

The Protégé-Owl SWRLTab and Temporal Data Mining in Surgery

G Tusch, M O'Connor, T Redmond, R Shankar and A Das

Stanford Medical Informatics

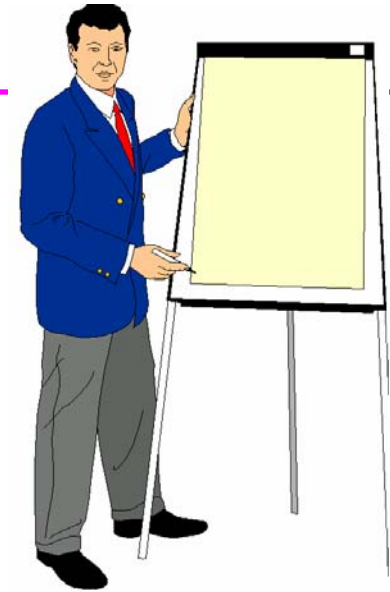
*Medical and Bioinformatics Program
School of Computing and Information Systems
Grand Valley State University
Allendale MI*



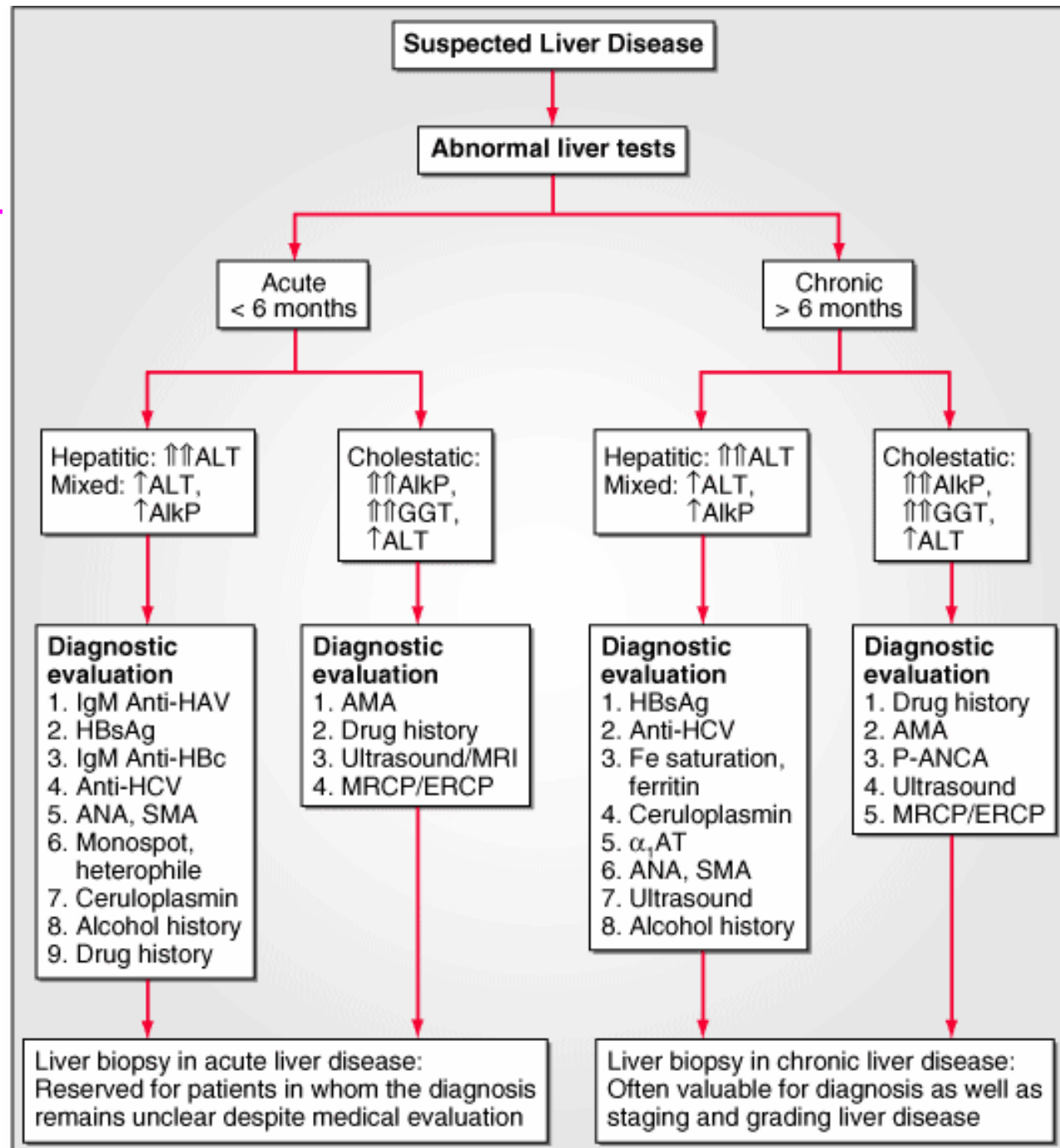
10th Intl. Protégé Conference - July 15-18, 2007 - Budapest, Hungary

