Swarming Along the Evolutionary Branches Sheds Light on Genome Rearrangement Scenarios

Nikolay Vyahhi
Information Management Research Group
St. Petersburg State University
St. Petersburg, Russia 198504
vyahhi@spbsu.ru

Adrien Goëffon
INRIA Bordeaux Sud-Ouest
351 cours de la Libération
33400 Talence, France
goeffon@labri.fr

Macha Nikolski
CNRS / LaBRI
Université Bordeaux I
351 cours de la Libération
33400 Talence, France
macha@labri.fr

David J. Sherman
INRIA Bordeaux Sud-Ouest
351 cours de la Libération
33400 Talence, France
david@labri.fr

ABSTRACT

A genome rearrangement scenario describes a series of chromosome fusion, fission, and translocation operations that suffice to rewrite one genome into another. Exact algorithmic methods for this important problem focus on providing one solution, while the set of distance-wise equivalent scenarios is very large. Moreover, no criteria for filtering for biologically plausible scenarios is currently proposed. We present an original metaheuristic method that uses Ant Colony Optimization to randomly explore the space of optimal and suboptimal rearrangement scenarios. It improves on the state of the art both by permitting large-scale enumeration of optimal scenarios, and by labeling each with metrics that can be used for post-processing filtering based on biological constraints.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; J.3 [**Life and Medical Science**]: Biology and genetics

Keywords

Genome rearrangement, Ant Colony Optimization.

1. INTRODUCTION

The genomes of contemporary species have been shaped by a number of evolutionary mechanisms, one of which is genome rearrangement. The most commonly studied evolutionary mechanism is that of *punctual mutations* that modifies the nucleotidic composition of a given genome. However, this type of analysis is not sufficient to infer evolutionary history. For example, it has been shown that the major part of genes within *Brassica olearacea* and *Brassica campestri* are identical up to 99% but their genomes differ in their size and gene order [17]. Large-scale mutations that involve genomic order rearrangements of large sengments of DNA mutations constitute a complementary approach to study evolutionary events. This field was pioneered in the 1930's by Dobzhansky and Sturtevant [6].

Formulating hypotheses as to which rearrangements took place between the ancestral and current genomes provides important insight into the mechanisms of molecular evolution operating on genomes. In this context, computing the minimal number of rearrangement steps defines a *distance measure* between genomes, which is a well-studied mathematical problem [10, 23, 16]. Exhibiting a sequence of steps that transforms one genome into another while respecting the rearrangement distance is a related question, and a response to it produces a *parsimonious rearrangement scenario*.

Hannenhalli and Pevzner developed the first polynomial-time $\mathcal{O}(n^4)$ algorithm (where n is the number of genomic elements) to compute a parsimonious rearrangement scenario [11]. Tannier and Sagot proposed in [22] a $\mathcal{O}(n^{3/2}\sqrt{\log n})$ algorithm for sorting reversals only, bound recently improved by Swenson *et al.* [21]. The drawback of these approaches is that they provide one, unique solution, while the solution space of all possible parsimonious scenarios is quite large [3]. It is for the same reason that enumeration approaches such as [19] are quite impractical.

Instead of computing one parsimonious scenario, the question then becomes how to find collections of rearrangement scenarios that are both parsimonious and biologically plausible. Reducing the search space by introducing additional biological constraints is one of the possible approaches. Lefebvre et al. [14] take into the account the length of reversed chromosomal segments, considering that small reversals are more frequent, based on the observation that they are the most numerous when comparing closely related species [18]. Bergeron et al. [2] consider only scenarios that conserve cer-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-325-9/09/07 ...$5.00.

tain common structures between the genomes, while in [3] the authors consider a way to build sets of equivalent scenarios and an algorithmic solution to do this is proposed in [4]. However, the latter remains impractical for large genomes.

A route different from classical combinatorial approaches has been chosen in the recent work by Darling et al. [5]. The authors apply a Bayesian statistical method in order to sample the genome rearrangements in *Y. pestis*. Genomic rearrangements are considered as a continuous Markov process operating on a given phylogenetic tree, with all rearrangement events being equally likely. From this the authors deduce the likelihood of a genome rearrangement scenario under the assumption that the mutation rates along the tree branches are known [15].

