2024（令和 6）年度　修士論文

適応度予測を用いた
データ可視化の半自動生成に関する研究

Research on Semi-automatic Generation of Data Visualization Using Fitness Prediction

2024/2/29 提出

東京都立大学大学院 システムデザイン研究科 情報科学域
博士前期課程
山上 遼馬

指導教員 高間 康史 教授

目次

1. はじめに 3

2. 関連研究 5

2.1. 可視化の描画手法 5

2.1.1. 可視化の描画の仕様と仕様に基づいた描画手法 5

2.1.2. 可視化文法を用いた描画手法 6

2.2. 可視化の自動生成手法 7

2.2.1. 情報推薦による可視化の自動生成 8

2.2.2. 機械学習による可視化の自動生成 8

2.3. 進化計算 9

2.3.1. 遺伝的アルゴリズムと遺伝的プログラミング 9

2.3.2 対話型進化計算 10

2.4 適応度予測 11

3. 適応度予測を用いた遺伝的プログラミングによる Vega-Lite コードの半自動生成 12

3.1. GTYPE の仕様 12

3.2. 初期個体集団の作成 16

3.3. 適応度の予測 16

3.3.1. 個体間の類似度と部分構造 17

3.4. 個体集団の更新 17

3.4.1. エリート個体の選択 18

3.4.2. 交叉 18

3.4.3 突然変異 21

3.5 対話型可視化生成システム 24

4. 実験 27

4.1. 評価実験 27

4.2. ユーザ実験 32

4.2.1. 実験協力者が生成した可視化 38

4.2.2. 実験結果の分析 43

5. おわりに 45

付録 A mark 構造内で設定した項目と encoding 構造内で設定した構造 46

参考文献 48

1. はじめに

本論文は，Vega-lite のコードを対象として適応度予測を用いた対話型遺伝的プログラミングを適用することにより，ユーザの評価負担を軽減しながらコードを進化させて新たな可視化を生成する手法を提案する．

近年では，データの活用・分析に対する関心が高まってきており，データの処理結果をわかりやすく分析者に示すことができるデータの可視化についても注目が集まっている．だが，適切な可視化を作成するには，可視化や統計の知識・スキルが必要となり，それらの知識がないユーザには困難である点が指摘されている[8].

可視化の描画に用いられるプログラミング言語としては, D3.js[1], Vega[2], Vega-Lite[3]などが挙げられる．これらは統計グラフなどの一般的に用いられる統計グラフをベースとした可視化を生成するために開発された言語であり，これらを用いることで，可視化の実装作業の負担を軽減することができる．しかし，可視化の描画には，適切な可視化表現を選択する知識・スキルも必要であり，それらの知識を持たないユーザが適切な可視化を描画するのは困難である．

この問題の解決策として，機械学習を用いた可視化の自動生成に関する研究が行われている[13]．適切な可視化表現を選択するなどの知識・スキルが不要となること，可視化生成作業の負担が軽減されることなどの利点から，データサイエンスでの活用が期待できるが，自動生成では，ユーザの意向，趣向を反映した可視化の生成を行うことができないと考える．

ユーザの意向を反映しつつ，ユーザの負担を軽減する手法として，ユーザに可視化の候補を提示し，ユーザが下した評価を基に新たな可視化を生成する，対話型遺伝的プログラミングを用いた手法が提案されている[23]．遺伝的アルゴリズム（Genetic Algorithm, GA）は，進化論の自然淘汰説の考えに基づいてデータを操作し，最適解探索を行う計算手法であり，進化計算の手法の一種である[17]．GA で扱うデータは，細胞内の染色体に相当する遺伝子型(GTYPE)と，遺伝子型の発現を示す表現型(PTYPE)に分かれる．GTYPE で複雑な構造を扱えるように GA を拡張したものが遺伝的プログラミング（Genetic Programing, GP)である[19]．また，進化計算に人の評価を組み込み，評価基準を明確に定義できない問題にも進化計算を適用できるようにしたのが，対話型進化計算であり，GA，GP に人の評価を組み込んだものをそれぞれ対話型 GA，対話型 GP と呼ぶ[20]．対話型進化計算を用いて，似顔絵の生成[30]や，フォントの生成などが行われている．前述の，対話型遺伝的プログラミングを用いた可視化の生成手法では，Vega-Lite のコードを対象に対話型遺伝的プログラミングを適用している[23]．ユーザ実験の結果からデータの意味を読み取りやすい可視化を生成可能であることが示されている．しかし，ユーザが，各世代の可視化(個体)をすべて評価する必要があるため，可視化の評価にかかるユーザ負担の軽減が課題となっていた．

本論文では，可視化の評価に要するユーザの負担を軽減するために，Vega-lite のコードを対象として対話型遺伝的プログラミングに適応度予測を導入した手法を提案する．提案手法では，ユーザに，可視化

---

[1] https://d3js.org

[2] https://vega.github.io/vega/

[3] https://vega.github.io/vega-lite/

の組み合わせと可視化の種類，可視化に含まれるデータについての情報の 3 つの観点について，ユーザの意図を満たす可視化をそれぞれ選択してもらい，選択された可視化との類似度を用いてその他の可視化の適応度を算出する．従来の適応度予測を使わない手法では，次世代集団の更新ごとにユーザ評価が必要になっており，ユーザの評価負担が増していたが，提案手法では，ユーザが選択した可視化を保存しておき次世代以降の可視化の適応度計算に用いる．これにより，ユーザが評価する過程をスキップしてある程度自動で個体集団の更新が行えるため．これによって，ユーザの評価負担の軽減が期待できる．

評価実験では，提案手法を実装したプログラムを用いて次世代個体集団を生成する実験を行い，実験結果より，ユーザの意向を反映した可視化を生成可能であることを示す．また工学系の大学院生を対象に，各都道府県の COVID-19 の感染状況のデータを利用して，提案手法を実装した可視化の半自動生成システムでユーザ実験を行い，実験結果より，ユーザ負担を軽減しながらも，ユーザの評価を反映した可視化を生成可能であることを示す．また，実際に生成した可視化例を示し，生成される可視化の特性などについて考察する．

2. 関連研究

2.1. 可視化の描画手法

近年，可視化の描画を行うライブラリや，可視化を簡潔なコードで描画する可視化文法が注目を集めている．以下では可視化の仕様決定と，仕様に基づく可視化の描画手法，可視化文法を用いた描画手法について述べる．

2.1.1. 可視化の描画の仕様と仕様に基づいた描画手法

近年，可視化の描画を行うツールが注目を集めている．可視化の描画には，可視化に特化したドメイン固有言語やツールを用いることが一般的となっている[3][4][5]．これらの言語では，可視化の描画の仕様に基づいて図の描画が行われる．可視化の描画の仕様は，可視化を描画する上で必要な処理の流れを示したものである．可視化の描画の流れは，まず，データセットからデータ変数を定義し，代数処理，尺度化処理，統計処理を行う．次にデータ変数に幾何処理，座標系処理，装飾処理を行い，可視化を描画する[1]．幾何処理，座標系処理，装飾処理については，以下のような処理が行われる[1]．

- 幾何処理 … データ変数を図形などの幾何的要素に変換する処理
- 座標系処理 … 幾何的要素を描画スペースにプロットする処理
- 装飾処理 … 描画した幾何的要素を色やテクスチャで装飾する処理

可視化の汎用的なツールが開発される以前においては，ユーザ自身でこの手順を実装する必要があり，実装作業の負担が重くなっていた[2]．また，プログラミングの知識やスキルが必要となり，それらを持たないユーザには困難である点が指摘されていた[2]．

実装作業の負担を軽減するために，可視化を用いたソフトウェアの設計を支援するデザインパターンが提案されている[2]．このデザインパターンでは，データ表現の処理とグラフィックスの描画処理とそれらの関係性が定義されている，このデザインパターンに基づいてソフトウェアを設計することで，可視化の描画処理の実装が容易になり，実装作業の負担を軽減に有効であることが報告されている[2]．

現在では，可視化の描画に特化したライブラリが用いられるようになり，様々な可視化を描画することが可能になっている．可視化の描画を行う言語としては，Java ライブラリの Prefuse[3]，JavaScript ライブラリの Protovis[4]，D3.js[5]などが知られている．

Prefuse は，実用的な可視化を作成およびカスタマイズが行える Java のライブラリであり，基本的な可視化を描画することが可能である[3]．図 2.1 に Prefuse を用いた可視化の例を示す．Java アプレットを用いて可視化の描画を行えるため，Web ページへ可視化の埋め込みも可能である．ただし可視化の表示には，専用プラグインのインストールが必要となり，プラグインがインストールされていない環境では可視化の表示ができないという問題もあった．

Protovis は，実用的な視覚化を作成およびカスタマイズが行える JavaScript のライブラリであり，Prefuse と同様に基本的な可視化を描画することができる[4]． JavaScript のライブラリであるため，追加でプラグインをインストールする必要がなくなり，環境を問わずに可視化の表示が行える利点がある．

D3.js は，Protovis の後継となる JavaScript のライブラリであり，多種多様な可視化を描画することができる[5]．Protovis では抽象的なインタフェースが採用されており，中間となる処理を挟んで可視化の描画を行っていたが，D3.js では DOM を直接操作して可視化の描画を行う．これによって，描画処理を

高速で行えるようになり，描画できる可視化の自由度が Prefuse や Protovis に比べて高くなっている．図 2.2 に D3.js で描画可能な可視化の例を示す．

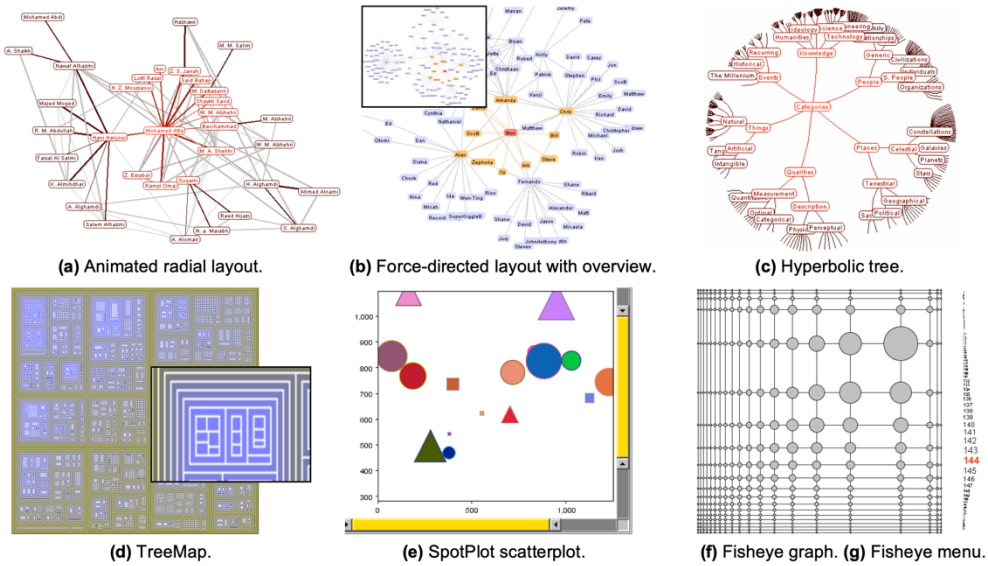


図 2.1 Prefuse による可視化の例([3]より抜粋)

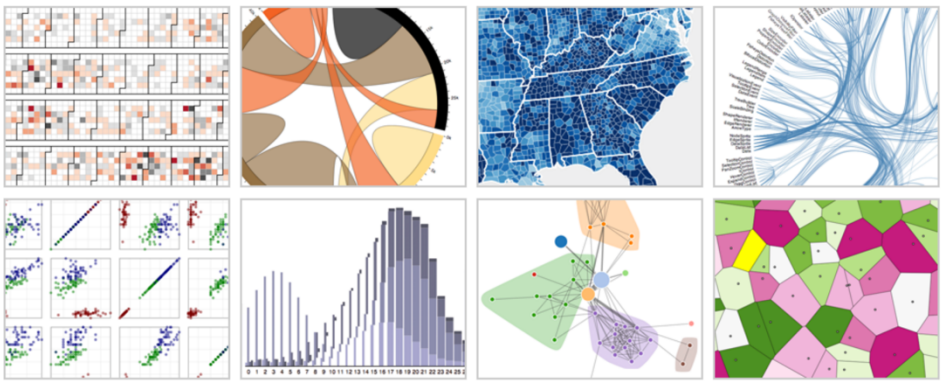


図 2.2 D3.js による可視化の例([5]より抜粋)

2.1.2. 可視化文法を用いた描画手法

D3.jsなどの可視化の描画に特化したライブラリは，プログラミング言語の知識や，スキルを持つユーザにとっては，容易に可視化の実装が行うことが可能だが，それらの知識やスキルを持たないユーザにとっては利用が困難である点が指摘されている[6]．

可視化文法は，プログラミングの知識やスキルを持たないユーザであっても，可視化の描画が行えるように．可視化の描画処理を簡潔なコードで記述可能としたものであり，VegaやVega-Liteなどが代表的である．

Vega は，json構文を採用した可視化文法で，D3.jsを用いて実装されている[6][7]．可視化のデザインやインタラクティブな動作をjson形式で記述することが可能である[7]．D3.jsと比較すると，自由度は下がるが，簡潔な記述でカスタマイズ可能な一般的な可視化を生成することが可能である[6]．また，Vegaは，高級言語や可視化システムなどといった高レベルのツールの基盤となることを目標として開発

