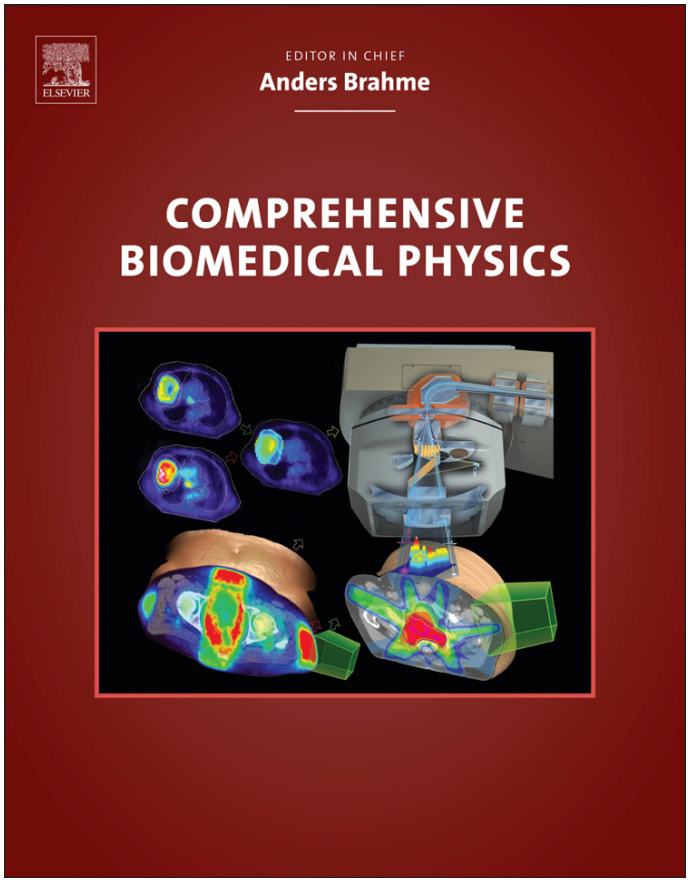




Komorowski J. (2014) Learning Rule-Based Models - The Rough Set Approach. In: Brahme A. (Editor in Chief.) Comprehensive Biomedical Physics, vol. 6, pp. 19-39. Amsterdam: Elsevier.

# 6.02 Learning Rule-Based Models – The Rough Set Approach

**J Komorowski**, Uppsala University, Uppsala, Sweden; Institute of Computer Science, Polish Academy of Sciences, Poland




| | | |
|---|---|---|
| **6.02.1** | **Introduction: Learning and Rule-Based Models** | 20 |
| **6.02.2** | **Basic Concepts of Rough Sets** | 21 |
| 6.02.2.1 | Information and Decision Systems | 21 |
| 6.02.2.2 | Approximation of Sets | 26 |
| **6.02.3** | **Quality Measures and Statistical Significance** | 27 |
| 6.02.3.1 | Accuracy of Models | 27 |
| 6.02.3.2 | Rule Quality | 27 |
| 6.02.3.3 | Statistical Significance of the Model | 27 |
| **6.02.4** | **The Modeling Process** | 28 |
| 6.02.4.1 | Rule Filtering Enhances the Model and Avoids Overfitting | 28 |
| 6.02.4.2 | Model Interpretation | 29 |
| 6.02.4.3 | Rule Tuning | 29 |
| 6.02.4.4 | Using the Models | 29 |
| **6.02.5** | **Advanced Rough Set Modeling** | 30 |
| 6.02.5.1 | Uneven Class Distribution | 30 |
| 6.02.5.2 | Feature Selection and Random Reducts | 30 |
| 6.02.5.3 | Approximate Reducts | 30 |
| 6.02.5.4 | Dynamic Reducts | 30 |
| 6.02.5.5 | Visualization | 30 |
| **6.02.6** | **Case Studies: Rough Sets in Bioinformatics** | 31 |
| 6.02.6.1 | Protein Analysis – From Sequences to Functions to Interactions | 31 |
| 6.02.6.2 | HIV-1 Modeling | 31 |
| 6.02.6.3 | Protein–Ligand Interactions | 32 |
| 6.02.6.4 | Function Prediction from Structure | 33 |
| 6.02.6.5 | Rough Sets in Genomics and Transcriptomics | 33 |
| 6.02.6.6 | Functional Genomics | 33 |
| 6.02.6.7 | Rough Sets in Cancer Research | 34 |
| 6.02.6.8 | Combinatorial Gene Regulation | 35 |
| 6.02.6.9 | Epigenetics – The Histone Code | 35 |
| 6.02.6.10 | Feature Interaction: Gene–Gene and Gene–Environment Interactions in Allergy | 36 |
| **6.02.7** | **Rough Sets Versus Statistical Classification** | 36 |
| **6.02.8** | **Other Learning Approaches in Bioinformatics** | 37 |
| 6.02.8.1 | Rough Set Resources | 37 |
| 6.02.8.2 | Software Availability | 37 |
| **Acknowledgments** | | 37 |
| **References** | | 38 |


## Glossary

**Accuracy** ACC = (TP+TN)/(TP+FP+FN+TN) is a measure of quality of a classifier.

**AUC** The area under ROC curve.

**Classifier** A classifier over a decision system is a function that outputs a predicted decision for an object in the decision system.

**Confusion matrix** A confusion matrix is a square matrix that compares predicted decisions (the values of the classifier) with the actual ones.

**Decision system** Decision systems (or decision tables) contain objects in rows and their features (or attributes, or conditions) in columns. Decision systems associate the values of features for each object with a decision (or outcome) for the object.

**FP, FN, TP, TN** Assume one class is positive and the other negative; False Positive (FP): the negative cases classified to the positive class, $ False Negative (FN): the positive cases classified to the negative class, $ True Positive (TP): the positive cases classified to the positive class, $ True Negative (TN): the negative cases classified to the negative class.

**MCFS** Monte Carlo Feature Selection: a method and software (dmlab) implemented in Java that builds ranked lists of significant features used for classification.

**Prime implicant** P is an implicant of Q if Q is true whenever P is true. Prime implicant is an implicant that is minimal.

**Reduct** Given a decision system, a reduct is a minimal set of features that gives the same discernibility between objects as



*Comprehensive Biomedical Physics,* (2014), vol. 6, pp. 19-39

the original set of features in the decision system. It has been proven that reduct correspond to prime implicants.
**ROC** Receiver operating characteristic, or ROC curve, is a graphical representation of the performance of a binary classifier under varying discrimination. The graph is drawn as the sensitivity against (1 - specificity) under varying threshold settings.
**ROSETTA** A system for developing rough set models under Windows and on multicore computers.
**Sensitivity** Also known as True Positive Rate: TPR = TP/(TP+FN).
**Specificity** Also known as True Negative Rate: SPC = TN/(FP+TN).

## 6.02.1 Introduction: Learning and Rule-Based Models

Machine learning has a long record of successful applications in many fields including the life sciences. This is witnessed, among others, by the influential book *Bioinformatics: The Machine Learning Approach* (Baldi and Brunak, 1998). In general, the dominating aim in the field of machine learning has been the construction of the best possible classifier for the task at hand. However, over the years of applying machine learning in bioinformatics, we have learnt that scientists working in many areas of the life sciences call for a deeper knowledge of the modeled phenomenon than just the information to classify the objects with a certain quality.

Many investigations today collate data into decision tables. For instance, a collection of octamer peptides is subjected to cleavage by a protease and the effect of the enzymatic activity is measured. The objects in the table are sequences of eight amino acids. A sequence may be represented by the names of the amino acids, or by the features of the amino acids in the given positions, and the decision or outcome associated with the object is the result of applying an enzyme to the object: cleaved or not cleaved. Being able to predict whether a sequence is cleavable or not is of course of high value. However, learning which features of objects representing the phenomenon, and which values of the features and in which combinations, define the phenomena under study provides a significant added value. In another experiment, for a collection of cells that are first halted and then stimulated by an addition of the medium, gene expression levels may be measured at several time points. The decision, or outcome, can here be the participation of a gene in a biological process. In addition to being able to predict this participation automatically for new genes, the molecular biologist model may be interested in learning minimal sets of features that define the membership of a gene in a particular biological process. The possibility of learning which features, in which order, and in which combinations define decisions becomes even more acute for high-throughput problems. For instance, we could be interested in learning which histone modification combinations might be associated with exon inclusion levels.

Hence, the standard definition of machine learning stating that given a set of examples with known decisions, classification decisions can be made automatically for future, unseen examples needs to be amended to include learning the structure of the classifier. Such tasks lend themselves well to rule-based modeling. The rules used here to construct our models are in the IF–THEN form:

IF Pos_2(AA) = not aromatic AND Pos_1′(AA) = has a large hydrophobic residue THEN the octamer is not cleavable
IF Gene A is up-regulated AND Gene D is down-regulated THEN the tissue is healthy
IF Transcription factor F binds AND Transcription factor V binds THEN Gene is co-regulated with Gene H
IF Protein contains motif J THEN Function is magnesium ion binding OR copper ion binding
IF Protein contains motif D AND Ligand water-octanol coeff. > c1 THEN binding affinity is high
IF Change in frequency of alpha-helix at position X > c3 THEN Resistant to drug W

Rules have many advantages. They are easily legible and users not trained in the theoretical aspects of that formalism will readily be able to understand them. Rules may also be applied to making predictions and, in certain cases, to generate new objects of a desired outcome that were not present in the learning set. Interactions in the data may also be explicitly modeled by rules.

The rough set formalism is particularly well suited to handling noise and ambiguity in the data by constructing approximate rules that have multiple outcomes. Furthermore, since neither the features nor the coordinate system need be modified in the process of constructing the rules, the original interpretation of the features is preserved. As a result, the user of a rough set model may inspect its structure, that is, learn which *minimal* sets of features are used to construct the model, what alternative models exist, and how the features may interact.

In this chapter, it is shown that rule-based models founded in rough sets answer many needs of complex modeling in bioinformatics. We first define the basic concepts of rough sets illustrating them with a running, albeit simplistic, example of modeling cleavability of octamers by a protease and then continue with a description of the methodology for developing rule-based models. There is a publicly available system for modeling with rough sets, called ROSETTA (Komorowski et al., 2002; Øhrn and Komorowski, 1997). The software can be downloaded from (http://rosetta.lcb.uu.se) together with a collection of data sets.

The structure of this presentation is as follows. After the introduction of the basic concepts in rough sets, the modeling process is described, with the focus on the statistical significance of the models. It is followed by a brief presentation of approaches to very large problems and some recently researched aspects of feature selection and rule visualization. A discussion on selected applications of rough sets to a wide spectrum of problems in virology, proteomics, structural biology, functional genomics, transcriptomics, and epigenetics is presented.

The presentation ends with a discussion of the relationship between rough sets and statistical classification, and a brief comparison of rough sets to other machine learning approaches.

## 6.02.2 Basic Concepts of Rough Sets

### 6.02.2.1 Information and Decision Systems

*Rough set theory* is due to Pawlak (1992). It is based on Boolean reasoning (Brown, 2003), which constitutes the mathematical framework for inducing rules from examples. The data is represented in the tabular form of an *information system*, which is a collection of objects (rows of the table) with features represented by the columns of the table. Features are functions that assign values to the objects. One row of the table is called an *information vector*. For more precision, an information system is defined as a pair

$$\mathcal{A} = (U, A)$$

where $U$ is a non-empty finite set called the universe and $A$ is a non-empty set of features (also called attributes) such that $a: U \to V_a$ for every $a \in A$. The set $V_a$ is called the value set of $a$.

Table 1 presents a sample information system. The objects in this table are octamer peptides and the features are positions in the octamers taking values from the set of amino acids. This is a simplified and small subset of the data used in Kontijevskis et al. (2007a). The task presented there was to model a property of the sequences such as cleavability of octamers by a protease in HIV-1. More precisely, the aim was to learn which positions need which values to make an octamer cleavable. Notice that this aim differs from the classical goal of learning machinery, which is obtaining the best possible classifier.

Our working example will be using discrete data. However, rough sets can also be used on continuous data after it is discretized.

Rough set modeling is built around the concept of *reduction*. First, objects that are indiscernible using the available features fall into *equivalence classes*. For example, since objects 1, 8, and 10 have exactly the same information vectors, they are indiscernible, and so are objects 2 and 9. They form equivalence classes (see Table 2).

Hence, instead of using multiple indiscernible objects, a much more parsimonious representation of their respective equivalence classes is chosen. We may either explicitly write all the members of the equivalence class $\{1, 8, 10\} = [1]_A$, where $A$ refers to the set of features, or use a more concise representation of the right-hand side of the equality. Notice that $[1]_A = [8]_A = [10]_A$.

Unless stated otherwise, hereafter we assume that one row corresponds to an equivalence class (cf. Table 3).

The second type of reduction is central to rough sets and concerns reduction of features that are redundant. The motivation for removing such superfluous features is especially evident in modeling tasks that collect a huge number of parameters such as next-generation sequencing or modeling protein functions from sequences. Even in applications that use a more modest number of features, it may be useful to learn which features are redundant. Working with a reduced set helps the researcher to focus on the most important factors and follows the centuries-old principle of Occam's razor.

