of the method are as follows: (1) selection of significant subsequences (2) classification of subsequences into groups of motifs (3) delimitation of each motif by multiple alignment (4) construction of an HMM for each group and (5) redefinition of subsequences for building motifs using local search technique. Steps (3) to (5) are iterated trying to expand the motif regions until the given condition is satisfied.

We applied the method to several sets of benchmark data and 400 human promoter sequences. Our computer experiments revealed the following. (1) the method has comparable capability to the existing methods but is much faster. (2) the method can successfully extract multiple motifs in a single run. (3) the method can successfully extract motifs from a large amount of data. (4) the method successfully extracts motifs observed in a small subset of data as well as in a large subset of data. (5) the method could be applied to practical problems in the field of molecular biology.

3.3 Genetic Programming

3.3.1 Introduction

Algorithms for the construction of hidden Markov model

In this section, we discuss a problem: motif pattern extraction with hidden Markov representation from a large amount of DNA sequences concerning a life phenomenon. This is a problem of how to optimize HMM topology and parameters against uncharacterized motif patterns hidden in DNA sequences.

A method based on the combination of genetic algorithm (GA) (Goldberg, 1989; Holland, 1992) and Baum-Welch algorithm (Rabiner, 1989) was developed for this problem (Yada *et al.*, 1996b). GA was used to optimize HMM topology and initial parameter values, and Baum-Welch algorithm was used to optimize parameters based on the initial values. This method is capable of constructing any topologies. Moreover, by adopting effective operations for topology generation, efficient search which avoids local optima was accomplished. The method, however, has two problems: (A) size of data which the method can be applied to, (B) encoding method of HMM. (A) At initial generation, the method generates a population whose HMMs consist of a small number of states and optimize their network as increment of number of states. Therefore, it takes much computational time to reach a network size large enough to represent motif pattern, and the size of data which the method can be applied to is strictly limited (up to several kilo bases). (B) In GA, strong specification method is used for encoding HMM network into chromosomes

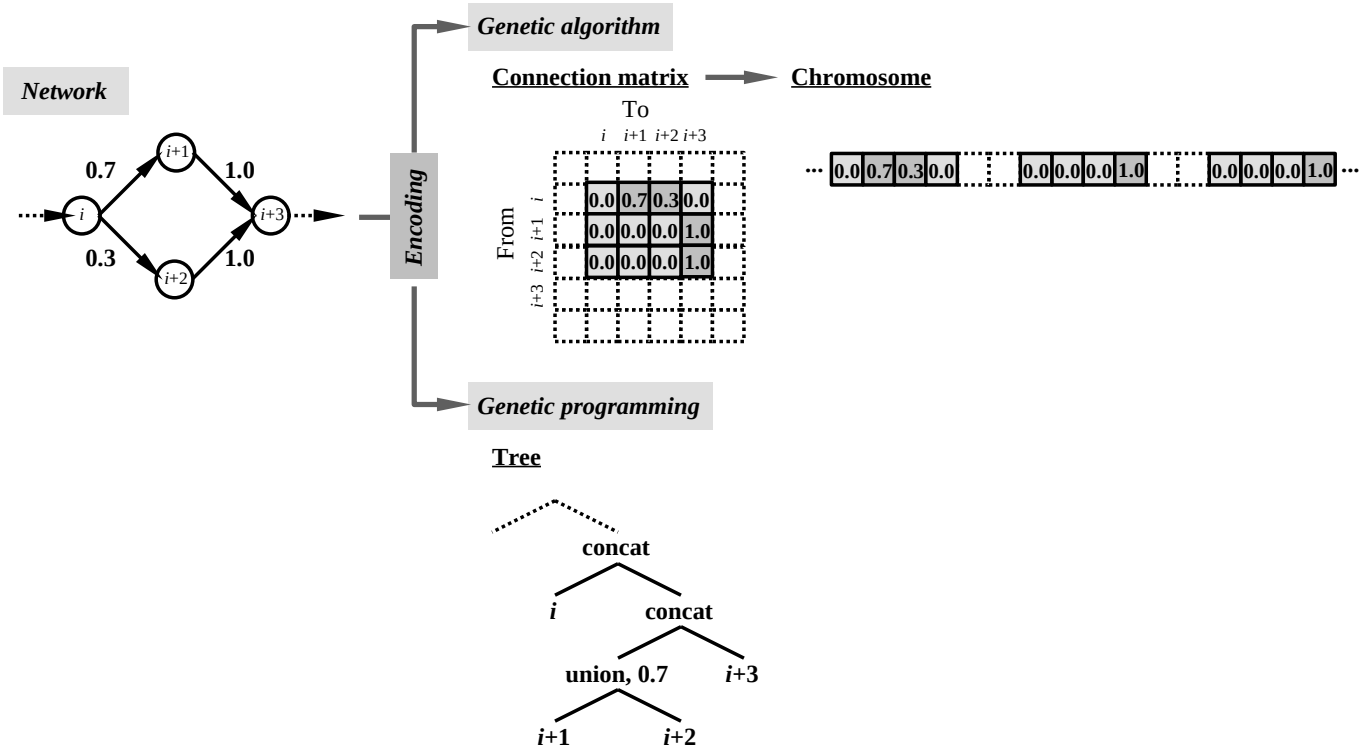

Figure 3.11: Comparison of encoding method of HMM network beween GA and GP

(Miller *et al.*, 1989). Figure 3.11 (upper) shows how to encode transition probabilities of an HMM subnetwork into a chromosome. First, transition probabilities are described by connection matrix. Transition probability from state i to j is assigned to element m_{ij} in the matrix. Next, a chromosome is constructed by connecting each row in the matrix. Figure 3.11 (upper) shows transition probabilities of a subnetwork that are frequently encoded into different parts of a chromosome. Therefore, the subnetwork is easily destroyed by the recombination operation of GA (destruction of building blocks).

Overview of a new approach

To solve the above problems, we designed a method for HMM construction as follows: (1) generation of a set of HMMs (motif HMMs) which correspond to each kind of motifs in given DNA sequences, (2) construction of motif pattern HMM which consists of motif HMMs using genetic programming (GP) (Koza, 1992; Koza, 1994). We call the former the motif extraction module and the latter the motif pattern extraction module.

We can solve the problem (A) by preparing a set of motif HMMs before the construction of motif pattern HMM. A large part of motif pattern HMM consists of subnetworks for

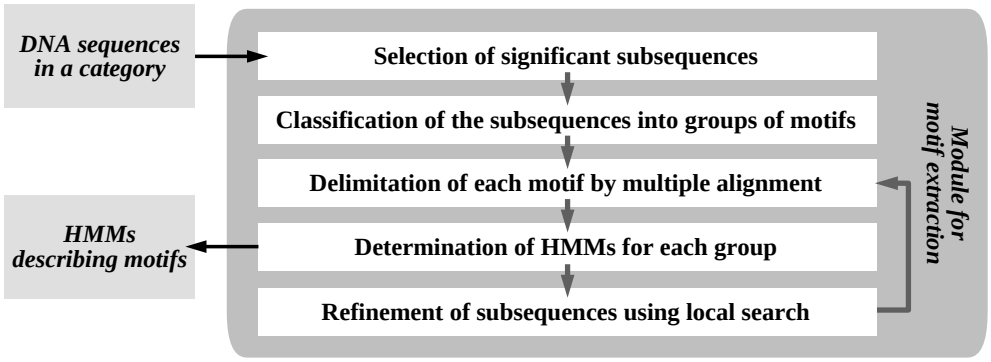

Figure 3.12: A method for design of motif HMMs

motif representations. On the other hand, Krogh *et al.* reported that a left-to-right HMM network is sufficient to represent motifs (Krogh *et al.*, 1994a). Therefore, the generation of left-to-right HMMs corresponding to each motifs in given sequences stands for the preparation of a large part of HMM network for motif pattern. By the preparation, we can start construction of motif pattern HMM from large network size, and can apply a method to a large amount of DNA sequences.

