Model Driven Paediatric European Digital Repository

Call identifier: FP7-ICT-2011-9 - **Grant agreement no**: 600932

Thematic Priority: ICT - ICT-2011.5.2: Virtual Physiological Human

Deliverable 16.3

Final Release of KDD & Simulation platform

Due date of delivery: 31-05-2016

Actual submission date: 22-06-2016

Start of the project: 1st March 2013
Ending Date: 31th May 2017

Partner responsible for this deliverable: ATHENA

Version: 2.0

Dissemination Level: Public

Document Classification

Title	Final Release of KDD & Simulation platform
Deliverable	16.3
Reporting Period	
Authors	ATHENA, HES-SO, Siemens Healthcare GmbH, UTVB
Work Package	WP16
Security	Public
Nature	Report
Keyword(s)	Knowledge discovery, Simulation platform

Document History

Name	Remark	Version	Date
MDP_D16.3_v0	Template	0.0	25/05/17
MDP_D16.3_v1.0	First draft	1.0	31/05/17
MDP_D16.3_v1.1	Siemens input	1.1	1/06/17
MDP_D16.3_v1.2	UTVB input	1.2	2/06/17
MDP_D16.3_v1.3	ATHENA updates	1.3	12/06/17
MDP_D16.3_v1.4	Internal review	1.4	20/06/2017
MDP_D16.3_v1.5	Updated after int.review	1.5	22/06/17
MDP_D16.3_v2	Final submitted version	2.0	22/06/17

List of Contributors

Name	Affiliation
Omiros Metaxas	ATHENA
Orfeas Aidonopoulos	ATHENA
Harry Dimitropoulos	ATHENA
Olivier Pauly	Siemens Healthcare GmbH
Lucian Itu	UTBV
Anamaria Vizitiu	UTBV

List of reviewers

Name	Affiliation
Henning Müller	HES-SO

Abbreviations

ANA	Anti-Nuclear Antibodies (blood test)
BN	Bayesian Network
CP	Cerebral Palsy
DAG	Directed Acyclic Graph
DCV	Data Curation and Validation (tool)
DIP	Distal Interphalangeal joints
DISC	Discretised (variable)
ESR	Erythrocyte Sedimentation Rate (blood test)
GPM	Graphical Probabilistic Models
IACI	Intra-Articular Corticosteroid Injections
JIA	Juvenile Idiopathic Arthritis
KDD	Knowledge Discovery and Data Mining / Knowledge Discovery in Databases
MDP	MD-Paedigree
MTX	Methotrexate (drug)
NND	Neurological and Neuromuscular Diseases
NSAIDs	Non-Steroidal Anti-Inflammatory Drugs
PCA	Principal Component Analysis
RF	Rheumatoid Factor
ROC	Receiver operation characteristic curves
SVD	Singular Value Decomposition
TMJ	Temporomandibular Joints
t-SNE	t-distributed Stochastic Neighbour Embedding
UDFs	User Defined Functions

Table of Contents

1. Introduction

The aim of this report is to describe the work carried out in WP16 between the beta and the final release of the Biomedical Knowledge Discovery (KDD) and simulation platform. Section 2, covers the work carried out by ATHENA in developing the final release of the KDD platform (DCV + knowledge discovery extension). Section 3, describes the work carried out by ATHENA with the AITION knowledge discovery tool. Section 4, outlines the work of Siemens on the DeepReasoner web-based prototype for supporting evidence-based decision making. Finally, Section 5 describes the updated method for personalizing the whole-body circulation model used to generate features for the DeepReasoner and is also related to the work performed in WP9.

2. Integration with DCV, interface, front-end tech-specs

As described in D16.2 (Beta Prototype of KDD & Simulation platform), ATHENA started to develop a web-based knowledge discovery platform focusing on biomedical cases. The last year of the project, this platform came to its final release by integrating with the Data Curation and Validation (DCV) tool [D15.2]. The algorithms and methods described in D16.1 (First report on Biomedical knowledge discovery and simulation for model-guided personalized medicine) remain, providing the user with an efficient pipeline: from preliminary analysis to advanced statistics like development of predictive models.

Workflow support: Each analysis step carried out by users is stored in a specific history-area where they can navigate in their history going back at any state they want. Every analysis can be stored as a workflow and hence when a user comes back to the platform s/he will find his/her workflows and rerun them.

2.1.User interface

2.1.1. Preliminary analysis

Although we mainly focused on the development of classification models to help clinicians automate the processes of patient' classification in medical categories, clustering and dimensionality reduction techniques were applied as preliminary steps for each data analysis. Clinicians spend a lot of time to make an informative profile of their datasets in order to then proceed with classifications and predictions. A key-problem they address is the well-known 'curse of dimensionality' where there is a large number of variables that produce a high-dimensional feature space. As the dimensions of the space increase, the predictive power of a model decreases, since it has to cope with much more noise and there usually aren't enough observations to get good estimates. Moreover, although most of the variables are very informative, the computational time to analyse them and gain information through a predictive model is usually prohibitive. Therefore, dimensionality reduction strategies are used to identify strong correlations among the variables and project them into a new lower-dimensional space. Especially, Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) integrated in the DCV extension are also used to visualize high-dimensional data. Later, t-distributed Stochastic Neighbour Embedding (t-SNE) was added in order to handle sparse data [1]. Clustering methods that were already added to the DCV KDD extension are also combined with dimensionality reduction for better visualization.

In Figure 1, a clustering example of the Juvenile Idiopathic Arthritis (JIA) knowledge discovery use-case is shown. Having patient cohorts derived from three main datasets (clinical measurements, measurements of biological factors by Luminex assays, and microbiota phyla categories) we aimed to develop a computational model that predicts short and long-term patient outcomes referring to their activity regarding the JIA disease. Such a case requires a preliminary analysis to first understand how each dataset affects the other. Following this, we proceeded to the best integration of the three datasets and made our training sets for the predictive model. Thus, we applied a k-means clustering (with a k=2, as we have a binary problem herein: active/inactive disease) simultaneously using a principal component analysis for the visualization of clusters in two-dimensional space (Figure 1). We observed that the enrichment of the clinical model with Luminex and microbiota variables can at least separate the patients into two clusters while the clinical set on its own fails to do so.

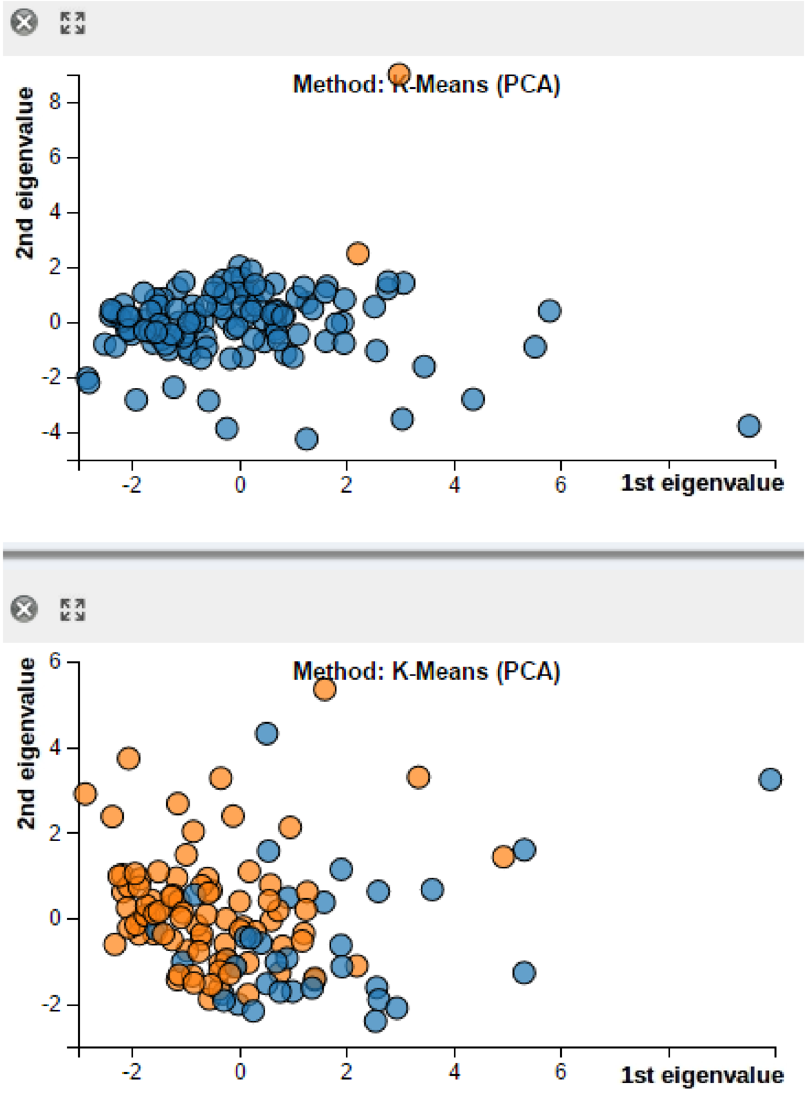
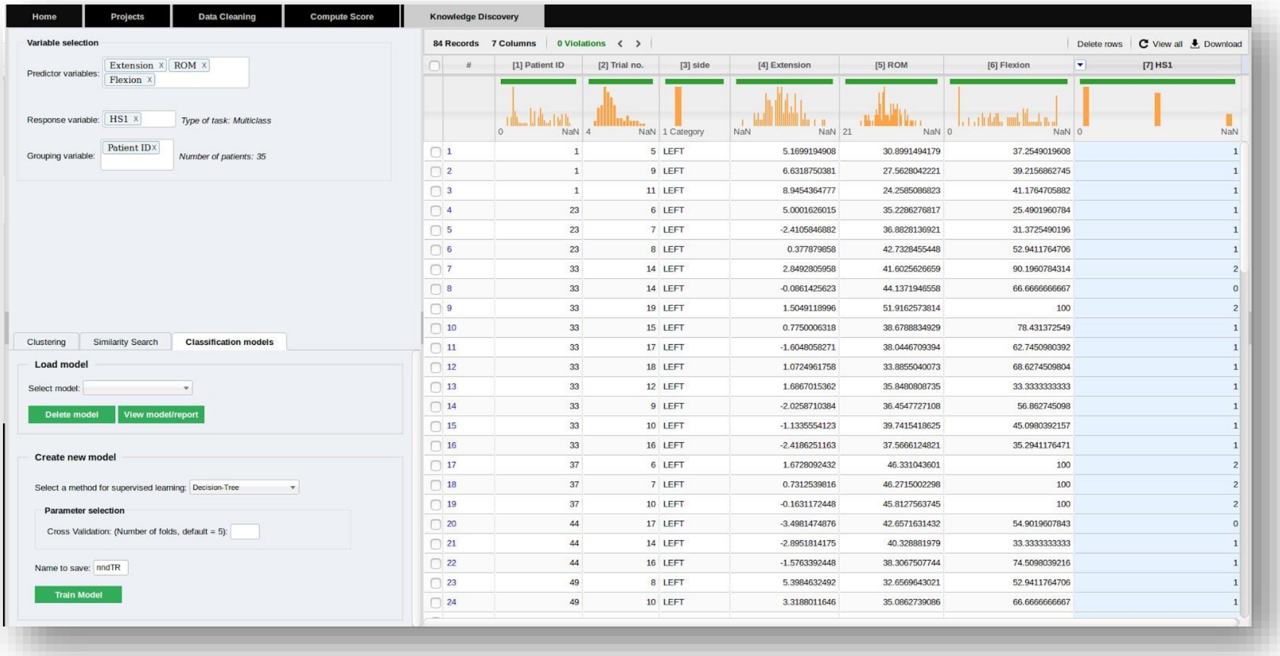


Figure 1: Clustering on clinical datasets VS mixed datasets: Patients could be separated in two clusters when enriching the clinical dataset with the Luminex and Microbiota sets.

A user-friendly interface was also designed following the prerequisites of the DCV interface to provide users with the steps, variables and parameters they want to add into the analysis (Figure 2). The interface consists of JavaScript, css and html5 scripts including four external libraries: *Jquery* [2] and *w2ui* [3] for the main interface fields and components, *slickgrid.js* [4] for the data representation in an interactive data-grid and finally *D3* [5] or any visualizations. Code snippets were written in JavaScript code in order to allow D3 to "communicate" with slickgrid. Thus, interactions between the data-table and the visualizations are available for exploring patient data through visualizations (Figure 3).



Variable selection

Predictor variables: Extension x ROM x Flexion x

Response variable: HS1 x *Type of task: Multiclass*

Grouping variable: Patient IDx *Number of patients: 35*

Load model

Select model:

Create new model

Select a method for supervised learning: Decision-Tree

Parameter selection

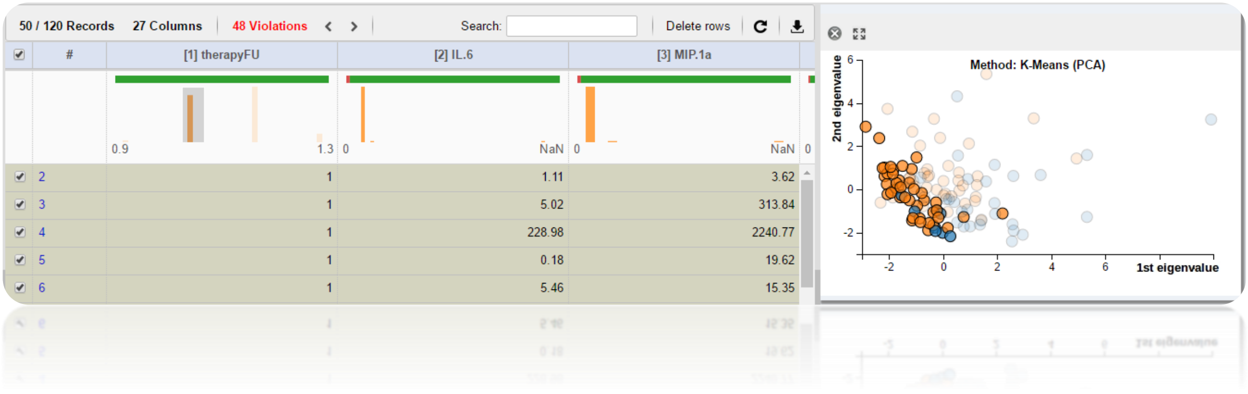
Cross Validation: (Number of folds, default = 5):

Name to save: nndTR

84 Records 7 Columns 0 Violations

#	[1] Patient ID	[2] Trial no.	[3] side	[4] Extension	[5] ROM	[6] Flexion	[7] HS1
□ 1	1	5	LEFT	5.1699194908	30.8991494179	37.2549019608	1
□ 2	1	9	LEFT	6.6318750381	27.5628042221	39.2156862745	1
□ 3	1	11	LEFT	8.9454364777	24.2585086823	41.1764705882	1
□ 4	23	6	LEFT	5.0001626015	35.2286276817	25.4901960784	1
□ 5	23	7	LEFT	-2.4105846882	36.8828136921	31.3725490196	1
□ 6	23	8	LEFT	0.377879858	42.7328455448	52.9411764706	1
□ 7	33	14	LEFT	2.8492805958	41.6025626659	90.1960784314	2
□ 8	33	14	LEFT	-0.0861425623	44.1371946558	66.6666666667	0
□ 9	33	19	LEFT	1.5049118996	51.9162573814	100	2
□ 10	33	15	LEFT	0.7750006318	38.6788834929	78.431372549	1
□ 11	33	17	LEFT	-1.6048058271	38.0446709394	62.7450980392	1
□ 12	33	18	LEFT	1.0724961758	33.8855040073	68.6274509804	1
□ 13	33	12	LEFT	1.6867015362	35.8480808735	33.3333333333	1
□ 14	33	9	LEFT	-2.0258710384	36.4547727108	56.862745098	1
□ 15	33	10	LEFT	-1.1335554123	39.7415418625	45.0980392157	1
□ 16	33	16	LEFT	-2.4186251163	37.5666124821	35.2941176471	1
□ 17	37	6	LEFT	1.6728092432	46.331043601	100	2
□ 18	37	7	LEFT	0.7312539816	46.2715002298	100	2
□ 19	37	10	LEFT	-0.1631172448	45.8127563745	100	2
□ 20	44	17	LEFT	-3.4981474876	42.6571631432	54.9019607843	0
□ 21	44	14	LEFT	-2.8951814175	40.328881979	33.3333333333	1
□ 22	44	16	LEFT	-1.5763392448	38.3067507744	74.5098039216	1
□ 23	49	8	LEFT	5.3984632492	32.6569643021	52.9411764706	1
□ 24	49	10	LEFT	3.3188011646	35.0862739086	66.6666666667	1

Figure 2: DCV knowledge discovery extension user interface.



