Using Temporal Reasoning for Genome Map Assembly

Olivier Schmeltzer

INRIA Rhône-Alpes - LIFIA
46, avenue Félix Viallet F-38031 GRENOBLE Cedex 1
Phone: (33) 76.57.45.78. Fax: (33) 76.57.46.95.
e-mail: Olivier.Schmeltzer@imag.fr

Abstract

Genomic maps are an indispensable tool for molecular biologists; their modelling has to take into account representation as well as computational issues. The algorithmic complexity of the assembly task is already huge and is even made worse when one wishes to deal with inconsistencies and provide generic tools. This work presents an algorithm tackling the assembly problem by using temporal reasoning techniques. The algorithm has to transform the initial input data, i.e. qualitative and quantitative relations between entities that appear on the maps, so that temporal reasoning algorithms can be applied successfully; this is achieved by performing a partition of these relations upon their relative orientation, creating islets of relations in which reasoning mechanisms are applied. The implementation of the algorithm is based on a temporal reasoning software, taken as is, which gives a high genericity since any improvement in this software (such as efficiency or the management of flexible constraints) can be immediately used by the algorithm.

Introduction

Genomic maps are a means of representing mapping information that are indispensable to molecular biologists because they model the data they manipulate, and make them at once visible, as well when building a map as when using it. The aim of this paper is to deal with the assembly of integrated genomic maps, i.e. using information from any of the three different kinds of map, cytogenetic, genetic and physical. This information consists typically of relations between entities belonging to the maps.

The integration provides new problems that have not been completely addressed yet. First, the entities that appear on the maps may have different lengths; for instance, a gene on a genetic map is represented by a point (length = 0), while it is an interval on a physical map (length $\neq$ 0). This aspect has to be taken into account because the relation between two entities depends upon their length (point or fragment). Second, quantitative information is not directly translatable from one map to another because there is not a constant scale factor between them. Third, in order to combine information from all the maps, it is necessary to define a framework in which all the knowledge can be deposited, with the aim of applying inferences on all the entities.

So far, existing systems have only considered a more or less restricted field of the genome map assembly problem. Letovsky and Berlyn (Letovsky & Berlyn 1992) have developed a system called CPROP to compute orders and distances in genetic maps; the choice of genetic maps has greatly simplified the problem since it allows them to consider only entities without length. Graves (Graves 1993) presented a representation scheme for distance and order relations, but he did not go so far as to propose an algorithm. Honda *et al.* (Honda *et al.* 1993) define an interesting framework to represent biological objects using object-oriented programming, but the algorithm they propose seems to be dedicated to the contig assembly problem. Lee *et al.* (Lee *et al.* 1993) developed a very interesting formalism to represent relations between map entities, which has the advantage of declaring both the relations and the connected inference mechanisms. Cui (Cui 1994) also presented the use of temporal and spatial reasoning for the representation of mapping information, but he did not design an algorithm able to compute the assembly.

There are still two important aspects that we wish to tackle: first is the declarativity of both the relations and the algorithm, so that it can be extended as easily as possible; second is the ability to manage inconsistencies that are bound to appear because the knowledge is experimental.

Formalising relations

Two kinds of relations are to be modeled: qualitative relations are the way to represent more or less imprecised knowledge between two entities; quantitative relations are used to model distances between positions (which are either the endpoints of interval entities or the positions of points).

Qualitative relations

Let Q_l be the set of all qualitative relation types, i.e. all the different generic relations that allow to build particular relations between two entities. Typically, *before* and *after* belong to Q_l, and are the basis for describing such knowledge as *gene D1S21 lies before gene TSHB*.

Unfortunately, since the DNA strand can be read one way or the other, some of these relations need a reference that specifies their orientation. The previous piece of knowledge does have a meaning only if it is also stated relatively to what the relation holds. The default value is the orientation given by the whole chromosome, but, very often, the reference is more local and has to be explicitly specified. As a consequence, let Q_l^+ be the set of these relations that, because they imply an orientation, need a reference, and Q_l^- those that do not need one.

When dealing with the genome map assembly task, since the equality of endpoint positions is not relevant to the problem, the relations are the following (see also figure 1):
$Q_l^+ = \{$before, after, overlaps_before, overlaps_after$\}$;
$Q_l^- = \{$contains, contained_by, disjoint, not_disjoint$\}$.

Relations between points and intervals use the same sets, provided that impossible relations are excluded (for instance, a point containing an interval).

What is then a particular relation? Either it is an oriented relation, i.e. a triplet composed of one element of Q_l^+, the two entities in relation and an entity that defines the orientation, or it is a couple of one element of Q_l^- and the two entities linked by the relation. It is convenient to add to these relations the relation *order* which expresses that a certain number of entities (greater than or equal to three) are locally ordered. Since this relation does not need a reference, it will be put in Q_l^-.

To end with, it is also necessary to have a relation stating that two entities have the same orientation or reverse orientations. These two relations do not need a reference either. They will be denoted *same_orientation* and *reverse_orientation*.