We adopted a method developed by Yada *et al.* (Yada *et al.*, in press) as a method for motif HMM generations (Figure 3.12). The method extracts motifs in the form of HMM from a given set of DNA sequences by the following steps: (1) selection of significant subsequences with fixed length (Pesole *et al.*, 1996) (2) classification of subsequences into groups of motifs (3) delimitation of each motif by multiple alignment (Ishikawa *et al.*, 1994) and statistical test (Wonnacott & Wonnacott, 1972) (4) construction of an HMM for each group and (5) redefinition of subsequences for building motifs using local search technique. Steps (3) to (5) are iterated trying to expand the motif regions until the given condition is satisfied. See section 3.2 for a detailed description of the algorithm. Yada *et al.* reported that the method could extract motifs from several sets of benchmark data (Frech *et al.*, 1997) with high accuracy (Yada *et al.*, in press). Further, the method could be successfully applied to a large amount of DNA sequence data (several hundred kilo bases in total) (Yada *et al.*, in press).

We can solve the problem (B) by applying GP. In GP, a network of HMM is encoded into a tree. Figure 3.11 (lower) shows how to encode transition probabilities of an HMM subnetwork into a tree. Components of HMM network are assigned to terminal nodes of a tree. Connection patterns of components are assigned to nonterminal nodes of a tree. HMM subnetwork in Figure 3.11 can be described by the combination of concatenation

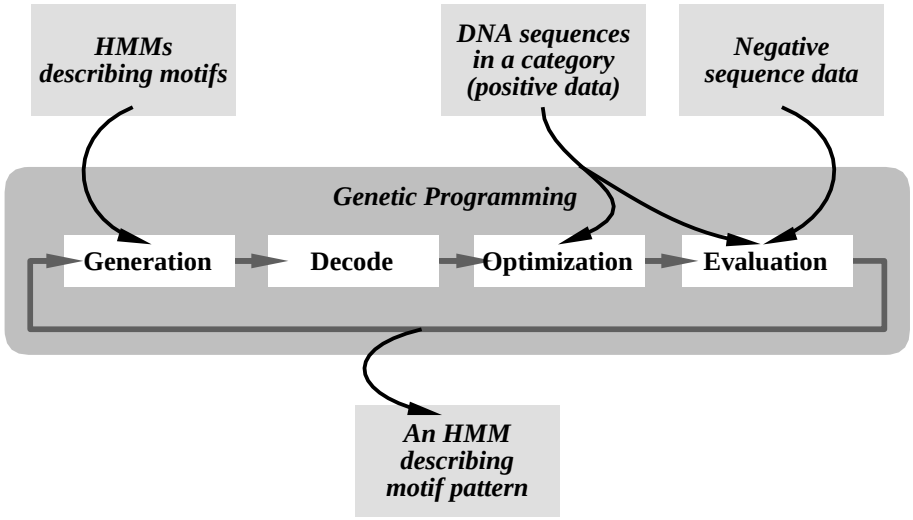

Figure 3.13: A method for design of motif HMMs

('concat') and probabilistic union ('union'). Value 0.7 assigned to a 'union' indicates transition probability from state i to i + 1. Transition probability from i to i + 2 is given by 1.0 – 0.7. Figure 3.11 (lower) shows that HMM subnetwork is encoded into a lump of subtree. Therefore, it becomes hard to destroy a subnetwork by genetic operations.

In this section, we discuss the validity of a method which constructs motif pattern HMM by using motif and motif pattern extraction modules. We introduce a detailed algorithm of the latter modules in the next subsection. After that, we report the results of computer experiments to construct a motif pattern HMM from a set of DNA sequences surrounding human acceptor sites. The size of data which the method was applied to is 122,360 bases, and this size can be regarded as a scale of practical problems in the field of genome biology. We also report the results of open test to validate the constructed HMM.

3.3.2 Method

We developed a method for the construction of a motif pattern HMM from given DNA sequences using GP 3.13. In the method, motif HMMs, positive and negative DNA sequences are given as inputs. The method outputs HMMs which represent motif pattern hidden in positive data. Positive data is randomly divided into training and test data. The division is performed every T_{renew} generations which is specified in advance. Such data division can prevent over learning of constructed HMMs against positive data.

We defined a probabilistic tree to describe HMMs. GP optimizes the tree structure so

as to represent motif pattern. Terminal nodes of the probabilistic tree correspond to components of HMM. Motif HMMs and an HMM for any a symbol are assigned to terminal nodes. Nonterminal nodes of probabilistic tree correspond to connection patterns of components. Concatenation, probabilistic union and probabilistic closure *etc.* are assigned to nonterminal nodes.

We implemented GP which is similar to cellular GA (Manderick & Spiessens, 1989). In cellular GA, individuals are placed on a two dimensional grid, the grid is transformed into a torus to reduce boundary effect. For an individual at a coordinate, a subpopulation which consists of individuals at neighbor sites is defined in the GA. Fitness calculation, selection and genetic operations are performed within the subpopulation. Since cellular GA is similar to island model (Sumida *et al.*, 1990), it can search large solution space.

Construction procedures of motif pattern HMM by GP are as follows. First, tree population is randomly generated at initial generation. Each tree is transformed into an HMM based on transformation rules, and the HMM parameters are optimized by Baum-Welch algorithm against positive training data. Next, HMMs are applied to positive and negative test data, and scores of positive training sequences are calculated by Viterbi algorithm (Rabiner, 1989). Based on the scores, we evaluated the performances of HMMs as a representation model for motif pattern. The results of the evaluation affects fitness of HMMs, and we select trees according to fitness values. Last, genetic operations such as recombination and mutations are performed on selected trees, and new tree population is generated. By the iteration of the above procedures, the method constructs HMMs which represent motif pattern hidden in positive data.

Encoding of hidden Markov models

We defined a probabilistic tree to describe HMMs (Figure 3.14). Components of HMM network (motif HMMs and an HMM for any a symbol) are assigned to terminal nodes. Connection patterns of components (concatenation, probabilistic union and probabilistic closure *etc.*)are assigned to nonterminal nodes. Nonterminal nodes consist of pattern and parameter description parts. The latter part is added as the need arises. Figure 3.14 (a) describes concatenation of s and t, and label "concat" is assigned to nonterminal node. Figure 3.14 (a) describes union of s and t, where appearance probability of s is a. Label "union, a" is assigned to nonterminal node. Figure 3.14 (c) describes closure of s, where repetition probability of s is a. Label "closure, a" is assigned to nonterminal node. Figure 3.14 (d) describes repetition of s one or more times, where repetition probability of s is a. Label "op1, a" is assigned to nonterminal node. Figure 3.14 (e) describes appearance of

