drug Ranking System Virtualisation</b>	<b>13</b>
4.1	Drug Ranking System Formats . . . . .	13
4.1.1	Retrogram Specification Document . . . . .	13
4.1.2	Retrogram Specification Formalised in Prolog . . . . .	14
4.1.3	Violab Prolog Rule Format . . . . .	14
4.1.4	ELF Executable . . . . .	15
4.2	Virtualisation components . . . . .	15
4.2.1	Derivation of the Retrogram Specification in Prolog . . . . .	15
4.2.2	Conversion of Prolog Retrogram Specification to Violab Prolog Rule Format . . . . .	16
4.2.3	ASI XML Parser to Violab Prolog Rule Format . . . . .	16
4.2.4	Violab Prolog Rule Unparser . . . . .	16
4.2.5	VPRF Publisher . . . . .	16
4.2.6	ASI Condition Parser . . . . .	16
4.2.7	VPRF Condition Optimiser . . . . .	17
4.2.8	VPRF Compiler . . . . .	17
4.3	Future work . . . . .	17
4.4	Abbreviations . . . . .	17

# List of Figures

4.1 Building the Drug Ranking System . . . . . 14

# 1 Executive summary

This deliverable describes the work carried out in Work Package 4 *'Retrogram: Virtualisation, Enhancement and Individual-based Interpretation Systems'*, Tasks 4.1 *'Virtualisation'* and 4.2 *'Automation'* in the period Months 1-12 of the Virolab project.

The work reported here must be seen in the overall context of Virolab project objectives [3, p7], in particular Objective 3: *'To virtualise and enhance the state of the art in genotypic resistance interpretation tools and integrate them into the virtual laboratory'*

This involved the virtualisation of the monolithic Retrogram interpretation into its components. These virtualised components would then serve as the interpretation system at the core of the Virolab virtual laboratory. In particular the virtualisation was targeted at providing an open and flexible system in order to allow easy addition of enhancements from:

- molecular dynamics calculations of drug binding affinities,
- cellular automata models of cell entry,
- biostatistics,
- agent-based models,
- subtyping & epidemiological results, and,
- literature mining for causal relations among HIV mutations and drugs.

For this reason a more flexible and expressive rule language was needed with a fully specified, formal semantics. This new rule language is expressive enough to allow the inclusion of not only the Retrogram interpretation system, but also three other ASI-based interpretation systems: Stanford HIVdb, Rega and ANRS.

The necessity for automation (Task 4.2) arises because different peripheral hospitals have different ways of processing blood samples to determine the genotypes of the virus. Once PCR and mutation detection have taken place, it is necessary to ensure that the resulting data fit the generic input format of decision support system. This automation should not be understood in the sense of hardware interfacing to the PCR apparatus in the hospitals but as providing the tools for hospital workflow and data interfacing.

## 2 Description

### 2.1 Clinical guidelines and protocols

In medicine in recent decades there has been a significant move towards the establishment of clinical guidelines and protocols for the formulation of knowledge. Such protocols are more or less precise recommendations about diagnoses, treatments or medical procedures and are based on the best evidence-based reviews available at the time. They have been described as “systematically developed statements to assist practitioners and patient decisions about appropriate health care for specific circumstances.” [4]

Evidence-based medicine is the “conscientious, explicit, and judicious use of current best evidence in making decisions about the care of individual patients”.[8]

It has been shown that formulation of clinical protocols can benefit the quality of healthcare outcomes; can help promote high quality practice; and can reduce variation in care.[11] It has further been shown that adherence to protocols can significantly reduce the costs of health care.[2]

Other benefits are improving quality assurance, guiding data collection, better interpretation and management of the patients status, activation of alerts and reminders and improved decision support.[7]

In addition a carefully formulated protocol provides a condensed presentation of current knowledge in its field of application. This greatly eases the burden on medical practitioners of having to keep abreast of an ever increasing flow of literature in their area.

Clinical protocols take many forms, from free text through flowcharts, decision tables, medical logic modules to expert systems.

### 2.2 HIV drug resistance interpretation systems

In the area of HIV drug resistance, such clinical protocols have taken the form of interpretation systems. Such systems have been and continue to be developed. They make predictions of resistance to available drugs for HIV suppression based on the mutations present in an infected patient. They vary in approach from mutation lists through rule-based systems and correlations of geno-

types with corresponding phenotypic susceptibilities.