The first observation about selecting subsets of features relates to the partitioning into equivalence classes induced by the choice of the features. Given an information system, we create a new information system by taking only a subset of the original set of features. For instance, we can select {P4, P3, P2, P1} and create a new information system. Selecting another subset of features usually leads to a different partitioning of the universe (cf. Table 4).

This concept of reduction is defined with the help of the *indiscernibility* relation. Given an information system $\mathrm{A} = (U, A)$ and a subset of features $B \subseteq A$, we define

$$\mathrm{IND}_{\mathcal{A}}(B) = \left\{ \left(x, x'\right) \in U^2 | \forall \mathrm{a} \in Ba(\mathrm{x}) = \mathrm{a}\left(\mathrm{x}'\right) \right\}$$

$\mathrm{IND}_{\mathrm{A}}(B)$ is called the $B$-indiscernibility relation, where $B$ refers to the subset of features. If $(x, x') \in \mathrm{IND}_{\mathrm{A}}(B)$, then objects $x$ and $x'$ are *indiscernible*. For simplicity, when it is clear which information system we mean, we omit the index in the name of the indiscernibility relation $\mathrm{IND}(B)$. In Table 4, objects {1, 6, 8, 10}, {2, 5, 9} {3}, {4}, and {7} form equivalence classes when using $B = \{\mathrm{P4}, \mathrm{P3}, \mathrm{P2}, \mathrm{P1}\}$. If we select {P1′, P2′, P3′, P4′} for the subset of features, the equivalence classes are instead {1, 8, 10}, {2, 9}, {3}, {4}, {5}, {6}, and {7}.

A very careful reader may have by now noticed that using only features P1′ and P2′, it is possible to discern between all the objects in the very same way as using all the available features. This can be expressed as the equality of the indiscernibility relation:

**Table 1** An example of an information system

| *Obj. id* | *P4* | *P3* | *P2* | *P1* | *P1′* | *P2′* | *P3′* | *P4′* |
|---|---|---|---|---|---|---|---|---|
| 1 | P | I | A | A | E | A | G | M |
| 2 | A | I | V | A | A | A | G | T |
| 3 | T | P | L | A | R | S | I | L |
| 4 | T | I | A | A | R | A | G | M |
| 5 | A | I | V | A | R | E | G | T |
| 6 | P | I | A | A | E | I | T | A |
| 7 | T | I | V | K | A | L | N | D |
| 8 | P | I | A | A | E | A | G | M |
| 9 | A | I | V | A | A | A | G | T |
| 10 | P | I | A | A | E | A | G | M |

**Table 2** The information system of Table 1 with equivalence classes marked with colors

| *Obj. id* | *P4* | *P3* | *P2* | *P1* | *P1′* | *P2′* | *P3′* | *P4′* |
|---|---|---|---|---|---|---|---|---|
| 1 | P | I | A | A | E | A | G | M |
| 8 | P | I | A | A | E | A | G | M |
| 10 | P | I | A | A | E | A | G | M |
| 2 | A | I | V | A | A | A | G | T |
| 9 | A | I | V | A | A | A | G | T |
| 3 | T | P | L | A | R | S | I | L |
| 4 | T | I | A | A | R | A | G | M |
| 5 | A | I | V | A | R | E | G | T |
| 6 | P | I | A | A | E | I | T | A |
| 7 | T | I | V | K | A | L | N | D |

The equivalence classes are {1, 8, 10}, {2, 9} and {3}, {4}, {5}, {6}, and {7}.

**Table 3** The information system of Table 1 with objects replaced by their corresponding equivalence classes

| *Equiv. cl.* | *P4* | *P3* | *P2* | *P1* | *P1′* | *P2′* | *P3′* | *P4′* |
|---|---|---|---|---|---|---|---|---|
| {1, 8, 10} | P | I | A | A | E | A | G | M |
| {2, 9} | A | I | V | A | A | A | G | T |
| {3} | T | P | L | A | R | S | I | L |
| {4} | T | I | A | A | R | A | G | M |
| {5} | A | I | V | A | R | E | G | T |
| {6} | P | I | A | A | E | I | T | A |
| {7} | T | I | V | K | A | L | N | D |

**Table 4** Two reduced information systems with different partitioning

| *Equiv cl* | *P4* | *P3* | *P2* | *P1* |
|---|---|---|---|---|
| {1, 6, 8, 10} | P | I | A | A |
| {2, 5, 9} | A | I | V | A |
| {3} | T | P | L | A |
| {4} | T | I | A | A |
| {7} | T | I | V | K |
| *Equiv. cl.* | *P1′* | *P2′* | *P3′* | *P4′* |
| {1, 8, 10} | E | A | G | M |
| {2, 9} | A | A | G | T |
| {3} | R | S | I | L |
| {4} | R | A | G | M |
| {5} | R | E | G | T |
| {6} | E | I | T | A |
| {7} | A | L | N | D |

$$\begin{aligned}\mathrm{IND}(A) &= \{\{1,8,10\},\{2,9\},\{3\},\{4\},\{5\},\{6\},\{7\}\}\\ &= \mathrm{IND}\left(\left\{\mathrm{P1}',\mathrm{P2}'\right\}\right)\end{aligned}$$

Indeed, a comparison of the information system of Table 3 with the information system of Table 5 shows that the equivalence classes are exactly the same for both systems. More interestingly, there may exist several such minimal sets of features.

The process of finding this reduced representation is easily formalized using Boolean reasoning. To this end, we build a square matrix $n \times n$, where $n$ is the number of equivalence classes. In every entry $c_{ij}$, where $i, j \in [1, n]$, we put a disjunction of the names of the features that discern between equivalence class $i$ and equivalence class $j$. This matrix is called a *discernibility matrix*. It is symmetrical along the diagonal, which contains the empty sets (Table 6).

For example, objects from class $\{2, 9\}$ may be distinguished from objects in class $\{5\}$ by feature $\mathrm{P1}'$ or feature $\mathrm{P2}'$. By taking the conjunction of all the non-empty entries, we obtain a Boolean expression that allows us to discern between any two equivalence classes in the system.

**Table 5** Two minimally reduced information systems

| *Equiv. cl.* | *P1′* | *P2′* |
|---|---|---|
| {1, 8, 10} | E | A |
| {2, 9} | A | A |
| {3} | R | S |
| {4} | R | A |
| {5} | R | E |
| {6} | E | I |
| {7} | A | L |
| *Equiv. cl.* | *P4* | *P4′* |
| {1, 8, 10} | P | M |
| {2, 9} | A | T |
| {3} | T | L |
| {4} | T | M |
| {5} | A | T |
| {6} | P | A |
| {7} | T | D |

$$\begin{gathered}(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P4}')\wedge(\mathrm{P4}\vee\mathrm{P3}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P4}\vee\mathrm{P3}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge(\mathrm{P4}\vee\mathrm{P1}')\wedge(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P3}\vee\mathrm{P2}\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P4}')\wedge(\mathrm{P1}'\vee\mathrm{P2}')\wedge\\ (\mathrm{P4}\vee\mathrm{P3}\vee\mathrm{P2}\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P2}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P4}\vee\mathrm{P3}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P4}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P1}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P4}\vee\mathrm{P1}\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge(\mathrm{P3}\vee\mathrm{P2}\vee\mathrm{P1}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P2}\vee\mathrm{P1}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge\\ (\mathrm{P4}\vee\mathrm{P1}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\wedge(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P2}\vee\mathrm{P1}\vee\mathrm{P1}'\vee\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}')\end{gathered}$$

This expression can be simplified by using logical laws such as $(a \vee b) \wedge b \equiv b$. The process is illustrated with the following coloring scheme. A minimal expression with respect to its length is chosen. First, the expression $\mathrm{P1}' \vee \mathrm{P2}'$ is selected and all its supersets are removed. Second, the expression $\mathrm{P2}' \vee \mathrm{P3}' \vee \mathrm{P4}'$ is chosen and its supersets are removed. Third, we take the expression $\mathrm{P4} \vee \mathrm{P1}'$ and remove its supersets. The final removal step is not colored. The minimal expression is then written down. Equal expressions are obviously written only once.

$$\left(\mathrm{P4}\vee\mathrm{P1}'\right)\wedge\left(\mathrm{P2}'\vee\mathrm{P3}'\vee\mathrm{P4}'\right)\wedge\left(\mathrm{P1}'\vee\mathrm{P2}'\right)\wedge\left(\mathrm{P4}\vee\mathrm{P2}\vee\mathrm{P2}'\vee\mathrm{P4}'\right)$$

After the simplification, we convert the expression to an equivalent disjunction of conjunctions:

$$\left(\mathrm{P1}'\wedge\mathrm{P2}'\right)\vee\left(\mathrm{P4}\wedge\mathrm{P2}'\right)\vee\left(\mathrm{P1}'\wedge\mathrm{P4}'\right)\vee\left(\mathrm{P4}\wedge\mathrm{P1}'\wedge\mathrm{P3}'\right)\vee\left(\mathrm{P2}\wedge\mathrm{P1}'\wedge\mathrm{P3}'\right)$$

In logic, such expressions are called prime implicants of the original expression because they logically imply the original expression, and no term can be removed without losing this implication. In the rough set terminology, these constructs are called *reducts*. Each reduct has the property of being a minimal expression that discerns between any pair of equivalence classes, and without any loss of discernibility, it can be used in lieu of all

**Table 6** Discernibility matrix for the information system in **Table 3**

| | *{1, 8, 10}* | *{2, 9}* | *{3}* | *{4}* | *{5}* | *{6}* | *{7}* |
|---|---|---|---|---|---|---|---|
| {1, 8, 10} | ∅ | | | | | | |
| {2,9} | P4 ∨ P2 ∨ P1′ ∨ P4′ | ∅ | | | | | |
| {3} | P4 ∨ P3 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P3 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | | | | |
| {4} | P4 ∨ P1′ | P4 ∨ P2 ∨ P1′ ∨ P4′ | P3 ∨ P2 ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | | | |
| {5} | P4 ∨ P2 ∨ P1′ ∨ P2′ ∨ P4′ | P1′ ∨ P2′ | P4 ∨ P3 ∨ P2 ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P2 ∨ P2′ ∨ P4′ | ∅ | | |
| {6} | P2′ ∨ P3′ ∨ P4′ | P4 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P3 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | |
| {7} | P4 ∨ P2 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P1 ∨ P2′ ∨ P3′ ∨ P4′ | P3 ∨ P2 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P2 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P2 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | ∅ |

features. (Strictly speaking, we should use another name for the logical variable, e.g., $P^*$ instead of $P$. However, this abuse of notation does not create any confusion.) Reducts have several interesting properties. For instance, we can overlay a reduct on the objects in the original information system to form an association rule set: Given the values of features in the reduct for a given object, it can be used to uniquely determine the values of all other features for the object. This relation is known in relational databases and is referred to as functional dependency. Although the case of reducts in information systems is interesting, it is not explored further here. Instead, a special case of information systems called *decision systems* is investigated.

It is often the case that objects have an outcome, such as binding affinity, or a decision such as benign or malicious, or an annotation, such as participation in a biological process. The decision may be the result of an experiment or it may be assigned from a database or another source such as human diagnosis. For the time being, let us assume that decisions are discrete (categorical). The *decision feature* is unique for the information system and is put in the last column. A *decision system* is any information system of the form

$$\mathcal{A} = (U, A \cup \{d\})$$

where $d \notin A$ is the decision (or outcome) feature. The elements of $A$ are often called *conditions*. The decision may take any finite number of values, but in practical applications, it is usually limited to a few, with binary and tertiary decisions being the most frequently used. A sample decision system is given in Table 7.

Several properties of decision systems are interesting. Only some of them are explored here. First, we see that the decision divides objects into *decision classes*. In our simplified case, there are two decision classes: cleavable and non-cleavable, customarily represented as 1 and 0, respectively. However, any finite number of decision classes is allowed, although large numbers will not be practical. Since these are names only, any other naming convention would do; for instance, we could use YES and NO instead of 1 and 0, respectively.

**Table 7** A sample decision system

| *Obj. id* | *P4* | *P3* | *P2* | *P1* | *P1′* | *P2′* | *P3′* | *P4′* | *Cleave* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | P | I | A | A | E | A | G | M | 1 |
| 8 | P | I | A | A | E | A | G | M | 1 |
| 10 | P | I | A | A | E | A | G | M | 0 |
| 2 | A | I | V | A | A | A | G | T | 1 |
| 9 | A | I | V | A | A | A | G | T | 1 |
| 3 | T | P | L | A | R | S | I | L | 0 |
| 4 | T | I | A | A | R | A | G | M | 1 |
| 5 | A | I | V | A | R | E | G | T | 1 |
| 6 | P | I | A | A | E | I | T | A | 0 |
| 7 | T | I | V | K | A | L | N | D | 0 |

The examples of objects with their annotation to the classes are the starting point for supervised learning. They are the definitions of the concepts represented by the decision feature and the task of a learning algorithm is to create as generalized a definition as possible for these concepts in the language of features and their values.

