ELSEVIER

Future Generation Computer Systems 18 (2002) 863–870

FGCS
FUTURE GENERATION COMPUTER SYSTEMS

www.elsevier.com/locate/future

Parallel creation of non-redundant gene indices from partial mRNA transcripts

Nishank Trivedi, Jared Bischof, Steve Davis, Kevin Pedretti, Todd E. Scheetz, Terry A. Braun, Chad A. Roberts, Natalie L. Robinson, Val C. Sheffield, M. Bento Soares, Thomas L. Casavant*

Parallel Processing Laboratory, and The Coordinated Laboratory for Computational Genomics, Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242, USA

Abstract

This paper describes the `UIcluster` software tool, which partitions expressed sequence tag (EST) sequences and other genetic sequences into "clusters" based on sequence similarity. Ideally, each cluster will contain sequences that all represent the same gene. `UIcluster` has been developed over the course of 4 years to solve this problem efficiently and accurately for large data sets consisting of tens or hundreds of thousands of EST sequences. The latest version of the application has been parallelized using the MPI standard. Both the computation and memory requirements of the program can be distributed among multiple (possibly distributed) UNIX processes. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Parallel cluster application; Expressed sequence tag; Genome project

1. Introduction

Partitioning of partial mRNA transcripts into non-redundant sets, or gene indices, is commonly referred to as clustering. In the high throughput gene sequencing activities of our laboratories, we generate large numbers of short mRNA transcript sequences—expressed sequence tags (ESTs)—and partition them into sets based on similarity. The importance of this problem bears on several aspects, but the principal of these are creating non-redundant indices of genes and assessing the novelty of sequencing. If done in a naïve fashion, such as an $N \times N$ comparison, this problem would be intractable for the data set sizes we produce (50K–300K ESTs). Although there are several existing software system [1,5–7] available that perform sequence clustering accurately, our program is unique in its ability to efficiently and accurately cluster EST sequences. Over the past 4 years, we have developed techniques to speedup the computation by using increasingly sophisticated heuristics along with parallel processing techniques. The usefulness of our program, `UIcluster`, has been demonstrated in conjunction with our gene discovery projects in identifying more than 100,000 novel clusters across three species (human, mouse, and rat).

2. Expressed sequence tags

From a biological perspective, ESTs are partial transcripts of genes. Specifically, they are sequenced from cDNA (complementary DNA) clones, typically synthesized from polyA-selected RNA. To prepare for EST sequencing, mRNA molecules are extracted

* Corresponding author.
E-mail address: tomc@eng.uiowa.edu (T.L. Casavant).

0167-739X/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-739X(02)00059-6

from cells and converted into cDNA through reverse transcription. The cDNAs are then cloned into a vector and electroporated into bacteria for growth, amplification, and storage. A collection of such cDNAs is referred to as a library. Each cDNA library potentially contains many unique and previously undiscovered genes. However, significant redundancy within a library (multiple copies of the same mRNA) and between libraries is normal.

High throughput EST sequencing for gene discovery involves sequencing the 3′ end of randomly chosen cDNA clones from a cDNA library. The use of a poly-T primer during reverse transcription allows for the preferential creation of cDNAs with a poly-A tail at their 3′ ends. Thus, sequencing can start from a known position (within poly-A tail).

For the purpose of this paper, and from the computational perspective, an EST is a character string made up of letters from the alphabet A, C, T, G, X, N, where A, C, T, and G represent the four nucleotide bases of DNA and X and N the bases that have been masked (i.e., due to low-complexity or similarity to known repetitive sequences) or that are of indeterminate identity. ESTs are typically several hundred letters, or bases, long. Comparing pairs of ESTs and looking for similarity is the basic element of sequence clustering. This comparison is complex because the underlying sequencing technology is error prone—bases can be inserted, deleted, or misread. Studies of our EST sequences have indicated that the error rate for EST sequencing is approximately 2–3% for misread errors, and 0.1–1% for insertion/deletion errors.

3. Uses of clustering

Clustering is used to assess the gene discovery rate of sequencing done from cDNA libraries. For novel assessment of individual libraries, the entire set of ESTs obtained from that library is used as an input to clustering. Clustering partitions the set into subsets, or clusters, based on sequence similarity. Each EST is a member of at most one cluster. Novelty is computed as the number of clusters identified divided by the number of sequences in the clustering.