されており[6]，実際に Vega-LiteなどがVegaを用いて実装されている．

Vega-Lite は，json構文を採用した可視化文法で，Vegaの簡易版として設計された[8]．D3.jsやVegaと比べて，描画できる可視化の種類は少ないが，コードがより簡潔になっているため，jsonのコードからデータと視覚的な出力の関係を読み取りやすくなっている．Vega-Liteには，Vegaと同様に．複数の可視化を重ね合わせるレイヤー機能，複数の可視化を並べて表示するマルチビュー機能，ある可視化の操作に合わせてほかの可視化が変化するインタラクティブな可視化を生成する機能がある[2]．またVegaと下位互換性があり，Vega-Liteのjsonコードは，Vegaのjsonコードに変換することが可能である．図2.3にVega-Liteのコードと可視化の例を示す．図2.3の例では，都道府県ごとの累計の陽性者数が棒グラフとして描画されている．

```
{
  "width" : 300,
  "height" : 500,
  "mark": "bar",
  "encoding": {
    "y": {
      "field": "name",
      "type": "nominal"
    },
    "x": {
      "field": "npatients",
      "type": "quantitative"
    }
  }
}
```

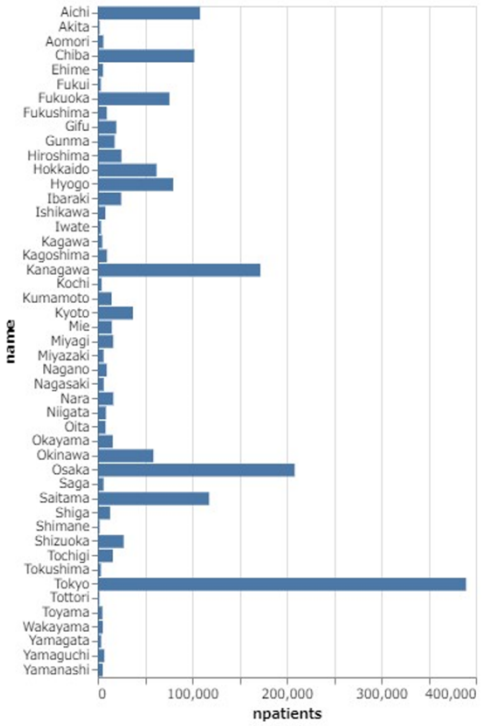


図 2.3 Vega-Lite のコードと可視化の例

2.2. 可視化の自動生成手法

Vega や Vega-Lite などの可視化文法は，プログラミング言語の知識やスキルを十分持たないユーザであっても，可視化の作成が行うことを可能にしているが，データから得られる知見をわかりやすく示すことができる可視化を作成するには，可視化や統計の知識・スキルが必要となる．それらの知識がないユーザには，適切な可視化の作成が困難である点が指摘されている．

近年では，可視化を自動的に生成する手法が注目を集めている．こうした手法を用いれば，可視化や統計の知識・スキルがないユーザであっても適切な可視化を作成可能でとなることが期待できる．可視化の自動生成手法は，Voyager や VizML，KG4Vis のように情報推薦を用いた手法と，Data2Vis や NL4DV のように機械学習を用いた手法に分けられる．

2.2.1. 情報推薦による可視化の自動生成

情報推薦による手法では，データセットから学習したモデルを用いてユーザに複数の可視化の提示を行う．

Voyager は，任意の json 形式と表形式のデータから可視化の推薦を行う対話型システムである[9]．Voyager では，ユーザが任意のデータ変数を選択すると，そのデータ変数を用いた Vega-Lite の可視化が複数生成される[9][10]．

Voyager は，ユーザが選択したデータ変数を用いて，可視化の推薦を行うが，VizML と KG4Vis ではデータセットと可視化のペアから特徴を抽出し，学習したモデルを用いて可視化の推薦を行う．

VizML では，データセットと可視化のペアから，データの特徴と，グラフの種類・軸といった可視化の構造を抽出し，可視化のデザインを推薦できるようにモデルを学習する．学習したモデルを用いることで，可視化の種類やデータ変数のプロット方法の推薦が行え，それに基づいて可視化の生成を行うことが可能である．

KG4Vis は，データセットと可視化のペアから特徴を抽出し，ナレッジグラフの構築を行う．ナレッジグラフから，抽出した特徴間の関係性を考慮して特徴ベクトルを学習する．学習結果を用いて推論を行い，可視化の推薦と可視化の推薦規則の生成を行う[11]．

VizML では，可視化の推薦理由を説明することが困難だが，ナレッジグラフベースの推薦を行う KG4Vis では，結果とともに推薦基準を示す可視化の推薦規則を提示するので，可視化が推薦された理由を簡単に説明可能である[12]．

2.2.2. 機械学習による可視化の自動生成

Data2Vis[13]は, 機械翻訳で使われているアルゴリズム Seq2Seq を用いてデータセットから Vega-Lite コードの自動生成を行う．Data2Vis のモデルは，エンコーダとデコーダの二つのニューラルネットワークに分かれている．エンコーダでデータセットを入力し，隠れ状態ベクトルをデコーダへ出力する．デコーダで隠れ状態ベクトルと文脈ベクトルを基にして，Vega-Lite の json コードを出力している．

NL4DV[14]は, 自然言語処理を用いて, 英語のクエリから Vega-Lite のコードの自動生成を行う python のライブラリである．NL4DV では，データセットと，それに関する自然言語のクエリを入力すると，Vega-Lite の文法を学習した言語モデルを用いて，Vega-Lite のコードが生成される[14]．例えば IMDB のデータセットと”Create a histogram showing distribution of IMDB ratings”を入力すると，IMDB の評価分布を示すヒストグラムを示す Vega-Lite のコードが生成される．

Mitra[15]らは，NL4DV を双方向の対話が行えるように拡張し，可視化生成を行うチャットボットを提案している. 拡張した NL4DV では, 自然言語のクエリを入力したときに, Vega-Lite のコードの他に, フォローアップとなる追加の質問文を出力し，チャットボットと利用者間で会話をしながら可視化の生成を行うことが可能となる[15]．

2.3. 進化計算

2.3.1. 遺伝的アルゴリズムと遺伝的プログラミング

遺伝的アルゴリズム(Genetic Algorithm, GA)と遺伝的プログラミング(Genetic Programing, GP)は，進化論の自然淘汰説の考えに基づいて，データを操作し，最適解探索を行う計算手法である[18]．GA,GP で扱うデータは，細胞内の染色体に相当する遺伝子型の GTYPE と，環境内での行動や構造といった遺伝子型の発現を示す表現型の PTYPE に分かれる[18]．GA の GTYPE には数値の配列が用いられるが，GP の GTYPE ではプログラムのような複雑な構造を扱える[19]．

一般的な GA，GP では，まずランダムに初期個体の集団を生成する．次に集団内の個体の PTYPE に対し，問題に応じて適切に定義した適応度関数を用いて適応度の計算を行う．個体の適応度に基づいて生き残る個体の選択が行われ，選択された個体の GTYPE にオペレータが適用されて，次世代個体が生成される．この適応度計算から次世代個体の生成までの処理を繰り返し，最終世代の個体の中で，最も適応度の高い個体を解として出力する[17]．

個体の選択のアルゴリズムは，ルーレット選択とトーナメント選択，エリート選択がある．
ルーレット選択は，式(2.1）で定義される各個体の選択確率を用いて生き残る個体を選択する．ここで，個体iの適応度をf_i，個体数をNとする[17]．

$$p_i = \frac{f_i}{\sum_{k=1}^{N} f_k} \quad \cdots (2.1)$$

ルーレット選択では，適応度が高い個体の選択確率が高くなるため，高適応度の個体が生き残りやすくなるが，個体間の適応度の格差が激しい場合は，低適応度の個体が選ばれる確率がかなり低くなり，解の多様性が失われ，探索が早期に収束する可能性がある．一方トーナメント選択では，集団の中から予め決めたトーナメントサイズの個体を取り出し，トーナメントの中で，最も適応度の高い個体を選択する．トーナメントサイズを小さくすれば，適応度の低い個体も生き残りやすくなるため，集団内の個体の多様性を保つことができる．エリート選択は，ルーレット選択やトーナメント選択と併用して使われ，適応度の高い個体を一定数残すことで，次世代個体に適応度の低い個体のみが残るという状況を避けられる[17]．

オペレータは，生物の遺伝子操作をモデルにしたもので，交叉と突然変異がある．交叉は親となる個体を選び，GTYPE の一部を入れ替え，子となる個体を生成する操作のことであり，オペレータの基本的な操作である[17]．

突然変異は，個体の GTYPE の一部を変化させる操作のことである[18]．一般的な GA，GP では，オペレータの適用箇所はランダムに選択される．
GA の派生の一つに，若林らは交差手法の適応的選択機能を取り入れた手法を提案している[19]．

若林らの提案した手法では，世代Tの個体 x_i^Tに対してエリート度 $ED(x_i{}^T)$を定義し，エリート度に応じて交叉手法を変えている．エリート度は，式(2.2)のように定義される．ここで$Anc_i^T(j)$ は個体 x_i^T の世代 $T-j$ における先祖の集合であり，$Elite_i^T(j)$ は$Anc_i^T(j)$ 内のエリート個体の集合，$\beta(0 \le \beta \le 1)$ はエリート影響度係数である[18]．

$$ED(x_i{}^T) = \frac{\sum_{j=1}^{l_max}\left\{\left|Elite_i^T(j)\right| \times \beta^j\right\}}{\sum_{j=1}^{l_max}\left\{\left|Anc_i^T(j)\right| \times \beta^j\right\}} \cdots (2.2)$$

$Elite_i^T(j)$ は，式(2.3)のように定義される[18]．ここで，世代Tの個体の平均適応度をμ_T，適応度の標

準偏差をσ_Tとする．

$$Elite_i^T(j) = \{x_k^{T-j} \mid x_k^{T-j} \in Anc_i^T(j),\ \mu_{T-j} + \alpha \times \sigma_{T-j} \leq f(x_k^{T-j})\} \ldots (2.3)$$

エリート度を用いた交叉的手法の適応的選択の手順は以下の通りである．

ステップ 1： 交差を行う 2 つの親について適応度の和 ED を求める．

ステップ 2： 2 つの親がエリートかどうかを判定し，一方の親のみがエリートの場合は 1，両方の親がエリートの場合は 2 を ED に加える．

ステップ 3： ED がしきい値以上ならば 2 点交差，そうでなければ一様交差を実行する．

これによって，エリート個体同士の交差では，2 点交叉を行い，変化を少なくすることで，優秀な個体が個体集団に残る可能性が上がることが期待できる．非エリート個体の交叉では，交換する GTYPE の範囲を指定し，範囲内の部分を一度に交換する一様交叉を行い，変化を大きくすることで，優秀な個体が誕生する可能性が上がることが期待できる[18].

2.3.2 対話型進化計算

対話型進化計算は，進化計算の適応度関数の代わりに人間の評価を組み込んだものである[21]．これによって，個人の好みが影響するような，適応度関数を定義できないものに対しても，進化計算を行うことが可能である[21]．また人間の主観的評価を組み込むため，結果に感性を反映させることが可能である．GA，GP に人の評価を組み込んだものをそれぞれ対話型 GA，対話型 GP と呼ぶ．対話型進化計算の問題点としては，評価基準が変動する点[]，通常の進化計算と比べて時間がかかるため，世代あたりの個体数や探索世代数を少なくせざるを得ない点[20]，評価の負担が利用者に発生する点[20]，計算結果が初期個体に大きく依存する点[22]が挙げられる．

中川らは，対話型進化計算を用いたフォント自動生成システムを提案している[21]．このシステムは欧文書体のみを対象にしている．フォントは，アルファベットや参考とした書体名を記述する書体情報，欧文の造形的特徴を表すエレメント情報，エレメントの輪郭を表す輪郭情報，輪郭情報に記述された座標の位置を明示的に表す位置情報で定義され，XML 形式で記述される．XML で記述されたエレメント情報の操作を行う関数を定義し，操作関数の履歴を対話型 GA の GTYPE，フォントを基に描画された文字列を PTYPE としている．このシステムでは初期個体の偏りが計算結果に影響しないように，個体を複数回突然変異させることで，初期個体の多様性を持たせている[22]．次世代個体の生成前に，突然変異させる個体をユーザ側に選択させることで，ユーザが希望するタイミングで解を収束し，計算を終了できるようにすることで，任意のタイミングで計算を打ち切れるため，ユーザの負担を減らすことができている．また，ユーザの評価を反映しやすくすることで，ユーザの求める最適解に近い個体の生成を可能としている．

是永らは，対話型進化計算を用いたインテリアレイアウトの自動生成システムを提案している[22]．このシステムでは，机と椅子の配置と配置された家具に使用する色を GTYPE，配置と家具の色を基に作られたレイアウトを PTYPE とする．各個体の適応度は，机どうしの距離の平均と，机と扉の距離の平均，家具の散布度を用いて算出する局所的評価値と，ユーザの判定による大局的評価を掛け合わせて求める．これによって，ユーザによる評価は世代によらず適応度に強く反映される．局所的評価も適応度に強く反映されるため，ユーザがシステムに慣れていない場合にも対応が可能となっている．

山上らは，対話型遺伝的プログラミングを用いた可視化の半自動生成システムを提案している[23]．こ

のシステムは，Vega-Lite のコードを GTYPE，コードから生成される可視化を PTYPE として対話型遺伝的プログラミングを適用している．ユーザ実験の結果から，データの意味を読み取りやすい可視化を生成可能であることが示されている．しかし，ユーザが，各世代の可視化をすべて評価する必要があるため，可視化の評価にかかるユーザの負担が指摘されている[23].

2.4 適応度予測

適応度予測は，一部の個体の適応度を求め，他の個体の適応度は，その個体のGTYPEと実際に求めた適応度を基に予測を行う手法である[25]．対話型進化計算においては，GTYPEの評価の回数が減るので，ユーザの負担の軽減が可能である[28]．また通常のGA，GPにおいても，適応度の計算の回数が減るので，計算速度も向上することが報告されている[24].

適応度予測は，個体間の類似度を用いた手法とニューラルネットワークを用いて予測する手法の二つが代表的である[26].

個体間の類似度を用いた手法では，一部の個体のみを実際に評価し，他の個体については評価した個体との類似度を基にして，適応度の予測を行う[27].