An advantage of encoding drug resistance information in a rule-based system is that a large quantity of such information can be encoded, checked and verified and then presented to the clinician in a user-friendly way.

An early such system was Retrogram, an expert-based interpretation system which weighed the effect of specified genotype changes on clinical drug activity. The decision support technology used in the implementation of RetroGram was AREZZO, based on PROforma and developed by InferMed Ltd., London. The medical content was developed by Virology Networks. RetroGram was jointly owned by Virology Networks and Hoffmann-La Roche but the rights were later transferred to the University of Amsterdam. It was used by > 250 clinicians in HIV treatment centres in 26 countries.

RetroGram relates complex genomic information on HIV-1 resistance to the clinical suitability of anti-retroviral drugs. The system generates a suitability ranking of all FDA approved drugs based on their expected clinical efficacy for an individual patient. It includes the experience and opinions of experts in the algorithm to weigh the effect of specific genotypic changes. Retrogram targets the Reverse Transcriptase and Protease proteins. It accepts input through a web browser in the form of amino-acid substitutions relative to the NL4-3 reference genome. The output is a ranking of each available drug based on (n-fold) resistance, confidence (based on evidence *in vivo* or *in vitro*) and suitability.

The most recent version of Retrogram was specified in the Clinical Specification (v1.6, 23 January 2004) and Rules & References (v1.6 2 July 2002).

We chose Retrogram for the rule system at the core of our Virolab Virtual Laboratory for several reasons:

**clinical validation** The main reason is that Retrogram has been clinically validated and repeatedly showed significant predictivity of virological outcomes using independent populations of HIV treatment experienced patients.

**comparison to other systems** Reports that compared the ability of available systems to predict clinical responses have found significant interpretation discordance with a remarkable number of systems failing to correctly predict virologic and immunologic outcome. Importantly, Retrogram always showed significant predictivity of virologic and immunologic outcomes using different populations of HIV treatment experienced patients. The largest

study that compared HIV-1 drug resistance interpretation tools was performed by De Luca *et al.*[5] This study included 261 subjects for whom a potent antiretroviral regimen was failing and who were therefore starting salvage therapy. De Luca *et al.* found that only five, including Retrogram, of the 11 evaluated interpretation systems showed independent correlation with virologic changes after six months. (The other four systems were all non-European: Stanford, Guidelines 3.0, HIVresistanceWeb and São Paulo University) Importantly, Retrogram was the interpretation tool that most strongly predicted the increase after six months of CD4-count. Torti *et al.*[9] studied 173 patients and found that the use of Retrogram was independently associated with undetectable viral loads after 16 weeks. Conversely, interpretation according to the biological cut-offs in the virtual or real phenotype system was not related to undetectable viral loads. Finally, Tural *et al.*[10] investigated the clinical utility of Retrogram compared to expert advice. It was found that Retrogram improved the virological outcome in both the presence and absence of expert advice.

**experience** Partners 1 and 2 of ViroLab have closely collaborated in the development of Retrogram and have therefore gained the most experience with this acclaimed HIV-1 drug resistance interpretation tool. All research groups involved in the comparative studies mentioned above participate in ViroLab as partner.

## 2.3 State of the Art

Since the most recent version of Retrogram, the state of the art in HIV drug resistance interpretation systems has progressed somewhat. The *de facto* standard for the expression of rules concerning mutation-influenced drug resistance has become the Algorithm Specification Interface (ASI)[1], developed at Stanford University as part of their HIV Drug Resistance Database project (<http://hivdb.stanford.edu/>).

Many institutes that develop rule-based interpretation systems publish their rule sets in the ASI format, among them Rega Institute (<http://www.kuleuven.ac.be/reg/>), Agence Nationale de Recherche sur la SIDA (<http://www.anrs.fr>) and Stanford HIVdb itself.

ASI provides a formal language for describing the rules which are encoded in a drug interpretation system, independent of the language and computing environment used to program the system. The language is defined as an XML Document Type Definition (DTD) and a compiler is provided to translate a rule set supplied in ASI into the programming language Perl. The ASI compiler is also written in Perl.

The availability of ASI has allowed developers of drug interpretation systems publish their systems and to compare them and has prevented to some degree lock-in to inaccessible proprietary formats. It also allows developers to focus on enhancing their algorithms rather than concentrating on developing software to encode them.

