BIOINFORMATICS ORIGINAL PAPER Vol. 25 no. 7 2009, pages 853–860
doi:10.1093/bioinformatics/btp070

Genome analysis

Accurate identification of orthologous segments among multiple genomes

Tsuyoshi Hachiya[1,*], Yasunori Osana[2], Kris Popendorf[1] and Yasubumi Sakakibara[1]

[1]Department of Biosciences and Informatics, Keio University and [2]Department of Computer and Informatics Science, Seikei University, Japan

Received on October 6, 2008; revised on January 28, 2009; accepted on January 29, 2009

Advance Access publication February 2, 2009

Associate Editor: Martin Bishop

ABSTRACT

Motivation: The accurate detection of orthologous segments (also referred to as syntenic segments) plays a key role in comparative genomics, as it is useful for inferring genome rearrangement scenarios and computing whole-genome alignments. Although a number of algorithms for detecting orthologous segments have been proposed, none of them contain a framework for optimizing their parameter values.

Methods: In the present study, we propose an algorithm, named OSfinder (Orthologous Segment finder), which uses a novel scoring scheme based on stochastic models. OSfinder takes as input the positions of short homologous regions (also referred to as anchors) and explicitly discriminates orthologous anchors from non-orthologous anchors by using Markov chain models which represent respective geometric distributions of lengths of orthologous and non-orthologous anchors. Such stochastic modeling makes it possible to optimize parameter values by maximizing the likelihood of the input dataset, and to automate the setting of the optimal parameter values.

Results: We validated the accuracies of orthology-mapping algorithms on the basis of their consistency with the orthology annotation of genes. Our evaluation tests using mammalian and bacterial genomes demonstrated that OSfinder shows higher accuracy than previous algorithms.

Availability: The OSfinder software was implemented as a C++ program. The software is freely available at http://osfinder.dna.bio.keio.ac.jp under the GNU General Public License.

Contact: hacchy@dna.bio.keio.ac.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

The term *orthologous segment* is defined as a set of genomic segments in different organisms descended from a common ancestor without large rearrangements (Dewey *et al*., 2006). The accurate detection of orthologous segments is essential for the following: inferring rearrangement-based phylogenies (Bourque *et al*., 2004; Tesler, 2002), reconstructing ancestral genomes (Bourque *et al*., 2005; Ma *et al*., 2006; Murphy *et al*., 2005), computing whole-genome alignments (Dewey *et al*., 2006; Gibbs *et al*., 2004; Waterston *et al*., 2002), identifying orthologous genes (Hubbard *et al*., 2005; Zheng *et al*., 2005) and detecting non-coding functional elements such as regulatory elements (Frazer *et al*., 2004). The problem of identifying orthologous segments is referred to as *orthology mapping* (Dewey *et al*., 2006).

The general strategy of orthology mapping is as follows: (i) take as input the positions of short homologous regions (also referred to as anchors) detected among the set of genomes under comparison. Homologous genes or bidirectional local sequence matches are commonly used as anchors. (ii) Detect *collinear* anchors which are distributed in the same order and have the same orientation. (iii) Connect closely located collinear anchors. (iv) Output connected components as orthologous segments.

One difficulty in orthology mapping concerns the fact that a non-negligible fraction of input anchors are non-orthologous rather than orthologous. In the case where the anchors are homologous gene pairs, paralogous gene pairs can be detected as non-orthologous anchors. In the case where the anchors are homologous sequence matches, repeat sequences can be detected as non-orthologous anchors. Conservation scores for anchors and distances between adjacent anchors constitute important features for distinguishing between orthologous genomic regions, in which anchors are distributed densely in off-diagonal positions, and non-orthologous genomic regions, in which anchors are distributed randomly. Existing orthology-mapping programs implicitly filter out non-orthologous anchors in the process of identifying orthologous segments. Pevzner and Tesler (2003) proposed the GRIMM-Synteny algorithm, which chains every pair of anchors if the distance between the two anchors is less than a certain distance threshold, removes chained components if the size of the components is smaller than a certain size threshold, and reports the remaining components as synteny blocks. In order to avoid detecting non-orthologous genomic regions as synteny blocks, it is important to set these two threshold values appropriately. However, GRIMM-Synteny does not provide a framework for determining optimal threshold parameters.

ADHoRe (Vandepoele *et al*., 2002) and SyMAP (Soderlund *et al*., 2006) are tools for detecting orthologous segments that are capable of automatically determining the distance threshold value. These tools perform detection by starting with a small value of the distance threshold and increasing it iteratively. This iteration process yields an appropriate distance threshold value which maximizes the length of the orthologous segments while retaining satisfactory quality (Soderlund *et al*., 2006). Both ADHoRe and SyMAP define

*To whom correspondence should be addressed.

the quality of the orthologous segments on the basis of the diagonal properties of the anchor positions. For a series of anchors, the anchor positions are fitted with a linear regression model, and the quality is computed as the coefficient of determination. Although these programs can determine the distance threshold automatically, they require a quality threshold to be set manually.

In addition to the above programs, other orthology-mapping algorithms, including DAGChainer (Haas *et al*., 2004), AXTCHAIN (Kent *et al*., 2003), DiagHunter (Cannon *et al*., 2003), FISH (Calabrese *et al*., 2003) and Cinteny (Sinha *et al*., 2007), also require the manual setting of key threshold parameters. Since these thresholds can affect the accuracy of orthology-mapping programs and are difficult to set manually, a more sophisticated approach for determining their parameter values is needed. Furthermore, the vast majority of existing orthology-mapping programs are applicable only in pairwise genome comparisons. Thus, the capability to compare multiple genomes is also desired.

In the present study, we propose an orthology-mapping algorithm, named OSfinder (Orthologous Segment finder), which uses a novel scoring scheme based on stochastic models. OSfinder explicitly discriminates orthologous anchors from non-orthologous anchors by using Markov chain models, which represent respective geometric distributions of lengths of orthologous and non-orthologous anchors. Such stochastic modeling makes it possible to optimize parameter values by maximizing the likelihood of the input dataset, and to automate the setting of the optimal parameter values. Moreover, OSfinder can be applied not only in pairwise genome comparisons, but also in multiple genome comparisons. There is no limit to the number of genomes which can be compared with our software.

