Exploring Microarray Time Series with Protégé

Guenter Tusch[1], Xiaohui Huang[1], Martin O'Connor[2], and Amar Das[2]
[1]Grand Valley State University, Allendale, MI, USA
[2] Stanford Medical Informatics, Stanford University, Stanford, CA, USA
tuschg@gvsu.edu, huangx@student.gvsu.edu,
{martin.oconnor, das}@stanford.edu

Abstract. Data mining of microarray data more often includes exploration of temporal data. E.g., researchers might look for temporal changes in gene expression to determine responses to an injury. Knowledge-based Temporal Abstraction does not assume that the pattern of time points is similar to the original experimental design. We extended this framework by creating an software tool using open-source platforms. It supports the R statistical package and knowledge representation standards (OWL, SWRL) using the open source Semantic Web tool Protégé-OWL.

1 Introduction

Advances in microarray technology have led to highly complex datasets often addressing similar are related biological questions. The statistical methodology of meta-analysis aims to combine results from independent but related studies. It is a relatively inexpensive option that has the potential to increase both the statistical power and generalizability of single-study analyses [1]. For example, a meta-analysis of five circadian microarray studies of Drosophila helped researchers to identify a novel set of rhythmically expressed genes [2].We advocate here a related approach to potentially extend confirmed results to other species or organs (translational research). If the researcher is not so much interested in confirmation of statistical differences but in exploring how established temporal patterns in one experiment or species translate into discovery of those pattern in non-related but similar experiments, he can select interesting experiments in an analogous way [1] and look for the same temporal patterns. We developed a tool that can be utilized for this kind of exploratory analysis of time series microarray data, e.g., rhythmically expressed genes as in circadian microarray studies or temporal changes in gene expression to determine reactive responses to a particular stimulus (e.g., the GEO GDS656 data set that we used as a pilot includes the gene expression response to retina injury in the rat). These studies become available for analysis in increasing numbers, we found in GEO alone more than 200 studies with at least five time points for the stimulus response study. The tool we developed (SPOT) is an implementation using open source and standardized tools: the Web Ontology Language (OWL; `http://www.w3.org/TR/owl-features`), the Semantic Web Rule Language (SWRL; `http://www.daml.org/2003/11/swrl`), Protégé plug-ins; `http://protege.stanford.edu/`, and open source statistical software (R; `http://www.r-project.org/`).

2 Methodology

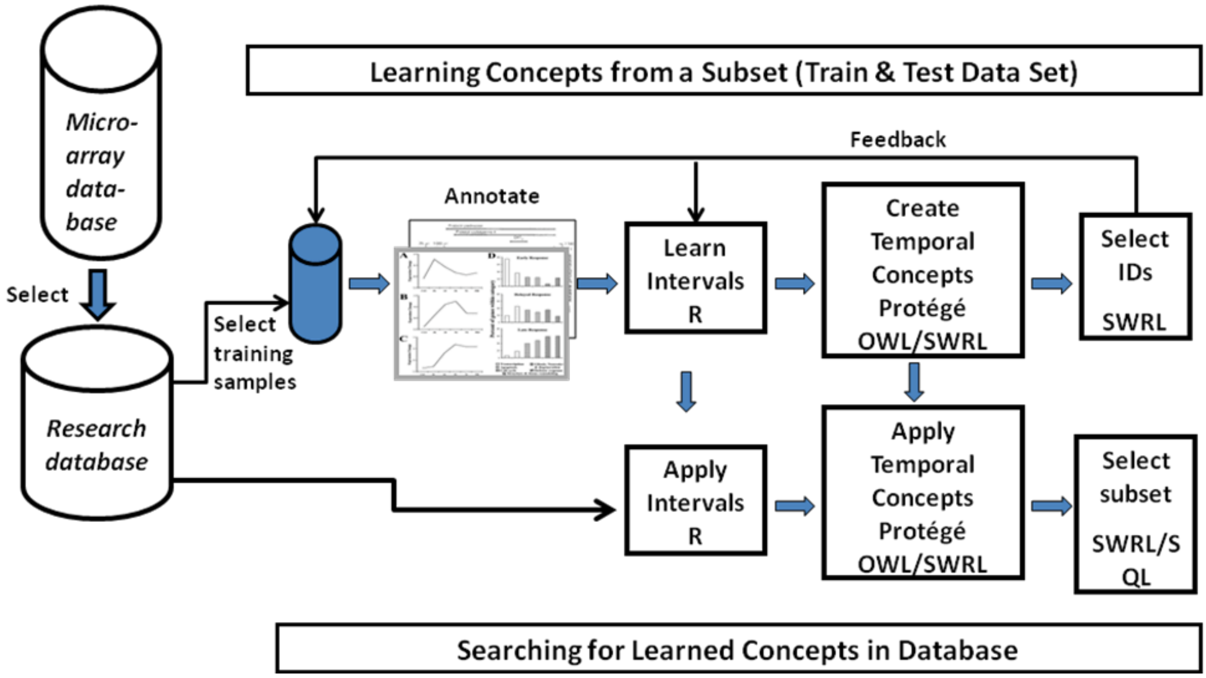
Traditional approaches to clustering gene expression temporal data include PCA, Pearson correlation, or software packages like GQL, CAGED, or STEM. Typically, for stimulus response microarray studies a researcher obtains a fold change expression profile and tries to retrieve similar profiles in one of the many web-accessible microarray databases like the US-based NCBI GEO, its European counterpart ArrayExpress, or clinical databases (that more frequently include microarray data). One frequent approach is to search for highly correlated profiles. We could show for an example data set (GEO GDS656) that this is a reasonable approach if you use the same experimental design (sensitivity and specificity in between 90% and 98%). However, this approach assumes that the pattern of time points in all selected experiments is identical or at least very similar to the experimental design of the initial study. Some meta-analyses have tried to solve this dilemma by using only experiments with similar design and only the intersection of all time points or a selection with only the data available at that particular time, see e.g. Keegan[2]. This is most likely only feasible if the studies for a few exceptions, but most likely will exclude a more general translational bioinformatics approach. For this approach, it is more realistic to look, for instance, for a peak in the profile instead of correlating the entire profile. This can be accomplished by using Knowledge-based Temporal Abstraction [4], where time-stamped data points are transformed into an interval-based representation. These intervals then form the basis for temporal concepts like peaks. We integrated this framework into a software tool based on an open-source platform, SPOT (see also [3]). It supports the R statistical package and knowledge representation standards (OWL, SWRL) using the open source Semantic Web tool Prote´ge´-OWL.