50 / 120 Records 27 Columns 48 Violations

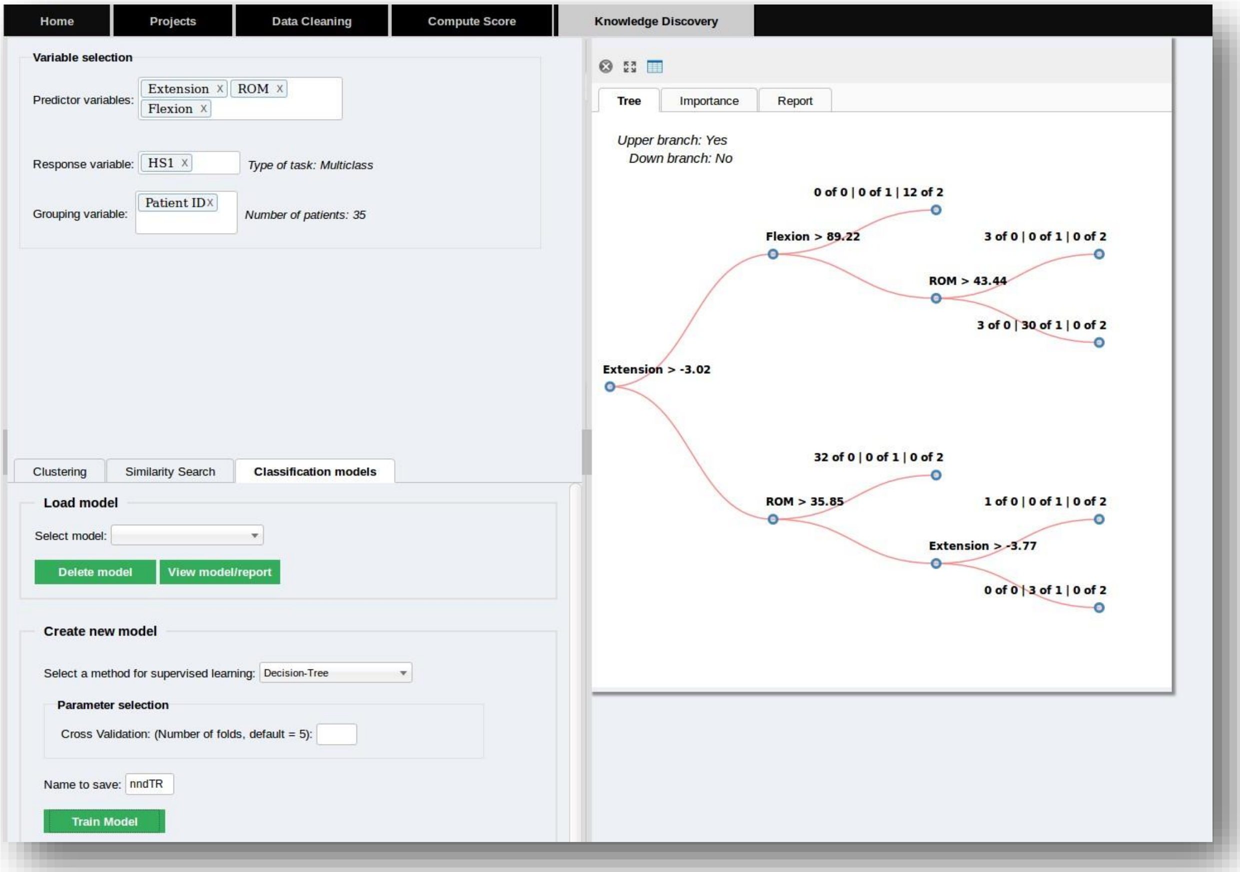
#	[1] therapyFU	[2] IL.6	[3] MIP.1a
☑ 2	1	1.11	3.62
☑ 3	1	5.02	313.84
☑ 4	1	228.98	2240.77
☑ 5	1	0.18	19.62
☑ 6	1	5.46	15.35

Figure 3: Interactive mode of the platform. A k-means clustering was applied on the JIA multivariable dataset and then the user selects a range of interest from the therapy column to explore how the distribution differs between the two clusters. Here, the clinician observes the statistical behaviour of patients receiving aggressive therapy (therapyFU=1 selected as range of interest).

2.1.2. Development of prediction models

Similarity and classification algorithms were already available in the KDD extension since February 2016 [D16.2]. However, a new container was added to visualize the KDD results. Especially for the prediction models, a detailed report is provided to users with both results and assessment metrics for each of the model trainings. Thus, the platform provides clinicians with a user-friendly environment where they can build models and assess them on their own (Figure 4). An optimized version of the C4.5 algorithm [6] is used for implementing "Decision Trees" for classification and regression tasks. The user is able to also improve his/her predictions through "Random Forests" that are constructed by an ensemble of decision trees. Decision tree learning is a 'white-box' strategy, as it provides the rules based on which the classification was done. Clinicians found this approach very useful (Figure 5) since they can compare these rules with the ones derived from their clinical model.

Moreover, there is the option to select a grouping variable in order to optimize the validation phase during the construction of the models. Selecting a grouping variable indicates that the cross-validation (CV) technique is used to split the data in training and test sets where each patient cohort of visits can never be separated into different splits. For instance, patient '23' has visited the hospital multiple times. His/her visits will always be in the same set during the CV procedure. Finally, each trained model is stored so it can be loaded to predict the outcome of new patient cases later.



Variable selection

Predictor variables: Extension x ROM x Flexion x

Response variable: HS1 x *Type of task: Multiclass*

Grouping variable: Patient ID x *Number of patients: 35*

Upper branch: Yes
Down branch: No

Load model

Select model:

Create new model

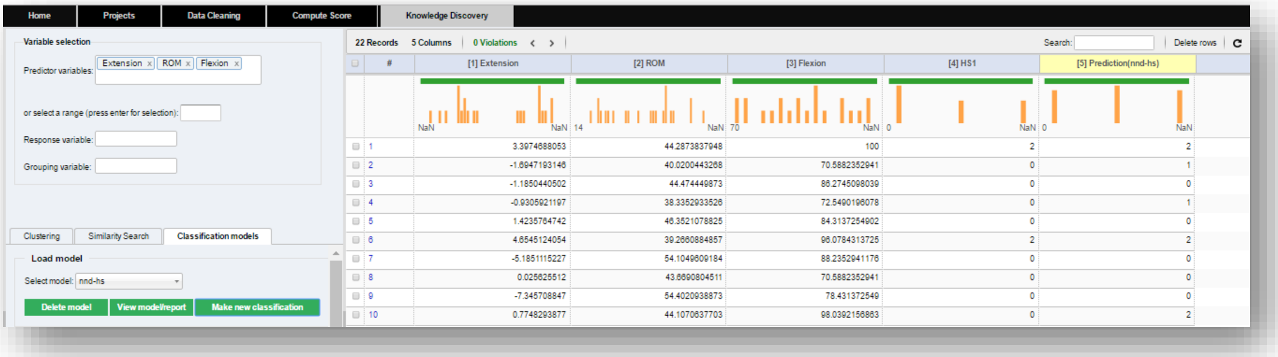
Select a method for supervised learning: Decision-Tree

Parameter selection

Cross Validation: (Number of folds, default = 5):

Name to save: nndTR

Figure 4: The platform gives a comprehensive view to the clinicians. Example from the NND use-case. Three predictors were selected and the Patient_ID column as a grouping variable. The model was trained and detailed classification rules are provided for each classification. The clinician is thus able to compare the model's rules with his/her own clinically designed rules.



Variable selection

Predictor variables: Extension x ROM x Flexion x

or select a range (press enter for selection):

Response variable:

Grouping variable:

Load model

Select model: nnd-hs

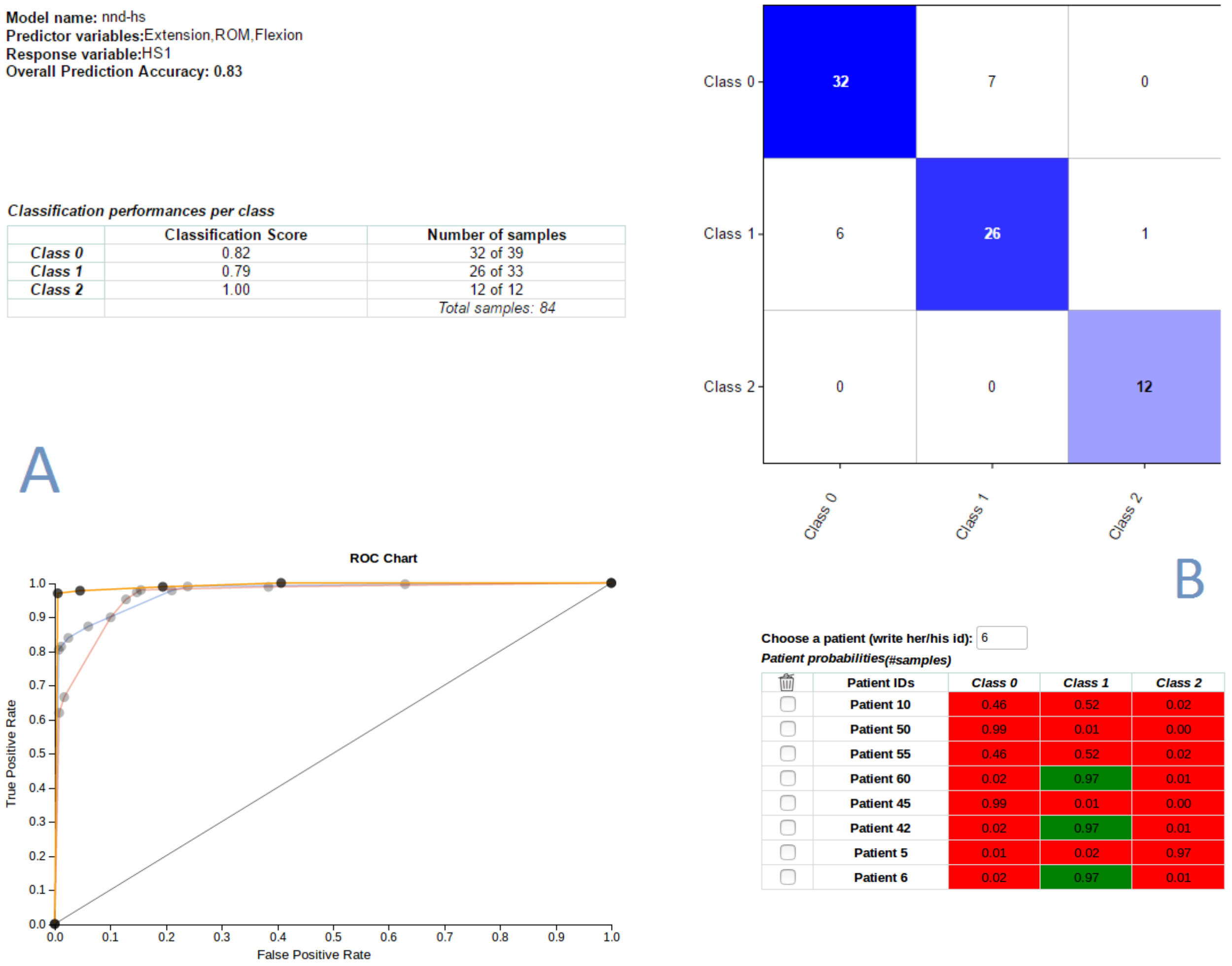
22 Records 5 Columns | 0 Violations

#	[1] Extension	[2] ROM	[3] Flexion	[4] HS1	[5] Prediction(nnd-hs)
1	3.3974688053	44.2873837948	100	2	2
2	-1.6947193146	40.0200443268	70.5882352941	0	1
3	-1.1850440502	44.474449873	86.2745098039	0	0
4	-0.9305921197	38.3352933526	72.5490196078	0	1
5	1.4235764742	46.3521078825	84.3137254902	0	0
6	4.6545124054	39.2660884857	96.0784313725	2	2
7	-5.1851115227	54.1049609184	88.2352941176	0	0
8	0.025625512	43.6690804511	70.5882352941	0	0
9	-7.345708847	54.4020938873	78.431372549	0	0
10	0.7748293877	44.1070637703	98.0392156863	0	2

Figure 5: The user selects a trained model from the database to classify new follow-up trials or entirely new patient cohorts.

2.1.3. Model assessment

A comprehensive report is produced for each training. Apart from the visualization of the classification-tree, classification performances, confusion matrices and ROC charts are provided to the user. The tool in addition offers the user the choice to search for a patient and view the posterior probabilities for belonging to each class separately. An optimal threshold derived by ROC analysis is applied so the clinician can also ask for the classifier's "trust" in the classification. In the case where the probabilities are below the threshold, the prediction is highlighted in red to indicate that is not to be trusted. For instance, in Figure 6B, Patient 10 is classified into Class1 with a 52% probability. Apparently, it is not a trustable classification, while patient 60 was classified with a certainty level of 97%. This specific feature was added on request of the NND clinicians, as they also use a confidence level from 1 to 5 for deciding whether they should proceed with a second follow-up classification. Users can also select a specific threshold from the ROC chart to change the threshold levels on the patient-based probabilities heat-map.



Model name: nnd-hs
Predictor variables: Extension,ROM,Flexion
Response variable:HS1
Overall Prediction Accuracy: 0.83

Classification performances per class

	Classification Score	Number of samples
Class 0	0.82	32 of 39
Class 1	0.79	26 of 33
Class 2	1.00	12 of 12
		Total samples: 84

	Class 0	Class 1	Class 2
Class 0	**32**	7	0
Class 1	6	**26**	1
Class 2	0	0	**12**

Choose a patient (write her/his id): 6

Patient probabilities(#samples)

🗑	Patient IDs	*Class 0*	*Class 1*	*Class 2*
☐	Patient 10	0.46	0.52	0.02
☐	Patient 50	0.99	0.01	0.00
☐	Patient 55	0.46	0.52	0.02
☐	Patient 60	0.02	0.97	0.01
☐	Patient 45	0.99	0.01	0.00
☐	Patient 42	0.02	0.97	0.01
☐	Patient 5	0.01	0.02	0.97
☐	Patient 6	0.02	0.97	0.01

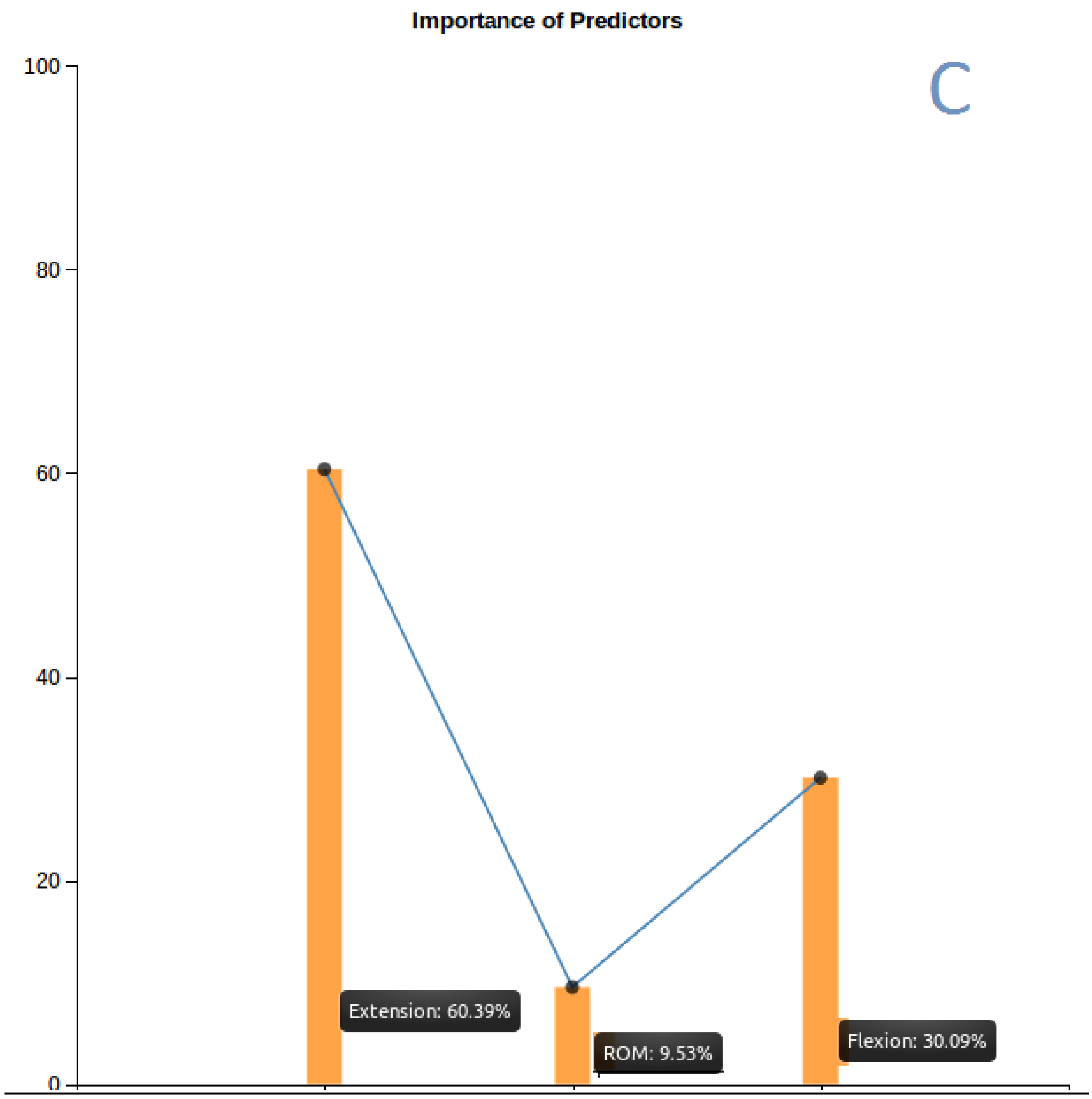


Figure 6: A comprehensive report is provided to users in order to be able to assess models based on their preferences: A) Overall mean accuracy, detailed classification performances for each class and confusion matrix; B) ROC curves and patients' posterior probabilities highlighted in red or green depending on a selected cut-off point from the ROC; C) Feature importance: how much each predictor contributes to the measured overall classification accuracy.

