Analysis of a Scientific Protocol: Selecting Suitable Resources

Zoé Lacroix, Christophe Legendre
Scientific Data Management Lab., Arizona State University
PO Box 875706, Tempe AZ 85287-5706, USA

Abstract
We analyze a scientific protocol developed to support a service devoted to alternative splicing analysis. We explain the process of selection of resources to implement each scientific task and present a methodology for scientific tools analysis and benchmarking to produce sound and efficient scientific protocols.
Keywords scientific protocol, workflow, scientific benchmark, semantic benchmark, sequence alignment benchmark.

1. Introduction

Public biological resources form a complex maze of heterogeneous data sources, interconnected by navigational capabilities and applications. Although this wide and valuable network offers scientists multiple options to implement and execute their scientific protocols, selecting the resources suitable to access and exploit the data of interest remains a tedious task. When designing a scientific protocol, they struggle to consolidate the best information about the scientific objects being studied and to implement it in terms of queries against biological resources. These protocols are typically implemented using the resources the scientist is most familiar with, instead of the resources that may best meet the protocol's needs. This implementation-driven approach to express scientific protocols may significantly affect the outcome of a scientific experiment. Indeed similar resources (e.g., alignment tools) do not all perform efficiently on the dataset of interest, or resources that seem similar may return dramatically different outputs [1].

In this paper we analyze a scientific protocol developed to support two Web services devoted to alternative splicing [2]. We distinguish its *design* that captures the scientific aim of the workflow, from its *implementation* that specifies the resources used to implement it. This two-layer representation of scientific workflows allows the discussion of alternate resources to implement a specific scientific task [3]. We discuss the bottlenecks typically found in implementations and how benchmarks of scientific tools could guide the scientist in the selection of resources suitable to the protocol and improve both the performance of the scientific workflow and the quality of the dataset returned by the workflow with respect to the protocol's needs.

2. Motivating Example: Alternative Splicing Protocol

Alternative Splicing (AS) is a splicing process of a pre-mRNA sequence transcribed from one gene that leads to different mature mRNA molecules thus to different functional proteins. Alternative splicing events are produced by different arrangements of the exons of a given gene. Experimental approaches were first the only way to analyze cellular processes. It still works well but it is a very long and expensive process. With the availability of complete genomes and transcripts published on the Web, it is possible to check whether a transcript belongs to a cluster of different transcripts of a gene. The Alternative Splicing Protocol (ASP) we present in this section is currently supporting the BioInformatics Pipeline Alternative Splicing Services (BIPASS[1]) [2].

2.1 Design of ASP

The Alternative Splicing Protocol takes a set of transcripts as input and returns clusters of transcripts aligned to a gene (see Figure 1). The process of alignment is simple and consists of an alignment of each transcript sequence against each genomic sequence of a whole genome of one or more organisms. This step is executed with all known transcripts extracted from different public databases. A clustering step immediately follows the alignment step. That

[1] The BioInformatics Pipeline Alternative Splicing Services (BIPASS) are available at http://bip.umiacs.umd.edu:8080/.

2007 IEEE Congress on Services (SERVICES 2007)
0-7695-2926-7/07 $25.00 © 2007 IEEE

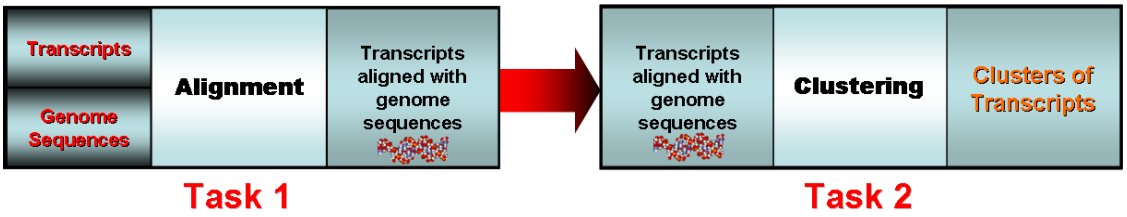

Figure 1: AS Protocol Design

step allows delimiting the transcript region of a gene excluding its regulation region. A cluster normally represents or may be representative of all intermediate transcripts (from the pre-messenger-RNA(s) to the mature messenger-RNA(s)) required to obtain one or several functional translated proteins from the same gene.