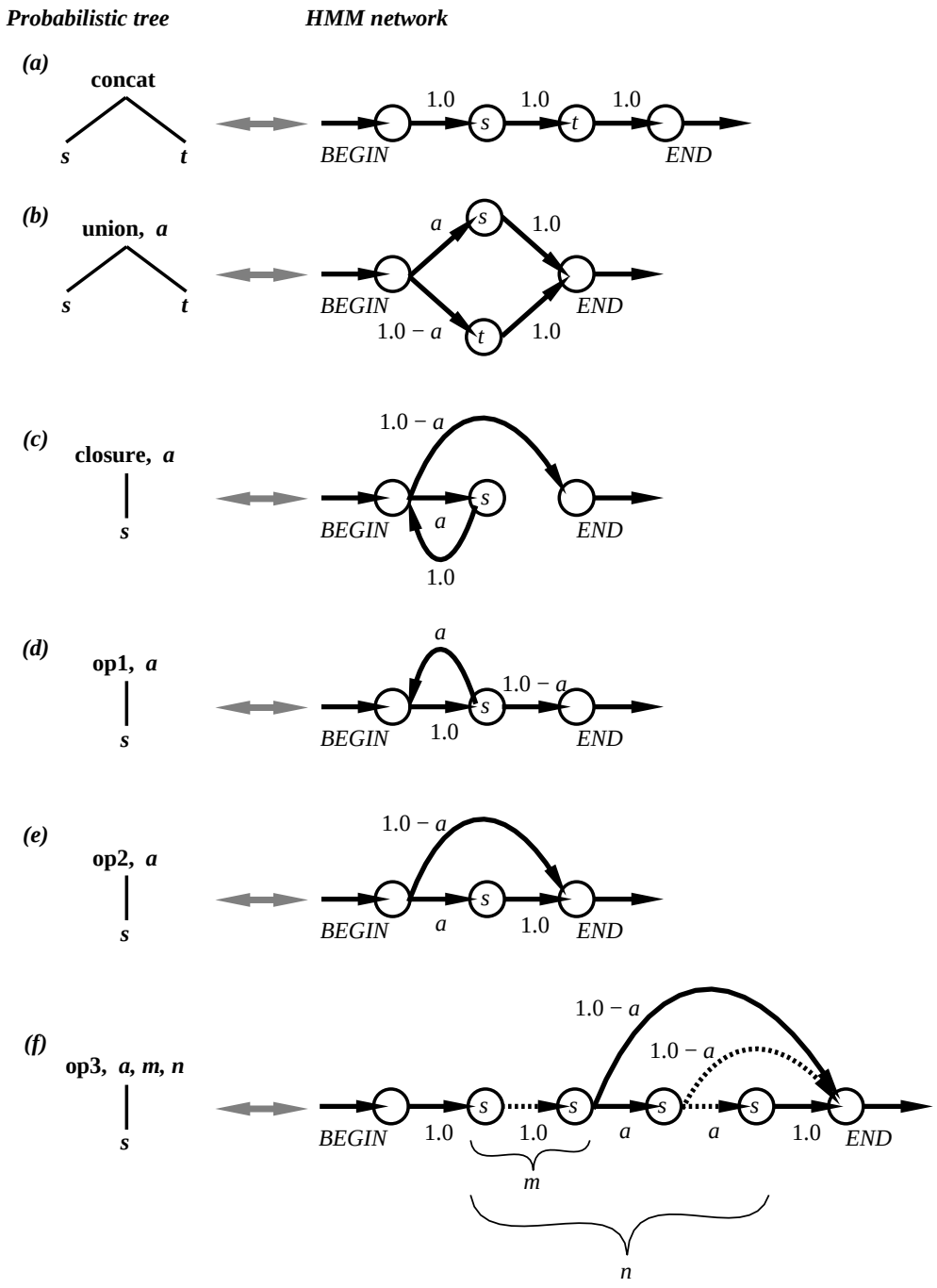

Figure 3.14: Probabilistic trees describing HMM networks

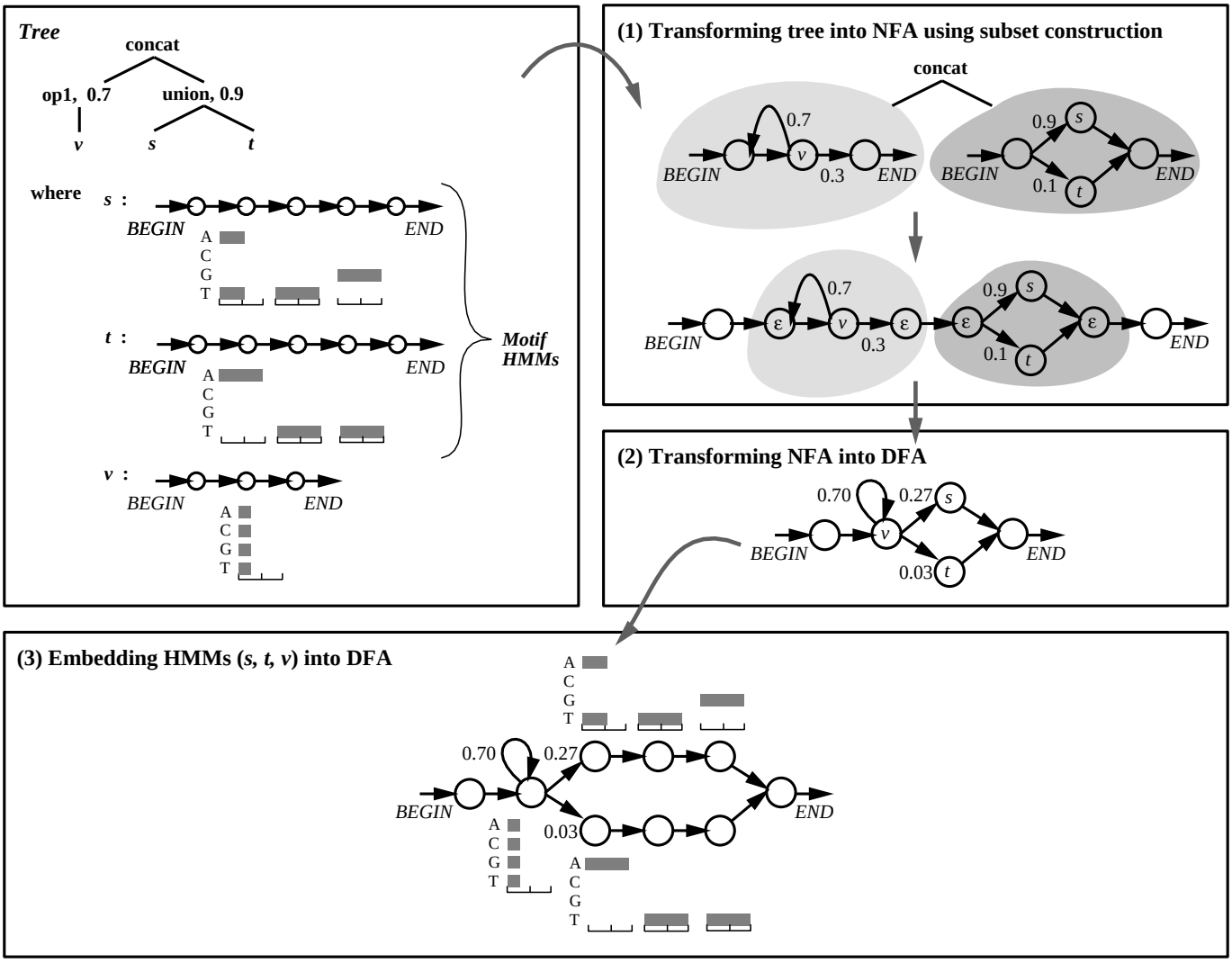

Figure 3.15: Decoding of a probabilistic tree

s zero or one time, where appearance probability of s is a. Label "op2, a" is assigned to nonterminal node. Figure 3.14 (e) describes repetition of s at least m times, but not more than n times, where repetition probability of s for all repetitions from m to n is a. Label "op3, a, m, n" is assigned to the nonterminal node. Hierarchical combination of the above descriptions enables us to encode more complex HMM network (see Figure 3.11 or 3.15).