2 METHODS

2.1 Detecting anchors

The term *anchor* generally refers to well-conserved short regions between two or multiple genomes, and is biologically defined as a group of homologous genes or a set of homologous sequence matches. In our experiments, anchors were detected between mammalian genomes and between bacterial genomes. Mammalian genomes included those of human, chimpanzee, macaque, mouse, rat, dog and opossum, and bacterial genomes included those of *Mycobacterium tuberculosis* (Mtu), *M.bovis* (Mbo), *M.leprae* (Mle) and *M.avium* (Mpa). Since the method for detecting anchors can affect the accuracy of orthology-mapping programs, two methods for detecting anchors were taken into account.

2.1.1 Homologous sequences Whole genome sequences of the seven mammals and the four bacteria were taken from the Ensembl genome browser (Hubbard *et al*., 2007) and the RefSeq database (Pruitt *et al*., 2007), respectively. When comparing two genomes x and x', the whole genome sequences of x and x' were input into Murasaki (Popendorf *et al*., 2007) with the repeat mask option. The genomic locations of the anchors were then output by Murasaki. After appropriate format transformation, the Murasaki output was used as input for the orthology-mapping programs. Both pairwise and multiple anchors can be computed by this work flow.

2.1.2 Homologous gene pairs Protein sequences encoded in the seven mammalian genomes and the four bacteria genomes were drawn from the Ensembl genome browser and the RefSeq database, respectively. When comparing two genomes x and x', all protein sequences encoded in genome x were compared with all protein sequences encoded in genome x' by using the BLASTP program (Altschul *et al*., 1990), and protein pairs whose E-values were $<10^{-100}$ were regarded as anchors. Then, pairs of gene IDs were transformed into pairs of genomic locations of the genes. The file containing the genomic positions of the anchors were used as input for the orthology-mapping programs. Only pairwise anchors can be computed by this work flow.

The statistics for the anchors detected between mammalian genomes are summarized in Supplementary Table S3.

2.2 Mathematical definitions

The OSfinder algorithm is based on the following mathematical definitions. Here, we denote the set of genomes under comparison as $\mathbf{G}$ and the set of pre-computed anchors as $\mathbf{a}$.

2.2.1 Properties of anchors The genomic position of an anchor a_i ($\in\mathbf{a}$) can be represented by four properties for each genome x ($\in\mathbf{G}$): chromosome ID ($a_i.chrom_x$), start position ($a_i.start_x$), end position ($a_i.end_x$) and strand information ($a_i.sign_x$) (Fig. 1a and b). We define that the values of $a_i.start_x$ and $a_i.end_x$ are positive integers, and represent coordinates in terms of forward strand positions in the chromosome $a_i.chrom_x$, where the first base in a chromosome is numbered 1. That is, for an anchor a_i on the reverse strand, the start and end positions of a_i are defined as the coordinates of the complementary region of a_i on the forward strand, and therefore these two values satisfy the condition $a_i.start_x < a_i.end_x$ regardless of the value of $a_i.sign_x$. Further, we assume that for the reference genome $\dot{x}$, the value of $a_i.sign_{\dot{x}}$ is '1' $^\forall a_i \in \mathbf{a}$. The value of $a_i.sign_x$ is '1' if the anchor region from the genome x is not inverted relative to the anchor region from the reference genome $\dot{x}$, and $a_i.sign_x = -1$ if the anchor region from the genome x is inverted relative to the anchor region from the reference genome $\dot{x}$. Note that the choice of the reference genome does not affect the *collinear* relation between the anchors.

2.2.2 Collinearity In comparative genomics, conservations that are distributed in the same order and have the same orientation are referred to as *collinear* conservations (Bennetzen and Ramakrishna, 2002; Song *et al*., 2002). Two anchors, a_i and $a_{i'}$, are *collinear* if the following conditions are satisfied:

$$\begin{aligned} &a_i.chrom_x = a_{i'}.chrom_x \\ &a_i.sign_x = a_{i'}.sign_x \\ &\begin{cases} a_i.end_x < a_{i'}.start_x & \text{when } a_i.sign_x = 1 \\ a_{i'}.end_x < a_i.start_x & \text{when } a_i.sign_x = -1, \end{cases} \end{aligned} \tag{1}$$

$^\forall x \in \mathbf{G}$. Let $a_i \prec a_{i'}$ denote the case where a_i and $a_{i'}$ satisfy the conditions shown in Equation (1).

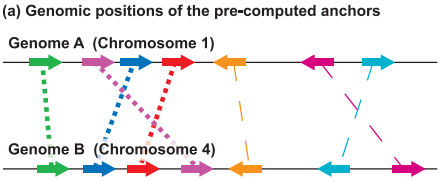
2.2.3 Anchor graph The collinear relation defines a partial order between the anchors. Since a partial order induces a directed acyclic graph (DAG), the collinear relations can be represented as a DAG (Fig. 1c). OSfinder constructs a DAG in which a node is an anchor and a directed edge is drawn from a_i to $a_{i'}$ if $a_i \prec a_{i'}$ and there is no anchor $a_{i''}$ satisfying $a_i \prec a_{i''} \prec a_{i'}$. We call this type of DAG *an anchor graph*.

2.2.4 Properties of edges The start and end positions of an edge e_j ($\in\mathbf{e}$) connecting two anchors (a_i and $a_{i'}$) are defined as follows:

$$\begin{aligned} e_j.start_x &\equiv \min\{a_i.end_x, a_{i'}.end_x\} + 1 \\ e_j.end_x &\equiv \max\{a_i.start_x, a_{i'}.start_x\} - 1. \end{aligned}$$

2.2.5 Length of anchors and edges The length of an anchor a_i and the length of an edge e_j are defined as follows:

$$\begin{aligned} a_i.length &\equiv \sum_{x\in\mathbf{G}} (a_i.end_x - a_i.start_x + 1) \\ e_j.length &\equiv \sum_{x\in\mathbf{G}} (e_j.end_x - e_j.start_x + 1). \end{aligned}$$



(b) Properties of the pre-computed anchors

Genome A				Genome B			
chrom	start	end	sign	chrom	start	end	sign
1	100	200	1	4	150	250	1
1	300	400	1	4	700	800	1
1	450	550	1	4	350	450	1
1	600	700	1	4	500	600	1
1	800	900	1	4	850	900	1
1	1100	1200	1	4	1350	1450	-1
1	1300	1400	1	4	1150	1250	-1