2.2 Implementation of ASP

The protocol is intended to support two services: BIPAS-SpliceDB and BIPAS-Align&Splice. The former provides access to a warehouse obtained by executing ASP on public transcript and genomic resources. Because ASP is executed off-line to update the SpliceDB database, the performance of its execution is less critical than the performance of the system that processes the queries. In contrast, BIPAS-Align&Splice executes the complete protocol on-line with transcripts provided by the user. Because ASP will be executed each time the service is used, an analysis of its performance is critical to insure the quality of the service.

3. Implementation issues

A naïve implementation of the protocol is to follow the protocol design that only specifies the scientific objects involved (e.g., transcripts and genome in the protocol design illustrated in Figure 1) and select data sources to provide needed inputs and services to execute each task. Naïve implementations typically fail for *syntactic*, *semantic*, or *efficiency* reasons.

Syntactic reasons are related to the problem of interoperability of resources. A data source may provide the needed input for a scientific task, but the service selected to perform the task requires a different format than the one used by the data source. Standardization efforts such as Web Services [4, 5] and BioMoby [6] provide a framework to communicate in a unified way the description of services and connectors including their input and output formats. Many projects have addressed problems of composition of services [7]. The focus of BioMoby is to provide an ontology-based messaging standard for automatic interaction with biological data, avoiding manual transformation of data formats between services. Workflow systems such as Taverna [7], Remora [8], SemanticBio [9], and Mobyle [10] enable the composition and execution of bioinformatics services. Combining a workflow approach with a service representation that guarantees compatibility of data formats offers a great value to the scientist who has selected the services to use and wishes to combine them in an executable workflow. For example, Taverna now allows the use of any service registered with BioMoby [11]. Although these approaches provide valuable support to the scientists, focusing on format-driven implementation of scientific protocols may lead to poor data quality (protocol semantics) and performance as a service may be chosen because it is compatible instead of its relevance with respect to the scientific aim of the protocol it implements or its efficiency.

Resource selection may affect dramatically to outcome of a scientific protocol [1]. An experiment conducted in February 2005 [12] demonstrates how the selection of resources may result in different datasets. Consider the simple query: *retrieve bibliographic references related to a genomic disorder*. To execute this query, a selection of resources may include OMIM and the PubMed Links provided by NCBI to retrieve PubMed entries related to each of the OMIM entries. These links offer a valuable contribution to the scientists as curated pre-computed joins between heterogeneous data sources. For the disease diabetes, 48,941 entries (without duplicates) were retrieved from PubMed linked to OMIM entries retrieved by conducting a search with 17 keywords related to diabetes. Alternative resource selections include: the same data sources (OMIM and PubMed) but different links, alternative data sources (e.g., an alternate resource for human genetic diseases is

GeneDis[2]), or alternative paths that may include additional resources. An alternative link between OMIM and PubMed consists in parsing the retrieved OMIM entries and extracting all PubMed references. This alternate execution of the same query retrieved 50,843 PubMed entries from the same set of OMIM entries, which is 1,902 more entries than the previous selection. This example illustrates that the adequate selection of resources is critical to the quality and completeness of the retrieved dataset, thus of the experiment. Scientists need to be provided with the ability to distinguish the protocol's aim from its various possible implementations. They also need to be able to compare, analyze, and integrate the results obtained with different implementations.

Finally, efficiency issues may drive the implementation of a scientific protocol. Indeed because of system or application limitations, a resource selected to implement a task may not complete or will take too much time to execute. The efficiency reasons are typically handled by calibrating selected algorithms or filtering significantly the dataflow.

In the following we address the problem of implementing a scientific protocol analysis the performance consequences of the selection of resources.

4. Naïve Implementation of ASP

We study in this section the problem of implementing a complete and efficient protocol in order to support the BIPAS-Align&Splice service that runs the protocol online on transcript inputs provided by the users. The problem of efficiency of the warehouse supporting BIPAS-SpliceDB was presented in [13].

4.1 Selection of data sources

The user submits transcripts and selects one or several organisms made available by the service. Input transcripts include mRNAs (un-spliced Pre-mRNA, in line spliced mRNA, and mature mRNA - CDS equivalent), cDNAs (mRNA Reverse Transcript - mRNA RT), and expressed sequence tags (ESTs) which are cDNAs between 200-600bp.

Genomic data consist of complete genomes of various organisms. Semantic criteria for selection of sources from where to download genomic data include: completeness, curation (low % error of sequencing, no sequence redundancy), and annotation. ASP will be executed against genomic data stored in the warehouse. Therefore only semantic criteria may be addressed for this selection as the protocol tasks must run against a complete genome.