Decoding of probabilistic trees

The procedures of transformation of probabilistic trees to HMMs are as follows (Figure 3.15). First, (1) by using subset construction (Rabin & Scott, 1959), nondeterministic finite automaton (NFA) is constructed from the probabilistic tree. We transform probabilistic

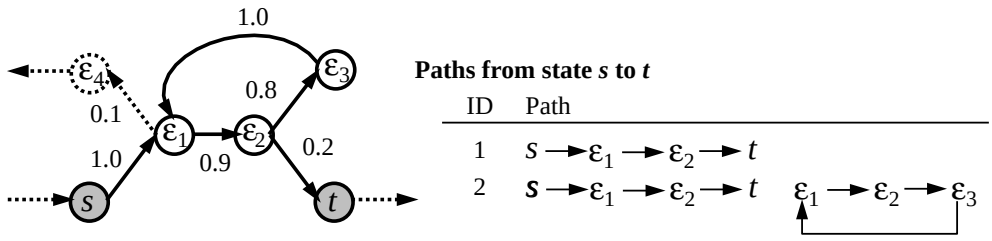

Figure 3.16: Recalculation of state transition probabilities

trees into NFAs according to transformation rules shown in Figure 3.14. State ϵ in Figure 3.15 indicates null state. Next, (2) NFA is transformed into deterministic finite automaton (DFA). Last, (3) HMMs (motif HMMs and an HMM for any a symbol) are embedded in the states of DFA. See Smith (Smith, 1983) for detailed description of transformation algorithm of non-probabilistic tree to DFA.

In the case of transformation of probabilistic trees to HMMs, we have to recalculate state transition probabilities at step (2). Transition probability $a_{s,t}$ from state s to t, both of which are connected via null states, is given by the following equations:

$$
\begin{aligned}
a_{s,t} &= \sum_i \alpha^i \beta^i \qquad (3.6)\\
&\text{with}\\
\alpha^i &= \prod_j p_j^i\\
\beta^i &= 1 \qquad \text{If } i \text{ has no loop paths}\\
&= \prod_k \frac{\prod_l q_l^{i,k}}{1 - \prod_l q_l^{i,k}} \qquad \text{Otherwise}
\end{aligned}
$$

where i is a path from state s to t via null states. Paths are discriminated by the combinations of paths not including loops and loops concerning the paths. p_j^i indicates probability of a state transition j on path i not including loops. $q_l^{i,k}$ indicates probability of a state transition l on loop k concerning path i. An example of the recalculation of state transition probabilities is as follows (see Figure 3.16). There are two kinds of paths from state s to t. One is a path (ID 1) which consists of a path not including loops. The other is a path (ID 2) which consists of ID 1 and a loop. Therefore, $\alpha^1\beta^1 = (1.0 \times 0.9 \times 0.2) \times 1 = 0.18$, $\alpha^2\beta^2 = (1.0 \times 0.9 \times 0.2) \times [0.9 \times 0.8 \times 1.0/(1.0 - 0.9 \times 0.8 \times 1.0)] \simeq 0.46$, then $a_{s,t} \simeq 0.18 + 0.46 = 0.64$.

Table 3.9: Parameters for evaluating the discrimination abilities of models

	Predicted positives (PP)	Predicted negatives (PN)
Actual positives (AP)	True positives (TP)	False negatives (FN)
Actual negatives (AN)	False positives (FP)	True negatives (TN)

Fitness and selection

We applied HMMs constructed from probabilistic trees to positive and negative test data and evaluated how valid each HMM is as the representation model of a motif pattern. Before the evaluation, Baum-Welch algorithm is used to optimize HMM parameters against positive training data. After that, Viterbi algorithm is applied to positive and negative test data using optimized HMM, and maximum logarithmic likelihood of each sequence are calculated. To evaluate the validity of HMM as motif pattern representation, we introduced correlation coefficient CC (Burset & Guigò, 1996) which is given by the following equation:

$$CC = \max_z \frac{(TP_z)(TN_z) - (FP_z)(FN_z)}{\sqrt{(PP_z)(PN_z)(AP)(AN)}} \tag{3.7}$$

CC varies from -1 to 1, and validity of HMM increases with increased value. z, which is a threshold for sequence discrimination, is determined so as to maximize CC. HMM regards sequences whose logarithmic likelihood is greater than z as positive examples and the others as negative examples. AP and AN are the number of positive and negative test data, respectively. TP_z is the number of positive data regarded as positive examples, and TN_z is the number of negative data regarded as negative examples. FP_z is the number of negative data regarded as positive examples, and FN_z is the number of positive data regarded as negative examples. PP_z is the number of data regarded as positive examples, and PN_z is the number of data regarded as negative examples. See Table 3.9 for relationship between the parameters.

In cellular GP, probabilistic trees are placed on a two dimensional grid, and subpopulations are defined for each tree. Fitness calculation and selection of a probabilistic tree are performed within a subpopulation which is defined by the tree. The subpopulation of a probabilistic tree on cell (i, j) is defined by probabilistic trees on neighbor cells $\{(x, y)|x \in X, y \in Y\}$ (where $X = \{i - 1, i, i + 1\}$, $Y = \{j - 1, j, j + 1\}$). The fitness $W_{k,l}$ of a probabilistic tree on cell (k, l) (where $k \in X, l \in Y$) is given by the following

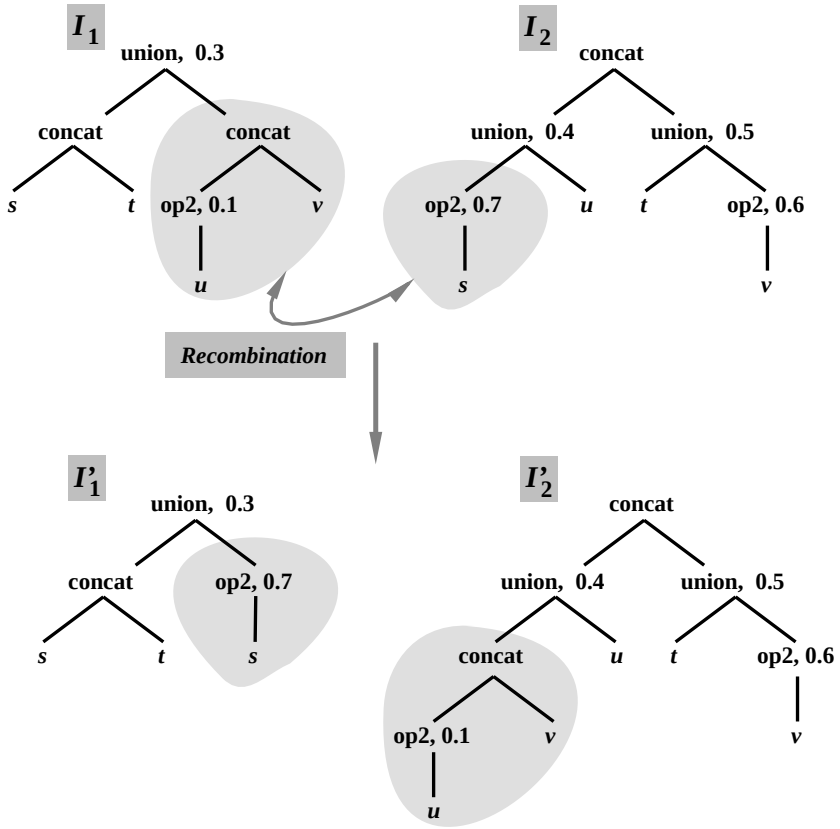

Figure 3.17: Recombination between probabilistic trees

equation:

$$W_{k,l} = \frac{CC_{k,l}}{\sum_x \sum_y CC_{x,y}} \tag{3.8}$$

where $CC_{k,l}$ is the correlation coefficient of HMM (probabilistic tree) on cell (k, l). Fitness is converted by sigma and linear scalings. Statistics which are necessary for the scalings are calculated within the subpopulation. According to fitness $W_{k,l}$ (where $k \in X, l \in Y$), roulette strategy is performed for the selection of probabilistic tree on cell (i, j). We adopted elite strategy: probabilistic trees whose correlation are above 5 % in rank (elites) survive on the same cells.

Genetic operations

We designed recombination and mutations as genetic operations of probabilistic trees. These operations are probabilistic operations based on parameters called recombination rate and mutation rate and can generate new probabilistic trees.

