秘密計算によるセキュアなDNB分析の実装及び評価

橋本 順子[1,a)]　五十嵐 大[1]　菊池 亮[1]　岸 寿春[1]　高橋 克巳[1]

概要：生体情報を分析することにより，人の健康状態を理解・支援することができる．DNB(動的ネットワークバイオマーカー) 理論は，生体状態の揺らぎから，未病状態を検出する数理的手法である．生体情報を分析することで新しい知見の発見が期待されるが，これら情報はプライバシーの観点から，広く提供されることへの抵抗がある．秘密計算は，元となるデータを暗号化したまま分析を行い，結果だけを平文で得ることができる技術である．秘密計算を用いて生体データの分析を行うことで，データ提供に関する障壁が減ることが期待される．今回，未病状態を検出する注目の DNB 分析を秘密計算で実現する方法を提案する．DNB 分析を秘密計算で行うにあたって，中央絶対偏差のアルゴリズムの工夫および階層型クラスタリングの相関係数対応を行った．また，性能測定を行い，実用的な速度で実行できることを確認した．

キーワード：秘密計算, 秘密分散, 秘密計算システム,DNB 理論,DNB 分析

Implementation and Evaluation of Secure DNB Analysis Using Secret Computation

Junko Hashimoto[1,a)]　Dai Ikarashi[1]　Ryo Kikuchi[1]　Toshiharu Kishi[1]　Katsumi Takahashi[1]

Abstract:
By analyzing biological data, it is possible to understand and support a person's health condition. DNB (Dynamic Network Biomarker) theory is a mathematical method for detecting pre-disease states from fluctuations in biological states. Although analyzing biometric data may lead to new insights, the biometric data required for such analyses may not be provided to external parties due to privacy concerns. Secure computing is a technology that analyzes the original data while keeping it encrypted, and outputs only the results in plain text. It is expected that data will be provided by analyzing biometric data using secure computation. In this paper, we propose a method to realize the notable DNB analysis for detecting pre-disease states using secure computation. In order to implement secure computation for DNB analysis, we devised a Median absolute deviation algorithm and supported the correlation coefficient of hierarchical clustering. We also conducted performance measurements and confirmed that it can be executed at a practical speed.

Keywords: Secure Computation, Secret Sharing, Secure Computation System, DNB Theory, DNB Analysis

1. はじめに

生体情報を分析することにより，人の健康状態を理解・支援することができる．DNB 理論 (動的ネットワークバイオマーカー（Dynamical Network Biomarker）理論)[1] では，健康状態から疾病状態へ状態遷移する過程において，遺伝子の mRNA 発現量や血中ホルモン濃度などの「揺らぎ」が現れることに注目し，「揺らぎ」から疾病の発症を予測する数理的手法である．未病を検出するこの手法は，超高齢化社会において，発病前に治療や予防を行う超早期治療を実現とする礎として期待されている [2]．しかし，これらの分析の対象となる生体情報は，プライバシーの観点から，分析のために広く提供されることが難しい．

一方，秘密計算は，元となるデータを暗号化したまま分析を行い，結果だけを平文で得ることができる技術である．秘密計算を用いて生体データの分析を行うことで，データ

[1] NTT 社会情報研究所,
NTT Social Informatics Laboratories
[a)] junko.hashimoto@ntt.com

提供に関する障壁が減ることが期待される．

本稿では，DNB 分析を秘密計算で実現する．DNB 分析を秘密計算化するにあたって必要な要素技術である，中央絶対偏差，順位への変換，相関係数を用いた階層型クラスタリングの各アルゴリズムおよび，それらを組み合わせた DNB 分析のアルゴリズムを提案，実装し，評価を行う．

1.1 関連研究

生体情報を秘密計算で分析した研究として，NTT は疾患と遺伝子の関連を調べる全ゲノム関連解析（Genome-Wide Association Study: GWAS) の重要な検定手法であるフィッシャー正確検定を秘密計算で実現している [3], [4].

DNB 分析を秘密計算化するにあたって必要な要素技術である階層型クラスタリングについては，三品らが提案した [5], [6] にて，メトリックにユークリッド距離, メトリックの更新に群平均法を用いた秘密計算でのアルゴリズムが提案されている．ただし，DNB 分析では，データ間の距離ではなく，相関をメトリックとして用いるため，これをそのまま適用することはできない．

2. 準備

2.1 DNB 分析

DNB 分析の対象とするデータ，解析方法は多様である．[7] では，DNB 分析の 2 つの手法，Two-step Method と，PCA-based Method が説明されている．Two-step Method は，対照群と実験群の遺伝子発現量を比較し，対照群に比べ，揺らぎが有意に大きい遺伝子を中央絶対偏差 (MAD) を用いて抽出するステップと，抽出した遺伝子に対して，実験群の遺伝子発現量をクラスタリングし，同じ揺らぎの傾向を持つ遺伝子群を抽出するステップの 2 つのステップから成る．

また，この方法に沿って，DNB 分析を行うツールが [8] で公開されている．

本稿では，Two-step Method を，秘密計算で実現する．以下に，[7], [8] をベースとした，DNB 分析の Two-step アルゴリズムを記す．

$\vec{g}$ は，遺伝子 ID リストで，長さ n_g とする．

C は，$n_g \times m_c$ の行列で，$c_{i,j}$ は，対照群の j 番目の個体について，$\vec{g}$ の遺伝子 ID g_i の遺伝子発現量．

E は，$n_g \times m_e$ の行列で，$e_{i,j'}$ は，実験群の j' 番目の個体について，$\vec{g}$ の遺伝子 ID g_i の遺伝子発現量．

θ_1 は，E の遺伝子発現量と，C の遺伝子発現量の揺らぎを比較して，遺伝子 ID リスト $\vec{g}$ から， 揺らぎが多い遺伝子 ID リスト $\vec{g'}$ をピックアップする際の，閾値．