In our case, the union of objects $\{1, 8, 2, 9, 4, 5\}$ defines the cleavable class of octamers, while objects $\{10, 3, 6, 7\}$ define the non-cleavable class. However, there is a caveat with this construction. There are objects in the same equivalence class that have different class membership, but they are indiscernible! In Table 7, objects $\{1, 8, 10\}$ form an equivalence class, but they have different decision labels. Such cases are typical of real data. They may be due to noise in the data, wrong measurements, or simply lack of knowledge. For instance, two patients may have apparently the same, that is, indiscernible, cancers, but in fact have different etiologies not illuminated by the collected features. Hence, examples serving as definitions may be contradictory (also called non-deterministic) and the question is how to treat such definitions. Rough sets offer an interesting approach to this real-life scenario.

As done previously, we shall search for minimal sets of features that discern between the equivalence classes, with one modification: equivalence classes having the same decision need not be discerned. As there are classes that do not have unique outcomes, we first transform a non-deterministic decision system into a deterministic one by replacing the decision with the set of all decisions for the given equivalence class. The decision system of Table 7 is now converted into a so-called generalized decision system and is shown in Table 8.

The construction of the discernibility matrix resembles the previous construction except that we now put the empty set of features for pairs of equivalence classes that have the same decision value (Table 9).

The simplification proceeds in the same way with the deeper shades of the colors indicating minimal expressions and the lighter colors their super-expressions. Simplification produces the following expression:

$$\left(\mathrm{P4} \vee \mathrm{P1}'\right) \wedge \left(\mathrm{P2}' \vee \mathrm{P3}' \vee \mathrm{P4}'\right)$$

which after conversion to the disjunctive form gives six reducts:

**Table 8** A transformed decision system with equivalence classes and generalized decision

| *Equiv. cl.* | *P4* | *P3* | *P2* | *P1* | *P1′* | *P2′* | *P3′* | *P4′* | *Cleave* |
|---|---|---|---|---|---|---|---|---|---|
| {1, 8, 10} | P | I | A | A | E | A | G | M | {0, 1} |
| {2, 9} | A | I | V | A | A | A | G | T | {1} |
| {3} | T | P | L | A | R | S | I | L | {0} |
| {4} | T | I | A | A | R | A | G | M | {1} |
| {5} | A | I | V | A | R | E | G | T | {1} |
| {6} | P | I | A | A | E | I | T | A | {0} |
| {7} | T | I | V | K | A | L | N | D | {0} |

**Table 9** The discernibility matrix for the decision system in **Table 8**

| | *{1, 8, 10}* | *{2, 9}* | *{3}* | *{4}* | *{5}* | *{6}* | *{7}* |
|---|---|---|---|---|---|---|---|
| {1, 8, 10} | ∅ | | | | | | |
| {2, 9} | P4 ∨ P2 ∨ P1′ ∨ P4′ | ∅ | | | | | |
| {3} | P4 ∨ P3 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P3 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | | | | |
| {4} | P4 ∨ P1′ | ∅ | P3 ∨ P2 ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | | | |
| {5} | P4 ∨ P2 ∨ P1′ ∨ P2′ ∨ P4′ | ∅ | P4 ∨ P3 ∨ P2 ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | ∅ | | |
| {6} | P2′ ∨ P3′ ∨ P4′ | P4 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | P4 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P2 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | |
| {7} | P4 ∨ P2 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P1 ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | P2 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | P4 ∨ P1 ∨ P1′ ∨ P2′ ∨ P3′ ∨ P4′ | ∅ | ∅ |

**Table 10** Rule measures

| *Rule* | *LHS Sup.* | *RHS Sup.* | *RHS Accuracy* | *LHS Coverage* | *RHS Coverage* |
|---|---|---|---|---|---|
| P4(P) AND P2′(A) => Cleave(0) OR Cleave(1) | 3 | 1, 2 | 0.333, 0.667 | 0.3 | 0.25, 0.333 |
| P4(A) AND P2′(A) => Cleave(1) | 2 | 2 | 1 | 0.2 | 0.333 |
| P4(T) AND P2′(S) => Cleave(0) | 1 | 1 | 1 | 0.1 | 0.25 |
| P4(T) AND P2′(A) => Cleave(1) | 1 | 1 | 1 | 0.1 | 0.1667 |
| P4(A) AND P2′(E) => Cleave(1) | 1 | 1 | 1 | 0.1 | 0.1667 |
| P4(P) AND P2′(I) => Cleave(0) | 1 | 1 | 1 | 0.1 | 0.25 |
| P4(T) AND P2′(L) => Cleave(0) | 1 | 1 | 1 | 0.1 | 0.25 |

$$\left(\mathrm{P4}\times\mathrm{P2}^{'}\right)\mathrm{v}\left(\mathrm{P4}\times\mathrm{P3}^{'}\right)\mathrm{v}\left(\mathrm{P4}\times\mathrm{P4}^{'}\right)\mathrm{v}\left(\mathrm{P1}^{'}\times\mathrm{P2}^{'}\right)\mathrm{v}\left(\mathrm{P1}^{'}\times\mathrm{P3}^{'}\right)\mathrm{v}\left(\mathrm{P1}^{'}\times\mathrm{P4}^{'}\right)$$

Overlaying the reducts on the objects of Table 8 produces IF–THEN rules. For example, the first reduct gives rise to the following rules:

IF P4 = P AND P2′ = A THEN Cleave = 1 OR 0
IF P4 = A AND P2′ = A THEN Cleave = 1
IF P4 = T AND P2′ = S THEN Cleave = 0
IF P4 = T AND P2′ = A THEN Cleave = 1
IF P4 = A AND P2′ = E THEN Cleave = 1
IF P4 = P AND P2′ = I THEN Cleave = 0
IF P4 = T AND P2′ = L THEN Cleave = 0

To quantify the generated rules, several numerical parameters are defined. *LHS Support* is the number of objects that match the left-hand side of the rule; right-hand support – *RHS Support* – is the number of the LHS objects of the respective classes; *RHS Accuracy* is defined for each class separately and is the proportion of objects of its RHS Support to LHS Support of the rule; *LHS Coverage* is the fraction of objects in LHS Support to the number of objects in the universe; and *RHS Coverage* is the proportion of RHS Support objects in each class to the number of objects of the class in the universe. Table 10 gives the values of the parameters for the rules generated from reduct P4^P2′.

### 6.02.2.2 Approximation of Sets

An equivalence relation, such as indiscernibility, induces a partitioning of the universe, in our case on the set of octamers. Such partitions may be used to construct new subsets of the universe. Of foremost interest are subsets that have the same decision or outcome. In the example in Table 8, the sets {{2, 9}, {4}, {5}} and {{3}, {6}, {7}} could be used to define the Cleave concept and its complement, respectively. This definition would, however, exclude objects {1, 8, 10}, whose cleavability status is uncertain. It may happen, and it, indeed, usually does happen in practical applications, that a concept cannot be defined crisply. We know which octamers are certainly cleavable and which are certainly not cleavable, and which octamers belong to the boundary region in between them. If the boundary region happens to be the empty set, the set is crisp, and if the region is not empty, the set is rough. Formally, the definition is as follows:

Let $A=(U, A)$ be an information system and let $B\subseteq A$ and $X\subseteq U$. We approximate $X$ using only features of $B$ by constructing the $B$-lower and $B$-upper approximations of $X$, respectively, denoted $\underline{B}(X)$ and $\bar{B}(X)$, where

$$\underline{B}(X)=\left\{x\middle|[x]_B\subseteq X\right\}$$

$$\bar{B}(X)=\left\{x\middle|[x]_B\cap X\neq\varnothing\right\}$$

The intention of these definitions is to identify the set $X$ with the decision and hence, we move our attention to decision systems of the form $A=(U, A\cup\{d\})$, where the value of feature $d$ defines whether the object belongs to the set or not. Informally, objects in the $B$-lower approximation are all uniquely classified as members of $X$, while objects in the $B$-upper approximation can possibly belong to $X$. The set $\mathrm{BN}_B(X)=\bar{B}(X)-\underline{B}(X)$ is called the $B$-boundary region of $X$. It contains all objects whose equivalence classes have multiple decision values and thus cannot be classified as members of $X$ with certitude. The set $U-\bar{B}X$ is called the $B$-outside region of $X$ and consists of objects that can be classified as certainly not members of $X$. It is important to bear in mind that we are using only a subset of all features. Using another subset of features, different approximations may be obtained.

A set is defined as *rough* (respectively, *crisp*) if its boundary region is non-empty (respectively, empty). When using all the features of a decision system, we usually omit the index in the approximations and boundary regions.

For example, let the set of features $B=A$ and consider Cleave = {1, 2, 4, 5, 8, 9}. We obtain $\underline{B}$(Cleave) = {2, 4, 5, 9} and $\bar{B}(\text{Cleave})=\{2,4,5,9\}\cup\{1,8,10\}=\{1,2,4,8,9,10\}$. Since the boundary region $\mathrm{BN}_B(X)=\{1, 8, 10\}$ is not empty, Cleave is rough.

There are several algebraic properties of approximations. The more important ones are mentioned here.

(1) $\underline{B}(X)\subseteq X\subseteq\bar{B}(X)$

(2) $\underline{B}(\varnothing)=\bar{B}(\varnothing)=\varnothing,\ \underline{B}(U)=\bar{B}(U)=U$

(3) $\underline{B}(X\cup Y)\supseteq\underline{B}(X)\cup\underline{B}(Y)$
(4) $\bar{B}(X\cap Y)\subseteq\bar{B}(X)\cap\bar{B}(Y)$

(5) $\underline{B}(-X)=-\bar{B}(X)$

The degree of roughness is based on frequency and is defined by the following formula:

$$\alpha_B(X)=|\underline{B}(X)|/|\overline{B}(X)|$$

where $|.|$ is the cardinality of any $X\neq\varnothing$. Naturally, $0\leq\alpha_B(X)\leq 1$, and if $\alpha_B(X)=1$, $X$ is *crisp* with respect to $B$ and if $\alpha_B(X)<1$, $X$ is *rough* with respect to $B$.

## 6.02.3 Quality Measures and Statistical Significance

### 6.02.3.1 Accuracy of Models

The learning process relies on the learning set and the test set. It is assumed that the learning set is representative of the test set. The reducts are computed using the learning set and all the generated rules may be used to classify the objects in the test set. Performance of the classifier can be measured with accuracy, which is the fraction of the correctly classified test cases. Accuracy may be insufficient when the cardinalities of the two classes are skewed or when making one type of error is more costly than the other one.

Let us assume one class is positive and the other negative. We consider predicted and actual classification and define the following quantities:

- *False positive* (FP): the negative cases classified to the positive class,
- *False negative* (FN): the positive cases classified to the negative class,
- *True positive* (TP): the positive cases classified to the positive class,
- *True negative* (TN): the negative cases classified to the negative class.

These values are customarily summarized in a confusion matrix (Table 11).

We can also define *sensitivity* and *specificity* as follows:

- sensitivity = TP/(TP + FN)
- specificity = TN/(TN + FP)

### 6.02.3.2 Rule Quality

In the classification process, we need to decide which decision class is to be chosen. This task is often effectuated by majority voting. For our running example, the six reducts for seven equivalence classes generate at most 42 rules, but since some of them are duplicates, the model contains 39 rules in total. Let one object from the test set be described by the following information vector and let it be cleavable.

| *Obj. id.* | *P4* | *P3* | *P2* | *P1* | *P1′* | *P2′* | *P3′* | *P4′* | *Cleave* |
|---|---|---|---|---|---|---|---|---|---|
| *x* | T | – | – | – | E | S | G | A | {1} |

**Table 11** The schema for a confusion matrix

| | | *Predicted* | |
|---|---|---|---|
| | | *Negatives* | *Positives* |
| Actual | Negatives | TN | FP |
| | Positives | FN | TP |

**Table 12** Classification result

| | | *Classes voted* | *Support per class* | *Fraction of votes per class* |
|---|---|---|---|---|
| P4 = T | P2′ = S | {0} | (1, 0) | |
| P1′ = E | P3′ = G | {0, 1} | (1, 2) | |
| P1′ = E | P4′ = A | {0} | (1, 0) | |
| | | | (3, 2) | [0.6; 0.4] |

We intentionally omit values for P3–P1 because they never appear in the reducts and therefore the values of the positions are not important to the classification. This suggests that these positions do not participate in defining cleavability. Application of the model to the object results in classifications as in Table 12.

Two rules gave two 'votes' for decision 1 (cleavable) and three for 0 (non-cleavable). If we accept majority voting, that is, request the threshold of 0.5, then the classification is 0 ($0.6 > 0.5$). Demanding a higher specificity, for example, 0.7 (the largest fraction of the votes is below the threshold: $0.6 < 0.7$), results in classification to the negative class. The confusion matrix may be extended to include the number of undefined objects. Using various values for the threshold allows obtaining models trained for higher specificity or sensitivity, whichever is preferred. Thresholds naturally lead to ROC curves. A ROC curve illustrates the performance of a (binary) classifier and is created by drawing *sensitivity* values versus 1-*specificity* values. ROC curves allow selecting models that are possibly optimal independently of the class distribution and of the cost associated with making a decision. Usually, the point closest to the North-West corner of the diagram, that is, point (0, 1), is selected as the possibly optimal choice (Figure 1). The area under the curve (AUC) is another measure of the quality of a classifier. It corresponds to the probability that the classifier will rank a random element of the positive class higher that a random element of the negative one.