This computation is used to calculate both instantaneous and overall novelty rates for individual cDNA libraries and for EST projects as a whole. Incremental novelty calculations are performed frequently to monitor the sequencing efforts and to determine when cDNA library subtractions should occur [2]. This procedure can dramatically increase novelty rates. However, the subtraction process is time consuming and cannot be performed on a continual basis.

Fig. 1 shows an example of the effectiveness of these procedures for a progression of four successive cDNA libraries, named C0, C1, C2p, and C3. Each sharp increase in novelty rates corresponds to the creation of the next subtracted library.

Another significant use of clustering is the generation of non-redundant gene indices, or UniGene sets [7]. As mentioned previously, ideally each cluster will uniquely represent a gene. Thus, the goal in constructing a UniGene set is to bring together all of the ESTs sequenced from a given gene into a single cluster. This information is useful for reducing redundant processing and for the annotation of EST sequences.

4. Program evolution

`UIcluster` has evolved as our laboratory's processing requirements have increased. Three generations of the clustering program have been developed to date. The first revision was developed to work well for moderately sized data sets of ESTs. As our data sets grew, this version required more than 24 h to cluster the entire set of ESTs. The primary goal of the second version of the program was improved performance for large data sets. A third, parallelized version provided higher performance and several additional features has recently been released. All revisions of `UIcluster` may be freely obtained from our project web site (http://genome.uiowa.edu).

The basic clustering program flow proceeds as follows: (1) read one sequence from the input file, (2) compare the sequence against every existing cluster, (3) based on sequence similarity, either add it to an existing cluster or make it the first member of a new cluster. This process is repeated until every sequence in the input file is examined. In step 3, the EST is only added to an existing cluster if the specified similarity criteria is met. The similarity criteria is run-time configurable and is of the form *N* out of *M* bases. For example, 38 out of 40 bases would mean two sequences are judged to be similar if the sequences being evaluated

