

# Temporal Logics for Phylogenetic Analysis via Model Checking

Roberto Blanco, Gregorio de Miguel Casado, José Ignacio Requeno and José Manuel Colom

*Department of Computer Science and Systems Engineering (DIIS), Aragon Institute of Engineering Research (I3A)*

*Universidad de Zaragoza*

*Zaragoza, Spain*

*{robertob, gmiguel, nrequeno, jm}@unizar.es*

**Abstract**—The need for general-purpose algorithms for the study of biological properties in phylogenetics motivates the research in formal verification frameworks so that researchers can focus their efforts exclusively on evolution modelling and property specification. To this end, model checking, a mature automated verification technique from computer science, is proposed for phylogenetic analysis. Three cornerstones found our approach: modelling evolution dynamics with transition systems; specifying phylogenetic properties using temporal logic formulae; and verifying the latter by means of automated computer tools. As prominent advantages stemming of studying phylogenetic properties with this approach, different models of evolution can be considered, complex properties can be specified as the logical composition of others, and the refinement of unfulfilled properties as well as the discovery of new ones can be undertaken by exploiting the verification results.

**Keywords**-phylogenetic analysis; formal verification; temporal logic; model checking

## I. INTRODUCTION

Since its inception, phylogenetics has adopted tree-based models as conceptual structures for arranging and analyzing the evolutionary relationships subject to study.

The successive incorporation of formal methods for characterizing key concepts such as rigorous evolutionary distances has provided valuable results. Thus, tangible numerical relationships between sets of populations or individuals characterised by particular biological features have been explored for tree-building through distance-based methods (e.g., neighbour joining), and character-based methods (parsimony, likelihood, Bayesian inference) so as to infer structural properties of interest to biologists [1].

The inherent temporal nature of phylogenetic data suggests the possibility of introducing novel formal methods capable of improving the lack of flexibility of conventional models to incorporate such features as future prediction, embedding of information from the past and combination of evolutive rules of various abstraction levels. In this sense, we explore the features of model checking, a paradigm stemming from computer science based on temporal logics, which has been successfully applied in the industry for system modelling and verification [2]. In the context of computational biology, this paradigm has been applied to fields in which (mostly) quantitative properties over temporal data are analysed. Cellular and molecular interaction networks

[3], including cell cycle kinetics [4], are the main areas of application, but problems as varied as prediction of protein folding dynamics have been considered [5].

Model checking is an automated verification technique that, given a finite state model of a system and a formal property, systematically checks whether this property holds for (a given state in) that model. The model checking process consists of three phases: modelling with a description language (formalize both system and properties), running (check property validity with a model checking computer package) and analysing the results (study the counterexamples if a property is not satisfied).

Bearing in mind that “any verification using model-based techniques is only as good as the model of the system”, one of the most interesting strengths of model checking is the fact that “it is a potential push-button technology” because its use “requires neither a high degree of user interaction nor a high degree of expertise” [6] and there already exists a plethora of software tools for automated verification [2].

Assuming this fact, the aim of this paper is to show that this can be a unifying formalism which permits the phylogeneticist to focus on models and specifications (phylogenetic properties) using propositions of temporal logic, instead of on implementation issues concerned with particular algorithms. Moreover, verification in this context works in a closed-loop helping the phylogeneticist to refine or discover properties using counterexamples obtained from unfulfilled properties. Furthermore, the introduction of this paradigm represents an opportunity to handle the increasing complexity of the qualitative structural properties required by biologists, of such importance as detection of potentially deleterious mutations.

The paper is arranged in four sections. After this introduction, Section II develops the central issues of model checking in relationship with those meaningful in phylogenetics: phylogenies as a dynamic models of evolution, phylogenetic specifications by means of computational logics (CTL), and system verification via model checking. Next, Section III typifies and exemplifies the modelling of non-trivial structural properties of cladistic classification and sequence composition. Finally, Section IV draws the conclusions from this research and outlines future research work.

## II. BRIDGING WORLDS: PHYLOGENETICS AND MODEL CHECKING

*Transition systems* [6, Def. 2.1] are powerful formal models for the study of concurrent systems. Given an adequate model counterpart of a system and a suitable logic to reason about its conduct, it is possible to achieve automated, exhaustive verification of properties of interest. In this section we show that phylogenies can be naturally understood as and represented by such models. We begin by identifying the foundations of evolution with those of transition systems, following with the study of their temporal nature and its logical formulation, and conclude with an overview of verification under the model checking paradigm.

