Vol.2016-MPS-108 No.7
Vol.2016-BIO-46 No.7
2016/7/4

# 文字列の集合上の確率分布における中央文字列および中心文字列に対する整数計画問題

林田 守広[1,a)]　小谷野 仁[2]

**概要**：数や数スベクトルからなるデータセットに対して，平均はデータの中心を把握するための最も基本的な尺度である．しかし，文字列からなるデータセットに対しては平均が定義されないため，データの中心の尺度として，中央文字列や中心文字列がしばしば用いられる．しかし，数データに対する平均を計算することと異なり，文字列データに対する中央文字列や中心文字列を求めることは簡単ではなく，中心文字列については厳密解を求めることが保証されているアルゴリズムは知られていない．本研究において，我々は，まず，文字列データに対する中央文字列と中心文字列の定義を，所与のアルファベットから作られる全ての文字列の集合上の確率分布に対する定義に一般化する．これは，数データの平均が数や数ベクトルの集合上の確率分布の期待値に一般化されることに対応している．我々は，次に，整数線形計画法を用いて，文字列の集合上の確率分布に対する中央文字列と中心文字列の厳密解を求める方法を開発する．これらの方法は，文字列の集合がレーベンシュタイン距離によって距離空間をなしている場合には，レーベンシュタイン距離が三角不等式を満たすことを利用して，より高速なものに改良される．最後に，数値実験を行い，提案した方法の実際の応用における有効性を検討する．

## 1. はじめに

データ解析において平均は基本的な統計手段である．本研究では平均が適用できない文字列の集合を扱う．DNA および RNA を構成する塩基配列あるいはタンパク質アミノ酸配列は文字列として表現される．これらの生物学的配列は急速にデータ量を増加させており知識獲得のための手法が求められる．生物進化においては共通祖先の配列を見つける目的に利用でき，またタンパク質配列に対しては機能モチーフの同定に利用できる．画像認識の分野においても OCR の事後処理 [1] や形状認識 [2] に利用されている．さらに生物学的配列の分類やクラスタリングに適用例がある [3].

文字列の集合に直接平均を適用することはできないため，その中心を表現するためのいくつかの尺度が提案されている．一つは中央文字列と呼ばれ，集合に含まれる各文字列との距離の和を最小にする文字列として定義される [4]．もう一つは中心文字列と呼ばれ，集合に含まれる各文字列との距離の最大を最小にする文字列として定義される [5]．文字列間の非類似度としては，レーベンシュタイン距離 [6]，ハミング距離 [7]，ジャロ-ウィンクラ距離 [8] などが提案されている．ただし，ジャロ-ウィンクラ距離は三角不等式を満たさないことが知られている．文字列 $s$, $t$ 間のレーベンシュタイン距離は挿入，削除，置換の編集を許し，動的計画法によって多項式時間 $O(|s||t|)$ で計算可能である．ここで $|s|$ は文字列 $s$ の長さを表す．ハミング距離もまた中央文字列や関連する問題に適用される [9], [10]．ハミング距離に対しては中心文字列の候補を削減する手法が開発されている [11]．しかし彼らのパラメータ化された手法ではレーベンシュタイン距離での中心文字列を見つけることはできない．この他，自然言語処理，著者属性などの分野で用いられるランク距離と呼ばれる距離の下で中央文字列を求める遺伝的アルゴリズム [12] が開発されている．

レーベンシュタイン距離の下で文字列の有限集合に対する中央文字列および中心文字列を求める問題は NP 完全であり [13]，構成する文字の種類が 2 つだけであっても NP 完全であること [14], [15] が示されている．さらに関連する CSCE 問題 (consensus string problem with consensus error) も距離関数が三角不等式を満たすとき NP 完全であることが示されている [16]．一方でレーベンシュタイン距離の下で中央文字列を求める動的計画法による厳密解法が提案されている [17] が，長さ $n$ の $N$ 本の文字列に対して

[1] 京都大学化学研究所
Gokasho, Uji, Kyoto 611-0011, Japan
[2] 理化学研究所生命システム研究センター
2-2-3, Minatojima Minami-cho, Chuo-ku, Kobe 650-0047, Japan
[a)] morihiro@kuicr.kyoto-u.ac.jp

$N$ 次元配列が必要であり，$O(n^N)$ の時間および空間計算量となり，例えば $n = N = 10$ のとき，$10^{10} \cdot 4$ B $= 40$ GB のメモリが少なくとも必要となる．