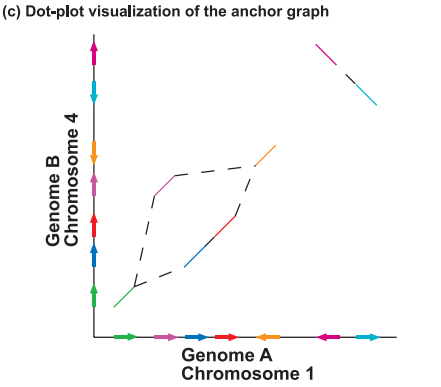


Fig. 1. These figures show a toy problem of detecting chains in which seven anchors are pre-computed between two genomes A and B. (**a**) The genomic positions of the seven anchors, in which colored arrows represent the anchors (either homologous genes or conserved sequences). In this figure, anchors located on the forward strand are depicted as right arrows, and anchors located on the reverse strand are depicted as left arrows. (**b**) The properties of the seven anchors, in which genome A is used as the reference genome. (**c**) The anchor graph visualized by using a dot plot, in which anchors are depicted by colored solid lines and edges are depicted by black broken lines. (**d**) This figure illustrates that chains (indicated by colored blocks) correspond to non-intersecting paths in the anchor graph.

2.2.6 Chains *Chains* are genomic segments in which anchors are distributed densely in off-diagonal positions. Chains correspond exactly to *non-intersecting* suboptimal paths in the observed anchor graph, where two paths are intersecting if their coordinate spans overlap with each other $^{\forall}x \in \mathbf{G}$ (Fig. 1d).

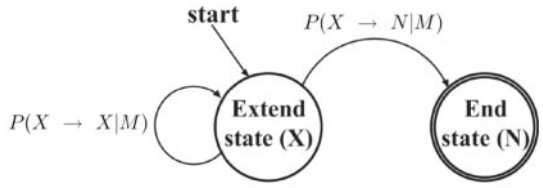


Fig. 2. Markov chain models in OSfinder.

2.2.7 Orthologous segments Orthologous segments are defined as genomic segments descended from a common ancestor without large rearrangements. An orthologous segment corresponds to a sequence of collinear chains, where the collinearity of chains is defined in a similar manner to that of anchors (Supplementary Fig. S1).

2.3 OSfinder algorithm

In order to identify orthologous segments accurately, orthology-mapping algorithms should be able to distinguish between orthologous and non-orthologous anchors. For this purpose, OSfinder introduces a set of hidden variables named *labels*. A label is assigned to an anchor or an edge, and its value is either '+' or '−', where '+' represents an orthologous anchor or edge and '−' represents a non-orthologous one.

The likelihood for the observed anchor graph is defined by two sets of variables, namely a set of model parameters **M** and a set of labels **L**. By computing the maximum likelihood solution for **M** and **L**, the respective length distributions of orthologous and non-orthologous anchors (edges) are fitted to geometric distributions defined by Markov chain models. The optimized model parameters not only determine the optimal length threshold for anchors (edges) which is used to discriminate between orthologous and non-orthologous anchors (edges), but also provide the score for anchors (edges) in the anchor graph. Based on the scores, non-intersecting suboptimal paths are efficiently extracted from the anchor graph by using a dynamic programming technique, and a set of chains is detected. Finally, a sequence of collinear chains is merged into an orthologous segment in order to fill large gap regions between collinear chains.

The overall algorithm of OSfinder is composed of the following steps. (i) take the genomic positions of the anchors as input. (ii) Construct an anchor graph. The definition of the likelihood for an anchor graph is described in Section 2.3.1. (iii) Compute the optimal values for labels and model parameters. Two optimization algorithms are described in Section 2.3.2. (iv) Extract non-intersecting suboptimal paths from the observed anchor graph and generate a set of chains. The description of the extraction algorithm can be found in Section 2.3.3. (v) Merge collinear chains and output the merged components as orthologous segments. The merge algorithm is described in Section 2.3.4.

2.3.1 Likelihood for an anchor graph OSfinder models the respective length distributions of orthologous and non-orthologous anchors (and edges) in the observed anchor graph by using Markov chain models. These models have two states, the extend state (X) and the end state (N), and two state transitions, $X \rightarrow X$ and $X \rightarrow N$ (Fig. 2). We denote the transition probability from state X to state X as $P(X \rightarrow X|M)$, and the transition probability from state X to state N as $P(X \rightarrow N|M)$, where M denotes a model. Note that $P(X \rightarrow X|M) + P(X \rightarrow N|M) = 1$. An anchor (or an edge) whose length is l indicates the transition sequence $(X \rightarrow X)^{l-1} \rightarrow N$. Thus, the likelihood for an anchor a_i and the likelihood for an edge e_j are defined by the following geometric distributions:

$$P(a_i|M) = P(X \rightarrow X|M)^{(a_i.length-1)} \times P(X \rightarrow N|M)$$

$$P(e_j|M) = P(X \rightarrow X|M)^{(e_j.length-1)} \times P(X \rightarrow N|M).$$

Next, we set up four Markov chain models, namely a model for representing orthologous anchors (M^{+}_{anchor}), a model for representing non-orthologous anchors (M^{-}_{anchor}), a model for representing orthologous edges (M^{+}_{edge}) and a model for representing non-orthologous edges (M^{-}_{edge}). OSfinder assumes that the average length of orthologous anchors is greater than the average length of non-orthologous anchors. This assumption is implemented in the constraint shown in Equation (2). Similarly, it is assumed that the average length of orthologous edges is shorter than the average length of non-orthologous edges. This assumption is implemented in the constraint shown in Equation (3).

$$P(X \to N|M^{+}_{\text{anchor}}) < P(X \to N|M^{-}_{\text{anchor}}) \tag{2}$$

$$P(X \to N|M^{+}_{\text{edge}}) > P(X \to N|M^{-}_{\text{edge}}). \tag{3}$$

Given a set of model parameters $\mathbf{M}$ and a set of labels $\mathbf{L}$, OSfinder defines the likelihood for an anchor a_i and the likelihood for an edge e_j as follows:

$$\begin{aligned} P(a_i|\mathbf{M},\mathbf{L}) &= \begin{cases} P(a_i|M^{+}_{\text{anchor}}) & \text{if } a_i.label \text{ is `+'}, \\ P(a_i|M^{-}_{\text{anchor}}) & \text{if } a_i.label \text{ is `}-\text{'}. \end{cases} \\ P(e_j|\mathbf{M},\mathbf{L}) &= \begin{cases} P(e_j|M^{+}_{\text{edge}}) & \text{if } e_j.label \text{ is `+'}, \\ P(e_j|M^{-}_{\text{edge}}) & \text{if } e_j.label \text{ is `}-\text{'}. \end{cases} \end{aligned} \tag{4}$$

