ALF—A Simulation Framework for Genome Evolution

Daniel A. Dalquen,*[,1,2] Maria Anisimova,[1,2] Gaston H. Gonnet,[1,2] and Christophe Dessimoz[1,2]

[1]Computational Biochemistry Research Group, Department of Computer Science, ETH Zurich, Universitätstrasse 6, Zürich, Switzerland
[2]Swiss Institute of Bioinformatics, Universitätstrasse 6, Zürich, Switzerland

***Corresponding author:** E-mail: ddalquen@inf.ethz.ch.

Associate editor: Ryan Hernandez

Abstract

In computational evolutionary biology, verification and benchmarking is a challenging task because the evolutionary history of studied biological entities is usually not known. Computer programs for simulating sequence evolution in silico have shown to be viable test beds for the verification of newly developed methods and to compare different algorithms. However, current simulation packages tend to focus either on gene-level aspects of genome evolution such as character substitutions and insertions and deletions (indels) or on genome-level aspects such as genome rearrangement and speciation events. Here, we introduce Artificial Life Framework (ALF), which aims at simulating the entire range of evolutionary forces that act on genomes: nucleotide, codon, or amino acid substitution (under simple or mixture models), indels, GC-content amelioration, gene duplication, gene loss, gene fusion, gene fission, genome rearrangement, lateral gene transfer (LGT), or speciation. The other distinctive feature of ALF is its user-friendly yet powerful web interface. We illustrate the utility of ALF with two possible applications: 1) we reanalyze data from a study of selection after globin gene duplication and test the statistical significance of the original conclusions and 2) we demonstrate that LGT can dramatically decrease the accuracy of two well-established orthology inference methods. ALF is available as a stand-alone application or via a web interface at http://www.cbrg.ethz.ch/alf.

Key words: simulation, genome evolution, codon models, indel, lateral gene transfer, GC-content amelioration.

Introduction

To unravel evolutionary relations among single molecular characters, genes, genomes, and species, computational evolutionary biology methods typically infer past events from current data. Because of the inherently unknown nature of these past events, model and method validation and comparison is notoriously difficult. Computer programs that simulate evolution in silico provide viable test beds to understand and characterize new models and methods. Since simulations rely on simplifying models, they necessarily lack realism. Nevertheless, they provide a way—often the only way—to investigate and test evolutionary models, algorithms, and implementations under controlled conditions. Thus, validation by simulation is often considered an insufficient but necessary step to propose a new method.

Programs to simulate biological sequences can be divided into two main categories. In population genetics, simulation takes into account changes within and across populations of individuals that arise by models of sex, recombination, or gene conversion. Simulation is performed backward in time under the coalescent (e.g., Hudson 2002; Spencer and Coop 2004; Schaffner et al. 2005) or forward in time (e.g., Peng and Kimmel 2005; Hoggart et al. 2007; Chadeau-Hyam et al. 2008; Hernandez 2008; O'Fallon 2010; Peng and Liu 2010). In phylogenetics and evolutionary biology, simulation involves single representatives of species related by a tree and is performed forward in time. In this latter context, various simulation programs have been developed for different evolutionary models. Most of them simulate evolution at the gene or protein sequence level, as opposed to the genome level. Darwin (Gonnet et al. 2000) offers functions for mutating sequences along a branch, including gaps. PAML evolver (Yang 1997), Seq-Gen (Rambaut and Grassly 1997), and its extensions PSeq-Gen (Grassly et al. 1997) and CS-PSeq-Gen (Tufféry 2002) were among the first popular programs that allowed synthetic evolution of a DNA (or protein) sequence along a given tree. They include support for several models of nucleotide substitution as well as site-specific rate heterogeneity. None of these programs support insertions and deletions (indels). Only very recently indel simulation was included alongside the point mutation process and additional sequence features such as variable over time mutation rates or sequence motifs: EvolveAGene (Hall 2008), MySSP (Rosenberg 2005), Hetero (Jermiin et al. 2003), indel-Seq-Gen (Strope et al. 2007), Rose (Stoye et al. 1998). Another program, SISSI (Gesell and Haeseler 2006), simulates site-specific interactions. Although most programs typically use biologically very simple indel simulation, some programs also incorporate more advanced models for indel formation and distribution: SIMPROT (Pang et al. 2005), Dawg (Cartwright 2005), and INDELible (Fletcher and Yang 2009). Additionally, PhyloSim (Sipos et al. 2011) simulates complex rate variations and selective constraints with multiple substitution and indel processes. To our knowledge, EvolSimulator (Beiko and Charlebois 2007) is the only program to go beyond single sequence simulation and allowing genomic effects such as gene duplication or lateral gene transfer (LGT). This program, however, is limited in the choice of evolutionary models and lacks support for insertions and deletions at the sequence level.

Here, we introduce Artificial Life Framework (ALF), which we developed with the long-term goal of simulating the entire range of evolutionary forces that act on genomes. In this first release, we primarily focus on species-level evolution. ALF evolves an ancestral genome, represented by an ordered set of sequences, along a tree into a number of descendant synthetic genomes. At the level of a gene, ALF can simulate evolution at the nucleotide, codon, or amino acid level with indels and among-site rate heterogeneity and supports most established models of character substitution. To mimic different types of sequences (e.g., coding sequence vs. noncoding, functional genes vs. pseudogenes, etc.), multiple sequence classes can be defined, each with their own models of substitution, insertion–deletion, and among-site rate variation. At a more global genomic level, ALF can simulate GC-content amelioration (Lawrence and Ochman 1997), gene duplication and loss, genome rearrangements, several types of LGT, and speciation events. The user can provide a starting (ancestral) genome or have one generated randomly. The output consists of the simulated genomes, multiple sequence alignments, and gene trees of all gene families, all ancestral sequences, the true species tree including LGTs, and for each pair of genomes the sets of orthologous, paralogous, and xenologous sequences. In future releases of ALF, we aim to incorporate more evolutionary models, including population-level events such as recombination.