2.2.Back-end functionalities and architecture

The main architecture of MDP's infrastructure extended DCV module is illustrated in Figure 7. The user interface described previously refers to the *front-end* section while the *back-end* consists of a worker, named *madIS*, enhanced with extra python modules for the knowledge discovery extension. The open source python library, *scikit-learn* [7] was incorporated that contains many well-established machine learning algorithms and techniques implemented in python libraries. These libraries can be imported on top of *EXAREME* easily [8] as specific User-Defined Functions (UDFs) of the madIS system [9]. Such UDFs, construct predictive or clustering models, estimate new values for unlabelled data, and categorize new data samples [D16.1]. An example of a madIS query for clustering is the following:

```
select skcluster(init, cols) from table t;
```

where:

- *skcluster* is the madIS UDF for clustering
- *init* is the initialization string and the desired parameters of any desired algorithm provided by scikit-learn.
  i.e.: “KMeans(n_clusters=3)” for k-means algorithm or “DBSCAN(*eps=0.5*, *min_samples=5*)” for Density-Based Spatial Clustering.
- *cols* are the attributes participating in clustering

In Figure 8 one can see the results of such a query. In this illustration, it is also depicted that in case of centroid-algorithms the cluster centres are returned too.

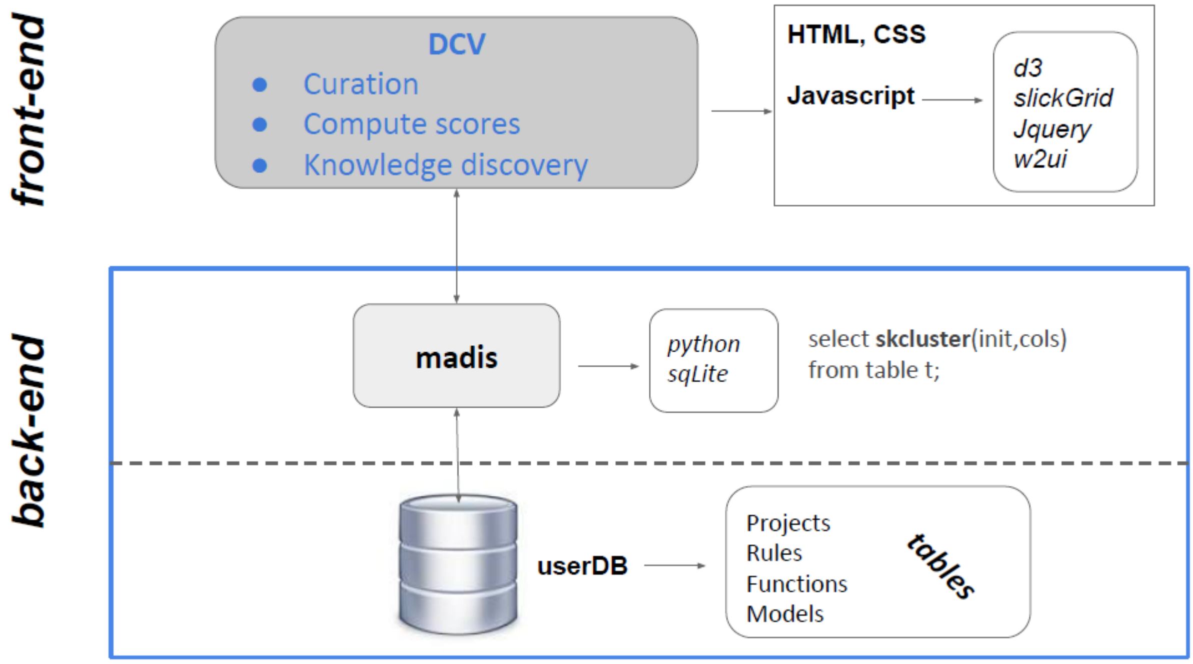
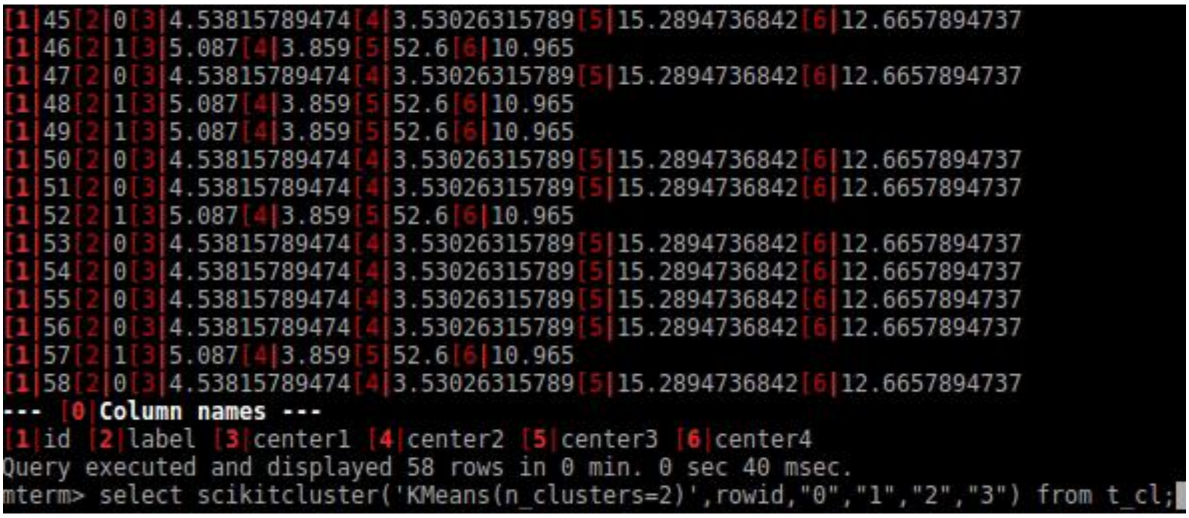


Figure 7: Description of the extended DCV architecture.



```
[1|45[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|46[2|1[3|5.087[4|3.859[5|52.6[6|10.965
[1|47[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|48[2|1[3|5.087[4|3.859[5|52.6[6|10.965
[1|49[2|1[3|5.087[4|3.859[5|52.6[6|10.965
[1|50[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|51[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|52[2|1[3|5.087[4|3.859[5|52.6[6|10.965
[1|53[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|54[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|55[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|56[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
[1|57[2|1[3|5.087[4|3.859[5|52.6[6|10.965
[1|58[2|0[3|4.53815789474[4|3.53026315789[5|15.2894736842[6|12.6657894737
--- [0|Column names ---
[1|id [2|label [3|center1 [4|center2 [5|center3 [6|center4
Query executed and displayed 58 rows in 0 min. 0 sec 40 msec.
mterm> select scikitcluster('KMeans(n_clusters=2)',rowid,"0","1","2","3") from t_cl;
```

Figure 8: K-means clustering of samples consisting of 4 variables (columns). Result (appearing above the query) depicts how the last 14 samples are clustered (1st column: id, 2nd: cluster, and the remaining columns are the centroids).

2.3.Knowledge Discovery use-cases and significant results

The integrated knowledge discovery methods were applied on two clinical use-cases with promising results. In particular, an automated classification system was developed and assessed to classify gait-movement joint patterns for patients from the Neurological and Neuromuscular Diseases (NND) group. All models achieve high prediction accuracy (above 80%) for each pattern. We also addressed the challenging approach of fitting a heterogeneous dataset from the Juvenile Idiopathic Arthritis (JIA) group on a prediction model.

2.3.1. Neurological and neuromuscular diseases (NND)

One of the primary needs of clinicians is a classification that characterizes each Cerebral Palsy (CP) gait by different degrees of membership for several gait patterns. Machine learning techniques fit very well such kind of problems. Predictive models can be created in order to automate the CP gait classification. Thus, our aim was to train classification models, based on rules developed by clinicians, and identify their prediction accuracy. Models with high accuracy (>80%) can be considered reliable in classifying new patients and can also predict possible future behaviour in their movement. Thus, having a classified dataset of 356 children with 1561 gait trials, we trained Random Forests and k-Nearest Neighbour models for the above-mentioned classification tasks. Random forests were used due to their “white-box” approach which fits very well in gait-classification, as the clinicians can explore and see the rules based on which the model classified each sample, and hence compare with their own rules. K-NN is used to classify the patients based on their similarities.

Both methods were applied on datasets where the predictor variables correspond to kinematic parameters extracted from the gait waveforms by the clinicians. After training and assessment, our classifiers proved to be reliable enough to classify each of the patterns successfully (overall accuracies>=85%). However, the number of parameters used as predictor variables in the models were too few (two to five at most) for trusting those results. Therefore, we also trained the classifiers directly on the waveform data in order to validate the good results. Kinematic waveforms consist of 50 intervals of a gait-cycle (initial contact of the foot to the ground until one step is completed). Thus, a large number of predictors is available; wavelet decomposition was applied to reduce the dimensions and avoid possible overfitting. The results are illustrated in Figure 9. The classifiers trained on extracted parameters outperformed the ones trained on waveforms. Nevertheless, waveform-based model accuracies still remained above 80% validating our analysis. Clinicians were trained on how to use the platform on their own and after were able to try more trainings on different variations; the classification performance was always high.

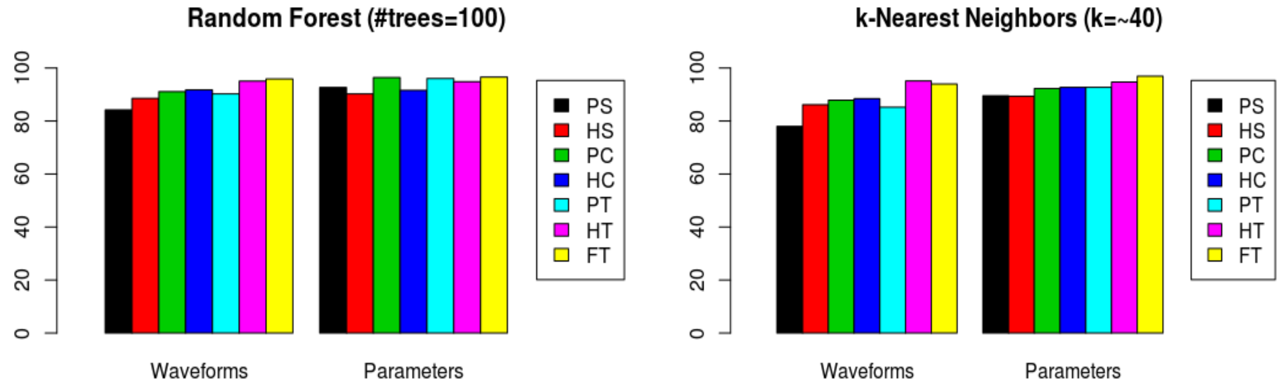
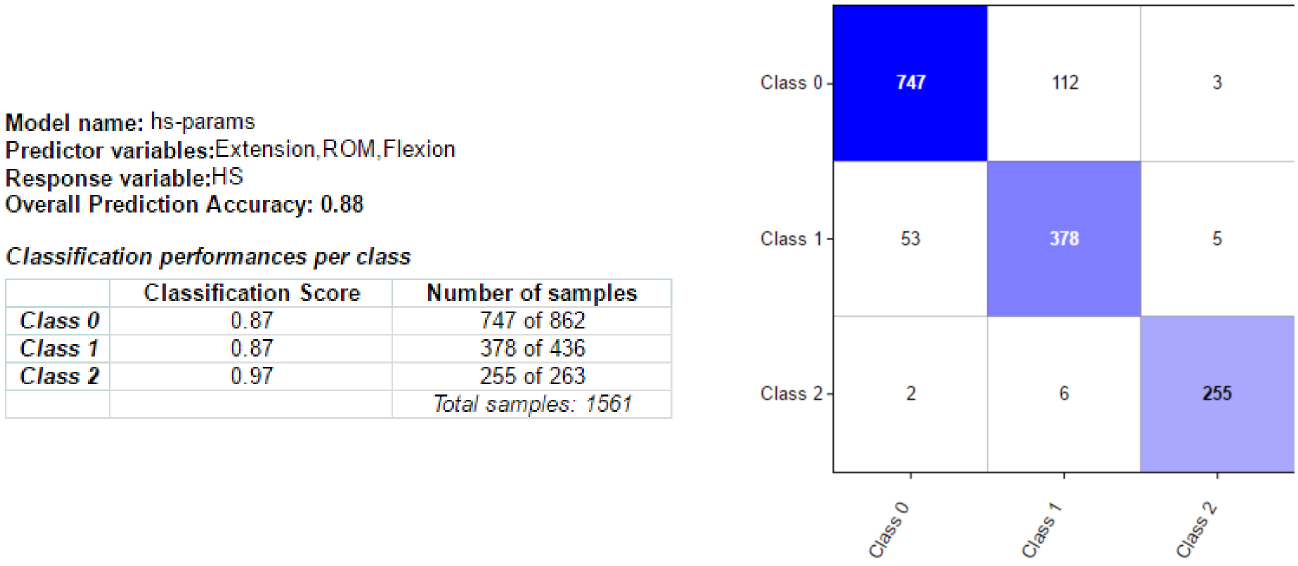


Figure 9: NND results: Random forest and k-NN algorithms used to train predictive models on kinematic waveforms and extracted parameters. PS: pelvis in sagittal plane, HS: hip in sagittal plane, PC: pelvis in coronal plane, HC: hip in coronal plane, PT: pelvis in transverse plane, HT: hip in transverse plane and FT: foot in transverse plane.

Looking deeper into the trained models, two screenshots from the DCV analytics report follow (Figure 10 & Figure 11). A classification model was trained on kinematic data measured for the hip joint in sagittal plain. The measured values refer to three parameters: Extension of hip, its range of motion (measurements of hip angle degree during one complete movement cycle), and its flexion (percentage of the gait cycle participating in flexion). As said before, these parameters are essentially the predictor variables of the model. As a response variable, the multiclass variable selected consisted of three classes: 0 for normal hip motion, 1 for hip extension deficit, and 2 for continuous excessive hip flexion. The overall mean prediction accuracy was 88% and the model was very discriminative. The percentages of true positives are very high as one can see from the confusion matrix and the area under the ROC curve.