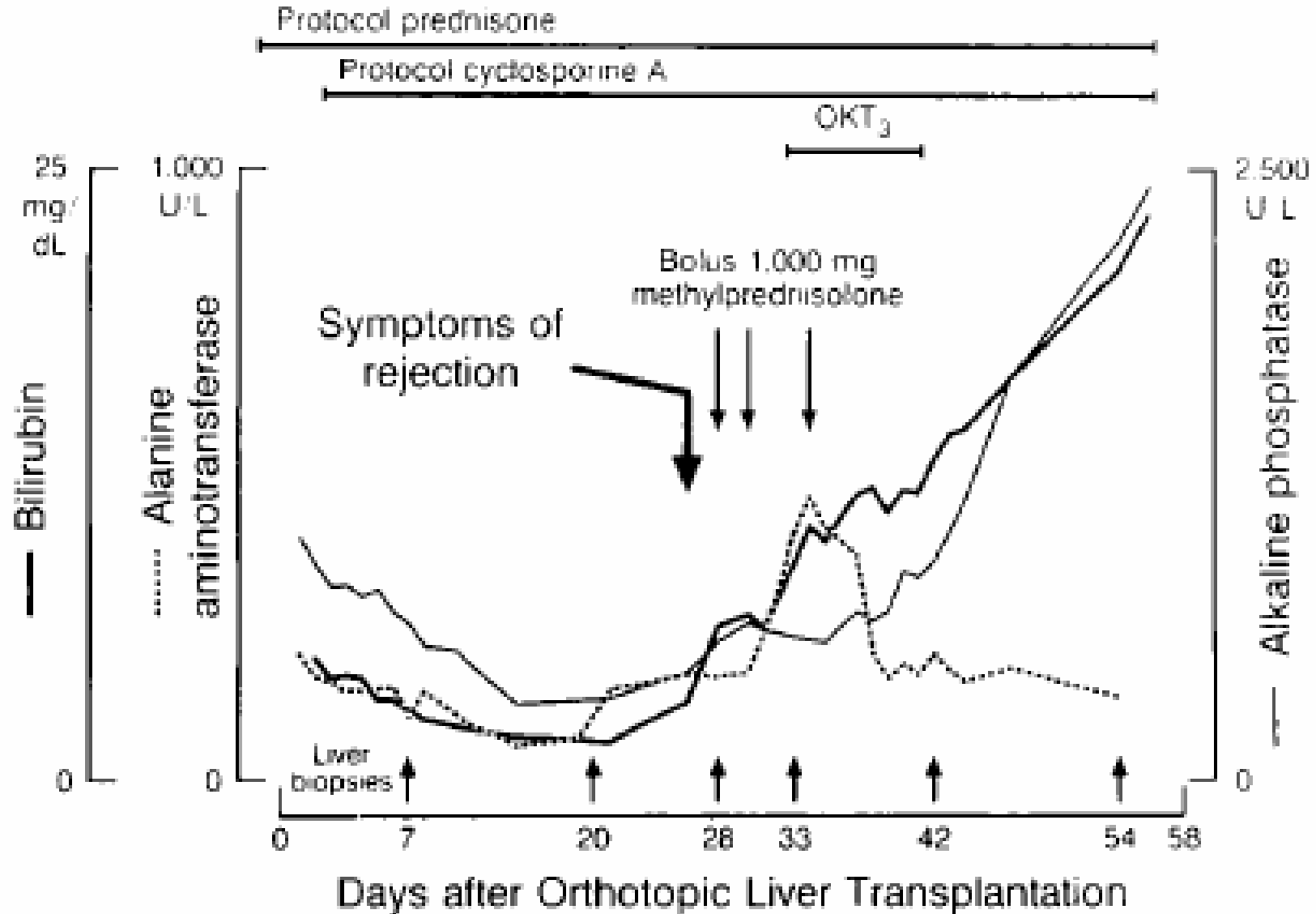
Outline

- Introduction (An Example of Transplantation Surgery)
- The SPOT Design
- Statistical Aspects
- SPOT in Surgery
- Conclusion



<http://www.ladybird.co.uk/favouriteCharacters/spot.html>





Wiesner et al. Hepatology. 1991 Oct;14(4 Pt 1):721-9.

SPOT and Temporal Abstraction

- **Purpose of SPOT (S - Protégé – OWL/SWRL – Temporal Abstraction):**
 - Mining large clinical databases including exploration of temporal data
 - Example liver transplantation: researcher looks for patients with an unusual pattern of potential complications of the transplanted organ
- TA is defined as the creation of high-level summaries of time-oriented data
- TA is necessary because
 - clinical databases usually store raw, time-stamped data
 - clinical decisions often require information in high-level terms

The Temporal-Abstraction Task (Shahar)

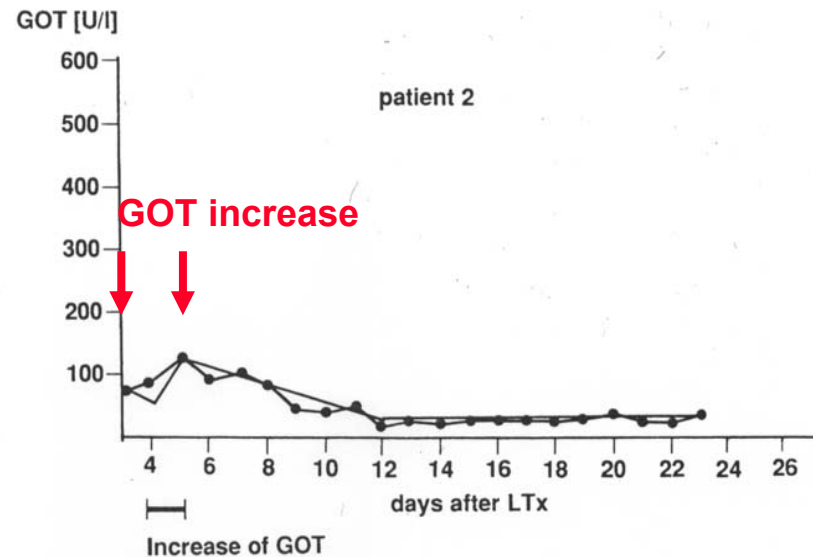
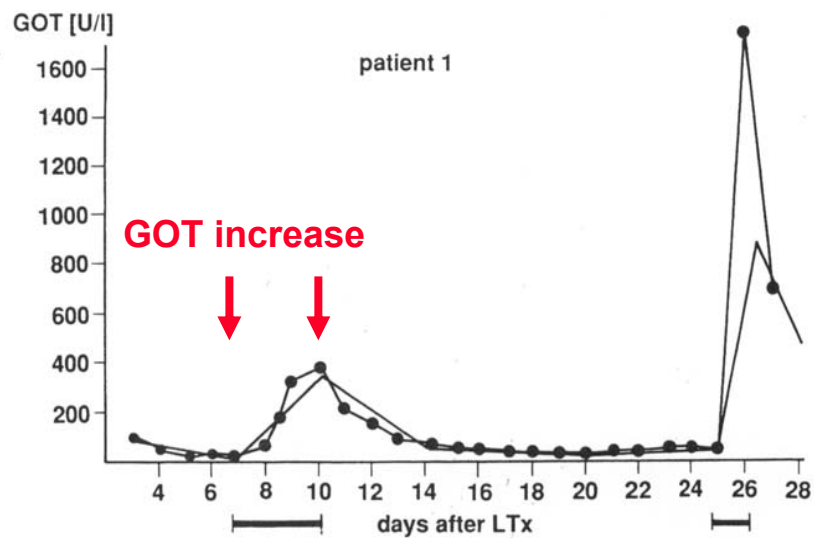
- **Input:** time-stamped clinical data and relevant events (interventions)
- **Output:** interval-based abstractions
- Identifies past and present trends and states