### A. Evolution as a Transition System

At the highest level of abstraction, phylogenies represent partial models of the evolution of sets of living organisms. Generally, they can be modelled as directed graphs, though phylogenetic trees suffice for most common purposes and will be adopted throughout the paper.

Trees offer a realistic model of aggregated evolution, in which each vertex represents a population of related individuals who mate among themselves and are denoted by a common, compatible heritage (DNA). Transformative processes that modify the heritable information give rise to new populations, which are reflected in oriented parent-child edges in the graph. Note that neither time nor ordering of child populations are part of the model, which is consistent with the descent semantics of the tree.

*Definition 1 (Rooted Labelled Tree):* Let  $\Sigma$  be a finite alphabet and  $l$  a natural number. A phylogenetic tree over  $\Sigma^l$  can be represented as a tuple  $P = (T, r, D)$ , where:

- $T = (V, E)$  is a tree graph,
- $r \in V$  is its root, and
- $D : V \rightarrow \Sigma^l$  is a dictionary function that labels each vertex of the tree with its associated taxon sequence.

Trees are typically built from finite sets of words of uniform length resulting from alignment algorithms. These words, which represent present-time taxa, are required to be in bijective correspondence with the set of leaves of the trees. Nevertheless, generalised encodings may be considered.

Restrictions can be added or removed to adjust the phylogeny model as needed, but in any case the nature of trees as *reactive systems* should become clear by now. They are composed of independent processes (individuals or populations) that interact profusely with their environment and may extend indefinitely in time, and whose most prominent feature is their *state*, i.e., their hereditary—though not invariable—information or a suitable portion thereof. Consequently, it is possible to naturally effect modelling and verification of evolutionary systems, and to this end data structures for representation of transition systems become a very attractive solution.

*Definition 2 (Kripke Structure):* Let  $AP$  be a set of *atomic propositions*, i.e., Boolean predicates that describe the observable properties of a state. A Kripke structure over  $AP$  is a finite transition system represented by a tuple  $M = (S, S_0, R, L)$ , where:

- $S$  is a finite set of states,
- $S_0 \subseteq S$  is the set of initial states,
- $R \subseteq S \times S$  is a total transition relation between states, i.e., for every state  $s \in S$ , there exist  $t \in S$  such that  $(s, t) \in R$ , and
- $L : S \rightarrow 2^{AP}$  is the labelling function that associates each state with the subset of atomic propositions that are true of it.

A Kripke structure models a system that is capable of an infinite number of behaviours or *paths*, infinite sequences of successive states  $\pi = s_0 s_1 s_2 \dots$  such that  $s_0 \in S_0$  and  $(s_i, s_{i+1}) \in R, i \in \mathbb{N}$ . The set of possible executions (paths) in a structure can be unfolded into its *computation tree*.

Here we focus on with Kripke structures that represent a certain tree as a (hopefully the real) computation of the evolutionary process, or rather the set of computations that result in the hypothesised patterns of evolution. Relations between states and atomic propositions, and between tree branches and state transitions, are chief in importance, and raise some interesting issues that need to be addressed:

- Ideally, the state of a process or population is uniquely identified by its DNA sequence. Sequences also determine the atomic propositions that form the basis of logical properties: the presence of a certain alphabet symbol at a given position. If separate vertices of the tree must share a sequence, their states can be enriched with auxiliary properties to preserve each its unique identity.
- Once a one-to-one correspondence between tree vertices and states has been established, it remains to resolve the infinite-path semantics of Kripke structures for what is essentially a present-time, local, tentative snapshot in the execution of a potentially infinite system. In order to deadlock terminal vertices, it suffices to add self-loops to their states, though finite-path relaxations of Kripke structures are also possible [7].