このため，現実的な時間で中央文字列を近似する手法が提案されてきている．入力となる文字列がお互いに類似するとき，動的計画法による最適な経路は多次元配列の対角に近くなると考えられるため，最適な経路となる候補を対角周辺に限定する手法が提案された [18]．Casacuberta らは，長さ 0 の文字列から開始し，誤差を最小化する文字を繰り返し追加していく貪欲アルゴリズムを提案した [19]．Jiang らは，文字列を一つずつ現在の近似中央文字列に加えていくオンラインアルゴリズムを提案した [20]．Olivares-Rodríguez らは文字列間に条件付き確率を定義し，EM アルゴリズムによって近似中央文字列を求めた [21]．Abreu らは，現在の文字列にスコアが最大となる編集操作をスコアが増加しなくなるまで繰り返す手法を提案した [22]．これらの手法は近似的な中央文字列を出力するが，厳密な中央文字列を出力できる手法はほとんど提案されていない．中心文字列については，厳密解を求める手法は提案されていない．

本研究では，整数線形計画法を用いた中央文字列および中心文字列を求める厳密解法を提案する．さらに，文字列の集合上の確率分布 [23] を導入し，この確率分布に対する中央文字列および中心文字列を定義することで，レーベンシュタイン距離の下での確率分布に対する中央文字列および中心文字列を求める手法を提案する．さらにレーベンシュタイン距離は三角不等式を満たすので，三角不等式による制約を加えた整数線形計画問題も提案する．最後に提案手法の効率性を検証するためにいくつかの確率分布について計算機実験を行った結果を示す．

## 2. 提案手法

レーベンシュタイン距離はよく使われる基本的な編集距離であるので本研究でもこの距離を扱う．本節では，レーベンシュタイン距離と中央文字列，中心文字列について定義した後，文字列の集合上の確率分布への一般化および提案手法である整数線形計画問題による定式化を説明する．以下では，$\mathcal{A} = \{a_1, \cdots, a_z\}$ を $z$ の異なる文字からなるアルファベットとする．例えば DNA 塩基配列については，$\mathcal{A} = \{A, T, G, C\}$ となる．$\mathcal{A}^*$ を $\mathcal{A}$ の文字を任意有限個並べた文字列のすべての集合とする．また $|s|$ を文字列 $s$ $(\in \mathcal{A}^*)$ の長さとする．

### 2.1 レーベンシュタイン距離

文字列 $s$ と $t$ の間のレーベンシュタイン距離 $d(s,t)$ は，$s = s_1 \cdots s_n$ から $t = t_1 \cdots t_m$ まで文字列を変換するときの編集操作の最小コストとして定義され，$\epsilon$ を空文字，$\gamma(s_i \to t_j)$，$\gamma(s_i \to \epsilon)$，$\gamma(\epsilon \to t_j)$ をそれぞれ一文字分の置換，削除，挿入のコストとするとき，次の動的計画法によって計算できる [24]．

$$D[0,0] = 0 \tag{1}$$

$$D[i,j] = \min \begin{cases} D[i-1, j-1] + \gamma(s_i \to t_j) \\ D[i-1, j] + \gamma(s_i \to \epsilon) \\ D[i, j-1] + \gamma(\epsilon \to t_j) \end{cases} \tag{2}$$

このとき $D[n,m]$ の値がレーベンシュタイン距離 $d(s,t)$ となる．

### 2.2 中央文字列および中心文字列

$\mathcal{A}^*$ 上の $N$ 本の文字列 $s^{(k)}$ $(k = 1, \cdots, N)$ に対して，中央文字列は

$$\mathrm{argmin}_{t \in \mathcal{A}^*} \sum_{k=1}^{N} d(t, s^{(k)}) \tag{3}$$

によって定義される．中心文字列は

$$\mathrm{argmin}_{t \in \mathcal{A}^*} \max_{k \in \{1, \cdots, N\}} d(t, s^{(k)}) \tag{4}$$

によって定義される．

$\mathcal{A}^*$ 上の確率分布 $p(s)$ が与えれられたとき，$p(s)$ に対する中央文字列および中心文字列をそれぞれ

$$\mathrm{argmin}_{t \in \mathcal{A}^*} \sum_{s \in \mathcal{A}^*} p(s) d(t,s), \tag{5}$$

$$\mathrm{argmin}_{t \in \mathcal{A}^*} \max_{s \in \mathcal{A}^*} p(s) d(t,s) \tag{6}$$

によって定義する．もしすべての $k = 1, \cdots, N$ に対して $p(s^{(k)}) = \frac{1}{N}$ であり，それ以外 $(s \notin \{s^{(k)}\})$ は $p(s) = 0$ なら，式 (3)，(4) は式 (5)，(6) とそれぞれ等価になる．

