Comparative Studies Made Simple in GPFlow

Lawrence Buckingham, James M. Hogan, Paul Roe, Jiro Sumitomo, Michael Towsey
Microsoft QUT eResearch Centre, Queensland University of Technology
{l.buckingham,j.hogan,p.roe,j2.sumitomo,m.towsey}@qut.edu.au

1. Introduction

Bioinformatic sequence data is typically analyzed via a pipeline of tools which may be realized manually or through some kind of script or workflow system. The explosive increase in the number of genomes available has made single sequence analyses almost obsolete. Bioinformaticians now wish to compare and analyze multiple versions of similar sequences, and the greater statistical significance afforded by automated comparisons is vital to scientific investigation.

This work describes recent extensions to the GPFlow scientific workflow system [1] in development at MQUTeR (www.mquter.qut.edu.au), which facilitate interactive experimentation, automatic lifting of computations from single-case to collection-oriented computation and automatic correlation and synthesis of collections. A GPFlow workflow presents as an acyclic data flow graph, yet provides powerful iteration and collection formation capabilities.

2. Data driven iteration

GPFlow allows the user to supply a set of values for each user input to the workflow. The workflow iterates over the Cartesian product of the user input sets, executing once for each combination. One implementation of this model is obvious, namely, to execute the entire workflow within a nested loop. While such a brute force procedure is trivial to implement, it is easy to construct workflows where such a regime would introduce unnecessary duplication of steps.

Instead, each workflow step consumes collections of input values and produces a corresponding collection of output values: the step iterates over a filtered subset of the Cartesian product of the input collections, producing one result for each combination. Each result is assigned a distinct key that encodes its provenance, and these keys are used to constrain the iteration, maintaining the structural integrity of the workflow.

3. Collection formation

Data driven iteration enables the generation of collections of disparate values. To process these collections as a whole, we provide two consolidation mechanisms: aggregation and key-slicing.

Aggregation is similar in operation to the familiar dot product operation of linear algebra. We merge the elements of two or more parallel arrays to form a single array of tuples. Key-slicing is more closely related to the “group by” clause of SQL. Here we collect some or all values produced by a component to perform some synthesis or summarizing operation.

4. Conclusions

The problem of managing collections arising in comparative studies is fundamental to post-genome bioinformatics. This work introduces a novel key based approach to tracking data tuples, ensuring correctness and enabling convenient selection from the available Cartesian product of data vectors. In addition to the guarantee of result integrity, the approach provides a ready made platform for provenance tracking and reporting. Above all, the system supports automated lifting of computations, allowing the user conveniently to prototype singleton computations and routinely apply them to full scale data sets.

5. References


We gratefully acknowledge the support of Microsoft Research, the Queensland Government and QUT.