BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 10 2005, pages 2544–2545
doi:10.1093/bioinformatics/bti311

Gene expression

The Graphical Query Language: a tool for analysis of gene expression time-courses

Ivan G. Costa[1], Alexander Schönhuth[2] and Alexander Schliep[1,*]

[1]Max Planck Institute for Molecular Genetics, Department of Computational Molecular Biology, Ihnestrasse 73, 14195 Berlin, Germany and [2]Center for Applied Computer Science, University of Cologne, Weyertal 80, 50937 Cologne, Germany

Received on January 05, 2004; accepted on February 7, 2005
Advance Access publication February 8, 2005

ABSTRACT

Summary: The Graphical Query Language (GQL) is a set of tools for the analysis of gene expression time-courses. They allow a user to pre-process the data, to query it for interesting patterns, to perform model-based clustering or mixture estimation, to include subsequent refinements of clusters and, finally, to use other biological resources to evaluate the results. Analyses are carried out in a graphical and interactive environment, allowing expert intervention in all stages of the data analysis.

Availability: The GQL package is freely available under the GNU general public license (GPL) at http://www.ghmm.org/gql

Contact: schliep@molgen.mpg.de

INTRODUCTION

Our application addresses the analysis of gene expression time-courses by identifying biologically relevant groups of genes undergoing the same transcriptional program. As the knowledge discovery process in the analysis of biological data is human-centric, a high degree of interactivity is an important characteristic of the Graphical Query Language (GQL). What we have striven for is a set of application tools which lets a user visualize and analyze time-course data interactively, evaluate hypotheses about the data and compare the results with other sources of biological data. GQL allows to integrate prior knowledge and it maintains a high degree of robustness with respect to noise and missing data, in order to arrive at unambiguous groups of time-courses. The main contributions of our method is the use of linear hidden Markov models (HMMs) to represent groups of genes showing the same qualitative behavior and their combination into a classical mixture model; this has been shown to be an intuitive and meaningful choice (Schliep *et al.*, 2003, 2004).

SOFTWARE DESCRIPTION

GQL is divided into two main applications: GQLQuery and GQLCluster. GQLQuery allows the user to either create a new HMM or load an existing one in order to query a set of time-courses for interesting temporal patterns (Fig. 1). Modifications of the model's parameters in the tool interface are simultaneously reflected in the time-courses panel. By changing the similarity rank threshold, the user can control the stringency of the query, and thus select only those time-courses which have been queried by the model with high probability. The modified models and query results can be saved for later analyses.

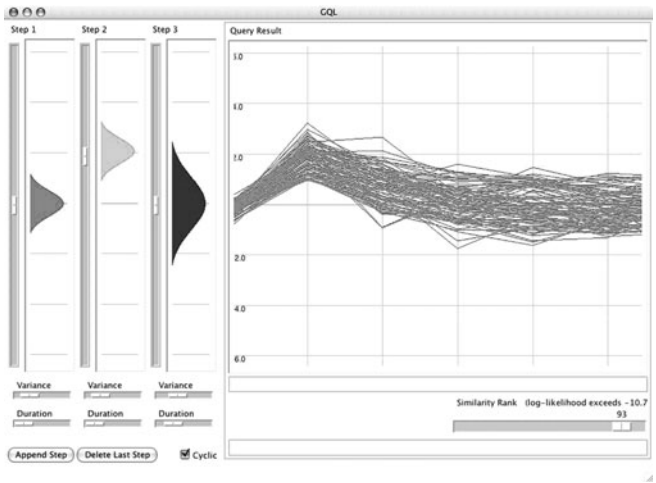

Fig. 1. The GQLQuery interface is divided into two main components: the left part is the model editor, where the user can view and change the model's parameters, and in the right part is the query result, where the queried time-courses are displayed.

GQLCluster implements the methods for estimating clusterings or mixtures of time-courses and for the post-analysis of the results. As a first step, the time-courses can be filtered to exclude non-expressed genes. GQLCluster provides an n-fold filter and a non-constant filter. There is no need to do any pre-processing concerning missing data, since the estimation methods can deal internally with these values. After filtering, different estimation procedures can be applied to find interesting groups of genes in the data. As model-based estimation procedures require the provision of an initial collection of models, GQLCluster has implemented three easy-to-use and well-justified methods for this. They can be either defined by the user, e.g. through saving query models from GQLQuery, randomly generated or estimated from the input data. In the case of randomly generated models, the Bayesian information criteria (BIC) can be used to infer a plausible number of components. Subsequent to the creation of an initial model set, three types of estimations are applicable

*To whom correspondence should be addressed.