### 2.3 整数線形計画問題への定式化

レーベンシュタイン距離の下で中央文字列および中心文字列を求める問題は NP 困難であること [14], [15] が知られているので，効率の良いアルゴリズムの開発が進んでいる整数線形計画法を利用する．もし文字列 $t$ と $s^{(k)}$ との間のレーベンシュタイン距離 $d(t, s^{(k)})$ を線形式で表現できるのであれば，中央文字列 $t$ を整数線形計画法で求めることができる．しかし，動的計画法における配列 $D[i,j]$ の値を直接整数線形計画問題として表現することは，式 (2) における最小値の選択を含むため困難である．

$\mathcal{A}^*$ 上の確率分布 $p(s)$ が与えられたとき，$p(s) > 0$ を満たす文字列 $s$ の数は $\sum_{s \in \mathcal{A}^*} p(s) = 1$ より有限であり，その数を $N$ とする (つまり $k = 1, \cdots, N$ に対して，$p(s^{(k)}) > 0$)．線形計画問題における変数は実数値を取るため，アルファベット $\mathcal{A}$ に含まれる文字を $1, \cdots, |\mathcal{A}|$ の整数として表現する．$s_i^{(k)}$ $(i = 1, \cdots, n_k)$ は $1, \cdots, |\mathcal{A}|$ のいずれかの値と

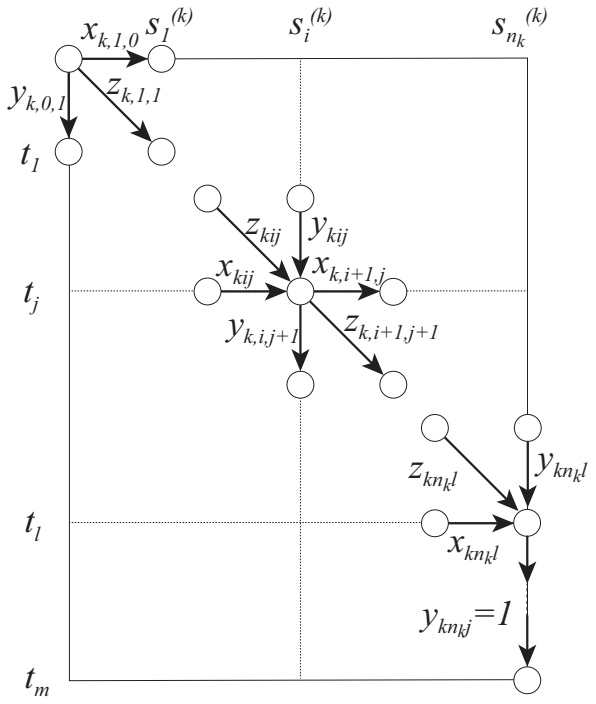



**図 1** 整数線形計画問題における文字列 $s^{(k)}$ と $t$ の間のレーベンシュタイン距離の計算．変数 $x_{kij}$，$y_{kij}$，$z_{kij}$ はそれぞれレーベンシュタイン距離計算における動的計画法での経路になっていれば 1 をとり，そうでなければ 0 をとる．変数 $l$ は最適文字列 $t$ の長さを表し，$j > l$ の $j$ に対して $y_{kn_kj} = 1$ を要請する．

して与えられる定数であり，長さ $n_k$ の文字列 $s^{(k)}$ の $i$ 番目の文字を表す．$t_j$ $(j = 1, \cdots, m)$ は $1, \cdots, |\mathcal{A}|$ のいずれかの値をとる変数であり，中央文字列あるいは中心文字列の $j$ 番目の文字を表す．このとき，置換，削除，挿入の各コストを $C_{sub}$，$C_{del}$，$C_{ins}$ とするレーベンシュタイン距離の下で，確率分布 $p(s)$ に対する中央文字列を求める整数線形計画問題 ILPMed を以下のように定義する．

最小化 $\displaystyle\sum_{k=1}^{N} p(s^{(k)}) \Big\{ \sum_{i=1}^{n_k} C_{del} x_{k,i,0} + \sum_{j=1}^{m} C_{ins} y_{k,0,j} + \sum_{i=1}^{n_k} \sum_{j=1}^{m} (C_{del} x_{kij} + C_{ins} y_{kij} + C_{sub} h_{kij}) \Big\} - C_{ins}(m - l)$

制約条件

すべての $k, i, j$ $(1 \le k \le N, i < n_k, j < m)$ に対して

$1 = x_{k,1,0} + y_{k,0,1} + z_{k,1,1}$ (a1)

$x_{k,i,0} = x_{k,i+1,0} + y_{k,i,1} + z_{k,i+1,1}$ (a2)