Model name: hs-params
Predictor variables: Extension,ROM,Flexion
Response variable:HS
Overall Prediction Accuracy: 0.88

Classification performances per class

	Classification Score	Number of samples
Class 0	0.87	747 of 862
Class 1	0.87	378 of 436
Class 2	0.97	255 of 263
		Total samples: 1561

	Class 0	Class 1	Class 2
Class 0	747	112	3
Class 1	53	378	5
Class 2	2	6	255

Figure 10: Classification performance per class (left) and distribution of classified samples in a confusion matrix (right).

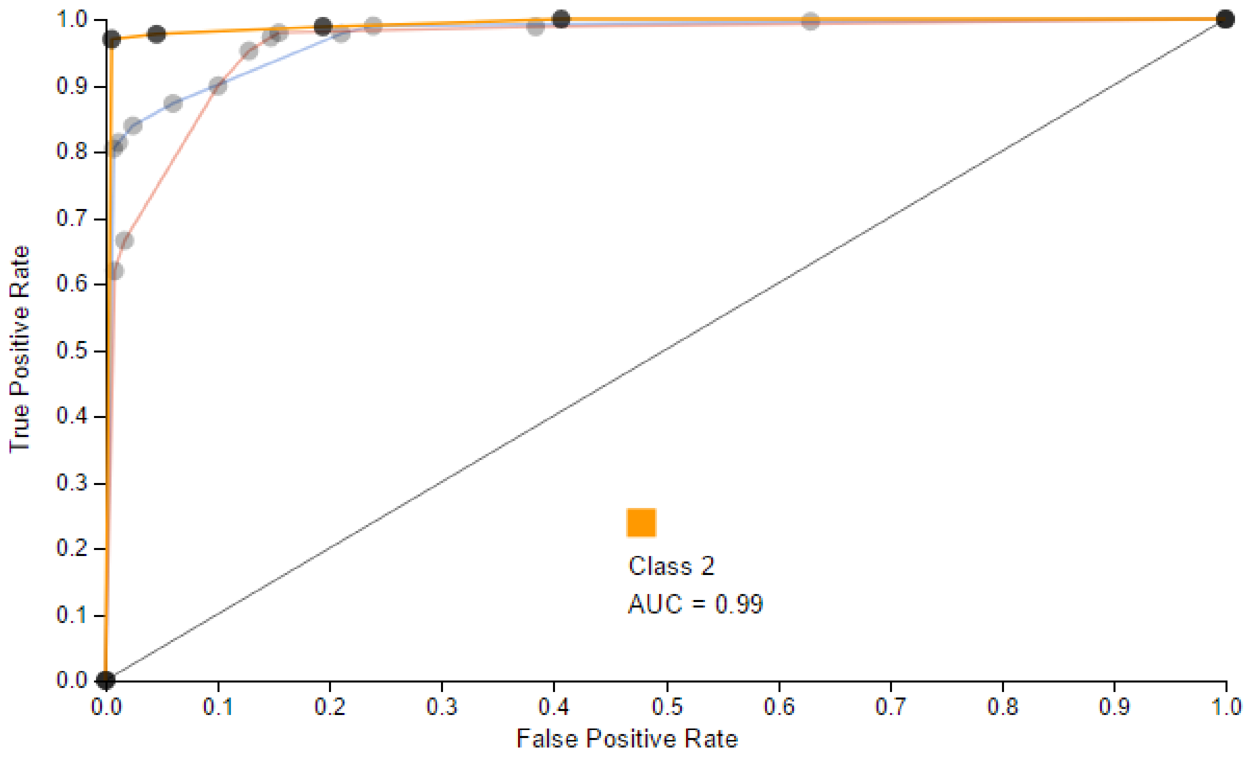


Figure 11: Receiver operation characteristic curves (ROC) for each class-pattern separately: plot of the true positive rate against the false positive rate for different possible cut-off points indicating the trade-off between sensitivity and specificity. Accuracy is measured by the area under the ROC curve (i.e. 99% of patients with deficit extension are correctly classified by the model).

In conclusion, the prediction models developed herein can be integrated with the classification procedure that clinicians follow. The models were also revised by clinicians. They were able to use them on their own and we are now working on incorporating the whole platform and the related produced models into the regular clinical validation process.

2.3.2. Juvenile Idiopathic Arthritis (JIA)

The second use-case on which DCV was applied on was designed to determine early in the course of JIA who needs more aggressive therapy. Using the same methods as for NND, we tried to fit efficient classification models in order to predict the disease outcome (active/inactive) using only baseline variables. This was a challenging task since we essentially tried to predict short and long-term outcomes (from 6 to 12 months) without having available follow-up visits. The dataset consisted of 151 patients with many variables from three heterogeneous datasets: clinical measurements, Luminex factors and microbiota phyla categories, which were to be concatenated and be fed to the models as one training set. Integrating the three datasets into one created a difficult problem due to the very large number of dimensions. In addition, several patient-cases were removed due to missing samples for Luminex and microbiota datasets. Hence, the amount of training cases was reduced after concatenation. For instance, from the available 151 patients some of them were missing Luminex measurements and some others microbiota measurements. After applying different methods of dimensionality reduction and normalization, the first results looked promising, especially for the 12 months' outcome. However, the diagram shown in Figure 12 refers to models trained without validation. Consequently, we applied a 10-fold cross-validation to assess the models only on test cases. The area under the ROC curve that was used for the assessment was below 70% and so it was incomparable to the training accuracies.

As a second trial, we then fitted three models: one clinical plus ultrasound, one clinical plus microbiota and one clinical plus Luminex, in order to see which of these performed best and, especially, if the Luminex and microbiota models would perform better than the clinical one (something shown also from the preliminary analysis via clustering - Figure 1). Our aim here was to predict only the 6 and 12 months' outcome as for the next two (18 and 24) the number of cases was dramatically small. In case of good performances, a deep feature selection process would be applied simultaneously with the training phase. Unfortunately, all models once again achieve poor performance and so no trustable predictors were selected from these datasets. Results in Figure 13 refer to the 6 month prediction case and one can easily observe the poor performance. Looking at the classification tables, we saw that the models were mainly unsuccessful is identifying patients who remained active.

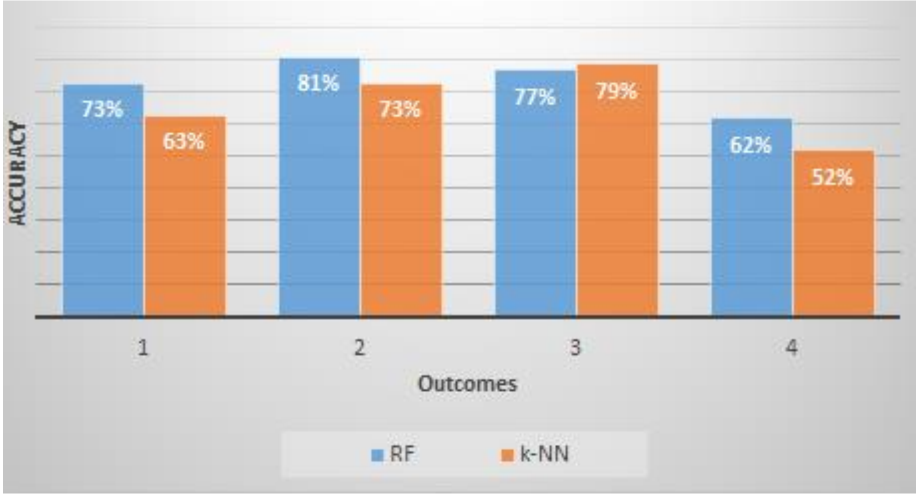
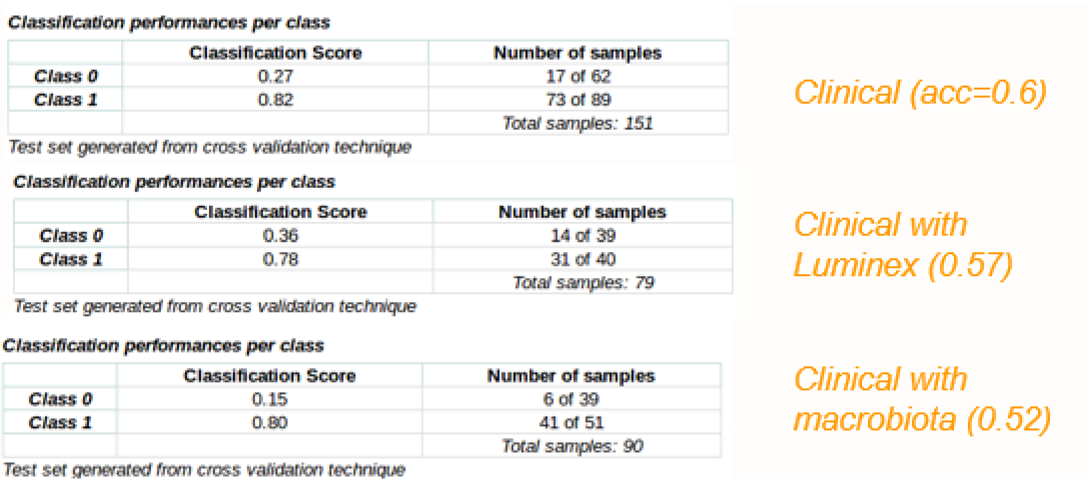


Figure 12: Overall mean accuracies for predicting JIA disease outcome on merged dataset for four future periods: 1) for 6 months - 77 patients, 2) for 12m - 66 patients, 3) for 18m - 46 patients, and 4) for 24m - 27 patients. No validation technique applied.



Classification performances per class

	Classification Score	Number of samples
Class 0	0.27	17 of 62
Class 1	0.82	73 of 89
		Total samples: 151

Test set generated from cross validation technique

Clinical (acc=0.6)

Classification performances per class

	Classification Score	Number of samples
Class 0	0.36	14 of 39
Class 1	0.78	31 of 40
		Total samples: 79

Test set generated from cross validation technique

Clinical with Luminex (0.57)

Classification performances per class

	Classification Score	Number of samples
Class 0	0.15	6 of 39
Class 1	0.80	41 of 51
		Total samples: 90

Test set generated from cross validation technique

Clinical with macrobiota (0.52)

Figure 13: The models re-performed on separated datasets and not in one concatenated like in Figure 12. In this step, cross-validation also applied to assess the models on test data. Poor performances were achieved by the classifiers tending to classify the majority of them as inactive. Class 0 refers to patients with active disease, class 1 to inactive patients.

3. AITION tool

The AITION desk tool, which was initially developed as part of the Health-e-Child ICT-FP6 project (#027749), provides feature selection, variable correlation analysis, statistical simulation modelling and reasoning based on Graphical Probabilistic Models (GPMs). GPMs are a popular and well-studied framework for compact representation of a joint probability distribution over a large number of interdependent variables, as well as for efficient reasoning about such a distribution [10,11].

In MD-Paedigree, new algorithms for Bayesian Network structure learning were implemented and tested following a Service Oriented Architecture, extending already existing algorithms in AITION. In particular, AITION desk implements state-of-the-art algorithms and techniques (exact [12-14] or approximate [15-19]) for Bayesian Network (BN) Structure and Parameter Learning, Markov Blanket induction and feature selection, and real-time inference.

Furthermore, ontologies and a-priori knowledge can be incorporated with the BN, defining topological constraints, in order to automate causal discovery and feature selection and provide semantic modelling under uncertainty [20]. This way, AITION presents a rich 'natural' framework for imposing structure and prior knowledge, providing the domain expert with the ability to seed the learning algorithm with knowledge about the problem at hand.

In the following section, we will describe how the AITION desk tool was used for analysis of JIA data, working on discretized variables utilizing both approximate and exact Bayesian Network structure learning algorithms.

3.1.Juvenile Idiopathic Arthritis (JIA)

Our focus has been on outcome prediction and early prognosis trying to identify disease signatures (factors and associations among specific sets of variables) that better predict short-term and long-term outcomes. In addition, based on the identified Directed Acyclic Graph (DAG), we have built specific Bayesian Network models for statistical simulation that can be used for precision medicine tasks, targeted predictions and "what-if" analysis.

In particular, we used the AITION tool on (i) a dataset of 60 patients consisting of 20 clinical, 4 Luminex, 5 microbiota variables, and on (ii) a dataset of 151 patients with just the 20 clinical variables. One dichotomised variable was used as an outcome variable in both sets. For these experiments, we used only variables that had already been selected by feature selection in collaboration with JIA clinicians at a previous step. The reason that the former dataset has fewer patients is that we had to exclude patients for which there were null values that could not be imputed reliably. In addition, we were instructed to exclude OPBG from the first set, since there was an issue with their Luminex measurements.

Variables were discretized using the following rules, based on known cut-off values taken from the literature or clinical practice. We also made sure that, for each variable, the discretisation used would allow for at least a few patient samples falling into each discrete category (e.g., for the "Age at Onset" variable, since we only had one baby in the sample, we had to merge the "Baby" and "Toddler" cases into one discrete value of "Baby/Toddler"). For variables for which appropriate discretisation cut-off values were not known, we decided to use quartiles.

Discretisation Rules:

GENDER: ifthenelse(gender=1,"1_Male","2_Female"ageonset)
AGEATONSET:
ifthenelse(ageonset<3,"1_BabyToddler",ifthenelse(ageonset<5,"2_Preschooler",ifthenelse(ageonset<12,"3_Gradeschooler",ifthenelse(ageonset<18,"4_Teen","6_Adult"))))
DISEASEDURATION:
ifthenelse(disdur<0.2,"1_0.1",ifthenelse(disdur<0.3,"2_0.2",ifthenelse(disdur<0.4,"3_0.3",ifthenelse(disdur<0.5,"4_0.4","5_0.5+"))))
ESR: ifthenelse(esr<15,"1_Normal",ifthenelse(esr<22,"2_Low",ifthenelse(esr<30,"3_Medium","4_High")))
PGA: ifthenelse(pga<2,"1_<2",ifthenelse(pga<4,"2_<4",ifthenelse(pga<6,"3_<6",ifthenelse(pga<8,"4_<8","5_High"))))
NEUTRO: ifthenelse(neutro<8,"1_Normal","2_High")

For ALL Joints discretise as follows:

JOINT: ifthenelse(joint=0,”1_Inactive”,”1_Active”)
Where joint is the column name/variable.
Leave KNEE as is for all clinical (i.e. 0,1,2)

Quartiles used for remaing variables.

Three network structure algorithms were used, combining the results in order to build a model as robust as possible. The first dataset, consisting of all types of JIA variables, did not produce very good results, explained by the low number of patient samples.

The following heat maps show the most important discovered dependencies between all variables (Figure 14) and between the clinical variables only (Figure 15), for the two datasets respectively.

