A Transcript Perspective on Evolution

Yann Christinat and Bernard M.E. Moret

Abstract—Alternative splicing is now recognized as a major mechanism for transcriptome and proteome diversity in higher eukaryotes, yet its evolution is poorly understood. Most studies focus on the evolution of exons and introns at the gene level, while only few consider the evolution of transcripts. In this paper, we present a framework for transcript phylogenies where ancestral transcripts evolve along the gene tree by gains, losses, and mutation. We demonstrate the usefulness of our method on a set of 805 genes and two different topics. First, we improve a method for transcriptome reconstruction from ESTs (ASPic), then we study the evolution of function in transcripts. The use of transcript phylogenies allows us to double the precision of ASPic, whereas results on the functional study reveal that conserved transcripts are more likely to share protein domains than functional sites. These studies validate our framework for the study of evolution in large collections of organisms from the perspective of transcripts; for this purpose, we developed and provide a new tool, TrEvoR.

Index Terms—Alternative splicing, transcript, evolution, phylogeny, protein domain, transcriptome reconstruction

1 Introduction

Gene duplication and loss are the main driving forces for transcriptome and proteome diversity. However, alternative splicing—a greatly underestimated mechanism 20 years ago—has now been shown to play a major role for diversity in higher eukaryotes [1], [2]. In many genomes, most genes are, thus, split into introns and exons. The standard splicing scheme keeps all exons and removes all introns, but alternative splicing permits removal of alternative exons. Some mRNAs are further translated into proteins—named isoforms—and alternative proteins can, therefore, vary in large regions or may even not overlap.

Alternative splicing is limited in plants and fungi but quite common in vertebrates. Some researchers conjecture that 90 percent of human multiexon genes are alternatively spliced [3]. Yet the study of evolution from a transcript perspective has not seen much work, while the evolution of the mechanism itself is poorly understood. Several articles focus on the evolution of the gene structure—exons and introns—but none address the problem of transcript evolution [3]. The few studies on this matter are limited to mouse and human and agree on the fact that alternative splicing is a fast evolving mechanism [4], [5]. For instance, Nurtdinov showed that only three quarters of the human isoforms have an ortholog in mouse [6].

The number of alternative isoforms is specific to species but also unevenly documented. For instance, the Ensembl database [7] reports numerous transcripts for human, mouse, rat, several apes, and a few fishes but almost no alternative splicing for dogs or cats. Some gene families also display a very different rates of alternative splicing across their species; a member can be extensively spliced in a species and have only one transcript in another species. The available data comes mainly from experiments and it is expected to be incomplete. Moreover, current automated pipelines for transcriptome reconstruction are not trusted, so that large-scale multispecies analysis is not doable at present. Confronted with the incompleteness of the data and the presence of extensively spliced genes, some researchers conjecture that all alternative transcripts are possible and that the observed set reflects a regulated distribution of all transcripts. Nonetheless, some transcripts are conserved among homologous genes. The question remains to be answered whether the function—be it gene ontologies, tissue or subcellular localization, or developmental stages—is also conserved or if they just represent noise in the splicing machinery.

In this paper, we extend our previous work [8] and present a transcript evolution framework where ancestral transcripts evolve along the gene tree through transcript gains or losses, and exon gains or losses. This entire process is represented by a forest of transcript phylogenies that links the observed transcripts of a gene family. This framework has applications in many transcript-related fields. Among these, we selected two to demonstrate the benefits gained through our method. In Section 3.1, we address the problem of transcriptome reconstruction, as it represents a core issue for transcript analysis. We refined the output of ASPic (the Alternative Splicing Prediction DataBase [9]): as measured using the RefSeq database [10], our method doubles the precision of ASPic. In Section 3.2, we study the distribution of protein domains in transcript phylogenies, using many Ensembl tracks. Our results indicate that transcripts are more likely to conserve their domains than their functional sites through evolution.

These two studies establish the usefulness of our transcript phylogeny framework for large-scale biological analyses. The tool we developed for these analyses

- *Y. Christinat is with the Laboratory for Computational Biology and Bioinformatics, EPFL, Lausanne, Switzerland, and the Institute of Molecular Health Sciences, HPL H16.2, Schafmattstrasse 22, CH-8093 Zurich, Switzerland. E-mail: yann.christinat@biol.ethz.ch.*
- *B.M.E. Moret is with the EPFL IC IIF LCBB, INJ 230 (Bâtiment INJ), Station 14, CH-1015 Lausanne, Switzerland. E-mail: bernard.moret@epfl.ch.*

Manuscript received 15 Aug. 2012; revised 23 Oct. 2012; accepted 1 Nov. 2012; published online 28 Nov. 2012.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-2012-08-0200.
Digital Object Identifier no. 10.1109/TCBB.2012.145.

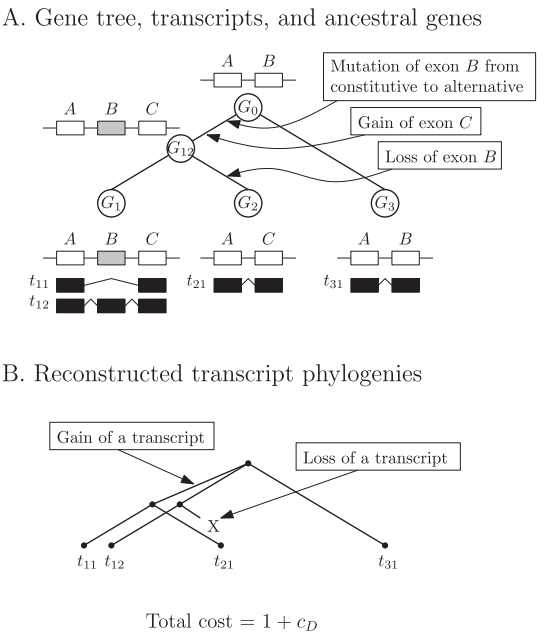


Fig. 1. Illustration of the two-level model. The first level is represented in A where the gene evolution happens. In B, one can see the transcript phylogeny. Transcripts t_{12} and t_{31} differ by exon C, which was gained during the evolution from G_0, to G_{12}. This event belong to the first level and, thus, has zero cost in the transcript phylogeny. There is one transcript loss ($\text{cost} = c_D$) between G_{12} and G_2, and there is a new splicing pattern in G_{12} that cannot be explained by the evolution of the gene structure. The total cost is, thus, $1 + c_D$.

