BIOINFORMATICS

Vol. 18 Suppl. 1 2002
Pages S303–S311

Multiple genome rearrangement: a general approach via the evolutionary genome graph

Dmitry Korkin and Lev Goldfarb*

Faculty of Computer Science, University of New Brunswick, Fredericton, NB, E3B 5A3, Canada

Received on January 24, 2002; revised and accepted on March 31, 2002

ABSTRACT

Motivation: In spite of a well-known fact that genome rearrangements are supposed to be viewed in the light of the evolutionary relationships within and between the species involved, no formal *underlying* framework based on the evolutionary considerations for treating the questions arising in the area has been proposed. If such an underlying framework is provided, all the basic questions in the area can be posed in a biologically more appropriate and useful form: e.g., the similarity between two genomes can then be computed via the nearest ancestor, rather than 'directly', ignoring the evolutionary connections.

Results: We outline an evolution-based general framework for answering questions related to the multiple genome rearrangement. In the proposed model, the evolutionary genome graph (EG-graph) encapsulates an evolutionary history of a genome family. For a set of all EG-graphs, we introduce a family of similarity measures, each defined via a fixed set of genome transformations. Given a set of genomes and restricting ourselves to the transpositions, an algorithm for constructing an EG-graph is presented. We also present the experimental results in the form of an EG-graph for a set of concrete genomes (for several species). This EG-graph turns out to be very close to the corresponding known phylogenetic tree.

Contact: dkorkin@unb.ca

Keywords: genome rearrangement; evolutionary model; gene transformations; longest common subsequence; ETS framework.

1 INTRODUCTION

The role of evolution in biology is so central that the two words are almost synonymous (Dobzhansky, 1973). As far as questions arising in connection with the multiple genome rearrangements are concerned, they have not yet been formulated and answered satisfactorily within an evolution-based formal framework. This is in spite of a well-known fact that genome rearrangements are supposed to be treated mainly as an integral part of the evolutionary information, or relations, between the species.

In this paper we outline an evolution-based general framework for embedding some questions related to the multiple genome rearrangement. It should be quite clear that if the proposed model proves satisfactory, many additional kinds of useful questions could be addressed within such a model.

The main distinctive feature of the proposed model, as compared with the conventional genome rearrangement approaches, is that our model allows one to reconstruct evolutionary relationships between the species based on the evolutionary genome graph (EG-graph). The model also allows one to reconstruct other than binary relations between the species; for example, EG-graph can accommodate three genomes that have one closest common ancestor. Moreover, the model allows for a *natural* integration of point mutations into the overall scheme.

The proposed model was inspired by a much more general 'evolution-based' formalism for structural object representation within an inductive framework—evolving transformations system (ETS) model—developed over last 15 years (Goldfarb *et al.*, 2001; Goldfarb and Korkin, 2001). The ETS formalism was motivated by the integration of the concept of similarity into a generative inductive class representation. The resulting class representation can be constructed based on a small set of class examples and is conceptualized via a weighted set of transformations acting on the class progenitor (the common ancestor).

This paper is organized as follows. In Section 2, we present the basic definitions, including that of EG-graph, and formulate two versions of the basic problem of genome development (which are analogues of the corresponding genome rearrangement problem). In Section 3, we introduce an evolution-based model with reversals and insertions as non-local evolutionary mutations (transformations). We define the corresponding evolutionary genome graph and discuss its properties. Next, we discuss the relationship between evolutionary transformations and the traditional genome rearrangement transformations.

*To whom the correspondence should be addressed.

We also introduce an evolutionary similarity measure for this model. At the end of this section, we introduce context-sensitive genome transformations and discuss how they could make the above model more accurate. In Section 4, we outline a method of constructing the EG-graph for a given set of genomes. We present the algorithm and analyse its complexity. In Section 5, we present the results of an experiment, i.e., an EG-graph for genomes of several species, and compare them to the corresponding known phylogenetic trees. In the last section, we conclude with a brief summary of the paper and propose some future directions.

2 BASIC DEFINITIONS AND PROBLEM FORMULATION

Some of the following concepts are standard, while the problems are formulated for the first time and rely on the evolutionary genome graph.

2.1 Basic definitions

DEFINITION 1. Let $\Sigma_0 = \{A, C, G, T\}$ be an alphabet. A **gene** is defined to be an element of the set of strings $\boldsymbol{S} = \Sigma_0^* \backslash \{\Lambda\}$, and a **genome** $\boldsymbol{G}$ is defined to be a sequence of genes: $\boldsymbol{G} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)$, $\boldsymbol{s}_k \in \boldsymbol{S}, 1 \leqslant k \leqslant n$. The corresponding string $G = \boldsymbol{s}_1 \circ \boldsymbol{s}_2 \circ \ldots \circ \boldsymbol{s}_n$ is called the **genome sequence**. The **length**, $|\boldsymbol{G}|$, of a genome is defined to be the length of its genome sequence. The set of all genomes is denoted as $\boldsymbol{\Gamma}$.

Note that the circular genomes can and will also be represented as sequences (see Section 4.1). The next definition formally introduces the concept of gene order transformation, which leaves the length of the genome unchanged.

DEFINITION 2. Let $\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n)$ and $\boldsymbol{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2 \ldots \boldsymbol{b}_n)$ be any genomes of the same length, such that there exists a permutation $\rho : \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$ satisfying $\boldsymbol{b}_i = \boldsymbol{a}_{\rho(i)}$ or $\boldsymbol{b}_i = \boldsymbol{a}'_{\rho(i)}$, where $\boldsymbol{a}'$ is a reversed $\boldsymbol{a}$. A mapping $f : \boldsymbol{\Gamma} \rightarrow \boldsymbol{\Gamma}$, $f(\boldsymbol{A}) = \boldsymbol{B}$, is called a **gene order transformation** *for* genome $\boldsymbol{A}$, or simply **gor-transformation**. The set of all gor-transformations is denoted as $\boldsymbol{\Phi}_{gor}$.