$x_{k,n_k,0} = y_{k,n_k,1}$ (a3)

$y_{k,0,j} = x_{k,1,j} + y_{k,0,j+1} + z_{k,1,j+1}$ (a4)

$y_{k,0,m} = x_{k,1,m}$ (a5)

$x_{kij} + y_{kij} + z_{kij} = x_{k,i+1,j} + y_{k,i,j+1} + z_{k,i+1,j+1}$ (a6)

$x_{kn_kj} + y_{kn_kj} + z_{kn_kj} = y_{k,n_k,j+1}$ (a7)

$x_{kim} + y_{kim} + z_{kim} = x_{k,i+1,m}$ (a8)

$x_{kn_km} + y_{kn_km} + z_{kn_km} = 1$ (a9)

$y_{kn_kj} \ge \frac{1}{m}(j - l)$ (b)

$s_i^{(k)} - t_j \le |\mathcal{A}| g_{kij}$ (c1)

$t_j - s_i^{(k)} \le |\mathcal{A}| g_{kij}$ (c2)

$h_{kij} \ge z_{kij} + g_{kij} - 1$ (d1)

$h_{kij} \le \frac{1}{2}(z_{kij} + g_{kij})$ (d2)

$x_{kij}, y_{kij}, z_{kij}, g_{kij}, h_{kij} \in \{0, 1\}$

$t_j \in \{1, \cdots, |\mathcal{A}|\}$, $0 \le l \le m$

ここで $m$ は十分大きな定数 (例えば $\sum_{k=1}^{N} n_k$) であり，$l$ は中央文字列の長さを表す変数である．

変数 $x_{kij}$ は $s_i^{(k)}$ が削除されるとき 1 をとり，そうでなければ 0 をとる (図 1 参照)．$y_{kij}$ は $t_j$ が挿入されるとき 1 をとり，そうでなければ 0 をとる．$z_{kij}$ は $s_i^{(k)}$ が $t_j$ に置換されるとき 1 をとり，そうでなければ 0 をとる．各 $s^{(k)}$ に対して $t$ とのアラインメントは左上から右下へ正確に 1 つの経路でなければならない．つまり，$x_{kij}$, $y_{kij}$, $z_{kij}$ のいずれかが 1 であるなら，$x_{k,i+1,j}$, $y_{k,i,j+1}$, $z_{k,i+1,j+1}$ のいずれかが必ず 1 でなければならない (式 (a6) 参照)．同様に $(i, j)$ の位置について，式 (a1-9) の制約が要請される．式 (b) は中央文字列 $t$ の長さが $l$ であることを表現し，$j > l$ である $j$ に対して $y_{kn_kj} = 1$ を要請する．変数 $l$ を総和の範囲としたレーベンシュタイン距離の式 $d(t, s^{(k)}) = \sum_{i=1}^{n_k} C_{del} x_{k,i,0} + \sum_{j=1}^{l} C_{ins} y_{k,0,j} + \sum_{i=1}^{n_k} \sum_{j=1}^{l} (C_{del} x_{kij} + C_{ins} y_{kij} + C_{sub} h_{kij})$ を整数線形計画問題において表現することは困難なため，十分大きな定数 $m$ を導入する．上式の $l$ を $m$ で置き換えた式は $j > l$ である $j$ に対して $y_{kn_kj} = 1$ の制約を課しているため，$C_{ins}(m - l)$ だけレーベンシュタイン距離よりも大きくなる．ILPMed ではこの分だけ目的関数から減算している．式 (c1-2) は $s_i^{(k)}$ と $t_j$ が等しくなるとき $g_{kij}$ が 1 となることを表現し，式 (d1-2) は $z_{kij}$ と $g_{kij}$ との両方とも 1 となるとき $h_{kij}$ が 1 となることを表現する．つまり $s_i^{(k)} \ne t_j$ のとき，$s_i^{(k)}$ から $t_j$ への置換コストが $C_{sub}$ となり，そうでなければ 0 となることを表す．中央文字列 $t$ の長さは高々 $m = \sum_{k=1}^{N} n_k$ であり，もし $m$ より長ければ $s^{(k)}$ のすべての文字列を連結した文字列よりも長くなることを意味する．

中央文字列のときと同様に，確率分布 $p(s)$ に対する中心文字列を求める整数線形計画問題 ILPCen を以下のように定義する．

最小化 $d$

制約条件

すべての $k, i, j$ $(1 \le k \le N, i < n_k, j < m)$ に対して

$p(s^{(k)}) \Big\{ \sum_{i=1}^{n_k} C_{del} x_{k,i,0} + \sum_{j=1}^{m} C_{ins} y_{k,0,j} + \sum_{i=1}^{n_k} \sum_{j=1}^{m} (C_{del} x_{kij} + C_{ins} y_{kij} + C_{sub} h_{kij}) - C_{ins}(m - l) \Big\} \le d$