### 6.02.3.3 Statistical Significance of the Model

Given a decision system and a resulting model, it is important to asses the statistical significance of the quality of the model. This significance is expressed as a *p*-value and is calculated as the fraction of permutation values that are at least as good as the original value derived from non-permuted data. In permutation tests, we define the null hypothesis thus: The labels

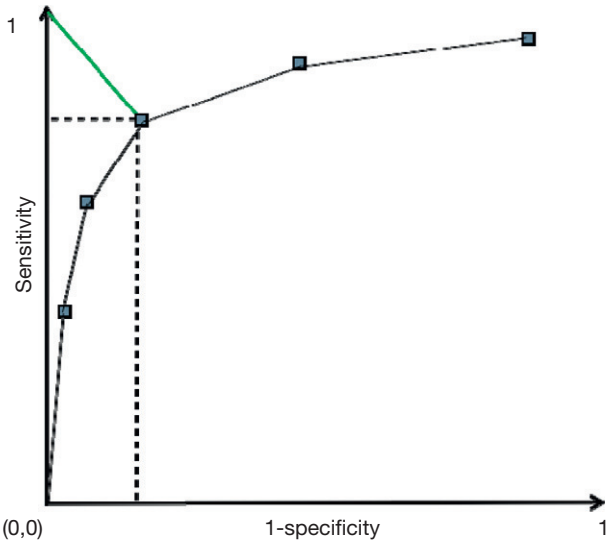


**Figure 1** A sample ROC curve.

assigning samples to classes are interchangeable (Edgington, 1969). To this end, decision values are randomly rearranged to form a new decision system, a model is constructed, and its accuracy computed. The procedure is repeated, for example, 100 times and the *p*-value is defined as the number of times when accuracy of the model was equal to or higher than the original accuracy divided by 100. Permutation tests are widely used non-parametric tests in bioinformatics (e.g., Subramanian et al., 2005; Tusher et al., 2001), since usually there is no evidence, or sufficient data, to assume a particular distribution model.

### 6.02.4 The Modeling Process

Developing a model is a complex process that requires a good understanding of the problem and the data describing the problem. It is also dependent on the experience of the user. We now highlight the main issues in this process, but it needs to be stressed that modeling is quite similar to programming: Quality comes with training and experience. As of today, there are no push-button software tools that would produce good models from the available data. We first describe the main steps of developing models and later review our published work in order to illustrate some practical problems that need to be resolved by the user. These models were developed with the ROSETTA system, but our observations are mostly of a general nature.

The process starts from a decision system. For simplicity, we assume that there are no missing data, that is, all values for all features are present in the decision system. If the system is sufficiently large, it should be divided into the learning and testing set. Usually, the split is 2:1, that is 67% of the data is used for the learning set and the remaining 33% for the test set. The learning set needs to be representative of the test set. As this cannot be guaranteed, the best approach is to perform a random split. This split must be done prior to any processing of the data so that no information about the test set is used in processing the learning set. If there is no testing set, the usual approach is to apply a cross-validation schema to estimate the quality of the model.

Because rough sets work with discrete data, also known as categorical data, all the real value features need to be discretized. There exist many discretization methods and, again, there is no easy answer for which method to choose. It usually pays to try a few approaches and experimentally find out which approach brings the best result. A rather elegant algorithm is again based on Boolean reasoning and finds minimal numbers of cuts that split real values into intervals; for a detailed description of the algorithm, see (Komorowski et al., 1999). The result of discretization is a set of cuts $\{c_1; c_2; \ldots; c_n\}$, which correspond to intervals, $(-\infty; c_1)$, $(c_1; c_2)$, …, $(c_n; +\infty)$. If the value set of the feature is an interval $[a, b]$, the $-\infty$ and $+\infty$ are replaced by $a$ and $b$, respectively. These intervals can be given easily interpretable names. A simple example is body temperature in hypothermia, which is usually considered to be in the interval (13.0 °C; 35.0 °C; **Table 13**). (Possibly the lowest body temperature of anyone who was saved from this condition was 13.0 °C in a 7-year-old girl in Sweden in December 2010.)

**Table 13** Discretization of hypothermia temperature values

| *Profound* | *Severe* | *Moderate* | *Mild* |
|---|---|---|---|
| (13.0 °C; 20.0 °C) | (20.0 °C; 28.0 °C) | (28.0 °C; 32.0 °C) | (32.0 °C; 35.0 °C) |

Such discretization may be derived from experience or algorithmically. The cuts are applied to the learning set.

Continuous decision values are another problem. In most practical approaches, 2–5 decision classes are used. One exception is described in Hvidsten et al. (2003) and Laegreid et al. (2003), where 23 gene functional classes were modeled. If the decision is continuous, the value is typically split into 2 or 3 intervals. For instance, in the case of modeling binding affinity, all objects that had a lower binding value than some heuristically chosen threshold were assigned to the negative class and the remaining ones to the positive class. In another example of exon expression values, the decision was experimentally split into three classes by taking 20%: 60%: 20% corresponding to highly expressed, medium expressed and low expressed exons; for details see Enroth et al. (2012). An approach to rough set modeling with continuous decisions may be found in Nguyen et al. (2005).

Given a discretized decision system, all reducts are computed. They need to be computed by heuristics because the problem in general is NP-hard. After the reducts are computed, the rules are generated and the quality of the model assessed. If there is a testing set, the test set is discretized with the cuts obtained from the training set and the model is used to classify the discretized examples of the test set. The test set provides the actual classification while the model gives the predicted one and, hence, a confusion matrix is built. If there is no testing set, one usually applies cross-validation. The entire data is randomly split into ten equal subsets, and learning is performed on nine subsets and validated on the tenth subset. The procedure is repeated ten times so that each of the ten subsets is used once as the testing set. It needs to be stressed that discretization must be performed each time on the selected nine subsets and the cuts used to discretize the tenth subset. The accuracy of the model is set to the average (or mean) of the accuracies and a standard deviation is usually computed.

#### 6.02.4.1 Rule Filtering Enhances the Model and Avoids Overfitting

Some rules may be based on only one or a few examples in the training set. This may introduce overfitting, which occurs when the model is too specific to the training data. To avoid overfitting, and to increase the accuracy of the predictions, a filtering procedure may be applied to the rules. There are several filtering options. Here, we will consider a filtering procedure based on the rule support (the number of objects in the training set covered by the rule) and the rule accuracy (the fraction of correct rule predictions in the training set).

For example, having the filtering conditions support $>2$ and accuracy $>0.6$ will only keep rules supported by at least three examples, and with more than 60% accuracy in the classifier. With more strict criteria for keeping the rules, achieving a higher fraction of correct classifications can be

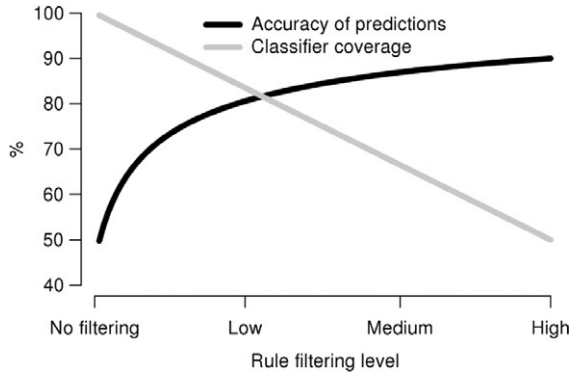


**Figure 2** A higher level of rule filtering results in more correct classifications but a lower classifier coverage.

expected, for the price of more objects that cannot be classified (Figure 2). Rule filtering may be preferable if the classifier accuracy is low.

Other methods of avoiding overfitting include generalization (Makosa, 2005) and variable precision rough sets (Ziarko, 1993).

### 6.02.4.2 Model Interpretation

In order to interpret the rules, we need to determine the ones that are the most significant, which can be estimated using the hypergeometric distribution:

$$p(x;N,n,k) = \sum_{z=x}^{\min(k,n)} \frac{\binom{k}{x}\binom{N-k}{n-x}}{\binom{N}{n}}$$

where $x$ is the number of objects that match the left-hand side (LHS) of the rule that had the predicted decision; $N$ is the number of objects in the dataset; $n$ is the number of objects that match the LHS of the rule; and $k$ is the number of objects in the dataset that have the predicted decision. Hence, the $p$-value is the probability that at least the observed number of objects that match the LHS of the rule would have the predicted decision by chance.

### 6.02.4.3 Rule Tuning

The legibility advantage of IF–THEN rule-based models will be decreased for very large rule sets. In addition to rule filtering, a technique that we call rule tuning can be applied. Mąkosa developed heuristics algorithms for rule tuning. The algorithms transform a set of rules into a smaller set containing more general rules with little loss of accuracy. This is achieved by algorithms for rule shortening and generalization. The algorithms are implemented in the ROSETTA system.

### 6.02.4.4 Using the Models

Models obtained from decision tables usually serve two purposes, prediction and description. One of the most important advantages of rule-based models is their legibility. Rule-based descriptions allow understanding which features and in which combinations are important in discerning between the classes, and for what values. In other words, rule-based models are legible even to non-experts.

Rough set models also have another property that allows for generation of new objects. It is particularly applicable in bioinformatics applications because new experiments can be generated for wet-lab validation. Provided that the features are discretized, it is straightforward to generate new objects that satisfy the definitions, from the reducts.

Let us consider again the octamers and cleavability, but now we take the physicochemical properties of amino acids instead of their names. These properties are usually numerical values and must be discretized before the reducts are computed. The corresponding rules use names of intervals. Following the idea presented in Kontijevskis et al. (2007a), the rules will refer, for example, to conjunctions of intervals. A simplified example could be as follows:

IF P1:hydrophobicity = high AND P1′:hydrophobicity = high THEN Cleave = YES.

This rule has been generated from examples of amino acids that are hydrophobic, but unless all the combinations of hydrophobic amino acids were present in the learning data set, it is possible to create new sequences of amino acids that satisfy these constraints. Hence, we can create new octamers that satisfy the model, but are absent from the learning set. This technique has been used in Kontijevskis et al. (2007b) to generate samples for wet-lab validation of the developed model. Also, Kierczak et al. (2009) uses this technique to suggest new, potential sites associated to drug resistance. For a summary, see case section '**Case Studies: Rough Sets in Bioinformatics**.'

The legibility of the IF–THEN rules is straightforward. The rules that have high support and accuracy are of obvious interest as they will be explaining a significant part of the universe and at a high quality. In interpreting the model at hand, one starts from these rules taking into account their significance, that is, $p$-values.

One should, however, bear in mind that the rules show only associations of the conditions to the outcomes, not causal explanations. For instance, rules that associate combinations of histone modifications to exon expression cannot be causal explanations. However, these associations may serve as hypotheses of causality and be used in experiments to confirm or reject them. For details, see Enroth et al. (2012).

When modeling dynamic systems, for example, gene expression, the association may be even more misleading because the changes may be associated to each other in time, but not causally. On the other hand, when interpreting properties of amino acids as in the cleavability case, the information provided by the reducts, which are minimal sets of discerning features, should be used. It has already been noticed that the reducts generated for that system do not include positions P3, P2, or P1. This is a strong indication that these positions do not play any significant role in cleavage. Now, this example is a gross simplification of the real case and should not be taken as a statement about true cleavability! A detailed investigation of the cleavage model is to be found in Kontijevskis et al. (2007a).

The original aim of classifiers was to make predictions. Predictions may also be used in a different, possibly innovative way, to validate human knowledge by reclassification and

inspection of the FP or/and FN outcomes. For details, see the description in the case studies given later.

## 6.02.5 Advanced Rough Set Modeling

### 6.02.5.1 Uneven Class Distribution

The classification may be biased when there is an uneven class distribution in the data; for example, if there are twice as many non-cleavable peptides as there are cleavable ones, then the rules may be expected to have a higher support for the larger class than for the smaller one(s). Furthermore, more rules may be generated for the more common class. As a result, the classification may be biased toward the larger class, and in the worst case, all objects will be predicted to belong to the largest class.

Two simple methods for handling uneven class distributions are undersampling and oversampling. In undersampling, all objects from the smallest class are selected and then objects are selected at random for the larger classes such that an equal number of objects from each class is chosen. Oversampling is done in the opposite way by first selecting all objects from all classes and then randomly selecting duplicates from the smaller classes until the same number of objects are chosen from each class. More complex methods exist as well, which may generate new objects from the smaller classes; see, for example, Chawla (2005).

### 6.02.5.2 Feature Selection and Random Reducts

More recently, the Monte Carlo approach was used to rank features in systems that have a very large number of attributes, much greater than the number of objects. Such systems are usually ill-defined and computing reducts for them is neither practical nor statistically sound. A Monte Carlo method for ranking features was described in Draminski et al. (2008) where decision trees were used to evaluate the quality of a classifier and find a ranking of the features. A Monte Carlo method called *random reducts* was defined and implemented by Kruczyk et al. (2013). It uses reduct computation in lieu of decision trees. Heuristics are used to identify suitable cut-off points to select significant features. Random reducts provide a computationally attractive way to find the significant features in cases where the number of features makes it infeasible to compute (approximate) reducts.