4.2 Selection of the alignment tool

Numerous alignment tools have been developed to align nucleic acid sequences. They use specific algorithms or derived improved algorithms from the most popular ones such as Smith & Waterman [14] or Needleman & Wunsch [15]. We need to select a resource that performs the alignment of input transcripts against a whole warehoused genome in a fast and accurate way. The selection of an alignment tool suitable to implement the alignment task is driven by the analysis of the characteristics of the sequences that are aligned.

Most of the sequences that are aligned against the selected genome are short sequences (from 200 to 1000 bp) because only a limited number of mRNAs or cDNAs are longer than 1000bp. In addition, we predict that among input transcripts, some or most have already undergone a splicing phase which has excised introns. For this reason, a local alignment tool based on the Smith-Waterman algorithm is preferable to a global alignment using the Needleman & Wunsch algorithm.

Once the semantic characteristics of the service that is expected to perform the protocol task are identified, the selection of the tool among all available services remains a tedious task. To implement the alignment task of ASP available local alignment tools include: BLAST, WU-BLAST, Fasta3, BLAT, SIM4, FDF, MPsearch, ClustalW, T-COFFEE, etc. All these tools use FASTA as a format therefore they are equivalent syntactically.

[2] GeneDis is available at http://life2.tau.ac.il/GeneDis.

Benchmarks may help select a tool suitable to implement tasks of scientific protocols.

5. Benchmarking scientific tools

Benchmarks are designed to access the relative performance of a tool. Database benchmarks consist of a *schema* representing a specific application domain, several *instances* of different sizes, and a list of *tasks* capturing the variety of actions (queries) performed in the application domain [16]. The main characteristic of a benchmark is that **it should not be biased** towards any specific system. A benchmark for scientific tools should follow the same approach. A benchmark for alignment tools should consist of a variety of sequences, each presenting some specific characteristics (these input sequences have the same role as queries in a database benchmark), and instances (sets of sequences which the input queries will be aligned against). The input sequences and instances need to be defined regardless of a specific alignment tool.

Definition 1
A benchmark for sequence alignment tools consists of:

- An *instance* that is a set of sequences against which the sequence alignment tool is run,
- A set of *queries* that are sequences to be aligned against the instance with the alignment tool, and
- A set of *measures* that are recorded while executing the alignment tool on each query against the instance.

Each benchmark query characterizes a property specific to the sequences of the case captured by the benchmark (e.g., phylogenic analysis). Existing benchmarks for sequence alignment address tasks such as sequence similarity searching, phylogenic analysis, multiple sequence alignment, genome-level alignment, sequence assembly, protein structure prediction, and the docking-docking protein. To the best of our knowledge no benchmark has been developed for micro-array, mass spectrophotometry or alternative splicing clustering-specific analysis.
Benchmarks developed for database systems did not address semantics discrepancies because a given query expressed in SQL on a given instance returns the same results regardless of the database management system used to execute it. Only measures of performances were analyzed to evaluate the different database systems tested. In contrast, although scientific tools may be designed to capture a given semantic property, two different tools typically return different results. Scientific benchmarks may be developed to evaluate the semantics of the tool as well as its performance.

Definition 2
A scientific *efficiency benchmark* is characterized by measures of space and time. A scientific *semantic benchmark* is characterized by measures of result quality.

Many candidates for benchmarks of alignment tools have been developed by the scientific community. They include *semantic benchmarks* such as BaliBASE 3.0 [17], OxBench [18], and ZDOCK [19] that compare the quality of the results obtained by the tools, and *efficiency benchmarks* Bioperf [20] and Biobench and Bioparallel [21, 22] that focus on space and time.

Definition 3
Evaluating a scientific alignment tool with a semantic benchmark processes as follows:

1. Let B={I,Q,M} be a semantic benchmark for sequence alignment tools where:
   - I is an instance
   - Q={s_1,…, s_n} a set of sequences
   - M={m_1,…, m_p} a set of measures
2. Let C_i=[c_1,…, c_n]$_i$ be the measures expected when running an alignment tool on s_i against I
3. Let T be an alignment tool
4. For each sequence s_i ($1 \le i \le n$) in Q, run T and record M_i the list of values of measures m_1,…, m_p
5. Compare [M_i] and C_i

For each sequence s_i, C_i results from a set of curated alignment.