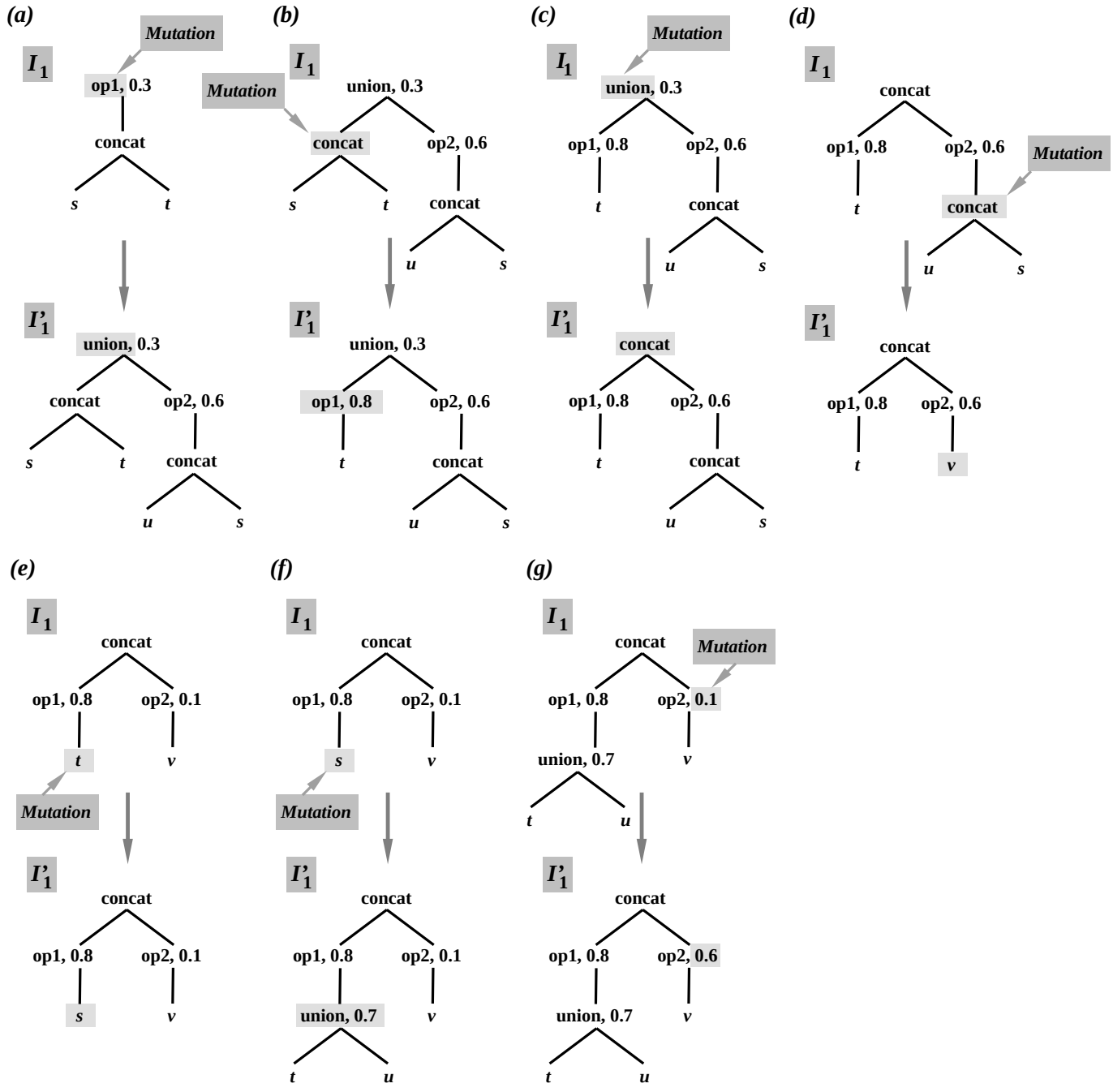

Figure 3.18: Mutations on a probabilistic tree

A probabilistic tree on a cell can recombinate with probabilistic trees placed at the neighbor sites. In recombination, subtrees are exchanged between two trees, and depths of the trees are occasionally changed (Figure 3.17). Figure 3.17 shows that the depth of tree I_1 is changed from 4 to 3, and that of tree I_2 is changed from 4 to 5 by recombination.

Procedures of recombination operation are as follows. First, tracing nodes of a probabilistic tree on cell (i, j) hierarchically, we determine whether a subtree below a node becomes a target of recombination based on the recombination rate. When subtree ST_1 is chosen as the target, we randomly select a tree from subpopulation $\{(x, y)|x \in X, y \in Y\}$ (where $X = \{i-1, i, i+1\}$, $Y = \{j-1, j, j+1\}$) defined by cell (i, j). Next, we choose another target of subtree ST_2 tracing nodes of the selected tree hierarchically. Last, subtrees ST_1 and ST_2 are exchanged. In the case of recombination occurring to an elite, no subtrees are exchanged, and a subtree of the elite is copied into the other tree.

Tracing nodes of a probabilistic tree hierarchically, we determine whether mutation occurs to a node based on the mutation rate. When mutation occurs, we randomly determine whether the node becomes a nonterminal node or not. Based on the determination, sometimes a subtree is generated or removed. These generation and deletion are performed at random. Mutation does not occur to elites.

Mutation operations are largely classified into three groups: (1) mutations occurring to pattern discrimination parts of nonterminal nodes [Figure 3.18 $(a) \sim (d)$], (2) mutations occurring to terminal nodes [Figure 3.18 (e) and (f)] and (3) mutations occurring to parameter discrimination parts of nonterminal nodes [Figure 3.18 (g)]. Group (1) is classified into two subgroups: (1A) mutations from nonterminal nodes to nonterminal nodes [Figure 3.18 $(a) \sim (c)$] and (1B) mutations from nonterminal nodes to terminal nodes [Figure 3.18 (d)]. The former is further classified into the following: (i) subtrees are generated [Figure 3.18 (a)], (ii) subtrees are removed [Figure 3.18 (b)] and (iii) subtrees do not change [Figure 3.18 (c)]. In Figure 3.18 (a), "op1" mutates into "union". "op1" has one child while "union" has two children (Figure 3.14), so a subtree is newly generated. Child of "op1" becomes one of the children of "union". In Figure 3.18 (b), "concat" mutates into "op1". "concat" has two children while "op1" has one child (Figure 3.14), so either subtree of "concat" is removed. In Figure 3.18 (c), "union" mutates into "concat". These have the same number of children (Figure 3.14), so the probabilistic tree after mutation inherits these children. Group (2) is classified into two subgroups: (2A) mutations from terminal nodes to terminal nodes [Figure 3.18 (e)] and (2B) mutations from terminal nodes to nonterminal nodes [Figure 3.18 (f)].

3.3.3 Computer Experiments

We prepared sets of DNA sequences surrounding human acceptor sites (positive data) and pseudo-acceptor sites (negative data) and applied the method (combination of motif and motif pattern extraction modules) to the construction of HMMs which represent motif pattern hidden in positive data. Further, we prepared open test data and evaluated the validity of the HMMs. Since GP is a stochastic algorithm, the motif pattern extraction module was performed five times. HMMs being the solution for each trial are selected by the following procedures. (1) For each generation after $T_{solution}$ at a trial, HMMs which have maximum correlation coefficient are collected. (2) Using collected HMMs, correlation coefficients (Equation 3.7) against positive and negative data are recalculated. (3) We regarded an HMM possessing maximum coefficient as a solution of the trial. Solution HMMs were applied in open test.

Data

We used DNA sequence data (Reese & Kulp, 1996) which is collected for benchmark tests of human gene recognition programs. The data is highly reliable because statistical biases and errors are removed. Focusing on 1,748 acceptor sites in the data, we extracted the upstream sequences (70 bases long) and assigned them as positive data. We extracted 856 upstream sequences (70 bases long) of pseudo-acceptor sites (di-nucleotide AG) from the data and assigned them as negative data. Each sequence of negative data has high scores calculated by position weight matrix (PWM) (Bucher, 1990) which models the upstream region of acceptor sites. PWM is widely used as a conventional model for motif pattern representation.