Given a set of labels $\mathbf{L}$, let $\mathbf{a}^{+}$ be a set of anchors labeled as '+' and $\mathbf{a}^{-}$ be a set of anchors labeled as '−' ($\mathbf{a}=\mathbf{a}^{+}\cup\mathbf{a}^{-}$). Similarly, let $\mathbf{e}^{+}$ be a set of edges labeled as '+' and $\mathbf{e}^{-}$ be a set of edges labeled as '−' ($\mathbf{e}=\mathbf{e}^{+}\cup\mathbf{e}^{-}$). Then, given $\mathbf{M}$ and $\mathbf{L}$, the likelihood for an anchor graph $G=(\mathbf{a},\mathbf{e})$ is defined as follows:

$$\begin{aligned} &P(\mathbf{a},\mathbf{e}|\mathbf{M},\mathbf{L}) \\ &= \Pi_{a_i\in\mathbf{a}^{+}} P(a_i|M^{+}_{\text{anchor}}) \times \Pi_{a_{i'}\in\mathbf{a}^{-}} P(a_{i'}|M^{-}_{\text{anchor}}) \\ &\quad \times \Pi_{e_j\in\mathbf{e}^{+}} P(e_j|M^{+}_{\text{edge}}) \times \Pi_{e_{j'}\in\mathbf{e}^{-}} P(e_{j'}|M^{-}_{\text{edge}}). \end{aligned} \tag{5}$$

2.3.2 Optimization algorithms The parameter values in our Markov chain models are optimized so as to maximize the likelihood for the observed anchor graph. Let $\tilde{\mathbf{M}}$ denote a set of *optimal* model parameters and $\tilde{\mathbf{L}}$ denote a set of *optimal* labels. $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{L}}$ are defined by the following equation:

$$(\tilde{\mathbf{M}},\tilde{\mathbf{L}}) = \text{argmax}_{(\mathbf{M},\mathbf{L})} P(G|\mathbf{M},\mathbf{L}). \tag{6}$$

Global maximization algorithm: given a set of labels $\mathbf{L}$, the *conditionally optimal* parameter set $\hat{\mathbf{M}}_{\mathbf{L}}$ is defined as follows:

$$\hat{\mathbf{M}}_{\mathbf{L}} = \text{argmax}_{\mathbf{M}} P(G|\mathbf{M},\mathbf{L}). \tag{7}$$

The conditionally optimal model parameters can be calculated from the following equations (see Proof 1 in Supplementary Materials):

$$\begin{aligned} P(X \to N|M^{+}_{\text{anchor}}) &= \frac{|\mathbf{a}^{+}|}{\sum_{a_i\in\mathbf{a}^{+}} a_i.length} \\ P(X \to N|M^{-}_{\text{anchor}}) &= \frac{|\mathbf{a}^{-}|}{\sum_{a_{i'}\in\mathbf{a}^{-}} a_{i'}.length} \\ P(X \to N|M^{+}_{\text{edge}}) &= \frac{|\mathbf{e}^{+}|}{\sum_{e_j\in\mathbf{e}^{+}} e_j.length} \\ P(X \to N|M^{-}_{\text{edge}}) &= \frac{|\mathbf{e}^{-}|}{\sum_{e_{j'}\in\mathbf{e}^{-}} e_{j'}.length}. \end{aligned} \tag{8}$$

Note that $P(X \to X|M)$ can be calculated from $P(X \to N|M)$ easily.

From Equation (7), the maximization problem shown in Equation (6) can be restated as follows:

$$\begin{aligned} \max_{\mathbf{M},\mathbf{L}} P(G|\mathbf{M},\mathbf{L}) &= \max_{\mathbf{L}} \max_{\mathbf{M}} P(G|\mathbf{M},\mathbf{L}) \\ &= \max_{\mathbf{L}} P(G|\hat{\mathbf{M}}_{\mathbf{L}},\mathbf{L}). \end{aligned}$$

Thus, a naive method to maximize the likelihood for an anchor graph is to enumerate all possible label sets and to find the label set that maximizes $P(G|\hat{\mathbf{M}}_{\mathbf{L}},\mathbf{L})$. However, the computation of the naive method is infeasible in terms of computational costs since the number of all possible label sets is $2^{(|\mathbf{a}|+|\mathbf{e}|)}$. Our global maximization algorithm reduces the number of label sets for enumeration to $(|\mathbf{a}|+|\mathbf{e}|-2)$ without losing the ability to find the global optimum. See 'Global maximization method' in Supplementary Materials for details. The total computational complexity of the global maximization algorithm is $\mathbf{O}(|\mathbf{a}|^2+|\mathbf{e}|^2)$.

Local maximization algorithm: the computation of the global maximization algorithm is also infeasible when the number of anchors or the number of edges is extremely large (Supplementary Table S1). Thus, a fast learning algorithm whose computational complexity is $\mathbf{O}(|\mathbf{a}|+|\mathbf{e}|)$ is also implemented in OSfinder.

Given a set of model parameters $\mathbf{M}$, the *conditionally optimal* label set $\hat{\mathbf{L}}_{\mathbf{M}}$ is defined as follows:

$$\hat{\mathbf{L}}_{\mathbf{M}} = \text{argmax}_{\mathbf{L}} P(G|\mathbf{M},\mathbf{L}). \tag{9}$$

The conditionally optimal labels are given by Equation (10).

$$\begin{aligned} a_i.label &= \text{argmax}_{label\in\{+,-\}} P(a_i|M^{label}_{\text{anchor}}) \\ e_j.label &= \text{argmax}_{label\in\{+,-\}} P(e_j|M^{label}_{\text{edge}}) \end{aligned} \tag{10}$$

The following algorithm is capable of finding a local optimum.

1. Set the initial model parameters $\mathbf{M}^0$. See 'Initialization of model parameters' in Supplementary Materials for details.
2. For each step t ($1 \le t \le t_{max}$),
   a. Calculate the conditionally optimal labels by using the parameter values calculated at step $(t-1)$. In other words, $\mathbf{L}^t = \hat{\mathbf{L}}_{\mathbf{M}^{(t-1)}}$.
   b. Calculate the conditionally optimal model parameters by using the labels calculated at step t. In other words, $\mathbf{M}^t = \hat{\mathbf{M}}_{\mathbf{L}^t}$.
   c. Stop the iteration if $\mathbf{M}^{(t-1)} = \mathbf{M}^t$.
3. Report the parameter values obtained at the end of the above iteration.