ニューラルネットワークを用いて予測する手法では，個体の評価値と GTYPE を訓練データとしてユーザの評価特性を学習し，評価しなかった個体は学習済みニューラルネットワークを用いて適応度の予測を行う[29].

佐藤らは，適応度予測を用いた対話型進化計算を用いて似顔絵生成を行うシステムを提案している[30]．GTYPE を，顔の輪郭や，髪型，鼻，口といった顔のパーツの組み合わせで表現し，顔のパーツの組み合わせで表現される似顔絵を PTYPE として，対話型 GA を適用している．似顔絵の顔と表情を分けて進化させられるように，個体の GTYPE を顔の評価に関する部分と表情の進化の部分に関する部分を区別している．ユーザに提示個体から好みの 1 個体のみを選択，評価させ，ユーザが選択しなかった個体に対しては，システムが評価値を見積もる．学習済みニューラルネットワークで予測適応度の算出を行い，次世代個体となる解候補の生成を行っている[30].

3. 適応度予測を用いた遺伝的プログラミングによる Vega-Lite コードの半自動生成

本章では適応度予測を用いて，ユーザの評価の負担を軽減しながら，対話型遺伝的プログラミングを用いてデータ可視化の半自動生成を行うシステムを提案する．図 3.1 にシステムの流れを示す．

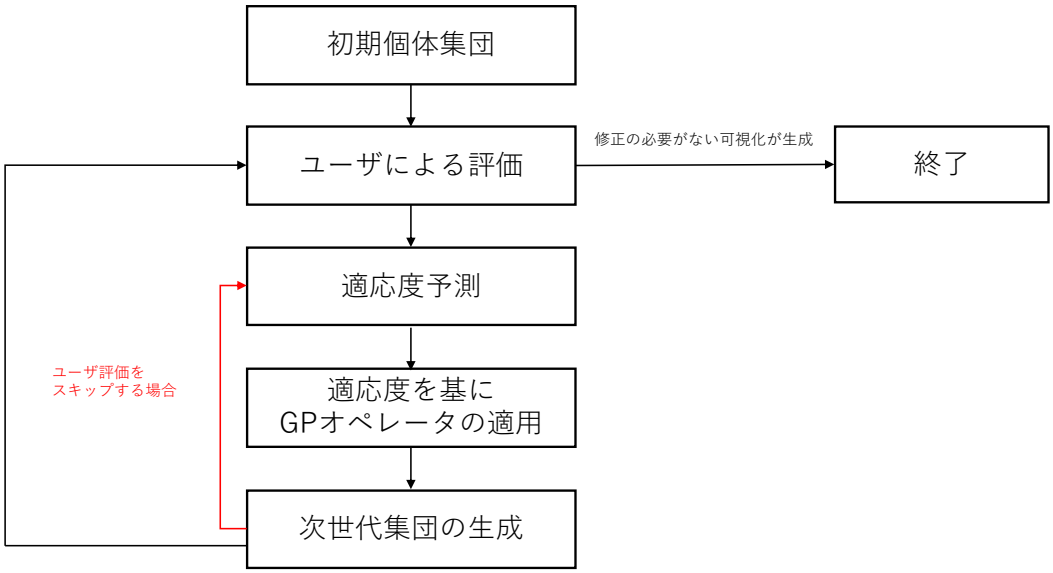


図 3.1 システムの処理の流れ

最初に初期個体集団の GTYPE から可視化を生成してユーザに提示する．ユーザは提示された各個体に対し，可視化の組み合わせと可視化の種類，可視化に含まれるデータについての情報の 3 つの観点について，自身の意図を満たすかを評価する．ユーザが 3 つの観点それぞれについて，意図を満たす個体をすべて選択する．このときに，これ以上修正が必要ないと判断した個体がある場合は，その個体を選びシステムを終了する．終了しない場合は，観点ごとにユーザが意図を満たすと判断した個体との類似度を用いてその他の個体の適応度を予測する．適応度を基に GP オペレータが適用され，次世代個体を生成し，次のステップに移行する．ユーザによる評価は，各世代について必ず行うのではなく，事前にに定めた世代分だけユーザ評価の過程をスキップすることで，評価に係るユーザの負担を軽減する．ユーザ評価の過程をスキップする場合は，次世代集団の各個体について，直前のユーザ評価でユーザが選択した個体から適応度の予測を行う．予測した適応度に基づいて，次世代個体の生成を行う．

可視化に使用するデータについては，任意の表形式のデータを指定できるようにし，各列が属性に対応した表形式のデータの可視化に対応できるようにした．

本章の以降の節では，GTYPE の仕様，初期個体の生成手法，各個体の適応度の予測手法，個体集団の更新手法について説明を行う．

3.1. GTYPE の仕様

本システムで使用する GTYPE は，以下に示す複数の構造から構成される．

- combination 構造 … 可視化の組み合わせ方法を指定する
- mark 構造 … 可視化の描画方法を指定する
- encoding 構造 …可視化に使用するデータの変数と変数のプロット方法を指定する

GTYPE の構造の例を図 3.2 に示す．mark 構造と encoding 構造についての説明は後述する．

```
{
  'combination': {
    'hconcat': [
      {'vconcat': [
        'v0','v1'
      ],
      'layer': [
        'v2','v3'
      ],
    }]}},
  'mark': {
    'v0': [...],
    'v1': [...],
    'v2': [...],
    'v3': [...]
  },
  'encoding': {
    'v0': [...],
    'v1': [...],
    'v2': [...],
    'v3': [...]
  }
}
```

図 3.2 GTYPE の構造

combination 構造では，hconcat 構造と vconcat 構造，layer 構造を用いて，可視化の配置と組み合わせ方法を指定する．各構造内のリストには可視化の id が内包されている．hconcat 構造はリスト内の可視化を水平方向に並べて配置することを示している．図 3.2 の例では vconcat 構造の可視化と layer 構造の可視化が水平方向に並べられている．vconcat 構造はリスト内の可視化を垂直方向に並べて配置することを示している．図 3.2 の例では v0 と v1 の可視化が垂直方向に並べられている．layer 構造は，リスト内の可視化を重ね合わせて配置することを示している．図 3.2 の例では v2 と v3 の可視化が重ねられている．この場合の layer 構造の可視化の例を図 3.3 に示す．この例では， 都道府県と PCR 検査実施人数についての可視化（v2）と，都道府県と PCR 検査実施人数についての可視化（v3）が重ねて配置されている．

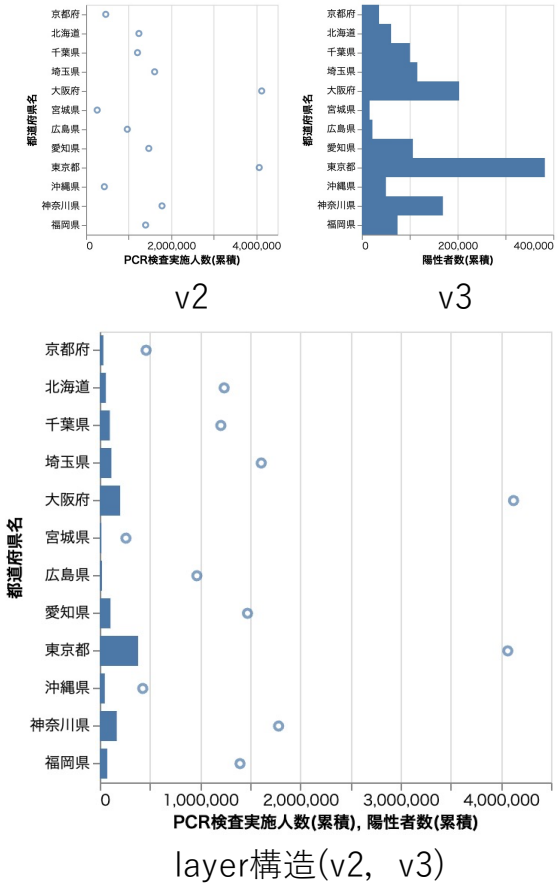


図 3.3 layer 構造の可視化の例

図 3.4 に mark 構造の例を示す.

```
'mark': {
  'v0' : {
    'type' : "point",
    'shape' : " triangle-up",
    'angle' : -209,
    'size' : 21,
    'strokeWidth' : 9.7
  }
}
```

図 3.4 mark 構造の例

図 3.4 の例では，可視化の種類を示す type， 散布図（point）で各データを表す点の形状を示す shape, 点(三角形)の角度を示す angle，点の大きさを示す size，点の枠線の大きさを示す strokewidth の項目が含まれている．この例では，可視化の種類は point（散布図）であり，プロットした点の形状が反時計回りに 201 度傾いた三角形(triangle-up)に設定されている．また，点のサイズが 21，点の枠線の太さが 21 にそれぞれ設定されている．可視化の種類ごとに設定できる項目は異なっており，本システムで設定で

きる項目については付録 A に示す.

図 3.5 に可視化の種類が text の場合の encoding 構造の例を示す.

```
'encoding': {
  'v1': {
    'x': {
      'field': "重傷者数",
      'type': "quantitative“
    },
    'y': {
       'field': "死亡者数(累積)",
       'type': "quantitative"
    },
    'text': {
       'field': "都道府県名",
       'type': "nominal“
    },
    'size': {
       'field': "陽性者数(累積)",
       'type': "quantitative"
    },
    'color': {
       'field': "都道府県名",
       'type': "nominal“
    },
    'opacity': {
       'field': "PCR 検査実施人数(累積)",
       'type': "quantitative"
    }
  }
}
```

図 3.5 encoding 構造の例

図 3.5 の例では，x，y，color，text，size，color，opacity といった構造が含まれている．これらの構造はデータ変数とプロット方法の関係を示している．各構造の役割は以下の通りである．

- x，y … 指定したデータ変数を x, y 軸に割り当てる．
- color … 指定したデータ変数の値に応じて，プロットしたオブジェクトの色分けを行う．
- opacity … 指定したデータ変数の値に応じて，プロットしたものの透明度を設定する．
- size … 指定したデータ変数の値に応じて，プロットしたものの大きさを調整する，

- text … 指定したデータ変数に格納されたテキストをオブジェクトとして描画する．

図 3.5 の例では，x 軸，y 軸に重症者数，死亡者数(累積)がそれぞれ割り当てられ，都道府県の名前（テキスト）が該当する位置に描画される．描画されたテキストのフォントサイズについては陽性者数(累積)の値の大小に応じて決定され，都道府県ごとに色分けされる．描画されたテキストの透明度については，PCR 検査実施人数(累積)の値の大小に応じて決定される．mark 構造内の type の値によって encoding で設定できる構造は異なり，本システムで設定できる項目は付録 A に示す．

3.2. 初期個体集団の作成

中川[14]によると，初期個体の多様性が最終的な結果に与える影響が大きいことが報告されている．そこで本論文では，多様な個体が含まれるように，以下の条件を満たすように初期個体集団を生成する．

- combination 構造 … layer，hconcat，vconcat のいずれの構造も持たない可視化が初期個体集団の 50%を占めるように生成し，残りについては，layer 構造の可視化と，hconcat 構造の可視化，vconcat 構造の可視化の個数が同等になるように生成を行う．異なる構造を組み合わせた個体の生成は行わない．
- mark 構造 … 提案システムで対応する全種類の可視化が，必ず一つは初期個体集団内に存在するように生成を行う．
- encoding 構造 … 各データ変数が含まれる可視化が，必ず一つは初期個体集団内に存在するように生成を行う．

これらの条件を満たすように生成することで，combination 構造については，シンプルな構造の可視化と複雑な構造の可視化の両方が生成される．mark 構造については，本システムで対応する area(面グラフ)，bar(棒グラフ)，line(線グラフ)など 10 種類の可視化が，必ず含まれるように生成される．encoding 構造については，各データ変数を使用した可視化が，初期個体集団内に必ず含まれるように生成される．これによって初期個体集団が偏らないことが期待できる．

3.3. 適応度の予測

個体集団内の個体iの適応度$f_i = (f_{i_{com}}, f_{i_{mark}}, f_{i_{enc}})$の予測は，以下の手順で行う．

1. ユーザが可視化の組み合わせ，可視化の種類，可視化に含まれるデータの各観点について，意図を満たすとして選んだ個体の集合をそれぞれ COM, MARK, ENC の 3 つの集合とする．
2. 個体iの GTYPE の combination 構造と COM 内の各個体の combination 構造の類似度を全て求め，類似度の最大値を$f_{i_{com}}$とする．
   mark 構造と encoding 構造についても同様に最大類似度を求め，求めた類似度の中で最も高いものをそれぞれ$f_{i_{mark}}$，$f_{i_{enc}}$とする．

COM は可視化の組み合わせについてユーザが意図を満たしたと評価した個体の集合であり，MARK は可視化の種類についてユーザが意図を満たしたと評価した個体，ENC は可視化に含まれるデータ変数についてユーザが意図を満たしたと評価した個体である．

例えば，$COM = \{a, b, c, d, e\}, MARK = \{a, b, d\}$，$ENC = \{c, e\}$の時，個体集団内の個体$i$の適応度$f_{i_{com}}$，$f_{i_{mark}}$，$f_{i_{enc}}$は，次のように求められる．ここで，個体 x,y 間の z 構造に関する類似度を$sim_z(x, y)$とする．

$$f_{i_{com}} = \max\left(sim_{com}(i, a), sim_{com}(i, b), sim_{com}(i, c), sim_{com}(i, d), sim_{com}(i, e)\right) \quad (3.1)$$

$$f_{i_{mark}} = \max\left(sim_{mark}(i, a), sim_{mark}(i, b), sim_{mark}(i, d)\right) \quad (3.2)$$

$$f_{i_{enc}} = \max\left(sim_{enc}(i, c), sim_{enc}(i, e)\right) \quad (3.3)$$

3.3.1. 個体間の類似度と部分構造

個体の構造の類似度については，部分構造の集合を用いて求める．部分構造の集合は，対象となる構造内のリスト構造や辞書構造，可視化 id を抽出し，抽出した構造の中にもリスト構造や辞書構造，可視化 id があれば，それも抽出することで求める． mark 構造の部分構造の集合の例をそれぞれ図 3.6 に示す．他の構造も同様に求める．