As for notations, we will use the following:
$\forall t \in Q_l^+, (e_1\, t_e\, e_2)$ links e_1 and e_2 through t with respect to the reference e;
$\forall t \in Q_l^-, (e_1\, t\, e_2)$ links e_1 and e_2 through t;
$order(e_1, \ldots, e_n)$ links $e_1, \ldots, e_n$, with the relation *order*.

Quantitative relations

Quantitative relations aim at representing distance with uncertainty between positions of entities. The existence of three different kinds of maps compels us to precise the type of map used when expressing a quantitative relation. This is the reason why such a relation consists of two positions, two positive real numbers for distance and uncertainty and the type of the map.

We will use the following notation to represent such a relation:

$$(P_i \leftrightarrow_p P_j = [distance, uncertainty]),$$

where p stands for the type of the map.

Temporal reasoning

Guidi and Roderick (Guidi & Roderick 1993) have noticed an evident correspondence between the qualitative relations that exist between biological entities and work on time representation. However, there are many differences and the application of temporal reasoning techniques is far from being straightforward. We first introduce temporal representation and reasoning formalisms and we look in the next section at how this can be successfully adapted to genome map assembly.

Allen's interval algebra

Allen (Allen 1983) introduced a formalism to represent the relations between intervals. There are 13 atomic relations, defined by the relations on their endpoints (table 1). A formula $(I_1 r I_2)$ links two intervals through the relation r, and is satisfied if the relations on the endpoints of I_1 and I_2 are satisfied. To allow uncertain information, the formulas are extended to disjunctions (or unions) of atomic relations, satisfied if any of the elements of the disjunction is satisfied. The set of these disjunctions is denoted $\mathcal{A}$ and contains 2^{13} relations.

Temporal reasoning aims at deducing new relations from an initial set, using as inference mechanism transitivity (or composition) of relations. For instance, the two facts *(I_1 before I_2)* and *(I_2 before I_3)* imply that *(I_1 before I_3)*. This inference mechanism can also be used with disjunctions of atomic relations thanks to the composition table of atomic relations alone because composition distributes over union.

From a set of formulas Θ (often represented in a network where nodes are intervals and edges relations between intervals), typical reasoning problems are:

1. Is there a model of Θ, i.e interpretations of the intervals that satisfy all the relations?
2. What is the strongest relation holding between two intervals (deductive closure (Vilain & Kautz 1986) or minimal labelling (van Beek 1990) problem)?

Seeing that these problems (which are kinds of *Constraint Satisfaction Problems* or CSP) are, for $\mathcal{A}$, NP-complete, either subsets of $\mathcal{A}$ have been defined or polynomial algorithms have been developed to compute local consistencies. Allen proposed a path-consistency algorithm, that ensures that whatever three intervals of the network, the composition of two relations between these intervals is included in the third. This algorithm runs in time $O(n^3)$ (where n is the number of intervals).

Different subsets of $\mathcal{A}$ have been introduced; the pointisable relations $\mathcal{P}$ is the subset of $\mathcal{A}$ which can be translated into the point algebra (Vilain & Kautz

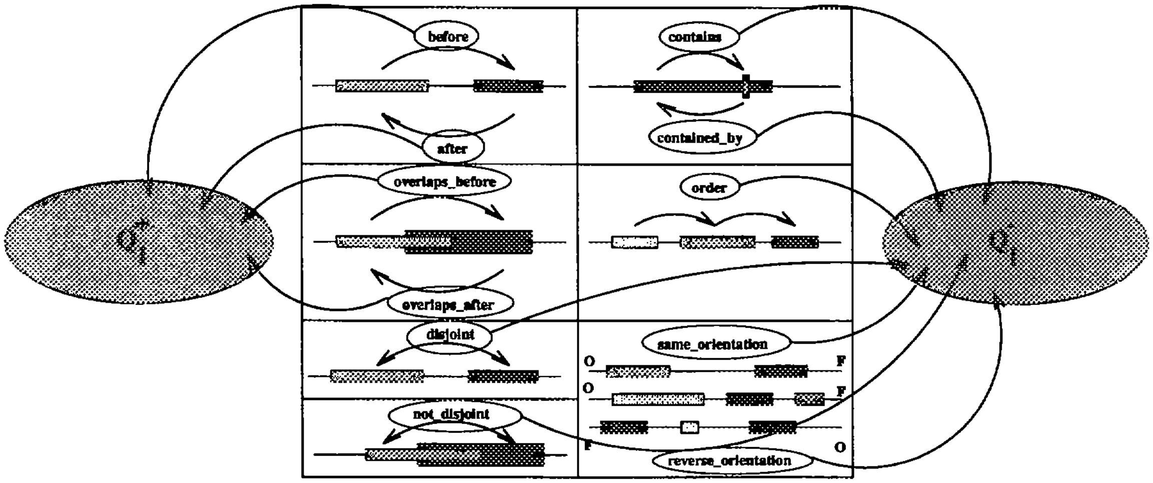

Figure 1: The sets of qualitative relation types between biological entities: Q_l^+ contains relation types that need a reference specifying the orientation, Q_l^- all the other relation types.