It can be proven that the parameters identified by the algorithm locally maximize the likelihood for the observed anchor graph (see Proof 2 in Supplementary Materials). The default value for $t_{\max}$ is set at 100 in the current version of OSfinder. Although the fast algorithm calculates a local optimum rather than the global optimum, our computational experiments using mammalian genomes show that the accuracy of the local maximization algorithm is almost the same as that of the global maximization algorithm (Supplementary Table S2).

2.3.3 Chain extraction algorithm Given a set of optimal model parameters $\tilde{\mathbf{M}}$, the score for an anchor a_i and the score for an edge e_j are defined as log-odds of two likelihoods as follows:

$$a_i.score = \log \frac{P(a_i|M^{+}_{\text{anchor}})}{P(a_i|M^{-}_{\text{anchor}})} \tag{11}$$

$$e_j.score = \log \frac{P(e_j|M^{+}_{\text{edge}})}{P(e_j|M^{-}_{\text{edge}})}. \tag{12}$$

A path in an anchor graph corresponds exactly to a sequence of collinear anchors. Let $\mathbf{a}^{p_k}$ denote a set of anchors included in a path p_k, and $\mathbf{e}^{p_k}$ denote a set of edges included in p_k. In this case, the score for path p_k is defined by the following equation:

$$p_k.score = \sum_{a_i\in\mathbf{a}^{p_k}} a_i.score + \sum_{e_j\in\mathbf{e}^{p_k}} e_j.score.$$

The path with the highest score can be efficiently found by using a dynamic programming technique with the following recursive formula for the best path ending at anchor a_i:

$$\begin{aligned} &path\_score(a_i) \\ &= a_i.score + \max \begin{cases} \max_{a_{i'}\prec a_i} \{path\_score(a_{i'}) + e_{a_{i'}\to a_i}.score\} \\ 0 \end{cases}, \end{aligned} \tag{13}$$

where $e_{a_{i'} \to a_i}$ represents the edge drawn from anchor $a_{i'}$ to anchor a_i. After the optimization of the model parameters, the chain extraction algorithm in OSfinder detects non-intersecting suboptimal paths from the observed anchor graph. It recursively executes the following operations: (i) calculation of the list whose i-th element deposits the value of $path\_score(a_i)$ defined by Equation (13), (ii) detection of the highest scoring path by tracing back from the element which has the highest value of $path\_score(a_i)$ and (iii) removal of the anchors and edges which are intersecting with the extracted path, until there are no paths scoring higher than zero (Supplementary Fig. S2). The re-calculation of the list is essential for the detection of the next highest scoring path because the removal of the anchors and edges changes the scores in the list. We call the suboptimal paths *chains*.

2.3.4 Merge algorithm Given a set of chains, the merge algorithm in OSfinder performs the following operation on the basis of a user-defined parameter named *minimum segment length*:

1. Construct a DAG, where a node is a chain and a directed edge is drawn from a chain c_l to a chain $c_{l'}$ if $c_l \prec c_{l'}$ and there is no chain $c_{l''}$ satisfying $c_l \prec c_{l''} \prec c_{l'}$. We call the DAG *a chain graph*.
2. Sort the edges in the chain graph in the order of increasing edge length.
3. For each edge e_j in the sorted order ($1 \leq j \leq |\mathbf{e}|$)
   a. Check whether there exists any merged component such that its coordinate span overlaps with the coordinate span of the edge e_j for at least one genome x ($\in \mathbf{G}$), and its length is greater than the *minimum segment length*.
   b. If no such case exists, merge chains connected through the j-th edge.
4. Report merged components whose length is greater than the *minimum segment length* as orthologous segments.

The *minimum segment length* controls the resolution of the orthology mapping. A large value for this parameter is appropriate for analyzing macrorearrangements, and a small value for the parameter is appropriate for drawing detailed dot plots.

2.4 Evaluation criteria

Since it is impossible to observe the course of evolutionary history, the evaluation of orthology-mapping programs is restricted to simulation experiments (Calabrese *et al*., 2003; Cannon *et al*., 2003; Hampson *et al*., 2003) or assessment on the basis of the consistency of the target program with other orthology-mapping programs (Cannon *et al*., 2003). However, simulation experiments require the mutation models to generate virtual evolutionary histories, and therefore the evaluation results are inevitably biased with respect to the mutation models used in the experiment. In addition, examining the consistency with other programs is not an efficacious methodology for estimating the accuracy if the compared programs are based on similar approaches.

In this article, we estimate the accuracy of orthology-mapping programs on the basis of their consistency with the orthology annotation of genes (Fig. 3). Let $\mathbf{G}$ be the set of genomes under comparison, $\mathbf{s} = \{s_1, s_2, \ldots, s_{|\mathbf{G}|}\}$ be an orthologous segment (a set of segments from different genomes), and $\mathbf{g} = \{g_1, g_2, \ldots, g_{|\mathbf{G}|}\}$ be an orthologous gene group (a set of genes from different genomes), where s_x (g_x) is a segment (gene) from a genome x. Here, we assume that orthologous gene groups do not contain in-paralogs (Koonin, 2005; Remm *et al*., 2001) and that orthologous relationships are necessarily one-to-one. Then, we define that $\mathbf{g}$ is *consistent* with $\mathbf{s}$ if, for all $x \in \mathbf{G}$, the coding region of g_x overlaps with the coordinate span of s_x and the orientation of g_x is the same as that of s_x. We also define that $\mathbf{g}$ is *inconsistent* with $\mathbf{s}$ if $\mathbf{g}$ is not consistent with $\mathbf{s}$ and there exists a genome x in which the coding region of g_x overlaps with the coordinate span of s_x.

Given a set of orthologous gene groups and a set of orthologous segments, let N be the number of orthologous gene groups and N_C be the number of orthologous gene groups with which at least one orthologous segment is consistent. Let $C(\mathbf{s}^i)$ denote the number of orthologous gene groups which are consistent with an orthologous segment $\mathbf{s}^i$ and $I(\mathbf{s}^i)$ denote the number of orthologous gene groups which are inconsistent with an orthologous segment $\mathbf{s}^i$. Then, the *sensitivity* and the *specificity* are defined as follows:

$$sensitivity = \frac{N_C}{N}$$

$$specificity = \frac{\sum_i C(\mathbf{s}^i)}{\sum_i C(\mathbf{s}^i) + \sum_i I(\mathbf{s}^i)}$$

The F-score is defined as $2pr/p+r$, where p represents the specificity and r represents the sensitivity. We use the F-score as an indicator of accuracy.