As open test data, we used DNA sequence (301,692 bases long) of APP locus on human chromosome 21 (Sakaki & Hattori, 1997). In the sequence, 18 acceptor sites are annotated, and 20,524 pseudo-acceptor sites are observed. The above positive data does not include the 18 acceptor sites.

Parameters

In the generation of motif HMMs, we used default parameter values (Yada *et al.*, in press) of the motif extraction module. In the construction of motif pattern HMMs, we defined a population consisting of 36 probabilistic trees and placed them on 6×6 grid. Searching generation was 50, and recombination and mutation rate were 0.05 and 0.02, respectively. We assigned 30% of positive data as training data and the remainder as test

data. The assignment was renewed every 2 generations (T_{renew} = 2). For each generation after $T_{solution}$, we selected solution HMMs.

3.3.4 Results and Discussion

Generation of motif HMMs

Figure 3.19 shows six motif HMMs which are generated from upstream sequences of human acceptor sites (positive data: 1,748 sequences). In Figure 3.19, typical sequence patterns (consensus sequences) for each motifs are also shown. Consensus sequences are described by regular expression which is commonly used in the field of molecular biology. Regular expression [ab](x) matches sequence fragments, whose lengths are x bases, consisting of the combination of base a and b. IDs attached to motifs indicate rank based on statistical significance.

Focusing on the consensus sequences of motifs in Figure 3.19 and their appearance positions in positive data, it is considered that motifs 1 and 6 correspond to sequence patterns from polypyrimidine tract to acceptor sites, motifs 2 and 3 correspond to branch point motifs, and motif 4 corresponds to polypyrimidine tract motif. These results suggest that polypyrimidine tract and branch point motif can be classified into some groups from the viewpoint of sequence similarity. Though motif 5 is an unknown motif, there is every possibility that it is a new one which is specially observed in the vicinity of human acceptor sites.

Table 3.10 summarizes the number of sequences in which each motif in Figure 3.19 was found. Branch point motifs 2 and 3 were found in 10.4% (182/1,748) and 0.9% (15/1,748) of positive sequences data. Since many motifs are only found in relatively small portion of data, it becomes difficult to extract such motifs. Table 3.10 shows that the motif extraction module extracted such motifs with high sensitivity.

Construction of motif pattern HMMs

Construction process Figure 3.20 shows changes of average and maximum correlation coefficients during searching generation at the five trials. The figure indicates that GP worked well in the processes of HMM construction. Namely, for all trials, average and maximum correlation coefficients rose dramatically at the early stages of searching generation, and fluctuating but stable situations of the coefficients are observed after the stages. These phenomena are the main feature of GP.

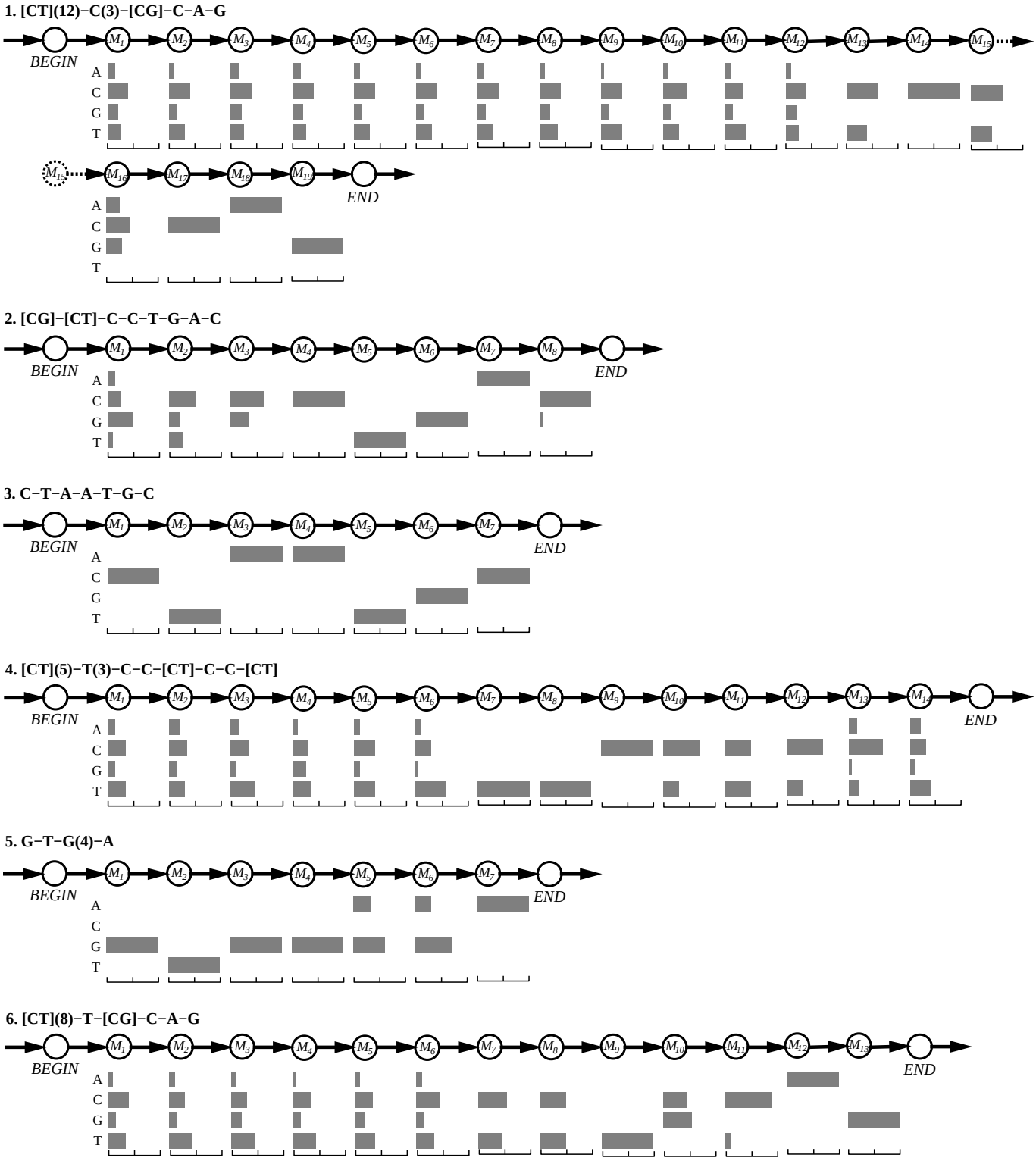

Figure 3.19: HMM descriptions of motifs extracted from upstream of acceptor sites

Table 3.10: Number of sequences in which each motif was found

Rank	Consensus sequence	# of seq.
1	[CT](12)-C(3)-[CG]-C-A-G	470
2	[CG]-[CT]-C-C-T-G-A-C	182
3	C-T-A-A-T-G-C	15
4	[CT](5)-T(3)-C-C-[CT]-C-C-[CT]	845
5	C-T-G(4)-A	58
6	[CT](8)-T-[CG]-C-A-G	438

Close test Table 3.11 summarizes the results of close tests using HMMs obtained by five trials. The table indicates that the method constructed HMMs which discriminate between positive and test data with high accuracy. Sensitivity (Sn) in Table 3.11 is given by TP/AP (Table 3.9), discrimination ability of HMM increases with increased value. To compare the discrimination ability, Table 3.11 shows Sn, Sp, CC of PWM and HMM constructed by HMMER program (Eddy, 1995). PWM is a motif pattern representation model which summarizes base contents at each position in sequences in the form of matrix. HMMER HMM is a model which is removed position dependency from PWM. For all tests, Table 3.11 shows that HMMs constructed by our method are superior in discrimination ability to PWM and HMMER HMM.