Feature selection with ranking is also attractive to the user of a rough set model because it suggests which features and in which order of significance are important in interpreting decision classes.

### 6.02.5.3 Approximate Reducts

Most real cases of decision systems are non-deterministic and some equivalence classes of objects will have more than one decision. For such systems, it can be proved that the set union of lower approximations of decision classes (concepts) will be a proper subset of the universe. Informally, crisp knowledge covers only a subset of the universe. When computing reducts for non-deterministic systems, we must recognize the fact that it is not possible to obtain crisp definitions of concepts.

Likewise, if we insist on computing very precise definitions, the reducts will become very long and specific, with a resulting overfitting of the model. It usually is advantageous to relax the requirement of finding perfect but long reducts and accept a degree of approximation. To this end, approximate reducts are computed with less important attributes omitted from the reducts. The price may be that the union of lower approximations of the decision classes is somewhat smaller, but often, the resulting model is more general and easier to compute. Further details about *approximate reducts* may be found in Komorowski et al. (1999).

### 6.02.5.4 Dynamic Reducts

When decision tables become very large, it may be computationally infeasible to compute reducts for the entire universe. Instead, a Monte Carlo-motivated approach can be taken to create samples of the universe, for example, 10% (20%, 30%) of the universe and reducts computed only for these samples. By repeating this process a sufficient number of times, it is possible to obtain an acceptable approximation of the whole universe, by selecting reducts that occur in a majority of subsystems. This concept is known as *dynamic reducts* and was introduced in Bazan et al. (1994). Random reducts allow constructing models of very large systems, with millions of objects, in the universe.

### 6.02.5.5 Visualization

Even rule-based models will be difficult to analyze if they are large. Rule filtering and tuning, discussed earlier, provide a partial solution to large cardinalities of rule sets. Another approach that aims at supporting exploration of large models is visualization.

Ciruvis (circular rule visualization) is a web-based tool for visualization and exploration of classification rules and can be used to produce publication-quality figures of classifiers. It is available at http://bioinf.icm.uu.se/~ciruvis, and ROSETTA-formatted rules as well as plain text rules are accepted as input (**Figure 3**).

The tool is focused on the rule conditions ('attribute = value') that co-occur in the rules and may be used as heuristics to find interacting attributes, which are expected to occur together. Each pair of rule conditions will receive a score based on how often they co-occur, and on the quality of the rules in which they co-occur (measures by the LHS Support and RHS Accuracy). The calculation of the scores is done decision-wise and the results are shown as circular networks, one for each decision, in which co-occurring rule conditions are connected by an edge with width relative to the score. The online versions of the networks also work interactively, and one may click on an edge between two rule conditions to view a list of all rules in which they co-occur. Thus, a large set of rules may be explored and visualized in a structured way and the resulting networks may be downloaded in a vector graphics format for use elsewhere. The visualization was applied in Enroth et al. (2012) and Bornelöv et al. (2012, 2013). For a description of the tool and its methods see Bornelöv et al. (2014).

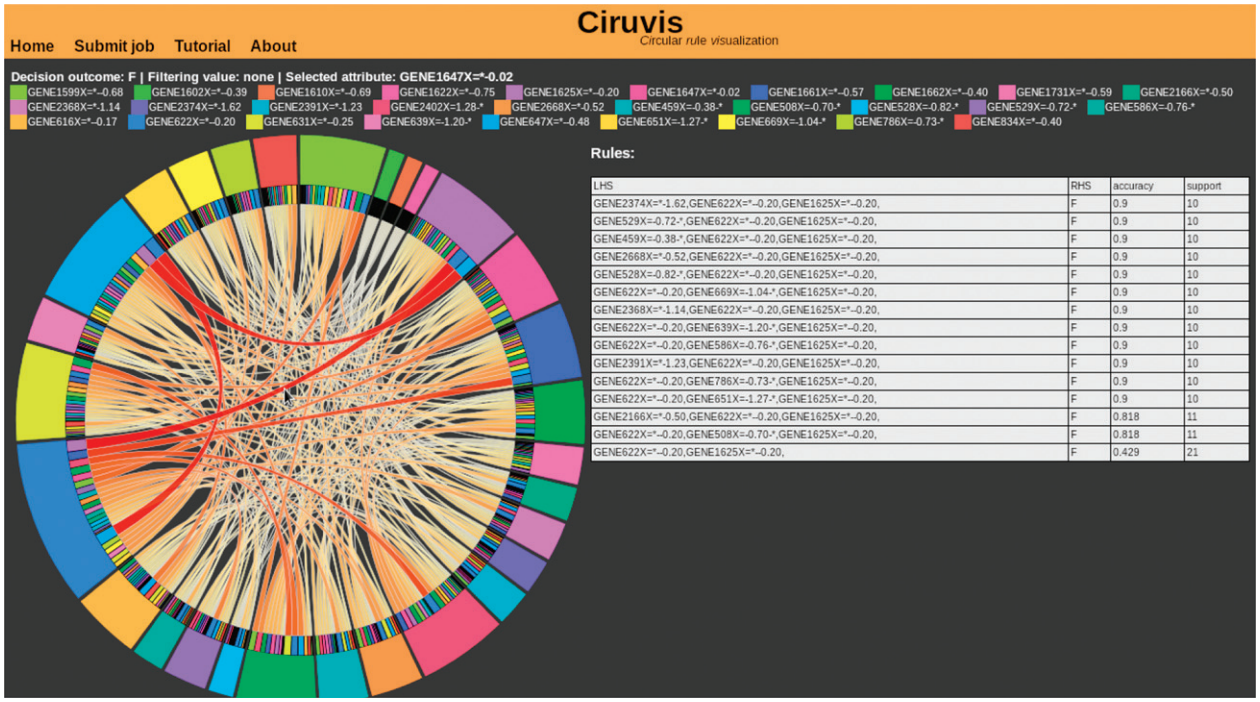


**Figure 3** Screen shot of the visualization tool Ciruvis.

### 6.02.6 Case Studies: Rough Sets in Bioinformatics

Rough sets have been used in many areas of the life sciences such as proteomics, genomics, and epigenetics and with applications that span cancer research, virology, molecular cell biology and pharmacology. This list is by no means complete. The applications presented here cover a period of over 10 years and therefore, some of the technologies used to generate data for experiments may have been significantly improved or entirely replaced since the original publication. Nevertheless, most of the results and methods seem to be applicable because even the more modern technologies suffer from noise or other data quality problems. In this section, selected research work in which rough set methodology has been used successfully is discussed, so the reader should be able to develop a good understanding of the opportunities offered by rough set methodology.

#### 6.02.6.1 Protein Analysis – From Sequences to Functions to Interactions

In this section, the results of applying rough set modeling to the analysis of proteins is discussed. The data used here are usually sequences of amino acids annotated with outcomes such as binding affinities, drug resistance, Gene Ontology labels, and other laboratory-derived or possibly human-assigned values. The models may characterize the substrates, or enzymes, or their combinations.

#### 6.02.6.2 HIV-1 Modeling

The HIV virus has a very high rate of replication characterized by many mutations and the ability to develop resistance to drugs. The HIV-1 protease plays an essential role in the replication process by cleaving the viral precursors Gag and Gag-Pol polyproteins into structural functional units. The HIV-1 protease has been an extensively studied target for the development of drugs that may inhibit the replication of the virus. The protease cleaves the polyprotein by recognizing a sequence of 8 amino acids, four on each side of the substrate's scissile bond.

Kontijevskis et al. (2007a) built a model for the HIV-1 protease cleavage of octamer peptides. (Prior to 2007, the best models of the cleavage had 40% accuracy.) The authors demonstrated that HIV-1 protease specificity is much more complex than had previously been anticipated. In particular, they showed that it cannot be defined based solely on the amino acids at the substrate's scissile bond or by any other single substrate amino acid position only. Their results showed that a combination of at least three particular amino acids is needed in the substrate for a cleavage event to occur. Only by combining and analyzing massive amounts of HIV proteome data was it possible to discover these novel and general patterns of physicochemical substrate cleavage determinants. These results clearly explained why it was not possible to obtain any alignment-based solution.

Data were collected from 16 years of HIV research (374 cleavable and 1251 non-cleavable substrates). Amino acids were replaced by their physicochemical properties and a rough set model was induced. Cross-validation showed a very high quality of the classifier: accuracy mean 93% and AUC mean 0.94. Results of the permutation test demonstrated further that it was very unlikely that valid models could be obtained based on random data (accuracy mean $76 \pm 2\%$, AUC mean $0.50 \pm 0.03$, $p$-value $<0.01$).

The analysis of the rules revealed that amino acids in at least three substrate positions define a pattern necessary for

processing by the HIV-1 protease, because no less than three physicochemical properties for three substrate positions are present in each rule for cleavage. This novel finding extended a traditional opinion at that time about the major importance of amino acids in positions P2, P1, P1′, and P2′ for HIV-1 substrates. Further analysis of the rules showed the most necessary physicochemical properties of amino acids for substrate cleavability by the HIV-1 protease, that is, hydrophobicity for the P1′ and P2 positions (present in 61 and 59 rules, respectively), polarity for the P4′ position (present in 53 rules), polarizability for the P2′ and P4 positions (found in 52 and 50 rules, respectively) and electronic effects of amino-acids for the P1 position. (The authors used slightly transformed features using principal components. In a later repetition of the construction of a rough set model, it was shown that equally good results may be obtained without this transformation (unpublished).) The authors' findings also demonstrated the major importance of hydrophobicity for substrate amino acids in the P1′ and P2 positions, and they captured new physicochemical features and their combinations preferential for the HIV-1 protease cleavage as well. The analysis of the model was also performed for the non-cleavable class.

The model created by the authors may also be used to generate substrate sequences that are cleavable and non-cleavable. Indeed, it is straightforward to convert the physicochemical ternary descriptors into amino acids. After substituting appropriate amino acids for the values of physicochemical properties, one can obtain descriptions of all objects that satisfy a given rule. An example substitution follows:

If P2 is Ala, Thr, or Val AND P2′ is Ala, Gly, Met, Glu, or His AND P3′ is Ala, Glu, Gly, His, Met, Trp, or Tyr AND P4′ is Ala, Met, Pro, Thr, or Val THEN Cleavable = YES.

Readers interested in further details are referred to the original publication (Kontijevskis et al., 2007a).

Another important drug target in HIV-1 is reverse transcriptase (RT). Kierczak et al. (2009) investigated resistance of RT to drugs. The data was obtained from the Stanford HIV Drug Resistance Database (Rhee et al., 2006). For each of the examined drugs, a number of amino acid sequences of the HIV-1 RT p66 subunit were extracted. Each sequence in the database has been annotated with the resistance value relative to the HXB2 wild-type strain. In total, there were 781 sequences of the p66 subunit (91% of them complete within the first 240 amino acid sites, 31% of them complete within all the 560 amino acid sites) that they could use for constructing data sets. Following the established clinical practice, each sequence was labeled as 'susceptible', 'moderately resistant' or 'resistant.'

Monte Carlo feature selection was used to limit a large number of features (560 amino acids and 7 properties) and a rough set model was developed with ROSETTA. Rules were of the following form:

IF (polarity at site 101 = $(-\infty, 2.100)$) AND (normalized freq. of turn at site 190 = $(0.045, \infty)$)
THEN resistant to Nevirapine

Models were simplified using Mąkosa's rule tuning and validated on external data sets. The accuracy range was [0.69, 0.89] depending on the drug, with SD in the range [0.03, 0.07]. Already, at the level of feature selection, known sites were confirmed and new ones discovered.

Similar to the earlier mentioned work on proteases, the rules are conjuncts of intervals of values of the physicochemical properties of amino acids. This allows seeing which amino acids fulfill the criteria imposed by a given rule and also when such amino acids were not represented in the training set. For instance, given the following rule,

IF P101 polarity $((-\infty, 2.100))$ AND P190 freq. turn $((0.045,\infty))$ THEN resistant to Nevirapine

it is easy to find which amino acids satisfy the conditions, and to substitute them into the rule:

IF P101(any of D,E,H,K,N,Q,R) AND P190(any but: A,G,N, P,Y) THEN resistant to Nevirapine

Even though asparagine (N) was not observed at site 101 in the available data, this general model is able to foresee that an occurrence of such a mutation may result in the acquisition of resistance.

A distinctive result of the first work was a comparison of the rough model with rules developed by experts. While accurate, the experts' rules were applicable to only a very limited fraction of sequences. Rough set models had a significantly higher coverage, showing a potential for new discoveries. This study was followed by a work on a network of interactions between RT positions in response to RT inhibitors (Kierczak et al., 2010).