Output types:

- State abstractions (LOW, HIGH)
- Gradient abstractions (INCREASE, DECREASE)
- Rate Abstractions (SLOW, FAST)
- Pattern Abstractions (CRESCENDO)
 - Linear patterns
 - Periodic patterns

Examples of patient courses in liver Tx

Concept: GOT (=AST) increase

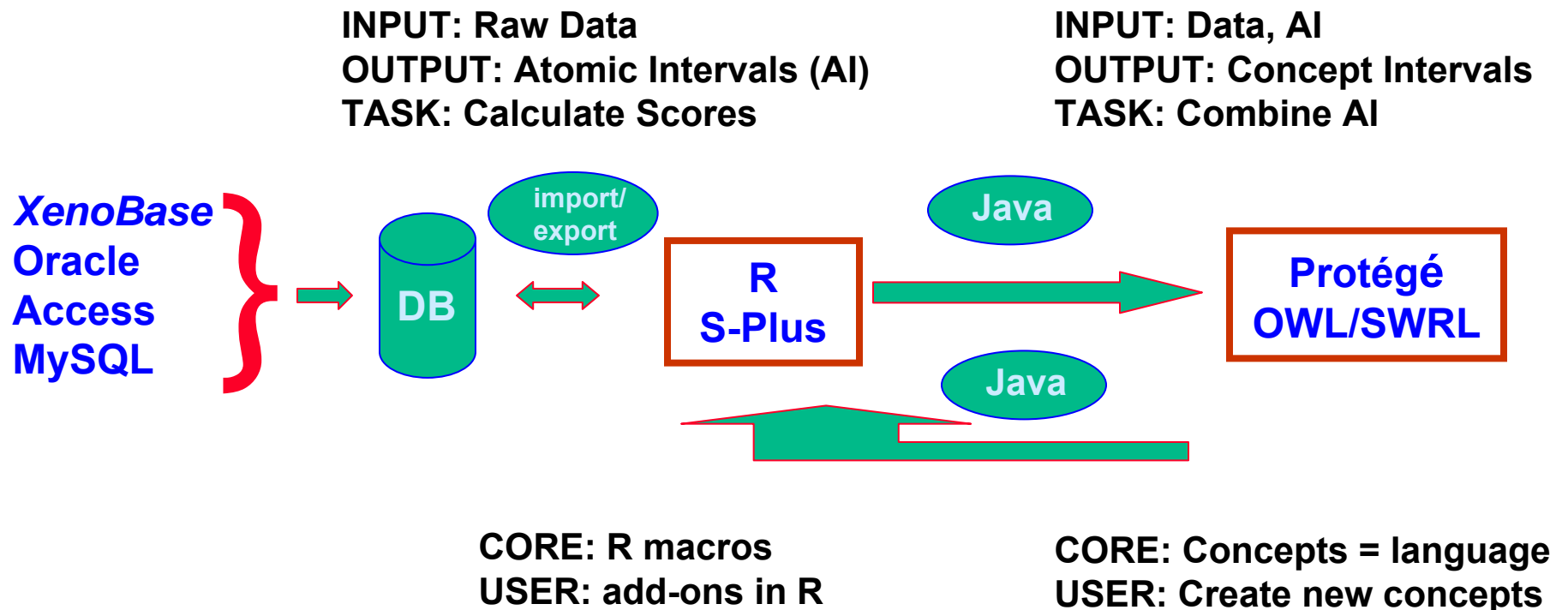


Tasks and Software

- Estimation of intervals from learning sample: S (R/S-Plus)
- Build high level concepts (Temporal Abstraction):
Protégé/OWL/SWRL
- Validate intervals: S (R/S-Plus)
- Run abstractions on original database: RASTA?

SPOT Overview

Learning Concepts from a Subset (Train & Test Data Set)