The following definition formally introduces the concept of genome rearrangement.

DEFINITION 3. A **rearrangement of genome** $\boldsymbol{G}$ terminating in genome $\boldsymbol{G}_m$ is a sequence $(f_1, f_2, \ldots, f_m)$, $m \geqslant 1$, $f_i \in \Phi_{gor}$, defined inductively as follows:

1. f_1 is a gor-transformation for genome $\boldsymbol{G}$ and $\boldsymbol{G}_1 = f_1(\boldsymbol{G})$
2. $\forall i, 1 < i \leqslant m$, f_i is a gor-transformation for genome $\boldsymbol{G}_{i-1}$ and $\boldsymbol{G}_i = f_i(\boldsymbol{G}_{i-1})$.

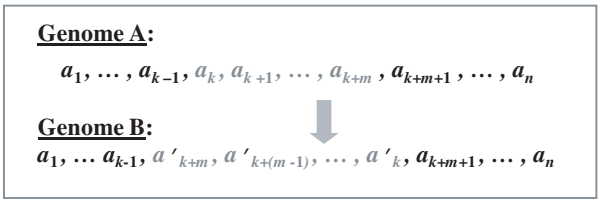

Fig. 1. A reversal of subsequence $\boldsymbol{A}_1 = \boldsymbol{a}_k, \boldsymbol{a}_{k+1}, \ldots, \boldsymbol{a}_{k+m}$.

The next two definitions review the concepts of subsequence and the multiple longest common subsequence.

DEFINITION 4. Let $\boldsymbol{a}$ be a sequence of length n over another alphabet $\Sigma = \{a, b, \ldots, z\}$. If sequence $\boldsymbol{a} = s_1 s_2 \ldots s_n$, $s_i \in \Sigma$, then sequence $\boldsymbol{b} = s_{i_1} s_{i_2} \ldots s_{i_k}$ is called a **subsequence** of $\boldsymbol{a}$, if $\forall j, 1 \leqslant j \leqslant k$:

$$1 \leqslant i_j \leqslant n,$$

and for all s and t, $1 \leqslant s < t \leqslant k : i_s < i_t$, where k is the length of $\boldsymbol{b}$.

DEFINITION 5. Let $S = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_d\}$ be a set of sequences over alphabet $\Sigma = \{a, b, \ldots, z\}$, of lengths $n_1, n_2, \ldots, n_d$, correspondingly. The **multiple longest common subsequence** (MLCS) for a set S is a sequence $\boldsymbol{b}$ such that:

1. $\boldsymbol{b}$ is a subsequence of $\boldsymbol{a}_i$, $\forall i$;
2. $\boldsymbol{b}$ is the longest sequence satisfying 1.

In case $d = 2$, MLCS is called simply the *longest common subsequence* (LCS).

There are two particular kinds of gene order transformations, i.e., reversals and transpositions, that are of interest.

DEFINITION 6. Let $\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n)$ and $\boldsymbol{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ be genomes and let f be a gor-transformation *for* genome $\boldsymbol{A}$ such that $\boldsymbol{B} = f(\boldsymbol{A})$. The gor-transformation f is called a **reversal** *of* a subsequence $\boldsymbol{A}_1 = (\boldsymbol{a}_k, \boldsymbol{a}_{k+1}, \ldots, \boldsymbol{a}_{k+m})$ of genes, if the corresponding permutation ρ (see Definition 2) is such that there exist integers $k, m \geqslant 1$ satisfying (see Figure 1):

$$\boldsymbol{b}_i = \boldsymbol{a}_i, 1 \leqslant i \leqslant k-1 \quad \text{or} \quad k+m+1 \leqslant i \leqslant n \quad (1)$$

$$\boldsymbol{b}_{k+i} = \boldsymbol{a}'_{k+(m-i)}, 0 \leqslant i \leqslant m. \quad (2)$$

DEFINITION 7. Let $\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n)$ and $\boldsymbol{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ be two genomes and let f be a gor-transformation *for* genome $\boldsymbol{A}$ such that $\boldsymbol{B} = f(\boldsymbol{A})$. The gor-transformation f is called a **transposition** *of* a subsequence $\boldsymbol{A}_1 = (\boldsymbol{a}_k, \boldsymbol{a}_{k+1}, \ldots, \boldsymbol{a}_{k+m})$ of genes, if the corresponding permutation ρ is such that there exist $j, k, m \geqslant 1$ satisfying (see Figure 2):

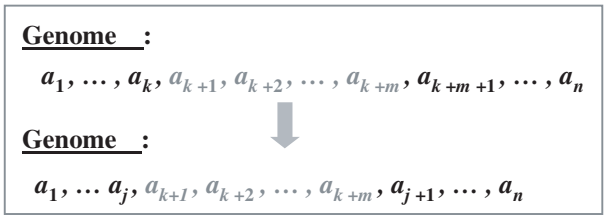

Fig. 2. A transposition of $A_1 = a_k, a_{k+1}, \dots, a_{k+m}$.

(1) $b_i = a_i$, $1 \leqslant i \leqslant \min(k, j)$ or $\max(k, j) + 1 + m \leqslant i \leqslant n$

(2) $b_{j+i} = a_{k+i}$, $1 \leqslant i \leqslant m$

(3) $b_{i+m} = a_i$, $j + 1 \leqslant i \leqslant k$

(note that we assume $j \leqslant k$; the case $j > k$ can be defined similarly).