$1 = x_{k,1,0} + y_{k,0,1} + z_{k,1,1}$

$x_{k,i,0} = x_{k,i+1,0} + y_{k,i,1} + z_{k,i+1,1}$

$x_{k,n_k,0} = y_{k,n_k,1}$,

$y_{k,0,j} = x_{k,1,j} + y_{k,0,j+1} + z_{k,1,j+1}$
$y_{k,0,m} = x_{k,1,m},$
$x_{kij} + y_{kij} + z_{kij} = x_{k,i+1,j} + y_{k,i,j+1} + z_{k,i+1,j+1}$
$x_{kn_kj} + y_{kn_kj} + z_{kn_kj} = y_{k,n_k,j+1}$
$x_{kim} + y_{kim} + z_{kim} = x_{k,i+1,m}$
$x_{kn_km} + y_{kn_km} + z_{kn_km} = 1$
$y_{kn_kj} \geq \frac{1}{m}(j - l)$
$s_i^{(k)} - t_j \leq |\mathcal{A}| g_{kij}$
$t_j - s_i^{(k)} \leq |\mathcal{A}| g_{kij}$
$h_{kij} \geq z_{kij} + g_{kij} - 1$
$h_{kij} \leq \frac{1}{2}(z_{kij} + g_{kij})$
$x_{kij}, y_{kij}, z_{kij}, g_{kij}, h_{kij} \in \{0, 1\}$
$t_j \in \{1, \cdots, |\mathcal{A}|\}$
$0 \leq l \leq m,\ d \geq 0$
ここで $d$ は式 (6) の最大値を表す変数である．

### 2.4 三角不等式制約

より効率的に中央文字列および中心文字列を求めるため，ILPMed と ILPCen に新たな制約を加える．レーベンシュタイン距離 $d(s,t)$ は三角不等式を満たすので，以下の三角不等式から導かれる制約を ILPMed，ILPCen のそれぞれに加えた整数線形計画問題 ILPMedTri，ILPCenTri を提案する．

すべての $k_1, k_2 (k_1 \neq k_2)$ に対して
$$d(s^{(k_1)}, s^{(k_2)}) + d(s^{(k_1)}, t) \geq d(s^{(k_2)}, t) \quad (7)$$
すべての $k_1, k_2 (k_1 < k_2)$ に対して
$$d(s^{(k_1)}, t) + d(s^{(k_2)}, t) \geq d(s^{(k_1)}, s^{(k_2)}) \quad (8)$$

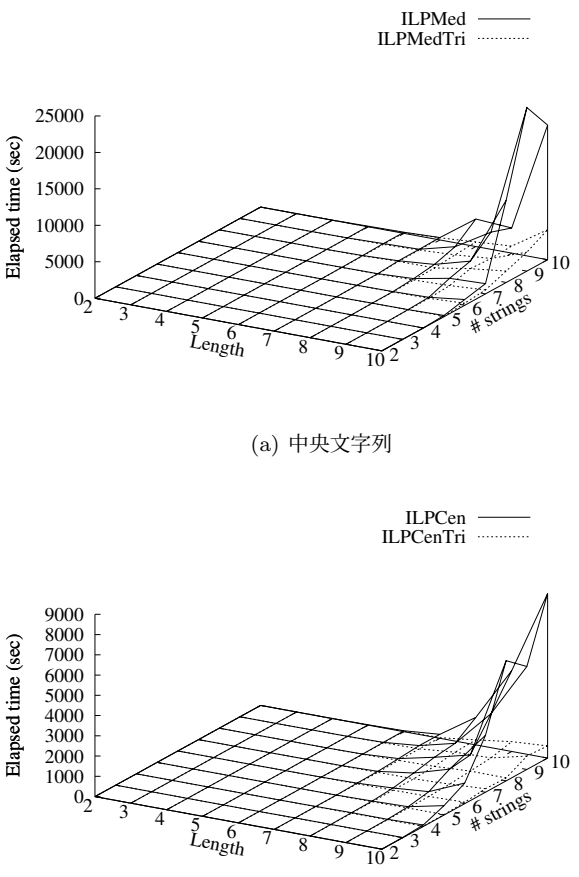
ここで，$d(s^{(k_1)}, s^{(k_2)})$ は与えられた文字列 $s^{(k_1)}, s^{(k_2)}$ から計算される定数であり，$d(s^{(k)}, t)$ は次の式で置き換えられる変数を含む式である．

$$\sum_{i=1}^{n_k} C_{del} x_{k,i,0} + \sum_{j=1}^{m} C_{ins} y_{k,0,j} + \sum_{i=1}^{n_k} \sum_{j=1}^{m} (C_{del} x_{kij} + C_{ins} y_{kij} + C_{sub} h_{kij}) - C_{ins}(m - l) \quad (9)$$