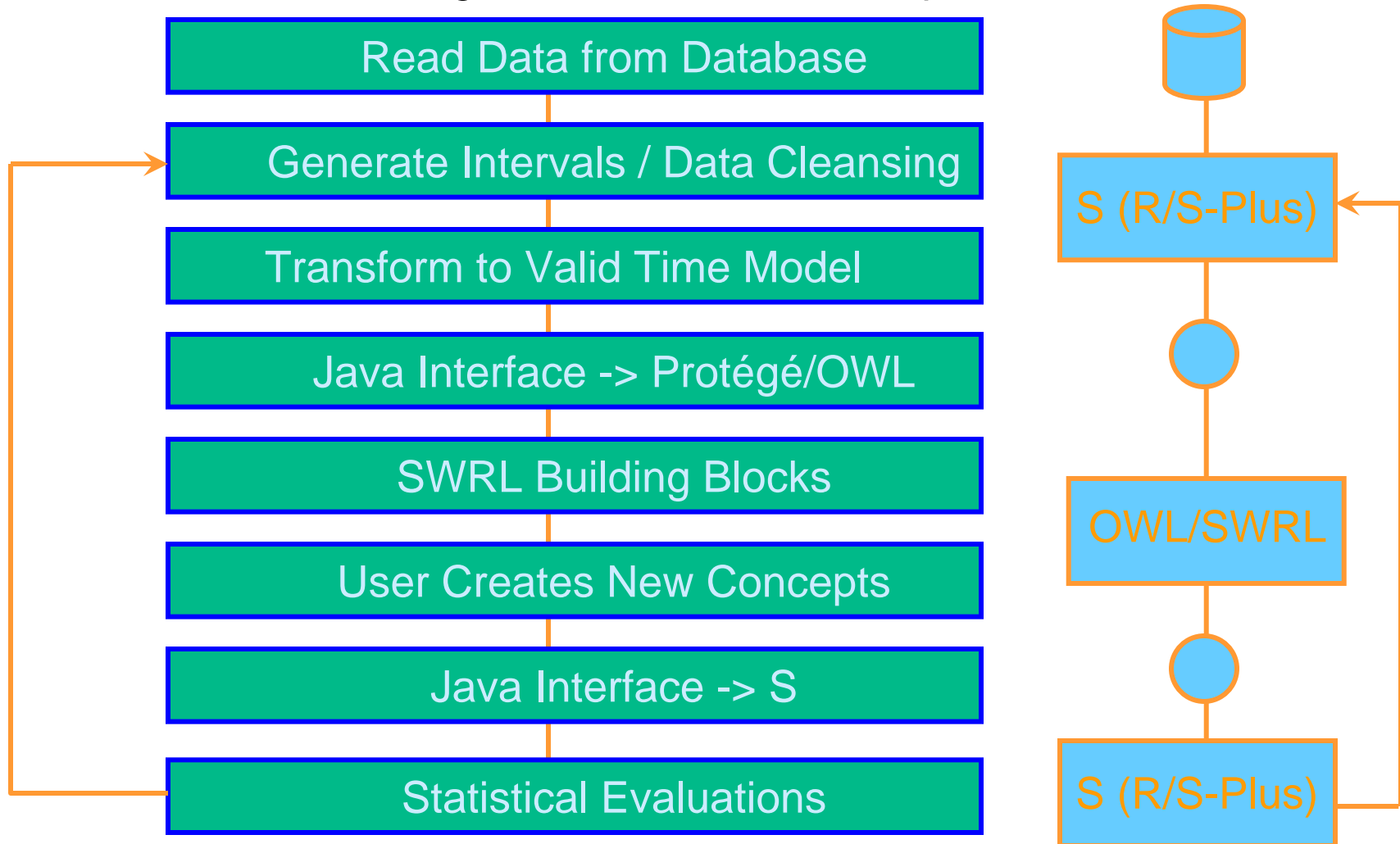


Searching for Learned Concepts in Database

TASK: - Search for patients with episodes and additional parameters (e.g., survival)

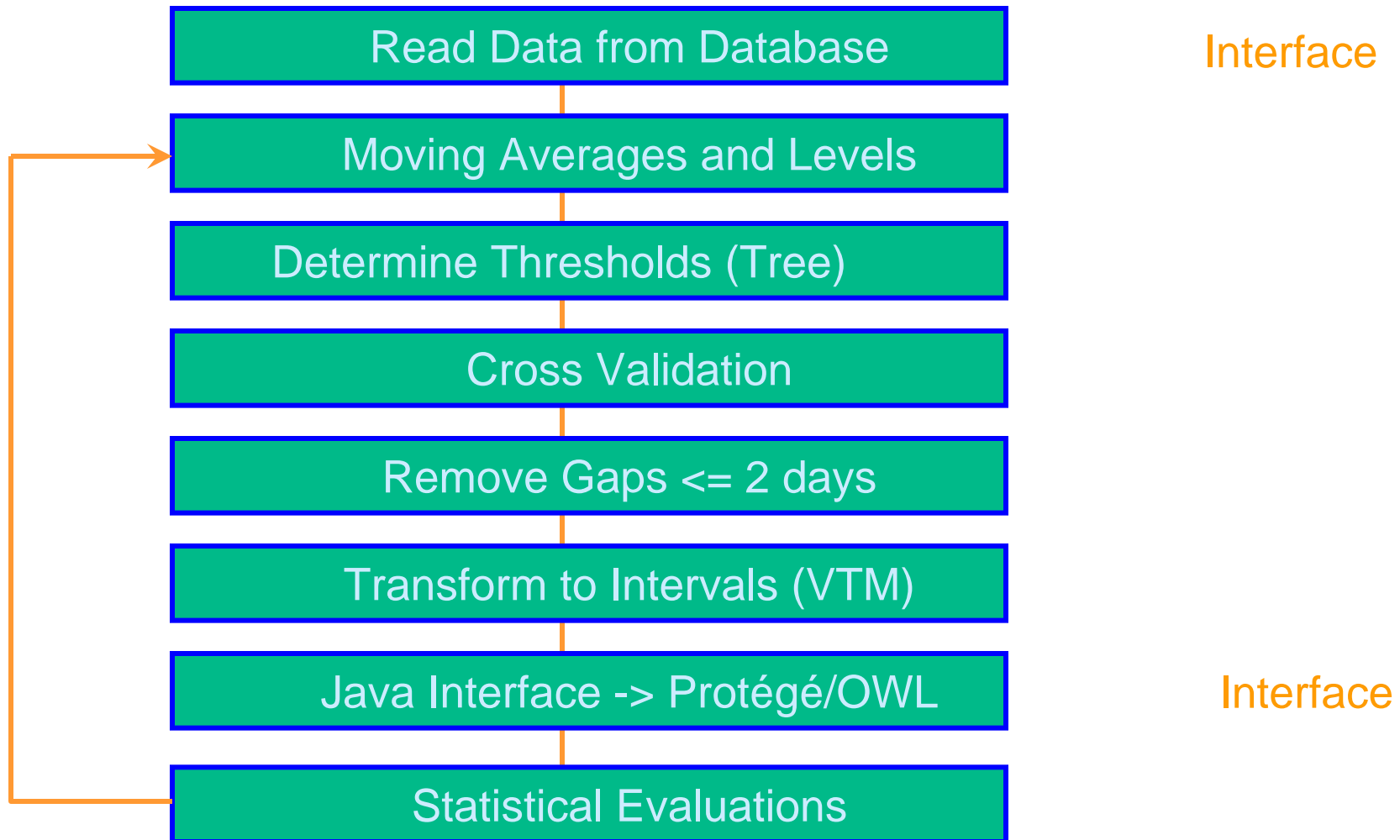
SPOT Structure

SPOT: S - Protégé – OWL/SWRL – Temporal Abstraction



SPOT Structure (S)