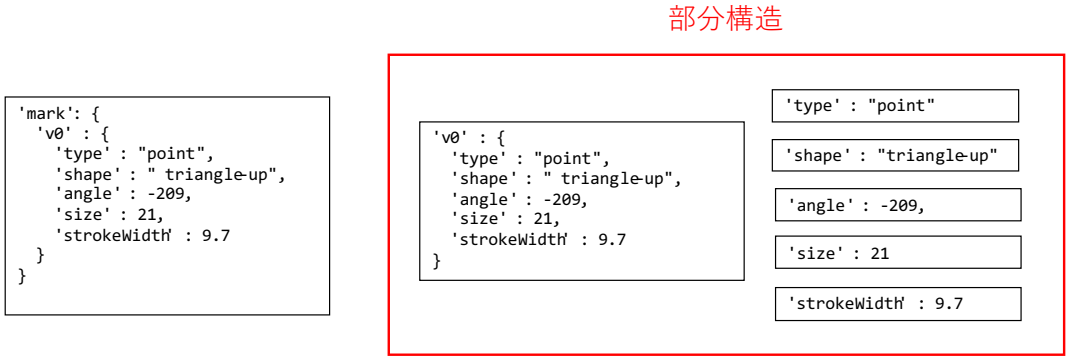


図 3.6 mark 構造の部分構造の例

$sim_z(x, y)$は以下のように求める．ここで$s_z(x)$はそれぞれ個体xの GTYPE に含まれる，観点 z に関連した部分構造の集合である．

$$sim_{z(x,y)} = \frac{|s_z(x) \cap s_z(y)|}{\max\left(|s_z(x)|, |s_z(y)|\right)} \cdots (3.4)$$

3.4. 個体集団の更新

個体集団の更新については，適応度に基づいて選択した個体に，GP オペレータを適用して行う．適用する GP オペレータは，交叉と突然変異の 2 種類である．後述するエリート個体以外は，適応度を基にルーレット選択で次世代個体の元となる個体を選んで GP オペレータを適用し，次世代個体の生成を行う．

3.4.1. エリート個体の選択

提案手法では，ユーザ評価と適応度に基づくエリート保存戦略を採用する．以下の 4 つの個体をエリート個体として保存を行う．ユーザ評価が選択した個体も保存し，次回のユーザ評価時に個体集団に戻して提示を行う．

- 3 つの観点の適応度の合計が最も高い個体
- COM の中で，他の観点の適応度の合計が最も高い個体
- MARK の中で，他の観点の適応度の合計が最も高い個体
- ENC の中で，他の観点の適応度の合計が最も高い個体

保存したエリート個体は，一切の改変を加えずに次世代の個体集団に加えられる．ユーザ評価を行う場合は，ユーザ評価適応度を基にルーレット選択を行い，集団からエリート個体の追加分だけ淘汰を行った後に，前回のユーザ評価時に選ばれたエリート個体を追加する．

3.4.2. 交叉

交叉は，ルーレット選択で選択した個体から 2 個体選び(親個体 1，親個体 2)，親個体 1 のコードの一部を，親個体 2 のコードの一部と差し替え，1 個体を生成する．combination 構造については，combination 構造全体や，構造内の layer 構造，hconcat 構造，vconcat 構造のいずれかを選択し交叉を行う．図 3.7 に combination 構造に対する交叉の例を示す．

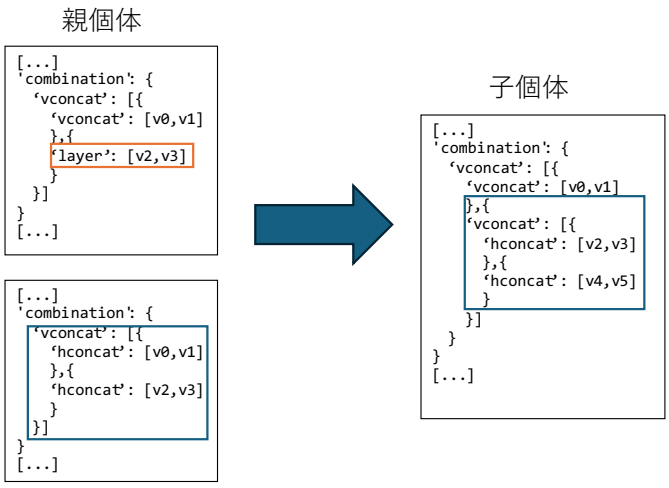


図 3.7 combination 構造の交叉の例

交叉時に，構造内に含まれている可視化 id は番号順に変更される．交叉前と比較して combination 構造内の可視化 id の数が増えた場合は，親個体 2 に含まれる可視化 id をランダムで増加分選び，その可視化 id に関する mark 構造と encoding 構造を，それぞれ子個体に追加する．図 3.8 に mark 構造と encoding 構造の追加の例を示す．

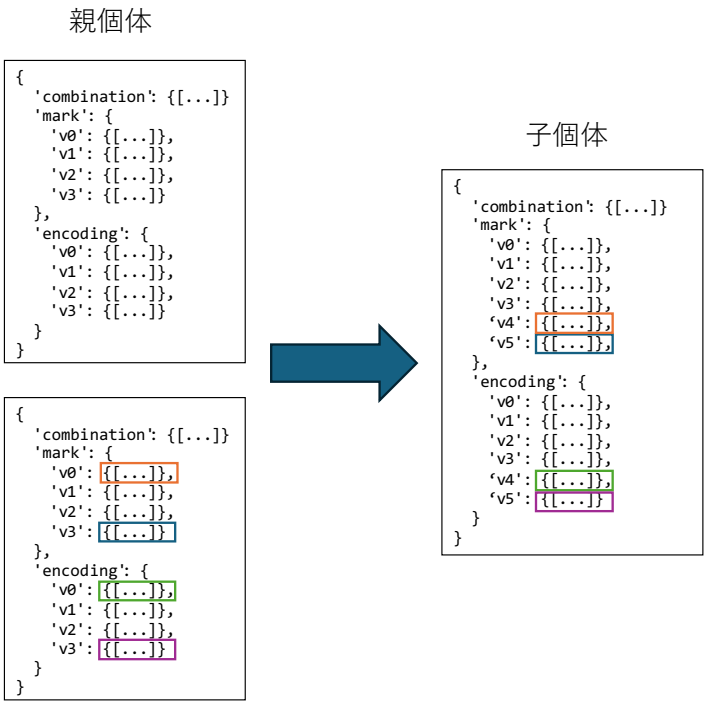


図 3.8 mark 構造と encoding 構造の追加の例

図 3.8 の例では，上の親個体は 4 個の id を含んでいたのに対し，子個体では 6 個に増えている．そのため，下の親個体の v0 ~ v3 の範囲から v0 と v3 が選択され，子個体の v4，v5 としてそれぞれ mark，encoding 構造に挿入されている．

mark 構造については，type の項目が一致したときは，構造内のいずれかの項目を選択し交叉を行う．図 3.9 に mark 構造の交叉の例を示す．

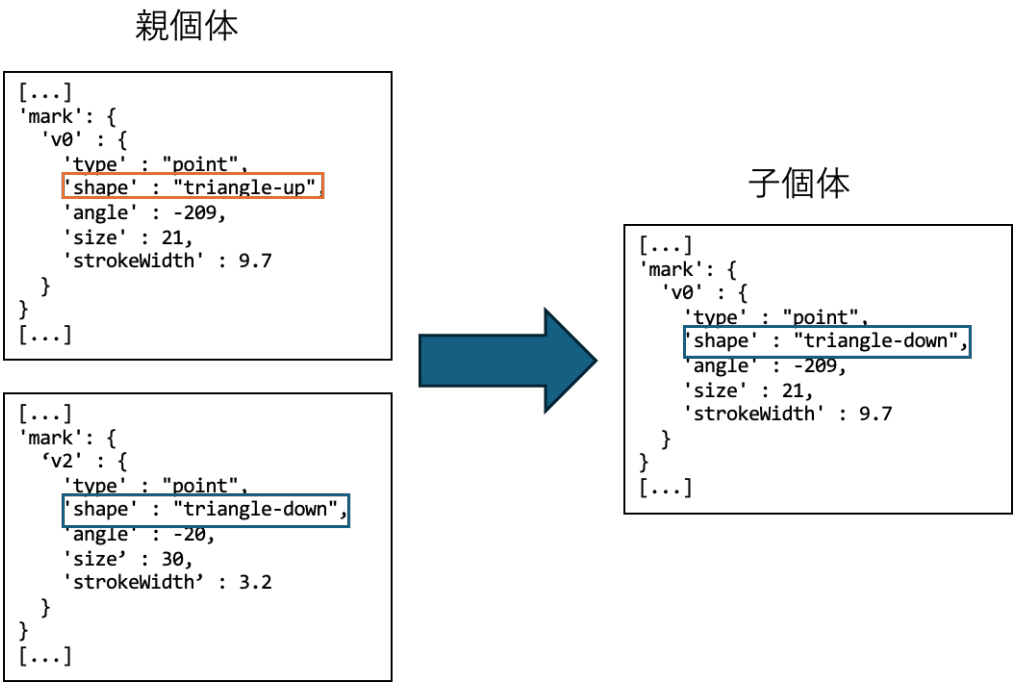


図 3.9 mark 構造の交叉の例

親個体の type の項目が一致しない場合は適用可能な項目が異なるため，上記と同様の交叉では描画可能な個体が生成される可能性が低くなる．そこで，mark 構造全体を差し替え，交叉を行う．

encoding 構造については，親個体 1 の個体の encoding 構造内の項目を，親個体 2 の個体のいずれかの項目で差し替えて交叉を行う．交叉を行った後に，必須条件を満たさない子個体が生成された場合は，必須条件を満たす子個体ができるまで，親個体 1 と親個体 2 の間で交叉を繰り返す．図 3.10 に encoding 構造の交差の例を示す．

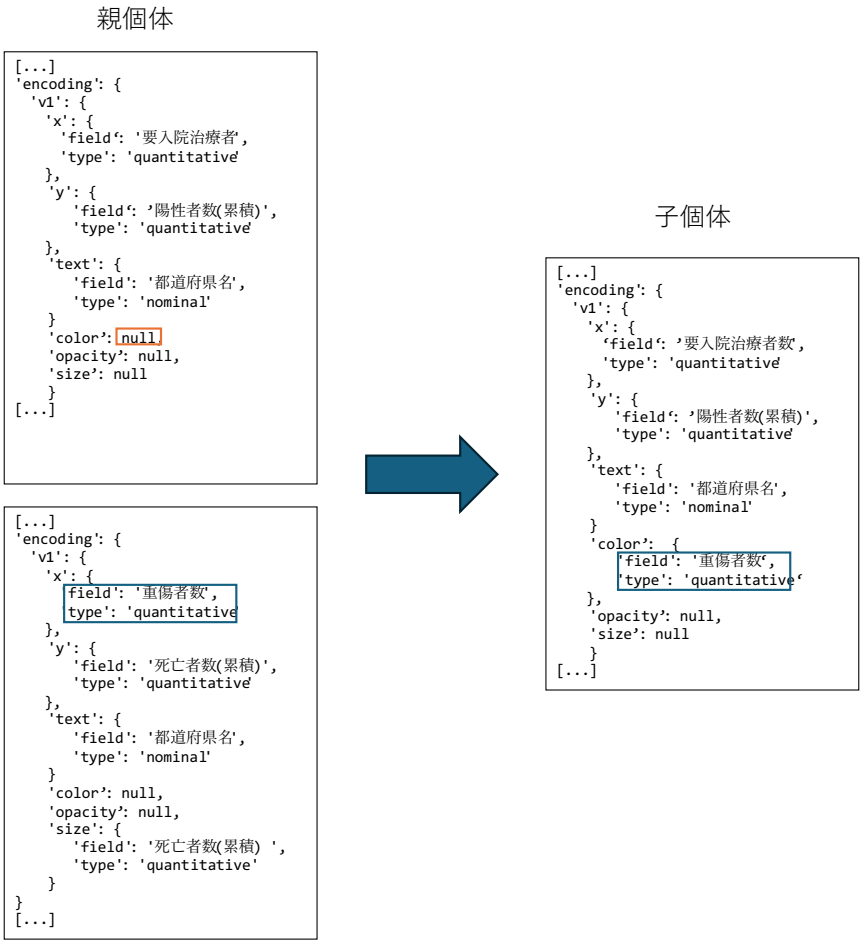


図 3.10 encoding 構造の交叉の例

図 3.10 の例では，親個体 1 の v1 の color 内の構造と，親個体 2 の v1 の x 内の構造が差し替えられており，描画可能である．

交叉の適用箇所については，親個体 1 の個体の適応度に基づいて，以下の順序で決定する．

1. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の和が，個体集団の適応度の和の平均を下回った場合は combination，mark，encoding 構造内の項目に適用する（三点交叉）．
2. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，$f_{i_{com}}$が最小値の場合は，combination 構造内の項目に適用する（一点交叉）．
3. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，$f_{i_{mark}}$が最小値の場合は，mark 構造内の項目に適用する（一点交叉）．
4. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，$f_{i_{enc}}$が最小値の場合は，encoding 構造内の項目に適用する（一点交叉）．
5. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，最小値が複数ある場合は，その中からランダムに選択し適用する（一点交叉）．

3.4.3 突然変異

突然変異は，ルーレット選択で選択した個体集団から 1 つ個体を選び，コードの一部をランダムに変更する．

combination 構造については，layer 構造や，vconcat 構造，hconcat 構造を別の構造に置き換える．

図 3.11 に combination 構造の突然変異の例を示す.