In this article, the mammalian orthologous gene database SPEED (Vallender *et al*., 2006) was employed as a reliable source of orthology annotation data for mammalian genes. Regarding the evaluation of the results of bacterial genome comparisons, orthologous gene pairs were identified on the basis of BLAST reciprocal best hits (Tatusov *et al*., 1997, 2003). Orthologous groups of bacterial genes were calculated by using the method described in Vallender *et al*. (2006).

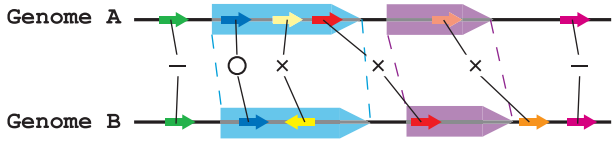


Fig. 3. Consistency and inconsistency between orthologous gene groups and orthologous segments. This figure shows the respective genomic locations of six pairs of orthologous genes (indicated by small arrows) and two pairs of orthologous segments (indicated by large arrows). The color of the arrows represents the orthologous relationship where pairs of orthologous genes or segments have the same color. '○' ('×') indicates that the gene pair is consistent (inconsistent) with a certain orthologous segment, and '−' indicates that the gene pair is neither consistent nor inconsistent with any orthologous segment.

3 RESULTS

3.1 Accuracy in pairwise genome comparisons

3.1.1 Comparison with DAGChainer and ADHoRe Anchors detected between pairwise genomes were input into DAGChainer (Haas *et al*., 2004), ADHoRe (Vandepoele *et al*., 2002) and OSfinder, where DAGChainer and ADHoRe were executed not only with the default parameter values, but also with optimized parameter values. The optimization for the DAGChainer and ADHoRe parameters was performed on the basis of a grid search in order to maximize the F-score. Two parameters in DAGChainer, the average expected distance between two orthologous anchors (-g option) and the maximum allowed distance between two anchors (-D option), were optimized. Two parameters in ADHoRe, the minimum r^2 value (r2_cutoff option) and the maximum distance between the anchors (max_dist option) were optimized. Since OSfinder automatically optimizes its parameter values, there was no need to perform grid searches.

Table 1 shows the accuracy of the three programs in the pairwise comparison of mammalian genomes. It is worth noting that OSfinder consistently achieved high F-scores (>85% on average), regardless of the anchor type. DAGChainer exhibited low F-scores when the anchors were homologous gene pairs (62.2% with a grid search), while ADHoRe exhibited extremely low F-scores both when the anchors were homologous sequences (47.1% with the default

Table 1. Accuracy in the pairwise comparison of mammalian genomes

Genomes	DAGChainer						ADHoRe						OSfinder		
	Default			Grid search			Default			Grid search			Default		
	Sn	Sp	*F*	Sn	Sp	*F*	Sn	Sp	*F*	Sn	Sp	*F*	Sn	Sp	*F*
Homologous sequences															
Human–chimpanzee	99.5	73.1	84.3	–	–	–	–	–	–	–	–	–	99.0	98.4	98.7
Human–macaque	99.1	77.1	86.7	–	–	–	–	–	–	–	–	–	97.9	96.9	97.4
Human–mouse	91.0	88.7	89.8	–	–	–	47.7	46.5	47.1	–	–	–	93.5	95.2	94.3
Human–rat	89.2	89.4	89.3	–	–	–	44.8	44.0	44.4	–	–	–	89.1	90.2	89.6
Human–dog	97.3	81.0	88.4	–	–	–	49.9	42.3	45.8	–	–	–	95.7	96.0	95.8
Human–opossum	50.8	93.8	66.0	–	–	–	48.4	53.8	50.9	–	–	–	76.4	83.1	79.6
Average[a]	82.1	88.2	83.4	–	–	–	47.7	46.7	47.1	–	–	–	88.7	91.1	89.8
Homologous gene pairs															
Human–chimpanzee	59.3	26.0	36.2	89.8	28.6	43.4	49.0	47.2	48.1	58.0	54.7	56.3	98.7	98.4	98.5
Human–macaque	54.9	30.0	38.8	87.2	31.2	46.0	37.9	35.9	36.8	46.4	42.2	44.2	90.6	90.8	90.7
Human–mouse	51.9	47.0	49.3	82.5	49.9	62.2	30.6	27.9	29.2	36.8	34.3	35.5	86.8	91.4	89.1
Human–rat	47.2	55.5	51.0	77.4	61.0	68.2	30.6	26.6	28.4	36.4	31.2	33.6	85.8	87.6	86.7
Human–dog	52.1	84.8	64.5	88.7	82.1	85.3	38.1	36.6	37.3	43.4	42.4	42.9	93.2	95.3	94.2
Human–opossum	31.2	78.3	44.6	70.3	65.6	67.9	24.2	19.1	21.2	27.3	20.4	23.3	55.9	55.8	55.9
Average	49.4	53.6	47.4	82.7	53.1	62.2	35.1	32.2	33.5	41.4	37.5	39.3	85.2	86.6	85.8

We show the respective accuracies of DAGChainer, ADHoRe and OSfinder in the pairwise comparisons of mammalian genomes. Here, '–' indicates that the calculation of orthologous segments was impossible in our environment due to either time or space limitations.

[a]For the purpose of a fair comparison, the respective accuracies in the human–chimpanzee and human–macaque comparisons were not used in calculating the average values.

parameters) and when they were homologous gene pairs (39.3% with a grid search). We discuss the reason for the low *F*-scores achieved by ADHoRe when comparing mammalian genomes in Section 4. These results demonstrate that OSfinder has greater accuracy than the other two programs. The average *F*-scores of OSfinder were notably higher than those of DAGChainer and ADHoRe, even though the latter two were executed with optimized parameter values.