In this paper we explore the search space of rearrangement scenarios using a novel Ant Colony Optimisation (ACO) [7] method. Our method makes it possible to explore parsimonious (optimal), as well as close to optimal, scenarios. The latter is of particular interest since there is no reason to suppose *a* priori that there exists a mathematically parcimonious solution that respects all of the biological constraints (such as for example never to create along the scenario a genome having a chromosome with no centromere). Our algorithm allows us to set metrics, such as pheromone quantity, along the scenarios that were explored that can be further used to select the most biologically pertinent scenarios. Moreover, we propose a framework that makes possible to compute and store (potentially) each feasible scenario between two genomes.

2. REARRANGEMENT SCENARIOS

2.1 Preliminaries

In the standard way we encode multichromosomal genomes as signed permutations of genomic markers. A *chromosome* $\pi = (\pi_1, ..., \pi_m)$ is represented by a sequence of genomic markers whose sign indicates their relative direction on the chromosome. A size-n *multichromosomal genome* Π is defined as a set of chromosomes $\{\pi^1, ..., \pi^N\}$ s.t. $\sum_i |\pi^i| = n$. Markers take their values from the set of ordinals $1, ..., n$; no given marker appears more than once in a given genome. Notice that there is no semantical order between chromosomes, and that $\pi = (\pi_1, ..., \pi_m)$ is equivalent to $-\pi = (-\pi_m, ..., -\pi_1)$.

In order to exclude equivalent genome representations from our study, we use the following *canonical form*: (1) chromosomes are sorted by their first markers, and (2) the absolute value of the first marker of a given chromosome is smaller than that of its last marker.

For example, a genome Π having three chromosomes $\pi^1 = (-3, 1, 5, 4)$, $\pi^2 = (7, 6)$, and $\pi^3 = (9, 2, -8)$, becomes in the canonical form Π^c with $\pi^1 = (-6, -7)$, $\pi^2 = (-3, 1, 5, 4)$ and $\pi^3 = (8, -2, -9)$. In the rest of this paper all genomes are considered to be in canonical form.

We follow the Hannenhalli-Pevzner theory [10] and consider the standard set of rearrangements that can be applied to a genome: reversals, translocations, fissions and fusions. Accordingly, we use the rearrangement distance measure d introduced in the same paper. Since [1], this distance can be computed in linear time.

The four rearrangement operations are informally defined herebelow:

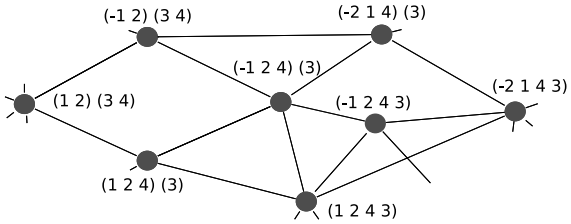

Figure 1: An example of a subgraph of permutation graph

- A *reversal* concerns one chromosome $\pi = (\pi_1, ..., \pi_i, ..., \pi_j, ..., \pi_n)$ and produces a chromosome $\pi' = (\pi_1, ..., -\pi_j, ..., -\pi_i, ..., \pi_n)$, $\forall (i, j) \neq (1, n)$.
- A *fusion* concerns two chromosomes $\pi^1 = (\pi^1_1, ..., \pi^1_m)$ and $\pi^2 = (\pi^2_1, ..., \pi^2_n)$ and produces one of the two possible chromosomes $(\pi^1_1, ..., \pi^1_m, \pi^2_1, ..., \pi^2_n)$ or $(\pi^1_1, ..., \pi^1_m, -\pi^2_n, ..., -\pi^2_1)$.
- A *fission* transforms one chromosome $\pi = (\pi_1, ..., \pi_i, \pi_{i+1}, ..., \pi_n)$ into two chromosomes $\pi' = (\pi_1, ..., \pi_i)$ and $\pi'' = (\pi_{i+1}, ...\pi_n)$, $\forall i < n$.
- A *translocation* exchanges pieces of two chromosomes $\pi^1 = (\pi^1_1, ..., \pi^1_i, \pi^1_{i+1}, ..., \pi^1_m)$ and $\pi^2 = (\pi^2_1, ..., \pi^2_j, \pi^2_{j+1}, ..., \pi^2_n)$. It produces one of the two possibles couples of chromosomes $\{(\pi^1_1, ..., \pi^1_i, \pi^2_{j+1}, ..., \pi^2_n), (\pi^2_1, ..., \pi^2_j, \pi^1_{i+1}, ..., \pi^1_m)\}$ or $\{(\pi^1_1, ..., \pi^1_i, -\pi^2_j, ..., -\pi^2_1), (-\pi^1_m, ..., -\pi^1_{i+1}, \pi^2_{j+1}, ..., \pi^2_n)\}$, $\forall i < m, j < n$.