This article is organized as follows. We first provide an overview of ALFs architecture and briefly describe how to set up a simulation scenario via ALFs web interface. We then summarize various control experiments conducted to ensure ALFs correctness. Finally, we present applications of ALF to two bioinformatics problems: First, we reanalyze data from a study of selection after globin gene duplication and test the statistical significance of the original conclusions; second, we demonstrate that LGT can dramatically decrease the accuracy of two well-established orthology inference methods.

Methods and Materials

Overview of the Simulation Process

ALF generates a set of species genomes starting from a single ancestral genome sequence. The ancestral genome may be represented by biological sequences supplied by the user or generated randomly according to user specifications. A species tree may also be specified by the user or randomly generated. In the course of the simulation, ALF evolves the root genome along the tree, where each node defines a speciation event. The emerging genomes are exposed to the evolutionary processes implemented in ALF.

Figure 1 gives a graphical overview of the ALF simulation pipeline. Character substitutions occur according to the substitution probability matrix of a selected amino acid, codon, or nucleotide model for a given branch length. Different models can be specified for simulation, for example, one codon and one nucleotide model could be used to distinguish coding and noncoding regions, respectively. The rate of substitution can differ over sites and genes. ALF allows for each species to have its own underlying equilibrium base frequencies, for instance, to simulate drift toward species-specific GC content.

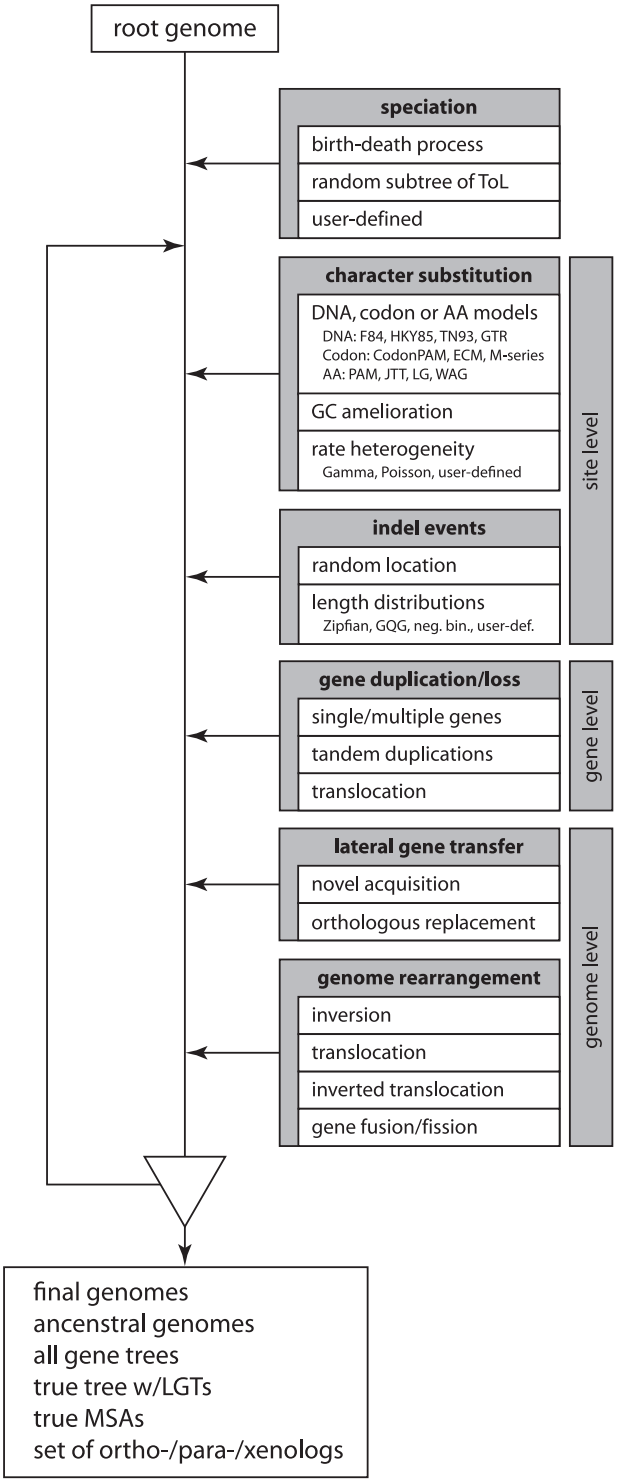

FIG. 1. Overview of the ALF simulation process. A root genome is evolved along a species tree. Events at the site, sequence and genome level are simulated iteratively.

The simulation of other evolutionary processes is based on Gillespie's algorithm with exponential waiting times (Gillespie 1977), providing for realistic scenarios with parallel simulations of events at the sequence and genome

level. Indel events occur additionally and independently of substitutions with separate rates for indels. Indels that would disrupt protein function by introducing frame shifts are not allowed when simulating coding DNA. In that case, ALF only simulates in-frame indels of nucleotide triplets or codons, and insertions will not contain any stop codons. Gene duplications, gene losses, and LGT alter the content of each genome. All three types of events can affect single or multiple genes. Genes in a genome can also be affected by gene fusion events that join neighboring genes into one and fission events that break an existing gene into two. Finally, rearrangement events lead to the shuffling of the genes within a genome.

The simulation starts by reading or generating a root genome and a species tree that defines speciation times. Then, gene- and genome-level events are generated on each branch by drawing a random waiting time from an exponential distribution using the total rate of all events. An event to occur after that time interval is selected based on its relative rate. Sequence level events (substitutions and indels) are delayed until a gene/genome level event depends on their execution or until the end of each branch. First, substitutions are performed using a substitution probability matrix. Afterward, indels are generated in a similar manner as gene/genome level events.