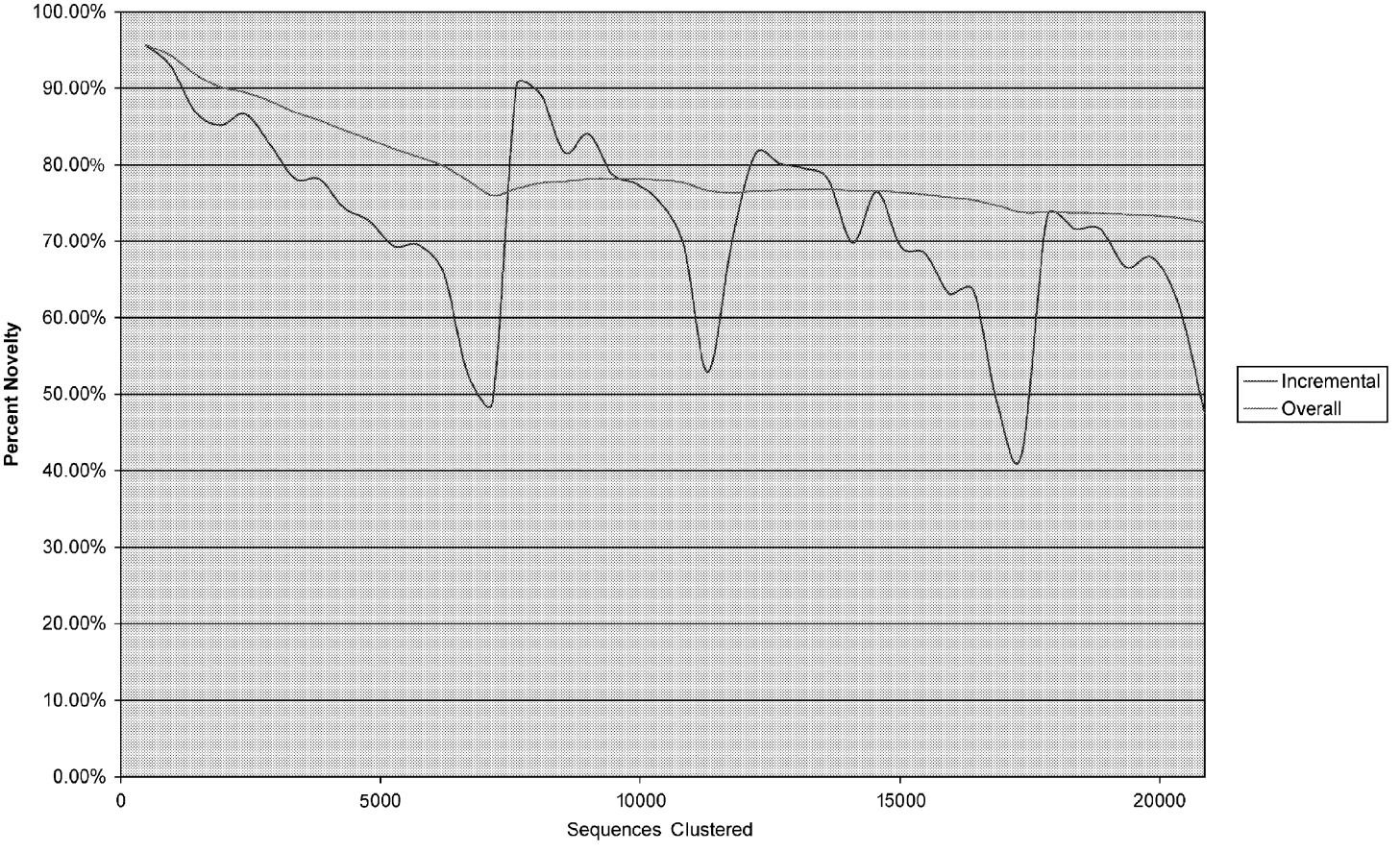

Fig. 1. Incremental library novelty.

contain regions of 40 bases in which at least 38 of the bases match. This comparison allows for insertion, deletion, and mismatch errors. The speed of the program is directly effected by these parameters. Higher error tolerance ($M - N$) increases program execution time significantly as does larger window sizes (M).

4.1. Revision 1.0

Revision 1.0 was useful for relatively small data sets (<30,000 ESTs). The program was structured so that clusters were stored in a 2D linked list. Each EST read from the input file was compared against a single representative element from each cluster. The longest EST from each cluster was used as a representative element for that cluster.

Evaluating the N of the M similarity criteria for two sequences is computationally intensive. Therefore, as a performance optimization, a hashing technique was used to avoid comparisons that cannot satisfy the N of the M criteria. A *hash* is simply an integer that uniquely represents a short string of characters. The general equation used to generate a hash is given as follows:

$$H = \sum_{i=0}^{\zeta-1} (K^i \phi_i). \tag{1}$$

In this equation, H is the generated hash value, ζ the string length, K the alphabet size, and ϕ_i the integer value assigned to the letter at position i in the string being hashed. The string length ζ that can be used to generate hashes is limited by the word size of the computer. For the DNA alphabet, each base requires 2 bits to represent it ($\lceil \log_2 K \rceil$, where $K = 4$ for DNA). Thus, the maximum value of ζ using a single word on a 32 bit system is 16.

When a sequence is hashed, Eq. (1) is used on every ζ length sub-string. Fig. 2 shows the first six hashes generated for a sample sequence with $\zeta = 8$.

When an EST is clustered, the N of the M similarity criteria is only evaluated for cluster representatives that contain one or more hashes in common with the EST being clustered. The length of the hash probe used is an important parameter that can significantly affect performance. Longer hash lengths will result in better performance for a given similarity criteria. It must also be chosen carefully so that potential

```
Sequence:  GCCACTTGGCGTTTTG
Hashes:
           Hash  1:  GCCACTTG  =  48406
           Hash  2:  CCACTTGG  =  44869
           Hash  3:  CACTTGGC  =  27601
           Hash  4:  ACTTGGCG  =  39668
           Hash  5:  CTTGGCGT  =  59069
           ...etc.
```

Fig. 2. Example of hashing a sequence.

similarities are not missed. The formula for calculating the maximum hash size is shown in (2). The rational underlying this equation is that for any chosen similarity criteria N of the M, there is at least one contiguous, error-free region of ζ bases. Thus, the comparison of two sequences can be accelerated by first searching for short exact matches of length ζ bases between the pair (i.e. searching for identical hashes). If such a match is found, a more exhaustive search that permits errors can be performed. If no length ζ hashes are identified, then the two sequences cannot possibly contain a window of M bases with N bases in common:

$$\zeta = \left\lfloor \frac{M}{M - N + 1} \right\rfloor. \tag{2}$$

The calculation to generate the hashes for a sequence is only performed once since the hash lists are stored in memory. However, the hashes are accessed many times during the programs execution. This amortizes the computational overhead of generating the hashes.