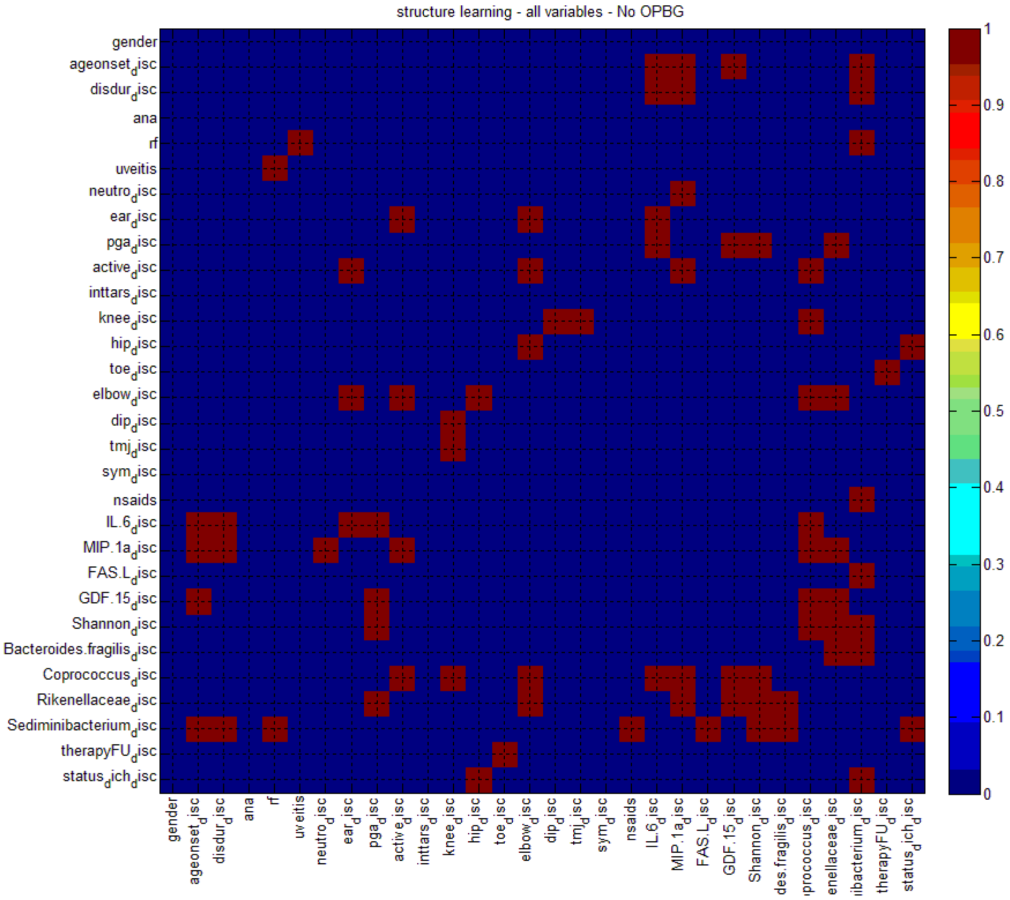


Figure 14: Heat map dependency matrix depicting most important dependencies in JIA clinical+Luminx+microbiota dataset.

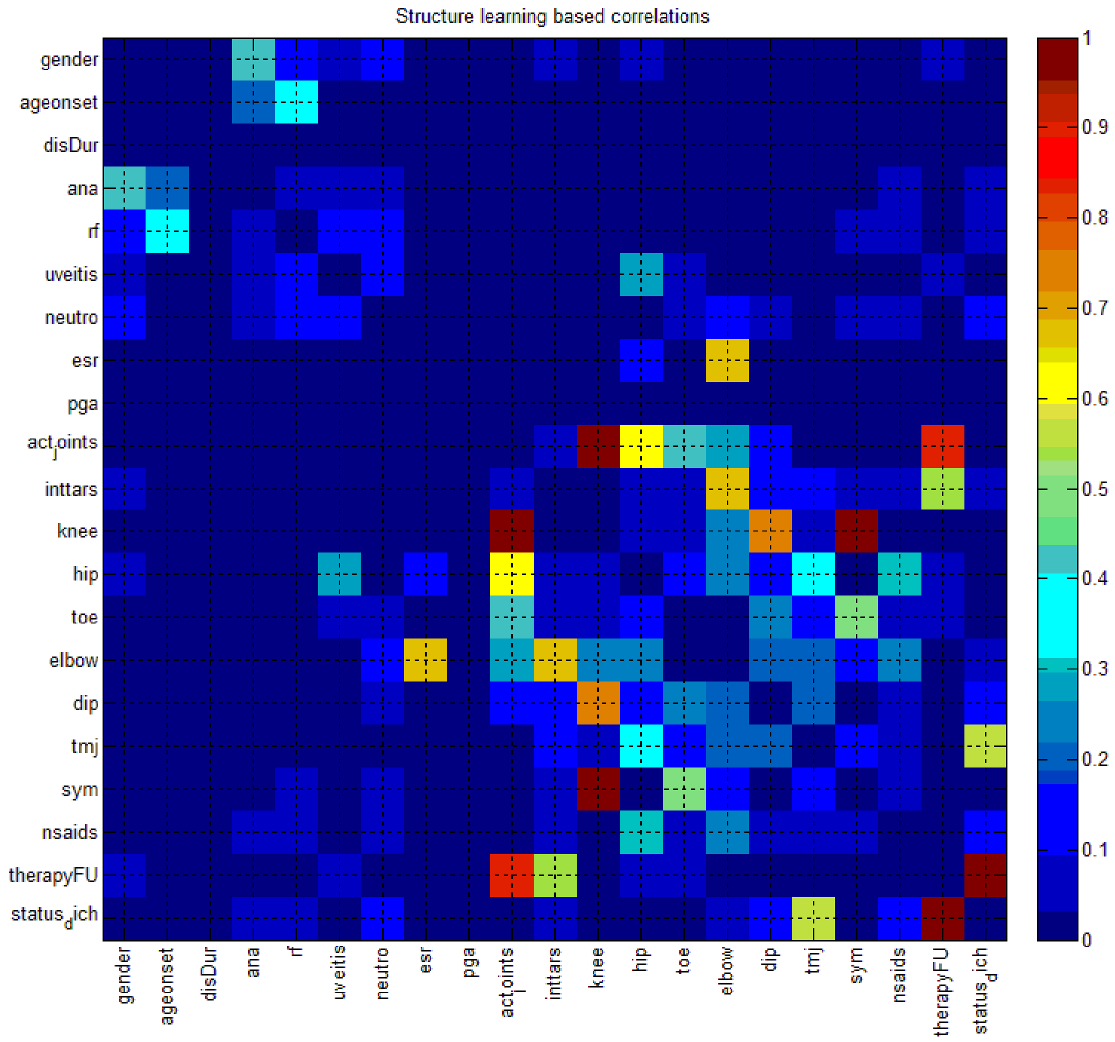


Figure 15: Heat map dependency matrix depicting discovered dependencies in JIA dataset of clinical variables only.

Based on the above-mentioned findings, AITION generated the most probable Directed Acyclic Graph (DAG) structure that was then used for quantitative analysis, as shown in the AITION screenshots that follow. In Figure 16, the AITION desk tool is shown with the JIA model loaded ready for "Inference" (reasoning). The four variables that appear as grey boxes are variables that AITION found were almost uncorrelated, and were excluded from the final DAG.

AITION can also perform a sensitivity analysis based on a selected variable. In the example depicted in Figure 17, a sensitivity analysis is carried out based on the "dichotomised output" variable (at the top of the graph). The length of the red bars, appearing on top of each variable box/node, give an indication of relative influence of each variable on "Outcome" (the selected variable).

Abbreviations: *ANA, antinuclear antibodies; DIP, distal interphalangeal joints; DISC, discretized; ESR, erythrocyte sedimentation rate; GA, global assessment; IACI, intra-articular corticosteroid injections; MTX, methotrexate; NSAIDs, non-steroidal anti-inflammatory drugs; RF, rheumatoid factor; TMJ, temporomandibular joints.*

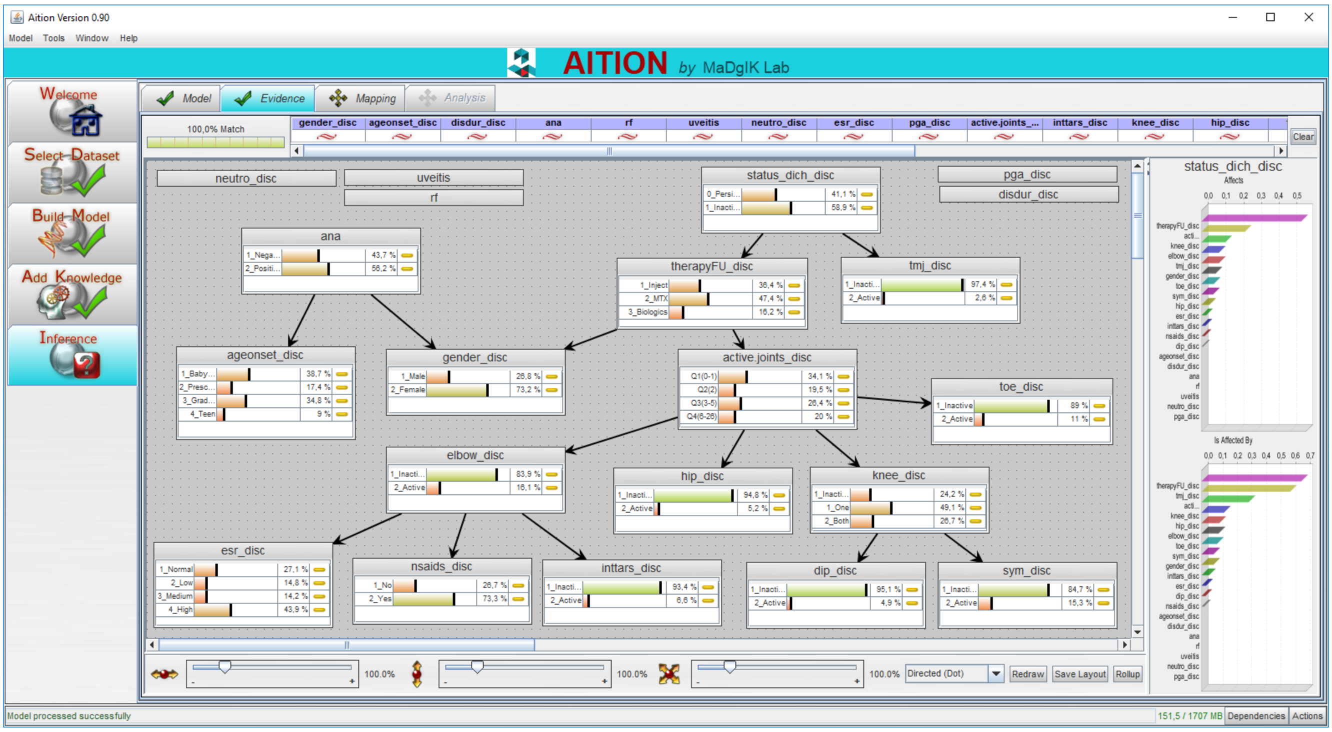


Figure 16: The JIA clinical variable model, as show within the “Inference” window of the AITION desk tool.

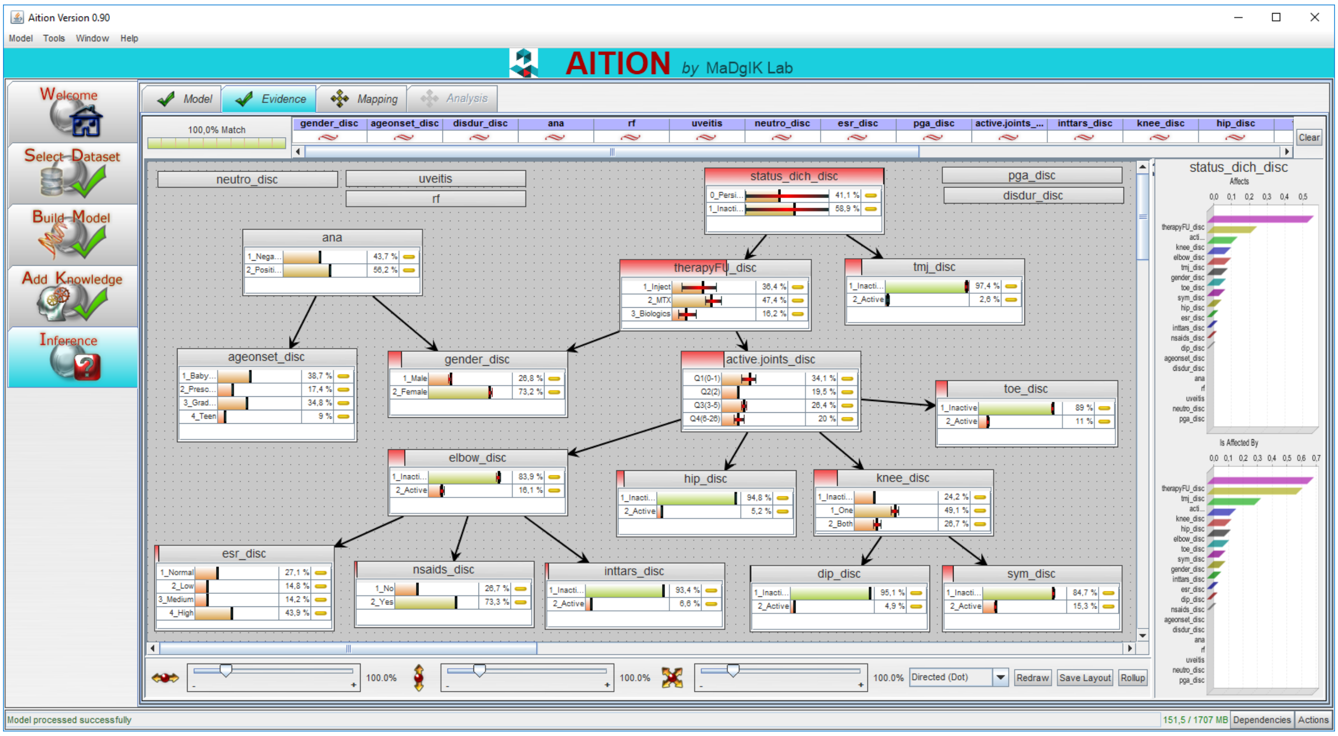
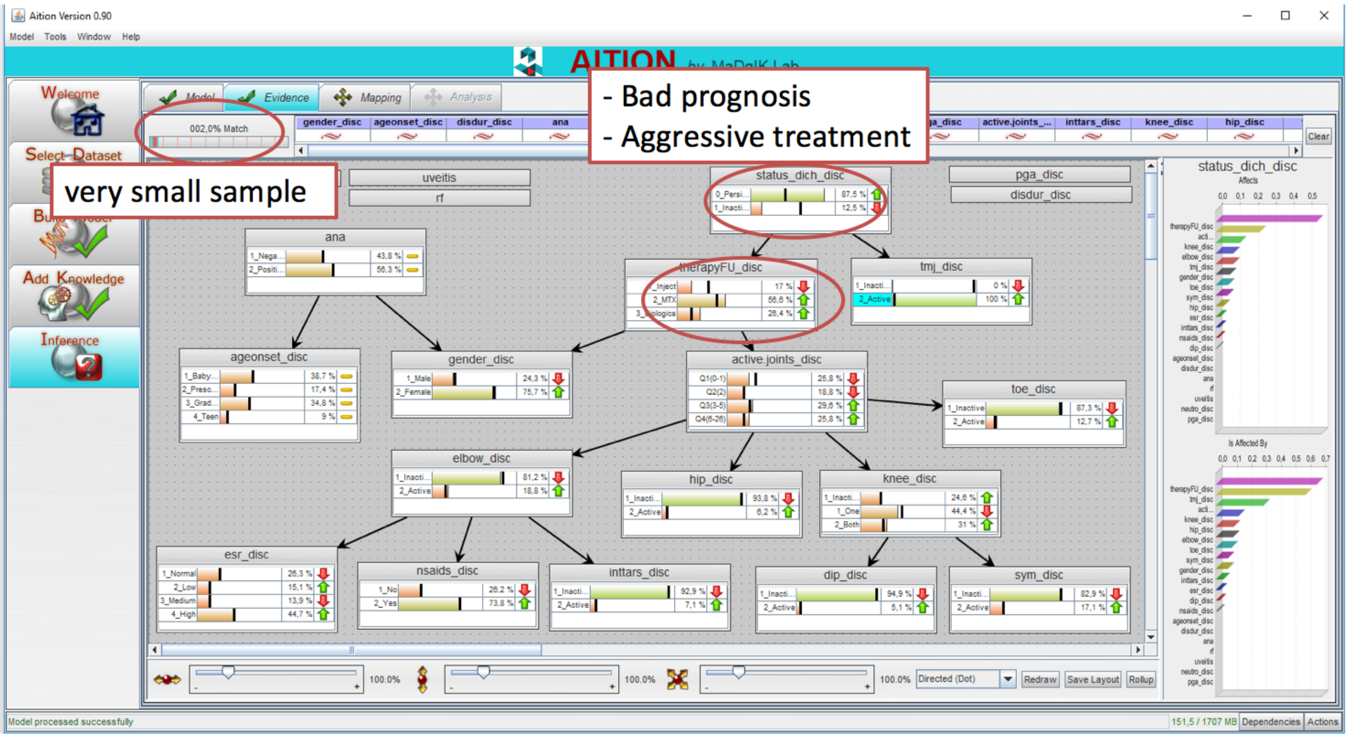


Figure 17: Sensitivity analysis on “Outcome” (‘status_dich_disc’) variable.