The next definition formally introduces the concept of insertion transformation, which changes the length of the genome.

DEFINITION 8. Let $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_{n+m})$, $m > 0$, be two genomes. A mapping $f : \Gamma \to \Gamma$, where $f(A) = B$, is called **insertion transformation** (*for* a genome A) *of* sequence $A_1 = (c_1, c_2, \dots, c_m)$ of genes, or simply **ins-transformation**, if there exists k, $1 \leqslant k \leqslant n$, such that:

(1) $b_i = a_i$, $1 \leqslant i \leqslant k$

(2) $b_{k+i} = c_i$, $1 \leqslant i \leqslant m$

(3) $b_{i+k+m} = a_{i+k}$, $1 \leqslant i \leqslant n - k$.

The set of all ins-transformations will be denoted as Φ_{ins}. The set of all gor- and ins-transformations will be denoted as Φ.

One should note the use of pronouns 'for' and 'of' in the above definitions (and below).

In the next definition, we introduce the evolutionary version of the concept of genome rearrangement by adding to the set of gor-transformations the set of ins-transformations.

DEFINITION 9. A **development of genome G** terminating in genome G_m is a sequence $(f_1, f_2, \dots, f_m)$, $m \geqslant 1$, $f_i \in \Phi$, defined inductively as follows:

1. f_1 is a gor- or ins-transformation for genome G and $G_1 = f_1(G)$;
2. $\forall i, 1 < i \leqslant m$, f_i is a gor- or ins-transformation for genome G_{i-1} and $G_i = f_i(G_{i-1})$.

We will also need the concept of closest common ancestor useful from the evolutionary point of view.

DEFINITION 10. A genome C is called a **common ancestor** for a set of genomes Γ_1, if $\forall G \in \Gamma_1$ either $G = C$ or there exists a development of genome C terminating in G.

Let Γ_{1A} be the set of all common ancestors of Γ_1. A genome C is called a **closest common ancestor** for Γ_1, if it belongs to set Γ_{1C}, $\Gamma_{1C} \subseteq \Gamma_{1A}$, such that for any common ancestor G (of Γ_1) not in Γ_{1C} there exists $C' \in \Gamma_{1C}$ and a development of G terminating in C.

2.2 Evolutionary genome graph and problem formulation

The current evolutionary approaches to genome rearrangement have roots in phylogenetic approaches to the study of protein sequences and have appeared during the last ten years (Hannenhalli *et al.*, 1995; Sankoff *et al.*, 1992, 1996; Sankoff and Blanchette, 1998, 1999; El-Mabrouk, 2001).

In our approach, the evolutionary genome graph is supposed to represent the evolutionary dependencies for the given set of genomes (see Figure 3).

DEFINITION 11. In a directed graph *DG, DG = (V, E)*, a vertex from V all edges of which are outcoming will be called a **source node**. A vertex from V all edges of which are incoming will be called a **sink node**.

DEFINITION 12. Given a finite set of genomes $\Gamma_1 \subseteq \Gamma$, the **evolutionary genome graph**, or simply EG-graph, for Γ_1 is defined as connected directed labeled graph DG, $DG = (V, E)$, $l_V : V \to \Gamma$, $l_E : E \to \Phi$ (See Defintion 8), such that:

(1) there exists only one source node, V_0, $V_0 \in V$

(2) l_V is an injective mapping and $\Gamma_1 \subseteq l_V(V)$

(3) if $G_0 = l_V(V_0)$, then G_0 is a closest common ancestor for Γ_1.

It is easy to see that, in general, EG-graph may not be a tree: there could be two or more different paths between two vertices, e.g., one path obtained by performing the insertion of a gene g_1 for a genome G followed by a reversal of another gene g_2 for G and another path obtained by performing the reversal of g_2 first followed by the insertion of g_1 for the genome G.

Having introduced the concept of EG-graph, we can formulate the basic problem of genome development (which is an analogue of the corresponding genome rearrangement problem).

PROBLEM 1. *Given a finite set of genomes* $\Gamma_1 = \{G_1, G_2, \dots, G_n\}$, *construct an EG-graph, DG,* $DG = (V, E)$, *such that* $\Gamma_1 \subseteq l_V(V)$.

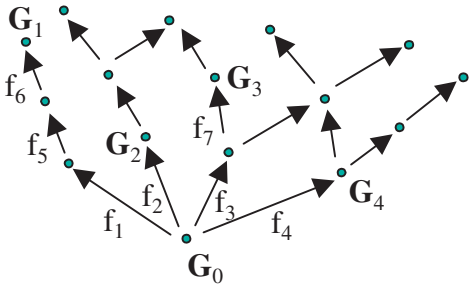

Fig. 3. An evolutionary genome graph.

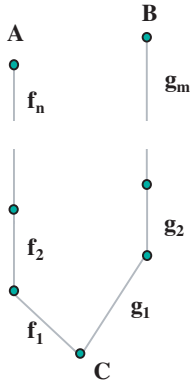

Fig. 4. The closest common ancestor of two genomes, ***A*** and ***B***

Next, given an EG-graph, DG, $DG = (V, E)$, in order to be able to compare a pair of genomes with respect to the set of transformations $l_E(E)$, one needs to specify a similarity measure

$$\mu_{DG} : l_V(V) \times l_V(V) \rightarrow \mathbb{R}^+.$$

In *evolutionary* approaches, one of the basic and most natural ways to introduce such a measure is by means of the evolutionary paths, i.e., by means of the sequence of transformations to the two genomes from their closest common ancestor (Figure 4). Since the set $l_E(E)$ of transformations may vary, depending on the EG-graph, we obtain the family of similarity measures:

$$M = \{\mu_{DG} | DG = (V, E) \text{ is an EG-graph}\}.$$

Having specified a similarity measure on a set of genomes, we can formulate an important, although computationally more complex, problem (which is an analogue of the corresponding optimal genome rearrangement problem).