### 6.02.6.3 Protein–Ligand Interactions

Strömbergsson and her colleagues proposed a new approach to modeling interactions between proteins and ligands (Strömbergsson et al., 2004, 2006a). The traditional approach is QSAR (Quantitative Structure Activity Relationship), which models the interactions between one protein and a series of ligands, and docking, where the three-dimensional structure of the protein is used to model the protein–ligand complex. Strömbergsson et al proposed to use a series of ligand–protein interactions for modeling. They used rough set-based rule learning to model interactions between G-protein-coupled receptors (GPCR) and ligands. GPCRs are membrane-bound proteins that share a conserved structural topology of seven transmembrane helices. GPCRs are of particular interest, as about 50% of all recently launched drugs are targeted toward these receptors. The main novel result of this study was that rules allowed a direct interpretation of the model, something that is not possible with the commonly used linear regression approach. For example, their rule model suggested that helix 2 was determinant of high and low binding affinity in three different data sets and ligands. This approach greatly reduced the number of known interactions needed for modeling and may predict cross-interactions between drugs and other proteins in the proteome.

In Strömbergsson et al. (2006b), the authors showed that using local descriptors of protein structure (Hvidsten et al., 2004), one can model vastly different proteins in terms of both sequence and structure. It was shown that the induced rule model combined local substructures and ligand descriptors to generalize beyond the enzyme–ligand interactions

present in the training set. An interesting interpretation from the rules was that strongly bound enzyme–ligand complexes were described in terms of the presence of specific local substructures, while weakly bound complexes were described by the absence of certain local substructures. This is intuitive, because there may be only one or a few ligands that geometrically fit the active site of a specific enzyme and form a strongly bound complex, while there may be many ligands that form only weakly bound complexes with the same enzyme. The preferred description of the latter is to point to the absence of the local substructure that, if present, would have resulted in a strongly bound complex.

### 6.02.6.4 Function Prediction from Structure

Although global structural similarity is often a sign of functional similarities (Pazos and Sternberg, 2004), many folds such as the TIM barrel and the Rossmann fold are found in proteins with many different functions. Thus, local similarity methods are more powerful in these cases (Orengo et al., 1999). By the middle of the first decade of 2000, researchers had started building tools that used a large number of different features including both local and global structure (Laskowski et al., 2005; Pal and Eisenberg, 2005). These so-called meta-servers obtained functional predictions by allowing a large number of different evidence to vote, and then selecting the most likely function. However, such approaches do not construct explicit models that are often very useful in further analysis.

Hvidsten et al. (2009) proposed a change in this paradigm by inducing IF–THEN rules that associate combinations of local substructures with specific protein functions (**Figure 4**). This approach differed from other studies in that the applied library of local substructures encompasses all recurring motifs and all annotated proteins using no prior knowledge of functional sites or any sequence information, and also in that the structure–function relationship is explicitly represented in a descriptive model. Moreover, the structure–function relationship in proteins was quantified by assessing the predictive performance of the model using cross-validation and AUC analysis. One of the findings in that work explained the observed better predictability of GO biological processes compared with GO molecular function (Brown et al., 2000; Hvidsten et al., 2005).

### 6.02.6.5 Rough Sets in Genomics and Transcriptomics

In the early 2000s, relatively large data sets provided by cDNA microarray technology enabled new forms of modeling in functional genomics. Large-scale measurements of RNA in cells became possible and researchers started to characterize the state of a cell in terms of RNA levels. Initially, clustering techniques were used (Iyer et al., 1999), but soon they gave room to supervised learning (Brown et al., 2000; Golub et al., 1999). Although the dominating RNA technology is RNA sequencing much of the experience obtained with cDNA microarrays remains valid and may be applied to new data types.

A series of papers (Hvidsten et al., 2001, 2005; Wabnik et al., 2009) explored the possibilities of modeling gene function and gene regulatory circuitry using expression data, which were eventually combined with other data such as binding sites. To our knowledge, the use of Gene Ontology terms (Ashburner et al., 2000) to label the outcomes of genes was possibly the very first use of Gene Ontology in supervised learning.

### 6.02.6.6 Functional Genomics

In Hvidsten et al. (2001), a method for modeling participation of gene products in biological processes was developed from temporal expression profiles. Earlier work had used hierarchical clustering to find associations between expression similarity and gene function (e.g., Brown and Botstein, 1999; Eisen et al., 1998; Iyer et al., 1999). A major problem with this work was that functionally related genes may be anti-co-regulated, and that genes have multiple functions. Brown et al. (2000) were the first to use a supervised learning approach using support vector machines, but their result was restricted to predicting only six functional categories.

The work of Hvidsten et al. (2001) introduced a template language to describe time profiles of gene activity in qualitative rather than absolute terms. For example, the rules

IF 0H–4H(constant) AND 0H–10H(increasing) THEN
GO(protein metabolism and modification) OR
GO(mesoderm development) OR GO(protein biosynthesis)

and

IF 0H–4H (increasing) AND 6H–10H (decreasing) AND
14H–18H (constant) THEN
GO (cell proliferation) OR GO (cell–cell signaling) OR
GO (intracellular signaling cascade) OR GO (oncogenesis)

associate a specific profile of expression level changes (between time point 0 and the 10th hour after the cells were stimulated, the expression level increases) with the specified biological processes. The support of the first rule is five genes, four of which have annotation *protein metabolism and modification*. The second one has a support set of four, out of which three are

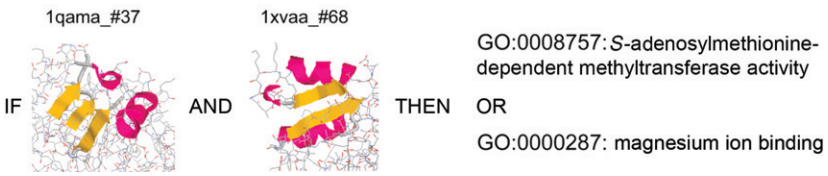


**Figure 4** The rule IF (1qama_#37 AND 1xvaa_#68) THEN (GO:0008757 OR GO:0000287) combining the substructure 1qama_#37 in A with the substructure 1xvaa_#68 to uniquely describe 12 of the proteins annotated with GO:0008757: S-adenosylmethionine-dependent methyltransferase activity. Two of these proteins are additionally annotated with GO:0000287: magnesium ion binding. The rule thus effectively combines local substructures to address only one of the three statistically significant GO classes related to 1qama_#37 (doi:10.1371/journal.pone.0006266.g001).

annotated to *cell proliferation*. Such *non-deterministic* rules have a significant advantage over clustering and even supervised learning approaches that return the strongest classification since the non-deterministic rules allow us to capture the multiple roles biological entities such as genes usually have.

The model was evaluated and found to be statistically significant for 23 out of 27 classes of the biological processes modeled. The predictions for uncharacterized, at the time, genes were evaluated by searching for homology information that could be used to formulate assumptions about the participation of these genes in biological processes. Eleven genes out of the twenty-four for which such assumptions could be made had one or more classification that matched this assumption.

Figure 5 shows on one side the rules generated for process transcription, and on the other side, the graphs of the learning set, that is, genes annotated to class transcription and the results of model application to unknown genes. It is highly instructive to see the difference between the rules and how different the profiles may be. Obviously, there is no straightforward (unsupervised) clustering that would put these genes into one and the same cluster. It should also be noticed how few features are sufficient to assign objects to a class. This is one of the main advantages of the minimization procedure (reduct computation) of rough sets.

In addition to predicting the biological process of uncharacterized genes, a model induced from all examples was also used to reclassify characterized genes. The resulting false positives were then used to guide a second literature search for possible missing annotations (i.e., information on biological process annotations existing in the literature, but overlooked during the initial literature search). Of the 14 genes with a false positive reclassification to DNA metabolism, four were found to actually participate in this process. Furthermore, it was revealed that 12 of the 24 false positive reclassifications to oncogenesis also represented missing annotations. Thus, it was shown that computational models could be used directly to both guide new literature searches for partially characterized genes and propose new functional hypotheses for unseen genes.

These studies used biological processes provided by GO to learn their definitions from gene expression profiles. In Midelfart et al. (2002), rough set rule classifiers learnt from the Gene Ontology taxonomy which biological processes would be the best predictors.

### 6.02.6.7 Rough Sets in Cancer Research

Dennis et al. (2005) used rough sets to construct a classifier for identifying the primary site of cancer based on expression levels in a sample taken from a secondary tumor. About 10–15% of cancers are discovered as metastases in solid organs, body cavities, or lymph nodes. Most of these secondary tumors are adenocarcinomas, for which the seven commonest primary sites are breast, colon, lung, ovary, pancreas, prostate, and stomach. Because prognosis and therapy are linked to the site of origin, and because histologically such tumors appear

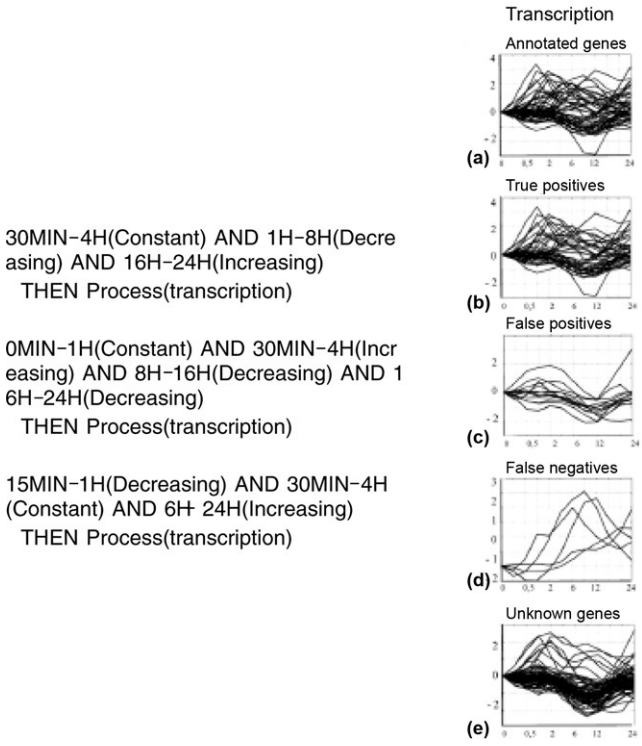


**Figure 5** Rules and expression profiles of annotated and classified genes for the transcription process. The *x*-axis shows time, and the *y*-axis shows log 2-transformed gene expression ratios (serum-treated vs. control). For each process, the following expression profiles are shown: (a) training example genes annotated with the process; (b) training example genes correctly classified to the process, that is, true positives; (c) training example genes classified but not annotated to the process, that is, false positives; (d) training example genes that the rule model failed to classify with the biological process to which they were annotated, that is, false negatives; and (e) unknown (uncharacterized) genes classified to the process.

similar, finding molecular markers for these sites may greatly improve treatment. The study assessed the expression patterns of 27 markers in 452 adenocarcinoma patients. 12 markers were scored as either present or absent (+ or −), while the remaining markers were scored as absent, weak, intermediate, or strong (0, 1, 2, or 3). Decision rules were induced from 352 adenocarcinomas and used to build a decision tree of ten markers. This tree was then used to predict the site of origin of 100 unseen adenocarcinomas with a success rate of 88%. This is a very high accuracy considering there were seven different sites to predict, and indicates a huge potential for molecular markers in identifying the primary site of these cancers.

### 6.02.6.8 Combinatorial Gene Regulation

Pilpel et al. (2001) found that genes sharing pairs of binding sites are significantly more likely to be co-expressed than genes with only single binding sites in common. This result was in agreement with the hypothesis that a limited number of transcription factors combine in various ways in order to respond to a large number of various stress conditions.

(Hvidsten et al. (2005) used rough set modeling to perform a comprehensive analysis of the combinatorial nature of gene regulation in yeast. The IF part of the rules consisted of conjunctions of minimal binding site combinations or modules shared by genes, and the THEN part was their common expression profile. The rules provided hypotheses on combinatorial co-regulation that may later be experimentally validated.

The approach was tested on a database of known and putative regulatory sequence motifs in yeast (Hughes et al., 2000) using six expression data sets including one cell cycle study and five studies of different stress conditions (Pilpel et al., 2001). The rule learning framework was subsequently applied to each gene to obtain rules that associate the expression profile of that gene with a minimal binding site combination shared by similarly expressed genes.

The results were statistically significant compared to genes associated with either a randomly chosen set of binding sites, or a similar expression, or neither of these constraints. Two rules were discussed as a case study and had support in the literature.

As an example, the rule

IF RAP1 AND MCM1 AND SWI5 THEN Similar expression

describes eight genes and suggests that the transcription factor RAP1 (which regulates genes that encode ribosomal proteins in growing yeast cells, and also other non-ribosomal genes) requires the cell cycle regulating transcription factors MCM1 and SWI5 to be present when specifically targeting ribosomal genes in growing yeast. In other words, the ribosomal genes targeted by RAP1 are regulated only when the cell is in the cell cycle (i.e., growing), that is, when MCM1 and SWI5 are present. RAP1 presumably combines with other transcription factors when regulating other non-ribosomal genes. By applying the method to expression data obtained under several different conditions, the authors were able to discover a number of binding site modules.

A follow-up study (Wilczyński et al., 2006) used expression similarity restricted to subintervals of cell cycle time profiles (similar to the template language discussed earlier), and showed that this improvement greatly increased the biological significance of the retrieved modules in addition to making it possible to retrieve modules that were not detectable using expression similarity over the whole time profile.