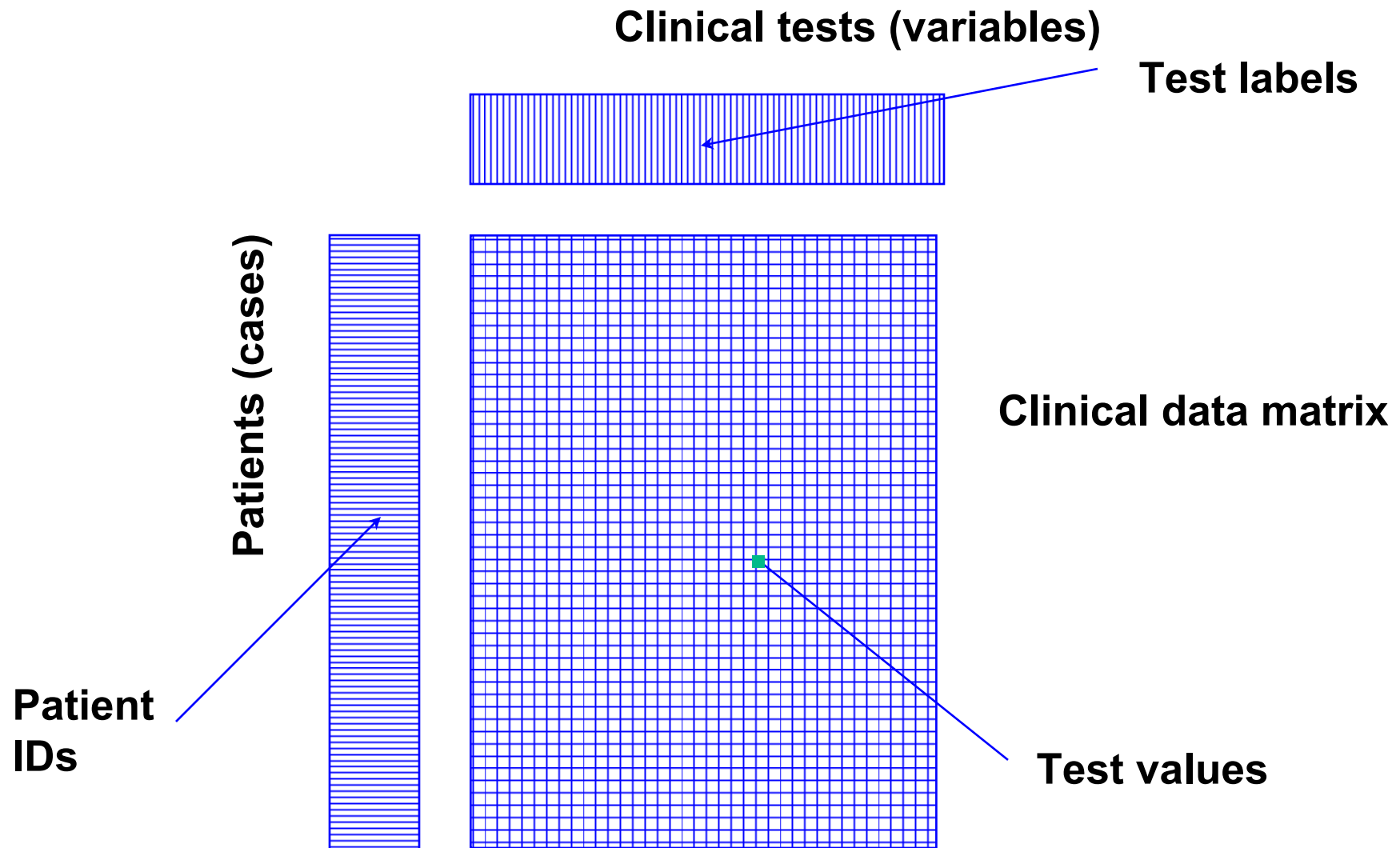
S Part



Input Data

- Time stamped data in database or time course graph e.g. in Xenobase
- Researcher (user) marks intervals per parameter (e.g. GOT)
 - Several different non-overlapping intervals are allowed, but only one parameter (independence assumption), i.e. mark as “increasing”, “decreasing”, “high”, etc.
 - Interval value is attached to time-stamped parameter value
 - Generate learning and test samples