2.2 Permutation graph

We consider the permutation graph $\mathcal{G} = \langle V, E \rangle$, where every vertex represent an unique size-n multichromosomal genome. Edges represent genome rearrangements, i.e. two genomes Π and Γ are adjacent in the graph if and only if $d(\Pi, \Gamma) = 1$ (see for an example Figure 1). Since the distance function d is symmetric, $\mathcal{G}$ has inverse edge for every direct edge.

PROPOSITION 1. *The number of distinct size-n multichromosomal genomes is given by:*

$$|V| = \sum_{c=1}^{n} \frac{C_n^{c-1} \cdot 2^{n-c} \cdot n!}{c!}$$

PROOF. Let us consider size-n genomes. There are $n!$ distinct sequences of n ordinals, hence $n!$ unichromosomal unsigned genomes, and consequently $2^n \cdot n!$ unichromosomal signed genomes. Let us first consider genomes with exactly c chromosomes. There are C_n^{c-1} partitions of one unichromosmal genome into c chromosomes. However, every distinct genome has $2^c \cdot c!$ equivalent versions since the set of chromosomes is not ordered ($c!$), and each chromosome can be read in both directions (2^c). Thus we obtain $(C_n^{c-1} \cdot 2^{n-c} \cdot n!)/c!$ distinct genomes of c chromosomes in canonical form. A size-n genome can have between 1 and n chromosomes; consequently there are $|V| = \sum_{c=1}^{n} (C_n^{c-1} \cdot 2^{n-c} \cdot n!)/c!$ distinct size-n multichromosomal genomes. □

The problem we focus on is to find parsimonious rearrangement scenarios between two genomes Π and Γ. If we consider that no intermediate genome should have either

fewer or more chromosomes than Π and Γ, then $\mathcal{G}$ can be reduced, with $|V| = \sum_{c=c-}^{c+}(C_n^{c-1} \cdot 2^{n-c} \cdot n!)/c!$ ($c-$ and $c+$ represent respectively the minimum and maximum number of chromosomes of Π and Γ).

For a given number of genomic markers n, the number of distance-1 neighbors for a genome varies slightly as a function of the chromosome number c. However, on each genome one can make $\mathcal{O}(n^2)$ reversal and translocation operations, $\mathcal{O}(n)$ fissions, and $\mathcal{O}(n)$ fusion operations. Thus, the size of the neighborhood for each size-n genome in the permutation graph is $\mathcal{O}(n^2)$.

2.3 Parsimonious scenarios

In this work we present a framework for the computation of sets of *good* rearrangement scenarios, with respect to the notion of parsimony.

DEFINITION 1. *A **scenario** of length m between two size-n multichromosomal genomes Π and Γ is a sequence $S = (\Pi_0, \ldots, \Pi_m)$ s.t.*

- $\forall\ i \in [0, m], \Pi_i$ *is a size-n multichromosomal genome,*
- $\Pi_0 = \Pi$ *and* $\Pi_m = \Gamma$, *and*
- $\forall\ i \in [0, m-1], d(\Pi_i, \Pi_{i+1}) = 1$.

*If $m = d(\Pi, \Gamma)$, then S is a **parsimonious scenario**. A sequence S s.t. $m = d(\Pi, \Gamma) + \alpha$ without cycles, that is, where $\Pi_i \neq \Pi_j \forall i, j \in \{0, \ldots, m\} (i \neq j)$, is called an $\alpha-$**parsimonious scenario***

Note that every path in $\mathcal{G}$ corresponds to one scenario, and that the set of parsimonious scenarios between two genomes Π and Γ corresponds to the set of shortest paths between Π and Γ.

Obviously, storing the complete graph $\mathcal{G}$ is not conceivable, even for small n. In the next section, we present an ACO-based heuristic that produces a reasonable subset of genomes and edges, in which pertinent and well-diversified rearrangement scenarios can be determined.

3. AN ACO ALGORITHM FOR COMPUTING REARRANGEMENT SCENARIOS

Given two genomes Π and Γ we define a hybrid ACO algorithm methood in order to find parsimonious and $\alpha-$parsimonious scenarios. ACO (Ant Colony Optimization) [7] is a well-known metaheuristic inspired by the behaviour of ants seeking a path from their nest to a food source.