ALF is highly configurable, allowing the simulation of either all or an arbitrary subset of evolutionary events. Size and number of the resulting organisms is only limited by computational power. For example, evolving 20 genomes based on the coding sequences of *Escherichia coli* (4,352 sequences with a total length of 1,368,902 codons), using the empirical codon model by Schneider et al. (2005) takes about 4.5 h on an single Intel Xeon core at 2.26 GHz.

Evolutionary Events

Speciation

Speciation events in ALF occur according to a species tree, fixed prior to simulation. ALF offers three options for obtaining this tree: 1) A tree is sampled according to the birth–death process with parameters λ and μ as described by Gernhard (2008); because the resulting trees are ultrametric, an exponentially distributed deviation is applied to each branch according to Guindon and Gascuel (2003), 2) a tree can be obtained by sampling uniformly from a variance weighted least squares tree on 1,038 species based on data from the OMA project (Dessimoz and Gil 2008), and 3) A custom tree (e.g., a priori inferred) may be specified by the user in Newick or DARWIN format.

At a speciation event, the two new species inherit the ancestral genome while the genome-specific parameters such as target GC content are adapted. As the simulation progresses, the genomes of the two species evolve independently and start undergoing different mutations and accumulating differences.

Sequence Types

For each segment of the root genome, a sequence type can be specified, which is defined by a substitution model, an indel model, and a model for rate heterogeneity among sites as described below. Switches between types can occur at gene duplication and speciation events, as specified by the user. Sequence types allow for simultaneous simulation of sequences with different characteristics, for example, for coding and noncoding sequences.

Substitution

The user can choose to simulate nucleotide, codon, and/or amino acid substitution. When simulating codon substitution, mutations that lead to the formation of a stop codon (nonsense mutations) are ignored. Although stop codons may occur in nature, in particular near the end of a sequence, they usually have a deleterious effect on protein function and will be selected against.

Nucleotide Substitution Models. Nucleotide substitution is simulated using one of four well-known Markov models. The HKY (Hasegawa et al. 1985) and F84 (Felsenstein and Churchill 1996) models allow for a different rate of transitions and transversions as well as unequal base frequencies. The TN93 model (Tamura and Nei 1993) is more general in that it models transitions with two parameters. Other simpler models, such as JC and K80, can be viewed as special cases of TN93. Finally, model GTR (Tavaré 1986) is the most general time reversible. In addition to the equilibrium base frequencies, it allows for six different rate parameters describing each type of substitution.

Codon Substitution Models. Codon models recently came into prominence, but very few programs allow simulation of the codon substitution process (for a review, see Anisimova and Kosiol 2009). ALF enables the simulation of protein-coding sequences under a range of codon models: the parametric site models M0, M2, M3, and M8 with variable selection pressure over sites (Yang et al. 2000), empirical codon models derived by Schneider et al. (2005) and Kosiol et al. (2007). Alternatively, a user-specified matrix can be used.

Amino Acid Substitution Models. For simulations at the amino acid level, ALF offers seven substitution models. These include PAM (Dayhoff et al. 1978), Gonnet (Gonnet et al. 1992), JTT (Jones et al. 1992), WAG (Whelan and Goldman 2001), LG (Le and Gascuel 2008), as well as two models for ordered and disordered proteins (Szalkowski and Anisimova 2011). The user can also choose to provide a custom exchangeability matrix and frequencies.

Rate Heterogeneity, Domains, and Motifs. Not all genes in an organism mutate at the same rate. Likewise, within genes, different regions or sites may evolve faster or slower—for example, transmembrane regions or active sites are known to evolve fast or slow, respectively. ALF acknowledges this fact in two ways. On the genome level, the overall rate of each gene can be modified by a factor drawn from the Gamma distribution (Γ) with parameter α_g. This factor also affects the indel rates. On the sequence level, ALF supports the Gamma model with invariable sites ($\Gamma + I$) (Gu et al. 1995). Alternatively, the sequence can be divided into a number of domains, all having their own mutation rate. The number

of domains for a gene is chosen at random from a uniform distribution between 1 and a user-defined maximal value, and the mutation rate for each domain is drawn from the Poisson distribution with user-specified mean.

When real sequence data are used for the root genome, the user has the possibility to provide custom rates. This tailors for simulating the evolution of well-characterized proteins or can be used for testing the robustness of estimates from empirical studies. Since domains can be as small as single amino acids and the mutation rate can be set to zero, it is possible to construct strictly conserved motifs.

GC-Content Amelioration. When ALF is used to simulate evolution at the nucleotide or codon level, differences in GC content can be simulated in two ways. One possibility is to switch to different substitution models specified by the user at the internal nodes of the species tree. The other possibility is to assign target nucleotide frequencies to the genomes at the leaves of the species tree, either globally for all substitution models or for each substitution model separately. During the simulation, these frequencies are used to compute the mutation matrix and to create sequence fragments for insertions. As a consequence, GC content converges over time to the target value defined for the leaves of the tree. The actual nucleotide frequencies follow the formula $\pi_t = \pi_0 \cdot e^{Qt}$, where π_0 and π_t are the base frequencies at the beginning and end of the branch, respectively, and the target frequencies are used in the rate matrix Q (Yang and Roberts 1995). The target frequencies for the internal nodes up to the root are computed as averages weighted by the lengths of the outgoing branches. All genes within an organism (using the same substitution model) share the same GC content. However, LGT may work against GC amelioration, keeping the GC content different for some genes.

Target frequencies can be set globally or per substitution model, and the user has the choice of having target frequencies generated randomly or supplying his own.

Indel Formation

Length distributions and rates can be specified separately for indels. Several possibilities for modeling the indel length distribution have been implemented. The first model uses the negative binomial distribution, which takes two parameters, an integer (r) and a proportion (q). By setting $r = 1$, the distribution is geometric and equivalent to the affine scoring model with gap open and gap extension costs (Gotoh 1982).