At this juncture, we can define a suitable *branching-time structure* for phylogenetic trees, which will form the basis for the interpretation of temporal logic formulae that express properties of the trees. The most common state formula determines whether the present state is associated with a sequence  $s = \sigma_1 \sigma_2 \dots \sigma_l \in \Sigma^l$  (or possibly a partial sequence or a set of sequences). Sequences will be manipulated symbolically, as will sets, as the aggregation of their parts. In logical terms:

$$s \equiv \bigwedge_{i=1}^l s[i] = \sigma_i \quad (1)$$

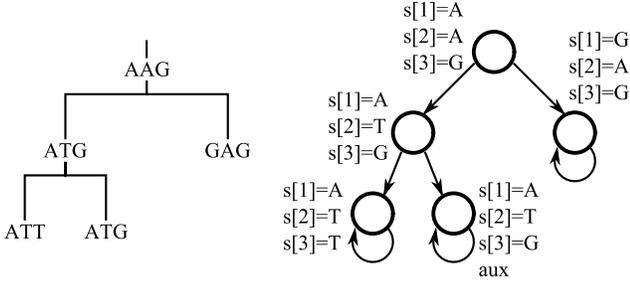


Figure 1. Translation from a tree to a Kripke structure.

**Definition 3 (Branching-time Phylogeny):** A tree (per Def. 1)  $P = (T, r, D)$  is univocally defined by the Kripke structure  $M_P = (V, \{r\}, R_P, L_P)$ , where:

- $R_P$  is the transition relation composed of the set of tree edges (directed from  $r$ ) plus self-loops on leaves:  $R_P = E \cup \{(v, v) : \nexists (v, w) \in E \wedge v, w \in V\}$ , and
- $L_P$  is the standard labelling function defined by  $AP_P$ , under which a state  $v$  mapped to  $D(v) = \sigma_1\sigma_2 \dots \sigma_l$  satisfies the family of properties  $s[i] = \sigma_i, 1 \leq i \leq l$ , plus any others necessary to preserve the unique logical identity of the state.

Figure 1 illustrates this translation process from a (phylogenetic) rooted labelled tree (Def. 1) to a Kripke structure (Def. 2). It should be noted that other models of evolution can be easily considered by adjusting Defs. 1–3 as needed.

### B. Temporal Logic as a Specification Language

Temporal logics are formal systems that allow the representation and manipulation of logical propositions qualified in terms of time [8]. In the context of transition systems, they are used to define properties on sequences of transitions between states of a system through a convenient abstraction of it (in the present case, a specific kind of Kripke structure). For example, properties may express whether it is possible at a point that one particular type of state may be reached or whether a certain property will always hold.

Temporal logics are principally classified according to how they treat sequences of events: whereas *linear-time logics* concern themselves with individual paths  $\pi = s_0s_1s_2 \dots$ , *branching-time logics* take into account the set of possible progressions from each state, hence reasoning about the whole computation tree. *Computational Tree Logic* (CTL) is a versatile exponent of the latter type which has been widely adopted by the model checking community [9]. As phylogenies represent evolutionary processes that are essentially branching in nature, branching-time logics in general, and CTL in particular, are remarkably well suited to their description.

CTL reinterprets the quantifiers of first-order logic as *path quantifiers*, expressing the fulfilment of a property throughout all computation paths (**A**), or at least one computation path (**E**). These two must be immediately qualified

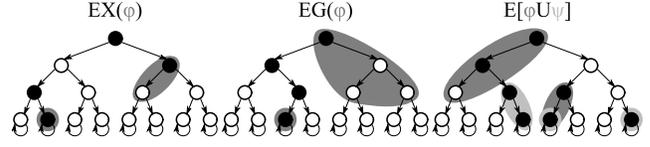


Figure 2. Evaluation of temporal logic operators.

by one of five *temporal operators*, of which three express the satisfaction of a property eventually in time (**F**), at all times (**G**), or in the next state (**X**); and two are conditional constructs in which a precedent is verified until a consequent comes into force (**U**), or until and including the moment when it does —if it does (**R**). A complete grammar and semantics of CTL formulae can be defined from a small subset of representative logical operators as follows; Fig. 2 illustrates the semantics of such a set.