1986), i.e. as conjunctions of relations between the endpoints. More recently, Nebel and Bürckert (Nebel & Bürckert 1993) have specified the maximal subclass of $\mathcal{A}$ containing all the basic relations for which the satisfiability – and the minimal labelling – are tractable. Nonetheless, it is important to notice that the satisfiability of a network whose relations belong to the subset Δ_0 containing {disjoint, not_disjoint, universal relation} is NP-complete (Golumbic & Shamir 1993).

Quantitative networks

A formalism of quantitative networks have been proposed by Dechter *et al.* (Dechter, Meiri, & Pearl 1991) in order to express inequalities between temporal points. The variables are now X_i, time points, and constraints are given as a set of intervals $\{I_1, \ldots, I_n\} = \{[a_1, b_1], \ldots, [a_n, b_n]\}$ to which a variable must belong. Unary constraints can be transformed into binary constraints by simply introducing the origin of time. So, all the constraints have the same expression:

$$(a_1 \leq X_j - X_i \leq b_1) \vee \ldots \vee (a_n \leq X_j - X_i \leq b_n).$$

Two problems appear also when dealing with quantitative networks: the satisfiability of the network and the minimal network computation, in which each variable is reduced to the smallest set of allowed values.

The consistency of the general network is a NP-complete problem, but, if the constraints are restricted to only one interval – instead of a union – (the network is then called *simple*), the Floyd-Warshall algorithm extracts the minimal network in $O(n^3)$.

Integrating qualitative and quantitative temporal reasoning

Two different formalisms have been proposed to integrate qualitative and quantitative reasoning (Kautz & Ladkin 1991; Meiri 1991). We only present the first one that is used by our genome map assembly algorithm.

This formalism aims at dealing both with Allen's interval algebra and simple quantitative relations like $m \leq x - y \leq n$ where x and y are interval endpoints. To achieve this, two translation procedures are defined, from quantitative to qualitative, and from qualitative to quantitative; these procedures are successively applied to the initial networks until the stability of the networks is reached.

The procedure that translates metric constraints into Allen relations is complete and requires $O(n^3)$ time. The counterpart procedure is complete only if the minimal network has been determined, and runs in time $O(n^2)$.

Time and genomic maps

The previous descriptions of genomic map requirements and temporal reasoning techniques have shown similarities but also many differences. Let us enumerate them.

Similarities

- Qualitative relations: both problems aim at expressing qualitative relations, though their set is not the same. Nonetheless, since *disjoint* and *not_disjoint* belong to the genomic map set, the full resolution of the network is NP-complete, as with Allen's set.
- Quantitative relations: quantitative relations on genomic maps express distance and uncertainty which can be easily translated into the formalism of simple quantitative networks, since we have the following equivalence (inside a unique type of map): $(P_i \leftrightarrow P_j = [d_{ij}, i_{ij}]) \Leftrightarrow d_{ij} - i_{ij} \leq P_j - P_i \leq d_{ij} + i_{ij}$.

Basic relation	Symbol	Example	Endpoint relations
X before Y	b	xxxx	$X^- < Y^-, X^- < Y^+$
Y after X	$\breve{b}$	yyyy	$X^+ < Y^-, X^+ < Y^+$
X meets Y	m	xxxxx	$X^- < Y^-, X^- < Y^+$
Y met-by X	$\breve{m}$	yyyyy	$X^+ = Y^-, X^+ < Y^+$
X overlaps Y	o	xxxxx	$X^- < Y^-, X^- < Y^+$
Y overlapped-by X	$\breve{o}$	yyyyy	$X^+ > Y^-, X^+ < Y^+$
X during Y	d	xxx	$X^- > Y^-, X^- < Y^+$
Y includes X	$\breve{d}$	yyyyyy	$X^+ > Y^-, X^+ < Y^+$
X starts Y	s	xxx	$X^- = Y^-, X^- < Y^+$
Y started-by X	$\breve{s}$	yyyyyyy	$X^+ > Y^-, X^+ < Y^+$
X finishes Y	f	xxx	$X^- > Y^-, X^- < Y^+$
Y finished-by X	$\breve{f}$	yyyyyy	$X^+ > Y^-, X^+ = Y^+$
X equals Y	$=$	xxxxx	$X^- = Y^-, X^- < Y^+$
		yyyyy	$X^+ > Y^-, X^+ = Y^+$

Table 1: The thirteen relations defined by Allen. These relations express all the relative positions of two intervals. The notation $\breve{}$ stands for the converse of the relation.