The second method models indel lengths using the Zipfian distribution with one parameter (Benner et al. 1993; Chang and Benner 2004).

Another available model is the generalized Qian–Goldstein indel length distribution (Qian and Goldstein 2001), which uses a mixture of exponentials and adapts to the branch lengths. Finally, the user can specify a customized general discrete distribution.

In order to cut the tail of the chosen indel length distribution, a maximum for the length of an indel can be defined at which the distribution is truncated. For insertions, the content of the inserted segment is drawn from the equilibrium character distribution for the species.

ALF offers two ways of placing deletions within the sequence: 1) Following Cartwright's model in Dawg (Cartwright 2005), each sequence is considered to be embedded in a longer virtual sequence. Deletions affecting the simulated sequence can start and/or end in this virtual sequence. A deletion of length L_d in a sequence of length L_s can therefore start at any position in the interval $[-L_d + 1, L_s]$. This method ensures equal deletion probability for all sites but biases the deletion length distribution toward smaller deletions because gaps overlapping with the beginning or end of sequence get truncated. 2) Deletions are placed within the simulated sequence in their entirety. Given a deletion length L_d and sequence length L_s, the deletion can start at any position in the interval $[1, L_s - L_p + 1]$. This way, the length of deletions matches more closely the distribution specified, but with this strategy, the deletion rate is not uniform over the entire sequence: the deletion probability for sites close to the ends of the sequence is lower than in the middle.

Gene Duplication and Gene Loss

Gene duplication and gene loss events occur randomly and in parallel to the evolutionary events at the sequence level. They can comprise one or several consecutive randomly selected sequences up to a user-defined maximum. The new copies from a duplication event are inserted as new sequences either directly after their originals or at a random position in the genome. Furthermore, ALF accounts for neofunctionalization (Ohno 1970) and subfunctionalization (Lynch and Conery 2000) by temporarily altering the mutation rate of duplicates or both, originals and duplicates, by a user-defined amount. In the case of neofunctionalization, this behavior corresponds to the idea that an organism can afford to mutate a copied gene at a higher rate while still maintaining the original function. The concept of subfunctionalization assumes both copies of a duplicated gene to be freed of evolutionary constraints and evolve in a complementary fashion.

Similarly, one or more genes are removed from the genome in a gene loss event.

Lateral Gene Transfer

Apart from gene duplication, LGT is the second evolutionary process allowing a genome to acquire new genes. ALF implements two kinds of LGT: orthologous replacement and "novel acquisition" (Doolittle et al. 2003). In the first case, the newly acquired gene replaces an orthologous gene in the recipient. In the second case, the new gene is added to the recipient genome without any replacement. For "novel acquisition," not only a single gene but a whole group of genes can be transferred.

The donor and recipient genomes as well as the genes to be transferred are chosen at random. In the case of "novel acquisition," the transferred genes are inserted at a random position in the recipient's genome. With orthologous replacement, only genes that exist in both genomes can be

transferred, and the transferred gene replaces its ortholog in the recipient.

Gene Fusion and Fission

ALF supports gene fission events that break a gene into two new sequences, either at a random location or at domain borders. Fusion events merge two or more existing genes into one new sequence.

Genome Rearrangement

In nature, the gene positions within a genome are not fixed. Several mechanisms may cause gene translocations or changes in the direction of the coding strand (Sankoff and Nadeau 2003). In ALF, these genome rearrangements are modeled as two independent phenomena.

First, genes can be moved to a different position within the genome. Such translocations occur at a user-defined rate and move a random number of consecutive genes to a random position within the genome.

The second process is the inversion of a segment of the genome, which results from the replacement of a segment of DNA by its reverse complement. Translocations and inversions can also occur simultaneously, leading to so-called inverted translocations.

Program Output

Although the probabilities of all evolutionary processes can be adjusted, all events occur at random positions at random points in time. Running ALF multiple times with the same parameters will therefore lead to generating different genomes with different histories in each run.

ALF saves the genomes of all species that arise during the simulation, including the ancestors of the final species, as a DARWIN (Gonnet et al. 2000) database containing all protein and DNA sequences (for simulations at the codon or nucleotide level) and their evolutionary history. Additionally, the sequences are saved in the FASTA format. Furthermore, true alignments and evolutionary histories (including all LGTs) are recorded during the simulation process for all gene families. A set of all orthologous genes of a gene family can also be assembled.

Web Interface and Stand-Alone Version

A web interface for ALF is available at http://www.cbrg.ethz.ch/alf. It provides the user with an intuitive way for setting simulation parameters and is organized hierarchically, reflecting the level upon which each force acts (fig. 1). To make usage of ALF as simple as possible, the user can rely on contextual help for all parameters as well as a selection of presets for typical applications (including those outlined below). The web interface can be used either to prepare a configuration file for the stand-alone version of ALF or to run the simulation directly on our servers.

For extensive simulations, we recommend using the stand-alone version of ALF that is available for Linux and Mac OS X and can be downloaded freely from the same web address.