## **2.4 Design decisions**

Having decided to use Retrogram as the core of our Virolab Virtual Laboratory and, recognising ASI as the *de facto* standard for representing HIV drug resistance genotype interpretation, we were faced with some design decisions. Expressing Retrogram in ASI appeared the obvious step but both Retrogram and ASI had shortcomings which needed to be addressed.

## **2.5 Shortcomings of Retrogram**

Retrogram was the result of a research project which had run its course and, as a result, it had not been updated for several years. It had been implemented on a monolithic proprietary medical expert-system platform called Arezzo which had three drawbacks: it was expensive, it was proprietary and closed-source and it did not lend itself to virtualisation.

The essential content of Retrogram, namely the rules and specifications for their interpretation, was however available in the form of the documents The Retrogram Clinical Specification (v1.6 23 January 2004) and Retrogram Rules & References (v1.6 2 July 2002). It was therefore decided to reimplement Retrogram using open-source tools in an established and standardised programming language, well suited for the purpose, namely Prolog.

## 2.6 Shortcomings of ASI

In order to describe the shortcomings of the Algorithm Specification Interface, we take a step back to consider the potential benefits of encoding a clinical protocol in such a way. ASI has undoubtedly been beneficial for the reasons mentioned above, however the ASI specification falls short of defining an adequate formal semantics for the language, limiting itself to the observation that “The language is very much fairly self-explanatory given that it reads similarly to an English sentence.” As a result, there is no basis for making judgements of the following kinds about interpretation systems expressed in ASI:

**ambiguity** Is it the case that a rule set is internally ambiguous? Does it allow more than one interpretation?

**completeness** Does a rule set have complete coverage?

**consistency** Are there rules in the set which make contradictory predictions?

**redundancy** Does one rule subsume another?

**predictive power** Can one rule set make more specific predictions than another or can it predict in cases where the other is silent?

This meant that ASI is inadequate for many of the key aims of the Virolab project, among which:

- comparison of rule systems,
- validation of rule systems,
- checking of rule-system anomalies,
- cross-validation of rule systems,
- investigation of rule-set discordances.

The provision of a declarative formal semantics for a rule-set specification language would not only allow all of these aims to be achieved, it would allow them to be automated using model checkers and automated theorem provers.

Other issues mitigating against ASI are:

- Despite providing an informal semantics for rule firing (<http://hivdb.stanford.edu/pages/asi/>), ASI has no semantics whatsoever for rule conditions. Several examples are given but among them are examples which do not adhere to the given BNF syntax definitions and others which are ambiguous.
- ASI is defined as a Document Type Definition (DTD), a formalism which has been more recently exposed as unsuitable for document validation and has been succeeded by improved formalisms such as RelaxNG.
- ASI provides no justification for apparently indiscriminate design choices such as the mixing of attributes with subelements.
- No semantics whatsoever is given for GLOBALRANGE values.
- And, perhaps most insidious, the day-to-day use of ASI seems to have overtaken the specification with features appearing in more recent versions of HIVdb rule sets which are not defined. As a result, the specification is outdated (“Last update 10/00.”) and the only current specification and semantics for ASI is its implementation in the Perl compiler, precisely such an eventuality as ASI was designed to prevent.

Aside from these informatics-related judgements concerning ASI, there is a virological need for a more expressive language. Recent findings have revealed the need to express multiplicative effects of certain mutations on certain drugs. ASI, in its present form, is limited to expressing linear combinations of resistance effects.

We note in passing a potential concomitant benefit of formally specifying medical protocols. An earlier STREP project, *Protocure: integrating formal methods in the development process of medical guidelines and protocols* ([www.protocure.org](http://www.protocure.org)), 2004-2006, carried out case studies[6] on the formalisation of two separate medical protocols and uncovered a significant number of anomalies in both protocols. This was particularly surprising since the protocols in question were of the highest quality produced by the medical profession. Formalisation can thus lead to important improvements in existing medical protocols.

## 2.7 Virtualisation

As part of the enhancements of Work Package 4, we have designed an enhanced language with a formal semantics for the expression of drug interpretation systems. This will be reported in Deliverable 4.3.