The high accuracy of OSfinder is supported further by the results in the pairwise comparison of bacterial genomes (Supplementary Table S4). When the anchors were homologous sequences, the average *F*-score of OSfinder was 85.3%, which was 14.2% higher than that of DAGChainer with a grid search optimization and 20.8% higher than that of ADHoRe with a grid search optimization. When the anchors were homologous gene pairs, the average *F*-score of OSfinder was 92.2%, which was 7.5% higher than that of DAGChainer with a grid search optimization and 56.2% higher than that of ADHoRe with a grid search optimization.

3.1.2 Comparison with syntenic nets The UCSC genome browser provides a common repository for genomic annotation data (Karolchik *et al*., 2008; Kuhn *et al*., 2007). Syntenic nets, which are unique annotations in the UCSC genome browser, are genomic regions descended from a single genomic segment in a common ancestor without macrorearrangements. In Table 2, we present the accuracy of syntenic nets together with the accuracy of OSfinder. For the purpose of a fair comparison, Table 2 displays the accuracy of OSfinder when homologous sequences were used as anchors.

We can see in Table 2 the trade off between the sensitivity and the specificity. OSfinder achieved a 25.6% higher average specificity than syntenic nets, while the average sensitivity of syntenic nets was 3% higher than that of OSfinder. Regarding the average F-score, OSfinder achieved a 14% higher value than syntenic nets.

Table 2. Accuracy of syntenic nets and OSfinder in the pairwise comparison of mammalian genomes

Genomes	Syntenic nets			OSfinder		
	Sn	Sp	*F*	Sn	Sp	*F*
Human–chimpanzee	98.9	85.6	91.8	99.0	98.4	98.7
Human–macaque	98.5	66.5	79.4	97.9	96.9	97.4
Human–mouse	97.2	69.5	81.0	93.5	95.1	94.3
Human–rat	97.1	66.3	78.8	89.1	90.2	89.6
Human–dog	98.2	60.7	75.0	95.7	96.0	95.8
Average	98.0	69.7	81.2	95.0	95.3	95.2

3.2 Accuracy in multiple genome comparisons

The accuracy of OSfinder in multiple genome comparisons was compared with that of the TBA program (Blanchette *et al*., 2004) and Mercator (Dewey *et al*., 2006; Dewey, 2007). The respective accuracies of these programs were evaluated in comparisons of mammalian X chromosomes. For the generation of input for OSfinder, anchors among multiple genomes were detected by using Murasaki (Popendorf *et al*., 2007). For the generation of input for the TBA program, the BLASTZ alignments (Altschul *et al*., 1997; Schwartz *et al*., 2003) between all pairs of genomes

Table 3. Accuracy in the comparison of multiple mammalian genomes

Genomes	TBA			Mercator			OSfinder		
	Sn	Sp	*F*	Sn	Sp	*F*	Sn	Sp	*F*
Human–chimpanzee–macaque	96.7	82.7	89.1	97.6	97.6	97.6	97.6	97.6	97.6
Human–chimpanzee–macaque–mouse	68.1	43.5	53.1	90.2	85.9	88.0	97.1	97.8	97.5
Human–chimpanzee–macaque–dog	96.5	55.0	70.1	95.8	96.9	96.3	96.9	96.9	96.9
Human–chimpanzee–macaque–mouse–dog	66.8	33.1	44.3	85.0	81.2	83.0	94.0	97.5	95.7
Average	82.0	53.6	64.2	92.2	90.4	91.2	96.4	97.5	96.9

under comparison were calculated. For the generation of input for Mercator, homologous gene pairs were detected between all pairs of genomes under comparison by using BLASTP program. Note that TBA and Mercator take as input the sets of anchors detected between all pairs of genomes under comparison, whereas OSfinder takes as input a set of anchors detected among multiple genomes. Thus, TBA and Mercator require the $\mathbf{O}(N^2)$ iterations of the calculation of anchors, where N represents the number of genomes under comparison, whereas it is sufficient to perform the calculation of anchors among multiple genomes only once for the input of OSfinder. The detailed procedures for performing calculations with TBA and Mercator are described in Supplementary Materials.

Table 3 shows the accuracy in the comparison of multiple mammalian genomes. The results contain two important points. The first is that the accuracy of OSfinder was extremely high, with F-scores of >95%. The average F-score of OSfinder was 96.9%, which was notably higher than that of TBA (64.2%) and Mercator (91.2%). The second point is that the F-scores of OSfinder in multiple genome comparisons were slightly higher than that in pairwise genome comparisons (Table 2). For example, the F-score in the human–chimpanzee–macaque–mouse comparison (97.5%) was higher than that in the human–mouse comparison (95.8%). This tendency is also visible in the results for the human–chimpanzee–macaque–dog comparison. These results imply that the accuracy of OSfinder in pairwise comparisons can be improved by adding closely related genome(s) and by comparing multiple genomes.

In Supplementary Table S5, we present the accuracy in the comparison of multiple bacterial genomes. These results also demonstrate the high accuracy of OSfinder with F-scores >90%. Supplementary Tables S4 and S5 show that the average F-score in the Mtu-Mbo-Mle (Mtu-Mbo-Mpa) comparison was higher than that in the Mtu-Mle (Mtu-Mbo) comparison. The results were the same as the results obtained from the comparison of multiple mammalian genomes.

4 DISCUSSION AND CONCLUSION

The results in this article have demonstrated the potential of stochastic models and learning algorithms in OSfinder to improve the accuracy of orthology mapping in both pairwise and multiple genome comparisons. We have shown that our novel algorithm makes it possible to identify orthologous segments with accuracy which is consistently higher than that of other algorithms, without any manual effort to determine the parameter values.

Quality-based methods, such as ADHoRe (Vandepoele *et al*., 2002) and SyMAP (Soderlund *et al*., 2006), estimate the quality by computing the coefficient of determination. ADHoRe with a grid search optimization showed an extremely high accuracy in the Mtu-Mbo comparison (98% in F-score when the anchors were homologous gene pairs), although its accuracy in the Mtu-Mle and Mtu-Mpa comparisons was profoundly low (5.1 and 4.9% in F-score, respectively). Moreover, ADHoRe also showed low F-scores in mammalian genome comparisons (47.1% in average F-score when the anchors were homologous sequences). These results imply that although quality-based methods are excellent approaches when comparing very closely related genomes, these methods are not adequate when comparing distantly related genomes where the positions of the orthologous anchors cannot be fitted with linear regression models.