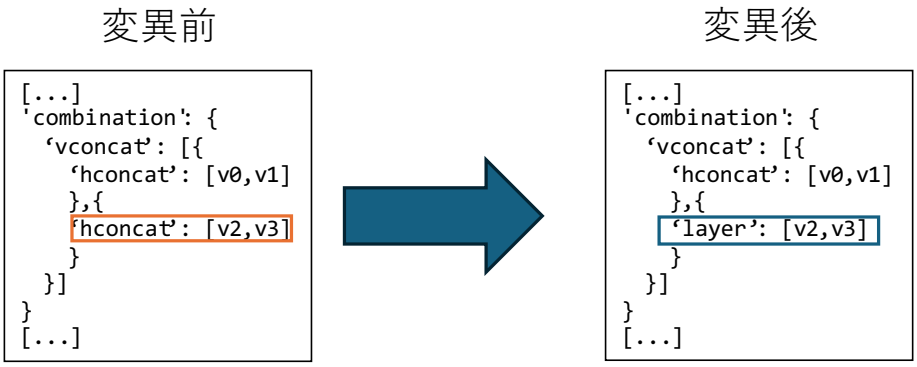


図 3.11 combination 構造の突然変異の例

mark 構造については，構造内のデータ変数の項目の値を別の値に変更する．図 3.12 に mark 構造の突然変異の例を示す.

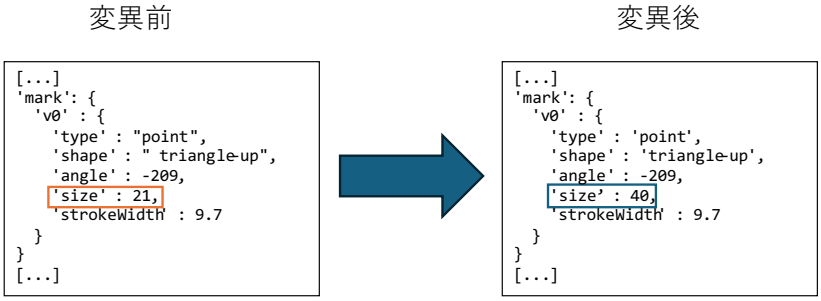


図 3.12 mark 構造の突然変異の例

mark 構造内の type の項目が変異する場合は，mark 内の設定可能な項目が変わるので，type 以外の項目も，変異に併せてランダムに設定する．図 3.13 に mark 構造の type が突然変異した例を示す.

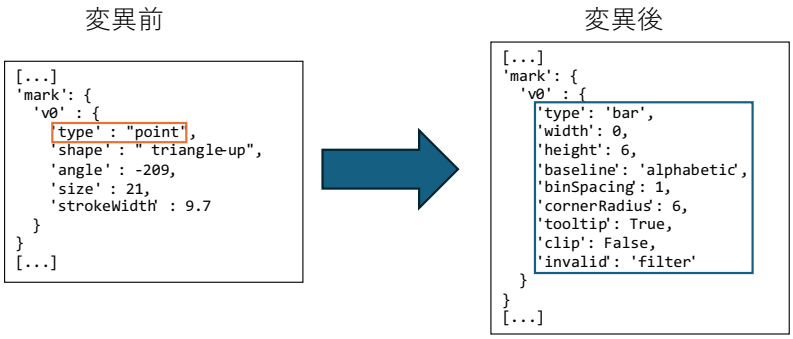


図 3.13 項目 type の突然変異の例

図 3.13 の例では，point と mark で共通して設定可能な項目が type 以外にないので，type 以外の項目を削除し，mark で設定可能な項目を挿入し，その値をランダムに設定している.

encoding 構造の突然変異は，encoding 構造内のデータ変数の項目を．他のデータ変数の項目で置き換える．図 3.14 に encoding 構造の突然変異の例を示す.

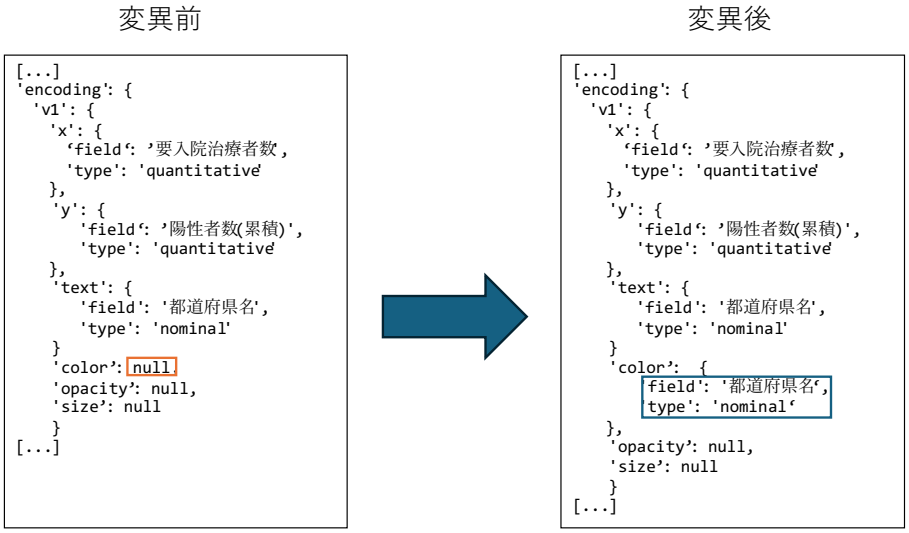


図 3.14 encoding 構造の突然変異の例

突然変異の適用箇所については，個体の適応度に基づいて，以下の順序で決定する．

1. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，$f_{i_{com}}$が最小値の場合は，combination 構造内の項目に適用する ．
2. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，$f_{i_{mark}}$が最小値の場合は，mark 構造内の項目に適用する．
3. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，$f_{i_{enc}}$が最小値の場合は，encoding 構造内の項目に適用する．
4. $f_{i_{com}}$, $f_{i_{mark}}$, $f_{i_{enc}}$の中で，最小値が複数ある場合は，その中からランダムに選択し適用する．

3.5 対話型可視化生成システム

提案手法のプロトタイプシステムを，Web アプリケーションフレームワーク Django を用いて実装した．図 3.15 に生成された可視化の評価画面，図 3.16 に可視化の選択画面をそれぞれ示す．生成された画像は縦一列に並んでいて，スクロールして閲覧可能である．

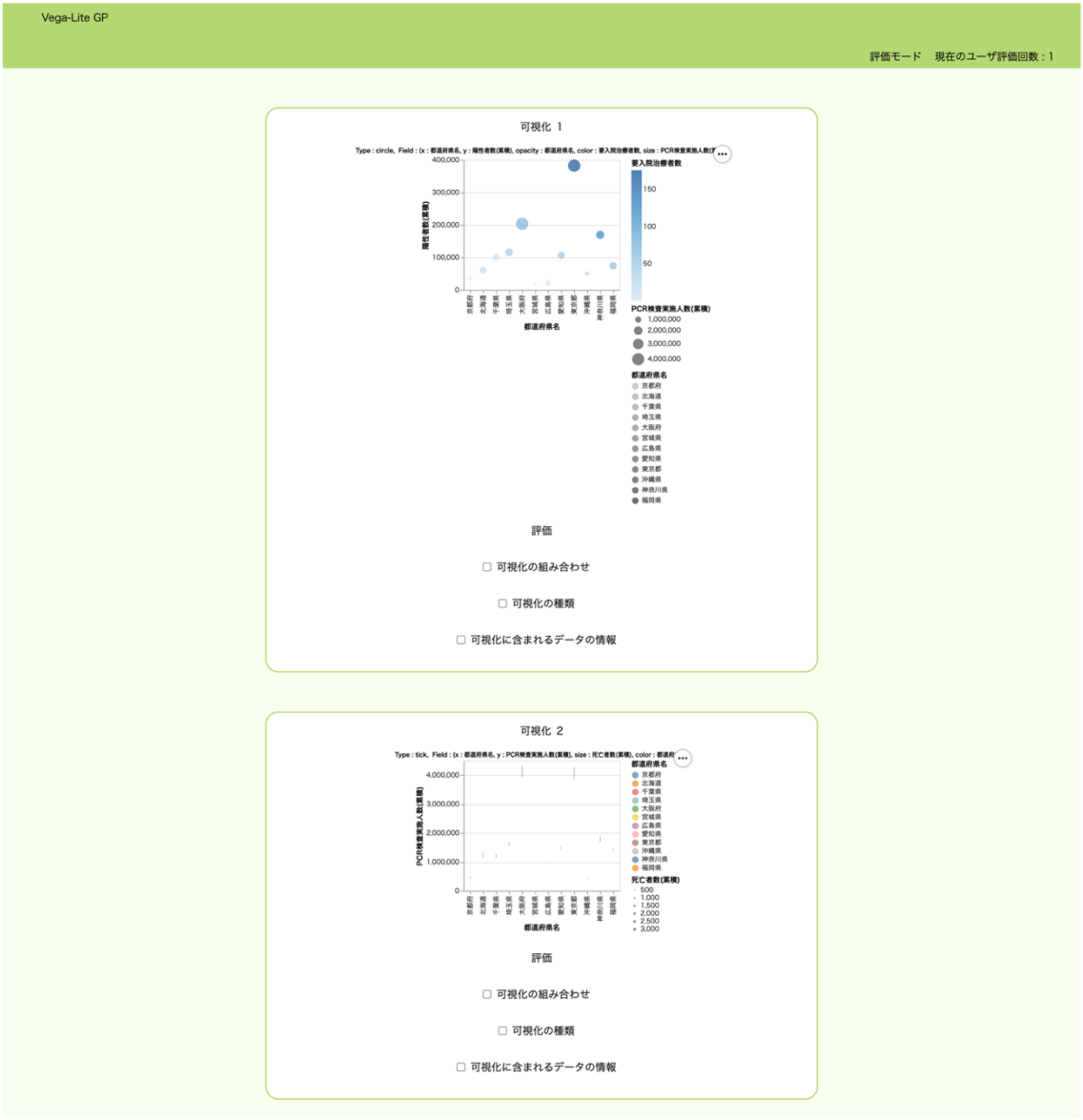


図 3.15 可視化の評価画面

図 3.16 可視化の選択画面

評価画面で可視化の評価を行い，納得のできる可視化が生成されれば，選択画面で最終結果とする可視化を選んで終了する．評価画面では，各生成画像の下に可視化の組み合わせ，可視化の種類，可視化に含まれるデータの種類それぞれの観点についての評価値を入力するためのチェックボックスが配置されている．システム利用者は，各可視化について，それが各観点について自身の意図を満たす場合，該当する観点をチェックボックスで選択する．ユーザ評価の回数が事前に定めた回数に達すると，選択画面に遷移可能となる．選択画面では，ラジオボタンで最終結果とする可視化を一つ選択する．

Vega-Lite で描画する可視化は，図 3.17 のように，データ変数と可視化の各要素との対応が凡例だけではわからない場合があるため，可視化の上部に可視化の種類，可視化に含まれるデータの情報を表示するようにしている．

図 3.17 の例では，x 軸，y 軸，テキスト（数値）とそのサイズが，それぞれ都道府県名，要入院治療者数，陽性者数(累積)，死亡者数(累積)と対応していることを示している．可視化の軸と判例だけでも，x，y，size の対応関係は確認できるが，text の対応関係は確認できない．

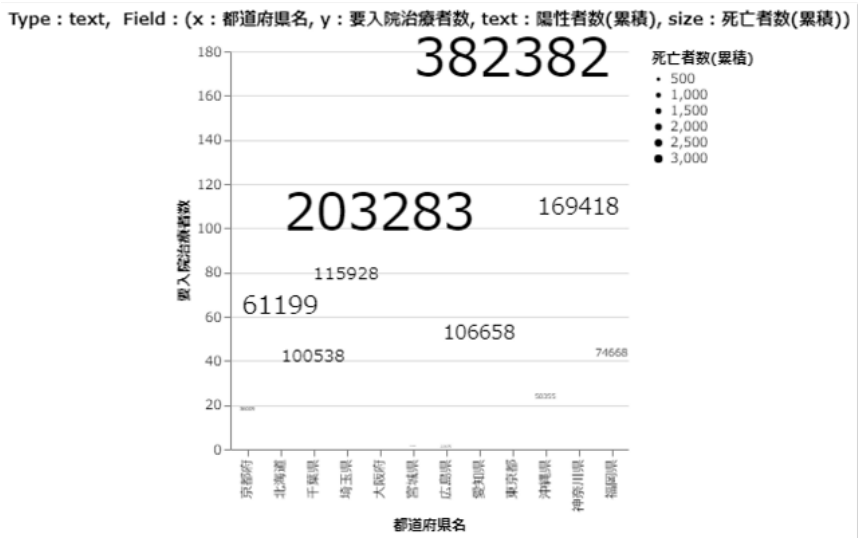


図 3.17 データ変数と可視化の各要素との対応の説明

4. 実験

4.1. 評価実験

ユーザ評価をスキップし，適応度予測に基づき個体集団を進化させても意図する可視化が得られるかを検証するために，評価を与えた初期個体集団に GP オペレータを適用し，その後はユーザ評価をせずに適応度予測に基づき，2~6 世代目の個体を生成する操作を 100 回行った．

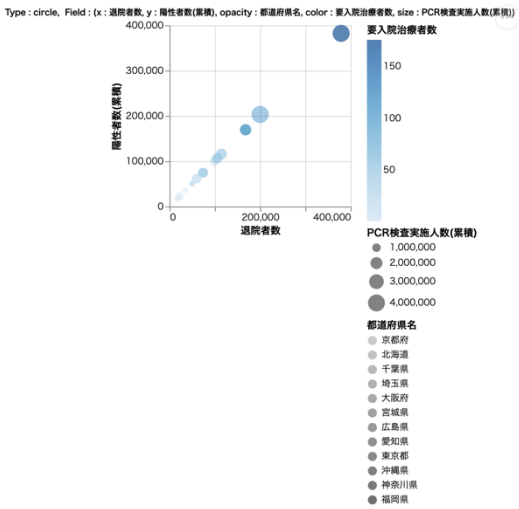
この実験は，ユーザが可視化の見た目(PTYPE)で判断し，取捨選択している行動を，Vega-Lite のコード(GTYPE)レベルで模倣できることを仮定して行った．