Transcript Evolution Reconstruction software (TrEvoR), is publicly available and can be downloaded at http://lcbb.epfl.ch/trevor.

2 A Model of Transcript Evolution

Our model of transcript evolution is a refinement of the extended model we presented in [8]. Transcripts evolve along the gene tree through three simple events: mutation (gain or loss of exons), fork (creation of new transcripts), and death (loss of transcripts). This process is represented in a set of *transcript trees*, one for each ancestral transcript present in the ancestral gene at the root of the gene tree. Mutations happen along the branches and affect the content of transcripts whereas forks and deaths affect the structure of the transcript trees.

A transcript tree is defined as a phylogenetic tree with transcripts as nodes. The root of the tree is an ancestral transcript (from the ancestral gene at the root of the gene tree) and leaves correspond to current transcripts of the gene family. Transcript deaths are represented by interrupted edges with a cross at their ends.

To maintain consistency with the gene tree, the following three conditions describe the relationship between the gene tree and a transcript tree and should hold for any transcript tree:

1. Transcripts evolve along the gene tree through the three events described earlier: mutation, fork, and death. Therefore, for each edge (t_a, t_b) in a transcript tree there exists an edge $(Gene(t_a), Gene(t_b))$ in the gene tree. ($Gene(t)$ represents the gene associated to transcript t.)

TABLE 1
Evolutionary Cost for a Particular Exon 0: Absent, 1: Present, 1_A: Alternative Exon, and 1_C: Constitutive Exon

Gene's exon evolution	Transcript's exon evolution	Cost
$0 \to 0$	$0 \to 0$	0
$0 \to 1_A$	$0 \to 0$	0
	$0 \to 1$	1
$0 \to 1_C$	$0 \to 1$	0
$1_A \to 0$	$0 \to 0$	0
	$1 \to 0$	0
$1_A \to 1_A$	$0 \to 0$	0
	$0 \to 1$	1
	$1 \to 0$	1
	$1 \to 1$	0
$1_A \to 1_C$	$0 \to 1$	0
	$1 \to 1$	0
$1_C \to 0$	$1 \to 0$	0
$1_C \to 1_A$	$1 \to 0$	1
	$1 \to 1$	0
$1_C \to 1_C$	$1 \to 1$	0

Note that this table only applies to mutations between two transcripts.

2. Recall that we have one transcript tree per ancestral transcript. Consequently every transcript tree has exactly one node t_0 such that $Gene(t_0)$ is the root of the gene tree. This condition ensures that no transcript can be created *de novo*; all transcripts must evolve from an ancestor.
3. A transcript death occurs whenever a transcript tree contains a node in a gene but none in a descendant of this gene. Formally, we have that given a gene tree edge (G_a, G_b) such that G_a is the ancestor of G_b, a transcript tree T will contain a transcript death at t_a ($Gene(t_a) = G_a$) if $\nexists (t_a, t) \in T$ s.t. $Gene(t) = G_b$.

Transcripts and genes are represented as sequences of exons. The evolutionary relationship between exons is inferred from a multiple alignment and was inspired by Takeda et al. [11]: exons that overlaps reciprocally with more than 70 percent are considered orthologous. Alternative 3′- and 5′- sites are modeled as multiple-state exons.

In the absence of prior work, we opted for a maximum parsimony framework. Every event is assigned a unit cost, except for transcript death, which is the sole event of the model to be parametrized (c_D). Other frameworks such as maximum likelihood or Bayesian networks can be used, but they tend to yield higher computational costs.

Our model aims at reflecting the cost of transcript evolution alone; thus, the cost of gene evolution is discarded. This implies a two-level model, where the evolution of the gene structure serves as a basis for the evolution of the transcriptome. For instance, the loss of an exon at the gene level implies that all transcripts lose this exon. In a classical maximum parsimony framework, these events would add their cost to the score. In our model, however, they do not since they are the unavoidable consequence of a gene event. This concept is illustrated through an example in Fig. 1. There are actually only a few events that have a cost in our model. Table 1 summarizes the cost for different exon events in a transcript with respect to the gene evolution. Note that this table only display events for cassette exons. Exons with multiple states behave like normal alternative exons with the addition that a change of state has a unit cost.

3 Results

3.1 Transcriptome Reconstruction

Next-generation sequencing methods yield an increasing amount of data and reconstructing transcripts from short reads is a complex problem [12]. Once reads are mapped on the genome and splice junctions are identified, a splice graph can be constructed—nodes represent exons and edges splice junctions. Any path on this graph is, thus, a potential transcript. The remaining problem is to identify the "true" transcripts within this graph.

Several methods exist to predict transcripts from ESTs. ESTGene, which is part of the Ensembl pipeline, reconstructs the minimal set of transcripts that cover the splice graph [7], [13]. ECGene, another method, parses the splice graph and clusters transcripts based on the nature of the splice sites [1]. ATP, the algorithm behind the ASPic database, is similar to ESTGene but includes additional rules to predict transcripts [9], [12]. Other methods such as Scripture [14], Cufflinks [15], or the EM algorithm by Xing et al. [16] also aim at transcriptome reconstruction but have no associated database. Remarkably, none of these methods make use of phylogenies.

3.1.1 Methods

We selected from the ASPic database 805 human genes that have an ortholog in rat, mouse, opossum, chimpanzee, marmoset, and macaque. These six species had the highest rate of alternative splicing in the Ensembl database while being the closest to human. Exons and transcripts were collected from the ASPic database for human and from Ensembl for the remaining species. Genes were aligned using MAFFT.

We refine the prediction of ASPic through a simple algorithm. For each human transcript present in the ASPic database, we collect the Ensembl transcripts of the homologous genes and reconstructed a transcript phylogeny on this set plus the ASPic transcript. The total cost of this transcript phylogeny—as defined in our model—is assigned as the score of the ASPic transcript. Note that the phylogeny will always contain only one human transcript; the rationale being the observation of how this particular transcript fits within the transcript phylogeny created by the homologous genes.