**Definition 4 (Phylogenetic Tree Logic):** An arbitrary temporal logic formula  $\phi$  is defined by the following minimal grammar, where  $p \in AP$ :

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \mathbf{EX}(\phi) \mid \mathbf{EG}(\phi) \mid \mathbf{E}[\phi \mathbf{U} \psi] \quad (2)$$

Formulae are checked against a structure  $M$  considering all paths  $\pi$  that originate from a certain state  $s_0$ .  $M, s_0 \models \phi$  signifies that  $s_0$  satisfies  $\phi$ . The semantics for verification of well-formed formulae is as follows (let  $\pi = s_0s_1s_2 \dots$ ):

- $M, s_0 \models p \Leftrightarrow p \in L(s_0)$ ,
- $M, s_0 \models \neg\phi \Leftrightarrow M, s_0 \not\models \phi$ ,
- $M, s_0 \models \phi \vee \psi \Leftrightarrow M, s_0 \models \phi$  or  $M, s_0 \models \psi$ ,
- $M, s_0 \models \mathbf{EX}(\phi) \Leftrightarrow \exists \pi : M, s_1 \models \phi$ ,
- $M, s_0 \models \mathbf{EG}(\phi) \Leftrightarrow \exists \pi : M, s_i \models \phi, \forall i \in \mathbb{N}$ , and
- $M, s_0 \models \mathbf{E}[\phi \mathbf{U} \psi] \Leftrightarrow \exists \pi, i \in \mathbb{N} : M, s_i \models \psi$  and  $M, s_j \models \phi, 0 \leq j < i$ .

A CTL formula  $\phi$  represents a property that may be verified at certain states in the computation tree. In this context, a system  $M$  satisfies  $\phi$  iff every one of its initial states does:  $\bigwedge_{s_0 \in S_0} M, s_0 \models \phi$ . For legibility, we may adapt the notation of conditional operators as per  $\mathbf{E}[\phi \mathbf{U} \psi] \equiv \{\phi\} \mathbf{EU} \{\psi\}$ .

A logic thus defined permits the formal expression of generic properties on evolving biological sequences and their eventual automated verification.

### C. Model Checking as an Inference Framework

The operating principle behind model checking is simple: the execution of verification software in a computer to check the correctness of a system, given a model of the system and a specification of its requirements, both provided by the user. In the event of failure to comply with the specification, the software outputs the scenarios which infringe the property as counterexamples. Nevertheless, the base algorithms alone cannot check the state space explosion, for which symbolic manipulation of systems and formulae are required [10].

Most importantly, the use of model checking techniques is completely transparent to the system under verification,

as they are domain- and interpretation-independent. This means that phylogeneticists can shift their efforts from implementation issues to modelling (before model checking, establishing required or desirable properties) and result analysis (after model checking, observing the success or failure of the process and ascertaining its significance).

### III. MODELLING OF STRUCTURAL PROPERTIES

This section is devoted to the illustration of a methodology to model biological properties of phylogenies employing to this end the logical framework introduced. By abstraction, a non-trivial property can be broken down into simpler, meaningful ones, and synthesised from these. The benefits of such logical decompositions are twofold: firstly, it simplifies formalization and favours readability; secondly, modular properties can be reused in complex constructs, and variations on a formula can be produced by local adjustments.

Two classes of queries can be identified: *global* or *tree properties*, in which the structure of the phylogeny itself is placed under scrutiny and relations between taxa inspected; and *local* or *sequence properties*, where compositional sequence features take center stage, possibly aided by placement constraints. In the remainder we furnish relevant properties of both groups and exemplify their formal modelling.

#### A. Tree Properties

Many typical tree properties are of a cladistic nature. One of the most frequent basic queries asks whether a set of extant organisms  $S$  under study constitutes a *monophyletic group* or *clade* in a phylogeny. That is to say, does it contain a subtree that has exactly those organisms as its leaves? Formally, it looks for a node somewhere in the tree ( $\mathbf{EF}$ ) which functions as the root of the group, and thus: a) everything that has to be in, is in; and b) everything that has to be out, is out (there are no outsiders, or conversely, only insiders can be found inside).

$$clade(S) \equiv \mathbf{EF}(in(S) \wedge out(S)) \quad (3)$$

The inclusion rule states that for each sequence of the set  $S$  (and-logic  $\wedge$ ) there is a path that finds it as its leaf; the exclusion rule demands that all paths end in a leaf from the same set. Within infinite computations, leaves are found through the pattern  $\mathbf{F} \circ \mathbf{AG}(s)$ , since by construction whole uniform computation subtrees can only be found precisely as a consequence of leaf states.

$$in(S) \equiv \bigwedge_{s \in S} \mathbf{EF} \circ \mathbf{AG}(s) \quad (4)$$

$$out(S) \equiv \mathbf{AF} \circ \mathbf{AG} \left( \bigvee_{s \in S} s \right) \quad (5)$$

As noted, individual properties in isolation have useful semantics in their own right. Here,  $in(S)$  is satisfied by all containing clades, and  $out(S)$  by all strict subclades.