Each ant navigates in space in search of a destination. After reaching it, ants deposit some quantity of non persistent pheromone along their paths, which creates a bias for consecutive searches. After some time, ants cluster along the paths with highest pheromone values, which are generally the shortest.

ACO algorithms are successfully applied to NP-hard problems like multidimensional knapsack problem [13] or graph coloring [8].

Here we propose to apply the ACO strategy for the search of shortest rearrangement scenarios. Considering the size of the rearrangement graph, the algorithm hybridizes ACO with local search (with a random walk mechanism) in order to guide the search.

3.1 General description

At the beginning, we record only two genomes from $\mathcal{G}$: the starting node Π (the *nest*) and the destination node Γ (the *food source*). Notice that Π and Γ can be switched with no influence on the result. Pheromone trail values required for ACO are stored on the edges and are denoted by τ_e, where e is an edge. For any ant sitting on a given vertex, the pheromone values on the outgoing edges can be seen as the probability of choosing this edge for the next move.

We consider S generations of ants with A ants in each generation. At every generation each ant starts at the nest node Π and tries to reach the food source Γ independently. A given ant builds its tour $T = (e_1, ..., e_l)$ in l steps. After all tours are calculated, new pheromone values are added to all edges participating in each tour, the objective being to encourage the next generation of ants to favor (or not) certain edges more than others. The pheromone value that an ant creates for an edge e is calculated as $\Delta\tau_e = 1/(l - d + 1)$ where $d = d(\Pi, \Gamma)$. For all edges present in T, pheromone values are updated, s.t. $\tau_e \leftarrow \tau_e + \Delta\tau_e$.

Finally, we simulate a pheromone evaporation mechanism with rate ρ, where ρ represents the persistance of a pheromone trail ($0 \leqslant \rho \leqslant 1$). When each tour of one generation is ended, every edge $e \in E$ has to be updated, s.t. $\tau_e \leftarrow \rho \cdot \tau_e$. Nevertheless, since $\mathcal{G}$ is huge, we do not consider all the theoretically possible edges $e \in E$ but only those that are currently recorded, as well as some constant default value of pheromone for all the other (virtual) edges.

After a tour T, we record each edge e from T having $\tau_e \geqslant \varepsilon$ (ε is a chosen threshold), as well as the associated vertices. In order to keep the the size of the graph datastructure modest, we delete elements that are considered to have insufficient support (after a tour or an pheromone evaporation process), namely:

- edges such that $\tau_e < \varepsilon$, and
- neighborless vertices (except those labelled by Π and Γ).

Thus, an ant following a tour T that corresponds to a parsimonious rearrangement scenario between Π and Γ (that is $|T| = d$) will add 1 unit of pheromone on its trail. An ant following a 1-parsimonious scenario (that is $|T| = d + 1$) will add $1/2$, and an ant following a 2-parsimonious scenario (that is $|T| = d + 2$) will add $1/3$, and so on.

The algorithm stops when the number of generations S is reached. S is tuned experimentally in order to reach some degree of convergence (see Experiments section, below) and a sufficient number of pertinent scenarios.

3.2 Hybridization with a random walk mechanism

A practical issue comes from two observations. First, $\mathcal{G}$ is very large and has many cycles. Consequently, for a given ant it is possible to have infinite tours. We deal with this issue by introducing an attractive force that pushes ants to Γ. We adapt the idea of default pheromone to the situation by splitting τ into three different default values of pheromone. The first value τ^g is for *good* edges that lead ants closer to Γ, the second value τ^n is for *neutral* edges that do not change the remaining distance, and the third value τ^b is for *bad* edges that lead the ants away from Γ. So, at each vertex v for an edge e between v and u, the choice of the particular τ value is

- τ^g if $d(u,\Gamma) = d(v,\Gamma) - 1$,
- τ^n if $d(u,\Gamma) = d(v,\Gamma)$, and
- τ^b if $d(u,\Gamma) = d(v,\Gamma) + 1$.

Furthermore, at each vertex these values are normalized by dividing them by the number of outgoing edges of the corresponding kind. This is done since, if there is a large number of egdes at a given vertex and only a small portion of them is good, there exists a bias for choosing the wrong direction. Thus, each ant simulates a local search process with a random walk mechanism, since we favor good moves and allow bad ones with lower probability. This mechanism requires that the values of τ are chosen such that $\tau^g \geqslant \tau^n \geqslant \tau^b$.