Our first steps, reported in this Deliverable, were to virtualise Retrogram and express it in ASI. In doing so, we wrote our own ASI compiler in Prolog, a programming language far more suited to the purpose, due to its recursive structure, its explicit data types for terms and partial structures and its superior parsing functionality.

While, at first sight, it may seem that we have done nothing more than reproduce the functionality of the Stanford system and cast Retrogram in that mould, there are two important benefits from this approach. Firstly we can expose any potential differences in interpretation between the Stanford ASI compiler and ours, potentially leading to a better confidence in both compilers. And secondly, the loosely-coupled nature of our implementation allows easy distributed deployment in the Grid context of the Virolab Virtual Laboratory.

## 3 Status

We have successfully performed the virtualisation of the Retrogram drug ranking system. The process of virtualisation required the following steps:

1. We formed a thorough understanding and analysis of the Retrogram specification document and Guidelines for Usage including rule format and application semantics
2. We conducted a survey of the state of the art of comparable systems, such as Stanford HIVdb, REGA, ANRS, etc. The standard format for expressing such genotypic drug-ranking rule systems is the Stanford Algorithm Specification Interface language (ASI). We gained an understanding of this language and performed an analysis of it. In the course of this work a number of deficiencies in the ASI language were identified, exemplified and ameliorated. In the course of our work, we intend to propose a more suitable language format for the expression of genotypic drug ranking rules.

3. We conceived a suitable syntax for expressing and re-encoding Retrogram rules and translated the Retrogram specification into that syntax.
4. We constructed an ASI compiler in the Prolog programming language including a parser, an interpreter and a compiler.
5. We tested the interpreter/compiler.
6. We performed a comparison with existing systems on publicly available rule sets (Stanford HIVdb, REGA).
7. We implemented the Drug Ranking System as a web service (WS).
8. We integrated the Drug Ranking System Web Service with the Virolab components for Data Access, Provenance Tracking, Resource Registry.

## 4 Drug Ranking System Virtualisation

The virtualisation of the Drug Ranking System relied on the definition of a number of formats together with a converters between those formats. The relationship between these formats and the converters is shown in Figure 4.1. They are described in more detail in the following two sections.

### 4.1 Drug Ranking System Formats

#### 4.1.1 Retrogram Specification Document

The Retrogram Drug Ranking system is described in the to be used in the “RetroGram HIV Drug Resistance Clinical Guideline (version 1.6)”.

The guideline provides suggested drug therapy rankings based on the analysis of the amino acid substitutions found in a patient’s HIV sample.

The document contains a number of tables describing how the presence of certain amino acid mutations in a patient’s HIV sample (with respect to the HIV reference genome NL4-3) influence resistance to 15 drugs (protease inhibitors, Nucleoside Reverse Transcriptase Inhibitors and Non-Nucleoside Reverse Transcriptase Inhibitors).