4.2. Revision 2.0

The main improvement in this revision was the implementation of the global hash table (GHT). As our EST data sets grew larger, the sequential nature of the traversal of the cluster representative linked list for every input sequence became a bottleneck. The GHT optimizes the program at a higher level than individual sequence comparisons by filtering the entire search space of cluster representatives into a subset of high-potential candidate targets.

When a new sequence is clustered, a list of hashes is generated for each ζ base window of its sequence. Each hash in the list is then used as an index into the

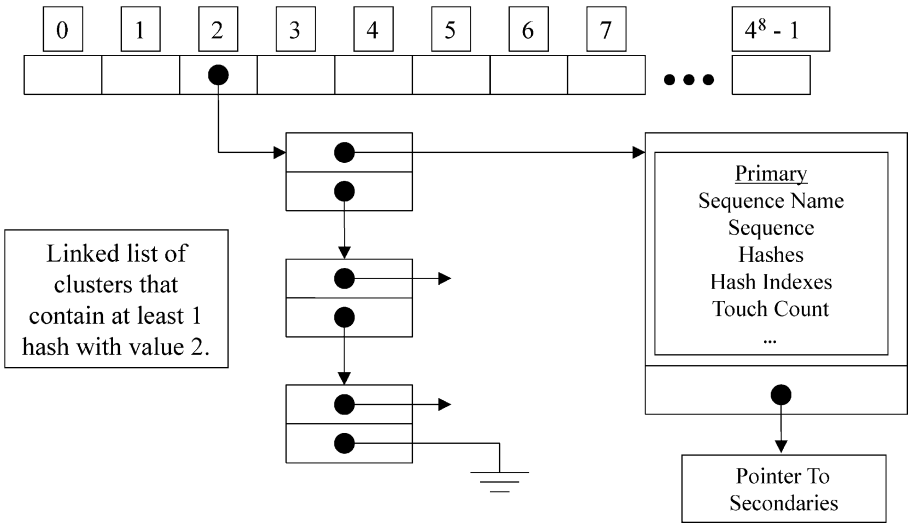

Fig. 3. GHT.

GHT. Fig. 3 shows a GHT with 4^8 elements, corresponding to $\zeta = 8$. Each element in the table points to a linked list of clusters that contain at least one occurrence of the hash equal to its index. In Fig. 3, there are three clusters that contain the hash 2. If the sequence being clustered also has a hash of two, the touch count field of each cluster linked from the second element in the GHT is incremented. If the touch count field of a cluster exceeds a run-time configurable threshold, a detailed sequence comparison is performed between the input sequence and the candidate cluster. This procedure is based on the premise that two similar sequences will likely have many hashes in common.

Care must be taken to adjust the touch count threshold appropriately. For a given similarity criteria (e.g. 38 out of 40 bases) and hash length ζ, if the threshold is too low the speedup due to the GHT will be small. Conversely, if the threshold is too high, some sequence similarities will be missed.

This revision demonstrated 28× speedup on an input data set of 80,766 rat EST sequences while calculating virtually identical results. The major trade-off of the GHT optimization is memory utilization. However, on a system with 2 GB of RAM we have been able to cluster data sets as large as 1 million ESTs. While theoretically the first revision could handle data sets this long, the computation time required would make it impractical.

4.3. Revision 3.0

The most recent version of the clustering program has been parallelized to divide the computational and memory requirements across several computers (compute nodes). The parallelization added performance and enabled computation of larger problem sizes. The MPI (message passing interface) [4] communication standard has been used for inter-process communication.

In this mode of execution, each cluster is stored on exactly one compute node. A given sequence is read in from the input file and processed in parallel on each compute node. This results in a parallel search of the cluster space. After each node has finished its search, it collectively communicates the local best match to all compute nodes. Only the node with the best match stores the sequence in its memory space. If no match is found on any of the compute nodes, the input sequence becomes a new cluster and is assigned to one of the compute nodes in a round-robin assignment.

This implementation uses a collective communication at the end of every sequence clustered, therefore, the amount of computation required for each sequence

is important. As the grain size increases, better performance should be observed because relatively less communication is being performed.

Performance scales poorly for default parameters and one sequence per job, actually decreasing when using two compute nodes. This is due to the computation being unevenly distributed and the additional communication overhead. With more compute nodes, performance increases somewhat but is never greater than double that of the serial case. The larger grain size when incoming sequences are searched against all clusters (instead of stopping after the first identified match) results in significantly improved speedup.