θ_2 は，階層型クラスタリングにおいて，結合可能なクラスタ間の相関係数の最小値．

θ_3 は，階層型クラスタリング結果として返却するクラスタの最小要素数の最大の要素数に対する割合．

Algorithm 1 DNB 分析 (平文)

Functionality: $\vec{g''}, \vec{cid}, \vec{cnt}, \vec{c''}, \vec{e''} \leftarrow DNB(\vec{g}, C, E, \theta_1, \theta_2, \theta_3)$

Input: $\vec{g}$, C, E, $\theta_1, \theta_2, \theta_3$

Output: $\vec{g''}, \vec{cid}, \vec{cnt}, \vec{c''}, \vec{e''}$

1. $\vec{c'} := MAD(C)$
2. $\vec{e'} := MAD(E)$
3. for each $i < length(\vec{g})$
   if $(\theta_1 \times c'_i \leq e'_i)$ then
   $(\vec{g'}, E')$ に $(g_i, E_{i,*})$ を追加する
4. $E'' := Rank(E')$
5. $L := AgglomerativeClustering(E'')$
6. L のうち，クラスタの結合距離が $1 - \theta_2$ 以下のものを，L' とする
7. L' から，クラスタの最大要素数 max_{cnt} を求める
8. 要素数が $\theta_3 \times max_{cnt}$ より大きいクラスタを抽出する
9. $\vec{g'}$ のうち，抽出されたクラスタに分類されたものを $\vec{g''}$ とする
10. $\vec{g''}$ の各要素に対応するクラスタ $\vec{cid}$, 要素数 $\vec{cnt}$,
    対照群の MAD$\vec{c''}$, 実験群の MAD$\vec{e''}$ を出力する

$\vec{g''}$ は，揺らぎが大きいとしてピックアップした遺伝子 ID リスト $\vec{g'}$ のうち，階層型クラスタリングで，θ_2,θ_3 の条件を満たしたクラスタに含まれる遺伝子 ID リスト．

$\vec{cid}$ は，$\vec{g''}$ のクラスタ ID.

$\vec{cnt}$ は, $\vec{cid}$ の要素数.

$\vec{c''}$ は, $\vec{g''}$ の対照群の MAD.

$\vec{e''}$ は, $\vec{g''}$ の実験群の MAD.

$MAD(X)$ は，X の各横方向のレコードに対する，各中央絶対偏差のベクトル $\vec{x}$ を出力する関数とする．

$\vec{x}$ の中央絶対偏差を $MAD(\vec{x})$ とすると，

$$MAD(\vec{a}) := Median(|\vec{a} - Median(\vec{a})|) \qquad (1)$$

と定義される．ここで，$Median(\vec{a})$ は，$\vec{a}$ の中央値とする．

$Rank(X)$ は，X の各行を，順位に変換した X' を出力する関数とする．X の i 番目の行 $\vec{x_i}$ を順位に変換したものが，X' の i 番目の行となる．同値が存在する場合は，平均値を順位とする．

$AgglomerativeClustering(X)$ は，X の階層型クラスタリングを行い，クラスタリング結果 L を出力する関数とする．L は，4 列の行列で，1 列目と 2 列目が結合するクラスタ ID，3 列目はクラスタ間の相関係数，4 列目は結合後のクラスタの要素数である [10]．1 行目が最初に結合したクラスタの情報，2 行目が次に結合したクラスタの情報となり，X の行数 n_x に対して，L の行数は，最大 $n_x - 1$ となる．また，本稿で実現する DNB 分析では，スピアマンの順位相関係数を用いた階層型クラスタリングを行うため，入力 X は Rank 関数により順位に変換されたものを用い，メトリックとしてピアソンの相関係数を用い，また距離の更新は群平均法で行うものとする．

ピアソンの相関係数の定義式を以下に示す．

$$r(\vec{p},\vec{q}) = \frac{\sum_i^n (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_i^n (p_i - \bar{p})^2 \sum_i^n (q_i - \bar{q})^2}} \quad (2)$$

ここで，$\vec{p},\vec{q}$ は順位, $\bar{x}$ は $\vec{x}$ の平均値とする．

階層型クラスタリングでは，ピアソンの相関係数から，以下の式により距離 d を求める．

$$d(\vec{p},\vec{q}) = 1 - r(\vec{p},\vec{q}) \quad (3)$$

2.2 秘密計算の記法および利用アルゴリズム

$k = 2,\ N \geq 2k-1$ を前提とし，メルセンヌ素数 p を modulus とした素体を利用した (k,N)-Shamir 秘密分散方式を用いる．乗算アルゴリズムは，Gennaro らの乗算 [9] を利用している．

DNB 分析のアルゴリズムを，秘密計算で実現するにあたり要素として利用する秘密計算アルゴリズムを説明する．

2.2.1 記法

本稿では下記の記法を用いる．

記法

a	平文値
$[\![a]\!]$	(k,N)-Shamir 秘密分散された値
$[a]$	(k,N)-加法的秘密分散された値
$[\pi]$	置換データ π が (k,N)-加法的秘密分散された値
$\{\{\sigma^{-1}\}\}$	ハイブリッド置換（詳しくは後述）
p	(k,N)-Shamir 秘密分散値の modulus
$[\![\mathcal{F}]\!]$	0 以上 p 未満の自然数の集合
$[\![a]\!] +_p [\![b]\!]$	$c := (a+b) \bmod p$ となる $[\![c]\!]$ を計算する.$-,\times$ も同様
$[\![a]\!] +_p b$	$c := (a+b) \bmod p$ となる $[\![c]\!]$ を計算する.$-,\times$ も同様
$-[\![a]\!]$	$b := -a \bmod p$ となる $[\![b]\!]$ を計算する
$[\![a]\!] <_p [\![b]\!]$	$c := (a<b)$ となる $[c]$ を計算する.$\leq,>,\geq,=$ も同様
$[\![a]\!] <_p b$	$c := (a<b)$ となる $[c]$ を計算する.$\leq,>,\geq,=$ も同様
$[\![a]\!] \&_p b$	$c := (a\&b)$ となる $[c]$ を計算する．
_	ダミー変数（内部で利用されない変数を表現）