The following set of figures show examples of AITION’s reasoning capabilities with the JIA model. Starting from Figure 18, we see one example of how we can do reasoning using inference in graphs with AITION, where the user has set the value of the JIA variable TMJ (Temporomandibular Joints) to “2_Active”, as shown by the turquoise coloured ‘tmj_disc’ node (and the 100% value displayed along the “2_Active” value-line). We can observe that the distributions of the “Outcome” variable (‘status_dich_disc’ on top of the graph) have changed, compared to the baseline distribution of the model, shown in the graph of Figure 16: the probability of value “1_Pesistent-or-Flair” has substantially increased and that of “2_Inactive” has greatly decreased, giving a bad prognosis. The small vertical black-line that appears inside each coloured value-box indicates the original percentage of the value in the dataset, before tweaking anything. We can reset the graph to its original form by clicking on the “Clear” button in the top right corner of AITION.

Similarly, the probability of more aggressive treatment (MTX) in this case has increased, as shown by the modified distributions of the “Therapy Follow-up” variable (‘therapFU_disc’), under the “Outcome” node. Note also that to the left of the long purple vector bar there appears a percentage indicating how many of the patients in our data sample match the specified vector (e.g. in this case, only 2% of patients in the JIA data have TMJ involvement). This shows that a very small sample supports this conclusion; however, JIA clinicians confirmed that TMJ involvement is a known ‘bad’ joint, validating this association.



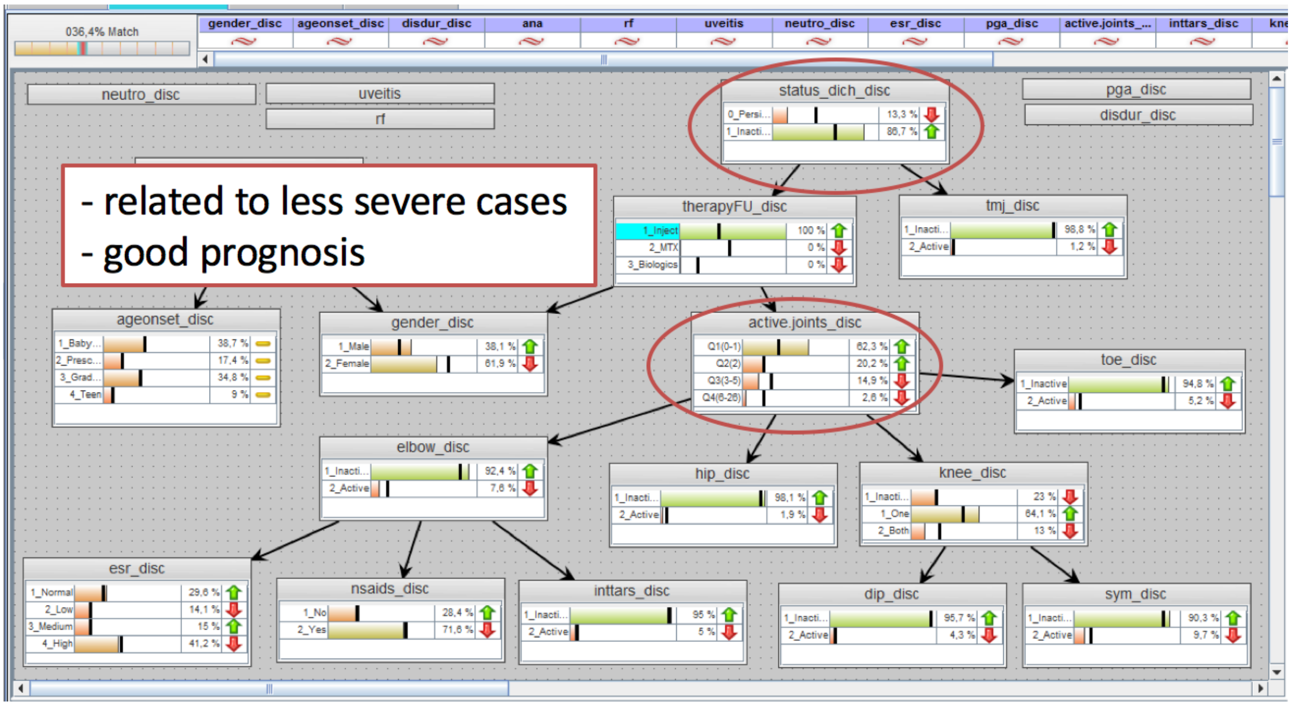
- Bad prognosis
- Aggressive treatment

very small sample

Figure 18: Inference example: setting TMJ (“tmj_disc” node) active.

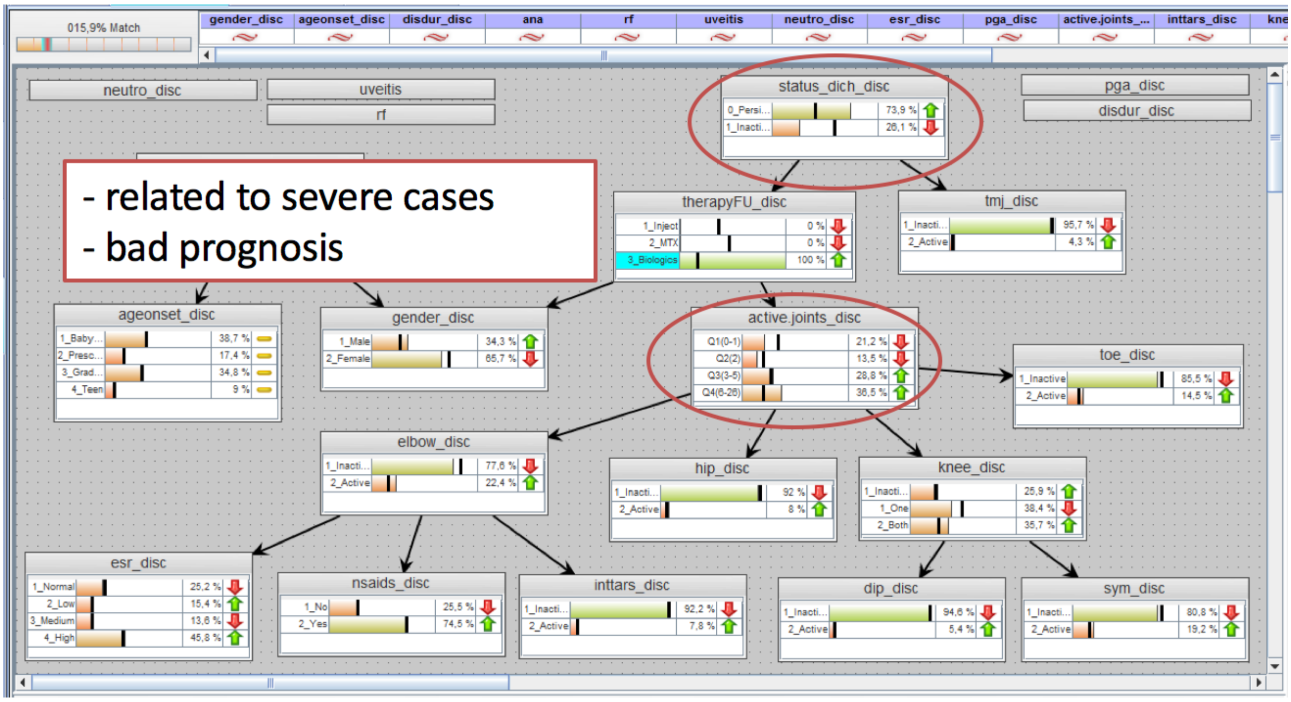
On the question of validating the graph, JIA clinicians were happy that the JIA model showed most expected dependencies, such as the ones discussed above (TMJ and Therapy Follow-Up being strongly connected to Outcome). Going ‘down the graph’, they were also not surprised that therapy is connected with the number of active joints and that this in turn is connected to the various joints in our data set. The dip-knee connection is an inverted one: those having dip involvement, have less probability of knee involvement (explainable by the various subtypes of JIA). They were a bit surprised by the connection between ESR, NSAIDs and the inter-tarsals on the one hand with elbow on the other hand, so they may explore this a little bit further. The problem here is that the elbow is not a typical joint of any subtype. On the other hand, the inter-tarsals are a typical joint of a JIA subtype which is associated with high ESR and which tends to be treated with NSAIDs, so those three are explainable. In the correlation graph, it seems that elbow is positively correlated with the inter-tarsals, so we should maybe start to say that also the elbow is a typical joint of this particular subtype. On the other hand, the connection between gender and therapy was surprising at first sight, but then again this may be due to different subtypes of JIA being more frequent in females or males and being treated differently. The connection between ANA positivity and young females at onset is very firm knowledge, so they were pleased that we found that too. The excluded variables were not so surprising to clinicians, although uveitis should be associated with ANA positivity, but it may have failed to pick up that relationship due to the small numbers. I general, they concluded that this was "very nice work and interesting to look at and to reason about", prompting them with ideas for follow-up studies.

In Figure 19 & Figure 20, the user selects different treatment values of the "Therapy Follow-up" variable and observes the effect on prognosis and other variables.



- related to less severe cases
- good prognosis

Figure 19: Inference: Treatment plan set to Injections.



- related to severe cases
- bad prognosis

Figure 20: Inference: Treatment plan set to Biologics.

Different "what-if?" scenarios are explored in the following screenshots. In Figure 21, the clinician explores what could happen if a new patient arrives with two active knee joints and symmetry (i.e. setting the values of three JIA variables). In addition, in the following three figures s/he also investigates what the model predicts for this patient if treated with injections (Figure 22), MTX drug (Figure 23), or Biologics (Figure 24).

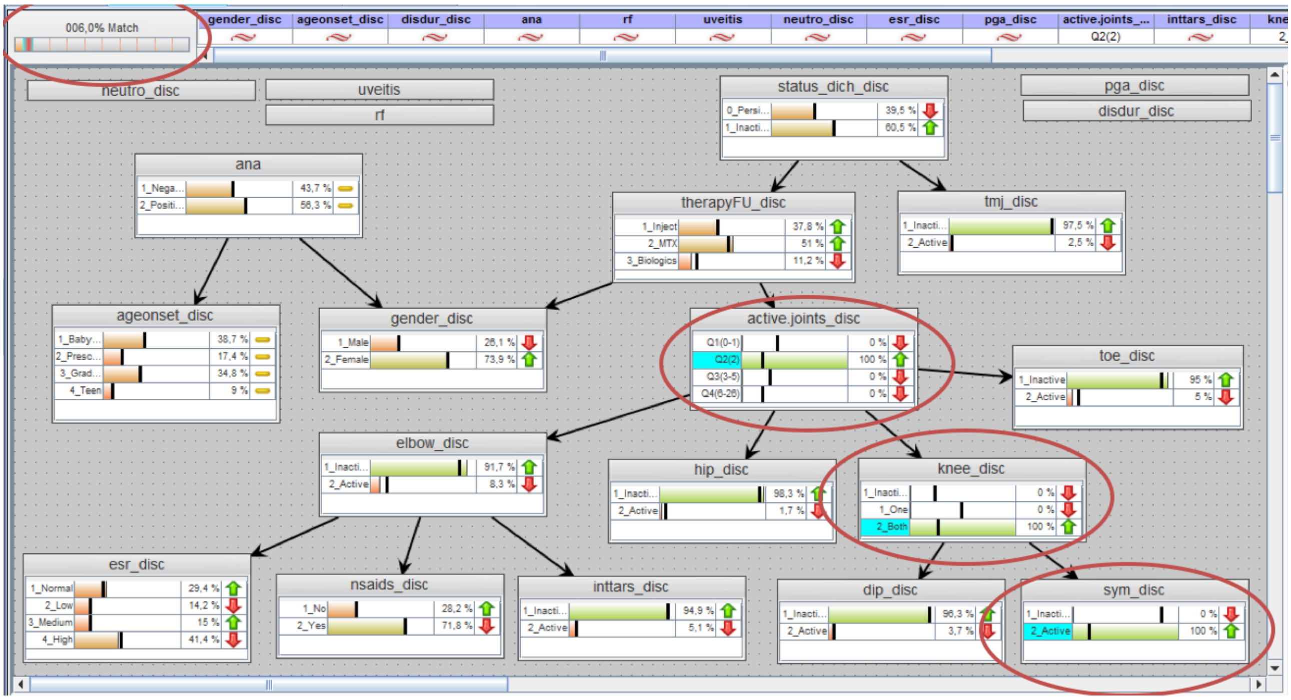


Figure 21: What-if a new patient arrives with 2 active knee joints & symmetry?

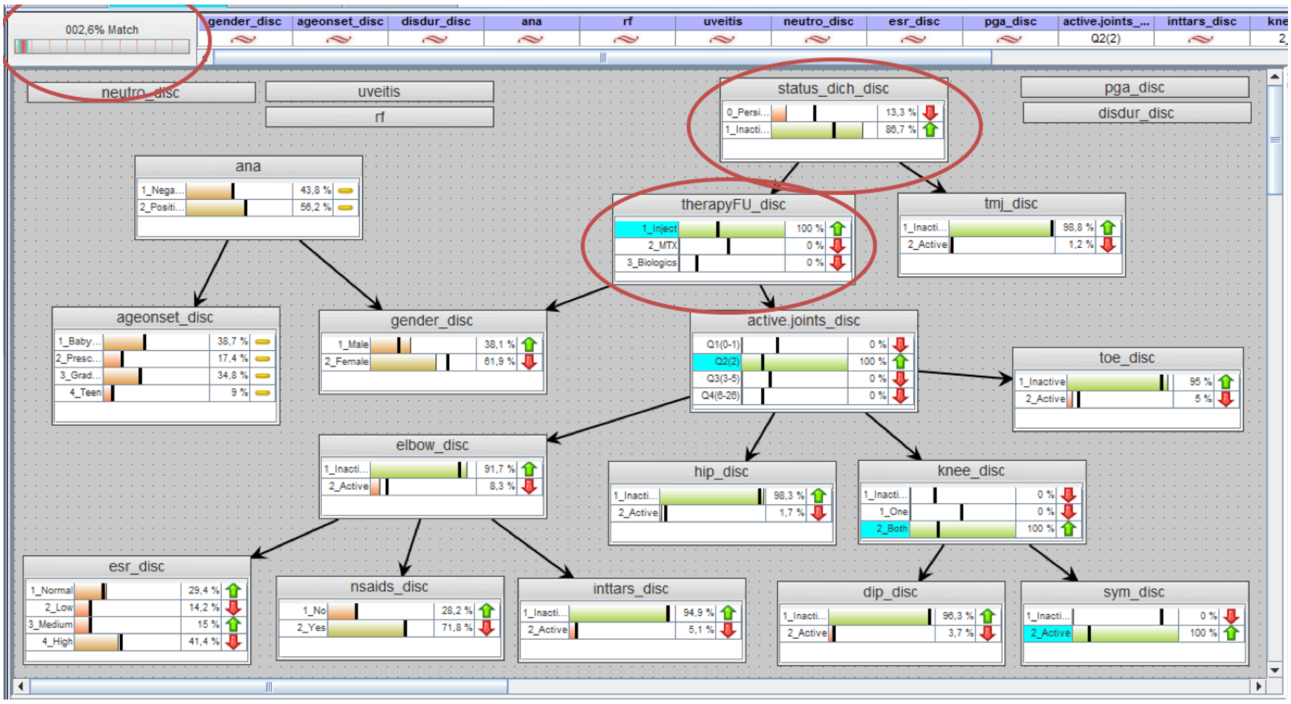


Figure 22: What if.. Therapy = inject?