Differences

- A non-oriented axis: since the DNA strand can be read one way or the other, contrary to time which goes from past to future, some qualitative relations have to add a reference that specifies the orientation to be taken into account. In particular, in order to apply an algorithm solving the network of qualitative relations, they should all possess the same orientation (i.e. every two references should be linked by the *same_orientation* relation). Quantitative relations are also affected by this difference since the application of the Floyd-Warshall algorithm needs the additivity of the distances.
- Different types of maps: this allows entities to have different representations on the different maps (for example a point on the genetic map and a fragment on the physical one). To tackle this problem, one is compelled to define allowed translations from one representation to the other. We won't go into details concerning this topic; once these translation rules have been settled, it is possible to only consider intervals. Another consequence lies in the compulsory partition of quantitative reasoning in each kind of map, since there is no precise scale factor from one to another. Practically, this means that every computation on quantitative relations has to be made inside a unique type of map.
- A subset of $\mathcal{A}$: the fact that, instead of 13 Allen relations, we only have to deal with 6 (Cf. next section), can be useful for efficiency considerations. Indeed, it seems then reasonable to store all the composition table for the disjunctions, without having to compute the composition every time it is needed.
- The existence of experimental uncertainties: this point is more constraining than in temporal reasoning because, when dealing with the map assembly problem, inconsistencies are much more likely to happen. They should be taken into account so that reasoning can be continued. We provide a partial answer to this problem, and more importantly, we show that it can be handled by temporal reasoning algorithms (much research work is being done to tackle these *dynamic and flexible constraints*).

An approximate composition table for genomic maps

Since relevant qualitative relations when assembling genomic maps do not use equalities on endpoints, only six relations from the 13 defined by Allen that are used (table 2). The relations *disjoint* and *not_disjoint* can be expressed using disjunctions of the set denoted $\mathcal{R}_{map}$ of these six atomic map relations. Let $\mathcal{C}$ be the set of disjunctions of relations of $\mathcal{R}_{map}$.

Map Relation	Allen's Equiv.	Converse
before $\preceq$	$\{b, m\}$	$\succeq$
overlaps_before $\ll$	$\{m, o, s, \breve{f}, =\}$	$\gg$
contained $\sqsubseteq$	$\{d, s, f, =\}$	$\sqsupseteq$
disjoint $\neq$	$\{b, \breve{b}, m, \breve{m}\}$	$\neq$
not_disjoint $\bowtie$	$\{o, \breve{o}, m, \breve{m}, s,$ $\breve{s}, d, \breve{d}, f, \breve{f}, =\}$	$\bowtie$

Table 2: Map relations in terms of Allen's. The two last relations *disjoint* and *not_disjoint* can be expressed as disjunctions of the six previous ones. They do not need a reference, though they seem to use relations that do need one, because they contain both a relation and its converse.

To ensure the stability of composition for $\mathcal{C}$, we de-

o	$\preceq$	$\succeq$	$\ll$	$\gg$	$\sqsubseteq$	$\sqsupseteq$
$\preceq$	$\preceq$	$\top$	$\preceq$	$\{\preceq, \ll, \sqsubseteq\}$	$\{\preceq, \ll, \sqsubseteq\}$	$\preceq$
$\succeq$	$\top$	$\succeq$	$\{\succeq, \gg, \sqsubseteq\}$	$\succeq$	$\{\succeq, \gg, \sqsubseteq\}$	$\succeq$
$\ll$	$\preceq$	$\{\succeq, \gg, \sqsupseteq\}$	$\{\preceq, \ll\}$	$\{\ll, \gg, \sqsubseteq, \sqsupseteq\}$	$\{\ll, \sqsubseteq\}$	$\{\preceq, \ll, \sqsupseteq\}$
$\gg$	$\{\preceq, \ll, \sqsupseteq\}$	$\succeq$	$\{\ll, \gg, \sqsubseteq, \sqsupseteq\}$	$\{\succeq, \gg\}$	$\{\gg, \sqsubseteq\}$	$\{\succeq, \gg, \sqsupseteq\}$
$\sqsubseteq$	$\preceq$	$\succeq$	$\{\preceq, \ll, \sqsubseteq\}$	$\{\succeq, \gg, \sqsubseteq\}$	$\sqsubseteq$	$\top$
$\sqsupseteq$	$\{\preceq, \ll, \sqsupseteq\}$	$\{\succeq, \gg, \sqsupseteq\}$	$\{\ll, \sqsupseteq\}$	$\{\gg, \sqsupseteq\}$	$\{\ll, \gg, \sqsubseteq, \sqsupseteq\}$	$\sqsupseteq$

Table 3: Composition of map relations (o is the composition operator). When both relations need a reference, the result is valid if and only if their references share the same orientation. When only one relation has to specify a reference, the resulting relation takes this reference (or, at least, a reference sharing the same orientation).

fine an approximate composition relation taking as result of the composition of two relations the smallest disjunction including the exact result obtained by applying Allen's composition relation (table 3).

The genome map assembly algorithm

Before describing in detail the algorithm, let us first summarise the incoming data and all the inference mechanisms that will be applied.