2.2.2 関数の定義

本稿でアルゴリズムを構成する要素として利用する，秘密計算を行う関数を定義する．ここでは入出力パラメータのみ定義する．内部のアルゴリズムについては，各参考文献を参照されたい．

- $Abs([\![\vec{x}]\!])$ は，$[\![\vec{y}]\!]$ を出力する関数とする．y_i は，x_i の絶対値である．
- $Sort([\![X]\!], \{\{\sigma^{-1}\}\})$ は，$[\![Y]\!]$ を出力する関数とする [12]．Y は，X を，置換データ σ^{-1} に従って並び替えたものである．ハイブリッド置換データ $\{\{\sigma^{-1}\}\}$ は，置換データの加法的秘密分散 $[\pi]$ と，$\pi \circ \sigma^{-1}$ から成る．
- $InvertSort([\![Y]\!], \{\{\sigma\}\})$ は，$[\![X]\!]$ を出力する関数とする．Y は，X を，置換データ σ^{-1} に従って並び替えたものである．つまり，上の $Sort$ 関数で並び替えた Y を，元の並び順 X に戻す関数である．
- $Max([\![\vec{x}]\!])$ は，$[\![max]\!]$ を出力する関数とする．max は，$\vec{x}$ の要素のうち，最大値である．x のソートを行い，最後の要素を返却することにより実現する．
- $GroupbyMedian([\![K]\!], [\![\vec{v}]\!])$ は，$[\![\vec{v'}]\!]$，$\{\{\sigma^{-1}\}\}$，$[\![K']\!]$，$[\![\vec{m}]\!]$ を出力する関数とする [13]．$\vec{v}$ は，中央値を求める列，K は，$\vec{v}$ の各要素に対するキー行列である．$\vec{v'}$ は，$\vec{v}$ を，ソートされたキー行列 K でソートし，かつ，同一キー値に対しては，$\vec{v}$ の値でソートした配列．σ^{-1} は，$\vec{v}$ を $\vec{v'}$ に並び替える置換データ．K' は，キー行列 K をユニーク化し，ソートしたもの．$\vec{m}$ は，K の各グループに対し，$\vec{v}$ の各中央値を求め，2 倍した配列を，K' に対応する順番に並べたものである．
- $GroupbyMean([\![K]\!], [\![\vec{v}]\!])$ は，$[\![\vec{v'}]\!]$, $\{\{\sigma^{-1}\}\}$, $[\![K']\!]$, $[\![\vec{m}]\!]$ を出力する関数とする．$\vec{v}$ は，平均値を求める列，K は，$\vec{v}$ の各要素に対するキー行列である．$\vec{v'}$ は，$\vec{v}$ を，ソートされたキー行列 K でソートした配列．σ^{-1} は，$\vec{v}$ を $\vec{v'}$ に並び替える置換データ．K' は，キー行列 K をユニーク化し，ソートしたもの．$\vec{m}$ は，K の各グループに対し，$\vec{v}$ の各平均値を，K' に対応する順番に並べたものである．
- $GroupbyCount([\![K]\!])$ は，$[\![K']\!]$, $[\![\vec{c}]\!]$ を出力する関数とする．K は，グループ化のキー行列である．K' は，キー行列 K をユニーク化し，ソートしたもの．$\vec{c}$ は，K の各グループに対し，度数をカウントしたものである．
- $Join([\![A]\!], [\![B]\!], i1, i2)$ は，$[\![C]\!]$ を出力する関数とする [14]．行列 A，行列 B を，それぞれの行列に含まれるキー列 ($i1, i2$ で指定) で内部結合した，行列 C を出力する．行列 C は，キー列，キー列を除く行列 A の各列，キー列を除く行列 B の各列，の順番に列が並んだ列を持つ．
- $Filter([\![\vec{x}]\!], [\vec{f}])$ は，$[\![\vec{y}]\!]$ を出力する関数とする．$\vec{y}$ は，$\vec{x}$ から，$f_i = true$ となる x_i のみを抜き出した配列である．$[\vec{f}]$) の降順ソートを行い，先頭から $Sum(f)$ 件のデータを抜き出すことで実現する．本関数では，内部で，$Sum(f)$ すなわち，出力結果の行数を復元することに注意．

また，以下に，行列/配列を並び替える関数を定義する．これらの関数は，値を計算しないため，入力，出力パラメータが平文，シェアどちらにも適用できる．

- $At(X, i)$ は，入力の行列 X の i 列目の配列を出力する関数とする．
- $Repeat(\vec{x}, n)$ は，入力の配列 $\vec{x}$ の長さが m のとき，長さ $n \times m$ の配列 $\vec{y}$ を出力する関数とする．$y_i := x_j, j := i \bmod n$ である．
- $Concatenate(X, Y, Z, \dots)$ は，入力パラメータ

$X, Y, Z, \ldots$ を，横方向に連結した行列 A を出力する関数とする．入力パラメータは，行列または配列とし，行列の行数，配列の長さは同じであること．

- $ChangeFormat(\vec{k}, V)$ は，keyID リスト $\vec{k}$ と,keyID に対応するデータを横方向に並べた行列 V を入力とし，keyID を 1 列目，対応する V の各要素を 2 列目に持つ，2 列の行列 A を出力する関数とする．A の行数は，V のセル数と等しい．m を，V の列数とすると，$A_{i,0} = k_j, i = j/m$, $A_{i,1} = V_{j,k}, i = j/m + k$ とする．
- $RevertFormat(A)$ は，$ChangeFormat$ の処理をもとに戻す．すなわち，$ChangeFormat$ の出力 A を入力とし，$ChangeFormat$ の入力 $\vec{k}$, V を出力する関数とする．