この三角不等式制約は ILPMed および ILPCen が最適解を見つけるのを妨げず，ILPMedTri，ILPCenTri もまた最適解である中央文字列および中心文字列を見つけることができる．Jiang らも三角不等式を利用した線形計画問題 [25], [26] を提案しているが，中央文字列に対する下限を求めるのみで本研究のように中央文字列自体は求められない．

## 3. 結果

提案手法の計算効率を評価するために計算機実験を行った．レーベンシュタイン距離の編集コストには $C_{del} = C_{ins} = C_{sub} = 1$ を用い，アルファベット $\mathcal{A}$ の大きさを 4 とした．文字列の集合 $\mathcal{A}^*$ 上の確率分布として以下のように $p_1(s)$，$p_2(s)$ をランダムに生成する．$p_1(s)$ では，$p_1(s) > 0$ を満たす $N$ 本の長さ $n_k$ の文字列 $s^{(k)}$ を生成する (ここで $N = 2, \cdots, 10$，$n_k = 2, \cdots, 10$)．$s_i^{(k)}$ は，$\alpha$ を平均 0，分散 1 の正規分布に従う確率変数の実現値として，$\min(1 + \lfloor |\alpha| \rfloor, |\mathcal{A}|)$ によって定める．$\lfloor \alpha \rfloor$ は $\alpha$ を越えない最大の整数である．文字列 $s^{(k)}$ の生起確率 $p_1(s^{(k)})$ は $\sum_{k=1}^{N} p_1(s^{(k)}) = 1$ を満たすような一様乱数によって定める．$p_2(s)$ では $N$ 本の文字列 $s^{(k)}$ を，$a_1$ ($\in \mathcal{A}$) が $n$ 個並んだ文字列 $a_1 \cdots a_1$ から始めて，置換，挿入，削除の編集操作からランダムに 1 つを選択し適用する手続きを 3 回繰り返して生成する (ここで $N = 2, \cdots, 10$，$n = 5, \cdots, 10$)．生起確率 $p_2(s^{(k)})$ は $p_1(s)$ のときと同様に一様乱数によって定める．$p_1(s)$，$p_2(s)$，$n_k$，$N$ のそれぞれの場合について，$p_1(s^{(k)})$ あるいは $p_2(s^{(k)})$ の確率をもつ $N$ 本の文字列 $s^{(k)}$ の集合を 10 回生成し，それぞれについて計測した実行時間の平均を取った．整数線形計画問題を解くソフトウェアとして CPLEX (version 12.5) を使用し，Xeon 2.9GHz，35GB メモリの計算機で実行した．




(a) 中央文字列

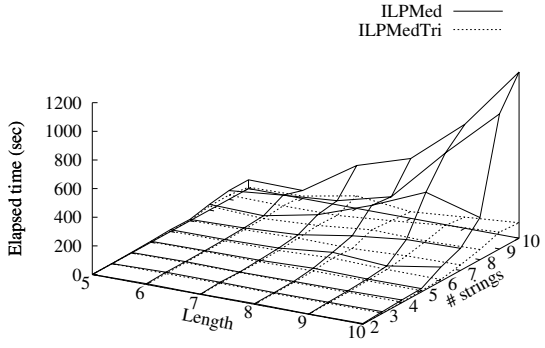
(b) 中心文字列

**図 2** 確率分布 $p_1(s)$ に対する ILPMed，ILPMedTri，ILPCen，ILPCenTri の平均実行時間 ($N = 2, \cdots, 10, n_k = 2, \cdots, 10$)．

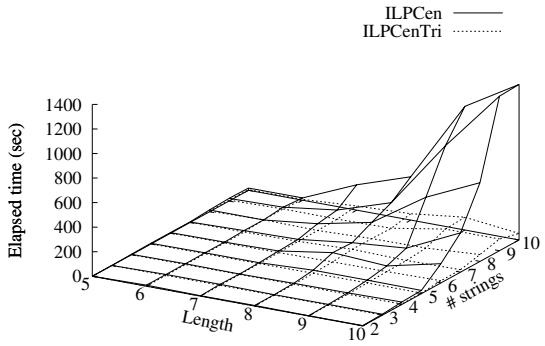
図 2 では，$N = 2, \cdots, 10, n_k = 2, \cdots, 10$ の各場合での，確率分布 $p_1(s)$ に対する ILPMed，ILPMedTri，ILPCen，