Data Structure: Clinical Data Example

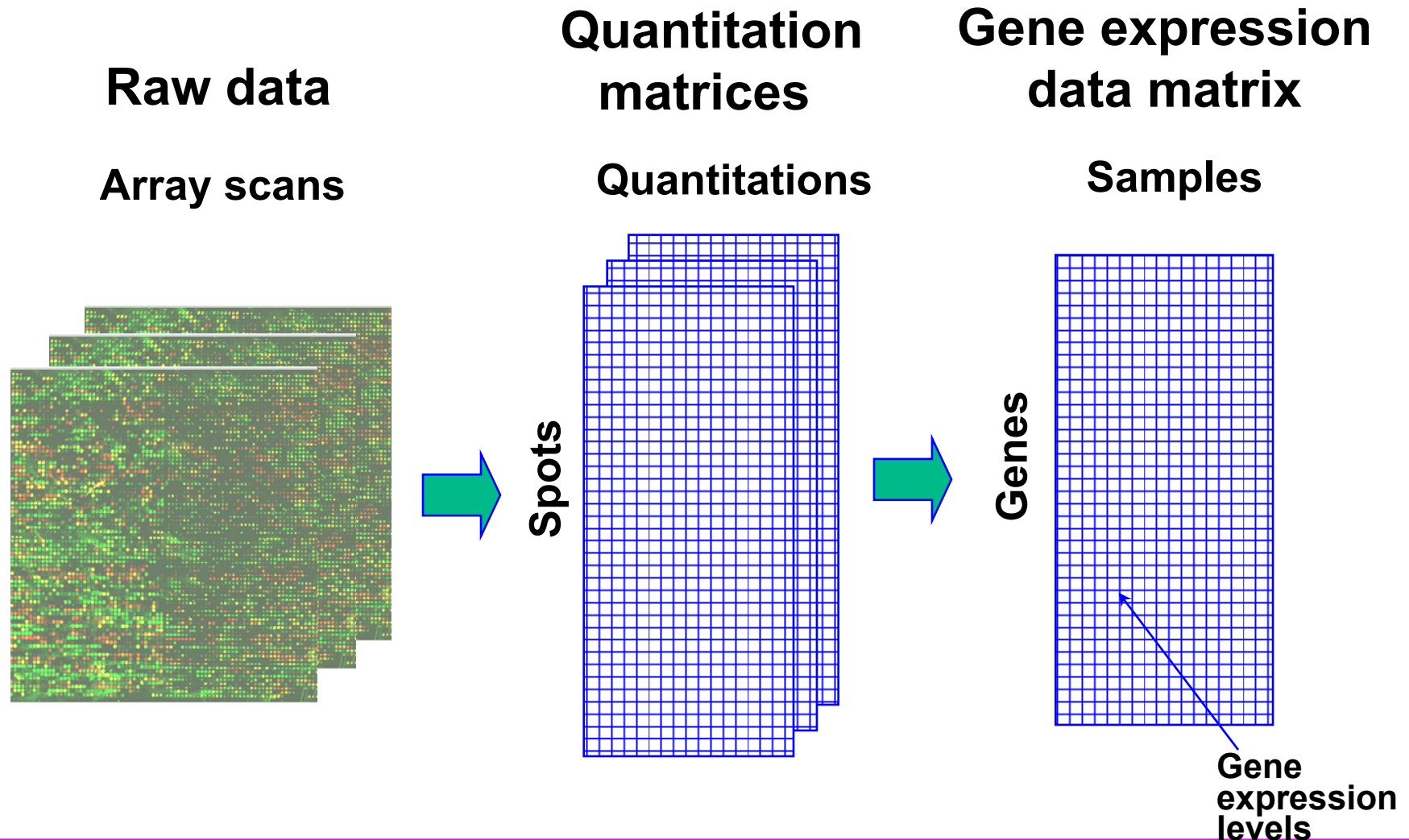


An Example Matrix

(not real patient data)

dtrans1	trans1	dbili1	bili1	dtrans2	trans2	dbili2	bili2	dtrans3	trans3	dbili3	bili3	group
0.92	0.99	-0.66	-0.66	-0.37	0.18	-1.12	-0.99	-0.39	-0.05	-0.85	-1.02	0.00
-0.28	-0.01	-0.07	-0.52	-0.34	-0.08	-0.73	-0.84	-0.42	3.00	-0.45	-0.72	0.00
-0.51	0.75	0.23	2.77	-0.13	-0.11	1.51	2.88	-0.22	-0.21	0.39	2.55	0.00
-0.66	0.15	-0.59	-1.27	-0.33	0.08	-0.61	-1.30	-0.41	-0.08	-0.35	-1.09	0.00
0.40	-0.18	-0.56	-1.53	-0.32	-0.14	-0.61	-1.59	-0.40	-0.16	-0.45	-1.49	0.00
0.92	1.17	-0.33	-1.48	-0.33	0.11	-0.38	-1.46	-0.40	-0.09	-0.28	-1.32	0.00
-0.28	-0.35	-0.13	1.03	-0.02	-0.31	-0.45	0.89	0.49	-0.41	-1.18	0.44	0.00
-0.12	0.75	-0.43	-1.62	-0.30	0.03	-0.40	-1.54	-0.38	-0.11	-0.38	-1.55	0.00
0.09	-0.93	-1.22	0.47	-0.39	-0.29	-1.59	0.20	-0.42	-0.35	-1.05	0.18	0.00
-0.51	-0.48	-0.59	-0.70	-0.13	-0.29	-0.84	-0.88	-0.08	-0.36	-0.69	-0.93	0.00
0.92	0.44	-0.46	-0.25	-0.38	0.03	-1.26	-0.67	-0.39	-0.11	-0.85	-0.63	0.00
0.09	1.08	0.26	0.46	-0.24	0.00	-0.15	0.34	-0.20	-0.24	-0.47	0.14	0.00
-0.28	1.08	-1.78	-0.46	-0.35	3.00	-0.45	-2.24	-0.48	0.11	-1.09	-0.58	0.00
-0.73	1.49	-0.49	-1.11	-0.26	0.27	-0.52	-1.12	-0.42	0.32	-0.38	-1.04	0.00
0.40	0.33	-0.20	-0.51	-0.41	0.03	0.20	-0.29	-0.45	0.02	-0.54	-0.73	0.00
2.01	0.83	-3.00	0.30	-0.35	0.14	-1.45	0.55	-0.40	0.00	-0.74	0.61	0.00
0.40	0.56	-0.59	0.08	-0.25	-0.14	-0.91	-0.10	-0.39	-0.15	-0.86	-0.25	0.00

Generating Data Matrices from Data



R, S and S-plus

S: an interactive environment for data analysis and a statistical programming language developed since 1976 primarily by John Chambers

Exclusively licensed by *AT&T/Lucent* to *Insightful Corporation*, Seattle WA. Product name: “S-plus”.

R: initially written by Ross Ihaka and Robert Gentleman during 1990s.

Since 1997: international “R-core” team of ca. 15 people with access to common CVS archive.

GNU General Public License (GPL), Open Source

What R does and does not

- data handling and storage:
numeric, textual
- matrix algebra
- hash tables and regular expressions
- high-level data analytic and statistical functions
- classes (“OO”)
- graphics
- programming language:
loops, branching, subroutines
- is not a database,
but connects to DBMSs
- has no graphical user interfaces, but connects to Java, TclTk
- language interpreter can be very slow, but allows to call own C/C++ code
- no spreadsheet view of data, but connects to Excel/MsOffice
- no professional / commercial support

R and statistics

- **Packaging: a crucial infrastructure to efficiently produce, load and keep consistent software libraries from (many) different sources / authors**
- **Statistics: most packages deal with statistics and data analysis**
- **State of the art: many statistical researchers provide their methods as R packages**

S Language Elements

- **Variables**
- **Missing values**
- **Functions and operators**
- **Vectors and arrays**
- **Lists**
- **Data frames**
- **Programming: branching, looping, subroutines**
- *apply*

Vectors, matrices and arrays

vector: an ordered collection of data of the same type

```
> a = c(1,2,3)
```

```
> a*2
```

```
[1] 2 4 6
```

Example: the mean spot intensities of all 15488 spots on a chip: a **vector** of 15488 numbers

matrix: a rectangular table of data of the same type

Example: the expression values for 10000 genes for 30 tissue biopsies: a matrix with 10000 rows and 30 columns.

array: 3-,4-,...dimensional matrix

Example: the red and green foreground and background values for 20000 spots on 120 chips: a 4 x 20000 x 120 (3D) array.

Data Frames Store Clinical/Biological Data Sets

data frame: is supposed to represent the typical data table that researchers come up with – like a spreadsheet.

It is a rectangular table with rows and columns; data within each column has the same type (e.g. number, text, logical), but different columns may have different types.