2.2.3 階層型クラスタリング

本稿では，三品らの階層型クラスタリング [5] をベースに，相関係数への対応を行う．ベースとする階層型クラスタリングのアルゴリズムを，Algorithm 2 に示す

Algorithm 2 秘密計算階層型クラスタリング

Functionality: $[\![Z]\!] \leftarrow AgglomerativeClustering([\![X]\!])$
Input: $[\![X]\!]$, $n \times m$ のテーブル
Output: $[\![L]\!]$

1. クラスタ ID の初期化 $[\![cid_{new}]\!] := n$
2. クラスタ ID テーブル $[\![C]\!]$ の初期化
3. 出力 $[\![L]\!]$ を空のテーブルとして初期化
4. 全クラスタ間の距離テーブル $[\![D]\!]$ を $[\![X]\!]$ から計算 (ユークリッド距離)
5. for $i = 0, 1, ..., n-2$ do
6. $[\![D]\!]$ から, 最も近いクラスタの ID($[\![cid_1]\!], [\![cid_2]\!]$) と，距離 $[\![d]\!]$ を取得
7. 新しいクラスタ ID$[\![cid_{new}]\!]$ と各クラスタ間の距離と，要素数 $[\![num]\!]$ を計算し，$[\![D]\!]$ を更新 (群平均法)
8. $[\![C]\!]$ を，新しいクラスタの情報で更新
9. $[\![L]\!]$ に，新しいクラスタの結合情報 ($[\![cid_1]\!], [\![cid_2]\!], [\![d]\!], [\![num]\!]$) を追加
10. $[\![cid_{new}]\!] := [\![cid_{new}]\!] + 1$
11. Output $[\![L]\!]$

3. 提案手法

本節では，秘密計算による DNB 分析アルゴリズムを提案する．DNB 分析アルゴリズムでは，中央絶対偏差を求めるために，遺伝子 ID リストの長さを n_g とすると，$4n_g$ 回ソートする必要がある．これを効率的に行うために，グループ化して行う．また，ピアソンの相関係数を用いた階層型クラスタリングのアルゴリズムを提案する．

3.1 基本的な考え方

DNB 分析は，Algorithm 1 に示したように，主に中央絶対偏差を求めるステップ 1，階層型クラスタリングを行うステップ 2 から構成される．

中央絶対偏差

中央絶対偏差は，式 (1) により求められる．この計算を行う手順は，以下の通りである．

(1) $\vec{a}$ をソートする
(2) 中央値 $Median(\vec{a})$ を選ぶ
(3) 中央値からの偏差の絶対値 $|\vec{a} - Median(\vec{a})|$ を計算する
(4) (3) の値をソートする
(5) 中央値を選ぶ，これが $MAD(\vec{a})$ となる

上記手順は，2 回のソートを含む．

DNB 分析アルゴリズムでは，中央絶対偏差を，C,E に対して各 n_g 回行うため，全体で，n_g 行の処理に対し,$O(n_g)$ 回のソートが必要である．

本稿では，これを秘密計算で効率的に行うために，グループ化中央絶対偏差を用いる．遺伝子 ID をキーとし, グループ化対象の値を遺伝子発現量としたグループ化中央絶対偏差を行うことで，n_g 行の処理を,$O(1)$ 回のソートで求めることができる．

グループ化中央絶対偏差の計算手順は，以下の通りである．入力値は，キー列 $\vec{k}$, 対応する値 $\vec{v}$ からなる 2 列のデータとする．あるキー値 k_i に対して，中央絶対偏差を求めたい配列 $\vec{a}$ の各要素が対応するようにする．

(1) $\vec{k}$ と $\vec{v}$ を横方向に結合し，ソートする．$\vec{v}$ の要素が下位ビットとなるように結合することで，$\vec{k}$ のキー値毎にソートされ，同一キー値に対しては $\vec{v}$ の各値でソートされた並び順となる
(2) 各グループごとに，中央値を計算する
(3) 各列で，中央値からの偏差の絶対値を計算する
(4) $\vec{k}$ と (3) の値を横方向に結合し，ソートする
(5) 各グループごとに，中央値を計算する

上記手順により，n_g 個の中央絶対偏差を，$O(1)$ 回のソートで求めることができる．

階層型クラスタリング

階層型クラスタリングについては，[5] にて，ユークリッド距離を用いた秘密計算でのアルゴリズムが提案されている．

DNB 分析では，スピアマンの順位相関係数を用いた階層型クラスタリングを行うため，本稿では，これに対応したアルゴリズムを提案する．

スピアマンの順位相関係数を用いた階層型クラスタリングを行う手順は，以下の通りである．

(1) 入力値を，順位に変換する．(各行ごとに，ソートを行う)
(2) 階層型クラスタリングを行う．その際，クラスタを結合する際の指標として，ピアソンの相関係数を用いる．

順位への変換

順位への変換は，素直に各行ごとでソートを行うと，対象とする行数を n とした場合，$O(n)$ 回のソートが必要である．本稿では，こちらについても，グループ化キーを用いてソートを行うことで,n 行の処理を,$O(1)$ 回のソートで求めるようにする．中央絶対偏差と同様，入力値は，キー列 $\vec{k}$, 対応する値 $\vec{v}$ からなる 2 列のデータとする．