Clearly, the influence of τ values is high when an edge is explored for the first time by an ant or is seldom visited, and is low for the highly-visited areas of $\mathcal{G}$ where pheromone values are high. If $\tau^g = \tau^n = \tau^b$ we obtain exactly the classic situation of default pheromone in ACO. On the contrary, a high τ^g value combined with small τ^b values increases the ants' attraction to Γ. In particular, if $\tau^g > 0 = \tau^n = \tau^b$, then only optimal scenarios are explored.

3.3 Computation improvements

The complexity of the ACO algorithm depends mainly on the neighbor selection process. Indeed, using the standard algorithm, all neighbors are evaluated at each step of the process, leading to a $\mathcal{O}(n^3)$ complexity: there is $\mathcal{O}(n^2)$ neighbors, and $\mathcal{O}(n)$ time to evaluate each unrecorded neighbor (required to calculate τ values).

In order to reduce the computational effort, we propose a faster version, which only differs by the neighbor selection process. At each step, only k random neighbors are evaluated (k being a constant parameter). Parameter k has a low influence on the quality of tours, except during the 3 or 4 last steps (improving neighbors become rarer when the food source is close). That is why in this version of our algorithm we use the complete neighborhood evaluation only in the three last steps of the search.

Resulting rearrangement scenarios. Running the algorithm results in a subgraph $\mathcal{H} \subset \mathcal{G}$ that contains vertices connected by edges with pheromone. This graph summarizes a set of scenarios. Extracting one scenario from $\mathcal{H}$ can be seen as an ant tour that visits only existing vertices and follows only existing edges in $\mathcal{H}$, from Π to Γ; furthermore, final pheromone values can be used as weights for the neighbor selections.

4. EXPERIMENTS AND APPLICATIONS

Benchmarks. The Génolevures Consortium[1] study the evolutionary history of the Hemiascomycetous yeasts. We selected for this experiment five completely sequenced and annotated genomes from the protoploid *Saccharamycetacae*: *Kluyveromyces lactis*, *Saccharomyces kluyveri*, *Zygosaccharomyces rouxii*, *Ashbya (Eremothecium) gossypii* and *Kluyveromyces thermotolerans*[2] [20], as well as a median genome M representing an ancestral genome for these yeasts [12]. Concerning the presentation of the Median Genome Problem and our algorithm FAUCILS which calculates median genomes, we refer the interested reader to [9].

Pairwise distances d between the contemporary genomes and the median are shown in Table 1. Recall that α represents the length difference between one scenario and the most parsimonious scenarios.

Parameters. The experimental setup depends on a number of parameters. First of all, the input data size, that is, the number of genomic markers n.

Our adaptation of ACO for computation of rearrangement scenarios has seven parameters:

- S - number of ant generations,
- A - number of ants in each generation,
- $\varepsilon \geqslant 0$ - pheromone threshold for edge deletion,
- $0 \leqslant \rho \leqslant 1$ - persistence of pheromone trail,
- $\tau^g \geqslant \tau^n \geqslant \tau^b$ - default pheromone values.

A first experiment run with default parameters $S = A = 50$, $\varepsilon = 10^{-4}$, $\tau^g = 1$, $\tau^n = 10^{-1}$, $\tau^b = 10^{-2}$ and $\rho = 0.75$ results in the scenario distributions shown on Figure 2.

	Sakl	Klth	Zyro	Ergo	Klla
size	135	135	135	135	135
chromosomes	8	8	7	7	6
d	23	29	69	74	89
min scenario	23	29	69	75	90
max scenario	34	38	88	94	112
mean α	3.62	0.25	5.67	7.41	7.26

Table 1: Rearrangement distances d from the 5 Hemiascomycetous yeasts genome to the ancestral genome M and statistics for the runs for M to Sakl, Klth, Zyro, Ergo and Klla instances with $S = A = 50$, $\varepsilon = 10^{-4}$, $\tau^g = 1$, $\tau^n = 10^{-1}$, $\tau^b = 10^{-2}$, $\rho = 0.75$: min and max scenario (shortest and longest simple paths from Π to Γ), and mean values for α. The size (135) is the number of common markers used for this study.