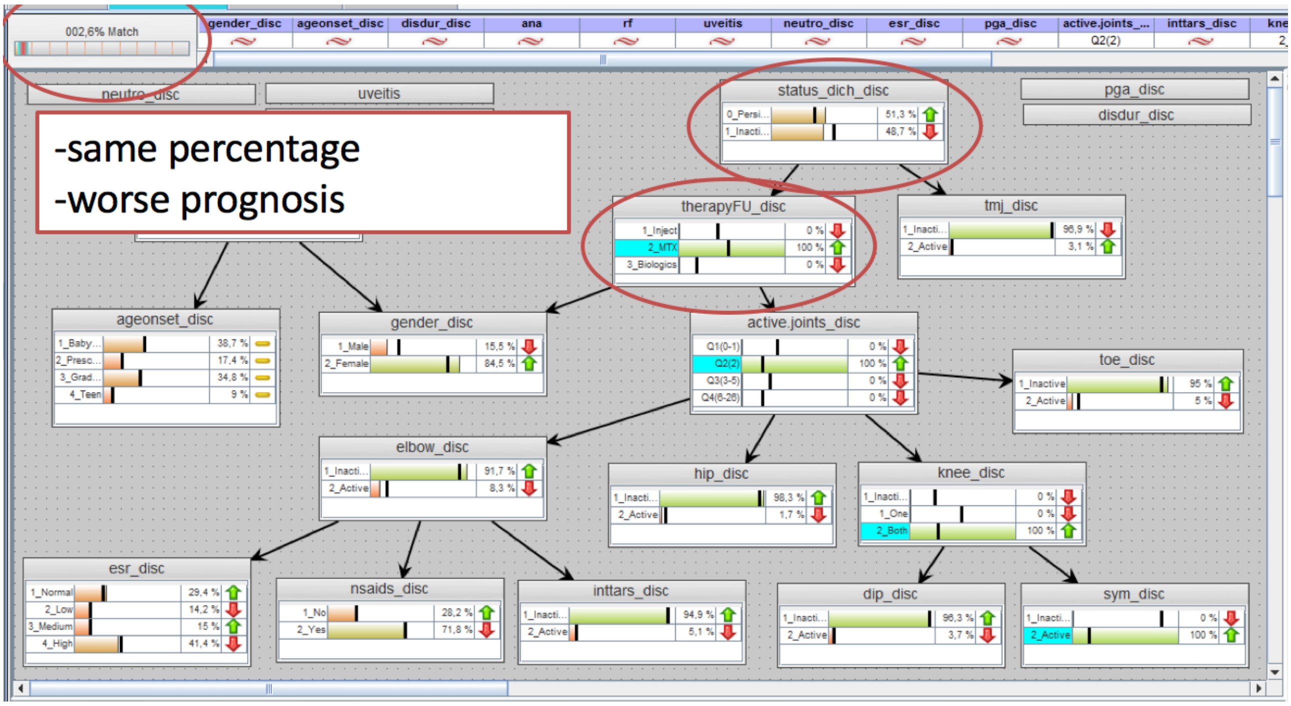

-same percentage
-worse prognosis

Figure 23: What if.. Therapy = MTX?

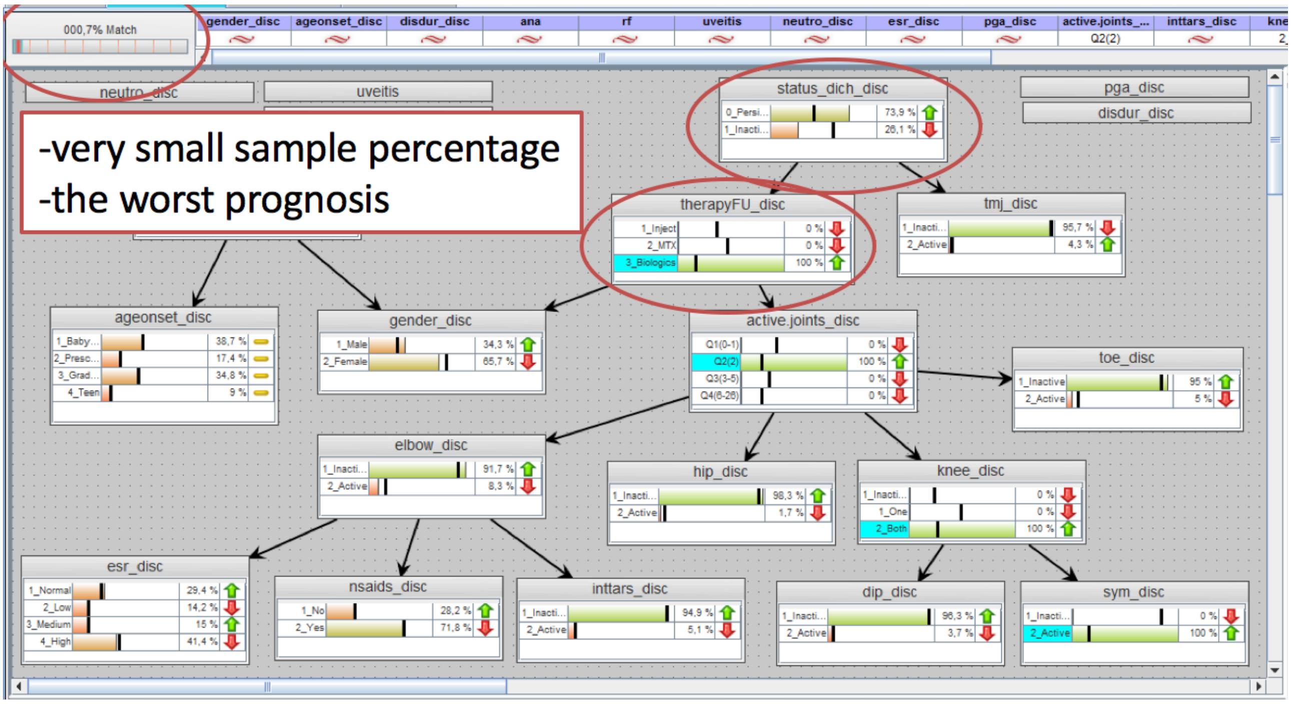

-very small sample percentage
-the worst prognosis

Figure 24: What if.. Therapy = Biologics?

4. DeepReasoner: web-based prototype for supporting evidence-based decision making[1]

As deep learning intrinsic mechanisms are very difficult to understand, deep neural networks are considered as black boxes. Assessing the confidence of a model or even exposing some evidence of how a decision is taken remains challenging. For this reason, we propose to integrate DL-based models within a case-based reasoning framework. Starting from our previous experience on this topic, we achieved a complete re-foundation of our case-based reasoning technology we call now DeepReasoner. Not only the prototype itself, where the front-end was completely re-designed and re-implemented as well as its back-end, but also the machine learning technology at the core of the case-based reasoning engine were improved. Our former web prototype implemented in Java and served by a Jetty server supports similarity searches that use classical distance functions, such as Euclidean distance, computed over the selected clinical variables. While classical distances are well-studied and therefore easy to interpret, they do not cope well with the large dimensionality, the presence of uninformative features, or the large difference in dimensionality and scales between different sources of information. All of these features are found in the heterogeneous and multi-modal data that are being acquired in this project. Moreover, classical distances are by design not aware of the context and thus cannot weight the importance of the parameters according to the current use case.

We proposed an enhancement of the underlying knowledge-discovery engine that has the potential to overcome these limitations of similarity search, based solely on classical distance functions. This new development uses the deep learning models we described in the previous section of this deliverable to derive compact task-specific patient representations that can then be analysed in several ways, including through similarity estimations. Moreover, we completely redesigned the front end as well as the back-end using modern web technologies such as html5, JavaScript and server side scripting node.js. The retrieval engine itself is implemented as an azure machine learning web service using python scripting language and the keras deep learning library. On the side of the front-end, results analysis and visualization such as computing prediction histograms and graph clustering are implemented in JavaScript.

In a nutshell, DeepReasoner consists of two main components: i) a deep-learning based encoder that converts patient parameters into a compact representation and ii) a retrieval system that permits to retrieve from a database relevant patients as well as their corresponding outcome.

As shown in the figure below, the web application is served by a virtual machine that communicates with the DeepReasoner back-end through a tunnel for querying predictions. Serving the web app as well as communicating with the back-end is done using https.

[1] For the reader’s convenience, this section also appears reprinted in D9.4 (Report on predictive risk models and their quantitative evaluation), where the DeepReasoner is used for predictive modelling.

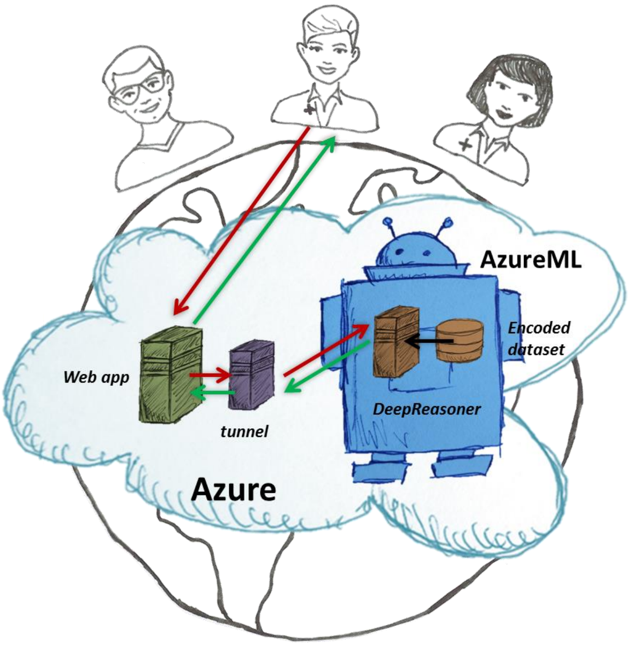
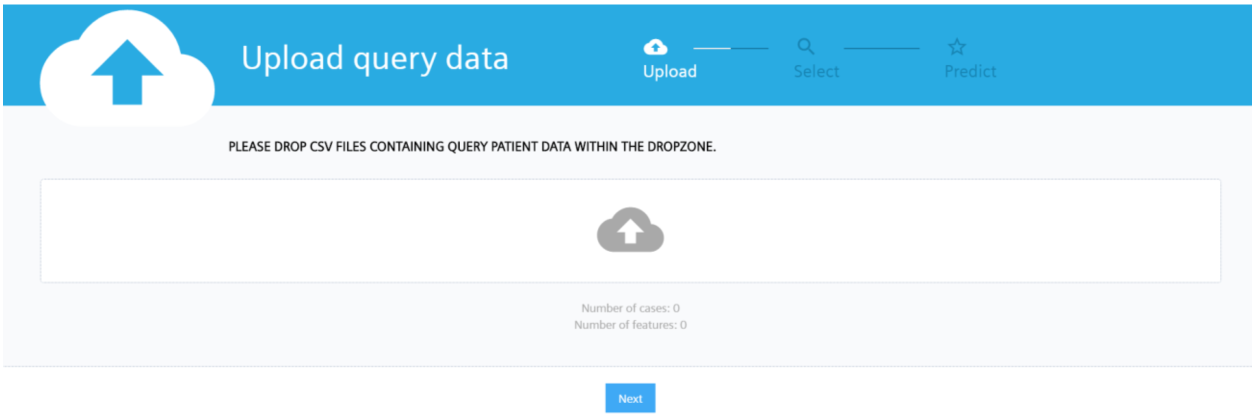


Figure 25: DeepReasoner in the cloud

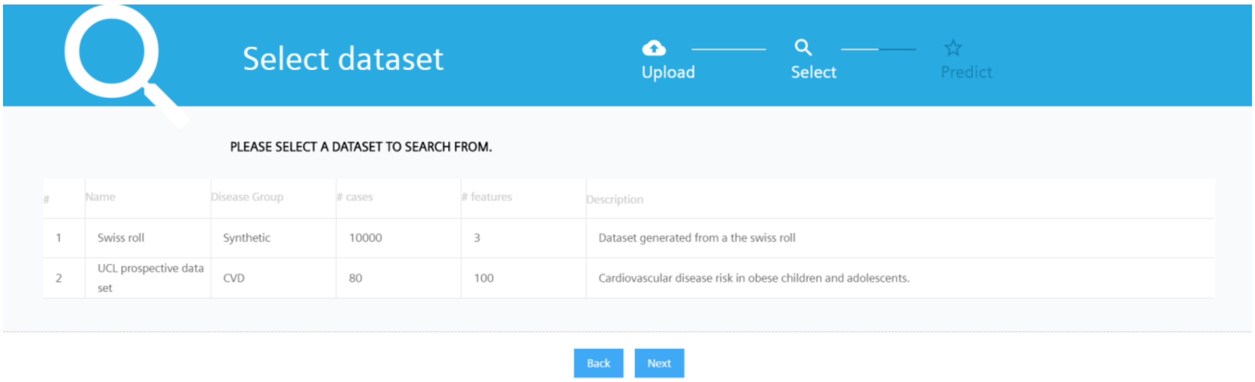
The DeepReasoner workflow consists of the following steps: 1) a query dataset is uploaded either from the Gnubila platform through the “Query Management” or directly from a user’s computer, 2) a use-case is selected from the list of available use cases, 3) a query case is selected, a number of retrieved cases is chosen and the query is sent to the back-end, 4) results are presented to the user in the form of a prediction histogram, a distance heat map as well as a patient graph. Interaction is possible with the different visualizations that are implemented using d3js technology. Users can change the number of bins for the prediction histogram, and the threshold for graph clustering can be changed. The latter triggers a re-clustering of the graph implemented in JavaScript and computed directly in the browser. The different steps are shown below.


Upload query data

PLEASE DROP CSV FILES CONTAINING QUERY PATIENT DATA WITHIN THE DROPZONE.

Number of cases: 0
Number of features: 0

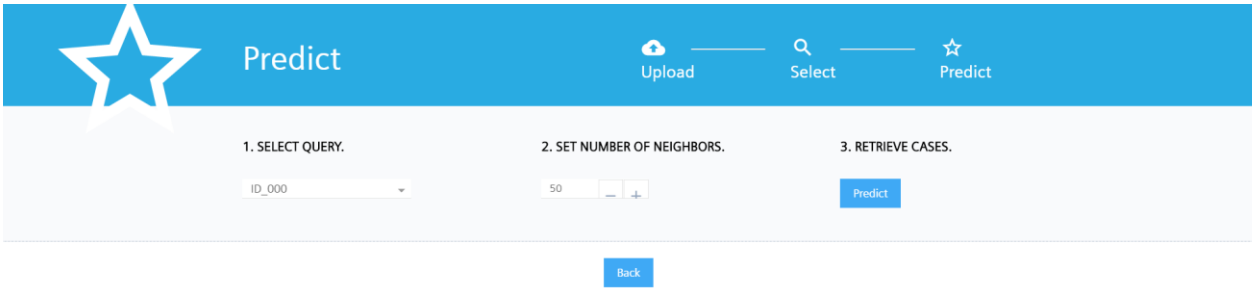
Figure 26: Upload query



PLEASE SELECT A DATASET TO SEARCH FROM.

#	Name	Disease Group	# cases	# features	Description
1	Swiss roll	Synthetic	10000	3	Dataset generated from a the swiss roll
2	UCL prospective data set	CVD	80	100	Cardiovascular disease risk in obese children and adolescents.