Moreover, if the clade property is extended to both ancestral and leaf sequences, the structure of the formula remains unaltered, and only the inclusion and exclusion rules need to be tuned as follows, so that target sequences must be found anywhere in the subtree, and the whole subtree is free from intruders, respectively.

$$in'(S) \equiv \bigwedge_{s \in S} \mathbf{EF}(s) \quad (6)$$

$$out'(S) \equiv \mathbf{AG} \left( \bigvee_{s \in S} s \right) \quad (7)$$

A more challenging question is whether, given a phylogenetic tree and a partition of its leaves (which arises from the application of a sequence classification scheme), that partition constitutes a *haplogroup classification* in the tree. Haplogroups are aggregations of related haplotypes identifiable by characteristic polymorphisms; thus, they define genetic populations and usually mark these geographically as well. Their study is focused on the non-recombining regions of the genome, in particular mitochondrial DNA, where the original cladistic notation for haplogroups originated [11], and the majority of the Y chromosome, for which this notation has been adapted [12].

Essentially, a haplogroup together with the set of populations (child haplogroups) that have sprung from it over time must form a clade. In other words, a haplogroup is a *nested clade*: its members occupy all the leaves of a subtree, except possibly a number of sub-subtrees, which are completely devoid from members (and have a nested clade structure, themselves). A phylogeny is an acceptable classification if every part has a haplogroup-like structure.

$$classifier(S_1, S_2, \dots, S_h) \equiv \bigwedge_{i=1}^h haplogroup(S_i) \quad (8)$$

Checking is trivial if the relations between parent and child haplogroups are known, symbolically by means not of a formula, but of a function  $children(S)$ .

$$haplogroup'(S) \equiv clade(S \cup children(S)) \quad (9)$$

However, it is often interesting to allow for flexibility in haplogroup placement in the ongoing study and refinement of coarse haplogroups and the exploration of alternative hypotheses. Whereas (9) may be extended to determine if a suitable set of child haplogroups exists, it suffices to check the local haplogroup-like quality of each individual part without resort to any additional information beyond the composition of the target part.

Formally, haplogroup-likeness is a relaxation of  $clade(S)$  through its local structure. While the inclusion rule is preserved, as is the general search for the root of the haplogroup, the exclusion rule is replaced by the nested clade property. Under this, all paths eventually reach a point

(**AU**) where they either arrive at a member subclade or at the root of a foreign clade, always through ancestral nodes ascribed to the haplogroup in question (for this, a membership function  $h_i$  must be provided for each  $S_i$ ).

$$\text{haplogroup}(S, h) \equiv \mathbf{EF}(\text{in}(S) \wedge \text{nested}(S, h)) \quad (10)$$

$$\text{nested}(S, h) \equiv \{h\} \mathbf{AU} \{\text{out}(S) \vee \text{nesting}(S)\} \quad (11)$$

$$\text{nesting}(S) \equiv \mathbf{AF} \circ \mathbf{AG} \left( \neg \bigvee_{s \in S} s \right) \quad (12)$$

Note  $\text{nesting}(S)$  is the opposite of  $\text{out}(S)$ . Obviously, terminal haplogroups (i.e., clades) are accepted by the formula. In cladistic terms, local haplogroup-like structure corresponds to the concept of *polyphyletic group*. It is precisely because the latter are indistinguishable from *paraphyletic groups* based on leaf content alone that ancestral membership information becomes necessary (see Sec. III-B).

Nevertheless, the assumption that ancestral members of a haplogroup satisfy its defining properties is certainly reasonable. As before, the incorporation of ancestral taxa derives a related family of properties which allow a more comprehensive evaluation of the process of evolution.