A second follow-up study (Andersson et al., 2007) refrained from using expression similarity altogether. Instead, this study used prior knowledge of the cell cycle period time to detect different classes of periodically expressed genes in three different synchronization studies, and then used rough set modeling to describe the regulatory mechanisms behind these classes. These mechanisms were then shown to be much more specific toward the cell cycle machinery than mechanisms discovered from expression clusters, and thus showed the advantage of incorporating biological knowledge into the data analysis process whenever possible.

The last work in this cycle (Wabnik et al., 2009) explored the ability of rough sets to integrate expression and protein data. Expression data, protein features, and Gene Ontology (GO) annotations were combined to describe general and biologically relevant patterns represented by IF–THEN rough set rules. These rules were used to predict the function of unknown genes in *Arabidopsis thaliana* and *Schizosaccharomyces pombe*. The models showed success rates of up to 0.89 (discriminative and predictive power for both modeled organisms), while models built solely of one data type (protein features or gene expression data) yielded success rates varying from 0.68 to 0.78. These models were applied to generate classifications for many unknown genes, of which a sizable number were confirmed either by PubMed literature reports or by electronically interfered annotations. Finally, the authors studied cell cycle protein–protein interactions derived from both tandem affinity purification experiments and *in silicium* experiments in the BioGRID interactome database and found strong experimental evidence for the predictions generated by the models. The results show that rough sets can be used to build very robust models that create synergy from integrating gene expression data and protein features.

### 6.02.6.9 Epigenetics – The Histone Code

The process of concatenating the exons into a complete transcript, splicing, involves elimination of introns and specific exons and is performed by the spliceosome, a massive complex containing hundreds of proteins. The constitution and function of the spliceosome is not yet fully known. Conceptually, splicing can be achieved in two ways, either post-transcriptional or co-transcriptional. The classical textbook model is post-transcriptional where the whole mRNA is first transcribed and then the introns and, possibly, some exons are removed. Recently, the co-transcriptional model has been proposed where inclusion/exclusion of a specific exon into the mRNA is decided before the whole mRNA is transcribed; see, for example, (Pandit et al. (2008). The co-transcriptional model puts the spliceosome close to the DNA during transcription and it thus has the possibility to read and recognize the histone code. Several DNA-binding proteins and chromatin remodelers have been shown to be important in controlling this process, and recently, post-translational modifications to the histone proteins have been shown to, at least partly,

regulate exon inclusion/exclusion in gene transcripts (e.g., Andersson et al., 2009; Luco et al., 2010; Tilgner and Guigó, 2010).

Investigations concentrated on finding the strongest relations between a *single* histone modification and the expression of the exon and the combinatorial aspects have not been comprehensively addressed. Following the study in Andersson et al. (2009) on nucleosome positioning and histone modifications over internal exons, Enroth et al. (2012) introduced a combinatorial model that better reflects a part of the complex biological machinery behind splicing. The publicly available data on histone methylation and acetylation were due to Barski et al. (2007) and Wang et al. (2008) and the exon expression data due to Oberdoerffer et al. (2008). Firstly, Monte Carlo feature selection (Draminski et al., 2008) was performed on the 114 features (3 positions, 38 histone modifications) and the significant features selected for rough set model construction. A typical rule would read as follows:

IF H2BK5me1 preceding exon is absent AND H3K4me1 succeeding exon is present AND
H3K36me3 succeeding exon is absent AND H4K29me1 preceding exon is present THEN exon is excluded

The resulting model was evaluated with cross-validation and had an average accuracy of 72% for 27% of the exons, which demonstrated that epigenetic signals mark splicing. The most common number of condition attributes was five (56%) for the 'spliced-out' class and six to eight (75%) for the 'included' class. In total, 66% of the rules contained 6–8 condition attributes and no rule consisted of a single attribute. Only 6 rules out of the total 165 did not require at least one histone modification to be absent from the region. The interplay between histone modifications was very high.

Histone modifications previously identified as related to exon expression (Andersson et al., 2009; Hon et al., 2009) were present in the rules (e.g., H2BK5me1 and H4K20me1) as well as previously less well studied modifications. However, the strongest univariate candidates, for example, H3K79me1, H3K79me3, and H3K36me3, were all selected as significant by the MCFS, but only H3K36me3 succeeding and preceding the exon were among the 20 highest ranked modifications and thus included in the rule model. Surprisingly, H3K36me3 was always required to be 'absent' in the rules for both decisions although it had previously been suggested that its presence is related to inclusion.

The model showed that a substantial proportion of alternative splicing events may be attributed to the combinatorial status of histone modifications on nucleosomes preceding, on, or succeeding the exon, and that the combination of specific combinations of histone modifications are often better predictors of exon inclusion levels than single histone modifications. This work supports the co-transcriptional view on inclusion/exclusion of a specific exon and is a contribution to deciphering of the histone code.

#### 6.02.6.10 Feature Interaction: Gene–Gene and Gene–Environment Interactions in Allergy

Genetic and environmental factors are important for the development of allergic diseases. However, a detailed understanding of how such factors interact is lacking. To elucidate the interplay between genetic and environmental factors in allergic diseases, Bornelöv et al. (2013) used Monte Carlo feature selection and rough set modeling. In two materials, PARSIFAL (a European cross-sectional study of 3113 children) and BAMSE (a Swedish birth-cohort including 2033 children), genetic variants as well as environmental and lifestyle factors were evaluated for their contribution to allergic phenotypes. MCFS and rough set models were used to identify and rank rules describing how combinations of genetic and environmental factors affect the risk of allergic diseases. Novel interactions between genes were suggested and replicated, such as between ORMDL3 and RORA, where certain genotype combinations gave odds ratios for current asthma of 2.1 (95% CI 1.2–3.6) and 3.2 (95% CI 2.0–5.0) in the BAMSE and PARSIFAL children, respectively. Several combinations of environmental factors appeared to be important for the development of allergic disease in children. For example, the use of baby formula and antibiotics early in life was associated with an odds ratio of 7.4 (95% CI 4.5–12.0) of developing asthma. Furthermore, genetic variants together with environmental factors seemed to play a role in the development of allergic diseases, such as the use of antibiotics early in life and COL29A1 variants for asthma, and farm living and NPSR1 variants for allergic eczema. Overall, combinations of environmental and life style factors appeared more commonly in the models than combinations solely involving genes. Interactions identified with this approach could provide useful hints for further in-depth studies of etiological mechanisms and also strengthen the basis for risk assessment and prevention.

### 6.02.7 Rough Sets Versus Statistical Classification

The field of classification has a long and rich development. The linear discriminant analysis (LDA) was the first method developed for multidimensional classification. LDA has been in use for decades and is still used and often compared to. Other methods that relaxed some of the limitations of LDA, such as multivariate normality, equality of dispersion matrices between groups, followed. They include quadratic discriminant analysis (QDA), logit and probit analysis, and the linear probability model for two-group problems.

Several non-parametric techniques, including rough sets, had already been developed by the end of the 1900s: multicriteria decision methods (e.g., Doumpos et al., 2000), mathematical programming (e.g., Freed and Glover, 1981), neural networks (e.g., Patuwo et al., 1993), and machine learning approaches (Quinlan, 1986). Rough sets, a machine learning technique, stands out in this group with several attractive properties such as data reduction, uncertainty handling, and ease of model interpretation. A Monte Carlo simulation study of the performance of rough sets was done by Doumpos and Zopounidis (2002). The study found that

> rough sets perform well in comparison to well-established approaches, at least in the cases where the data originate from asymmetric distributions. The major rival of the rough sets, the QDA also performs well in many cases, but the form of the group dispersion matrices heavily affects its performance. [...] there are two significant features of the rough sets theory that should be emphasized, with regard to providing meaningful decision support.

First, they provide a sound mechanism for data reduction, and second they enable the development of rule-based classification models that are easy to understand. This second feature is very significant in terms of decision support, and it is hardly shared by the statistical approaches explored in this study.

### 6.02.8 Other Learning Approaches in Bioinformatics

Machine learning has a long tradition in bioinformatics, which can be exemplified by the well-known book 'Bioinformatics – The Machine Learning Approach' (Baldi and Brunak, 2001). On examining its content, one can see how widespread the field is. Among the methods used in the book, we find no mention of decision trees or other rule-based formalisms including rough sets, or support vector machines, for that matter. Obviously, the field is moving and new formalisms are coming in while others are being phased out. Rather than discussing which approach is superior, we should look at the needs of a bioinformatician.

If the requirement is the best possible classification, then the user should make his choice accordingly. Often, the answer given to this question is neural network, support vector machines, or random forest. It appears that these methods, while powerful classifiers, are *opaque*, that is, it is difficult if not impossible to interpret the model and understand how the decision was made. If, on the other hand, the requirement is an interpretable model with explanatory power, *transparent* methods are the choice. Transparent methods are exemplified by rule-based approaches of several different kinds. They include decision trees.

There are also pragmatic criteria, which refer to the ease of use. It can be best compared to the support provided by a programming environment. A programming language may offer very useful abstractions but without a good support for writing, debugging, and editing the programs, it will be much less attractive. We believe that a user interface, or indeed, a programming environment for learning should support these aspects. Classifier development is very much a trial and error process. Hence, the user should investigate which learning system provides tools to this end. In this context, the ROSETTA system offers an environment that supports the process of developing a classifier, including an automatic scripting facility that allows future repetition of classifications that are often based on random choices and need to be repeated several times.

#### 6.02.8.1 Rough Set Resources

Rough sets were introduced in 1982 by the late Pawlak (1982). A second important publication by this author was Pawlak (1997). The first publication has 10 899 citations in Google Scholar (October 2013). The appearance of tools for modeling with rough sets such as LERS (Grzymala-Busse, 1992), RSES (Bazan and Szczuka, 2001), and ROSETTA systems (Komorowski et al., 2002; Øhrn and Komorowski, 1997). ROSE (Predki et al., 1998) have enabled practical applications of rough sets. Already, in 1999, a survey conducted by Goebel and Gruenwald (1999) counted rough sets among the nine most used methodologies. Rough sets have been used in many areas. For example, in multicriteria choice and ranking problems, Greco et al. (1999) introduced rough approximation of a preference relation by dominance relations. **Table 14** gives the results of searching for keywords on rough sets and bioinformatics in PubMed (http://www.ncbi.nlm.nih.gov/pubmed) and Google Scholar (http://scholar.google.com) in years 2007 and 2013. There is roughly a sixfold increase in publications on rough sets and a tenfold increase on rough sets and bioinformatics.

Transactions on Rough Sets (http://www.springer.com/series/7151) and Lecture Notes Transactions on Rough Sets (http://www.springer.com/computer/lncs?SGWID=0-164-2-99627-0) are the specialized publishing arenas for the rough set community. The *Fundamenta Informaticae* journal (http://fi.mimuw.edu.pl/index.php/FI) publishes a significant number of theoretical papers devoted to rough sets. A comprehensive database of rough set publications can be found at http://rsds.ur.edu.pl. It currently contains 3407 authors and 5183 articles.

A mathematically founded introduction to rough sets was presented in Komorowski et al. (1999). An earlier overview of the applications of rough sets in bioinformatics can be found in Hvidsten and Komorowski (2007). The original presentation of rough sets with examples is to be found in Pawlak (1992).

The presentation of the applications is by its nature brief and concentrates on the most significant findings. The reader should consult the original publications for a complete description.

#### 6.02.8.2 Software Availability

The ROSETTA system, which served the development of all rough set models described in this chapter, is available at http://www.lcb.uu.se/tools/rosetta/. A free version for academic use, which is limited to decision tables of at most 25 attributes and 500 objects, can be freely downloaded there.

## Acknowledgments

I am indebted to the late Ździsław Pawlak for introducing me to his rough sets. My thanks also go to Andrzej Skowron who

**Table 14** PubMed: titles and abstracts search, Google Scholar: whole text search

| *Source* | *Year* | *Keyword* | *Keywords* |
|---|---|---|---|
| | | 'Rough set*' | 'Rough set*' AND 'bioinformatics' |
| PubMed | 2007 | 69 | n/a |
| | 2013 | 171 | n/a |
| Google scholar | 2007 | 18 100 | 694 |
| | 2013 | 66 500 | 2780 |

has carried out the main theoretical developments in rough sets and inspired me and literally hundreds of researchers worldwide to work with this powerful and elegant formalism. I wish also to acknowledge his and his team's generous help in letting us use the early version of RSES in ROSETTA. The work presented here would not have been possible without some great contributions from my past and present postdocs, PhD, and Master's students in Norway and Sweden. The list would be long, so please check the bibliography as my thanks go to all the authors. Among them, Torgeir R. Hvidsten deserves a special mention because he and I published together the largest number of papers on rough sets and a majority of the work described here comes from our collaboration. The case studies section of this chapter is based on our earlier paper 'Rough Sets in Bioinformatics'. And so was the collaboration with colleagues in the life sciences who fruitfully provided knowledge to interpret our results and had the patience to learn about this new and unusual, but in their opinion, very attractive approach. Final thanks are due to my colleagues, Bengt Persson who invited me to write this chapter and provided continuous encouragement; Susanne Bornelöv, my PhD student, for contributing to this chapter; and Nicholas Baltzer, research assistant, for running the examples in ROSETTA.