PROBLEM 2. *Given a finite set of genomes,* $\mathbf{\Gamma}_1 = \{\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_n\}$, *and a family of similarity measures,* $M = \{\mu_{DG}\}$, *construct an EG-graph,* DG, $DG = (V, E)$, *such that* $\mathbf{\Gamma}_1 \subseteq l_V(V)$ *and at which the minimum of the following function (the mean value) is achieved over all EG-graphs with* $\mathbf{\Gamma}_1 \subseteq l_V(V)$*:*

$$\rho_{DG}(\Gamma_1) = \sum_{G_i, G_j \in \Gamma_1} \mu_{DG}\left(G_i, G_j\right)/n.$$

3 A NOVEL GENOME DEVELOPMENT MODEL

In this section, we introduce a novel evolution-based framework for genome development. First, we consider the case of two basic types of transformations, reversals and ins-transformations, and discuss their relationships with gen-transformations. Then, we introduce a similarity measure based on these basic types of transformations. Finally, we briefly discuss a new concept—the context of transformation—and show how it can be encapsulated in the model.

3.1 An evolution-based genome development model

Consider a set of genes $\boldsymbol{S} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_n\}$, i.e., a set of strings over the alphabet $\Sigma_0 = \{A, C, G, T\}$. We now specify some biologically reasonable restrictions we impose on the above set $\mathbf{\Gamma}$ of genomes (Definition 1) and EG-graphs (Definition 12). Set $\mathbf{\Gamma}$ of genomes is now restricted to those genomes that do not have repeated genes:

$$\mathbf{\Gamma} = \{\boldsymbol{G} | \boldsymbol{G} = \left(g_{i_1}, g_{i_2}, \ldots, g_{i_k}\right) \quad \text{and} \quad i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, n\}, i_1 \neq i_2 \neq \ldots \neq i_k\}.$$

We also specify a particular class of the corresponding EG-graphs (Definition 12). Given a finite set of genomes $\mathbf{\Gamma}_1 \subseteq \mathbf{\Gamma}$, the restrictions on an EG-graph DG, $DG = (V, E)$, for $\mathbf{\Gamma}_1$ are specified as follows:

(1) $\forall f \in l_E(E)$, f is either a reversal or an insertion

(2) $\forall \boldsymbol{G} \in \mathbf{\Gamma}_1$, $\forall$ genome development of $\boldsymbol{G}_0$ terminating in $\boldsymbol{G}$, $\forall g \in \boldsymbol{S}$, such that g is present in genome $\boldsymbol{G}$, g cannot be reversed twice (back to original form) by any of the reversals from the genome development

(3) if V_S is the set of sink nodes (Def. 11) of DG, then $l_V(V_S) = \mathbf{\Gamma}_1$.

Note that the additional restrictions on the EG-graph, in particular, allow us to avoid the case where a reversal transformation can be applied infinitely many times for a genome. Moreover, the following lemma is valid:

LEMMA 1. *With the above restrictions, the connected directed labeled graph* $DG = (V, E)$ *is also an acyclic graph.*

The above Lemma implies two related and very important properties: first, the genome can never return to one of its previous states, and, as a consequence of this, second, if one allows copying (that is, insertion) of each of the genes only once (in a more general model, several times), then the process of construction of all possible genomes is finite.

The model specified above has strong connections with the traditional genome transformations. Namely, it is not difficult to obtain the following results. (Henceforth, when necessary, we will denote a transformation f *of* a sequence of genes $\boldsymbol{X}$ as f_X.)

LEMMA 2. *Let* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *be two genomes,* $\boldsymbol{A}, \boldsymbol{B} \in \boldsymbol{\Gamma}$, *and let genome* $\boldsymbol{C}$ *be their closest common ancestor. Then:*

(1) if $\boldsymbol{B} = f_X(\boldsymbol{A})$, *where* f_X *is a transposition of a sequence* $\boldsymbol{X}$ *of genes, then there exist two insertions of this sequence of genes* f'_X *and* f''_X, *such that* $\boldsymbol{A} = f'_X(\boldsymbol{C})$, $\boldsymbol{B} = f''_X(\boldsymbol{C})$;

(2) if $\boldsymbol{B} = f(\boldsymbol{A})$, *where* f *is a reversal, then* $\boldsymbol{A} = \boldsymbol{C}$.

In other words, those genomes that are related to each other in a traditional model by either a reversal or a transposition, are also related in our model, but now via the closest ancestor and the corresponding gor- or ins-transformations. This partly explains the need for an evolutionary similarity measure for genomes to be based on the transformations leading to these genomes.

Next, we introduce a similarity measure for the above model. Before specifying the similarity measure formally, we want to assign a weight to each of the reversal and insertion transformations used in the corresponding EG-graph. There are many ways to assign a weight to a genome transformation. One of the standard ways is considering it with the size of the inserted, reversed, or transposed substring. In our model, we will use the following weighting scheme.

DEFINITION 13. For any reversal or insertion f_A of a sequence of genes $\boldsymbol{A}$, the **weight** of this transformation is defined as $w(f_A) = |\boldsymbol{A}|$, where $|\boldsymbol{A}|$ is the length of $\boldsymbol{A}$ (see Definition 1).