The smaller numbers of scenarios in Ergo, Zyro and Klla instances can be explained by the larger number of long scenarios, where $\Delta\tau = 1/(l - d + 1)$ was so small that it was negligible and quickly evaporated. The bigger difference between best and average tours can be explained by the fact that distances between M and Ergo, Zyro and Klla are greater than between M and Klth or Sakl. For example, Sakl-M distance is 3 times smaller than Zyro-M, hence ants have 3 times fewer steps in which to make "mistakes" (following bad or neutral edges). Since the same τ^g, τ^n and τ^b are used for all instances, the probabilities of making these mistakes at each step are the same. Consequently, the difference between the best and the average ant's tour for the Sakl-M instance is 3 times smaller than for Zyro-M. No parsimonious scenario was found for Ergo-M and Klla-M for same reason described above (see Table 1). We conclude that the greater the distances, the greater the attractive force has to be used in order to counter this effect.

[1] `http://genolevures.org`

[2] Abbreviations: Klla, *K. lactis*; Sakl, *S. kluyveri*; Zyro, *Z. rouxii*; Ergo, *A. gossypii*; Klth, *K. thermotolerans*.

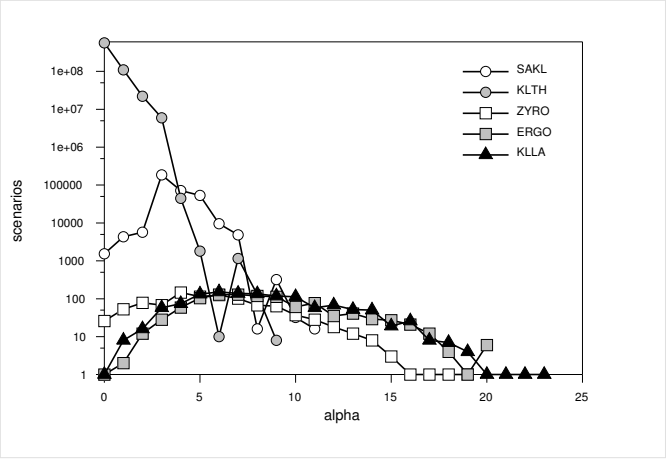

Figure 2: Distribution of α values for computed scenarios

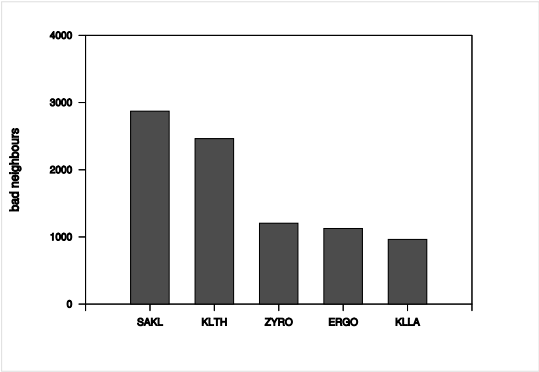

Figure 3: Number of bad to good edges for M to Sakl, Klth, Zyro, Ergo and Klla instances

The large number of scenarios for Klth-M instance can be explained by the high degree of sharing of edges between the optimal tours, and hence the growth of possible scenarios in the resulting graph.

In Figure 3 we show the average number of bad neighbors per one good edge for every instance. Closer instances have a higher proportion of bad edges than distant instances. Intuitively, when an ant starts its tour and is quite far from Γ, it has a large choice of good edges at each vertex. The closer an ant gets to Γ, the smaller is the number of good edges. Hence the ratio of bad to good edges is lower for vertices that are more distant from Γ. And the larger the distance between Π and Γ, the smaller is the overall ratio.

As expected, influence of τ^g versus τ^n and τ^b is high. Indeed, in Table 2, we can see that considering sufficiently high τ^g/τ^b ratios, returned α values are small and the proportion of parsimonious scenarios out of all uncovered scenarios is greater than 70%. For fixed values of τ, the value of ε has a similar (but less important) effect of the proportion of parsimonious scenarios. Notice, that for more complicated instances such as Ergo (greater ratio d/size) choosing sufficient attractive force τ^g is essential since when this value is too small, no parsimonious scenarios are found (see Table 2 as well as 1 where only default values are shown).