Once every ASPic transcript is assigned a score, transcripts are grouped by genes. Then, for each gene, the algorithm discards all transcripts that have an unreasonable evolutionary score within the gene's transcript set. The evolutionary score of a transcript depends on the number of gene exons, and its range may vary greatly between genes. Consequently, we considered the difference ratio within a gene. The algorithm searches for the two groups of transcripts that maximize the difference of their respective mean scores with respect to the maximum score—a two-mean clustering algorithm. For a given gene g, we have

$$r = \max_{T^\star \subset T_g} \left(\frac{E[T^\star] - E[T_g \setminus T^\star]}{S_{max}} \right), \quad (1)$$

where T_g is the set of all human transcripts in gene g, S_{max} the maximum score in T_g, and $E[X]$ is the average score of all transcripts in X. If r is larger than a threshold t, then the high-cost group is discarded. Therefore, if $t = 1$ the performance of the refinement algorithm will be equal to the original algorithm since no transcript is removed. The value of r can be easily determined by a simple algorithm: First sort T_g by scores, then test all possible splits.

TABLE 2
Results of the Different Algorithms Using Exact Matches on the RefSeq Database

	Sensitivity	Precision
ECGene	0.9%	1.2%
ASPic	7.23%	5.2%
Ref. ASPic ($c_D = 1$)	6.4%	11.5%
Ref. ASPic ($c_D = 10$)	7.0%	9.7%

Our refinement on the ASPic prediction yielded a slight decrease in sensitivity and a twofold increase in precision. Both refined results were achieved with the same threshold t but different transcript loss cost.

3.1.2 Results

We tested the performance of ECGene and ASPic by matching their predicted transcripts (from their online database) to the RefSeq database [10]—a gold standard for RNA sequences. A true positive was defined as an exact sequence match between the query and a sequence in RefSeq. A false positive is a transcript predicted by ASPic that could not be matched in RefSeq.

As shown in Table 2, both methods performed quite poorly. The best method, ASPic, can only recover 7 percent of all RefSeq transcripts. The use of transcript phylogenies yielded a significant increase in precision while withstanding a minor decrease in sensitivity. Note that since we filter the ASPic's transcripts, sensitivity cannot increase. The Ensembl database had a precision of 97 percent on the same set of genes. It is, however, impossible to assess the part played by ESTGene as the Ensembl database includes all sequences from RefSeq. The ECGene results could not be refined with our algorithm as exon information was not available; only transcript RNA sequences are available for download.

Following Bonizzoni et al. [12], we opted for a gentler setup where exact matches are not required at the end of the sequences—namely, the "loose" setup. A predicted transcript is a true positive if it is a substring of a sequence in RefSeq and if no additional exons could have been added by the algorithm. That is, there exists no predicted transcript that contains the missing ends. (ECGene does not provide any exon information and could not be assessed under this setup.) The ASPic database performed better—13.4 percent precision and 22.8 percent sensitivity—and our refinement method pushed the precision to 25.7 percent, again a twofold increase, while maintaining the sensitivity at 21.5 percent ($c_D = 10$).

We then varied the threshold t in our refinement algorithm and investigated the influence of different transcript death costs, models, and phylogenies on the performance of the refinement. (All experiments were performed under the loose setup because it provides a larger set of true positive.)

Two main setups were tested for the cost of transcript death: a fixed cost and a cost that is dependent on the average number of gene exons ($c_D = \alpha e$, where e is the average number of exons and $\alpha \in \Re^+$). Curves on Fig. 2

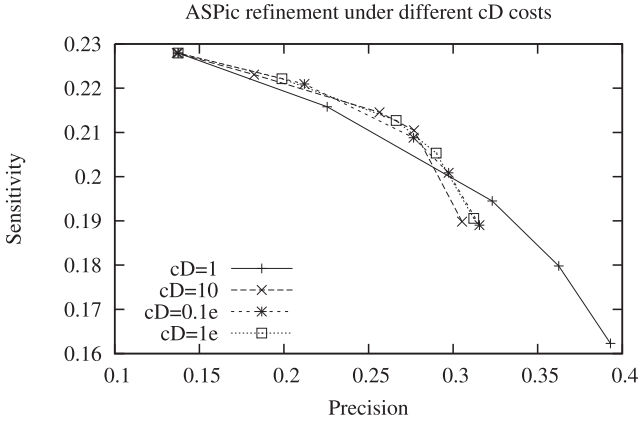


Fig. 2. Loose matches on RefSeq under the different transcript death costs and threshold t. All curves converge toward the ASPic performance at the top left corner. An optimal curve would go horizontally from the ASPic performance toward the right. Points on a curve represent results for $t = \{0.0, 0.06, 0.08, 0.16, 1.0\}$. $c_D = \alpha e$ represents a transcript death cost dependent on the average number of exons in the gene family; e being the average number of exons.

show that different costs have little influence on the final improvement. $c_D = 1$ is the only exception. The average number of exons across all gene families is 13.2. Thus, a cost of $0.1e$ (that is 0.1 times the average number of gene exons) should yield equivalent results. A quick study showed that the difference in sensitivity and precision between the two setups ($c_D = 0.1e$ and $c_D = 1$) is mainly due to gene families with 20 exons or less. If $c_D = 1$ or $c_D = 0.1e$ and $e = 10$, the loss or gain of an exon in a transcript is equivalent to losing the whole transcript. A small change in the cost can, thus, break the arbitrary choice and lead toward another phylogeny.

We then tested the influence of our two-level model of transcript evolution—gene events separated from transcript events—versus a simple parsimony model where every exon modification has a unit cost. From Fig. 3, one can observe the superiority of the two-level model over the simple model. (The simple model curve is always below the two-level model curve.) For instance at $t = 0.08$, the two-level model allows a gain of 3 percent in precision over the simple model with no loss in sensitivity.

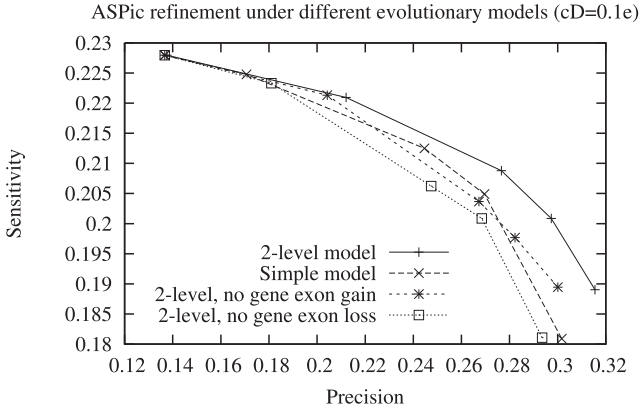


Fig. 3. Loose matches on RefSeq under the different evolutionary models. "Two-level model": our standard model of transcript evolution where gene events have zero cost. "simple model": our standard model of transcript evolution but with gene cost. For instance, the cost of losing an exon at the gene level is passed onto each transcript. The last two setups only affect the content of the ancestral genes. Points on a curve represent results for $t = \{0.0, 0.06, 0.08, 0.16, 1.0\}$; $t = 1.0$ being in the top left corner.