Example:

> a

	localization	tumorsize	progress
XX348	proximal	6.3	FALSE
XX234	distal	8.0	TRUE
XX987	proximal	10.0	FALSE

apply

```
apply( array, margin, function )
```

Applies the function `function` along some dimensions of the array `array`, according to `margin`, and returns a vector or array of the appropriate size.

```
> x
```

	[,1]	[,2]	[,3]
[1,]	5	7	0
[2,]	7	9	8
[3,]	4	6	7
[4,]	6	3	5

```
> apply(x, 1, sum)
```

```
[1] 12 24 17 14
```

```
> apply(x, 2, sum)
```

```
[1] 22 25 20
```

Data Frame Example (not real patient data)

\$"alk phos"

```
[1] 984 254 237 258 857 807 439 329 254 237 171 197 157 141 154  
[16] 140 157 228 248 415 954 594 733 834 1785 3124 3582 3820 3459 3223  
[31] 2259 2549 2111 1652 1098 1057 1098 1219 1803 1592 1525 943 1340 3268 4614  
[46] 5900
```

\$alt

```
[1] 26 63 360 141 179 44 28 21 27 22 19 19 14 17 18 27 22
```

\$\$"JHU Hb"

```
[1] 14.6 10.0 10.3 11.3 14.1 12.9 11.8 10.3 10.8 10.4 9.5 9.7 9.5 9.1 8.4  
[16] 7.5 8.6 8.6 7.0 5.9 7.8 8.7 10.2 8.1 7.9 11.1 10.9 11.8 12.1 12.9  
[31] 12.6 12.3 11.7 12.3 11.7 12.6 13.1 13.1 11.4 9.6 10.0 7.6 7.1 8.0 9.3  
[46] 8.8
```

\$"JHU ICE COMBO"

```
[1] NA NA NA
```

\$"neut absolute"







```
[1] 2.250 1.030 1.680 0.983 0.740 0.854 0.981 0.785 1.060 0.857 0.570 3.600  
[13] 2.690 2.900 1.100 1.100
```

\$platelet

```
[1] 220 202 317 222 194 159 180 273 268 172 80 47 223 241 93 26 163 130 35  
[20] 25 22 57 179 31 85 171 211 112 156 131 137 110 100 86 100 112 157 141  
[39] 125 105 86 84 73 30 30 13 26 22
```

Ontologies for Events and Time Intervals

- Temporal Description Logic²
 - 13 basic temporal interval relations (Allen notation)

Relation	Abbr.	Inverse	i	j
before(i, j)	b	a		
meets(i, j)	m	mi		
overlaps(i, j)	o	oi		
starts(i, j)	s	si		
during(i, j)	d	di		
finishes(i, j)	f	fi		

²A. Artale and E. Franconi. "A temporal description logic for reasoning about actions and plans". *Journal of Artificial Intelligence Research*, 9:463--506, 1998

Example: Concept “Clinical Type II Rejection”

- **Type-II-Rejection:**
OVERLAPS(Bili_Fever,
 UNION(Int(“GOT=increase”),
 Int(“GPT=increase”)),
 “days”) AND
OVERLAPS([4,21], Bili_Fever, “days”)
RESULT:
Start(Bili_Fever),Finish(Bili_Fever)
- **Bili_Fever:**
DURING(Int(“MaxTemp=Fever”),
 High_Bili_Increase, “days”)
RESULT:
Start(High_Bili_Increase),Finish(High_Bili_Increase)
- **High_Bili_Increase:**
During(Int(“Bilirubin=high”),
 Int(“Bilirubin=increase”),”days”)
RESULT:
Start(Bili_Increase),Finish(Bili_Increase)

Retrieve all occurrences of patient episodes, where the interval representing increase of bilirubin with at least partly fever episodes overlaps an interval representing an increase of transaminases (GOT or GPT) within day 4 and day 21 after liver transplantation.

This concept is characterized by the interval of bilirubin increase.

The concept bili_increase represents occurrences with values at least partially over 100 umol/l