Open test The results of open test are shown in Figure 3.21. This graph shows that HMMs constructed by our method are valid as motif pattern representation. In Figure 3.21, relationships between true positives (TP) and false positives (FP) of our HMMs are shown in broken lines. Relationships between TP and FP of PWM and HMMER HMM are shown in solid lines. For all TPs, FPs of our HMMs are smaller than those of PWM and HMMER. Moreover, our HMMs can detect 14 $\sim$ 16 acceptor sites while PWM and HMMER HMM can detect 12 and 14 acceptor sites, respectively.

Motif pattern HMMs An example of motif pattern HMM which was constructed at the second trial is shown in Figure 3.22. In Figure 3.22, thick solid lines, solid lines, thin solid lines and broken lines indicate state transition probabilities $a \geq 0.9$, $0.5 \leq a < 0.9$, $0.1 \leq a < 0.5$ and $a < 0.1$, respectively. Shaded boxes indicate motif HMMs (Figure 3.19) generated by motif extraction module.

The HMM in Figure 3.22 indicates that co-occurrence relationship between motifs ex-

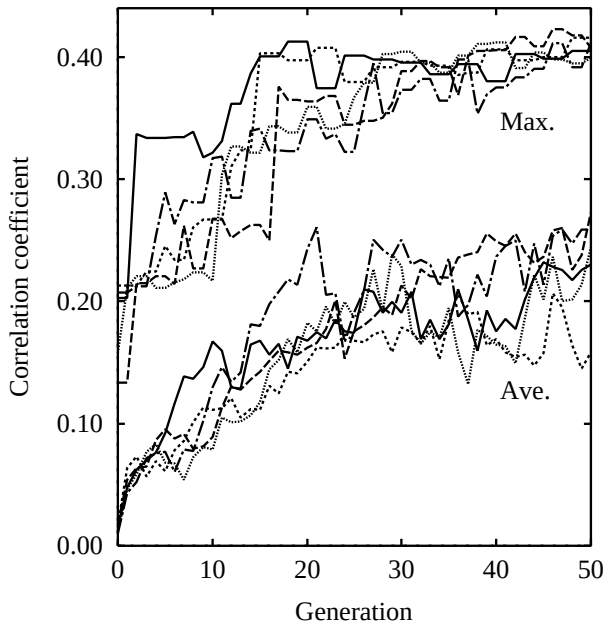

Figure 3.20: Process of HMM generation by genetic programming

ists in the upstream region of acceptor sites, and the region can be classified into three groups from the viewpoint of the relationship. The first group suggests that the GC-rich region (state M_A) is located upstream of motif 1 or 6. The second group suggests that the GC-rich region (state M_A) is located upstream of motif 2 or 3, and the C-rich region (state M_C) appears downstream of the motif, and motif [CT](3)-C-[CT](2)-C-[CG]-C-A-G is located downstream of the C-rich region. Motif [CT](3)-C-[CT](2)-C-[CG]-C-A-G, which is a C abundant motif, corresponds to the region from the polypyrimidine tract to the acceptor site. Since the transition probability of self-loop at M_C is 0.93, Markov analysis (Wonnacott & Wonnacott, 1972) reveals that the average distance between motif 2 or 3 and motif [CT](3)-C-[CT](2)-C-[CG]-C-A-G is 14.3 bases $[1/(1-0.93)]$. The third group suggests that the CT-rich region (state M_E) is located downstream of the uniform base contents region (state M_D), and motif [CT](6)-C-A-G appears downstream of the CT-rich region. Output symbol distribution at M_E suggests that M_E corresponds to a part of the polypyrimidine tract. Since the transition probability of self-loop at M_E is 0.92, Markov analysis reveals that the average base length corresponding to M_E is 12.5 $[1/(1-0.92)]$.

Efficiency of the construction of motif pattern HMMs It is very important for the efficient construction of motif pattern HMMs to prepare proper sets of probabilistic trees (Fig-

Table 3.11: Results of close tests of generated HMMs

		Sn	Sp	CC
Our HMMs	1	0.934	0.758	0.405
	2	0.961	0.748	0.408
	3	0.938	0.757	0.405
	4	0.934	0.756	0.398
	5	0.958	0.749	0.408
PWM		0.797	0.781	0.344
HMMER HMM		0.893	0.758	0.359

ure 3.14). Probabilistic trees corresponding to subnetworks enclosed with outline boxes (Figure 3.22) are described in the form of Figure 3.14 (f). If we did not prepare it, a hierarchical pile of Figure 3.14 (a) is necessary to describe the subnetworks. Namely, many times of genetic operations are necessary to construct such subnetworks, and construction efficiency of motif pattern HMMs significantly decreases. Therefore, it is very important for the efficient construction of motif pattern HMMs by GP to list a set of typical motif patterns and prepare their probabilistic trees. From the similarity viewpoint, introduction of automatically defined functions (ADF) (Koza, 1992; Koza, 1994) would be effective for the efficiency. ADF can construct the combinations of probabilistic trees and use them as components of motif pattern HMMs.

Though we did not adopt it in the current study, introduction of duplication operation as genetic operations would also be effective for the efficient construction of motif pattern HMMs (Figure 3.23). When duplication occurs at a node, the subtree below the node is copied. Original and copied subtrees are connected via concatenation or probabilistic union, in which the connection type is determined at random. This operation means that a subnetwork of HMM is copied, and parallel or serial connection is generated between original and copied subnetworks. In other algorithms for HMM network construction (Fujiwara *et al.*, 1994; Yada *et al.*, 1996b), procedures which are similar to the duplication have been adopted as efficient procedures of network generation.

3.3.5 Conclusion

We developed a method for the construction of HMMs which represents motif patterns hidden in a set of DNA sequences. First, the method generates a set of HMMs each of

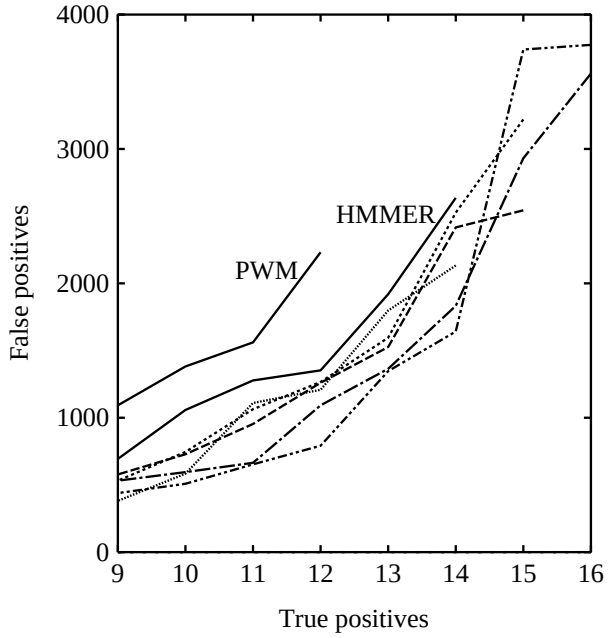

Figure 3.21: Results of open tests of generated HMMs

which represents a motif in the sequences using an algorithm developed by Yada *et al.* (Yada *et al.*, in press). Next, the method constructs HMMs which represent motif patterns by combining motif HMMs. We applied the method to the construction of motif pattern HMMs from DNA sequences surrounding human acceptor sites. Further, we identified acceptor sites in uncharacterized human DNA sequence using motif pattern HMMs.

Our computer experiments revealed the following. (1) the method successfully constructed hidden Markov representations of motif patterns from a large amount of data (122,360 bases). (2) the method successfully generated hidden Markov representations of motifs observed in a small subset of data as well as ones of motifs observed in a large subset of data. (3) GP worked well in the construction processes of motif pattern HMMs. (4) the constructed HMMs are capable of identifying motif patterns in uncharacterized DNA sequences with high accuracy. (5) the constructed HMMs are highly suggestive of new knowledge of motif patterns.