Figure 27: Select use case



1. SELECT QUERY.

ID_000

2. SET NUMBER OF NEIGHBORS.

50

3. RETRIEVE CASES.

Figure 28: Setup query

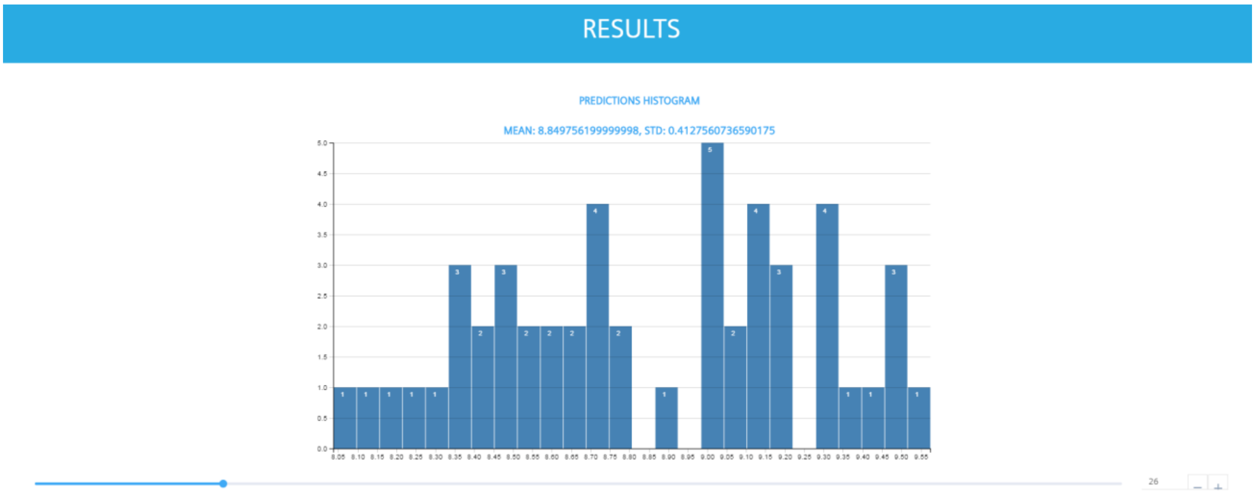


Figure 29: Visualize prediction distribution

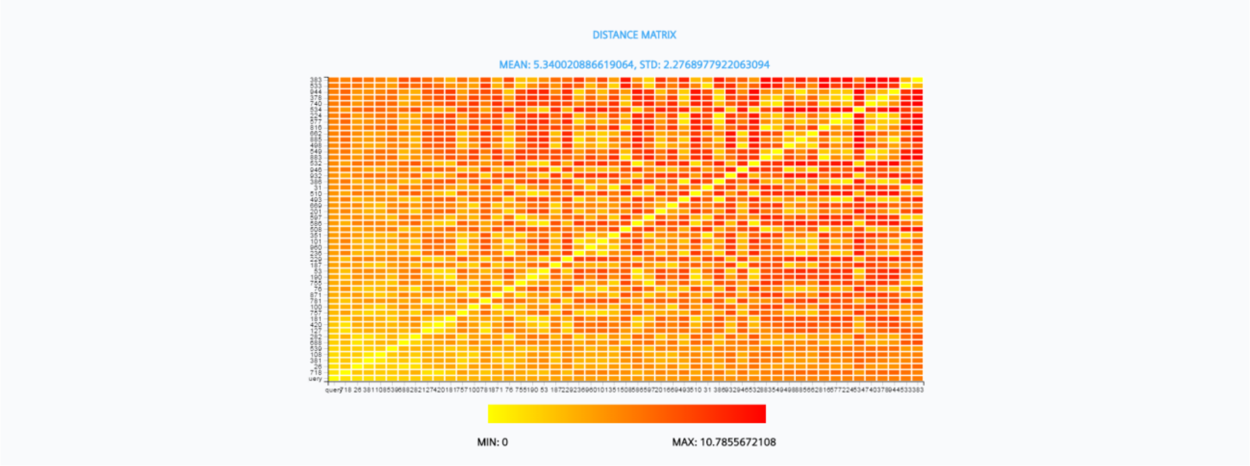


Figure 30: Visualize similarity between cases through distance heat map

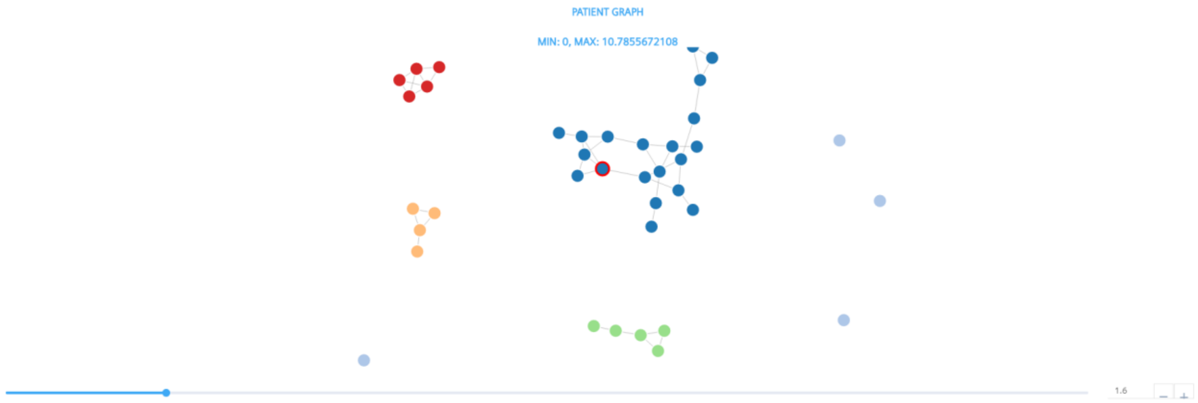


Figure 31: Visualize patient graph and perform clustering in the browser

5. Personalization of the Whole-Body Circulation Model

For the personalization of the whole-body circulation model we use a set of standard objectives, formulated based on systolic, diastolic and average arterial/venous aortic pressure, interval of time during which the aortic/pulmonary valve is open, and maximum and minimum left/right ventricular volume. The personalization of the whole-body circulation model, which is used to provide input features for the DeepReasoner, was updated as follows: beyond this set of 12 objectives, the personalization based on the following objectives was improved.

1. Pressure based objectives

Based on the shape of the pressure curves, the following objectives are defined (Figure 1):

- $Slope_1$: the slope of the systolic pressure during early systole (before the valve opens)
- P_{Dia}: the ventricular pressure at mid diastole
- $Slope_2$: the slope of the aortic pressure decay during diastole

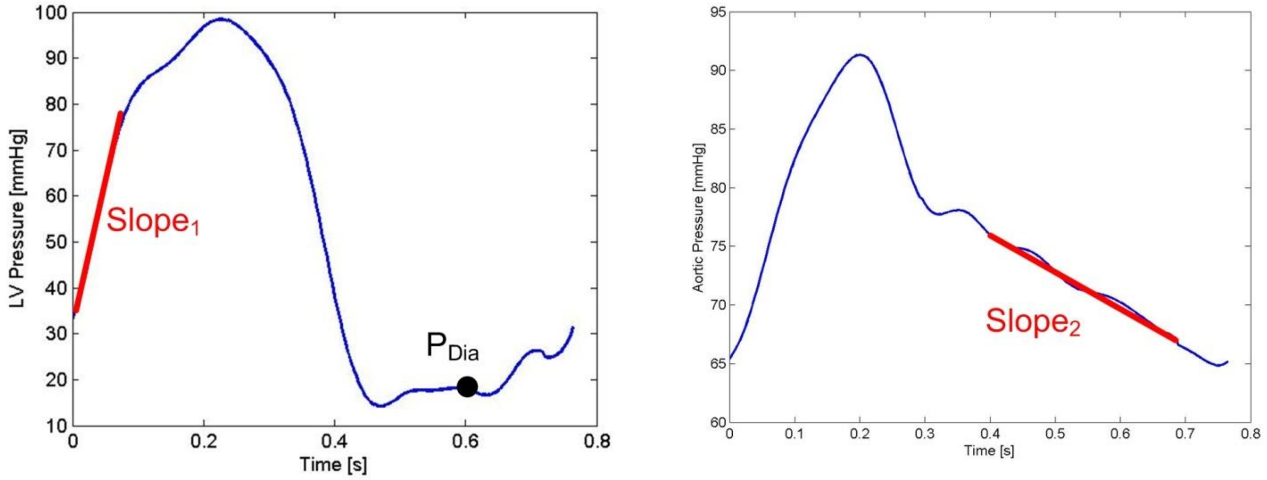


Figure 32: Additional pressure based objectives.

2. Volume based objectives

Based on the shape of the volume curve, the patients are classified into three categories (Figure 2):

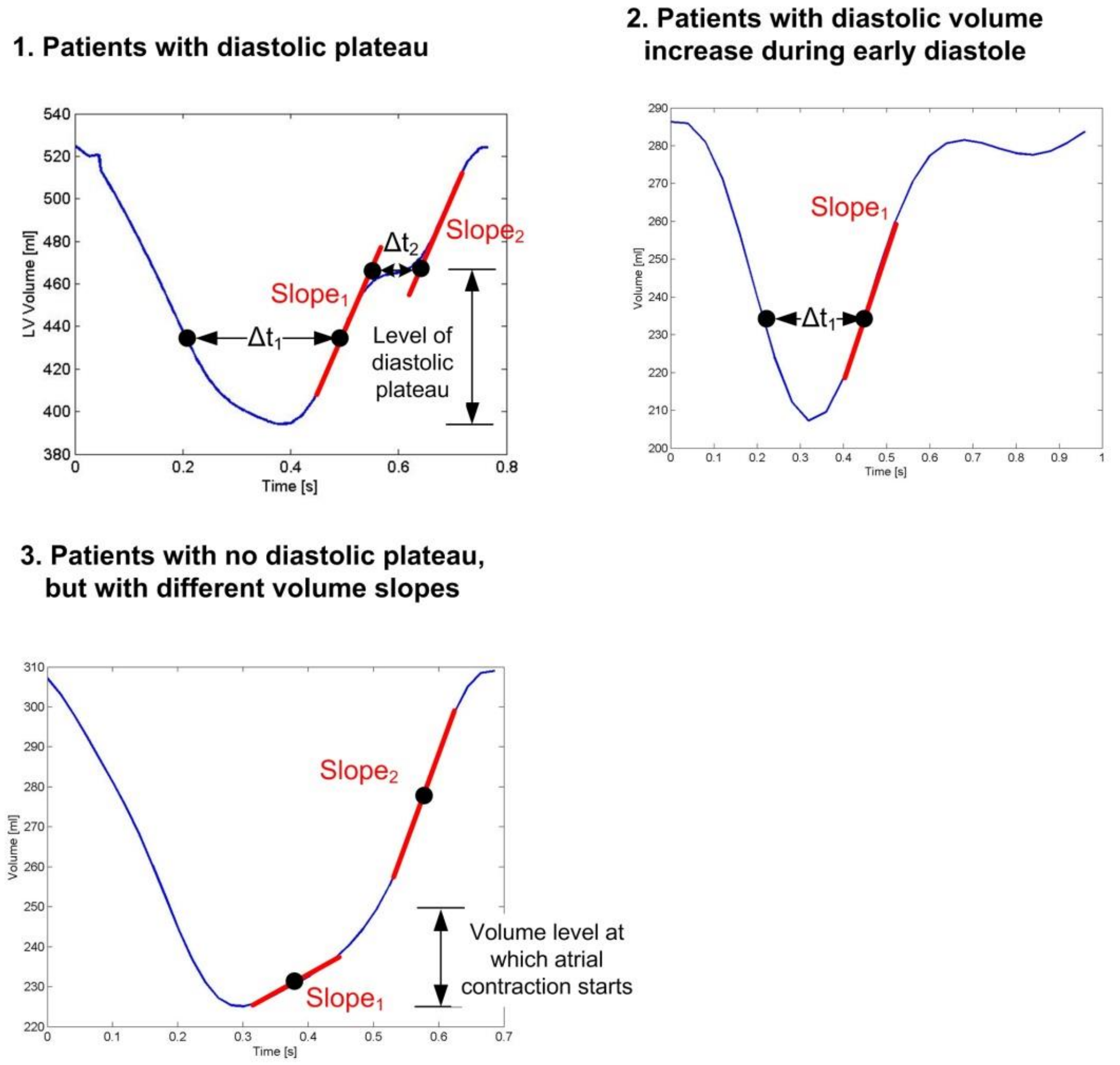


Figure 33: Additional volume based objectives.

a. Patients with diastolic plateau

- Δt_1: the delta time on the volume curve between two identical volume values (one in systole and one in diastole), at a volume level of 30% between minimum and maximum volume
- Slope_1: the slope of the volume curve in diastole at the above defined volume value
- Δt_2: the duration of the diastolic plateau
- Slope_2: the slope of the volume curve when the atrium contracts
- Level of diastolic plateau, expressed in percentages between the min and max volume

b. Patients with diastolic volume increase during early diastole

- Δt_1: the delta time on the volume curve between two identical volumes (one in systole and one in diastole), at a volume level of 30% between minimum and maximum volume
- Slope_1: the slope of the volume curve in diastole at the above defined volume value

c. Patients with no diastolic plateau but with different slopes during diastole

- Slope_1: the slope of the volume curve in early diastole, before atrial contraction
- Slope_2: the slope of the volume curve in late diastole, after atrial contraction
- Volume level at which atrial contraction starts

The updated personalized whole-body circulation model was used to re-personalize all patients with CVD risk (WP4). The figures below display two samples which exemplify the excellent match between computed and measured volume curves.

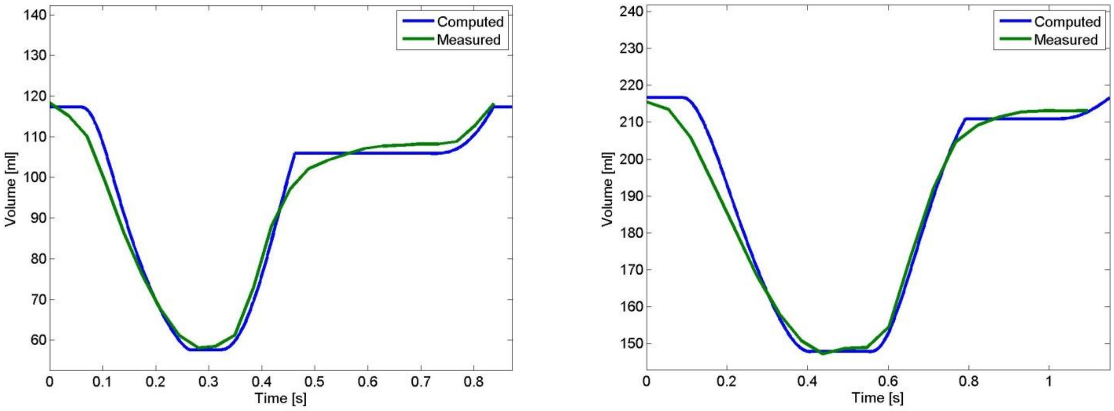


Figure 34: Comparison between computed and measured LV volume for an OPBG patient (left) and a UCL patient (right).

6. References

[1] van der Maaten, L.J.P.; Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9:2579-2605, 2008.

[2] Edward McCormick and Kris De Volder. 2004. JQuery: finding your way through tangled code. In Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications (OOPSLA '04). ACM, New York, NY, USA, 9-10. DOI: http://dx.doi.org/10.1145/1028664.1028670

[3] w2ui JavaScript UI-w2ui http://w2ui.com/web. Retrieved February 24th 2014.

[4] M. Leibman and other contributors. Slick grid. Online available at https://github.com/mleibman/SlickGrid [accessed 27- April-2015].

[5] Bostock, M.: D3.js. http://mbostock.github.com/d3/

[6] S. Ruggieri, "Efficient C4.5 [classification algorithm]," in IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 2, pp. 438-444, Mar/Apr 2002. doi: 10.1109/69.991727

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[8] Chronis, Y. et al. A Relational Approach to Complex Dataflows, in Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference (2016).

[9] https://github.com/madgik/madis

[10] D. Koller and N. Friedman, Probabilistic Graphical Models, MIT Press, 2009.

[11] J. Pearl, Probabilistic Reasoning in Intelligent Systems, 2nd revised ed., Morgan Kaufmann, San Mateo, 1988.

[12] D. Eaton and K. Murphy, “Bayesian structure learning using dynamic programming and MCMC,” in UAI, 2007.

[13] M. Koivisto, “Advances in exact Bayesian structure discovery in Bayesian networks,” in UAI, 2006.

[14] M. Koivisto and K. Sood, “Exact Bayesian structure discovery in Bayesian networks,” J. Mach. Learn. Research, vol. 5, pp. 549-573, 2004.

[15] C. Aliferis, I. Tsamardinos, and A. Statnikov, “HITON: A novel Markov Blanket algorithm for optimal variable selection,” presented at AMIA Annu. Symp. Proc., 2003.

[16] L. Brown, I. Tsamardinos, and C. Aliferis, “A novel algorithm for scalable and accurate Bayesian network learning,” in MEDINFO2004.

[17] L. Brown, C.Aliferis, and I. Tsamardinos, “A comparison of novel and state-of-the-art polynomial Bayesian network learning algorithms,” in Amer. Assoc. Artificial Intell. (AAAI), 2005.

[18] I. Tsamardinos, L. Brown, and C. Aliferis, “The max-min hill climbing Bayesian network structure learning algorithm,” Mach. Learn., vol. 65, pp. 31-78, 2006.

[19] M. Wainwright, P. Ravikumar, and J. Lafferty, “High-dimensional graphical model selection using l1-regularized logistic regression,” in NIPS, 2007.

[20] P. C. G Costa, K. Laskey, and G. AlGhamdi, “Bayesian ontologies in AI systems”, presented at 4th Bayesian Modell. Applicat. workshop, held at 22nd Conf. Uncertainty in AI, 2006, Cambridge, MA, USA.