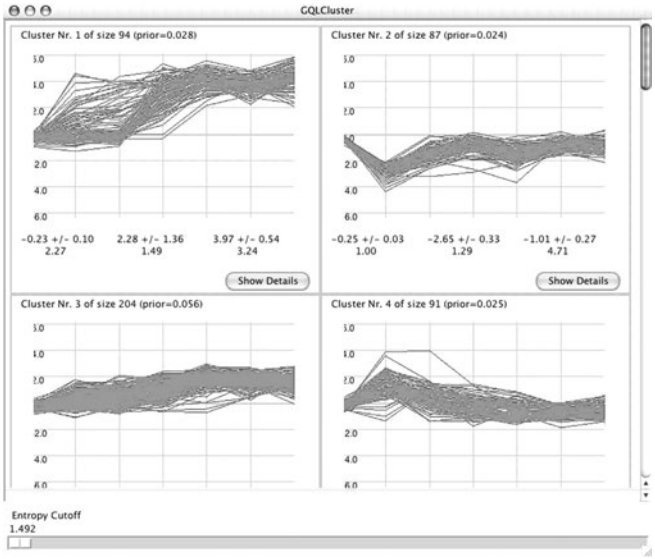

Fig. 2. After estimation, GQLCLuster displays the time-courses assigned to each cluster. The user can then do more detailed inspection of the modules, such as looking for gene annotation in known databases, inspect for GO enrichment or compute a sub-grouping.

in GQLCluster: first, clustering estimation, where the time-courses are uniquely assigned to one model; second, mixture estimation, where the time-courses have a probability of being assigned to each model; and third, mixture estimation additionally using labeled data. The last method allows the user to include prior knowledge in the estimation process in a partially supervised approach (Schliep *et al*., 2004).

After the estimation has been successfully carried out, GQLCluster offers several tools for the analysis of the results. As a starting point, the graphical interface creates panels, which contain the time-courses of each of the clusters/components (Fig. 2). Then, for each cluster, it is possible to inspect the list of gene identifiers, which are linked to known web databases, or look for enriched GO terms through an external link to the web tool GOStat (Beissbarth and Speed, 2004). In the mixture estimation case, the time-courses are assigned to the most likely model. The user can choose only genes that can be unambiguously assigned to one model by increasing the entropy cut-off threshold. By the inspection of probability distributions of the time-courses over the models, it is also possible to find genes interacting in more than one context. A further refinement of the clusters can be obtained by the application of a Viterbi decomposition analysis, which finds sub-groups of synchronous time-courses.

Another feature of GQLCluster is the use of other sources of biological data to evaluate the groupings. Currently only annotations from gene ontology are supported, but further classes of data such as gene regulation or protein–protein interactions are in preparation. It provides a number of statistics such as sensitivity, specificity and corrected Rand as well as a contingency table allowing the user to find correlations between the groupings and gene annotation. These statistics are also available when benchmark data is given. Furthermore, a procedure for finding an 'optimal' entropy cut-off threshold given a gene annotation dataset is provided, by finding a threshold value, which maximizes the specificity of annotations. All results, estimated models and graphs can be saved for subsequent use.

IMPLEMENTATION

The graphical interface and high-level implementations of the methods are implemented in Python. It is also possible to access the GQL functionality through Python scripts, e.g. experiments requiring more computational time. GQL is based on GHMM, a C-library for HMMs (GHMM, 2003, http://www.ghmm.org). The tools run on most platforms (Unix, Linux, MacOS X and Windows) and require GHMM, Python2.3, Swig 1.3.17, GSL and PYGsl. A tutorial, detailed installation instructions and sample data can be found at http://www.ghmm.org/gql

ACKNOWLEDGEMENTS

Thanks to Christine Steinhoff for her valuable contributions during the development of the method. The authors would like to acknowledge funding from the DAAD/CNPq (Brazil) and the BMBF through the Cologne University Bioinformatics Center (CUBIC). Thanks also to Wasinee Rungsarityotin, Benjamin Georgi, Xue Li, Olof Persson and Tim Beissbarth.

REFERENCES

Beissbarth,T. and Speed,T.P. (2004) GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics*, **20**, 1464–1465.

GHMM (2003) The General Hidden Markov Model library.

Schliep,A. *et al*. (2003) Using hidden markov models to analyze gene expression time course data. *Bioinformatics*, **19**, i255–i263.

Schliep,A. *et al*. (2004) Robust inference of groups in gene expression time-courses using mixtures of HMMs. *Bioinformatics*, **20**(Suppl. 1), i283–i289.