Input data

What follows are all the input data that are the basis of the assembly algorithm.

1. The entities:
   (a) $\mathcal{E} = \{e_i\}$, a set of entities;
   (b) $Pos = \{P_i\} = \{O_i\} \cup \{E_i\}$, a set of positions corresponding to entity endpoints (origin and end).
2. The relations:
   (a) Qualitative relations: $\mathcal{R}_{quali} = \mathcal{R}^{+}_{quali} \cup \mathcal{R}^{-}_{quali}$, a set of qualitative relations: $\mathcal{R}^{+}_{quali} = \{(e_i\, t^{+}_{e}\, e_j)\}$ and $\mathcal{R}^{-}_{quali} = \{(e_i\, t^{-}\, e_j)\} \cup \{order(e_k, \ldots, e_{k+n})\}$, with $e_i, e_j, e_k, \ldots e_{k+n} \in \mathcal{E}, t^{+} \in \mathcal{Q}^{+}_{l} = \{\preceq, \succeq, \ll, \gg\}, t^{-} \in \mathcal{Q}^{-}_{l} = \{\sqsubseteq, \sqsupseteq, \neq, \bowtie, \rightrightarrows, \rightleftarrows\}$, where $\rightrightarrows$ stands for the same orientation relation and $\rightleftarrows$ for the reverse orientation relation.
   (b) Quantitative relations: $\mathcal{R}_{quanti} = \{(P_i \leftrightarrow_p P_j = [d_{ij}, i_{ij}])\}$, a set of quantitative relations where $P_i, P_j \in Pos, p \in \{\text{cyto, gen, phys}\}$.

Inference mechanisms

Among all the following inference mechanisms, some have already been introduced while some are new and are elucidated in this section.

1. Generating orientations from oriented qualitative relations: the problem can be settled the following way: what is the orientation relation holding between e and e' knowing the two relations $(e_1\, t_e\, e_2)$ et $(e_1\, t'_{e'}\, e_2)$ (where t and t' are disjunctions of relations in $\mathcal{Q}^{+}_{l}$)? For instance, if $(e_1 \{\preceq, \ll\}_e e_2)$ and $(e_1 \{\preceq\}_{e'} e_2) \in \mathcal{R}^{+}_{quali}$, then $(e_1 \{\preceq\}_e e_2) \wedge (e \rightrightarrows e')$. Table 4 synthesises the possible inferences (let us note that it is similar to the one described in (Lee *et al.* 1993)).
2. Composition table for qualitative relations (table 3). Nonetheless, it should be taken care of the relative orientation of the references when dealing with oriented qualitative relations.
3. Constraint networks:
   (a) from a qualitative relation network sharing the same reference, path-consistency, satisfiability or minimal labelling algorithms can be applied;
   (b) from a quantitative relation network for which additivity is true, Floyd-Warshall algorithm gives the minimal intervals of the constrained positions.
4. Position ordering (to ensure that additivity is valid):
   (a) Thanks to the following table and to orientation and qualitative relations:

$(e_1\, r_e\, e_2)$		$(e_1$ same_orientation $e_2)$
$r = \preceq$	$(e \rightrightarrows e_1)$	$ord(O, O_1, E_1, O_2, E_2, E)$
	$(e \rightleftarrows e_1)$	$ord(O, E_1, O_1, E_2, O_2, E)$
$r = \ll$	$(e \rightrightarrows e_1)$	$ord(O, O_1, O_2, E_1, E_2, E)$
	$(e \rightleftarrows e_1)$	$ord(O, E_1, E_2, O_1, O_2, E)$
$r = \sqsupseteq$		$ord(O_1, O_2, E_2, E_1)$

Table 5: Endpoint ordering thanks to qualitative relations. For the three relations *before*, *overlaps_before* and *contains*, depending on the relative orientation of e_1 et e_2, the table gives the ordering of the endpoints according to the relative orientation of e and e_1; the first element of the column is valid when e and e_1 share the same orientation. A similar mechanism exists when $(e_1$ reverse_orientation $e_2)$

When two entities are linked through a disjunction of relations, if, for every element of this disjunction, the relative positions of endpoints are

$(e_1\, t'_e\, e_2)$ $(e_1\, t_e\, e_2)$	$\preceq$	$\ll$	$\succeq$	$\gg$	$\{\preceq,\ll\}$	$\{\succeq,\gg\}$	$\{\preceq,\gg\}$	$\{\succeq,\ll\}$
$\preceq$	$M(e\rightrightarrows e')$	C	$M(e\rightleftarrows e')$	C	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$
$\ll$	C	$M(e\rightrightarrows e')$	C	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$
$\succeq$	$M(e\rightleftarrows e')$	C	$M(e\rightrightarrows e')$	C	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$
$\gg$	C	$M(e\rightleftarrows e')$	C	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$
$\{\preceq,\ll\}$	$M(e\rightrightarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	?	?
$\{\succeq,\gg\}$	$M(e\rightleftarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	?	?
$\{\preceq,\gg\}$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	?	?	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$
$\{\succeq,\ll\}$	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightrightarrows e')$	$M(e\rightleftarrows e')$	?	?	$M(e\rightleftarrows e')$	$M(e\rightrightarrows e')$

Table 4: Generating orientations. The letter M means that the two relations can merge into one, and it specifies the relative orientation of e and e'. The letter C indicates a clash situation. The question mark shows an indetermination.

conserved, then this information can be used to apply Floyd-Warshall algorithm.