Having assigned the weights to transformations, we can choose the following similarity measure.

DEFINITION 14. Given an EG-graph DG, let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two genomes, $\boldsymbol{A}, \boldsymbol{B} \in \boldsymbol{\Gamma}$, and let genome $\boldsymbol{C}$ be their closest common ancestor. Moreover, let $\boldsymbol{A} = f_n \circ f_{n-1} \circ \ldots \circ f_1(\boldsymbol{C})$, $\boldsymbol{B} = g_m \circ g_{m-1} \circ \ldots \circ g_1(\boldsymbol{C})$ (see Figure 4). Then, the **similarity measure** is defined as

$$\mu_{DG}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{n} w\ (f_i) + \sum_{i=1}^{m} w\ (g_i).$$

Thus, having specified the class of EG-graphs, weighting scheme, and the similarity measure, it becomes possible to introduce the relationships among the genomes. Namely, given a set of genomes $\boldsymbol{\Gamma}_1$, we can, first, construct a particular EG-graph, thus specifying the particular mutation pathways that lead to each of the genomes in $\boldsymbol{\Gamma}_1$. Then, any genome in $\boldsymbol{\Gamma}_1$ can be compared with any other genome in $\boldsymbol{\Gamma}_1$ and their similarity measure, in terms of similar transformations occurring in the closest common ancestor of the above genomes, can be calculated.

3.2 The case of context-sensitive gene transformations

In this section, we discuss the concept of context-sensitive genome transformations. Why is such a concept useful? It is natural to assume that the process of a genome rearrangement (and development) should depend on the structure of genome being rearranged. In other words, the transformation for a genome may depend on a particular region of this genome (the case of a *local context*), or it may depend on the regions that are not close to each other and even on the entire genome (the case of a *global context*). Since the concept of a global context-sensitive transformation is very complex, in this introductory paper, we will discuss the local context-sensitive transformations only. We next consider the concept of the local context for both types of transformations, reversals and insertions.

DEFINITION 15. Given a set of genomes, $\boldsymbol{\Gamma}_1$, and a transformation f_X, where f_X is either reversal or ins-transformation, the **local context** of f_X is a pair of sequences $(\boldsymbol{C}_1, \boldsymbol{C}_2)$, $\boldsymbol{C}_1, \boldsymbol{C}_2 \in \Sigma^*$, such that

(1) if f_X is a reversal and $\boldsymbol{X}'$ is the reversed $\boldsymbol{X}$, then $\forall \boldsymbol{A}, \boldsymbol{B} \in \boldsymbol{\Gamma}_1$, where $\boldsymbol{B} = f_X(\boldsymbol{A})$,

$$\boldsymbol{A} = \boldsymbol{A}^1 \circ \boldsymbol{C}_1 \circ \boldsymbol{X} \circ \boldsymbol{C}_2 \circ \boldsymbol{A}^2$$

and

$$\boldsymbol{B} = \boldsymbol{B}^1 \circ \boldsymbol{C}_1 \circ \boldsymbol{X}' \circ \boldsymbol{C}_2 \circ \boldsymbol{B}^2,$$

where $\boldsymbol{A}^1, \boldsymbol{A}^2, \boldsymbol{B}^1, \boldsymbol{B}^2 \in \Sigma^*$;

(2) if f_X is an ins-transformation, then $\forall \boldsymbol{A}, \boldsymbol{B} \in \boldsymbol{\Gamma}_1$, where $\boldsymbol{B} = f_X(\boldsymbol{A})$,

$$\boldsymbol{A} = \boldsymbol{A}^1 \circ \boldsymbol{C}_1 \circ \boldsymbol{C}_2 \circ \boldsymbol{A}^2$$

and

$$\boldsymbol{B} = \boldsymbol{B}^1 \circ \boldsymbol{C}_1 \circ \boldsymbol{X} \circ \boldsymbol{C}_2 \circ \boldsymbol{B}^2,$$

where $\boldsymbol{A}^1, \boldsymbol{A}^2, \boldsymbol{B}^1, \boldsymbol{B}^2 \in \Sigma^*$.

Note that $\boldsymbol{C}_1$ and/or $\boldsymbol{C}_2$ can be the null string. In the case when both $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, are null strings, the corresponding transformation f_X is said to be **context-free**.

How does the introduction of context affect the similarity measure? There are different ways to redefine the similarity measure to reflect the presence of a context. It goes without saying that the context-sensitive transformation, when applied to some genome, acts more discriminatively than the 'same' transformation but without any context. Therefore, after the application of a context-free operation f_X, the resulting genome $\boldsymbol{B} = f_X(\boldsymbol{A})$ might be considered *farther* away from the original genome $\boldsymbol{A}$ than the one obtained by application of the 'same' transformation f_X but with non-empty context. In other words, in view of the additive nature of the similarity measure, the weight of a context-free transformation should be larger than that of the same transformation but with a non-empty context. Below, we give one of the possible ways to define such weighting scheme for context-sensitive transformations.

DEFINITION 16. Let f_X be a transformation and $(\boldsymbol{A}, \boldsymbol{B})$ be its context. Then, the **context-sensitive** weight of transformation f_X is defined as

$$w_{CS}(f_X) = w_{CF}\frac{|X|}{|X| + |A| + |B|},$$

where w_{CF} is a 'standard' weighting scheme for a (context-free) transformation introduced in Definition 13.