5.1 Semantic benchmarks

BaliBASE 3.0 [17] and OxBench [18] address similarity issues for protein sequences as well as nucleotide sequences, whereas ZDOCK [19] only focuses on

protein sequences. They respectively use NCBI Nucleotide and PDB or Swiss-Prot as instances. Their queries are sets of sequences, such as Nucleic Acid sequences or Amino Acid sequences capturing classes or families of sequences with known specific features such as the size of sequences, the percentage of similarities between sequences, or already known domains. The curated measures of alignments provide a comparison vector to evaluate the quality of alignment obtained by the alignment tool.

5.2 Efficiency benchmark

Their measures are based on the analysis of instructions profile, basic block lengths, IPC, L1, L2 data cache, branch prediction accuracy, the share of loads, time-sharing, multiplexing, or analysis of instructions at the memory level in order to identify bottlenecks in computers and give feedback implementation for the tools. Biobench, Bioparallel and BioPerf perform these tests.

5.3 Comparing scientific tools

Once a benchmark is defined, it may be used to compare similar sequence alignment tools. Two local sequence alignment tools may be compared with a benchmark while a local alignment tool may not be compared to a global alignment tool.

Definition 4
Benchmarking scientific alignment tools processes as follows:

1. Let B={I,Q,M} be a benchmark for sequence alignment tools where:
2. I is an instance
3. Q={s_1,…, s_n} a set of sequences
4. M={m_1,…, m_p} a set of measures
5. Let T and T' be two comparable alignment tools
6. For each sequence s_i (1≤i≤n) in Q, run T and record M_i the list of values of measures m_1, …, m_p
7. For each sequence s_i (1≤i≤n) in Q, run T' and record M'_i the list of values of measures m_1, …, m_p
8. Compare [M_i] and [M'_i]

6. Improving protocol implementations

The careful selection of tools to implement each task of the design protocol using syntactic, semantic, and efficiency analysis may not produce an executable protocol. Indeed, the overall organization of tasks may create execution bottlenecks that may affect the overall space and time requirements for its execution and reach limits of tools evoked in the protocol or of the system it runs on. Prediction models exploiting statistical information of the resources selected in the protocol implementation may produce more efficient equivalent (semantically) protocols implementations [1]. Ultimately, the solution to transform a protocol implementation in a manageable protocol implementation is by introducing addition tasks aiming at limiting the dataflow.

A naïve implementation of ASP may consist of the selection of SIM4 to implement Task 1 and a clustering algorithm [23] to implement Task 2, as illustrated in Figure 2.

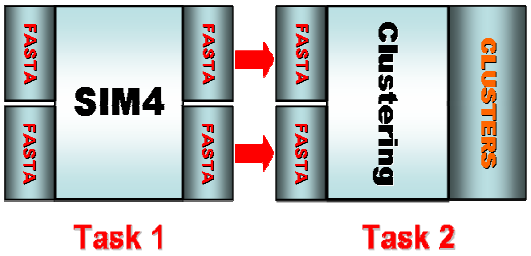

Figure 2 - Naive ASP Implementation

Limitations of execution of scientific protocols may originate from hardware limitations such as CPU, processors, motherboard chip, hard-drives speed, RAM, ROM, virtual memory, etc. Software limitations include calibration of the tool (selection of the parameters) and inclusion of filtering steps that limit the dataflow submitted to the overwhelmed tool.
Because ASP processes very large data inputs (maybe several whole genomes), the fine alignment step achieved by SIM4 is overwhelmed. In order to avoid such a bottleneck, a pre-alignment task that eliminates roughly all sequences not likely to produce a fine alignment with the input transcripts may be used. This pre-alignment consists in the use of Blat, an alignment tool that allows finding fairly accurate alignment positions for transcripts. Filters are then

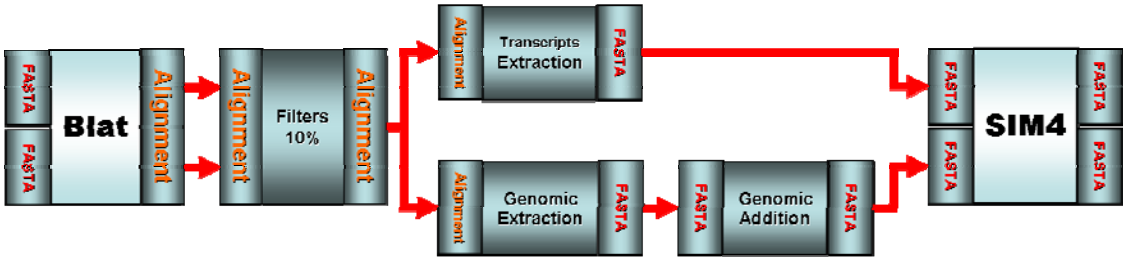

Figure 3 - Efficient implementation of ASP

applied to the alignment results (e.g., by keeping only 10% of the top scoring HSPs and by keeping only specific sections of the genomes that are aligned with transcripts). These filters decrease the number of inputs for the alignment refinement performed by SIM4. In summary, both queries and instances, i.e., transcripts and genome sequences, have been filtered. It results that the manageable implementation of a single scientific task "Alignment" consists of a workflow composed of six tasks illustrated in Figure 3.