Validation

To validate our simulation framework, we ran separate simulations to test the various processes and models of the framework:

- We ascertained that basic properties of the simulation were not violated. For example, without the simulation of gene loss, sequences should not disappear from the genomes. Similarly, the sequence lengths should not change when no indels are simulated.
- We ensured that evolutionary distances between pairs of resulting sequences matched the distances from the input tree when estimated using the same model (supplementary fig. 5, Supplementary Material online).
- For parametric models, we used codeml (Yang 1997) to reestimate the parameters of the substitution rate matrices used for the simulation. In all cases, the parameter estimates were close to the true values (supplementary table 2, Supplementary Material online for codon models and supplementary table 3, Supplementary Material online for nucleotide models).
- We compared the distribution of indel length specified in the simulation to the resulting distribution of gap lengths in the true alignment between gene pairs, using a χ^2-test for goodness of fit. As long as there were no or few overlapping indel events, no significant differences were observed. As the proportion of overlapping indels increases, the length distribution of gaps becomes biased toward longer gaps (supplementary figs. 6 and 7, Supplementary Material online)—an expected behavior because overlapping indels appear as a single gap.
- In simulations with GC-content amelioration, we confirmed that, when the branch lengths are sufficiently long, the base frequencies converge to the specified target frequencies, both for nucleotide and codon models.
- We ensured correctness of the Gillespie algorithm by comparing the distribution of waiting times between events to the theoretical exponential distribution using a χ^2-test for goodness of fit (supplementary fig. 8, Supplementary Material online).

Results and Discussion

In order to illustrate the utility of ALF, we discuss two example studies below. First, we investigate the accuracy of a clade codon model (Bielawski and Yang 2004) by looking at the empirical distribution of the parameters. In the second study, we analyze how LGT affects prediction accuracy of two orthology inference methods.

Testing Selection Regimes in Globin Gene Family

In vertebrates and some other species, oxygen is transported from lungs to tissues by means of binding with hemoglobin. In most vertebrates, hemoglobin is a tetramer of two pairs of subunits designated α and β. In placental mammals, two

Table 1. ML Estimates of Model Parameters for the Globin Data Set and the "Globin-Like" Simulated Data.

Data Set	Model	Parameters Estimates	*l*
Globin data set (real data)	**M0**	$\omega = 0.191$	−2477.82
	M3	$\omega_0 = 0, p_0 = 0.134$ $\omega_1 = 0.072, p_1 = 0.604$ $\omega_2 = 0.607, p_2 = 0.262$	−2442.4
	MD	$\omega_0 = 0.053, p_0 = 0.716$ $\omega_1 = 0.649, p_1 = 0.154$ $\omega_{2\epsilon} = 0.045, \omega_{2\gamma} = 0.991, p_2 = 0.130$	−2435.65
ALF simulation (100 replicates)	**M0**	$\bar{\omega} = 0.202$	
	M3	$\bar{\omega}_0 = 0.036, \bar{p}_0 = 0.426$ $\bar{\omega}_1 = 0.229, \bar{p}_1 = 0.374$ $\bar{\omega}_2 = 0.761, \bar{p}_2 = 0.200$	
	MD	$\bar{\omega}_0 = 0.060, \bar{p}_0 = 0.673$ $\bar{\omega}_1 = 0.640, \bar{p}_1 = 0.188$ $\bar{\omega}_{2\epsilon} = 0.191, \bar{\omega}_{2\gamma} = 1.150, \bar{p}_2 = 0.139$	

NOTE.—ω: selective pressure (d*N*/d*S* ratio), p_i: proportions of ω classes. For real data, log likelihoods are shown. MD is the preferred model according to the likelihood ratio test, Akaike information criterion (AIC) and Bayesian information criterion (BIC).

paralogs (ϵ and γ) are expressed during early development instead of β. The persistence of these two genes in the genome since the duplication event about 80–100 Ma is believed to be an example of genetic cooption, that is, the shift of a trait or gene to a new function.

Selective pressures after gene duplication in the globin gene family has been studied using the codon model D (MD), and a Bayesian approach was used to detect sites that have experienced changes in selective pressure following the genetic cooption event (Bielawski and Yang 2004). Instead of assuming the same parameters for the whole tree, MD allows for different selective pressure in two clades of the gene tree following the duplication event. The selective pressure on the protein is measured by the dN/dS ratio (ω), which reflects negative selection if $\omega < 1$ and positive selection if $\omega > 1$. Among-site variation of selective pressure is modeled using k discrete site classes, one of which allows different ω parameters in the two a priori specified clades. Based on their analyses, the authors suggested that a fraction of sites evolves with different selective pressures in the two clades, with strict negative selection affecting clade ϵ and relaxed purifying selection on clade γ. The complexity of their model, combined with the short sequence length of the genes in question, makes it difficult to assess the confidence in their estimate, in particular, whether the differing selection class for the γ globin clade indeed corresponds to mild purifying, neutral, or even positive selection. Our reanalysis of the data (a list of genes is included in the supplementary table 1, Supplementary Material online) resulted in slightly different values, with $\omega_{2\gamma}$ being very close to 1, which we attribute to the removal of columns with gaps. The maximum likelihood (ML) estimates of the model parameters are summarized in table 1.

To assess the accuracy of MD, we used ALF to simulate three runs of 100 "globin-like" data sets under MD ($k = 3$), fixing $\omega_{2\gamma} = 1$ and keeping all other parameters at their ML estimates for the original data set. We then reestimated the model parameters for each replicate using codeml. To avoid local optima, we ran codeml three times using different starting values and chose the result with the highest likelihood. Finally, we compared the ML estimate of $\omega_{2\gamma}$ obtained from the real data to the distribution of ML estimates obtained from simulated replicates with $\omega_{2\gamma} = 1$. As can be seen from figure 2*a*, the ML estimate for $\omega_{2\gamma}$ (0.991 here or 0.79 in the original study) lies well within the variance of this parameter. Therefore, our observations indicate that neutrality for this class in clade γ cannot be rejected, and there is no evidence for relaxed purifying selection on this clade, contrary to the conclusions of Bielawski and Yang (2004).

In addition, we also simulated another 100 replicates with 10,000 codons to ensure that the mean estimate converges to the true value for long sequences. Figure 2*b* shows that this is indeed the case.

Our results therefore suggest that MD might be too complex to give reliable estimates for the short sequences from the globin gene family. For the simpler models M0 and M3, on the other hand, this did not appear to be a problem, as the variances of their parameters were much smaller on the simulated data (mean values: table 1; distributions: supplementary figures 2 and 3, Supplementary Material online).