The weighting scheme in Def. 16 has two important features. First of all, for any context-free transformation f_X, its context-sensitive weight, $w_{CS}(f_X)$, is equal to the 'standard' weight of f_X, and the bigger the context of f_X, the smaller its context-sensitive weight. The latter, as was already mentioned above, can be explained by the fact that the bigger the context of a transformation, the more specific this transformation (when applied to a genome), and thus, the resulting new genome should be closer to its ancestor than the one obtained by applying the context-free version of the same transformation f_X. Second, when solving Problem 2, defined above, a smaller weight of a context-sensitive transformation makes it preferable, since the similarity measure between two genomes based on context-sensitive transformations will also be smaller in comparison with the same measure based on the same transformations but without the contexts.

4 IMPLEMENTATION: GENOME REARRANGEMENT USING TRANSPOSITIONS ONLY

In this section, we discuss how to reconstruct an EG-graph, given a set of genomes. We consider a basic type of rearrangement that uses only transpositions. The EG-graph reconstruction algorithm, described in this section, uses the idea of multiple longest common subsequence (see Definition 5).

4.1 Some basic ideas and assumptions

The basic, and computationally the simplest, is the case when we restrict ourselves to the set of genomes Δ, $\Delta \subseteq \Gamma$, possessing the property that any of its genomes can be obtained from some other of its genomes by applying a finite set of transpositions of genes from S. In an evolution-based model, this means that the same set of genes is to be inserted (possibly in a different order) by the corresponding ins-transformations into the protogenome. This assumption results in the following lemmas:

LEMMA 3. *Each of genomes from* $\boldsymbol{\Delta}$ *consists of the same number of genes from* $\boldsymbol{S}$.

LEMMA 4. *Given a finite set of genomes* $\boldsymbol{\Gamma}_1 \subseteq \boldsymbol{\Delta}$, *let DG, DG* $= (V, E)$, *be any EG-graph for* $\boldsymbol{\Gamma}_1$ *(see Section 3.1). Then, for any set* $V_0 \subseteq V$, *the MLCS of* V_0 *will contain at least one gene from* $\boldsymbol{S}$.

The last lemma follows from the previous one and allows us to rely on the concept of MLCS in the algorithm discussed next.

4.2 Algorithm

The algorithm outputs one of the possible EG-graphs, $DG = (E, V)$, given a set of genomes $\boldsymbol{\Gamma}_1$ composed of genes from a set $\boldsymbol{S}$. As a preprocessing stage, to obtain a string representation for each of the circular genomes, i.e., to find a common gene along which each of the given circular genome will be cut, the following simple algorithm is applied. For each of the genes, all of the (circular) genomes are cut along this gene and for the resulting strings the MLCS is constructed. The final cut is determined by choosing genome cuts associated with a gene for which the MLCS found is the longest one (ties are broken arbitrarily). The algorithm consists of two parts. In the first part, the common ancestor of all genomes in $\boldsymbol{\Gamma}_1$ is constructed. In the second (main) part, the transformations leading to each of the genomes are consecutively extracted. The pseudocode of the algorithm is presented below (// . . . // marks the comments).

Algorithm EG-graph

Input: $\boldsymbol{\Gamma}_1, |\boldsymbol{\Gamma}_1| = N$
Output: $DG = (V, E)$, where $V = \{v_1, v_2, \ldots, v_K\}$ is the set of labeled vertices and $E = \{e_1, e_2, \ldots, e_M\}$ in the set of labeled edges.

// **1. Construct ancestor** //
Ancestor $= v_0 =$ MLCS($\boldsymbol{\Gamma}_1$)
// **2. Construct transformations** //
cur_level $=$ V $= \{v_0\}$;
level $= 0$; cur_$G_v(v_0) = \{1, 2, \ldots, N\}$;
// *contains indices of all the genomes* //
While cur_level is not empty do

```
{ For all vertices v in cur_level
 {New_level = ∅;
  // cur_G_v: indices of all current genomes corresponding to a current vertex from Cur_level //
  cur_S = MLCS(cur_G_v);
  cur_Tr = Transform(S);
  // Transforms the MLCS cur_S to the sequence of ins-transformations, cur_Tr //
  For each transformation f in cur_Tr
  {(V_1, E_1, cur_G_v) = Children_v(v, f, cur_G_v, V, E);
   // Returns new vertices (V_1) together with edges (E_1), connecting vertex v and new vertices, based on the transformation f and the set of genomes corresponding to vertex v, cur_G_v, plus the set of indices Cur_G_v(w) corresponding to genomes that are children of the genome corresponding to a vertex w in V_1//
  V = V ∪ V_1; E = E ∪ E_1;
  } // End For //
 New_level = New_level ∪ V_1;
 // Adds to a new level the vertices, corresponding to the last transformation in the sequence cur_Tr//
 } //End For all vertices v //
 cur_level = new_level;
 level = level + 1;
} //End While//
END //Algorithm//
```

The algorithm's pseudocode presented above depends on two basic subroutines. While the first subroutine, MLCS( ), is discussed in Hakata and Imai (1998), the second one, Children_v ( ), can be described as follows. Given: a vertex v_A, corresponding to a genome $\boldsymbol{A}$, the set of indices cur_G_v (v) of those genomes in $\boldsymbol{\Gamma}_1$ which have $\boldsymbol{A}$ as a common ancestor, and an ins-transformation f; the subroutine Children_v ( ) constructs

(1) the set of vertices $V_1 = \{v_{A1}, v_{A2}, \ldots, v_{Ad}\}$ corresponding to the set $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_d\}$ of immediate descendants of $\boldsymbol{A}$ obtained by applying f to $\boldsymbol{A}$;

(2) the set of edges E_1, each of which is labeled by f, connecting each vertex in V_1 with v_A;

(3) for each of $\boldsymbol{A}_i$—the set of indices $cur\_G_v(\boldsymbol{A}_i)$ corresponding to those genomes in $\boldsymbol{\Gamma}_1$ which have $\boldsymbol{A}_i$ as a common ancestor.