We are trying to apply the method to two challenging problems in the field of molecular biology. One is the problem of gene structure prediction, and the other is the problem of gene function prediction.

The problem of gene structure prediction is a problem of predicting gene structures in uncharacterized DNA sequences. Several kinds of prediction methods based on mathe-

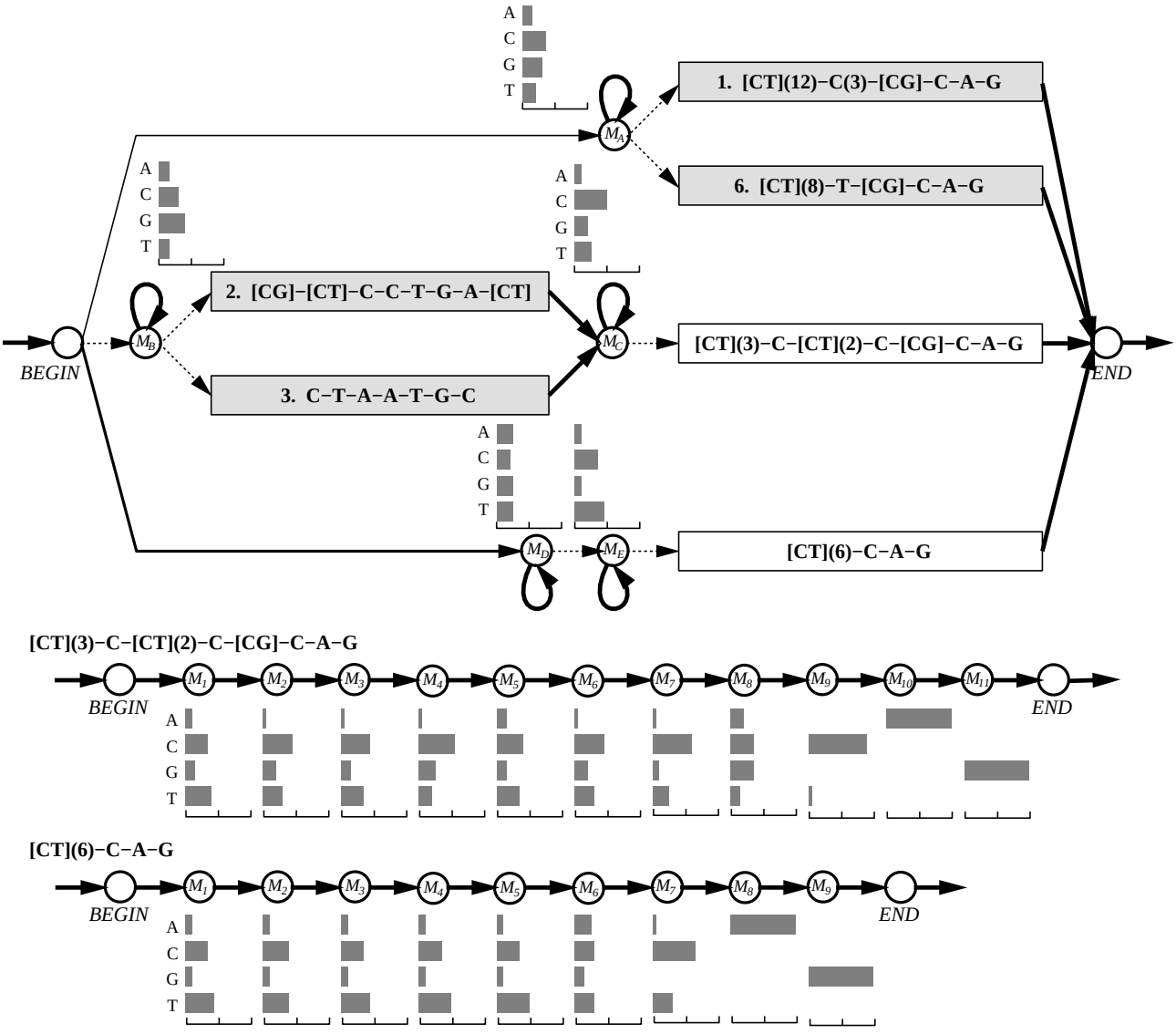

Figure 3.22: HMM descriptions of motif patterns extracted from upstream of acceptor sites

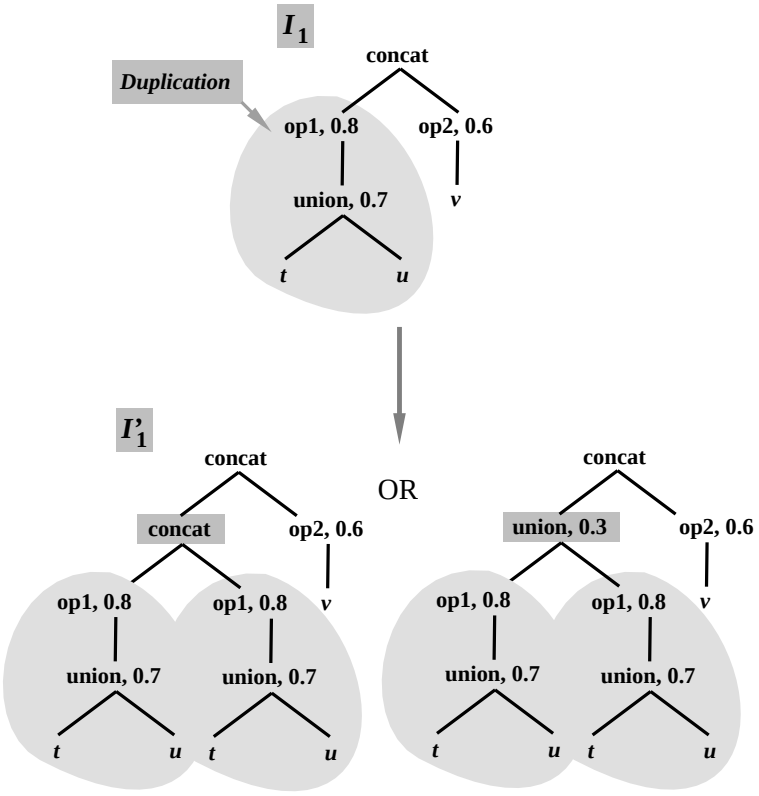

Figure 3.23: Duplication on a probabilistic tree

matical models have been developed. Among them, the method based on HMM marked high accuracy (Krogh *et al.*, 1994b; Yada & Hirosawa, 1996b; Burge & Karlin, 1997; Henderson *et al.*, 1997; Krogh, 1997). However, the prediction accuracy of boundaries of gene structures is insufficient (acceptor sites are also one of the boundaries) because motif patterns surrounding boundaries are not well characterized. Therefore, by applying the method to such regions, we expect the discovery of new knowledge of motif patterns and construction of a reliable model for boundary prediction.

The problem of gene function prediction is a problem of predicting gene function from DNA sequence information. Recently, a new approach, the inference of gene functions from their promoters, was proposed (Bucher *et al.*, 1996; Fondrat & Kalogeropoulos, 1996; Yada *et al.*, 1997). Promoters are located upstream of genes and possess motif patterns to regulate gene expression. All genes of an organism are not always expressed, and promoters regulate so as to express a proper subset of genes as the need (stress response, starvation response, *etc.*) arises. Therefore, similar motif patterns should be observed in promoters whose genes are expressed against a need. Namely, if we can discriminate motif patterns against a need, we can infer in what situation the gene will be expressed.

However, previous methods can extract only a set of motifs from promoters due to the underdevelopment of information processing technique for the extraction of motif patterns from DNA sequences. Therefore, by applying the method to promoter data, we expect the extractions of motif patterns specially found in a subset of promoters and establishment of a new gene prediction method.