世代あたりの個体数は 20，交叉率は 70%，突然変異率は 20%とし，乱数のシード値は 1~100 の一様分布から試行ごとに設定して行った．初期個体集団については，3.2 節で述べた初期個体の条件を満たすように生成した．図 4.1，図 4.2 に初期個体となる可視化の図とそれぞれの適応度を示す．図 4.1 に示した個体は，layer 構造，hconcat 構造，vconcat 構造のいずれも持たない単一の可視化であり，図 4.2 に示した個体は layer 構造や hconcat 構造，vconcat 構造のいずれかを持つ可視化である．

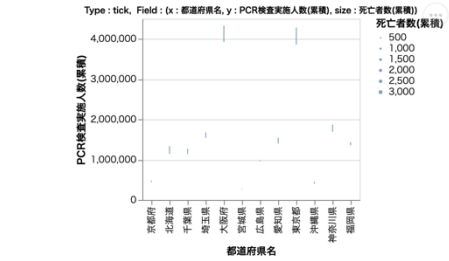
初期個体に対するユーザ評価は，主観に左右されず客観的に判断するために，Vega-Lite のコード（GTYPE）に基づき観点ごとに以下のように評価を行った．

- 可視化の構造
  - GTYPE の combination 構造内に layer 構造が含まれている個体を評価
- 可視化の種類
  - GTYPE の mark 構造内で項目 type が bar の個体を評価
- 可視化に含まれるデータ
  - GTYPE の encoding 構造内に要入院治療者数と退院者数の両方が含まれている個体を評価

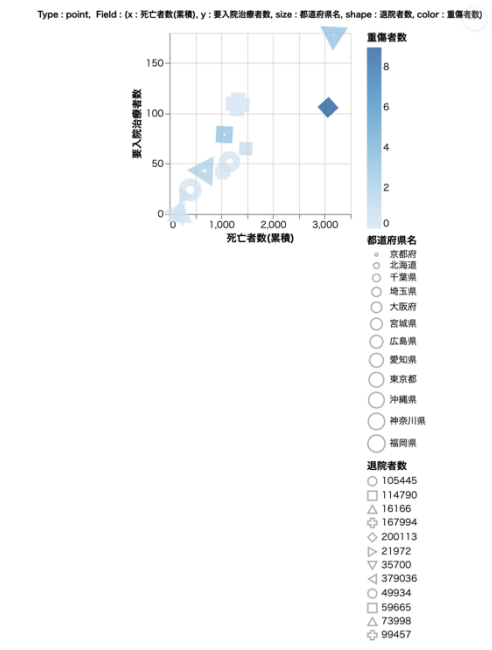
表 4.3 に，各世代集団内の可視化の構造ごとの平均個体数，表 4.4 に，各世代集団内の可視化の種類ごとの平均個体数，表 4.5 に各世代集団内の可視化に含まれるデータ変数ごとの平均個体数を示す．複数の可視化を複合した個体の場合は，それぞれについてカウントしている．layer 構造，vconcat 構造，layer 構造のいずれも持たない個体については，構造なしとしてカウントを行った．それぞれの表の上部には，初期世代集団の個体数も示している．



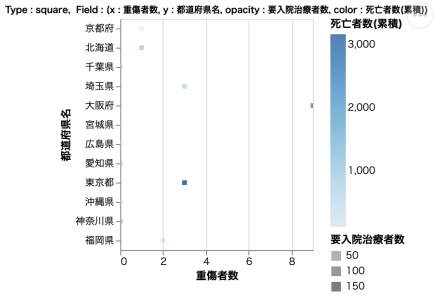
$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.15, 1.0)$$



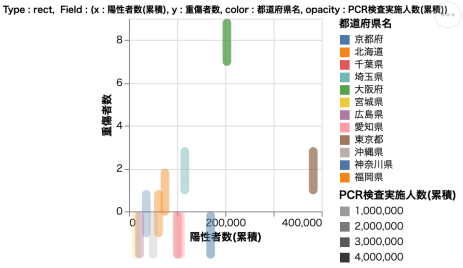
$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.1, 0.6154)$$



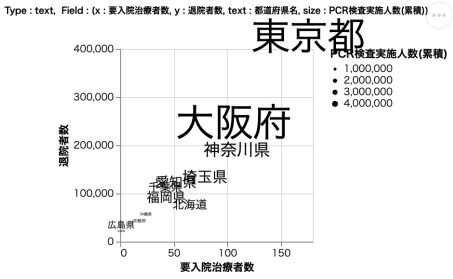
$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.0, 1.0)$$



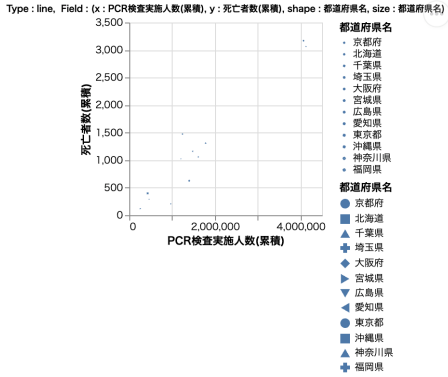
$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.2, 0.5417)$$



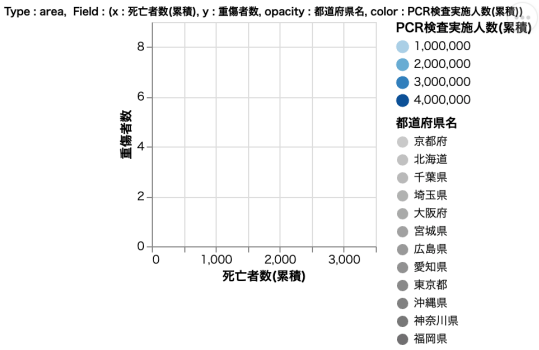
$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.2, 0.4706)$$



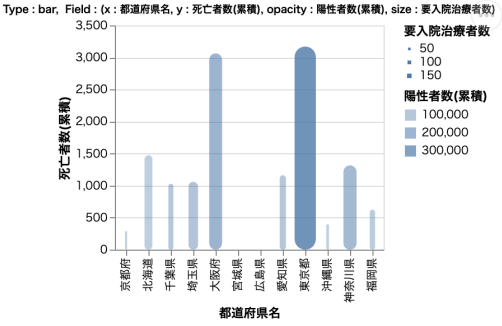
$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.2, 1.0)$$



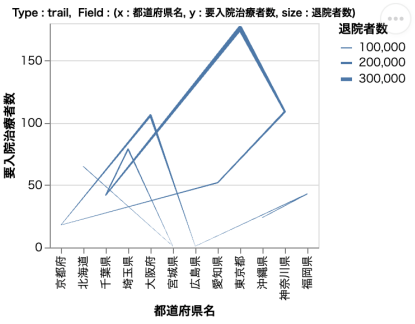
$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.1, 0.5625)$$



$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.1, 0.5)$$



$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 1.0, 0.6)$$



$$(f_{com}, f_{mark}, f_{enc}) = (0.2, 0.2, 1.0)$$

図 4.1 初期個体とその評価(構造なし)

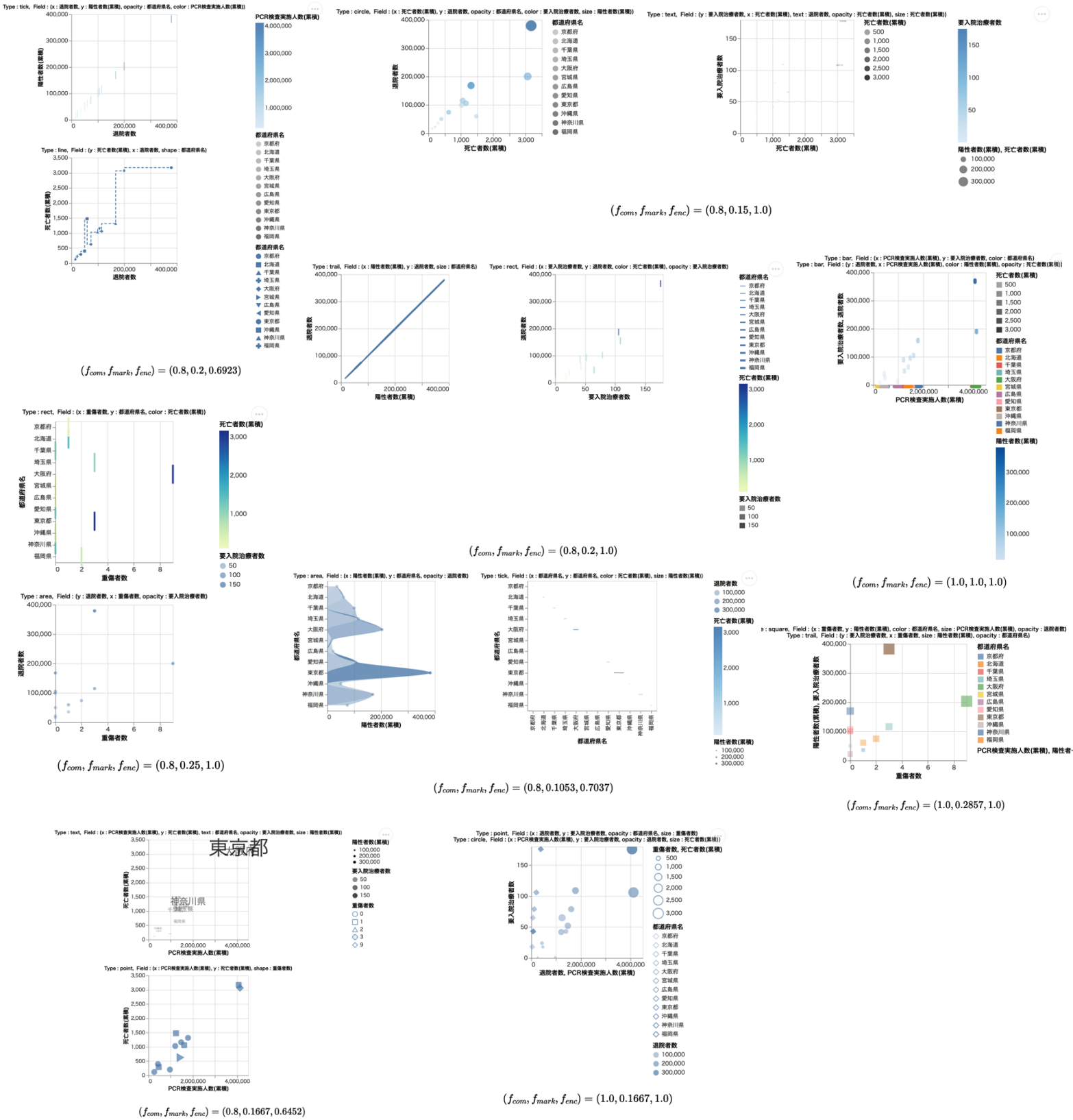


図 4.2 初期個体とその評価(構造あり)

表 4.3 可視化の構造ごとの個体数

	layer	vconcat	hconcat	構造なし
初期世代	3	4	3	10
2 世代	4.59	5.5	4.47	6.52
3 世代	5.76	5.97	4.61	4.62
4 世代	6.94	6.15	4.99	2.96
5 世代	8.07	5.93	5.17	1.91
6 世代	9.68	5.68	5.05	1.09

表 4.4 可視化の種類ごとの個体数

	area	bar	circle	line	point	rect	square	text	tick	trail
初期世代	3	2	3	3	3	3	3	3	3	3
2 世代	2.82	3.65	3.02	3.17	2.86	2.94	2.92	2.99	2.91	3.22
3 世代	2.87	5.21	2.89	3.12	2.74	2.87	3.02	2.72	2.64	3.15
4 世代	2.8	6.79	2.66	3.1	2.75	2.7	2.99	2.62	2.41	2.91
5 世代	2.76	8.07	2.44	2.86	2.54	2.51	2.77	2.42	2.35	2.8
6 世代	2.66	9.64	2.1	2.49	2.34	2.21	2.37	2.18	2.15	2.57

表 4.5 データ変数ごとの個体数

	陽性者数(累積)	要入院治療者数	退院者数	死亡者数(累積)	重傷者数	PCR 検査実施人数(累積)
初期世代	11	14	13	15	9	12
2 世代	12.49	15.45	14.79	15.86	8.26	11.19
3 世代	13.37	16.31	15.45	16.67	8.32	11.1
4 世代	14.5	16.89	16.27	17.54	7.81	10.98
5 世代	15.31	17.13	16.55	17.82	7.26	11.34
6 世代	16.14	17.44	16.89	18.04	6.68	12.12

表 4.2 より，layer 構造の可視化の個体数が 2 世代から 6 世代にかけて増加しており，可視化の構造についてはユーザの評価を反映していると考える．layer 構造，hconcat 構造，vconcat 構造のいずれも持たない個体数の平均が急激に減少しているのは，それらの個体に含まれる部分構造が少なく，他の個体と一致する部分構造の個数も必然的に少なくなるため，適応度が他の構造を持つ個体と比較して低くなることが原因だと考えられる．

表 4.3 より，bar 構造の可視化の個体数が 2 世代から 6 世代にかけて増加しており，可視化の構造についてはユーザの評価を反映していると考える．bar を含む個体の数が急激に増えているのは，combination 構造の交叉時に，可視化 id が増えた場合は，親個体 2 の一部の mark 構造と encoding 構造が追加されるため，世代が進むにつれて mark 構造の type が bar の個体が増えていったことが原因だと考えられる．