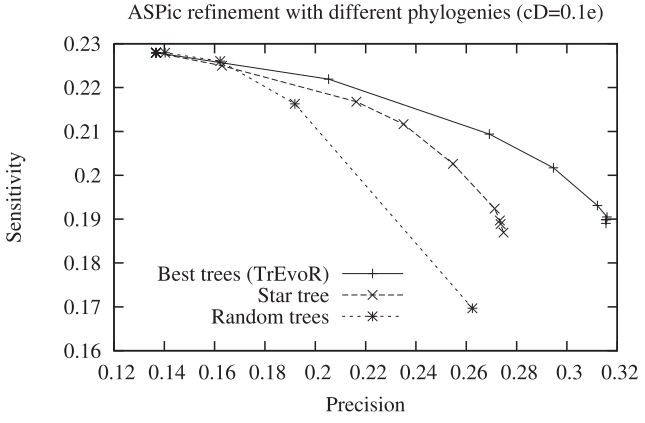


Fig. 4. Loose matches on RefSeq under the different phylogenies. Random transcript trees convey wrong information and thus display the worst behavior.

Additionally, we assessed the influence of the ancestral gene reconstruction procedure on the final result. The algorithm reconstructs the ancestral gene content through a simple Fitch's algorithm for the small parsimony problem. Every event—exon gain, loss, and mutation—has unit cost. Two extreme scenarios were designed. One that prohibits exon gain and another that excludes exon loss. As seen in Fig. 3, the absence of exon loss is more detrimental than the lack of exon gain. The absence of exon loss forces an evolutionary scenario where the same exon can be created in different branches of the gene tree; a statement that would imply convergent evolution. The latter is unlikely and, thus, explains the bad performance. The absence of exon gain conserves the evolutionary relationship but slightly alters its content. Nonetheless, this slight change is sufficient to lower the accuracy of our refinement method on the ASPic database.

A subsequent question that arose was the influence of the structure of the transcript trees on the final result. We, thus, tested three different phylogenies:

- *Best trees.* The reconstructed phylogeny output by our algorithm (TrEvoR).
- *Random trees.* Random transcript trees that still agree with the gene tree (average on 1,000 runs).
- *Star tree.* All transcripts are directly linked to the root. The gene tree is not considered any more but an ancestral gene is reconstructed and used in the cost computation (two-level model).

As expected, transcript trees reconstructed by our algorithm perform much better than a star phylogeny or random trees (see Fig. 4). The use of ancestral gene information for the star phylogeny—our two-level model—is also beneficial despite the use of a single gene as common and direct ancestor to all genes (data not shown). The latter demonstrates that the separation of transcript evolution and gene evolution is the correct model for studies related to alternative splicing. Unsurprisingly, a star phylogeny outperforms random trees. In random trees, the signal is completely lost as orthologous transcripts may not be on the same tree, thus conveying wrong information, whereas

on a star phylogeny no information is provided beyond the content of the transcripts. It then performs a simple comparison of the human transcript to all nonhuman transcripts. This is enough to gain some precision, but it cannot reach the performance of the best tree setup.

To further validate the method, a random score was assigned to each transcript (normal distribution). In this case, a variation of the threshold show a rapid decrease in sensitivity without any gain in precision. At $t = 0.0$, we reach 13.9 percent in precision (+0.2 percent) and 10.1 percent in sensitivity (−12.7 percent).

The previous experiment tested how a single transcript fits within the phylogeny. However, transcriptome reconstruction is about sets of transcripts. Therefore, we computed the cost of evolution for the Ensembl transcripts in the homologous genes plus different sets of human transcripts: all ASPic transcripts, no human transcripts (NO), and ASPic transcripts that were matched in RefSeq under the exact setup ($EXACT$) and loose setup ($LOOSE$). In this experiment, TrEvoR is only used to score different sets of human transcripts; our refinement algorithm is not applied. Note that human is the only species to be affected by these changes because we selected human genes from the ASPic predictions only. The NO setup corresponds to an evolutionary scenario where all human transcripts were lost in this gene family. The expected result is that the set of exactly matched transcript should have the lowest cost. We expect the loose match setup to sometimes have the lowest cost as the RefSeq database may not be exhaustive. We ran the algorithm on the 213 genes that had at least one exact match and observed that, for any transcript loss cost, the set of all transcripts always yielded a score equal or higher than any other set. With a transcript loss cost of 1, removing all transcripts yielded the minimum cost in 77 percent of the gene families. In many cases, the minimum score is shared among the NO, $EXACT$, and $LOOSE$ sets and the minimum was unique to the NO setup in 58 percent of the gene families. (Note that only 28 gene families have differences between the $EXACT$ and $LOOSE$ sets.) As shown in Fig. 5, a transcript loss cost of 10, $1e$, or $10e$ guarantees that the $EXACT$ or $LOOSE$ sets will have the minimum score.

A cost of $0.1e$ might not be the best choice as the removal of all human transcripts is more profitable in 36 percent of the cases. However, the first experiment (Fig. 2) showed that the performance of our refinement method under this transcript loss cost is similar to other cost setups. It might thus be advantageous to remove transcripts, but it does not affect the phylogeny reconstruction. However, a transcript loss cost equal to the cost of losing or gaining an exon should be avoided—the ideal cost should be larger.

3.2 Functional Study on Transcripts

To demonstrate the broad scope of our framework, we applied transcript phylogenies to the study of function in transcripts. We inquired whether transcript phylogenies carry any functional information and, in the positive case, if two different transcript trees vary in functions. Unfortunately gene ontology terms could not be used as most of the annotations were not transcript-specific. Moreover, the set of shared annotations across the seven species is quite small and mainly uninformative. We, thus, studied the correlation between protein domains—structurally stable regions that often correspond to specific functions—and transcript phylogenies.