7. Related Works

Benchmarks were first developed to compare hardware performances. Those first efficiency benchmarks measure several parameters such as CPU usage, memory access, etc. They consist of program(s) that will be run on the machine that is tested. Because the performance may vary with respect to the context of application, *application benchmark*s that capture a specific usage were developed in contrast to *generic (synthetic) benchmarks*. For example, 056.ear is a benchmark that simulates the human ear [16] using extensive complex Fast Fourier Transform functions. It takes a sound file as input and produces a cochleagram in an output file of 1Mb. 056.ear is a CPU intensive single-precision floating point benchmark and is executed on different hardware systems in order to compare their CPU usage, time, etc.

Benchmarks can also be used to evaluate the relative performance of two software products running on the same computer. They first have been developed as efficiency benchmark comparing for example queries per minute and the price/performance ratio.

Database benchmarks [16] are domain specific and consists of a schema, instance(s), and queries that capture a particular domain (see Definition 1). XML management systems (XMS) recently developed also required evaluation and benchmarks were developed to address the specificities of XMS [24]. Because the evaluation of a query (SQL, OQL, or XQuery) is deterministic, only efficiency was measured.

In contrast, bioinformatics services use different algorithms and services implementing similar scientific tasks may return different results. These discrepancies led to the need of evaluating not only the efficiency of the execution of the services, but also the quality of the results they return.

Existing bioinformatics "benchmarks" typically do not clearly describe the parameters they measure. Some seem to focus on traditional efficiency measures (e.g., time) and provide an instance on which similar applications have been run and for which the execution time was recorded. However the proposed instances are not designed to capture an application domain and rather seem to be a testing instance for an application calibrated to run very efficiently on this dataset, and more efficiently than known similar tools. Such an instance would be biased towards a particular tool, thus would not qualify as a benchmark as defined in Section 5.

Other candidates for bioinformatics benchmarks focus on the quality of results returned by the bioinformatics tools. They consist of an instance, queries, and a curated expected output for the scientific task (Definition 3). Semantic benchmarks are difficult to develop because they require significant data curation which limits their size. Because semantic benchmarks are yet not large enough, they cannot be used to also measure efficiency adequately.

8. Conclusion

In this paper we analyze a scientific protocol developed to support a service devoted to

alternative splicing analysis. We explain the process of selection of resources to implement each scientific task and lay the ground for scientific tools analysis and benchmarking to produce sound and efficient scientific protocols. We define the concept of *benchmark* for scientific tools, and acknowledge the need for *semantic benchmarks* that evaluate data quality and *efficiency benchmarks* that focus on fast data processing. Future work will focus on the development of a benchmark for alternative splicing analysis and a rigorous analysis of performance of existing alignment tools.

Acknowledgements

We are grateful to Dr. Louiqa Raschid for her help and comments and to Ben Snyder for providing technical information regarding his implementation of BIPASS. We thank Dr. Bahar Taneri and Dr. Terry Gaasterland for their input developing BIPASS. This research was partially supported by the National Science Foundation grants[3] IIS0222847, IIS0430915, IIS 0223042, and IIS 0222847.

References

[1] Z. Lacroix, L. Raschid, and B.A. Eckman, "Techniques for Optimization of Queries on Integrated Biological Resources," *Journal of Bioinformatics and Computational Biology*, 2(2):375-411, 2004.

[2] Z. Lacroix, C. Legendre, R. Louiqa, and B. Snyder, "BIPASS: BioInformatics Pipeline Alternative Splicing Services," *Nucleic Acids Res., Web services special issue*, July 2007, DOI:10.1093.

[3] N. Kwasnikowska, Z. Lacroix, and Y. Chen "Modeling and Storing Scientific Protocols," in Proc. International Workshop on Knowledge Systems in Bioinformatics, Lecture Notes in Computer Science (LNCS), Vol. 4277, pp 730-739, 2006.

[4] D. Booth and C. K. Liu, "Web Services Description Language (WSDL) Version 2.0 Part 0: Primer," World Wide Web Consortium, www.w3.org/TR/2007/WD-wsdl20-primer-20070326/, 2006.