(b) From order relations:
$\forall e_1, ..., e_n \in \mathcal{E}, order(e_1, ..., e_n) \Rightarrow ord(O_1, ..., O_n)$
$\wedge ord(E_1, ..., E_n)$;

(c) From quantitative relations only:
$\forall P_1, P_2, P_3, d_{12}+i_{12} \leq d_{13}-i_{13} \Rightarrow ord(P_1, P_2, P_3)$;

(d) From the translation of qualitative relations into quantitative relations: this translation computes information on interval endpoints such as $P_i - P_j < 0$.

5. Translation of quantitative relations into qualitative relations.

Applying temporal reasoning algorithms

From the preceding description of data and inference mechanisms, how can we satisfy the constraints allowing us to apply the algorithms shown previously?

To apply an algorithm dealing with Allen's interval algebra, say for instance path-consistency, we have to overcome the non-global orientation of the qualitative relations expressed between the entities. To do this, the set of entities is partitioned[1] into islets such as, whatever two entities inside a same islet, their relative orientation is known. So, each islet is itself divided into two sub-islets such as, whatever two entities inside a sub-islet, they share the same orientation, and whatever two entities belonging to opposite sub-islets, they have an opposite orientation (figure 2).

The goal of this operation is to project every islet on the set of $\mathcal{R}^+_{quali}$ in the following way: for every entity in an islet, we keep only the qualitative relations that have this entity as reference. Then, provided that we take the converses of the relations for the entities of a sub-islet, all the relation references share the same orientation (figure 3). It is then possible to apply any of the CSP algorithms.

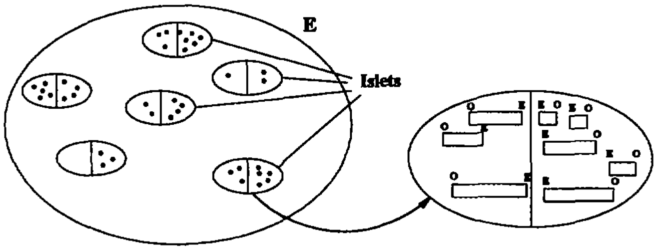

Figure 2: Partitioning the set of entities through orientation. The relative orientation of every two entities in an islet is known, and every two entities of the same sub-islet share the same orientation.

Moreover, the infered relations may be able to group islets thanks to inference mechanism number 1.

It is also possible to get information from quantitative relations in order to apply Floyd-Warshall algorithm. To do so, we have to check if the relative positions of endpoints are always the same, so that additivity of distances is true. We have at our disposal four means to ensure that (see inference mechanisms 4a, 4b, 4c, 4d).

Description of the algorithm

Figure 4 shows the different steps of the algorithm; they are more detailed hereafter.

Partitioning the sets of entities and relations into islets This step has already been explained earlier; it results in islets for which the relative orientation of their entities is known. The projection of these entities on the relations whose reference belongs to islets creates the corresponding islets in $\mathcal{R}^+_{quali}$. All the relations that have as reference an entity of I^+ are kept

[1] This is indeed a partition of $\mathcal{E}$ because the *same_orientation* relation is an equivalence relation.