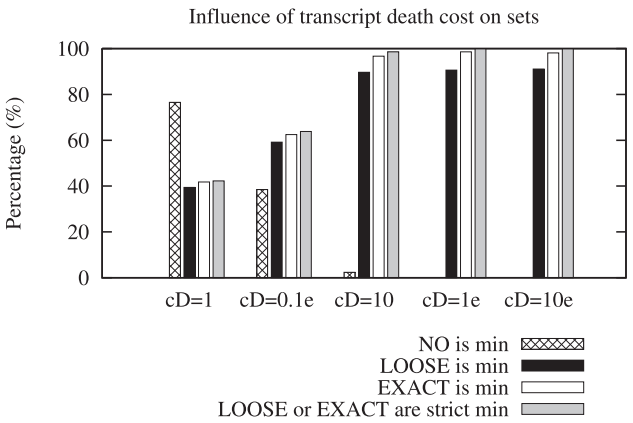


Fig. 5. Statistics on the evolutionary cost of different subsets of the predicted transcripts. Values indicate the percentage of genes that fit the condition. The first three setups represent cases when the minimum score belongs to one of these sets. ("NO is min" means that the NO set contain the minimum value but might not be unique.) The last setup indicates the occurrences where the choice of $LOOSE$ or $EXACT$ will yield the minimum score. One can see that higher cost value have a higher correspondence with the expected behavior.

3.2.1 Methods

We used the same data set, seven species and 805 genes, as for the transcriptome reconstruction problem. Transcripts and domain annotations were retrieved from the Ensembl database and transcript phylogenies were reconstructed with our algorithm. The sole parameter, the cost of transcript loss, was set as a proportion of the average number of exons in a set of homologous genes and three values were tested: $0.1e$, $1e$, and $10e$. (e is the average number of exons in the gene family.) Different domain annotation databases, all available as tracks in Ensembl, were selected.

- *InterPro.* An integrated resource for protein families, domains, regions, and functional sites. It combines data from several databases such as *PROSITE*, *PRINTS*, *SMART*, *SUPERFAMILY*, *Pfam*, and many others.
- *Pfam.* A database of protein families identified by sequence alignments and hidden Markov models.
- *PROSITE.* A collection of protein domains, families, and functional sites identified through patterns and profiles.
- *SMART.* A database of well-annotated protein domains.
- *SUPERFAMILY.* A set of hidden Markov models that represent domains at the superfamily level in SCOP (structure-based classification of proteins).
- *SEG.* A software that divides the sequence into low- and high-complexity regions.
- *Transmembrane.* Identification of transmembrane helices with TMHMM.

We tested the transcript phylogenies for robustness and found 135 gene families where transcripts were grouped under the same parents across the three different transcript

loss costs (c_D). (The complete gene list is available in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2012.145.)

A test of robustness is necessary as a slightly different c_D may affect the transcript phylogeny (for instance through an extra join operation). If the phylogeny is robust, that change will occur at the higher levels of the phylogeny were uncertainty is high and multiple solutions exist. Hence, the impact on the tree structure close to the leaves will be minimized. Consequently, gene families that have the lower structure of their transcript phylogeny unaffected by parameter changes are good candidates for having well-conserved ancestral transcripts.

In this study, we investigated the difference in functional content between the different transcript trees of a transcript phylogeny. If the ancestral transcripts had the same functional content, then the distribution of the domain content in the leaves of each transcript tree should be roughly equal to the distribution of the domains across all transcripts of the gene family. Therefore, for each gene family F, we computed the probability of a domain d to appear in a transcript. This is our background probability,

$$P_F[d] = \frac{\text{Nb of transcripts in } F \text{ that contain } d}{\text{Nb of transcripts in } F}. \quad (2)$$

A similar value was computed for each transcript tree t of the transcript phylogeny

$$P_t[d] = \frac{\text{Nb of leaf-transcripts in t that contain } d}{\text{Nb of leaf-transcripts in } t}. \quad (3)$$

Note that we only account for the presence of the domain. The number of occurrences per transcript does not matter. Should the transcript phylogeny contain only one tree, then we have that $\forall\, d\ P_F[d] = P_t[d]$.

Consequently, we selected phylogenies with multiple trees and computed, for all domains, the deviation from the background probability. For a given domain, we have, thus,

$$Dev[d] = \frac{1}{|T|}\sum_{t\in T}(P_t[d] - P_F[d])^2, \quad (4)$$

where T represent the set of transcript trees in the phylogeny. The mean value over all domains gives us an indication of the global deviation of the domain content in the trees from the domain content in all transcripts. That is,

$$E_{Dev} = \frac{1}{|D|}\sum_{d\in D} Dev[d], \quad (5)$$

where D is the set of domains in a given Ensembl track. Note that the E_{Dev} value depends on the Ensembl track (D), the gene family (F), and the transcript phylogeny (T).

To test if the 135 "stable" genes had a different deviation from the rest, we performed a random sampling among the 805 genes. The sampling was repeated a hundred times then averaged. We refer to this sampling as the "unstable set." This set still represents valid transcript phylogenies—as reconstructed by our algorithm and model—but may not be as informative as the stable set. Additionally, we compared the stable set to random phylogenies. For each phylogeny in the 135 stable gene families, we randomly reassigned all transcripts to another transcript trees. (The number of transcript trees remain unchanged.) The random assignment was performed a hundred times and results averaged.

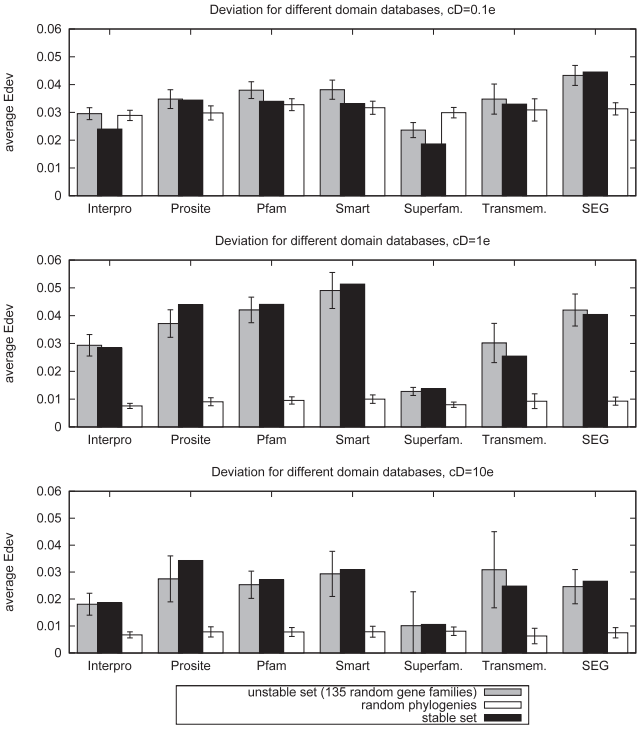


Fig. 6. Deviation from the expected domain presence for different domain annotation databases. For each database, values are computed by averaging E_{Dev} across the gene families. As the unstable set contains 670 gene families, 135 families were randomly sampled from it to achieve a fair comparison with the stable set. The *PROSITE* is the only database to display a significantly higher deviation in the stable set when compared to the unstable set. In most databases, the reconstructed tree differs by more than one standard deviation from a random phylogeny, especially if the transcript loss cost is larger than $0.1e$.