## References

Andersson R, Enroth S, Rada-Iglesias A, Wadelius C, and Komorowski J (2009) Nucleosomes are well positioned in exons and carry characteristic histone modifications. *Genome Research* 19(10): 1732–1741.

Andersson C, Hvidsten T, Isaksson A, Gustafsson M, and Komorowski J (2007) Revealing cell cycle control by combining model-based detection of periodic expression with novel cis-regulatory descriptors. *BMC Systems Biology* 1(1): 45.

Ashburner M, Ball CA, Blake JA, et al. (2000) Gene Ontology: Tool for the unification of biology. *Nature Genetics* 25(1): 25–29.

Baldi P and Brunak S (1998) *Bioinformatics: The Machine Learning Approach*, vol. 1. Cambridge, MA: The MIT Press.

Baldi P and Brunak S (2001) *Bioinformatics: The Machine Learning Approach*. Cambridge, MA: MIT Press.

Barski A, Cuddapah S, Cui K, et al. (2007) High-resolution profiling of histone methylations in the human genome. *Cell* 129(4): 823–837.

Bazan J, Skowron A, and Synak P (1994) Dynamic reducts as a tool for extracting laws from decisions tables. In: Raś Z and Zemankova M (eds.) *Methodologies for Intelligent Systems*, vol. 869, pp. 346–355. Berlin/Heidelberg: Springer.

Bazan JG and Szczuka M (2001) RSES and RSESlib – A collection of tools for rough set computations. In: Ziarko W and Yao Y (eds.) *Rough Sets and Current Trends in Computing*, vol. 2005, pp. 106–113. Berlin/Heidelberg: Springer.

Bornelöv S, Enroth S, and Komorowski J (2012) Visualization of rules in rule-based classifiers. In: Watada J, Watanabe T, Phillips-Wren G, Howlett RJ, and Jain LC (eds.) *Intelligent Decision Technologies*, pp. 329–338. Berlin/Heidelberg: Springer.

Bornelöv S, Marillet S, and Komorowski J (2014) Ciruvis: A web-based tool for rule networks and interaction detection using rule-based classifiers (under revision).

Bornelöv S, Sääf A, Melén E, et al. (2013) Rule-based models of the interplay between genetic and environmental factors in childhood allergy. *PLoS One* 8(11): e80080.

Brown FM (2003) *Boolean Reasoning: The Logic of Boolean Equations*. Mineola: Dover Publications.

Brown PO and Botstein D (1999) Exploring the new world of the genome with DNA microarrays. *Nature Genetics* 21: 33–37.

Brown MP, Grundy WN, Lin D, et al. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences* 97(1): 262–267.

Chawla NV (2005) Data mining for imbalanced datasets: An overview. In: *Data Mining and Knowledge Discovery Handbook*, pp. 853–867. Berlin: Springer.

Dennis JL, Hvidsten TR, Wit EC, et al. (2005) Markers of adenocarcinoma characteristic of the site of origin: Development of a diagnostic algorithm. *Clinical Cancer Research* 11(10): 3766–3772.

Doumpos M and Zopounidis C (2002) Rough sets and multivariate statistical classification: A simulation study. *Computational Economics* 19(3): 287–301.

Doumpos M, Zopounidis C, and Pardalos PM (2000) Multicriteria sorting methodology: Application to financial decision problems. *Parallel Algorithms and Application* 15(1–2): 113–129.

Draminski M, Rada-Iglesias A, Enroth S, Wadelius C, Koronacki J, and Komorowski J (2008) Monte Carlo feature selection for supervised classification. *Bioinformatics* 24(1): 110–117.

Edgington ES (1969) Approximate randomization tests. *The Journal of Psychology* 72(2): 143–149.

Eisen MB, Spellman PT, Brown PO, and Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* 95(25): 14863–14868.

Enroth S, Bornelov S, Wadelius C, and Komorowski J (2012) Combinations of histone modifications mark exon inclusion levels. *PLoS One* 7(1): e29911.

Freed N and Glover F (1981) Simple but powerful goal programming models for discriminant problems. *European Journal of Operational Research* 7(1): 44–60.

Goebel M and Gruenwald L (1999) A survey of data mining and knowledge discovery software tools. *SIGKDD Exploration Newsletter* 1(1): 20–33.

Golub TR, Slonim DK, Tamayo P, et al. (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286(5439): 531–537.

Greco S, Matarazzo B, and Słowiński R (1999) Rough approximation of a preference relation by dominance relations. *European Journal of Operational Research* 117(1): 63–83.

Grzymala-Busse JW (1992) LERS – A system for learning from examples based on rough sets. In: Słowiński R (ed.) *Intelligent Decision Support*, vol. 11, pp. 3–18. Netherlands: Springer.

Hon G, Wang W, and Ren B (2009) Discovery and annotation of functional chromatin signatures in the human genome. *PLoS Computational Biology* 5(11): e1000566.

Hughes JD, Estep PW, Tavazoie S, and Church GM (2000) Computational identification of *Cis-regulatory* elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *Journal of Molecular Biology* 296(5): 1205–1214.

Hvidsten TR and Komorowski J (2007) Rough sets in bioinformatics. In: Peters J, Skowron A, Marek V, Orłowska E, Słowiński R, and Ziarko W (eds.) *Transactions on Rough Sets VII*, vol. 4400, pp. 225–243. Berlin/Heidelberg: Springer-Verlag.

Hvidsten TR, Komorowski J, Sandvik AK, and Lægreid A (2001) Predicting gene function from gene expressions and ontologies. *Pacific Symposium on Biocomputing* 6: 299–310.

Hvidsten TR, Kryshtafovych A, Komorowski J, and Fidelis K (2004) A novel approach to fold recognition using sequence-derived properties from sets of structurally similar local fragments of proteins. *Bioinformatics* 20(2): 293.

Hvidsten TR, Laegreid A, and Komorowski J (2003) Learning rule-based models of biological process from gene expression time profiles using gene ontology. *Bioinformatics* 19(9): 1116–1123.

Hvidsten TR, Lægreid A, Kryshtafovych A, Andersson G, Fidelis K, and Komorowski J (2009) A comprehensive analysis of the structure–function relationship in proteins based on local structure similarity. *PLoS One* 4(7): e6266.

Hvidsten TR, Wilczynski B, Kryshtafovych A, Tiuryn J, Komorowski J, and Fidelis K (2005) Discovering regulatory binding-site modules using rule-based learning. *Genome Research* 15(6): 856–866.

Iyer VR, Eisen MB, Ross DT, et al. (1999) The transcriptional program in the response of human fibroblasts to serum. *Science* 283(5398): 83–87.

Kierczak M, Draminski M, Koronacki J, and Komorowski J (2010) Computational analysis of molecular interaction networks underlying change of HIV-1 resistance to selected reverse transcriptase inhibitors. *Bioinformatics and Biology Insights* 4: 137–146.

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

Komorowski J, Øhrn A, and Skowron A (2002) ROSETTA rough sets. In: Klösgen W and Zytkow J (eds.) *Handbook of Data Mining and Knowledge Discovery*, pp. 554–559. New York, NY: Oxford University Press.

Komorowski J, Pawlak Z, Polkowski L, and Skowron A (1999) Rough sets: A tutorial. In: *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, 3–98.

Kontijevskis A, Petrovska R, Mutule I, et al. (2007a) Proteochemometric analysis of small cyclic peptides' interaction with wild-type and chimeric melanocortin receptors. *Proteins* 69(1): 83–96.

Kontijevskis A, Wikberg JE, and Komorowski J (2007b) Computational proteomics analysis of HIV-1 protease interactome. *Proteins* 68(1): 305–312.

Kruczyk M, Baltzer N, Mieczkowski J, Dramiński M, Koronacki J, and Komorowski J (2013) Random reducts: A Monte Carlo rough set-based method for feature selection in large datasets. *Fundamenta Informaticae* 127(1): 273–288.

Laegreid A, Hvidsten TR, Midelfart H, Komorowski J, and Sandvik AK (2003) Predicting gene ontology biological process from temporal gene expression patterns. *Genome Research* 13(5): 965–979.

Laskowski RA, Watson JD, and Thornton JM (2005) ProFunc: A server for predicting protein function from 3D structure. *Nucleic Acids Research* 33(supplement 2): W89–W93.

Luco RF, Pan Q, Tominaga K, Blencowe BJ, Pereira-Smith OM, and Misteli T (2010) Regulation of alternative splicing by histone modifications. *Science* 327(5968): 996–1000.

Makosa E (2005) *Rule Tuning*. Sweden: Uppsala University pp. 1–51.

Midelfart H, Komorowski J, Nørsett K, Yadetie F, Sandovik AK, and Lægreid A (2002) Learning rough set classifiers from gene expressions and clinical data. *Fundamenta Informaticae* 53(2): 155–183.

Nguyen HS, Luksza M, Makosa E, and Komorowski J (2005) Rough set approach to mining data with continuous decision value. In: *Rough Set Techniques in Knowledge Discovery and Data Mining*, 45.

Oberdoerffer S, Moita LF, Neems D, Freitas RP, Hacohen N, and Rao A (2008) Regulation of CD45 alternative splicing by heterogeneous ribonucleoprotein, hnRNPLL. *Science* 321(5889): 686–691.

Øhrn A and Komorowski J (1997) ROSETTA – A rough set toolkit for analysis of data. In: *Proceedings of the Third International Joint Conference on Information Sciences* Citeseer.

Orengo CA, Todd AE, and Thornton JM (1999) From protein structure to function. *Current Opinion in Structural Biology* 9(3): 374–382.

Pal D and Eisenberg D (2005) Inference of protein function from protein structure. *Structure* 13(1): 121–130.

Pandit S, Wang D, and Fu X-D (2008) Functional integration of transcriptional and RNA processing machineries. *Current Opinion in Cell Biology* 20(3): 260–265.

Patuwo E, Hu MY, and Hung MS (1993) Two-group classification using neural networks. *Decision Sciences* 24(4): 825–845.

Pawlak Z (1982) Rough sets. *International Journal of Computer and Information Sciences* 11(5): 341–356.

Pawlak Z (1992) *Rough Sets: Theoretical Aspects of Reasoning About Data*. Dordrecht: Kluwer Academic Publishers.

Pawlak Z (1997) Rough set approach to knowledge-based decision support. *European Journal of Operational Research* 99(1): 48–57.

Pazos F and Sternberg MJ (2004) Automated prediction of protein function and detection of functional sites from structure. *Proceedings of the National Academy of Sciences of the United States of America* 101(41): 14754–14759.

Pilpel Y, Sudarsanam P, and Church GM (2001) Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics* 29(2): 153–159.

Predki B, Słowiński R, Stefanowski J, Susmaga R, and Wilk S (1998) ROSE – Software implementation of the rough set theory. In: Polkowski L and Skowron A (eds.) *Rough Sets and Current Trends in Computing*, vol. 1424, pp. 605–608. Berlin/ Heidelberg: Springer.

Quinlan JR (1986) Induction of decision trees. *Machine Learning* 1(1): 81–106.

Rhee S-Y, Taylor J, Wadhera G, Ben-Hur A, Brutlag DL, and Shafer RW (2006) Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences* 103(46): 17355–17360.

Strömbergsson H, Kryshtafovych A, Prusis P, et al. (2006a) Generalized modeling of enzyme–ligand interactions using proteochemometrics and local protein substructures. *Proteins* 65(3): 568–579.

Strömbergsson H, Prusis P, Midelfart H, Lapinsh M, Wikberg JE, and Komorowski J (2006b) Rough set-based proteochemometrics modeling of G-protein-coupled receptor–ligand interactions. *Proteins* 63(1): 24–34.

Strömbergsson H, Prusis P, Midelfart H, Wikberg JE, and Komorowski J (2004) Proteochemometrics modeling of receptor–ligand interactions using rough sets. In: *Proceedings of the German conference on Bioinformatics,* pp. 4–6.

Subramanian A, Tamayo P, Mootha VK, et al. (2005) Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* 102(43): 15545–15550.

Tilgner H and Guigó R (2010) From chromatin to splicing: RNA-processing as a total artwork. *Epigenetics* 5(3): 180–184.

Tusher VG, Tibshirani R, and Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences* 98(9): 5116–5121.

Wabnik K, Hvidsten TR, Kedzierska A, et al. (2009) Gene expression trends and protein features effectively complement each other in gene function prediction. *Bioinformatics* 25(3): 322–330.

Wang Z, Zang C, Rosenfeld JA, et al. (2008) Combinatorial patterns of histone acetylations and methylations in the human genome. *Nature Genetics* 40(7): 897–903.

Wilczyński B, Hvidsten T, Kryshtafovych A, Tiuryn J, Komorowski J, and Fidelis K (2006) Using local gene expression similarities to discover regulatory binding site modules. *BMC Bioinformatics* 7(1): 505.

Ziarko W (1993) Variable precision rough set model. *Journal of Computer and System Sciences* 46(1): 39–59.