(Π, Γ)	τ^g	τ^n	τ^b	ε	min	max	mean α	% opt.
(Klth,M) $[d = 29]$	1	10^{-1}	10^{-2}	10^{-3}	29	40	1.63	22.6%
	1	10^{-1}	10^{-2}	10^{-4}	29	38	0.25	80.3%
	1	10^{-1}	10^{-2}	10^{-5}	29	35	1.00	35.7%
	1	10^{-1}	10^{-3}	10^{-3}	29	32	0.22	79.4%
	1	10^{-1}	10^{-3}	10^{-4}	29	32	0.27	76.0%
	1	10^{-1}	10^{-3}	10^{-5}	29	32	0.04	96.1%
	1	10^{-2}	10^{-4}	10^{-3}	29	31	0.25	75.6%
	1	10^{-2}	10^{-4}	10^{-4}	29	32	0.25	76.8%
	1	10^{-2}	10^{-4}	10^{-5}	29	31	0.19	81.3%
(Zyro,M) $[d = 69]$	1	10^{-1}	10^{-2}	10^{-3}	70	87	6.84	0%
	1	10^{-1}	10^{-2}	10^{-5}	70	88	7.00	0%
	1	10^{-1}	10^{-3}	10^{-3}	69	73	0.36	71.6%
	1	10^{-1}	10^{-3}	10^{-4}	69	74	0.29	75.6%
	1	10^{-1}	10^{-3}	10^{-5}	69	72	0.14	86.3%
	1	10^{-2}	10^{-4}	10^{-3}	69	72	0.32	73.3%
	1	10^{-2}	10^{-4}	10^{-4}	69	73	0.27	77.1%
	1	10^{-2}	10^{-4}	10^{-5}	69	73	0.30	73.0%
(Ergo,M) $[d = 74]$	1	10^{-1}	10^{-2}	10^{-3}	76	94	7.44	0%
	1	10^{-1}	10^{-2}	10^{-4}	75	94	8.41	0%
	1	10^{-1}	10^{-2}	10^{-5}	76	94	7.27	0%
	1	10^{-1}	10^{-3}	10^{-3}	74	79	0.38	72.1%
	1	10^{-1}	10^{-3}	10^{-4}	74	82	0.37	70.7%
	1	10^{-1}	10^{-3}	10^{-5}	74	78	0.05	95.4%
	1	10^{-2}	10^{-4}	10^{-3}	74	80	0.33	73.3%
	1	10^{-2}	10^{-4}	10^{-4}	74	78	0.32	74.2%
	1	10^{-2}	10^{-4}	10^{-5}	74	79	0.36	67.9%

Table 2: Large-scale scenario computations between Ergo, Klth, Zyro instances and the common median genome, for selected parameters. Each run was 50 generations of 50 ants, with $\rho = 0.75$. 'min' and 'max' indicate minimum and maximum scenarios. '% opt.' is the proportion of optimal scenarios among all computed.

Finally, preliminary tests of results convergence were performed on the most representative distance-wise Zyro-M instance. Varying the number of ant generations, and keeping all the other parameters at default values (see Table 1) yields to the consistent growth of the proportion of the number of parsimonious scenarios (out of the total number of scenarios) that follows the growth of the ants generations. This indicates that the required completeness of the exploration of the parsimonious and α-parsiminious scenarios has to be tuned by using this parameter to the particular application case data. Maintaining a sufficient total number of scenarios can be done by decreasing the value of ε. Indeed, since ants are converging to Γ in an unstructured fashion, even optimal scenarios can be less and less visited. These scenarios can be kept in memory by using a smaller value of ε.

5. CONCLUSIONS

Computing biologically plausible genome rearrangement scenarios is a challenging problem in large-scale comparative genomics, and current exact methods fail to meet the needs of the application domain by only computing a single scenario out of a very large pool of biologically plausible results. Using metaheuristic methods based on Ant Colony Optimization, we have developed a novel algorithm that computes and stores a large population of parsimonious and α-parsimonious (that is, suboptimal) solutions, labeled with metrics permitting useful post-processing. Experimental results on five challenging eukaryote genomes from the protoploid *Saccharomycetacae* reveal that this algorithm works

quite well in practice and can generate hundreds of optimal scenarios amenable to further biological analysis.

The data structure and general framework defined here can be used in several ways, depending on the application. It can be used for brute-force generation of a very large number of different scenarios, useful for studying the statistical structure of the population of plausible reconstructions. It can be used to focus in on probable sub-scenarios, around a postulated event mapped (for example) to an ancestor node of a phylogenetic tree. Data structures for related instances can be crossed to find common sub-scenarios between several related genomes.

Two obvious improvements to the method can be foreseen. Based on feedback from large-scale application, specific strategies for pheromone persistence and initial values can be devised. Second, perhaps combining this approach with our FAUCILS method for rearrangement trees[9], it should be possible to compute consensus Steiner trees of rearrangements, through a large number of iterations and final node selection based on cumulative pheromone quantities.