Note that for this experiments, the tree structure of the phylogeny does not matter. The only information considered is the different groups of transcripts created by the phylogeny—a group being equal to the leaves of a transcript tree.

3.2.2 Results

In most cases, the phylogeny reconstructed under our model was significantly different—more than one standard deviation—from a random phylogeny (see Fig. 6). With a low transcript loss cost, $0.1e$, the *Pfam*, *SMART*, and *SUPERFAMILY* databases showed results similar to random. That is, under this transcript loss cost, any phylogeny with the same number of trees will display a similar deviation in domain content. However, for $c_D = 1e$ or $c_D = 10e$, the behavior is very far from random. *SUPERFAMILY* is the exception as its deviation is only one standard deviation away from the random phylogeny. The *SUPERFAMILY* database clusters domains with a higher abstraction level than families and may not be very informative across transcripts—all members will contain the domain.

The variance for both the unstable and random set seems to increase with the transcript loss cost. That is explained by

the design of our analysis. We only consider phylogenies with multiple transcript trees and we known that an increasing transcript loss cost tends to lower the total number of trees. Consequently, as the number of data points decreases, the variance is bound to increase.

As shown in Fig. 6, the deviation between the stable and the unstable sets differ mainly for low values of c_D. The stable set, except for *PROSITE*, has always a lower deviation than the average on the unstable set. As the cost increases, the stable set converges towards the unstable set. Remarkably, *Transmembrane* and *SEG* did not display any significant differences between unstable and stable sets for any values of c_D. A study by Cline et al. [17] showed that transmembrane regions are not likely correlated with alternative splicing—a finding that endorses our results. Interestingly, the *PROSITE* database is the only one to exhibit a higher deviation for the stable set (at $c_D = 1e$). It is also the only database to include functional sites. The *InterPro* database does include the *PROSITE* database, but its behavior resembles the other databases. *InterPro* collects data from 11 databases. The *PROSITE* functional sites annotations may thus be a minority and have little influence on the global result.

To test if the difference in *PROSITE* for $c_D = 1e$ was indeed significant and not a visual artifact, we performed a one-way ANOVA on *PROSITE*, *Pfam*, *SMART*, and *SUPERFAMILY* and another on the same databases but without *PROSITE*. The first analysis returned a p-value of $1.5036 \cdot 10^{-28}$, while the second returned a value of 0.1632. Consequently, we can confidently reject the null hypothesis—all means are equal—in the first case but not in the second. This indicates that the *PROSITE* database has a significantly larger deviation than the other databases between the two sets. Note that these databases may cover similar domains and consequently may not be independent, which poses a problem for statistical analyses.

All databases, except *PROSITE*, have a lesser deviation than the unstable set for low c_D value. The *PROSITE* exception can be explained by an averaging effect. Proteins domains tend to yield a smaller deviation while functional sites create a larger deviation. The combination of both averages the score and results in a deviation similar to the random set. The *SUPERFAMILY* has, globally, a lower deviation than any other database. As mentioned before, the *SUPERFAMILY* database clusters domains with a higher abstraction level than families. For instance, two different domain families may be reunited under a single superfamily and thus lower the deviation.

We also tested the stable set against the 805 genes for GO enrichment with GOrilla [18] but could not find any over- or underrepresented ontologies.

Based on these results, one could conjecture that well-conserved transcripts contain similar sets of domains but different functional sites. A deeper study could focus on the functional sites and potentially identify unknown functions in some transcripts. Conserved exons are also known to display a high correlation with protein domain boundaries [4]. Another study could thus include the age of domains and exons into the analysis. The stable set displayed a behavior that is consistent across all c_D and quite often differs significantly from a random phylogeny. Hence, it is more informative than the unstable set. These gene families may have strong conservation across transcripts and such sets—gene families where transcript phylogenies are consistent across different c_D—should be used for further analysis.

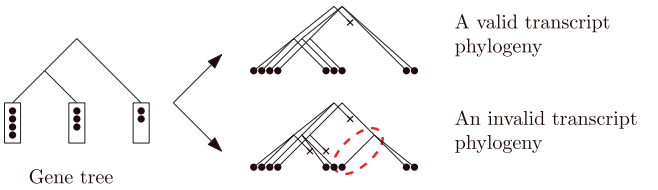


Fig. 7. Small example on a gene family with three genes containing 4, 3, and 2 transcripts. A transcript tree has to agree with the gene evolution (gene tree). The faulty connection in the invalid transcript phylogeny is encircled.

4 Reconstruction of Transcript Phylogenies

Reconstructing transcript phylogenies is nontrivial: The problem is at least as hard as the standard phylogeny reconstruction problem, which is NP-hard. In a standard tree reconstruction, the tree structure is unknown, but we know that only one tree exists. In a transcript phylogeny reconstruction, the tree structure is partially known, as it has to agree with the gene tree, but the number of trees is unknown. Fig. 7 show some valid and invalid transcript phylogenies.

Our previous algorithm did not scale well; hence, we designed a heuristic based on neighbor-joining and packaged it into a convenient tool: TrEvoR. The latter is available online and can be downloaded at http://lcbb.epfl.ch/trevor. A manual and some toy examples are also present.

4.1 TrEvoR Algorithm

Our two-level model of transcript evolution depends on the gene structure and, similar to our previous algorithm, the first step is thus to reconstruct the exons of the ancestral genes. Sankoff's algorithm for the small parsimony problem is applied and backtracking yields the ancestral states [19]. Our algorithm searches then for the most parsimonious forest of transcript trees.