DAGChainer (Haas *et al*., 2004) measures the diagonal properties of the input anchors by utilizing a scoring scheme which is more relaxed than linear regression models. The scoring scheme makes it possible to compare distantly related genomes while maintaining a relatively high F-score (∼80% when the anchors are homologous gene pairs in the pairwise comparison of bacterial genomes). The scoring scheme, however, suffers from low specificity when a large fraction of the input anchors are non-orthologous (e.g. when the anchors are homologous sequences in the pairwise comparison of bacterial genomes). Therefore, the accuracy of distinguishing between orthologous anchors and non-orthologous anchors is the key to performing orthology mapping with consistently high accuracy.

The scoring scheme of OSfinder takes into account the distance between collinear anchors instead of the coefficient of determination of the linear regression. Furthermore, stochastic models are employed in OSfinder for accurately distinguishing between orthologous anchors and non-orthologous anchors. Thus, the OSfinder algorithm consistently achieves high accuracy even when distantly related genomes are compared and when a large fraction of the anchors are not orthologous.

With the rapidly increasing amount of sequence data, the automation of orthology mapping and the ability to compare multiple genomes will continue to become ever more important for high-throughput genome analysis. In addition to the seven mammalian genomic sequences used in our analysis, draft sequences for orangutan (*Pongo pygmaeus abelii*), cow (*Bos taurus*) and horse (*Equus caballus*) have already been made available in the Ensembl genome browser. Furthermore, it is expected that over 20 mammalian genome sequences will become available in the

near future. It is expected that the calculation results of OSfinder can be further improved by using the increasing number of closely related genome sequences.

ACKNOWLEDGEMENT

We thank Kengo Sato for helpful discussions.

Funding: Ministry of Education, Culture, Sports, Science and Technology of Japan Grant-in-Aid for Scientific Research on Priority Area 'Comparative Genomics' (No. 17018029).

Conflict of interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res.*, **25**, 3389–3402.

Bennetzen,J. and Ramakrishna,W. (2002) Numerous small rearrangements of gene content, order and orientation differentiate grass genomes. *Plant Mol. Biol.*, **48**, 821–827.

Blanchette,M. *et al*. (2004) Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, **14**, 708.

Bourque,G. *et al*. (2004) Reconstructing the genomic architecture of ancestral mammals: lessons from human, mouse, and rat genomes. *Genome Res.*, **14**, 507–516.

Bourque,G. *et al*. (2005) Comparative architectures of mammalian and chicken genomes reveal highly variable rates of genomic rearrangements across different lineages. *Genome Res.*, **15**, 98–110.

Calabrese,P.P. *et al*. (2003) Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics*, **19** (**Suppl. 1**), 74–80.

Cannon,S.B. *et al*. (2003) DiagHunter and GenoPix2D: programs for genomic comparisons, large-scale homology discovery and visualization. *Genome Biol.*, **4**, R68.

Dewey,C.N. *et al*. (2006) Parametric alignment of Drosophila genomes. *PLoS Comput. Biol.*, **2**, e73.

Dewey,C.N. (2007) Aligning multiple whole genomes with Mercator and MAVID. *Methods Mol. Biol.*, **395**, 221–236.

Frazer,K.A. *et al*. (2004) VIATA: computational tools for comparative genomics. *Nucleic Acids Res.*, **32**, W273–W279.

Gibbs,R.A. *et al*. (2004) Genome sequence of the Brown Norway rat yields insights into mammalian evolution. *Nature*, **428**, 493–521.

Haas,B.J. *et al*. (2004) DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics*, **20**, 3643–3646.

Hampson,S. *et al*. (2003) LineUp: statistical detection of chromosomal homology with application to plant comparative genomics. *Genome Res.*, **13**, 999–1010.

Hubbard,T. *et al*. (2005) Ensembl 2005. *Nucleic Acids Res.*, **33**, 447–453.

Hubbard,T. *et al*. (2007) Ensembl 2007. *Nucleic Acids Res.*, **35**, 610–617.

Karolchik,D. *et al*. (2008) The UCSC Genome Browser Database: 2008 update. *Nucleic Acids Res.*, **36**, D773–D779.

Kent,W.J. *et al*. (2003) Evolutions cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proc. Natl Acad. Sci. USA*, **100**, 11484–11489.

Koonin,E.V. (2005) Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.*, **39**, 309–338.

Kuhn,R.M. *et al*. (2007) The UCSC genome browser database: update 2007. *Nucleic Acids Res.*, **35**, 668–673.

Ma,J. *et al*. (2006) Reconstructing contiguous regions of an ancestral genome. *Genome Res.*, **16**, 1557–1565.

Murphy,W.J. *et al*. (2005) Dynamics of mammalian chromosome evolution inferred from multispecies comparative maps. *Science*, **309**, 613–617.

Pevzner,P. and Tesler,G. (2003) Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res.*, **13**, 37–45.

Popendorf,K. *et al*. (2007) Murasaki – homology detection across multiple large-scale genomes. In *Fifth Annual RECOMB Satellite Workshop on Comparative Genomics*. San Diego, USA.

Pruitt,K.D. *et al*. (2007) NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.*, **35**, D61–D65.

Remm,M. *et al*. (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, **314**, 1041–1052.

Schwartz,S. *et al*. (2003) Human-mouse alignments with BLASTZ. *Genome Res.*, **13**, 103–107.

Sinha,A.U. and Meller,J. (2007) Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. *BMC Bioinformatics*, **8**, 82.

Soderlund,C. *et al*. (2006) SyMAP: A system for discovering and viewing syntenic regions of FPC maps. *Genome Res.*, **16**, 1159–1168.

Song,R. *et al*. (2002) Mosaic organization of orthologous sequences in grass genomes. *Genome Res.*, **12**, 1549–1555.

Tatusov,R.L. *et al*. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.

Tatusov,R.L. *et al*. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.

Tesler,G. (2002) GRIMM: genome rearrangements web server. *Bioinformatics*, **18**, 492–493.

Vallender,E.J. *et al*. (2006) SPEED: a molecular-evolution-based database of mammalian orthologous groups. *Bioinformatics*, **22**, 2835–2837.

Vandepoele,K. *et al*. (2002) The automatic detection of homologous regions (ADHoRe) and its application to microcolinearity between Arabidopsis and rice. *Genome Res.*, **12**, 1792–1801.

Waterston,R.H. *et al*. (2002) Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**, 520–562.

Zheng,X.H. *et al*. (2005) Using shared genomic synteny and shared protein functions to enhance the identification of orthologous gene pairs. *Bioinformatics*, **21**, 703–710.