### B. Sequence Properties

In general, sequence properties are based on state formulae (i.e., those that are evaluated within a node and without resort to temporal operators) in composition with simple temporal patterns, typically to extend their application to an entire phylogeny or to inspect their vicinities. Such types of state formulae will be called *patterns* ( $p$ ). They offer a powerful descriptive formalism for formulating general restrictions without the limitations of ad hoc approaches. Often, these properties may be used not necessarily to forbid patterns, but as queries and alerts to signal unusual, possibly anomalous behaviour, and mark it for further study.

A first group of patterns represent global correctness constraints that are supposed to hold across the whole phylogeny. They can be categorised as follows:

- *Conservation* is modelled as a restriction on the symbols that can occur at a given position in a sequence. Commonly, the pattern will codify a unidimensional Boolean table that classifies each symbol as permissible or impermissible. However, it is possible to define general families of compatible elements, not necessarily bound to specific positions, and restrict usage to exactly one of these, among other extensions.
- *Covariation* imposes a relation of dependence between two (or more) positions in a sequence. In its general form, it can be represented as a bidimensional Boolean table which states, for each symbol in the first column, the set of symbols that may appear in the second column. Typically, for the property to be meaningful, associations between symbols will be sparse.

- A combination of both.

A global pattern thus defined is easily verified by extending it over the computation tree.

$$\text{global}(p) \equiv \mathbf{AG}(p) \quad (13)$$

Exceptions to the aforementioned properties may in fact indicate suspicious or potentially deleterious mutations, which are of great interest in applied phylogenetic studies [13]. Furthermore, known or suspected mutations of this kind can be explicitly modelled as patterns and their positioning in a phylogeny assessed. In particular, mutations affecting important metabolic functions are expected to prevent or hinder the reproduction of the organism, and consequently should be found confined in or near terminal leaves.

Just as some mutations may be ordinarily forbidden altogether as global patterns, observed and feasible deleterious mutations may be permitted subject to certain restrictions. Specifically, it may be demanded that, if a hazardous pattern appears, it has no offspring, i.e., it is a leaf in the phylogeny; or, to provide some flexibility, it may be allowed that all descendants, if any, are reached in at most  $k$  steps ( $\mathbf{AX}^k$ ).

$$\text{terminal}(p) \equiv \mathbf{AG}(p \rightarrow \text{leaf}) \quad (14)$$

$$\text{terminal}(p, k) \equiv \mathbf{AG}(p \rightarrow \mathbf{AX}^k(\text{leaf})) \quad (15)$$

In this case, leaves (self-loops in the Kripke structure) must be detected without reference to any particular sequence. This is easily achieved by performing an equality comparison between the unique *state vectors* (the valuations of  $AP$ ) of the target state and all its successors.

$$\text{leaf} \equiv \bigwedge_{p \in AP} p \leftrightarrow \mathbf{AX}(p) \quad (16)$$

This last example is representative of properties that perform *conditional explorations* of the phylogeny. Lineage-specific verification represents a further step forward, where patterns would be used to define the sets of states of concern to a formula, though their study is beyond the scope of this paper. It is worth noting that pattern-based checking of haplogroup classifications falls in this category.

## IV. DISCUSSION

The aim of this paper has been to erect a bridge between two such apparently removed worlds as phylogenetics and formal verification. Our proposal is founded on three pillars:

- 1) The interpretation of the traditional phylogenetic tree as a transition system, whose states correspond to populations characterised by a shared genetic legacy (DNA); when members of these populations undergo change in their DNA through mutation (“evolve”), a transition between states occurs in the system. Straightforward transformations in the structure of the

tree assimilate it to the Kripke structure used by model checking techniques.

- 2) The definition of an adequate temporal logic for the expression of phylogenetic properties, the semantics of which is based on Kripke structures derived from phylogenetic trees. In the context of phylogenetics, in which evolution is represented by sequences of causal events throughout a phylogeny, temporal operators with branching-time semantics are indispensable. Thus, temporal properties inquire about forward and backward (future and past) relations in the tree.
- 3) The use of standard model checking techniques for the automated verification of phylogenetic properties expressed as temporal logic formulae on phylogenetic trees interpreted as Kripke structures. Nowadays, model checking is a mature field with a well-founded theory supported by widespread used software tools.