Ortholog Prediction in Presence of LGT

LGT is widely recognized as a major force in prokaryotic genome evolution, although the extent of LGT is still disputed (Ragan and Beiko 2009). Nevertheless, orthology prediction projects only consider vertical inheritance of genes.

We used ALF to analyze the influence of LGT on the performance of two well-established programs for orthology prediction, Inparanoid 4.1 (Remm et al. 2001) and OMA (Roth et al. 2008). We simulated data sets with different amounts of gene duplications and LGT for a tree with 30 species, sampled from the tree of γ-proteobacteria (supplementary fig. 4, Supplementary Material online). The root genome of each simulation consisted of 200 randomly generated sequences using amino acid frequencies

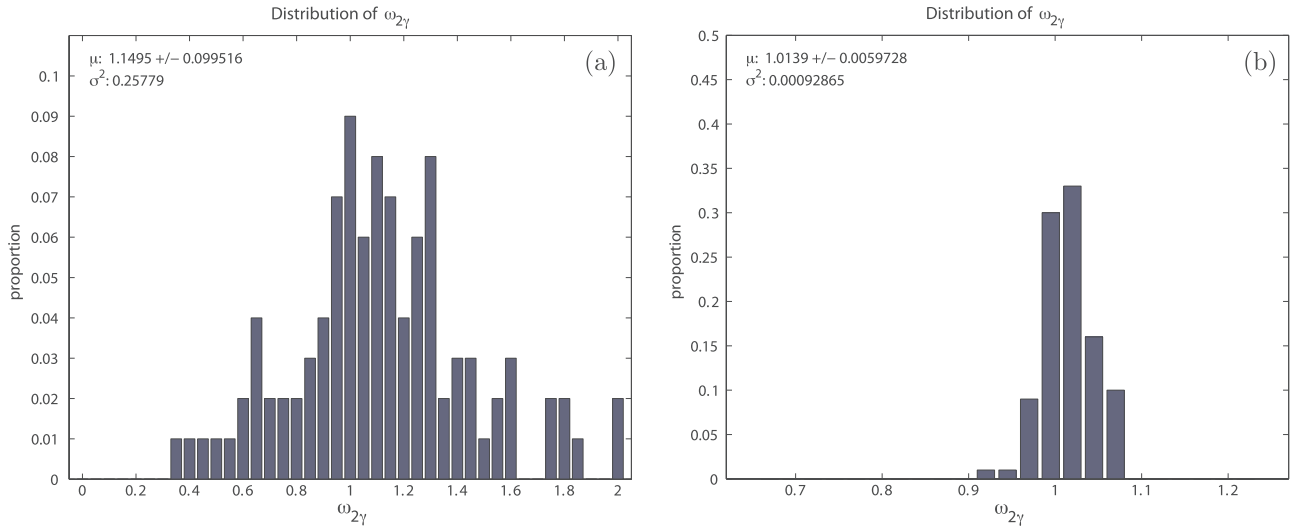

FIG. 2. The distribution of ML estimates for $\omega_{2\gamma}$ from simulation with ALF (a) for one run with sequence length matching the real data (144 codons; other data shown in supplementary fig. 1, Supplementary Material online), and (b) for sequences of 10,000 codons. Data simulated under MD with $\omega_{2\gamma} = 1$, all other parameters are as in table 1.

from the WAG model (Whelan and Goldman 2001), which was also used for substitutions. Sequence lengths followed the Γ length distribution that we fitted on data from γ-proteobacteria. A gene loss rate was chosen that kept the number of genes roughly constant. We then used the resulting synthetic genomes as input for the two prediction pipelines. To avoid differences attributable to homology inference, we used the same procedure for both OMA and Inparanoid, namely all-against-all Smith–Waterman alignment with score cutoff of 181 (roughly corresponding to an *E* value of 10^{-14}).

The results of the analysis are summarized in a precision-recall plot (fig. 3), where precision is defined as the fraction of true positives among predicted positives, and recall corresponds to the fraction of true positives recovered by a method, that is, its statistical power. For both methods, varying the LGT and gene duplication rates appear to have an almost orthogonal effect on prediction accuracy. Increasing the duplication rate mainly affects recall, decreasing the fraction of recovered true positives. When a larger fraction of genomes consists of duplicated genes, the orthology prediction task becomes more challenging because effects such as differential gene loss complicate the inference problem. Thus, consistent with previous studies of OMA and Inparanoid (Altenhoff and Dessimoz 2009; Boeckmann et al. 2011; Linard et al. 2011), we observe that these methods are relatively conservative in their predictions of difficult scenarios.

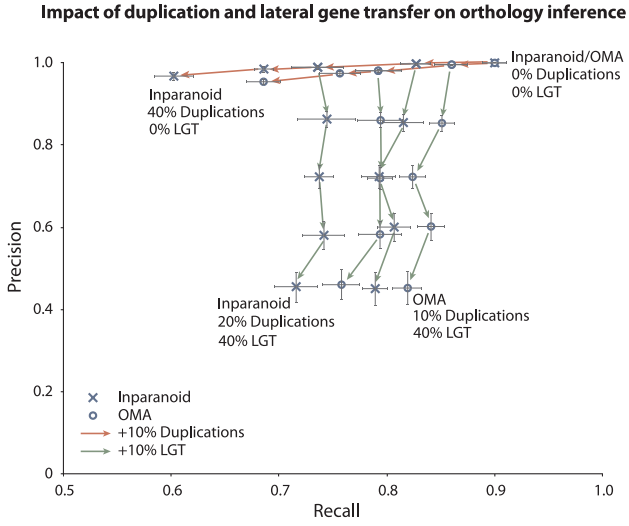

FIG. 3. Precision/recall of orthology predictions with different proportions of genes with a history of duplications and/or LGT. Each data point corresponds to the mean of five independent runs using the same parameters (with 95% confidence interval in both dimensions).