(1) $\vec{k}$ をグループ化のキーとして，$\vec{v}$ をキーごとにソートする
(2) ソート後の $\vec{v}$ を，順位 $0, 1, 2, ..., m-1, 0, 1, 2, ..., m-1, ...$ に置き換える
(3) 元の順番に戻し，順位に置き換えた $\vec{r}$ を得る，ウィ
(4) 同値を平均の順位に変換するため，$\vec{k}, \vec{v}$ をキー値とし，$\vec{r}$ をグループごとに平均する $GroupbyMean$ を行う
(5) $\vec{k}, \vec{v}$ をキー値として，(3),(4) のテーブルの $Join$ を用い，順位 $\vec{r}$ を平均順位に置き換える
(6) 元の順番に戻す

ピアソンの相関係数

階層型クラスタリングでは，式 (2) に示すピアソンの相関係数を用いる．除算，ルートの計算はコストが高いため，式 (2) を以下のように変形し，平均値を求める除算を削減する．さらに，残ったルート，逆数の計算は，参考文献 [15] に示す平方根の逆数を用い，1 回で行う．

$$r(\vec{p},\vec{q}) = \frac{\sum_i^n (np_i - Sum(\vec{p}))(nq_i - Sum(\vec{q}))}{\sqrt{\sum_i^n (np_i - Sum(\vec{p}))^2 \sum_i^n (nq_i - Sum(\vec{q}))^2}} \tag{4}$$

ただし，$Sum(\vec{x}) := \sum_i^n x_i$.

3.2 秘密計算によるグループ化中央絶対偏差

秘密計算によるグループ化中央絶対偏差のアルゴリズムを，Algorithm 3 に示す．

Algorithm 3 GroupbyMAD

Functionality: $[\![K']\!], [\![\vec{med}4v]\!] \leftarrow GroupbyMAD([\![K]\!], [\![\vec{v}]\!])$
Input: $[\![K]\!], [\![\vec{v}]\!]$
Output: $[\![K']\!], [\![\vec{med}4v]\!]$

1. $[\![\vec{v'}]\!], \{\{\sigma^{-1}\}\}, [\![K']\!], [\![\vec{med}2v]\!] := GroupbyMedian([\![K]\!], [\![\vec{v}]\!])$
2. $[\![\vec{m}2]\!] := Repeat([\![\vec{med}2v]\!], length(\vec{v'})/length(\vec{med}2v))$
3. $[\![\vec{d}2]\!] := Abs([\![\vec{v'}]\!] \times_p 2 -_p [\![\vec{m}2]\!])$
4. $[\![\vec{d2'}]\!] := InvertSort([\![\vec{d}2]\!], \{\{\sigma^{-1}\}\})$
5. $\_, \_, \_, [\![\vec{med}4v]\!] := GroupbyMedian([\![K]\!], [\![\vec{d2'}]\!])$
6. Output $[\![K']\!], [\![\vec{med}4v]\!]$

入力パラメータ，出力パラメータの平文値での関係を説明する．

入力パラメータ $\vec{v}$ は，中央絶対偏差を求める列，K は，$\vec{v}$ の各要素に対するキー行列である．

出力パラメータ K' は，キー行列 K をユニーク化し，ソートしたもの．$\vec{med}4v$ は，K の各グループに対し，$\vec{v}$ の各中央絶対偏差を求め，4 倍した配列である．

中央値は，要素数が偶数の場合中央の 2 値の平均値を求めるが，秘密計算では除算はコストがかかるため，都度除算せず，中央値の 2 倍の値を返却するとする．そのため，中央絶対偏差も，4 倍の値を返却する．

3.3 秘密計算による順位への変換

秘密計算による順位への変換アルゴリズムを, Algorithm 4 に示す．

Algorithm 4 Rank

Functionality: $[\![E'']\!] \leftarrow Rank([\![\vec{k}]\!], [\![E]\!])$
Input: $[\![\vec{k}]\!]$, $[\![E]\!]$
Output: $[\![E'']\!]$

1. $[\![E']\!] := ChangeFormat([\![\vec{k}]\!], [\![E]\!])$
2. $\{\{\sigma^{-1}\}\} := GetSortingPermutation(E')$
3. $[\![\vec{seq}]\!] := [\![1, 2, ..., m_e, 1, 2, ..., m_e, ...]\!]$
4. $[\![\vec{seq'}]\!] := InvertSort([\![\vec{seq}]\!], \{\{\sigma^{-1}\}\})$
5. $[\![\vec{v'}]\!], \{\{\sigma'^{-1}\}\}, [\![K']\!], [\![\vec{mean}v]\!] := GroupbyMean([\![E']\!], [\![\vec{seq'}]\!])$
6. $[\![L]\!] := Concatenate([\![K']\!], [\![\vec{mean}v]\!])$
7. $[\![R]\!] := Concatenate([\![E']\!], [\![\vec{seq}]\!])$
8. $[\![J]\!] := Join([\![L]\!], [\![R]\!], [0, 1], [0, 1])$
9. $[\![SK]\!] := Concatenate(at([\![J]\!], 0), at([\![J]\!], 2))$
10. $\{\{\sigma''^{-1}\}\} := GetSortingPermutation([\![SK]\!])$
11. $[\![S]\!] := Concatenate(at([\![J]\!], 0), at([\![J]\!], 3))$
12. $[\![\vec{S'}]\!] := Sort([\![S]\!], \{\{\sigma''^{-1}\}\})$
13. $[\![\vec{x}]\!], [\![E'']\!] := RevertFormat([\![S']\!])$
14. Output $[\![E'']\!]$

入力パラメータ，出力パラメータの平文値での関係を説明する．

入力パラメータ E は，順位に変換したい行列，$\vec{k}$ は，E の各行に対応するキー ID である．キー ID は，ユニークである必要がある．

出力パラメータ E'' は，E を，順位に変換した行列である．同値（同順位）は，平均の順位に変換する．

3.4 秘密計算によるスピアマンの順位相関係数を用いた階層型クラスタリング

階層型クラスタリングでは，Algorithm 2 に示したように，入力された各データを初期クラスタとし，全初期クラスタの組み合わせについて，距離テーブルを作成する．その後，距離テーブルの中で，一番近い距離のクラスタ同士を結合し，結合元のクラスタの距離を距離テーブルから削除し，新しいクラスタの距離を距離テーブルに追加する．距離の計算にかかわる処理は，(1) 初期クラスタからの距