To our knowledge, this approach is novel in its application to phylogenetics. It presents a number of interesting advantages arising from the circumscription of the field by the robust framework of model checking, which can be summarised in the following acquired capacities:

- 1) The study of phylogenetic properties under different models of evolution. The standard phylogenetic tree has been considered as the default model throughout the paper; nevertheless, the described framework can easily consider alternative models (phylogenetic networks, etc.) that explicitly include recombination events and other complex transformations.
- 2) The understanding of relations between properties within a formal reasoning framework derived from the temporal logic of choice. Among other uses, this allows to inspect the underlying structure of a property, which may actually be composed of simpler subproperties, possibly verified (and stored) previously; from these may result a significant improvement in the performance of the model checking process.
- 3) The exploitation of verification results. Whereas satisfaction of a formula represents a simple positive result, failure to fulfil a property by model checking involves the generation of counterexamples where misbehaviour can be observed (conflicting states and the chains of events that cause them). Phylogeneticists may subsequently use these results to refine properties of interest or discover previously undetected ones.
- 4) Last, but not least, the relief from software issues in phylogenetic analysis. Because the verification algorithms are general-purpose and not subject to change, the efforts of researchers are shifted from implementation concerns to specification of properties and interpretation of results.

Notwithstanding these statements, some essential aspects are deserving of further study. Prominent among these are

extensions to the specific expressive power of the phylogenetic tree logic (which comprises everything from extensions to the set of atomic propositions to enunciation of quantitative properties on phylogenies), space- and time-efficient representations of phylogenies, and experimental studies, as well as others mentioned throughout.

#### ACKNOWLEDGMENT

This work was supported by the Spanish Ministry of Science and Innovation (MICINN) [TIN2008-06582-C03-02]; and the Spanish Ministry of Education [AP2008-03447].

#### REFERENCES

- [1] J. Felsenstein, *Inferring phylogenies*. Sunderland, MA: Sinauer, 2003.
- [2] O. Grumberg and H. Veith, Eds., *25 years of model checking: history, achievements, perspectives*. Berlin: Springer, 2008.
- [3] P. T. Monteiro, D. Ropers, R. Mateescu, A. T. Freitas, and H. de Jong, "Temporal logic patterns for querying dynamic models of cellular interaction networks," *Bioinformatics*, vol. 24, pp. i227–i233, 15 Aug. 2008.
- [4] A. Rizk, G. Batt, F. Fages, and S. Soliman, "On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology," in *Proc. CMSB 2008*, Rostock, Oct. 2008, pp. 251–268.
- [5] C. J. Langmead and S. K. Jha, "Predicting protein folding kinetics via temporal logic model checking," in *Proc. WABI 2007*, Philadelphia, PA, Sep. 2007, pp. 252–264.
- [6] C. Baier and J.-P. Katoen, *Principles of model checking*. Cambridge, MA: The MIT Press, 2008.
- [7] A. Arnold, *Systèmes de transitions finis et sémantique des processus communicants*. Paris: Masson, 1992.
- [8] Z. Manna and A. Pnueli, *The temporal logic of reactive and concurrent systems: specification*. Berlin: Springer, 1991.
- [9] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching time temporal logic," in *Proc. Workshop on Logics of Programs*, Yorktown Heights, NY, May 1981, pp. 52–71.
- [10] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE T. Comput.*, vol. C-35, pp. 677–691, Aug. 1986.
- [11] M. B. Richards, V. A. Macaulay, H.-J. Bandelt, and B. C. Sykes, "Phylogeography of mitochondrial DNA in western Europe," *Ann. Hum. Genet.*, vol. 62, pp. 241–260, May 1998.
- [12] The Y Chromosome Consortium, "A nomenclature system for the tree of human Y-chromosomal binary haplogroups," *Genome Res.*, vol. 12, pp. 339–348, Feb. 2002.
- [13] J. Montoya, E. López-Gallardo, C. Díez-Sánchez, M. J. López-Pérez, and E. Ruiz-Pesini, "20 years of human mtDNA pathologic point mutations: carefully reading the pathogenicity criteria," *Biochim. Biophys. Acta*, vol. 1787, pp. 476–483, May 2009.