Note that orthologous exons have first to be assigned. The representation of alternative transcripts across different species is an issue that have been recently addressed by Ouangraoua et al. [20]. Similar to their model, we define an orthologous position—loci—as a set of exons that overlap in a multiple alignment of the gene sequences. Following Takeda et al., we set the overlap threshold at 70 percent [11]. An exon is then uniquely assigned to a loci. Within a loci, exons that do not overlap reciprocally are assigned to different identifiers.

In a classical tree reconstruction problem, standard parsimony methods use a specific operation to search the tree space—nearest neighbor interchange, subtree pruning and regrafting, or tree bisection and reconnection. In our case, we define the *join* operation, which merges two transcripts into one. The join operation simply assigns all children of transcript t_A to transcript t_B, deletes transcript t_A, and updates transcript deaths in t_B. To preserve the consistency between the gene tree and the transcript tree,

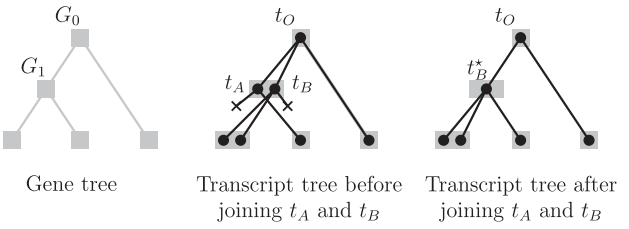


Fig. 8. A join operation on a simple transcript tree. JOIN(t_A, t_B) is valid because they share a common ancestor and belong to the same gene, G_1. Note that the two transcript deaths are lost after the operation.

a join operation is only possible if these three conditions are met:

1. t_A and t_B are part of the same transcript tree.
2. They belong to the same gene. That is $Gene(t_A) = Gene(t_B)$.
3. They share a common ancestor in the transcript tree ($\exists t$ such that (t, t_A) and (t, t_B) are edges of the transcript tree) or are ancestral transcripts associated with the ancestral gene at the root of the gene tree.

Our algorithm explores the forest space through a recursive neighbor-joining algorithm.

The algorithm starts with a phylogeny that contains the maximum number of transcript trees—one per current transcript—then applies a neighbor-joining algorithm under the join operation until a single transcript tree is left. At each iteration, all possible join operations on the root of two transcript trees are tested, the best candidates are retained and their roots are joined. The algorithm moves then to the next iteration. A sketch of the algorithm is described in Fig. 9. This part of the algorithm is not truly greedy, as the selected forest may have a suboptimal score. However, it forces a full traversal of the search space. (A classical greedy algorithm would stop as soon as the solution cannot be improved.) When testing for a possible join operation of two roots, we apply a recursive procedure to find the lowest possible score of this new transcript tree. The score of a tree is tested by propagating the join operation from the root to the leaves. For each possible join operation on two roots, we apply a recursive algorithm to find the best transcript tree—Algorithm 1. The latter is a truly greedy algorithm and thus cannot guarantee an optimal solution. Note that any join operation that was done when computing the score of the new transcript tree is undone before passing on to the next iteration.

Algorithm 1. A recursive Neighbor-Joining algorithm using the JOIN operation.

Procedure RECNJ(Transcript t)
  if no children of t can be joined **then**
    return COMPUTESCORE(t) ▷ Compute the score of the tree containing t.
  $s_{best} = \infty$
  while some children of t can be joined **do**
    $s^\star = \infty$
    for (a, b) s.t. $a, b \in$ CHILDRENOF(t) **do**
      JOIN(a, b) ▷ Assume that joining a and b is feasible.
      $s =$ RECNJ(b)
      UNJOIN(a, b) ▷ Revert to the situation before joining a and b.
      if $s < s^\star$ **then**
        $s^\star = s$
        $a^\star = a$ and $b^\star = b$ ▷ Save the best join.
    JOIN$(a^\star, b^\star)$ ▷ Apply the best join.
    $s =$ RECNJ$(b)^\star$ ▷ Iterate on the "new" node.
    if $s < s_{best}$ **then**
      $s_{best} = s$
  return s_{best}

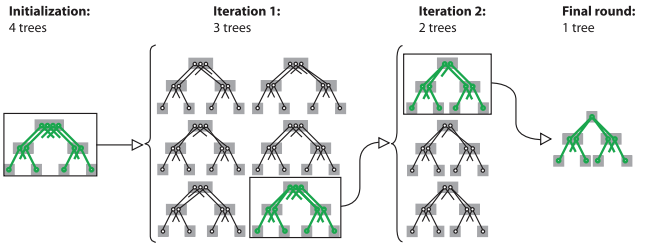


Fig. 9. Example of traversal of the forest space on a full binary gene tree (four genes) with one transcript per gene. Transcripts are represented by circles and genes by gray rectangles. At each step, all possibilities obtained by a join operation on the roots are evaluated and the best tree (after the recursive procedures) is selected for the next step. The best phylogeny at each iteration is retained as a candidate for the final decision.

Fig. 8 illustrates the join operation on a simple example.

Tested on simulated data, this algorithm (a mix of greedy and nongreedy neighbor-joining methods) yielded the best compromise between accuracy and rapidity ($O(n^4)$) when compared to a fully greedy approach.

5 Conclusion

We presented a model of transcript evolution and an associated tool, TrEvoR, to reconstruct transcript phylogenies. The model represents the evolution of transcripts as a second layer above the exon evolution.

On 805 genes from the ASPic database, we demonstrated that transcript phylogenies can enhance transcriptome reconstruction from ESTs. The use of transcript phylogenies doubled the precision while retaining a similar sensitivity. Results also showed that our two-level model performed better than a gene-centric model. This implies that a transcript-focused approach is more powerful for this particular task.

Additionally, we broadened the scope of transcript phylogenies by correlating them with the protein domains of their isoforms. It turned out that transcript trees indeed contain useful functional information and may be used in studies on function evolution. Domain information was gathered from different tracks in Ensembl and results revealed that conserved transcripts show a greater variability in functional sites than in protein domains.

Future work can be directed in several directions. Different models—for instance, a model based on splice sites and not exons—and different hypotheses can be tested through TrEvoR. The accuracy of automated pipelines for transcriptome reconstruction could be improved by developing a method that includes transcript phylogenies of model organisms. Deeper studies on functional sites within

a transcript phylogeny framework could shed some light on the evolution of functions.

In previous work, we proposed the concept of transcript phylogenies and demonstrated its feasibility. Here, we applied this concept to two large-scale analyses, demonstrated good improvements on transcriptome reconstruction, new findings on the evolution of function in transcripts, and consequently validated the usefulness of our method for transcriptome studies.