離テーブルの作成，(2) 結合クラスタの距離計算，の 2 か所となる．(2) の計算は群平均法を用いており，今回のアルゴリズム変更には関係ない．そのため，今回，(1) の計算を，ユークリッド距離から，ピアソンの相関係数の計算に変更した．(1) の計算は，Algorithm 2 の step4 において行っているため，ここでの距離の計算を，ピアソンの相関係数に変更する．

また，DNB 分析では，Algorithm 1 に示した通り，階層型クラスタリングの結果を，閾値 θ_2, θ_3 で，修正する．

$1-\theta_2$ は，結合可能なクラスタ間の最大距離である．そのため，Algorithm 2 の step5 の for ループを最後まで実施する必要はなく，ループ内で，距離 $[\![d]\!]$ と $1-[\![\theta_2]\!]$ を比較し，処理を中断することとする．この時，処理を打ち切るための判定値は，サーバ上で復号する．

θ_3 は，クラスタ結果から，$[\![\theta_2]\!] \times max_{cnt}$ 以上のクラスタのみ取り出し，出力するための閾値である．max_{cnt} はクラスタ結果のうち，最大の要素数である．この処理を行うため，階層型クラスタリングの出力結果に，クラスタ ID テーブル $[\![C]\!]$ を追加する．C は，1 列目が，入力 X に対応する行番号 (=初期クラスタ ID)，2 列目が，クラスタ ID となるテーブルである．

変更後の階層型クラスタリングのアルゴリズムを，Algorithm 5 に示す．

Algorithm 5 秘密計算階層型クラスタリング (相関係数, 閾値 θ_2 対応)

Functionality: $[\![L]\!], [\![C]\!] \leftarrow$ $AgglomerativeClusteringS([\![X]\!], [\![\theta_2]\!])$
Input: $[\![X]\!], [\![\theta_2]\!]$. X は，順位で，$n \times m$ のテーブル
Output: $[\![L]\!], [\![C]\!]$

1. クラスタ ID の初期化 $[\![cid_{new}]\!] := [\![n]\!]$
2. クラスタ ID テーブル $[\![C]\!]$ の初期化
3. 出力 $[\![L]\!]$ を空のテーブルとして初期化
4. 全クラスタ間の距離テーブル $[\![D]\!]$ を計算
   $[\![D]\!] := CalcPeasonr([\![X]\!])$
5. $i := 0, max := n-2$
6. $[\![D]\!]$ から, 最も近いクラスタの ID($[\![cid_1]\!], [\![cid_2]\!]$) と，距離 $[\![d]\!]$ を取得
7. $cond := Reveal([\![d]\!] \leq_p (1-[\![\theta_2]\!]))\&(i \leq max)$
8. while $cond$
9. 新しいクラスタ ID$[\![cid_{new}]\!]$ と各クラスタ間の距離と，要素数 $[\![num]\!]$ を計算し，$[\![D]\!]$ を更新 (群平均法)
10. $[\![C]\!]$ を，新しいクラスタの情報で更新
11. $[\![L]\!]$ に，新しいクラスタの結合情報，($[\![cid_1]\!], [\![cid_2]\!], [\![d]\!], [\![num]\!]$) を追加
12. $[\![cid_{new}]\!] := [\![cid_{new}]\!] +_p 1$
13. $i := i+1$
14. $[\![D]\!]$ から, 最も近いクラスタの ID($[\![cid_1]\!], [\![cid_2]\!]$) と，距離 $[\![d]\!]$ を取得
15. $cond := Reveal([\![d]\!] \leq_p (1-[\![\theta_2]\!]))\&(i \leq max)$
16. Output $[\![L]\!], [\![C]\!]$

$CalcPeasonr$ で，ピアソンの相関係数の計算を行うとする．計算内容については，基本的には，式 (4) を秘密計算の四則演算，実数除算を用いて置き換えたものとなり，記述は割愛する．

3.5 DNB 分析

秘密計算による DNB 分析のアルゴリズムを, Algorithm 6 に示す．

Algorithm 6 秘密計算 DNB 分析

Functionality: $[\![O]\!] \leftarrow DNB([\![\vec{g}]\!], [\![C]\!], [\![E]\!], [\![\theta_1]\!], [\![\theta_2]\!], [\![\theta_3]\!])$
Input: $[\![\vec{g}]\!]$, $[\![C]\!]$, $[\![E]\!]$, $[\![\theta_1]\!]$, $[\![\theta_2]\!]$, $[\![\theta_3]\!]$
Output: $[\![O]\!]$