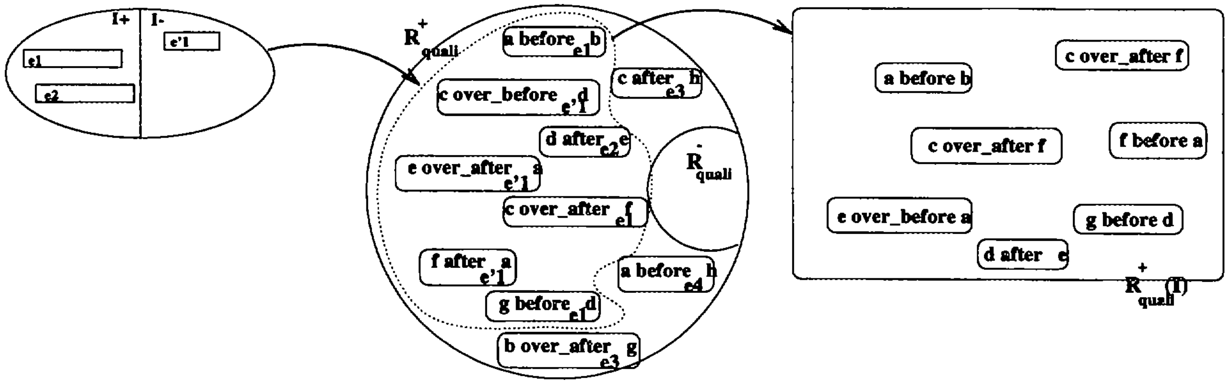

Figure 3: Projection of the islets onto qualitative relations. An islet I is projected onto $\mathcal{R}^{+}_{quali}$ by selecting all the qualitative relations whose reference is an entity of I; if this entity belongs to I^{+}, the relation is kept as is (for instance, a before b); if it belongs to I^{-}, the relation is inversed (for instance, e overlaps_after a). Then, a constraint satisfaction technique can be applied on $\mathcal{R}^{+}_{quali}(I)$.

as they are, all the other relations whose reference belongs to I^{-} are inversed to be valid with the common orientation of the previous relations.

Merging islets Merging islets can be realised when oriented relations linking two identical entities appear in two different islets. Then, it may be possible to decide the relative orientation of their references using table 4.

Concerning non oriented relations, if a qualitative relation contains $\sqsubseteq$ or $\sqsupseteq$, but its counterpart in the other islet does not, this relation is removed from the disjunction. On the contrary, if each relation contains one of these two relations, nothing can be infered since such a relation does not specify an orientation.

Ordering the quantitative relations and solving the corresponding CSP This ordering is made thanks to the previously described inference mechanisms and allows to apply additivity rules, and thus the Floyd-Warshall algorithm.

Infering new qualitative relations This is done by the temporal reasoning software from quantitative relations.

Stability? If stability is reached, i.e. no new relation has been discovered, the algorithm ends. If not, it iterates on the previous steps. Of course, the algorithm is guaranteed to stop because there is a finite number of relations between entities and positions. The time complexity of the algorithm depends upon the algorithm chosen for the resolution of the qualitative network.

Implementation

There are two separate aspects in the implementation: the first one deals with the resolution of the CSP, the second one deals with what is peculiar to genomic maps.

Solving the CSP

One of the biggest advantages of the algorithm is that we have been able to use already existing temporal reasoning algorithms without any modification. We have been using the software MATS[2] (which stands for *Metric/Allen Time System*) developed by Henry Kautz. MATS implements the resolution of temporal constraint problems; input constraints are either difference inequalities on the endpoints of intervals or Allen-style qualitative constraints.

MATS provides the following functionalities:

- reduction of the qualitative relation network using a path-consistency algorithm;
- translation of qualitative constraints into quantitative ones;
- reduction of the quantitative relation network using the Floyd-Warshall algorithm;
- translation of quantitative constraints into qualitative ones.

We have introduced map relations inside MATS representation model. For the time being, the system only translates the qualitative relations into Allen's formalism and uses consequently the whole composition table. The final result is then a disjunction of the thirteen basic interval relations instead of a disjunction of the six map relations.

[2] MATS is available by anonymous ftp on the site research.att.com in the directory /dist/ai/ (file mats.shar.Z).