4.3 Example

This example is presented for illustrative purposes only. However, it captures all the necessary aspects of the algorithm as well as allowing to comparison of the results of 'traditional' genome rearrangement with those for the evolutionary genome rearrangement represented by the corresponding EG-graph. Figure 5 and Figure 6 present the initial data and the resulting EG-graph, respectively.

$$\begin{aligned}
&\mathrm{G}_1 = \mathrm{D\ A\ B\ E\ C\ F\ G\ H};\\
&\mathrm{G}_2 = \mathrm{A\ D\ G\ B\ C\ F\ H\ E};\\
&\mathrm{G}_3 = \mathrm{A\ G\ B\ D\ C\ F\ E\ H};\\
&\mathrm{G}_4 = \mathrm{G\ A\ B\ D\ C\ F\ E\ H};\\
&\mathrm{G}_5 = \mathrm{E\ G\ C\ A\ F\ D\ B\ H};\\
&\mathrm{G}_6 = \mathrm{C\ E\ A\ F\ B\ G\ H\ D}.
\end{aligned}$$

Fig. 5. The input data, $\boldsymbol{\Gamma}_1$, consisting of 6 genomes composed of the following genes $\{A, B, \ldots, H\}$.

4.4 Complexity

To estimate the computational time complexity of the algorithm, we use the following result.

LEMMA 5. *Suppose* $|\boldsymbol{\Gamma}_1| = d$, $|\boldsymbol{S}| = s$. *Then:*

1. *The algorithm traverses the EG-graph only once;*
2. *For each level, the MLCS algorithm is performed no more than* $d/2$ *times;*
3. *There are no more than* s *levels in an EG-graph;*
4. *Transform ( ) is O(dL), where*

$$L = \max\{|\boldsymbol{G}_1|, |\boldsymbol{G}_2|, \ldots, |\boldsymbol{G}_d|\};$$

5. *In the main algorithm, Children ( ) is* $O(ds)$.

Based on the above Lemma, it is not difficult to estimate the time complexity of the EG-graph algorithm. Namely, let $O(T)$ be the time complexity for MLCS subroutine. Then the following result is true.

THEOREM 1. *The time complexity of the algorithm EG-graph is O(sd (T + L)).*

Finally, we note that the following estimation of the time complexity of the MLCS algorithm, based on the dominant point approach (Hakata and Imai, 1998), points to a substantial advantage of the latter approach as compared to the known dynamic programming approaches for solving this problem.

THEOREM 2 (HAKATA AND IMAI, 1998). *The MLCS problem for* $d (d \geqslant 3)$ *strings of length n can be solved in time*

$$O(nsd + |D|sd(\log^{d-3} n + \log^{d-2} s)),$$

where $|D|$ *is the size of the set of all dominant positions.*

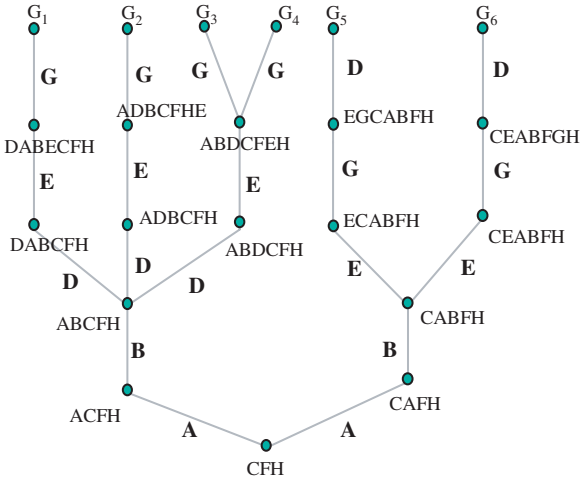

Fig. 6. An EG-graph for the set of genomes Γ_1 in Figure 5.

N	Organism	Group	Our genome cod
1	Albinaria coerulea	MOLluscs	(FAEOILKCDJBHG)
2	Asterina pectinifera	ECHinoderms	(JCDFGOBAHEILK)
3	Drosophila yakuba	ARThropod	(BHILKJCFDEGOA)
4	Homo sapiens	CHOrdat	(ABHILKJCEDFGO)
5	Katharina tunicata	MOLluscs	(HIKLFDEOGAJCB)
6	Lumbricus terrestri	ANNeli	(HILJGOKFEDACB)

Our abbreviations for the genes

ND1	**A**	ND4	**D**	ND6	**G**	COX3	**J**	CYTB	**O**
ND2	**B**	ND4L	**E**	COX1	**H**	ATP6	**K**		
ND3	**C**	ND5	**F**	COX2	**I**	ATP8	**L**		

Fig. 7. Mitochondrial genomes and their assumed monophyletic groupings used in our experiments (adapted from (Blanchette *et al.*, 1999), Table 1).

5 EXPERIMENTAL RESULTS

Selected results of our experiments are shown in Figures 7–9. Note that under the assumed constraints the corresponding EG-graph must be a tree. The resemblance of our tree with those shown in Figure 8 is quite apparent. However, instead of being an unrooted tree, the resulting EG-graph is always a rooted tree whose root is a common ancestor of all six genomes and encapsulates all their common conserved parts.