1. $[\![C']\!] := ChangeFormat([\![\vec{g}]\!], [\![C]\!])$
2. $[\![E']\!] := ChangeFormat([\![\vec{g}]\!], [\![E]\!])$
3. $[\![\vec{k}]\!], [\![\vec{c_m}]\!] := GroupbyMAD([\![C']\!])$
4. $\_, [\![\vec{e_m}]\!] := GroupbyMAD([\![E']\!])$
5. $[\vec{f}] := ([\![\theta_1]\!] \times_p [\![\vec{c_m}]\!]) \leq_p [\![\vec{e_m}]\!]$
6. $[\![GCE]\!] := Concatenate([\![\vec{k}]\!], [\![\vec{c_m}]\!], [\![\vec{e_m}]\!])$
7. $[\![GCE']\!] := Filter([\![GCE]\!], [\vec{f}])$
8. $[\![\vec{g'}]\!] := At([\![GCE']\!], 0)$
9. $[\![GE]\!] := Concatenate([\![\vec{g}]\!], [\![E]\!])$
10. $[\![GE']\!] := Join([\![\vec{g'}]\!], [\![GE]\!], 0, 0)$
11. $[\![GE']\!]$ の 1 列目を $\vec{g''}$, $[\![GE']\!]$ の 2 列目以降を $[\![E'']\!]$ とする
12. $[\![E'''\!] := Rank(\vec{g''}, [\![E'']\!])$
13. $[\![L]\!], [\![DC]\!] := AgglomerativeClusteringS([\![E'''\!], [\![\theta_2]\!])$
14. $[\![\vec{k'}]\!], [\![\vec{cnt}]\!] := GroupbyCount(At([\![DC]\!], 1))$
15. $[\![th_{cnt}]\!] := Max([\![\vec{cnt}]\!]) \times_p [\![\theta_3]\!]$
16. $[\vec{f'}] := [\![th_{cnt}]\!] \leq_p ([\![\vec{cnt}]\!] \times_p 100)$
17. $[\![CN]\!] := Concatenate([\![\vec{k'}]\!], [\![\vec{cnt}]\!])$
18. $[\![CN']\!] := Filter([\![CN]\!], [\vec{f'}])$
19. $[\![CGCE]\!] := Concatenate(At([\![DC]\!], 1), [\![GCE']\!])$
20. $[\![O]\!] := Join([\![CGCE]\!], [\![CN']\!], 0, 0)$
21. Output $[\![O]\!]$

Algorithm 6 では，提案した $GroupbyMAD$, $Rank$, $AgglomerativeClusteringS$ を呼んで DNB 分析を実現した．

Step4 までで中央絶対偏差を求めている．

Step5〜7 で，閾値 θ_1 により，揺らぎが大きい遺伝子 ID のピックアップを行う．

Step8〜12 で，順位への変換を行う．

Step13 で階層型クラスタリング及び，閾値 θ_2 によるフィルタを行う．

Step14 以降で，閾値 θ_3 による出力対象の選定及び，出力の整形を行う．閾値 θ_3 として，0.5 を用いるが，これを百分率に変換し，$\theta_3 = 50$ として入力する．出力 $[\![O]\!]$ の，O は，最終的に選択されたクラスタ ID，クラスタに対応する遺伝子 ID リスト，遺伝子に対する対照群の MAD，遺伝子に対する実験群の MAD, クラスタの要素数，からなる行列である．

4. 評価

4.1 システム構成

秘密計算システムとして，MEVAL3[16] を用いた．MEVAL3 は，サーバクライアント型のシステムであり，秘密計算クライアント上に秘密計算プログラムを定義し，3 台の秘密計算サーバへリクエストを送信して，秘密計算サーバが秘密計算を実行し，秘密計算クライアントへ計算結果のシェアを返却する．

4.2 環境

Azure 環境上に，秘密計算サーバ 3 台および，秘密計算クライアント 1 台を構築した．表 1 に測定環境を示す．測定は 3 回行い，平均値を記載する．

表 1 測定環境

	サーバ	クライアント
VM サイズ	Standard D4s v4	Standard D4 v4
vCPU 数	4	4
RAM	16Gbyte	16Gbyte
NW	5〜30Gbps	5〜30Gbps
OS	Rokcy Linux	Rokcy Linux

4.3 使用データ

DNB ツール [8] に同梱されたサンプルデータを用いた．データ件数は，6,467 件，対照群の個体数は 17 個体，実験群の個体数は 18 個体である．

つまり，遺伝子情報リスト $\vec{g}$ は，長さ 6,467 の配列，対照群の遺伝子発現量の行列 C は，遺伝子 6,467 個× 17 個体の遺伝子発現量を値に持ち, 実験群の遺伝子発現量の行列 E は，遺伝子 6,467 個× 18 個体の遺伝子発現量を値に持つ．

なお，基本的なパラメータとして，$\theta_1 := 2$, $\theta_2 := 0.75$, $\theta_3 := 0.5$ を用いた．

4.4 計算結果

DNB 分析の Two-step Method のステップ１，ステップ２について，以下の結果となった．

(1) ステップ 1 の揺らぎの大きい遺伝子のピックアップについて，平文アルゴリズムと同じ遺伝子をピックアップすることができた．

(2) ステップ 2 のクラスタリングについて，入力値を順位に変換し，階層型クラスタリングを実施し,θ_2,θ_3 のフィルタリングを行った．θ_2 でフィルタリングされた範囲の結合情報について，結合対象のクラスタ ID，結合順序は 1 か所を除いて一致し，ピアソンの相関係数は，平文アルゴリズムの結果と，有効数字 5 桁以上が一致した．結合順序が一致しなかった箇所は，ピアソンの相関係数が同値となるペアが，2 組存在していた箇所であったため，計算結果として問題ないと考える．また，θ_3 のフィルタリングを行った最終的な出力 $[\![O]\!]$ について，表 2 に示す．$\vec{g}$ を対応する遺伝子名に変換すると，平文アルゴリズムと同じ結果となる．

表 2 step2 の出力 O

$\vec{g}$	$\vec{cid}$	$\vec{cnt}$	$\vec{e}/4 \times 10^{-3}$	$\vec{c}/4 \times 10^{-3}$
93	334	16	0.076	0.1655
237	334	16	0.102	0.2665
1032	334	16	0.083	0.1705
1102	334	16	0.180	0.400
1269	334	16	0.059	0.1825
1551	334	16	0.075	0.2835
1768	334	16	0.392	1.0515
1923	334	16	0.061	0.1645
2090	334	16	0.059	0.176
2369	334	16	0.263	0.590
3082	334	16	0.241	0.5395
4918	334	16	0.074	0.1905
5295	334	16	0.020	0.0755
5422	334	16	0.049	0.132
5480	334	16	0.101	0.3785
5935	334	16	0.027	0.061