表 4.4 より，要入院治療者数，退院者数が含まれている可視化の個体数が 2 世代から 6 世代にかけて増加しているが，死亡者数，陽性者数も同様に増加している．初期個体集団内には，死亡者数(累積)を含む個体が 20 個中 15 個，陽性者数を含む個体が 20 個中 11 個存在しており，評価した可視化はそれらのデータ変数も含んでいたと考える．これらのデータ変数についても他の個体と一致する部分構造を持つ個数が多くなった結果，適応度が通常よりも高くなったと考える．このように可視化に含まれるデータについては，意図したデータ変数以外を持つ個体も増える場合があると考える．

4.2. ユーザ実験

工学系の大学生・大学院生 10 人に対して，提案システムを用いて，各都道府県の COVID-19 の感染状況を示す可視化を作成してもらう実験を行った．実験協力者には，遺伝的プログラミング，Vega-Lite，描画対象のデータの簡単な解説と、システムの使い方を予め説明し，簡単なチュートリアルでシステムの使い方に慣れてもらった後に実験に取り組んでもらった．実験協力者を 2 グループに分けて以下の条件のもとで行った．

- 乱数のシード値を固定
- 世代あたりの個体数は 20 に設定
- 世代のスキップ数を A グループは 2, B グループは 4 にそれぞれ設定

図 4.3，図 4.4 に初期個体として使用した可視化の図を示す．図 4.3 に示した個体は，layer 構造，hconcat 構造，vconcat 構造のいずれも持たない単一の可視化であり，図 4.4 に示した個体は layer 構造や hconcat 構造，vconcat 構造のいずれかを持つ可視化である．

ユーザの評価回数が 10 回に到達した段階で，可視化の選択画面に移れるようにし，30 回に到達した段階で，強制的に可視化の選択画面に遷移するようにした．

実験協力者には，以下の条件を満たす可視化を作成してもらった．

- 必須条件
  - 複数の可視化を並べた可視化であること．
  - 「都道府県名」と「陽性者数（累積)」が、可視化で使用されているデータに含まれていること．
  - 「現在の重症者数」,「累積の死亡者数」,「要入院治療者数」のいずれかが，可視化で使用されているデータに含まれていること．
- 可能条件
  - 都道府県間の感染状況の比較が行いやすい可視化であること．
  - 意味が読み取りやすい可視化であること．

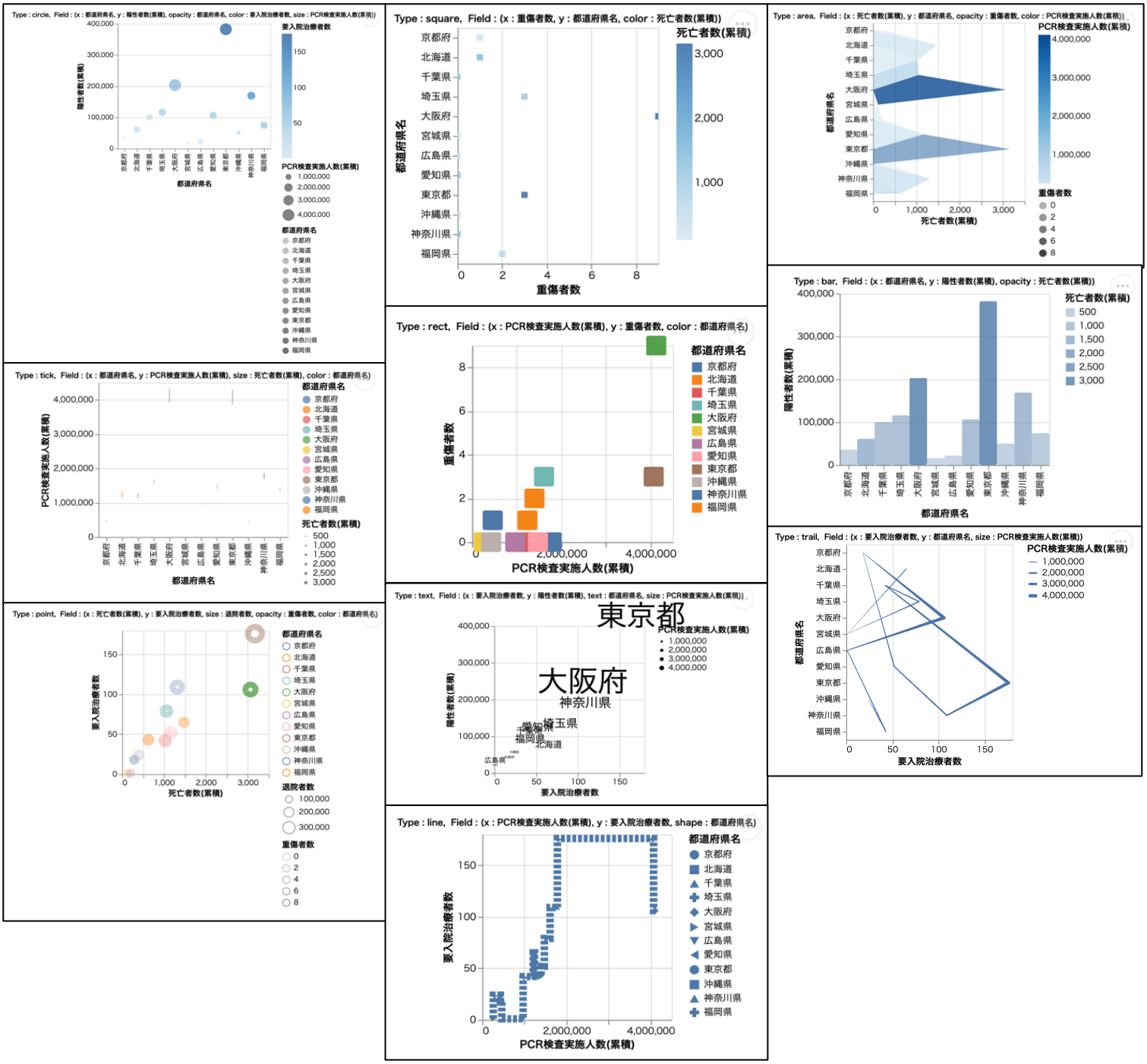


図 4.3 ユーザ実験の初期個体(構造なし)

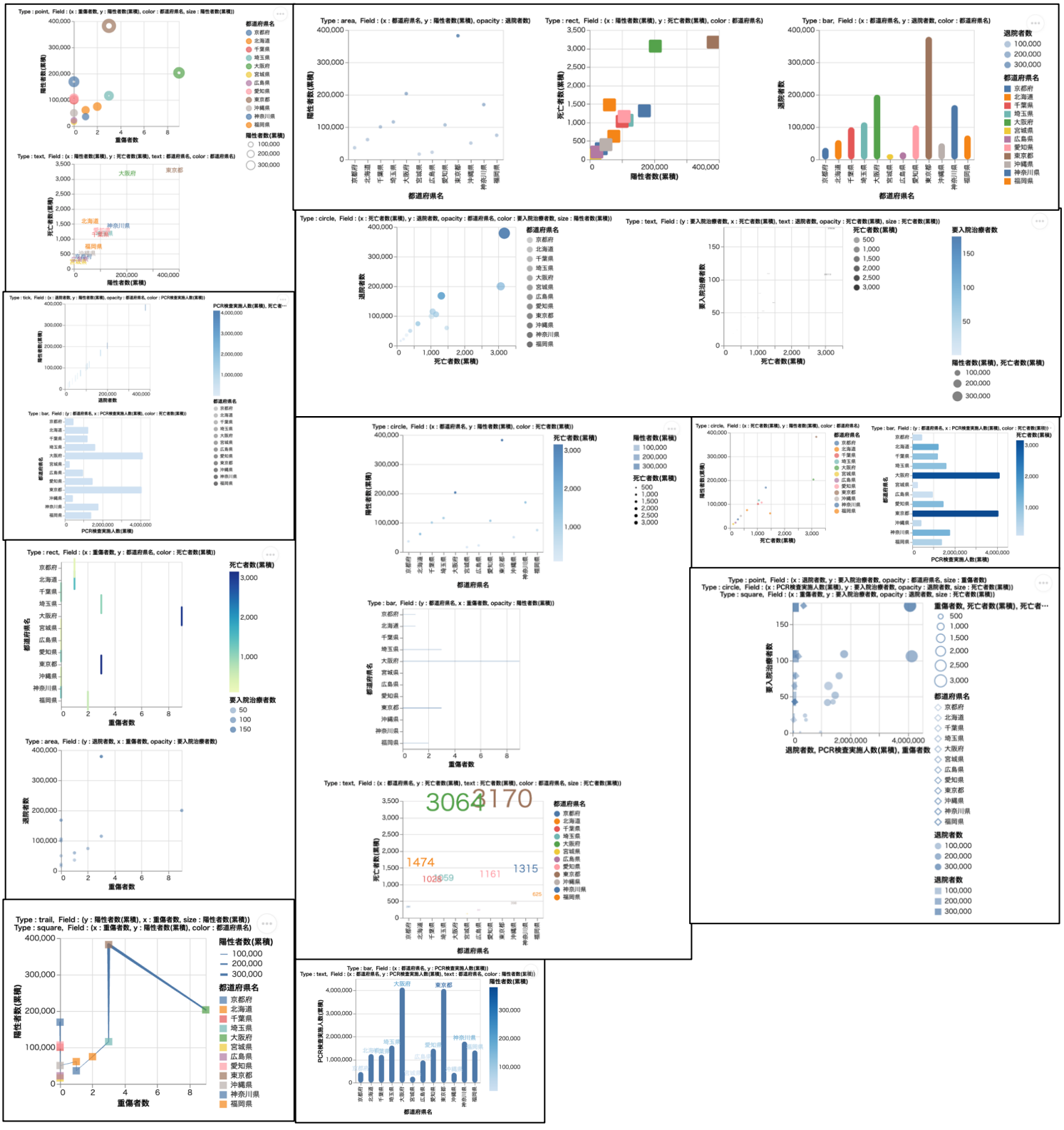


図 4.4 ユーザ実験の初期個体(構造あり)

以下に実験後にとったアンケートを示す.

●最後に選んだ可視化は必須条件を満たしていますか？
- はい
- いいえ

●最後に選んだ可視化の満足度について
- 1~5 の 5 段階で回答

●最後に選んだ可視化は，実験開始前に想定していたものとどの程度近かったですか？
- 1~5 の 5 段階で回答

●最後に選んだ可視化から読み取れることをすべて回答してください
- 自由記述

●最後に選んだ可視化の情報の読み取りやすさについて
- 1~5 の 5 段階で回答

●次世代個体の生成時にどの程度評価が反映されていると感じましたか？
- 1~5 の 5 段階で回答

●どのタイミングで一番ユーザ評価が反映されたと感じましたか？
(評価の反映の度合いの質問で 4,5 を選んだ人のみ)
- 実験の前半
- 実験の後半

●どのような点に着目して評価していましたか？
- 自由記述

●最後の可視化の選択で最も重視した点はどれですか？
- 可視化の構造
- 可視化の種類
- 可視化に含まれるデータ
- その他(自由回答)

表 4.5，表 4.7 に，A グループと B グループの最後に選んだ可視化から読み取れることと，どのような点に着目して評価したかについてそれぞれの回答を示す．その他の選択肢による回答をグループごとに表 4.6, 4.8 にそれぞれ示す.

表 4.5　A グループのアンケートの結果（読み取れること，評価の着目点）

ユーザ ID	最後に選んだ可視化から読み取れること	どのような点に着目して評価したか
A01	左の可視化から，PCR 検査実施人数が多いにもかかわらず, 陽性者数が少ない地域があることがわかり，右の可視化から，その地域が大阪府であることがわかる.	陽性者数と都道府県の情報が含まれている可視化を評価するようにした.
A02	都道府県ごとの陽性者数 陽性者数と死亡者数の相関 陽性者数と重症者数の相関	地域間で比較しやすい棒グラフと相関が読み取りやすい散布図の組み合わせを選ぶようにした.
A03	左図の点から東京都における陽性者数が多いく、京都府や宮城県は少ない。右図から陽性者と死亡者数は正の相関がある。	都道府県別陽性者数が一目でわかるようなグラフを選んだ。陽性者数別の死亡者数のデータが含まれる可視化を評価した。都道府県別陽性者数がヒストグラムの図になるよう 3,4 度試したがプロットの図が出力されてしまった。
A04	死者数(累積)が多い都道府県は陽性者数(累積）と PCR 検査実施人数（累積）が多いことが読み取れる。	複数の可視化を含む場合は、過半数が必須条件を満たしていることに着目して評価していました。
A05	都道府県別で陽性者数と退院数は比例することが読み取れる 陽性者数と死亡者数の間には相関関係があることが読み取れる	都道府県が x 軸になるように評価した

表 4.6　A グループのアンケートの結果（選択肢による回答）

ユーザ ID	必須条件を満たしたか	可視化の満足度	想定していたものとどの程度近かったか	可視化の読み取りやすさ	評価の反映の度合い	一番評価が反映されたタイミング	選択で最も重視した点
A01	はい	3	2	5	3		可視化に含まれるデータ
A02	はい	4	4	4	5	実験の前半	可視化の種類
A03	はい	2	2	3	4	実験の後半	可視化の種類
A04	はい	4	3	4	4	実験の前半	可視化に含まれるデータ
A05	はい	5	4	5	4	実験の後半	可視化の種類