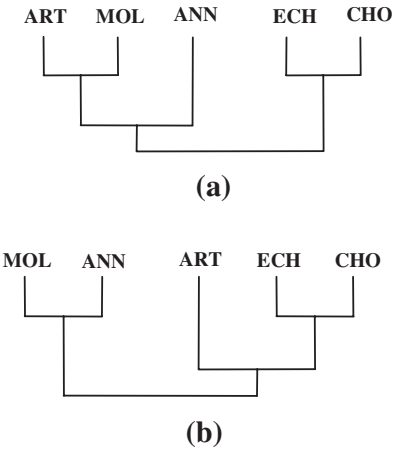

Fig. 8. Two alternative evolutionary trees for the species in Figure 7(a) currently most widely accepted view (b) a minimal breakpoint tree (adapted from Blanchette *et al.* (1999, Figures 1, 4)).

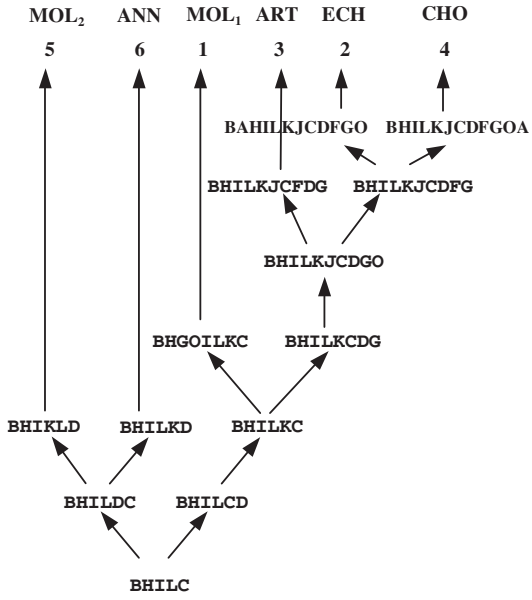

Fig. 9. The constructed evolutionary genome graph for the species in Figure 7.

6 DISCUSSIONS AND FUTURE RESEARCH

In this paper, we introduced an alternative, evolution-based, approach to the study of genome rearrangements. Within the approach, the development of genomes can be represented by a special directed graph, called an EG-graph, with labeled edges and vertices. One of the main advantages of the evolution-based approach is that it allows one to reconstruct other than just binary relations among the genomes: e.g., an EG-graph can represent three genomes that have one closest common ancestor. In order

to represent such relationships, we relied on the idea of a multiple longest common subsequence. Although the classical dynamic programming methods for computing MLCS can be used, practically, for the case of only two or three strings, the dominant-points based method allows one to obtain a MLCS for a much larger set of genome sequences.

As far as the future research directions are concerned, there are many that one can choose to follow. First of all, as the next step, one can consider the reconstruction of an EG-graph for more complex models of genome rearrangement, e.g., models based on the following genome transformations:

(1) *transpositions and reversals of genes*
(2) *transpositions and insertions of genes*

These models are not obvious and need careful study.

Next, one can consider the computationally more complex problem of reconstructing an optimal EG-graph (see Problem 2 in Section 2.B).

Another direction is the reconstruction of the context-sensitive transformations. One of the possible approaches, when the optimal EG-graph is not necessary, is to search for the context of transformations on the basis of the context-free EG-graph. Finally, one can consider a model of genome rearrangement with the presence of noise, i.e., point mutations. The weighting scheme should take this fact into consideration in such a way that the presence of some point mutations affects the similarity measure between the two genome sequences.

ACKNOWLEDGEMENTS

We would like to thank Patricia Evans for suggesting to the first author the genome rearrangement problem as an interesting topic (for the appplication of the new model) as well as the source for the data.

REFERENCES

Blanchette,M., Kunisawa,T. and Sankoff,D. (1999) Gene order breakpoint evidence in animal mitochondrial phylogeny. *J. Mol. Evol.*, **49**, 193–203.

Dobzhansky,T. (1973) Nothing in biology makes sense except in the light of evolution. *The American Biology Teacher*, **35**, 125–129.

El-Mabrouk,N. (2001) Reconstructing an ancestral genome using minimum segments duplications and reversals. *J, Comput. Syst. Sci.,* special issue on computational molecular biology, to appear.

Goldfarb,L., Golubitsky,O. and Korkin,D. (2001) What is a structural representation? Submitted to the special issue of *Pattern Recognition* on the ETS model. In Goldfarb,L. (ed.),

Goldfarb,L. and Korkin,D. (2001) Molecular classes and the structural representation: Towards theoretical foundations and unification of cheminformatics and chemistry. Submitted to the special issue of *Pattern Recognition* on the ETS model.

Hakata,K. and Imai,H. (1998) Algorithms for the longest common subsequence problem for multiple strings based on geometric maxima. *Optimiz. Meth. Software*, **10**, 233–260.

Hannenhalli,S, Chappey,C., Koonin,E. and Pevzner,P. (1995) Genome sequence comparison and scenarios for gene rearrangements: a test case. *Genomics*, **30**, 299–311.

Sankoff,D. and Blanchette,M. (1998) Multiple genome rearrangement and breakpoint phylogeny. *J. Comput. Biol.*, **5**, 555–570.

Sankoff,D. and Blanchette,M. (1999) Probability models for genome rearrangement and linear invariants for phylogenetic inference. *RECOMB*, 302–309.

Sankoff,D., Leduc,G., Antoine,N., Paquin,B., Land,B.F. and Cedergren,R. (1992) Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proc. Natl Acad. Sci. USA*, **89**, 6575–6579.

Sankoff,D., Sundaram,G. and,J.Kececioglu (1996) Steiner points in the space of genome rearrangements. *Int. J. Foundations Comput. Sci.*, **7**, 1–9.