4.5 処理時間

データ件数を 1000 件, 5000 件, 全件に変えて測定した結果を，表 3 に示す．測定時，サーバ間の遅延は，2.46 ミリ秒, サーバ-クライアント間の遅延は，1.71 ミリ秒であった．6,467 件の処理時間は全体で 23,128 ミリ秒であり，そのうち階層型クラスタリングの処理時間は 20,502 ミリ秒であった．件数が増えるにつれ，階層型クラスタリングの処理時間が全体に占める割合が大きくなることから，DNB 分析の性能向上のポイントは，階層型クラスタリングの高速化であることが分かる．

表 3 DNB 分析処理時間 (ミリ秒)

件数	全体	内訳		
		MAD	Rank	クラスタリング
1,000	1,463	469	275	625
5,000	14,377	1,266	775	12,220
6,467	23,128	1,572	924	20,502

今回，階層型クラスタリングの処理をすべて終えてから閾値 θ_2 によるフィルタリング処理をするのではなく，階層型クラスタリングの処理の中で閾値 θ_2 の判定を行い，処理を打ち切る方針とした．これによる処理時間削減の効果を表 4 に示す．$\theta_2 = 0$ は閾値による打ち切りを行わない場合，$\theta_2 = 0.75$ は相関係数が 0.25 を超えた際に階層型クラスタリングの処理を打ち切る場合の処理時間である．6,467 件では，処理時間が 4 分の 1 以下になっていることが分

かる．

表 4 θ_2 に対する階層型クラスタリング処理時間比較 (ミリ秒)

件数	$\theta_2 = 0.75$	$\theta_2 = 0$
1,000	625	8,917
5,000	12,220	64,818
6,467	20,502	95,183

今回，階層型クラスタリングを相関係数に対応した．メトリックをユークリッド距離とした場合に比べ，相関係数とした場合の性能が劣化していないかを確認した．その結果を，表 5 に示す．大きな性能劣化はないが，6,467 件の時，相関係数の方が 4%処理時間が遅い．

表 5 メトリックに対する階層型クラスタリング処理時間比較 (ミリ秒)

件数	ユークリッド距離	相関係数
1,000	8,977	8,917
5,000	62,707	64,818
6,467	91,411	95,183

5. おわりに

本稿では，秘密計算を用い，DNB 分析を暗号化したまま処理するアルゴリズムを提案，実装した．6,467 件，対照群の個体数 17 個体，実験群の個体数 18 個体のデータに対し，平文アルゴリズムと同等の計算結果を得ることができた．また，処理時間については，23 秒となり，実用上問題ない数値と考える．

謝辞 本研究を実施するにあたり，DNB 解析のアルゴリズムについて，DNB 解析ソフトウェア [8] を参考にしています．本ツールの利用を許諾いただいた JST および，ツールの提供元である，合原ムーンショットプロジェクトに感謝します．

参考文献

[1] L. Chen, R. Liu, ZP. Liu, M. Li, K. Aihara. Detecting early-warning signals for sudden deterioration of complex diseases by dynamical network biomarkers. Sci Rep 2, 342 (2012). https://doi.org/10.1038/srep00342

[2] 合原ムーンショットプロジェクト. https://www.jst.go.jp/moonshot/program/goal2/21_aihara.html

[3] 複数の研究機関が持つゲノムデータを相互に開示せず分析する解析手法を開発 ～プライバシー保護データマイニング技術によるフィッシャー正確確率検定を世界で初めて実現～ https://group.ntt/jp/newsrelease/2016/07/12/160712a.html.

[4] K. Hamada, S. Hasegawa, K. Misawa, K. Chida, S. Ogishima, M. Nagasaki. Privacy-Preserving Fisher's Exact Test for Genome-Wide Association Study, 4th International Workshop on Genome Privacy and Security (GenoPri'17), 2017.

[5] 三品 気吹，五十嵐 大，濱田 浩気，菊池 亮．大規模データにも対応した秘密計算階層型クラスタリング．In SCIS, 2022.

[6] 三品 気吹，五十嵐 大，濱田 浩気，菊池 亮．秘密計算によるプライバシー保護階層型クラスタリング．In CSS, 2021.

[7] M. Oku. Two Novel Methods for Extracting Synchronously Fluctuated Genes. IPSJ Transactions on Bioimformatics, 12, 9-16, doi: 10.2197/ipsjtbio.12.9 (2019).

[8] DNB 解析ソフトウェア. https://www.sat.t.u-tokyo.ac.jp/moonshot/software/dnb-tool.html

[9] R. Gennaro, M. O. Rabin, and T. Rabin. Simplified VSS and Fact-Track Multiparty Computations with Applications to Threshold Cryptography. In PODC, pp. 101–111, 1998.

[10] scipy. https://www.scipy.org/

[11] 五十嵐 大，千田 浩司, 濱田 浩気，高橋 克巳. 軽量検証可能 3 パーティ秘匿関数計算の効率化及びこれを用いたセキュアなデータベース処理. In SCIS, 2011.

[12] 五十嵐 大，濱田 浩気，菊池 亮，千田 浩司．超高速秘密計算ソートの設計と実装 秘密計算がスクリプト言語に並ぶ日. In CSS, 2017.

[13] 菊池 亮, 濱田 浩気, 五十嵐 大, 高橋 元, 高橋 克巳. 横断的動線分析を秘密計算でやってみよう. In SCIS, 2020.

[14] 桐淵 直人，五十嵐 大，諸橋 玄武, 濱田 浩気 属性情報と履歴情報の秘匿統合分析に向けた秘密計算による高速な等結合アルゴリズムとその実装．In CSS, 2016.

[15] 五十嵐 大, M op/s を超える秘密計算上の初等関数群. In SCIS, 2020.

[16] 桐淵 直人，五十嵐 大，濱田 浩気，菊池 亮．プログラマブルな秘密計算ライブラリ MEVAL3. In SCIS, 2018.