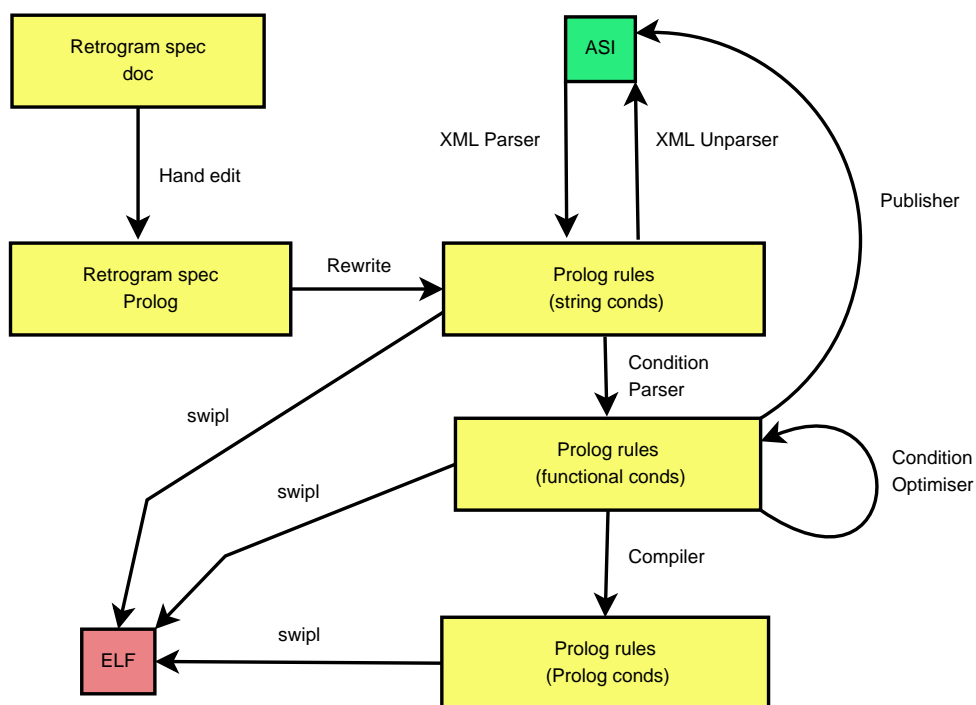


Figure 4.1: Building the Drug Ranking System

### 4.1.2 Retrogram Specification Formalised in Prolog

This is a Prolog-readable format that superficially resembles the presentation in the clinical guideline. Examples of predications are:

```
amino_acid(aspartic_acid, asp, 'D').
drug('NRTI', 'tenofovir', 'TDF').
rule(nrti, 'ZDV10', '215YF', 'ZDV', 2, 2, 3).
```

Expressing the Retrogram Specification in this way has two benefits:

1. The rules can be parsed by Prolog.
2. The rules can be seen by eye to correspond to those in the clinical guideline. This gives confidence that no errors have crept in during editing.

### 4.1.3 Virolab Prolog Rule Format

Virolab Prolog Rule Format (VPRF) is a Prolog format for describing drug resistance rules and associated information such as drug classes, resistance level interpretations, comments and rule provenance.

Common predications, such as `level_definition/5`, `drugclass/4`, `drug/5`, declare information concerning terms used in the rules. The rules themselves can be expressed using one of three levels of VPRF which differ in how rule conditions are represented. These levels, together with examples, are:

**text conditions** are strings which are parsed and interpreted during ranking.

These strings are those which are loosely defined in the ASI specification.

```
'184VI AND SELECT ATLEAST 2 FROM (41L,67N,69AD)'
```

**functional condition syntax** is an intermediate syntax which is interpreted recursively.

```
'and(184VI, interval(2, infinity, or(41L,67N,69AD)))'
```

**Prolog-encoded conditions** Functional conditions are compiled into Prolog forms for direct execution, such as:

```
relevant_mutation(MutationList):-  
    ( member(mut(184,'V'), MutationList)  
      ; member(mut(184,'I'), MutationList)  
      ),  
    ...
```

We refer to these three levels as VPRF-text, VPRF-functional and VPRF-Prolog.

The standard format for storing and representing Virolab rule sets will be VPRF-functional since it achieves the balance between expressivity and interpretability and further allows for optimisation of rule conditions.

#### 4.1.4 ELF Executable

Ultimately each of the above Prolog rule format can be compiled to an executable binary (e.g. ELF format on Linux) using a Prolog compiler. Our Prolog compiler of choice is SWI-Prolog (version 5.2.0).

## 4.2 Virtualisation components

### 4.2.1 Derivation of the Retrogram Specification in Prolog

This step was done by hand. The rules from the Retrogram Specification Document were copied from the tables in the (Microsoft Word) document and lightly

hand-edited into a Prolog-readable format that superficially resembles the presentation in the clinical guideline.

#### **4.2.2 Conversion of Prolog Retrogram Specification to Virolab Prolog Rule Format**

This is performed by a relatively straightforward Prolog program which amounts to little more than a set of rewrite rules.

#### **4.2.3 ASI XML Parser to Virolab Prolog Rule Format**

This is a Prolog program which relies on the XML parsing library of SWI-Prolog. A set of rewrite rules corresponding to the Document Type Definition (DTD) for the ASI specification language transforms the rule specifications and associated information in ASI into VPRF-text.

#### **4.2.4 Virolab Prolog Rule Unparser**

The Virolab Prolog Rule Unparser is a Prolog program which performs the opposite of the ASI XML Parser to Virolab Prolog Rule Format, namely transforming a rule set in VPRF-text format into ASI format. Although it is not a functional inverse, the resulting rule set will give the same drug rankings as the original.

#### **4.2.5 VPRF Publisher**

The VPRF Publisher is similar to the Prolog Rule Unparser but it operates on VPRF-functional format. This will be necessary for publishing rule sets which have been optimised by the Condition Optimiser.