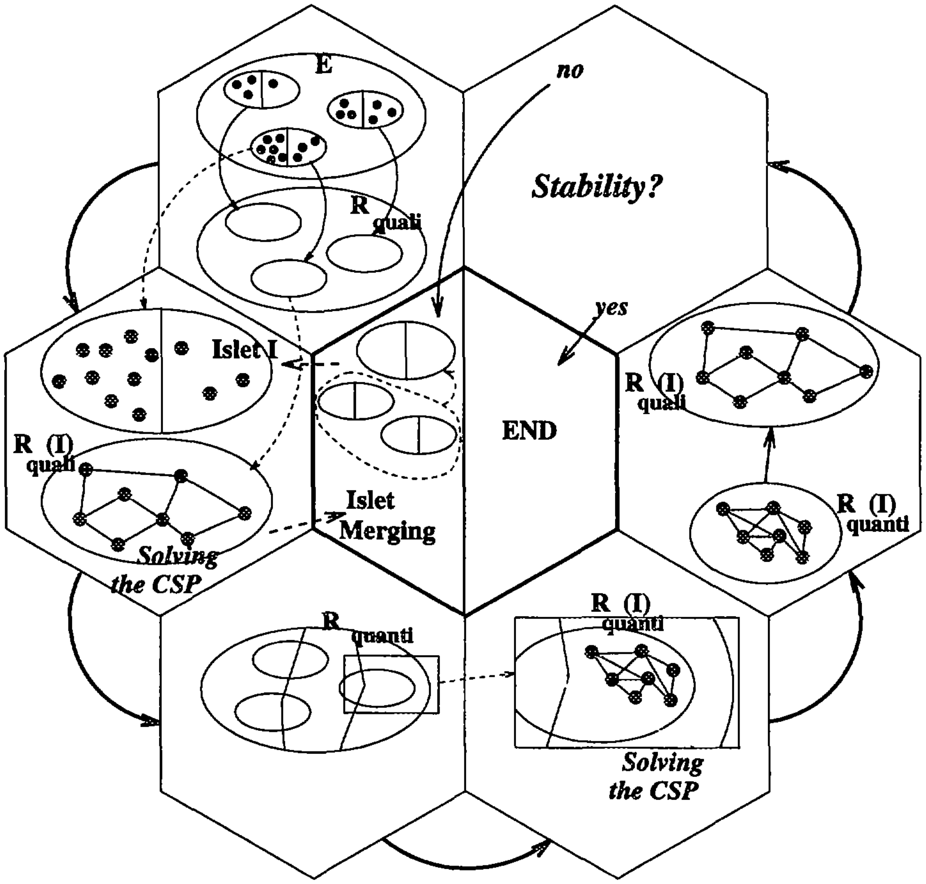

Figure 4: Genome map assembly algorithm. See text for explanations

Part specific to genomic maps

The rest of the implementation has been made in Talk2.0, an object-oriented Lisp dialect into which MATS has been translated. Each islet is implemented as a class whose attributes are the entities of the islet and its relations (qualitative and quantitative).

Discussion

Thanks to the similarities between genome map assembly and temporal reasoning, it has been made possible to use efficient algorithms dealing with the resolution of temporal constraint networks. This alleviates the burden to develop one's own algorithms and allows to profit by each improvement, as well theoretical as practical, in this field where there is still huge research work on. In particular, these improvements should include the management of inconsistencies through the use of dynamic constraints, which will provide a better interactivity between a user and the algorithm, allowing for instance a user to add, remove or relax constraints according to the results. These possibilities are absolutely necessary because one can not have a complete confidence in the experimental biological data, but they are to be handled at the CSP level; they will then be included without any effort into the genome map assembly algorithm, since the temporal reasoning algorithms are used as they are.

Moreover, the algorithm can be customised relatively to the amount of data. If the data are huge, local consistency will be chosen, so that a polynomial algorithm can be applied. On the contrary, if they are less numerous, or if the precision of the results is important, the algorithm would rather solve the networks completely.

Another point emphasised in the algorithm is the declarativity of the relations used. One can not only add new relations in the networks without having to do the whole computation again because what has already been done is kept, but also easily add new types of relation that one would wish to use, provided that the inference mechanisms are extended to tackle these new types.

For the time being, tests have only be made on very small data, and have shown the discovery of new relations. This work is to be included in a bigger frame that includes also the implementation of a modelisation of genomic maps in a knowledge representation system, including the analysis of DNA sequences through tasks and a generic map interface.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843.

Cui, Z. 1994. Using interval logic for order assembly. In *Proc. of the 2nd International Conference on Intelligent Systems for Molecular Biology*, 103–111. Stanford, CA.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1-3):61–95.

Golumbic, M. C., and Shamir, R. 1993. Complexity and algorithms for reasoning about time: A graph-theoretic approach. *Journal of the ACM* 40(5):1108–1133.

Graves, M. 1993. Integrating order and distance relationships from heterogeneous maps. In *Proc. of the 1st International Conference on Intelligent Systems for Molecular Biology*, 154–162. Washington, DC.

Guidi, J. N., and Roderick, T. H. 1993. Inference of order in genetic systems. In *Proc. of the 1st International Conference on Intelligent Systems for Molecular Biology*, 163–169. Washington, DC.

Honda, S.; Parrot, N. W.; Smith, R.; and Lawrence, C. 1993. An object model for genome information at all levels of resolution. In *Proc. of the 26th Annual Hawaii International Conference on System Sciences*, 564–573. IEEE Computer Society Press.

Kautz, H. A., and Ladkin, P. B. 1991. Integrating metric and qualitative temporal reasoning. In *Proc. of the 9th National Conference on Artificial Intelligence, AAAI-91*, 241–246.

Lee, A. J.; Rundensteiner, E. A.; Thomas, S.; and Lafortune, S. 1993. An information model for genome map representation and assembly. In *Proc. of the 2nd ACM International Conference on Information and Knowledge Management, CIKM '93*, 75–84.

Letovsky, S., and Berlyn, M. B. 1992. CPROP: a rule-based program for constructing genetic maps. *Genomics* 12:435–446.

Meiri, I. 1991. Combining qualitative and quantitative constraints in temporal reasoning. In *Proc. of the 9th National Conference on Artificial Intelligence, AAAI-91*, 260–267.

Nebel, B., and Bürckert, H. J. 1993. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. Technical Report 11, DFKI GmbH, Saarbrücken, Germany.

van Beek, P. 1990. Reasoning about qualitative temporal information. In *Proc. of the 8th National Conference on Artificial Intelligence, AAAI-90*, 728–734.

Vilain, M., and Kautz, H. 1986. Constraint propagation algorithms for temporal reasoning. In *Proc. of the 5th National Conference on Artificial Intelligence, AAAI-86*, 377–382.