**表 1** 確率分布 $p_1(s)$ に対する $n_k = 10$ のときの ILPMed, ILPMedTri, ILPCen, ILPCenTri の平均実行時間 ($N = 2, \cdots, 10$).

| $N$ | ILPMed | ILPMedTri | ILPCen | ILPCenTri |
|---|---|---|---|---|
| 2 | 14.4 | 2.4 | 6.4 | 1.9 |
| 3 | 21.5 | 5.3 | 18.1 | 6.3 |
| 4 | 23.7 | 22.5 | 72.1 | 7.9 |
| 5 | 168.0 | 68.6 | 464.8 | 20.0 |
| 6 | 798.6 | 283.7 | 1015.6 | 72.7 |
| 7 | 1400.6 | 542.5 | 2810.3 | 173.4 |
| 8 | 11269.8 | 1570.0 | 5936.2 | 142.9 |
| 9 | 22438.1 | 2589.5 | 5086.0 | 659.6 |
| 10 | 18467.8 | 4104.7 | 8120.3 | 597.5 |




(a) 中央文字列




(b) 中心文字列

**図 3** 確率分布 $p_2(s)$ に対する ILPMed, ILPMedTri, ILPCen, ILPCenTri の平均実行時間 ($N = 2, \cdots, 10, n = 5, \cdots, 10$).

ILPCenTri の平均実行時間を示す．表 1 は，$n = 10$ のときの具体的な平均実行時間を示す．ILPMed と ILPCen の平均実行時間は文字列の本数 $N$ および長さ $n_k$ とともに指数関数的に急速に増加しているが，中央文字列，中心文字列を求める問題がともに NP 困難であることから妥当であると考えられる．また三角不等式制約による ILPMedTri, ILPCenTri は元の ILPMed，ILPCen に比べ大幅に実行時間を削減していることが分かる．

図 3 では，$N = 2, \cdots, 10, n = 5, \cdots, 10$ の各場合での，

**表 2** 確率分布 $p_2(s)$ に対する $n = 10$ のときの ILPMed, ILPMedTri, ILPCen, ILPCenTri の平均実行時間 ($N = 2, \cdots, 10$).

| $N$ | ILPMed | ILPMedTri | ILPCen | ILPCenTri |
|---|---|---|---|---|
| 2 | 4.7 | 1.1 | 2.7 | 0.4 |
| 3 | 12.4 | 2.5 | 7.5 | 1.4 |
| 4 | 18.9 | 4.1 | 19.5 | 3.4 |
| 5 | 23.3 | 8.9 | 26.6 | 5.0 |
| 6 | 98.6 | 11.8 | 157.0 | 18.1 |
| 7 | 154.3 | 38.5 | 320.4 | 20.3 |
| 8 | 293.7 | 71.2 | 641.3 | 55.8 |
| 9 | 943.4 | 117.3 | 1259.2 | 94.8 |
| 10 | 1160.4 | 108.1 | 1266.7 | 53.0 |

確率分布 $p_2(s)$ に対する ILPMed，ILPMedTri，ILPCen，ILPCenTri の平均実行時間を示す．表 2 は，$n = 10$ のときの具体的な平均実行時間を示す．$p_2(s)$ においても，三角不等式制約が有効にはたらいていることが分かる．

## 4. まとめ

中央文字列および中心文字列の定義を文字列の集合 $\mathcal{A}^*$ 上の確率分布 $p(s)$ に一般化し，レーベンシュタイン距離の下で NP 困難であるこれらの問題の最適解を求める新たな整数線形計画問題 ILPMed，ILPCen を提案した．さらにレーベンシュタイン距離が三角不等式を満たすことから，ILPMed，ILPCen に三角不等式による制約を加えた ILPMedTri および ILPCenTri を提案した．計算機実験による結果は，ILPMedTri および ILPCenTri が大幅に ILPMed および ILPCen の実行時間を削減できることを示した．