On the other hand, increasing LGT worsens precision, reducing the fraction of true positives among predicted positives. For both methods, it appears that laterally transferred genes replacing an existing sequence in the recipient species are more difficult to distinguish from true orthologs by either algorithm. Indeed, both Inparanoid and OMA have been developed under the assumption that the sequences are related through speciation and duplication events only, not lateral transfer events. This analysis confirms that ignoring lateral transfer events can lead to a significant fraction of false-positive orthology predictions.

Conclusion

The lack of knowledge of the evolutionary history is a major challenge when developing new models and methods in computational biology. Although a computer program will never be able to describe the entire evolutionary reality and might ignore potentially important factors, simulation packages have proven to be useful tools for analyzing and comparing the performance of new algorithms. In contrast to the majority of existing tools, ALF can simulate processes at the genomic level, rendering itself useful for a broad range of analyses in gene and genome evolution. With the two ex-

ample case studies, we illustrated what such analyses might look like. Other possible applications include benchmarking of alignment or tree building methods (including methods for multiple loci or based on gene rearrangement) and strategies for gene and species tree reconciliation.

Although ALF already implements more evolutionary models than any other publicly available simulation tool, the long-term goal of ALF is to realistically simulate the entire range of evolutionary forces that act on genomes. Currently, ALF is particularly well suited for simulation of prokaryotic evolution, where it covers many of the evolutionary processes typically relevant. And yet, many important aspects of evolution have yet to be implemented. As next steps, we plan to incorporate models of recombination and of promiscuous domain evolution (Basu et al. 2008). Possible further improvements could include models of interactions between sequences as well as of patterns such as codon biases or tandem repeats on the sequence level.

As new evolutionary forces are discovered, we are confident that these can be included in ALF, allowing for more realistic simulations and more thorough testing of algorithms.

Supplementary Material

Supplementary table S1 and figures S1–S4 are available at *Molecular Biology and Evolution* online (http://www.mbe.oxfordjournals.org/).

Acknowledgments

We would like to thank Daniel Margadant and Sereina Riniker who helped in the beginning of the project and Prof. Z. Yang as well as two anonymous reviewers for their helpful comments on ALF and this manuscript. The project was funded by ETH Zurich. M.A. received additional funding from the Swiss Science Foundation (ref. 31003A/127325). The open access charges were funded by the Swiss Institute of Bioinformatics. The funders had no role in study design, analysis, decision to publish, or preparation of the manuscript.

References

Altenhoff AM, Dessimoz C. 2009. Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Comput Biol*. 5:e1000262.

Anisimova M, Kosiol C. 2009. Investigating protein-coding sequence evolution with probabilistic codon substitution models. *Mol Biol Evol*. 26:255–271.

Basu MK, Carmel L, Rogozin IB, Koonin EV. 2008. Evolution of protein domain promiscuity in eukaryotes. *Genome Res*. 18:449–461.

Beiko RG, Charlebois RL. 2007. A simulation test bed for hypotheses of genome evolution. *Bioinformatics* 23:825–831.

Benner SA, Cohen MA, Gonnet GH. 1993. Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J Mol Biol*. 229:1065–1082.

Bielawski JP, Yang Z. 2004. A maximum likelihood method for detecting functional divergence at individual codon sites, with application to gene family evolution. *J Mol Evol*. 59:121–132.

Boeckmann B, Robinson-Rechavi M, Xenarios I, Dessimoz C. 2011. Conceptual framework and pilot study to benchmark phylogenomic databases based on reference gene trees. *Brief Bioinform*. 12:423–435.

Cartwright RA. 2005. DNA assembly with gaps (Dawg): simulating sequence evolution. *Bioinformatics* 21(3 Suppl):iii31–iii38.

Chadeau-Hyam M, Hoggart CJ, O'Reilly PF, Whittaker JC, De Iorio M, Balding DJ. 2008. Fregene: simulation of realistic sequence-level data in populations and ascertained samples. *BMC Bioinformatics* 9:364.

Chang MSS, Benner SA. 2004. Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *J Mol Biol*. 341:617–631.

Dayhoff MO, Schwartz RM, Orcutt BC. 1978. A model for evolutionary change in proteins. In: Dayhoff MO, editor. Atlas of protein sequence and structure. Vol. 5, Suppl. 3. Washington (DC): National Biomedical Research Foundation. p. 345–352.

Dessimoz C, Gil M. 2008. Covariance of maximum likelihood evolutionary distances between sequences aligned pairwise. *BMC Evol Biol*. 8:179.

Doolittle WF, Boucher Y, Nesbø CL, Douady CJ, Andersson JO, Roger AJ. 2003. How big is the iceberg of which organellar genes in nuclear genomes are but the tip? *Philos Trans R Soc Lond B Biol Sci*. 358:39–57; discussion 57–58.

Felsenstein J, Churchill GA. 1996. A hidden Markov model approach to variation among sites in rate of evolution. *Mol Biol Evol*. 13: 93–104.

Fletcher W, Yang Z. 2009. Indelible: a flexible simulator of biological sequence evolution. *Mol Biol Evol*. 26:1879–1888.

Gernhard T. 2008. The conditioned reconstructed process. *J Theor Biol*. 253:769–778.

Gesell T, Haeseler AV. 2006. In silico sequence evolution with site-specific interactions along phylogenetic trees. *Bioinformatics* 22:716–722.

Gillespie D. 1977. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem*. 81:2340–2361.

Gonnet GH, Cohen MA, Benner SA. 1992. Exhaustive matching of the entire protein sequence database. *Science* 256:1443–1445.

Gonnet GH, Hallett MT, Korostensky C, Bernardin L. 2000. Darwin v. 2.0: an interpreted computer language for the biosciences. *Bioinformatics* 16:101–103.