REFERENCES

[1] N. Kim, S. Shin, and S. Lee, "ECgene: Genome-Based EST Clustering and Gene Modeling for Alternative Splicing," *Genome Research,* vol. 15, no. 4, pp. 566-576, 2005.

[2] B. Modrek and C. Lee, "A Genomic View of Alternative Splicing," *Nature Genetics,* vol. 30, no. 1, pp. 13-39, Jan. 2002.

[3] H. Keren, G. Lev-Maor, and G. Ast, "Alternative Splicing and Evolution: Diversification, Exon Definition and Function," *Nature Rev. Genetics,* vol. 11, no. 5, pp. 345-55, May 2010.

[4] I.I. Artamonova and M.S. Gelfand, "Comparative Genomics and Evolution of Alternative Splicing: The Pessimists' Science," *Chemical Rev.,* vol. 107, no. 8, pp. 3407-3430, Aug. 2007.

[5] B. Harr and L.M. Turner, "Genome-Wide Analysis of Alternative Splicing Evolution among MUS Subspecies," *Molecular Ecology,* vol. 19, Suppl 1, pp. 228-239, Mar. 2010.

[6] R.N. Nurtdinov, "Low Conservation of Alternative Splicing Patterns in the Human and Mouse Genomes." *Human Molecular Genetics,* vol. 12, no. 11, pp. 1313-1320, June 2003.

[7] P. Flicek et al., "Ensembls Tenth Year," *Nucleic Acids Research,* vol. 38, no. suppl 1, pp. D557-D562, 2010.

[8] Y. Christinat and B. Moret, "Inferring Transcript Phylogenies," *Proc. IEEE Int'l Conf. Bioinformatics and Biomedecine,* pp. 208-215, 2011,

[9] P. Martelli et al., "ASPicDB: A Database of Annotated Transcript and Protein Variants Generated by Alternative Splicing," *Nucleic Acids Research,* vol. 39, no. suppl 1, pp. D80-D85, 2011.

[10] K. Pruitt et al., "NCBI Reference Sequences: Current Status, Policy and New Initiatives," *Nucleic Acids Research,* vol. 37, no. suppl 1, pp. D32-D36, 2009.

[11] J.-i. Takeda, Y. Suzuki, R. Sakate, Y. Sato, T. Gojobori, T. Imanishi, and S. Sugano, "H-DBAS: Human-Transcriptome Database for Alternative Splicing: Update 2010." *Nucleic Acids Research,* vol. 38, pp. D86-D90, http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2808982&tool=pmcentrez&rendertype=abstract. Jan. 2010.

[12] P. Bonizzoni et al., "Detecting Alternative Gene Structures from Spliced ESTs: A Computational Approach," *J. Computational Biology,* vol. 16, no. 1, pp. 43-66, 2009.

[13] E. Eyras et al., "ESTGenes: Alternative Splicing from ESTs in Ensembl," *Genome Research,* vol. 14, no. 5, pp. 976-987, 2004.

[14] M. Guttman et al., "Ab Initio Reconstruction of Cell Type-Specific Transcriptomes in Mouse Reveals the Conserved Multi-Exonic Structure of Lincrnas," *Nature Biotechnology,* vol. 28, no. 5, pp. 503-510, 2010.

[15] C. Trapnell et al., "Transcript Assembly and Quantification by RNA-Seq Reveals Unannotated Transcripts and Isoform Switching During Cell Differentiation," *Nature Biotechnology,* vol. 28, no. 5, pp. 511-515, 2010.

[16] Y. Xing et al., "An Expectation-Maximization Algorithm for Probabilistic Reconstructions of Full-Length Isoforms from Splice Graphs," *Nucleic Acids Research,* vol. 34, no. 10, pp. 3150-3160, 2006.

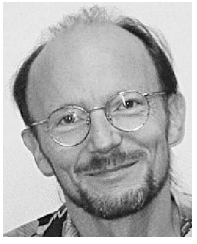
[17] M. Cline et al., "The Effects of Alternative Splicing on Transmembrane Proteins in the Mouse Genome," *Proc. Pac Symp. Biocomput,* vol. 2004, pp. 17-28, 2004.

[18] E. Eden et al., "GOrilla: A Tool for Discovery and Visualization of Enriched Go Terms in Ranked Gene Lists," *BMC Bioinformatics,* vol. 10, no. 1, p. 48, 2009.

[19] D. Sankoff, "Minimal Mutation Trees of Sequences." *SIAM J. Applied Math.,* vol. 28, no. 1, pp. 35-42, 1975.

[20] A. Ouangraoua, K.M. Swenson, and A. Bergeron, "On the Comparison of Sets of Alternative Transcripts," *Proc. Eighth Int'l Conf. Bioinformatics Research and Applications (ISBRA),* pp. 201-212, 2012,

Yann Christinat received the scientific maturité at the Gymnase de Nyon and then studied biology at the University of Geneva. After a successful first year, he switched to computer sciences at EPFL to study bioinformatics where he received the MSc degree with a specialization in biocomputing in 2006. During his studies, he spent a year at the Linköping Institute of Technology in Sweden as an exchange student and accomplished his master thesis at Siemens Corporate Research, Princeton, where he developed a novel biclustering algorithm for microarray data. At the beginning of 2007, he joined professor Bernard M.E Moret's laboratory for Computational Biology and Bioinformatics in EPFL as a PhD student where he obtained the title of Docteur ès Sciences in June 2012. His research interests range from protein domain prediction to the mechanism alternative splicing and evolution. His work on the evolution of alternative transcripts received the Best Student Paper Award at the IEEE Conference on Bioinformatics and Biomedecine in 2011.



Bernard M.E. Moret received the PhD degree from the University of Tennessee in 1980, and was on the faculty of the Department of Computer Science at the University of New Mexico until 2006, serving as a chairman from 1991 to 1993. He is a professor of computer science, holding the chair of bioinformatics at the EPFL, the Swiss Federal Institute of Technology in Lausanne, Switzerland. His research interests include the area of algorithms and applications, particularly in computational molecular biology. He founded the *ACM Journal of Experimental Algorithmics* in 1995, serving as its editor-in-chief for seven years. Since 2000, he has focused on the development of models and algorithms for evolutionary genomics, publishing around 100 peer-reviewed articles in the area and founding, in 2001, the annual Workshop on Algorithms in Bioinformatics.