6. REFERENCES

[1] D. A. Bader, B. Moret, and M. Yan. A linear-time algorithm for computing inversion distances between signed permutations with an experimental study. Journal of Computational Biology, 8(5):483491, 2001.

[2] S. Berard, A. Bergeron, C. Chauve, and C. Paul. Perfect sorting by reversals is not always difficult. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(1):4–16, 2007.

[3] A. Bergeron, C. Chauve, T. Hartman, and K. Saint-Onge. On the properties of sequences of reversals that sort a signed permutation. In *Proceedings of JOBIM 2002*, pages 99–108, 2002.

[4] M. Braga, M.-F. Sagot, C. Scornavacca, and E. Tannier. Exploring the solution space of sorting by reversals, with experiments and an application to evolution. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(3):348–356, 2008.

[5] A. Darling, I. Mikls, and M. Ragan. Dynamics of genome rearrangement in bacterial populations. *PLoS Genetics*, 4(7):e1000128 doi:10.1371/journal.pgen.1000128, 2008.

[6] T. Dobzhansky and A. H. Sturtevant Inversions in the chromosomes of drosophila pseudoobscura, *Genetics*, 23(1):28–64, 1938.

[7] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

[8] K.A. Dowsland and J.M. Thompson An improved ant colony optimisation heuristic for graph colouring *Discrete Applied Mathematics*, 156(3):313–324, 2008.

[9] A. Goëffon, M. Nikolski, and D.J. Sherman. An Efficient Probabilistic Population-Based Descent for the Median Genome Problem. *Proceedings of the 10th annual ACM SIGEVO conference on Genetic and evolutionary computation (GECCO 2008)*, pages 315–322, 2008.

[10] S. Hannenhalli and P.A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). *Proceedings of twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 178–189, 1995.

[11] S. Hannenhalli and P.A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 581–592, 1995.

[12] G. Jean and M. Nikolski. Mining the semantics of genome super-blocks to infer ancestral architectures. *Journal of Computational Biology*, accepted for publication.

[13] L. Ke, Zuren Feng, Zhigang Ren and Xiaoliang Wei An ant colony optimization approach for the multidimensional knapsack problem *Journal of Heuristics*, 2008.

[14] J.F. Lefebvre, N. El-Mabrouk, E. Tillier, and D. Sankoff. Detection and validation of single gene inversions. In *ISMB (Supplement of Bioinformatics)*, pages 190–196, 2003.

[15] I. Milos. Mcmc genome rearrangement. *Bioinformatics*, 19 Suppl 2:ii130–ii137, 2003.

[16] M. Ozery-Flato and R. Shamir. Two notes on genome rearrangement. *J. Bioinformatics Comput. Biol.*, 1(1):71–94, 2003.

[17] J.D. Palmer and L.A. Herbon Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence *Journal of Molecular Evolution*, 28:87–97, 1988.

[18] C. Seoighe, Federspiel N.J.T., Hansen N., Bivolarovic V., Surzycki R., Tamse R., Komp C., Huizar L., Davis R.W., Scherer S., Tait E., Shaw D.J., Harris D., Murphy L., Oliver K., Taylor K., Rajandream M.A., Barrell B.G., and Wolfe K.H. Prevalence of small inversions in yeast gene order evolution. *Proc. Natl. Acad. Sci. USA*, 97:14433–14437, 2000.

[19] A. Siepel. An algorithm to enumerate sorting reversals for signed permutations. *Journal of Comp. Biol.*, 10(3-4):575–597, 2003.

[20] D. Sherman, T. Martin, M. Nikolski, C. Cayla, J.-L. Souciet, and P. Durrens. Genolevure: protein families and synteny among complete hemiascomycetous yeast proteomes and genomes. *Nucleic Acids Research (NAR)*, Database Issue: D550-D554, 2008

[21] K. Swenson, V. Rajan, Y. Lin, and B. Moret Sorting Signed Permutations by Inversions in O(n log n) Time. To appear at *RECOMB'09*.

[22] E. Tannier, A. Bergeron, and M.-F. Sagot. Advances on sorting by reversals. *Discrete Appl. Math.*, 155(6-7):881–888, 2007.

[23] G. Tesler. Efficient algorithms for multichromosomal genome rearrangements. *J. Comput. Syst. Sci.*, 65(3):587–609, 2002.