Gotoh O. 1982. An improved algorithm for matching biological sequences. *J Mol Biol*. 162:705–708.

Grassly NC, Adachi J, Rambaut A. 1997. PSeq-Gen: an application for the Monte Carlo simulation of protein sequence evolution along phylogenetic trees. *Comput Appl Biosci*. 13:559–560.

Gu X, Fu YX, Li WH. 1995. Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites. *Mol Biol Evol*. 12:546–557.

Guindon S, Gascuel O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol*. 52:696–704.

Hall BG. 2008. Simulating DNA coding sequence evolution with EvolveAGene 3. *Mol Biol Evol*. 25:688–695.

Hasegawa M, Kishino H, Yano T. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol*. 22:160–174.

Hernandez RD. 2008. A flexible forward simulator for populations subject to selection and demography. *Bioinformatics* 24: 2786–2787.

Hoggart CJ, Chadeau-Hyam M, Clark TG, Lampariello R, Whittaker JC, De Iorio M, Balding DJ. 2007. Sequence-level population simulations over large genomic regions. *Genetics* 177:1725–1731.

Hudson RR. 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18:337–338.

Jermiin LS, Ho SYW, Ababneh F, Robinson J, Larkum AWD. 2003. Hetero: a program to simulate the evolution of DNA on a four-taxon tree. *Appl Bioinformatics*. 2:159–163.

Jones DT, Taylor WR, Thornton JM. 1992. The rapid generation of mutation data matrices from protein sequences. *Comput Appl Biosci*. 8:275–282.

Kosiol C, Holmes I, Goldman N. 2007. An empirical codon model for protein sequence evolution. *Mol Biol Evol*. 24:1464–1479.

Lawrence JG, Ochman H. 1997. Amelioration of bacterial genomes: rates of change and exchange. *J Mol Evol*. 44:383–397.

Le SQ, Gascuel O. 2008. An improved general amino acid replacement matrix. *Mol Biol Evol*. 25:1307–1320.

Linard B, Thompson JD, Poch O, Lecompte O. 2011. OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics* 12:11.

Lynch M, Conery JS. 2000. The evolutionary fate and consequences of duplicate genes. *Science* 290:1151–1155.

O'Fallon B. 2010. TreesimJ: a flexible, forward time population genetic simulator. *Bioinformatics* 26:2200–2201.

Ohno S. 1970. Evolution by gene duplication. New York: Springer-Verlag.

Pang A, Smith AD, Nuin PAS, Tillier ERM. 2005. SIMPROT: using an empirically determined indel distribution in simulations of protein evolution. *BMC Bioinformatics* 6:236.

Peng B, Kimmel M. 2005. simuPOP: a forward-time population genetics simulation environment. *Bioinformatics* 21:3686–3687.

Peng B, Liu X. 2010. Simulating sequences of the human genome with rare variants. *Hum Hered*. 70:287–291.

Qian B, Goldstein RA. 2001. Distribution of indel lengths. *Proteins* 45:102–104.

Ragan MA, Beiko RG. 2009. Lateral genetic transfer: open issues. *Philos Trans R Soc Lond B Biol Sci*. 364:2241–2251.

Rambaut A, Grassly NC. 1997. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput Appl Biosci*. 13:235–238.

Remm M, Storm CE, Sonnhammer EL. 2001. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J Mol Biol*. 314:1041–1052.

Rosenberg MS. 2005. MySSP: non-stationary evolutionary sequence simulation, including indels. *Evol Bioinform Online*. 1:81–83.

Roth ACJ, Gonnet GH, Dessimoz C. 2008. Algorithm of OMA for large-scale orthology inference. *BMC Bioinformatics* 9:518.

Sankoff D, Nadeau JH. 2003. Chromosome rearrangements in evolution: from gene order to genome sequence and back. *Proc Natl Acad Sci U S A*. 100:11188–11189.

Schaffner SF, Foo C, Gabriel S, Reich D, Daly MJ, Altshuler D. 2005. Calibrating a coalescent simulation of human genome sequence variation. *Genome Res*. 15:1576–1583.

Schneider A, Cannarozzi GM, Gonnet GH. 2005. Empirical codon substitution matrix. *BMC Bioinformatics* 6:134.

Sipos B, Massingham T, Jordan GE, Goldman N. 2011. PhyloSim—Monte Carlo simulation of sequence evolution in the R statistical computing environment. *BMC Bioinformatics* 12:104.

Spencer CCA, Coop G. 2004. SelSim: a program to simulate population genetic data with natural selection and recombination. *Bioinformatics* 20:3673–3675.

Stoye J, Evers D, Meyer F. 1998. Rose: generating sequence families. *Bioinformatics* 14:157–163.

Strope CL, Scott SD, Moriyama EN. 2007. indel-Seq-Gen: a new protein family simulator incorporating domains, motifs, and indels. *Mol Biol Evol*. 24:640–649.

Szalkowski AM, Anisimova M. 2011. Markov models of amino acid substitution to study proteins with intrinsically disordered regions. *PLoS One* 6:e20488.

Tamura K, Nei M. 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol Biol Evol*. 10:512–526.

Tavaré S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lect Math Life Sci*. 17:57–86.

Tufféry P. 2002. CS-PSeq-Gen: simulating the evolution of protein sequence under constraints. *Bioinformatics* 18:1015–1016.

Whelan S, Goldman N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol Biol Evol*. 18:691–699.

Yang Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. *Comput Appl Biosci*. 13:555–556.

Yang Z, Nielsen R, Goldman N, Pedersen AM. 2000. Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics* 155:431–449.

Yang Z, Roberts D. 1995. On the use of nucleic acid sequences to infer early branchings in the tree of life. *Mol Biol Evol*. 12:451–458.