3 Temporal Data Mining of Microarrays

An overview over the mining process with SPOT is depicted in figure 1. The user has to perform the following steps: He/she selects interesting studies, e.g., from the GEO database, that are stored in a research database. A training sample (including all applicable array types) is selected from there. Then he/she selects algorithms, that allow the system to train patterns to recognize, for instance, an increase in fold changes of the temporal expression data regarding one particular gene or a group of genes. The program generates R macros and OWL/SWRL code. SWRL allows users to write rules that can reason about OWL individuals. The Prote´ge´ OWL plug-in allows to easily building ontologies. The researcher (user) can define different gene expression profile peaks, e.g., "Early", "Delayed", or "Late" in the time course, and search for similar profiles in the database of interesting studies. Estimation of intervals from a learning sample, e.g., learning thresholds for increasing fold change values for one array type, e.g., Affymetrix, in order to model an "increase" interval, building of high level concepts using temporal abstraction (implemented in Protégé-OWL/SWRL Tab), and validation of the generated intervals. The statistical tasks (learning the thresholds) are implemented in R.

The user might go through that process several times until the classification error for the biological concepts he/she models is sufficiently small (feedback loop). Adjustments can be made by changing thresholds or adding additional constraints to the SWRL concepts. When the user is satisfied with the performance, the entire research database is transformed using the selected R-macros, the resulting intervals submitted to Protégé and gene IDs selected with the user-created SWRL rule abstractions.

Besides using the time stamped data from the microarray research database, the user needs to identify intervals for the training data only, one gene expression profile at a time (annotate step in figure 1). Several different non-overlapping intervals are allowed, i.e. the user mark intervals for the gene as "increasing", "decreasing", and "high" in one step. The interval value is attached to the time-stamped gene expression value.



Learning Concepts from a Subset (Train & Test Data Set)
Micro-array data-base
Select
Research database
Select training samples
Feedback
Annotate
Learn Intervals R
Create Temporal Concepts Protégé OWL/SWRL
Select IDs SWRL
Apply Intervals R
Apply Temporal Concepts Protégé OWL/SWRL
Select subset SWRL/SQL
Searching for Learned Concepts in Database

Fig. 1. The SPOT (***S*** – ***P***rotégé – ***O***WL/SWRL – ***T***emporal Abstraction) mining process.

4 Implementation Details

For the implementation we use the Semantic Web tool Protégé-OWL and the SWRLTab [3] including temporal built-ins (`http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTemporalBuiltIns`) and extensions built-ins (`http://protege.cim3.net/cgi-bin/wiki.pl?SWRLExtensionsBuiltIns`). For a previous version of SPOT for clinical data see [3]. Ontologies are used in OWL to formally specify meaning of annotations by providing a vocabulary of terms. New terms can be formed by combining existing ones.