しかし膨大な生物学的データへ適用するには現実的ではなく，新たな制約の導入や線形計画問題への緩和などによる改良が必要である．

## 謝辞

本研究は JSPS 科研費 24500361, 26610037 の助成を受けたものです．

### 参考文献

[1] Bunke, H., Jiang, X., Abegglen, K. and Kandel, A.: On the weighted mean of a pair of strings, *Pattern Analysis and Applications*, Vol. 5, pp. 23–30 (2002).
[2] Chen, S., Tung, S., Fang, C., Cherng, S. and Jain, A.: Extended attributed string matching for shape recognition, *Computer Vision and Image Understanding*, Vol. 70, pp. 36–50 (1998).
[3] Martínez-Hinarejos, C., Juan, A. and Casacuberta, F.: Median strings for k-nearest neighbour classification, *Pattern Recognition Letters*, Vol. 24, pp. 173–181 (2003).
[4] Kohonen, T.: Median strings, *Pattern Recognition Letters*, Vol. 3, pp. 309–313 (1985).
[5] Gusfield, D.: *Algorithms on strings, trees and se-*

*quences*, Cambridge University Press (1997).
[6] Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals, *Doklady Adademii Nauk SSSR*, Vol. 163, No. 4, pp. 845–848 (1965).
[7] Hamming, R.: Error detecting and error correcting codes, *The Bell System Technical Journal*, Vol. 29, No. 2, pp. 147–160 (1950).
[8] Winkler, W.: String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage, *Proceedings of the Section on Survey Research Methods*, pp. 354–359 (1990).
[9] Gramm, J.: Fixed-parameter algorithms for the consensus analysis of genomic data, PhD Thesis, Universität Tübingen (2003).
[10] Gramm, J., Niedermeier, R. and Rossmanith, P.: Fixed-parameter algorithms for closest string and related problems, *Algorithmica*, Vol. 37, pp. 25–42 (2003).
[11] Hufsky, F., Kuchenbecker, L., Jahn, K., Stoye, J. and Böcker, S.: Swiftly computing center strings, *BMC Bioinformatics*, Vol. 12, p. 106 (2011).
[12] Dinu, L. and Ionescu, R.: An efficient rank based based approach for closest string and closest substring, *PLoS ONE*, Vol. 7, No. 6, p. e37576 (2012).
[13] de la Higuera, C. and Casacuberta, F.: Topology of strings: Median string is NP-complete, *Theoretical Computer Science*, Vol. 230, pp. 39–48 (2000).
[14] Nicolas, F. and Rivals, E.: Complexities of the centre and median string problems, *Lecture Notes in Computer Science*, Vol. 2676, pp. 315–327 (2003).
[15] Nicolas, F. and Rivals, E.: Hardness results for the center and median string problems under the weighted and unweighted edit distances, *Journal of Discrete Algorithms*, Vol. 3, pp. 390–415 (2005).
[16] Sim, J. S. and Park, K.: The consensus string problem for a metric is NP-complete, *Journal of Discrete Algorithms*, Vol. 1, pp. 111–117 (2003).
[17] Kruskal, J.: An overview of sequence comparison: Time warps, string edits, and macromolecules, *SIAM Review*, Vol. 25, No. 2, pp. 201–237 (1983).
[18] Lopresti, D. and Zhou, J.: Using Consensus Sequence Voting to Correct OCR Errors, *Computer Vision and Image Understanding*, Vol. 67, No. 1, pp. 39–47 (1997).
[19] Casacuberta, F. and de Antoni, M.: A greedy algorithm for computing approximate median strings, *Proceedings of National Symposium on Pattern Recognition and Image Analysis*, pp. 193–198 (1997).
[20] Jiang, X., Abegglen, K., Bunke, H. and Csirik, J.: Dynamic computation of generalised median strings, *Pattern Analysis and Applications*, Vol. 6, pp. 185–193 (2003).
[21] Olivares-Rodríguez, C. and Oncina, J.: A stochastic approach to median string computation, *Structural, Syntactic, and Statistical Pattern Recognition*, Springer, Berlin, pp. 431–440 (2008).
[22] Abreu, J. and Rico-Juan, J.: A new iterative algorithm for computing a quality approximate median of strings based on edit operations, *Pattern Recognition Letters*, Vol. 36, pp. 74–80 (2014).
[23] Koyano, H. and Kishino, H.: Quantifying biodiversity and asymptotics for a sequence of random strings, *Physical Review E*, Vol. 81, No. 6, p. 061912 (2010).
[24] Wagner, R. and Fischer, M.: The string-to-string correction problem, *Journal of the ACM*, Vol. 21, No. 1, pp. 168–173 (1974).
[25] Jiang, X. and Bunke, H.: Optimal lower bound for generalized median problems in metric space, *Structural, Syntactic, and Statistical Pattern Recognition*, Springer, Berlin, pp. 143–151 (2002).
[26] Jiang, X., Wentker, J. and Ferrer, M.: Generalized median string computation by means of string embedding in vector spaces, *Pattern Recognition Letters*, Vol. 33, pp. 842–852 (2012).