Ontology Example

Concept

conceptual entity of the domain

Property

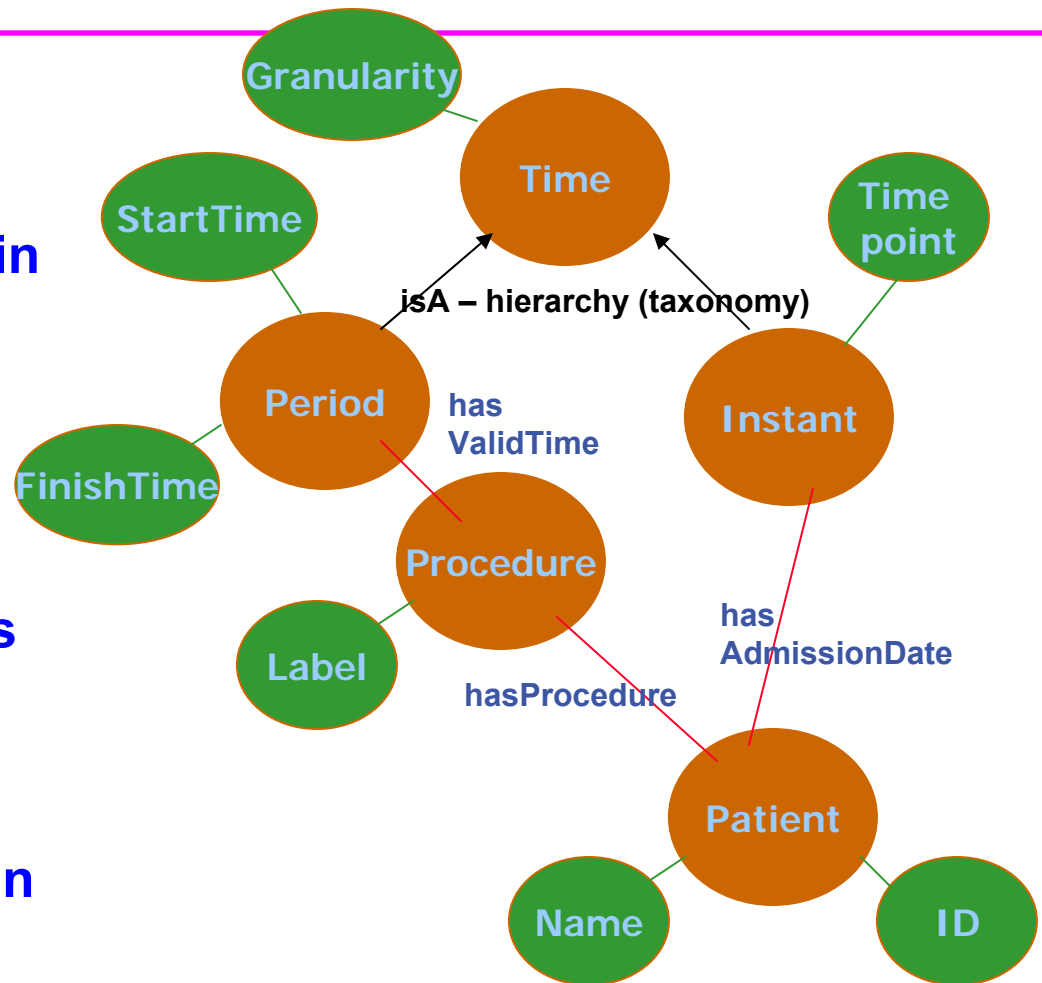
attribute describing a concept

Relation

relationship between concepts or properties

Axiom

coherency description between Concepts / Properties / Relations via logical expressions

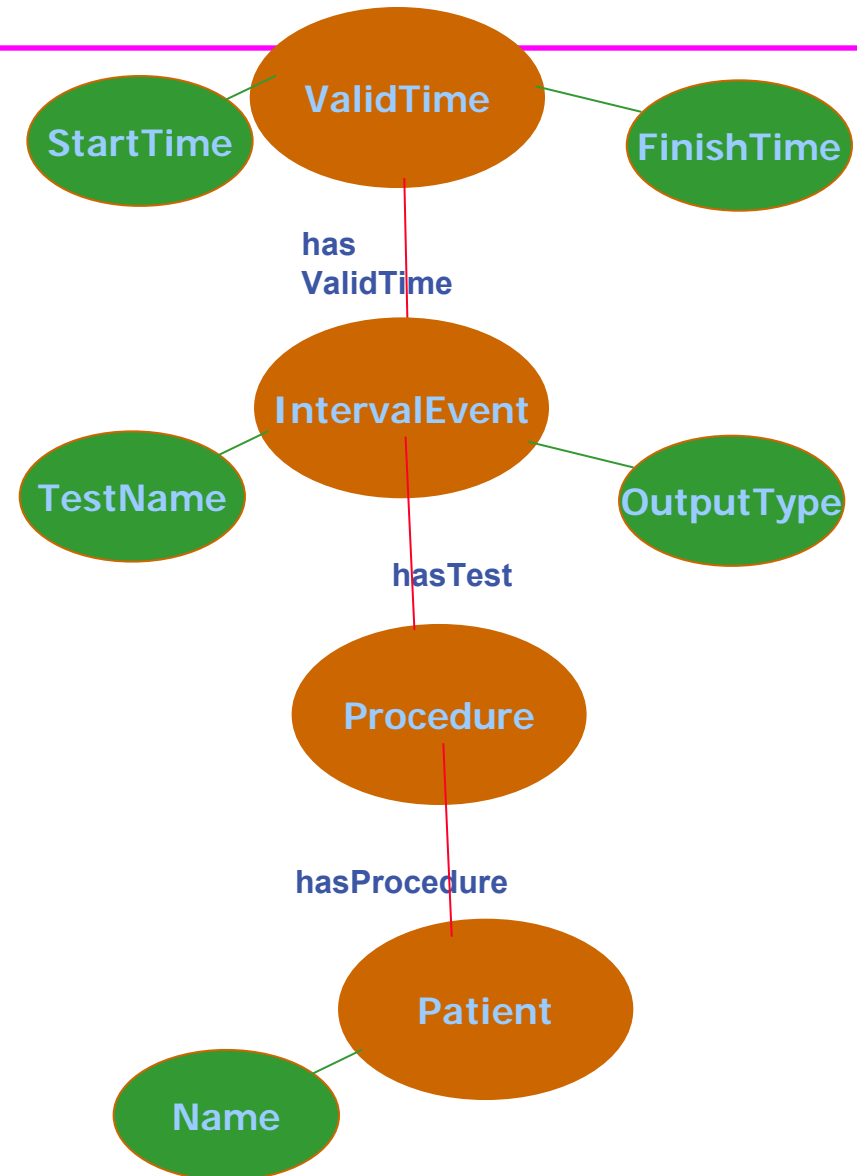


hasAdmissionDate(Patient)
=> admitted(Patient)

SWRL

High and Increasing Bilirubin

```
Patient(?p) □  
hasProcedure(?p, ?proc) □  
  hasTest(?proc, ?test) □  
    hasTestName(?test, ?testName) □  
    swrlb:equal(?testName, "BILIRUBIN") □  
    HasOutputType(?test, ?testType) □  
    swrlb:equal(?testType, "INCREASE") □  
    temporal:hasValidTime(?test, ?tVT) □  
hasTest(?proc, ?test2) □  
hasTestName(?test2, ?testName2) □  
swrlb:equal(?testName2, "BILIRUBIN") □  
HasOutputType(?test2, ?testType2) □  
swrlb:equal(?testType2, "HIGH") □  
temporal:hasValidTime(?test2, ?tVT2) □  
  temporal:overlaps(?tVT, ?tVT2, "days") □  
temporal:hasStartTime(?tVT, ?stTime) □  
temporal:hasFinishTime(?tVT, ?fiTime) □  
swrlx:createOWLThing(?hbVT, ?proc) □  
->temporal:ValidPeriod(?hbVT) □  
temporal:hasStartTime(?hbVT, ?stTime) □  
temporal:hasFinishTime(?hbVT, ?fiTime) □  
  hasHighBiliIncrease(?proc, ?hbVT)
```



Discussion



- Proof of concept
- SPOT is a feasible approach to use open source and standards based software
- Different solutions to “translate” logic from OWL/SWRL into S
- Currently, concept intervals are passed from OWL/SWRL through the Java interface and “relearned” through a classification tool in R, e.g., discriminant analysis.
- SWRL interface improved with modularization since object instantiation is possible
- Need of GUI for researcher

Acknowledgements

Thank you

- Mark Musen
- Tania Tudorache
- Samson Tu
- Ted Hopper
- The Protégé Team at Stanford



Thank you
for your attention