SWRL allows users to write rules that can be expressed in terms of OWL concepts and that can reason about OWL individuals. In the SPOT application, we use a temporal ontology implementing the valid time model, and a hierarchical patient ontology with classes: Patient (has) Procedure (has) Interval/Event (has) Valid Time.

As an example from the GEO GDS656 data set, we describe a concept of an “early peak” in fold change expression data of one particular gene and one array type. The concept is described in terms of SWRL code in figure 2, using SWRL built-ins and extensions. Each concept creates a “result interval”, i.e., an interval instance in the valid time model, using the SWRL extension `createOWLThing`. This allows for modularization of time concepts, i.e., breaking down concepts into (potentially in other clinical domains re-usable) sub concepts.

```
Explanation           SWRL - Code

init variables          Patient(?p)  ∧
 to capture the         hasProcedure(?p, ?proc)  ∧
underlying                  hasTest(?proc, ?test)  ∧
ontology                           hasTestName(?test, ?gene)  ∧
increasing                  HasOutputType(?test, ?testType)  ∧
interval part               swrlb:equal(?testType, "INCREASE")  ∧
of peak→?tVT                temporal:hasValidTime(?test, ?tVT)  ∧
                        hasTest(?proc, ?test2) ∧
                        hasTestName(?test2, ?gene2)  ∧
decreasing              swrlb:equal(?gene2,?gene) ∧
interval part           HasOutputType(?test2, ?testType2)  ∧
of peak→?tVT2           swrlb:equal(?testType2, “DECREASE")  ∧
meets intervals         temporal:hasValidTime(?test2, ?tVT2) ∧
?tVT and ?tVT2              temporal:meets(?tVT, ?tVT2, "days")∧
Length≤1 day            swrlb:lessThanOrEqual(?finishTime, 1)  ∧
save ?startTime         temporal:hasStartTime(?tVT, ?startTime)  ∧
and                     temporal:hasFinishTime(?tVT2, ?finishTime)∧
?finishTime             swrlx:createOWLThing(?hbVT, ?proc)
 variables              ->  temporal:ValidPeriod(?hbVT) ∧
create                      temporal:hasStartTime(?hbVT,?startTime)∧
result interval             temporal:hasFinishTime(?hbVT,?finishTime)∧
and attach                  hasEarlyPeak(?proc, ?hbVT)
```

Fig. 2. *SWRL rule for the concept “Early Peak”*
(`?tVT`, `?tVT2`, and `?hbVT` are interval instances in the valid time model, `temporal:` denotes temporal built-ins based on Allen’s temporal relationships[6], `swrlx:` denotes SWRL extensions.)

5 Conclusion and Future Aspects

We developed tools to support this process using the Protégé-OWL ontology development toolkit (compare [5]), which was tested using GEO GDS656 as the reference study. The newly created GUI using the R and Protégé APIs allows now for easier access and manipulation by the user, especially creating macro like constructs that generate part of the SWRL code automatically. The system is currently tested by users to help improve the GUI.

Acknowledgments. We acknowledge the valuable input and support by Timothy Redmond, PhD, Tania Tudorache, PhD, Jeremy Miller, PhD, Craig Webb, PhD, and Mark Musen, MD, PhD.

References

1. Ramasamy A, Mondry A, Holmes CC, Altman DG. Key Issues in Conducting a Meta-Analysis of Gene Expression Microarray Datasets. PLoS Med (2008) 5(9).
2. Keegan KP, Pradhan S, Wang J-P, Allada R, (2007) Meta-Analysis of Drosophila Circadian Microarray Studies Identifies a Novel Set of Rhythmically Expressed Genes. PLoS Comput Biol 3(11)
3. Tusch, G., O’Connor, M., Redmond, T., Shankar, R., Das, A. (2007). The Protege-OWL SWRL Tab and Temporal Data Mining in Surgery. Proceedings 10th International Protege Conference, Budapest, Hungary.
4. Shahar, Y., Musen, M.A.: Knowledge-based temporal abstraction in clinical domains. Artif. Intell. Med. 8(3) (1996) 267-98.
5. O'Connor MJ, Shankar RD, Parrish DB, Das AK. Knowledge-Data Integration for Temporal Reasoning in a Clinical Trial System. Int J Med Inform. In Press in 2009 [Epub ahead of print] .
6. Allen, J.F.: Towards a general theory of action and time. Artif. Intell. 23(2) (1984) 123-154.