#### **4.2.6 ASI Condition Parser**

The ASI Condition Parser is another Prolog program. It parses the conditions of the rules in VPRF-text format into conditions in VPRF-functional format.

### 4.2.7 VPRF Condition Optimiser

As mentioned above, the standard format for storing and representing Virolab rule sets will be VPRF-functional since it not only achieves the balance between expressivity and interpretability but allows for optimisation of rule conditions. Rule conditions can be analysed for internal inconsistency, parsimony, unreachable subconditions, among others. Such optimisation has not yet been carried out but will be considered in the enhancements phase of the work.

### 4.2.8 VPRF Compiler

The VPRF Compiler cross-compiles a rule set in VPRF-functional format into Prolog code which can be directly compiled by a Prolog compiler such as SWI-Prolog.

## 4.3 Future work

The Drug Ranking System is now in a state where it can be enhanced, as planned in Task 4.3. In addition to the planned enhancements such as, improving rule quality, augmenting the system with rules derived from simulations, experiments and the literature, the work done in developing the system to this point has revealed a number of ways in which it can be inherently improved. We will implement these improvements and the system will be made publicly available.

## 4.4 Abbreviations

Abbreviation/Term	Explanation
ViroLab	a virtual laboratory for decision support in viral diseases treatment
ASI	Stanford Algorithm Specification Interface language
VPRF	Virolab Prolog Rule Format
WS	Web Service
DTD	Document Type Definition
PCR	Polymerase chain reaction
swipl	The SWI-Prolog compiler/interpreter

# Bibliography

- [1] Bradley J. Betts and Robert W. Shafer. Algorithm specification interface for human immunodeficiency virus type 1 genotypic interpretation. *Journal of Clinical Microbiology*, 41(6):2792–4, 2003.
- [2] P. Clayton and G. Hripsak. Decision support in healthcare. *Int. J. of Biomedical Computation*, 39:59–66, 1995.
- [3] The Virolab Consortium. A virtual laboratory for decision support in viral diseases treatment, description of work. Technical Report 027446, Sixth Framework Programme, Priority 2, Information Society Technologies, October 2005.
- [4] M. Field and K Lohr, editors. *Clinical Practice Guidelines: Directions for a New Program*. National Academy Press, Washington D.C., 1992.
- [5] Andrea De Luca, Antonella Cingolani, Simona Di Giambenedetto, Maria Paola Trotta, Francesco Baldini, Maria Gabriella Rizzo, Ada Bertoli, Giuseppina Liuzzi, Pasquale Narciso, Rita Murri, Adriana Ammassari, Carlo Federico Perno, , and Andrea Antinori. Variable prediction of antiretroviral treatment outcome by different systems for interpreting genotypic human immunodeficiency virus type 1 drug resistance. *Journal of Infectious Diseases*, 187:1934—43, 2003.
- [6] M. Marcos, H. Roomans, A. ten, and F. van Harmelen. Improving medical protocols through formalisation: a case study, 2002.
- [7] L. Ohno-Machado, J. H. Gennari, S. Murphy, N. L. Jain, S. W. Tu, D. E. Oliver, E. Pattison-Gordon, R. A. Greenes, El H. Shortliffe, and G. O. Barnett. The GuideLine interchange format: a model for representing guidelines. *Journal of American Medical Informatics*, 5(4):357–372, 1998.
- [8] L. Sackett, William M. C. Rosenberg, J. A. Muir Gray, R. Brian Haynes, and W. Scott Richardson. Editorial from the british medical journal. *BMJ*, 312:71–2, 1996.
- [9] C. Torti, E. Quiros-Roldan, W. Keulen, L. Scudeller, C. S. Lo, and C. Boucher. Comparison between rules-based human immunodeficiency

virus type 1 genotype interpretations and real or virtual phenotype: concordance analysis and correlation with clinical outcome in heavily treated patients. *Journal of Infectious Diseases*, 188(2):194–201, 2003.

[10] C. Tural, L. Ruiz, C. Holtzer, J. Schapiro, P. Viciano, and J. Gonzalez. Clinical utility of hiv-1 genotyping and expert advice: the havana trial. *AIDS*, 16(2):209–218, 2002.

[11] S. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw. Potential benefits, limitations, and harms of clinical guidelines. *British Medical Journal*, 318:527–530, 1999.