[5] R. Chinnici, J. J. Moreau, A. Ryman, and S. Weerawarana, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language," World Wide Web Consortium, www.w3.org/TR/2006/CR-wsdl20-20060327/, 2006.

[6] M. D. Wilkinson and M. Links, "BioMOBY: an open source biological web services proposal," *Briefings in Bioinformatics*, 3(4):331-341, 2002.

[7] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: A tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, 20(17): 3045-3054, 2004.

[8] S. Carrere and J. Gouzy, "REMORA: a pilot in the ocean of BioMoby web-services," *Bioinformatics*, 22(7): 900-901, 2006.

[9] Z. Lacroix and H. Ménager, "SemanticBio: building conceptual scientific workflows over web services," in Proc. 2nd International Workshop on Data Integration in the Life Sciences, in Lecture Notes in Bioinformatics (LNBI), Vol. 3615, pp 296-299, 2005.

[10] B. Néron, P. Tufféry, C. Letondal, "Mobyle: a Web portal framework for bioinformatics analyses," poster presented at NETTAB, 2005.

[11] E. Kawas, M. Senger, and M. D. Wilkinson, "BioMoby extensions to the Taverna workflow management and enactment software," *BMC Bioinformatics*, vol. 7:523+, 2006.

[3] Any opinion, finding, and conclusion or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[12] Z. Lacroix, "Evaluating similar implementations of a scientific protocol on NCBI resources," Technical Report, Arizona State University, 2005.

[13] B. A. Eckman, T. Gaasterland, Z. Lacroix, L. Raschid, B. Snyder, and M.-E. Vidal, "Implementing a Bioinformatics Pipeline (BIP) on a Mediator Platform: Comparing Cost and Quality of Alternate Choices," in Proc. *Workshop on Workflow and Data Flow for Scientific Applications*. Atlanta, Georgia: IEEE Press, pp. 67+, 2006.

[14] T. F. Smith, M. S. Waterman, and W. M. Fitch, "Comparative biosequence metrics," J*ournal of Molecular Evolution*, 18(1): 38-46, 1981.

[15] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, 48(3):443-53, 1970.

[16] J. Gray, *The Benchmark Handbook for Database and Transaction Systems (2nd Edition)*, Digital Equipement Corporation ed: Morgan Kaufmann Publishers, 1993.

[17] J. D. Thompson, P. Koehl, R. Ripp, and O. Poch, "BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark," *Proteins*, 61(1):127-36, 2005.

[18] G. P. Raghava, S. M. Searle, P. C. Audley, J. D. Barber, and G. J. Barton, "OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy," *BMC Bioinformatics*, vol. 4, pp. 47+, 2003.

[19] J. Mintseris, K. Wiehe, B. Pierce, R. Anderson, R. Chen, J. Janin, and Z. Weng, "Protein-Protein Docking Benchmark 2.0: an update," *Proteins*, 60(2):214-6, 2005.

[20] D. A. Bader, Y. Li, T. Li, and V. Sachdeva, "BioPerf: a benchmark suite to evaluate high-performance computer architecture on bioinformatics applications," in Proc. IEEE International Symposium on Workload Characterization, IEEE Press, pp 163-173, 2005.

[21] K. Albayraktaroglu, A. Jaleel, X. Wu, M. Franklin, B. Jacob, C. W. tseng, and D. Yeung, "BioBench: A benchmark suite for bioinformatics applications," in Proc. IEEE International Symposium on Performance Analysis of Systems and Software, IEEE Press, pp 2-9, 2005.

[22] A. Jaleel, M. Mattina, and B. Jacob, "Last-level cache (LLC) performance of data-mining workloads on a CMP--A case study of parallel bioinformatics workloads," in Proc. International Symposium on High Performance Computer Architecture, IEEE Press, pp 88-98, 2006.

[23] B. Taneri, B. Snyder, A. Novoradovsky, and T. Gaasterland, "Alternative splicing of mouse transcription factors affects their DNA-binding domain architecture and is tissue specific," *Genome Biol*ogy, 5(10):R75, 2004.

[24] S. Bressan, M. L. Lee, Y. Li, Z. Lacroix, and U. Nambiar, "XML Management System Benchmark," in *XML Data Management: Native XML and XML-Enabled Database Systems*, A. B. Chaudri, A. Rashid, and R. Zicari, Eds.: Addison-Wesley, 2003, pp. 477-498.