5. Current research

Our current research on `UIcluster` focuses primarily on transcript-centered methods—unifying the information present from every constituent of a cluster to provide a more comprehensive representation of the entire gene. The major change this encompasses is the definition of a "virtual primary" that represents all of the unique subsequences. The impact of this is to enable more comprehensive storage of transcript information, including alternative splicing information. This in turn allows unification of ESTs derived from alternatively spliced mRNAs that might otherwise not have overlapped.

6. Conclusion

The evolution of an EST clustering program has been discussed. Background information on the problem has been presented along with details of two sequential implementations and a parallel implementation. Planned extensions to `UIcluster` include utilizing the recently released human genome sequence [3,8], and others applicable to the organism under study (e.g., mouse, rat) to improve the accuracy of clustering, and to aid in identification of alternative splice forms and intron/exon boundaries. Other extensions planned include improved performance for long sequences (e.g., full length cDNA sequences), automated and semi-automated cluster merging and merge candidate identification, and tools for manual curation of clustering results by expert human operators.

References

[1] M.D. Adams, A.R. Kerlavage, R.D. Fleishmann, R.A. Fuldner, C.J. Bult, N.H. Lee, E.F. Kirkness, K.G. Weinstock, J.D. Gocayne, O. White, et al., Initial assessment of human gene diversity and expression patterns based upon 83 million nucleotides of cDNA sequence, Nature 377 (1995) 3–17.
[2] M.F. Bonaldo, G. Lennon, M.B. Soares, Normalization and subtraction: two approaches to facilitate gene discovery, Genome Res. 6 (1996) 791–806.
[3] Initial sequencing and analysis of the human genome, International Human Genome Sequencing Consortium, Nature 409 (2001) 860–921.
[4] MPI: a message-passing interface standard, Technical Report CS-94-230, Message Passing Interface Form, University of Tennessee, 1994.
[5] R.T. Miller, A.G. Christoffels, C. Gopalakrishnan, J.A. Burke, A.A. Ptitsyn, T.R. Broveak, W.A. Hide, A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus knowledge base, Genome Res. 9 (1999) 1143–1155.
[6] J.D. Parsons, S. Brenner, M.J. Bishop, Clustering cDNA sequences, Comput. Appl. Biosci. 8 (1992) 461–466.
[7] G.D. Schuler, Pieces of the puzzle: expressed sequence tags and the catalog of human genes, J. Mol. Med. 75 (1997) 694–698.
[8] J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, et al., The sequence of the human genome, Science 291 (2001) 1304–1351.

Nishank Trivedi received his Bachelor of Technology degree in electrical engineering from University of Lucknow, India. He is currently pursuing his Master of Science degree in electrical engineering from The University of Iowa, and is a research assistant in the Coordinated Laboratory for Computational Genomics. His research interests include bioinformatics, parallel and distributed computing, computer architecture and hardware/software codesign.

Jared Marshall Jerome Bischof is working on his BS degree (2002) in Electrical and Computer Engineering at the University of Iowa. He is currently a research assistant in the Coordinated Laboratory for Computational Genomics at the University of Iowa. Some of his research interests include sequence clustering, computer systems, and detecting alternative splicing among genomically aligned ESTs.

Steve G. Davis received the BS degree in Computer Science in December of 2001 from the University of Iowa where he was also on the Division One Iowa Mens Gymnastics Team. He is currently pursuing his MS degree in Electrical and Computer Engineering while working as a Computational Scientist in the Coordinated Laboratory for Computational Genomics at the University of Iowa. His research interests include bioinformatics, data-mining, and artificial intelligence.

Kevin Pedretti received his Bachelor of Science in electrical engineering, and his Master of Science in electrical and computer engineering from The University of Iowa in 2001. He has been an active member in several gene discovery projects lead by The University of Iowa, and is currently a system software developer at Sandia National Laboratories in Albuquerque, NM. Current research interests include scalable operating systems, scalable computer architectures, and bioinformatics.

Todd E. Scheetz received his Bachelor of Science degree in electrical engineering, Master of Science degree in electrical and computer engineering, and PhD in genetics from The University of Iowa in 2001. He is currently a Senior Computational Scientist in the Coordinated Laboratory for Computational Genomics at The University of Iowa. Current research involvement includes a rat EST mapping project, gene discovery projects in human, rat, mouse, and pig, and several human disease-related projects. He has co-authored several papers in the areas of high performance computer architecture and parallel systems, and bioinformatics. Research interests include bioinformatics, disease gene identification, mapping, data-mining, parallel and distributed processing, and operating systems.

Terry Allen Braun received a PhD in Genetics (2001), and MS (1995) and BS (1993) degrees in Electrical and Computer Enigeneering from the University of Iowa. Currently, he is a Senior Computational Scientist in the Coordinated Laboratory for Computational Genomics at Iowa. He has co-authored several papers in the area of high performance computer architecuture before research interests focussed on genetics, bioinformatics and computational biology. Current research interests include disease gene identification and prioritization using automated knowledge discovery and sequence analysis.

Natalie Robinson received her Bachelor of Science in Biology and her Master of Science in Electrical and Computer Engineering from the University of Iowa in 2001. She is currently a Computational Scientist with the Coordinated Laboratory for Computational Genomics at the University of Iowa College of Engineering. Her current research interests are in EST sequence processing and quality assessment, Microarray probe set design and development of computational algorithms to support gene discovery.

Val C. Sheffield received his BS degree in Zoology in 1974 and his MS degree in Developmental Biology in 1977 from Brigham Young University in Provo Utah. In 1983 he received in PhD in Developmental Biology and in 1985 his MD with Honors from The University of Chicago, Chicago, IL. From 1985–1987 he was a Pediatric Resident at The University of California, San Francisco, and from 1987–1990 a Fellow in Medical Genetics at The University of California, San Francisco. He joined The University of Iowa in 1990, and is currently a Porfessor of Pediatrics and the Director of the Division of Medical Genetics and the Interdepartmental Research Program in Human Molecular Genetics. In 1998 he became an Associate Investigator with the Howard Hughes Medical Institute. Dr. Sheffield has over 150 publications in peer-reviewed journals. Dr. Sheffield's current projects in the lab include the study of inherited blindness, deafness, obesity, hypertension, and autism. In addition, his laboratory has been active in genomic research.

Dr. Marcelo Bento Soares received a Bachelor's degree in Genetics in 1979, and has subsequently earned an MS in 1982, specializing in Genetics from Federal University of Rio de Janeiro, and a PhD with distinction from Columbia University in 1986, specializing in both Molecular Biology and Genetics. Dr. Soares is currently the Principal Investigator for a Gene Discovery laboratory and also a Professor of Pediatrics, Biochemistry, and of Physiology and Biophysics at the University of Iowa. He has over 50 publications in peer-reviewed journals. His current projects include normalization, subtraction and construction of cDNA libraries, chondrasarcoma, cystic fibrosis, SAGE and creating full length cDNA libraries. He has over 50 publications.

Thomas L. Casavant is currently Professor of Electrical and Computer Engineering at the University of Iowa. He received the BS degree in Computer Science in 1982, the MS degree in Electrical and Computer Engineering in 1983, and the PhD degree in Electrical and Computer Engineering from the University of Iowa in 1986. In 1986, Dr. Casavant joined the faculty of the School of Electrical Engineering at Purdue University, West Lafayette, Indiana specializing in the design and analysis of parallel/distributed computing systems, environments, algorithms, and programs. From 1987–1989, he was Director of the PASM Parallel Processing Project, and the Purdue EE School's Parallel Processing Laboratory. He has developed graduate courses in advanced computer architecture, distributed computing, parallel processing, and computational biology. In 1989, he joined the faculty of the Iowa ECE Department and was promoted to Professor in 1999. There, he is director of both the Coordinated Laboratory for Computational Genomics, as well as the Parallel Processing Laboratory. Since 1996, he has led Computational Molecular Biology efforts in Gene Discovery, Mapping, and Disease Gene Identification/Isolation. From 1993–1994, he was a guest professor with the Department of Infor-

matik at the ETH (Eidgenossisch Technische Hochschule—Swiss Federal Institute of Technology) in Zurich, Switzerland. In 2000, he was a guest researcher in the Biochimie et Biophysique des Systemes Integres Laboratory and the CEA/CNRS (Centre National de la Recherche Scientifique/Commissariat a l'Energie Atomique) in Grenoble, France. Dr. Casavant has authored or co-authored over 100 technical articles in Computer Science/Engineering and Computational Biology/Bioinformatics, edited two books on Parallel and Distributed Computing, served as editor for IEEE Transactions on Parallel and Distributed Processing and the Journal of Parallel and Distributed Computing, and has presented numerous tutorials worldwide.