表 4.7　B グループのアンケートの結果（読み取れること，評価の着目点）

ユーザ ID	最後に選んだ可視化から読み取れること	どのような点に着目して評価したか
B01	東京、大阪の PCR 検査実施人数（累積）が多い。 東京、大阪の重症者数が多い。 東京、大阪の陽性者数（累積）が多い。	以下の二点を重視して評価した。 条件を満たす ぱっと見で情報を得やすい図が含まれる
B02	左の可視化は東京都の感染者数が多いことが読み取れる 中の可視化は東京都の感染者と死亡者数が多いことが読み取れると大阪府の死亡率は高いことを読み取れる 右の可視化は東京都の退院者が多いことが読み取れる	地域間で比較しやすいような可視化の種類を選ぶようにした 死亡者と退院者のデータが含まれる可視化を評価するようにした
B03	左図から，各都道府県の累積陽性者数と累積死亡者数に相関関係があることが読み取れる． 各都道府県のポイントの分布が右上がりの直線に近似できることから，強い正の相関があると読み取れる． 右図からは，同じく各都道府県について，累積死亡者数と要入院治療者数の相関関係が読み取れる．基本的には要入院治療者数が多い都道府県ほどヒストグラムの色が濃くなっているので，こちらも強い正の相関があると読み取れる．	・どちらの可視化図にも都道府県が含まれていること ・死亡者数．陽性者数の相関がわかりやすい可視化図であること を意識して評価した．
B04	大阪と東京は陽性者数と死亡者数が多い、陽性者数と死亡者数には相関がある	陽性者数が含まれるものを選んだ、可視化が横に 2 枚表示されているものが見やすいので評価した
B05	「右の図からは、東京都と大阪府の感染者数が最も多く、死亡者数も最も高いことが直感的にわかります。一方、宮城県と広島県の感染者数は最も少なく、死亡者数も低い状況です。」「右の図からは、北海道の感染者数はそれほど多くないものの、死亡率が比較的高いことが分かります。一方で、神奈川県は感染者数が多いものの、死亡率は比較的低いことが読み取れます。」「左の図からは、広島県の死亡者数は比較的少ないが、PCR 検査の実施数は多いことが分かります。一方、北海道では死亡者数が多いにも関わらず、PCR 検査の実施数は少ない状況が読み取れます。」	1.可視化を水平に一列に並べて、画面のサイズに合わせるようにしました。 2.①地域間の違いを表現する種類を選んだのです。チュートリアルでは、異なる地域を異なる形状で区別することに成功しましたが、実際の実験では 7 回試しても再現できませんでした。（それにもかかわらず、実験を重ねる中でコツを掴み、可視化の効果も徐々に良くなってきました）②できる限り異なる可視化の種類を組み合わせるようにしました。例えば、一般的な point 図から、異なる形状（異なる地域を表す）と色の深さ（その地域の PCR 検査実施者数を示す）を持つ point 図へと変化させました。 3.可視化する時に「都道府県名」というデータ特性を、x 軸、y 軸、色、形状以外の部分に表示しないようにしました。

表 4.8　B グループのアンケートの結果（選択肢による回答）

ユーザ ID	必須条件を満たしたか	可視化の満足度	想定していたものとどの程度近かったか	可視化の読み取りやすさ	評価の反映の度合い	一番評価が反映されたタイミング	選択で最も重視した点
B01	はい	4	4	3	4	実験の前半	可視化の種類
B02	はい	4	4	4	4	実験の後半	可視化に含まれるデータ
B03	はい	5	5	5	5	実験の後半	可視化に含まれるデータ
B04	はい	5	1	5	4	実験の前半	可視化の構造
B05	はい	5	4	5	4	実験の前半	可視化の種類

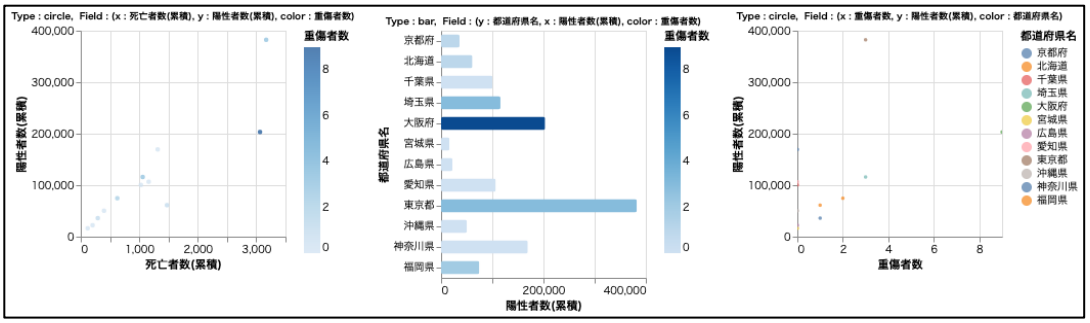
4.2.1. 実験協力者が生成した可視化

A グループの実験協力者が生成した可視化を, 最終的に可視化を選ぶときに重視した点で分け, 図 4.9, 図 4.10 にそれぞれ示す．図 4.9 は可視化の種類，図 4.10 は可視化に使用したデータを重視して選んでいる．同様に B グループの実験協力者が生成した可視化を，図 4.11，図 4.12，図 4.13 にそれぞれ示す．図 4.11 は可視化の種類，図 4.12 は可視化に含まれるデータ，図 4.13 は可視化の構造を重視して選んでいる．各図の下には作成した実験協力者の番号を記している．

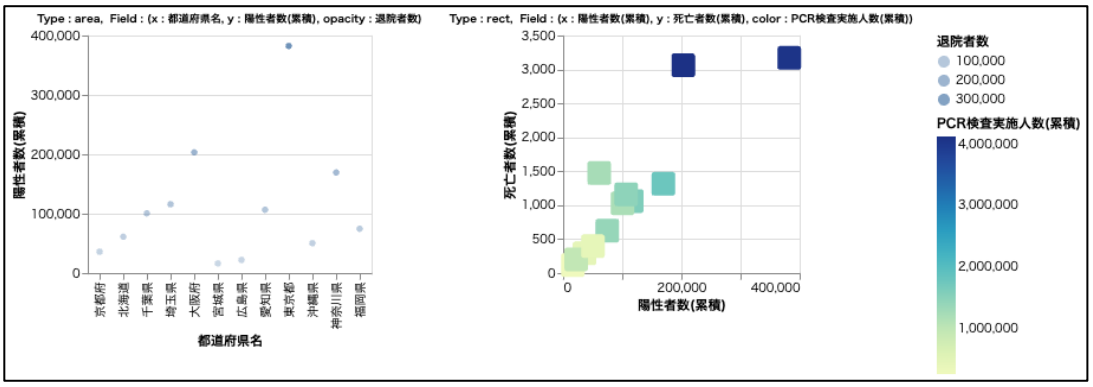
可視化の種類を重視して選んだ可視化については, 図 4.9, 図 4.11 より, 実験協力者 5 名のうち, A02, A05，B01，B05 は，棒グラフを含んだ可視化を選んでいる．また表 4.5 より，A03 は「都道府県別陽性者数がヒストグラムの図になるよう 3,4 度試した」と回答しているため，都道府県の陽性者数の棒グラフを作ろうとしていたと考える．評価で重視した点については，表 4.5，表 4.7 より，都道府県間の比較のしやすさを挙げている回答が多かったことがわかる．

可視化に含まれるデータを重視して選んだ可視化については, 図 4.10, 図 4.12 より, 実験協力者 A01, A04，B02，B03 の 4 名全員が陽性者数(累積)のデータを含んだ可視化を生成していることがわかる．また，A04，B02，B03 の 3 名が死亡者数(累積)のデータを含んだ可視化を生成している．表 4.5，表 4.7 より，評価においてはこれらのデータを含む可視化を重視していたことがわかる．

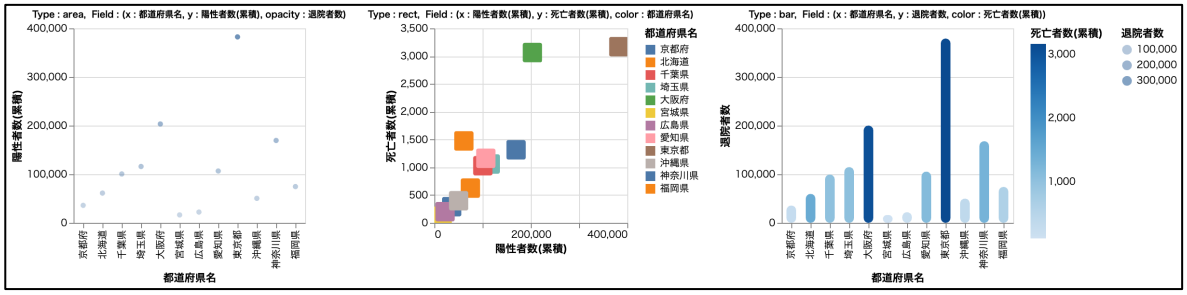
可視化の構造を重視して選んだ実験協力者は B04 の 1 名であった．図 4.13 より，実験協力者 B04 が選んだ可視化は，hconcat 構造の可視化であることがわかる．また評価で重視した点については，表 4.7 より,「可視化が横に 2 枚表示されているものが見やすいので評価した」と回答しており，意図した可視化が生成されているといえる．



A02

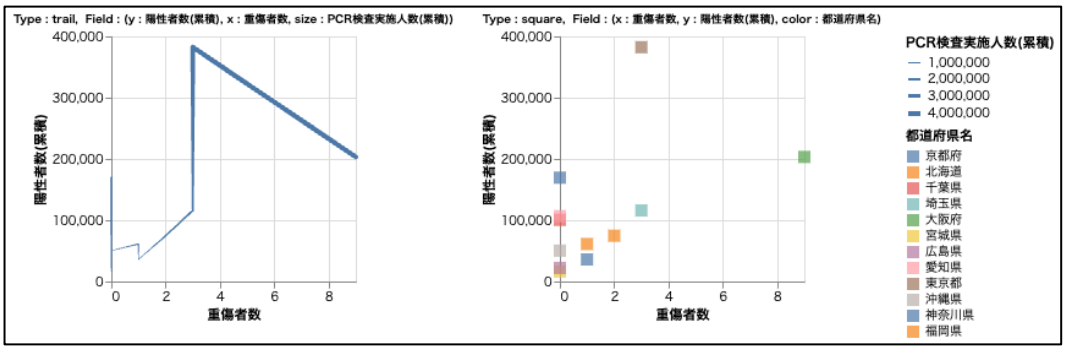


A03

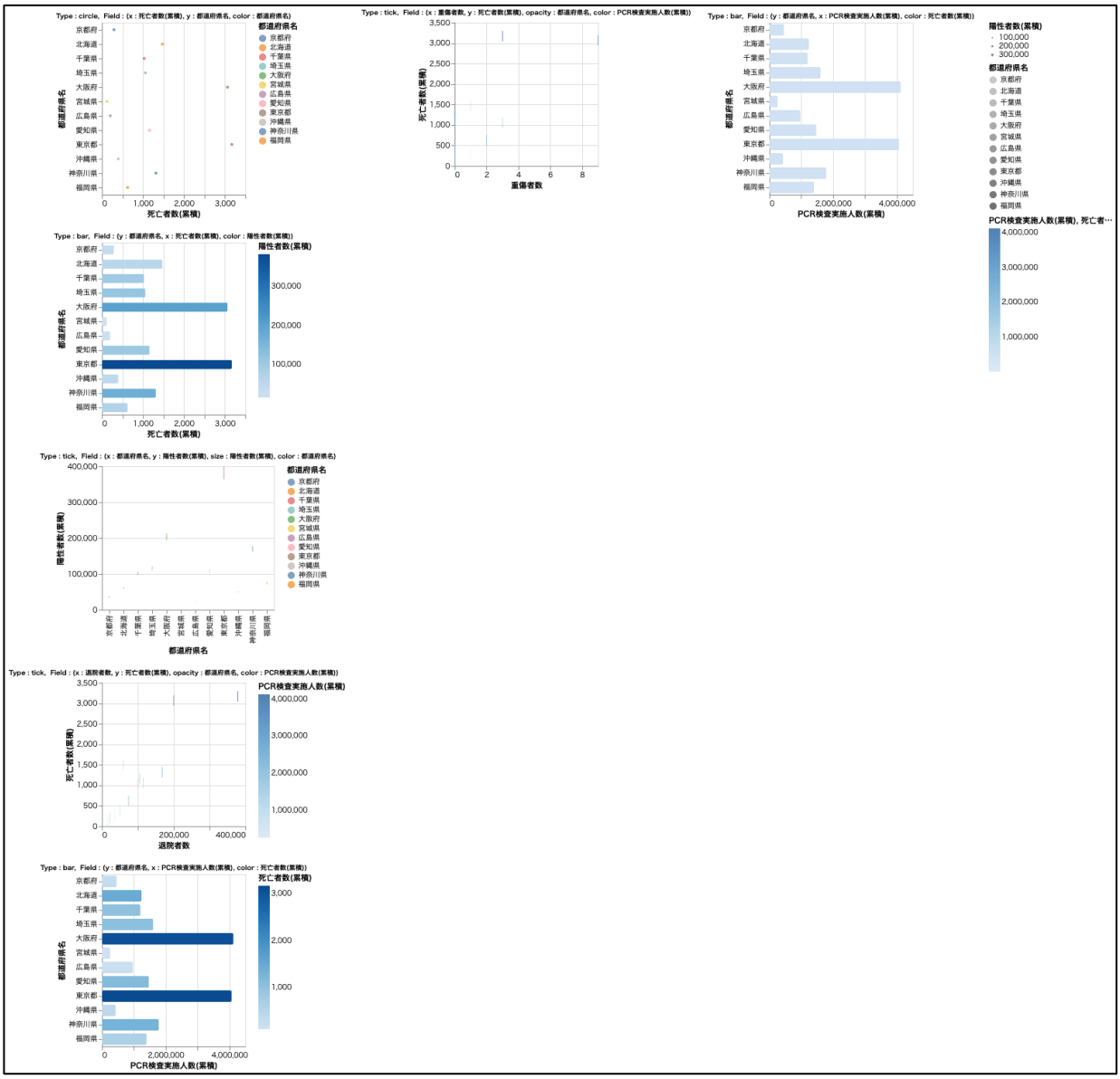


A05

図 4.9　A グループの生成結果(種類を重視した可視化)

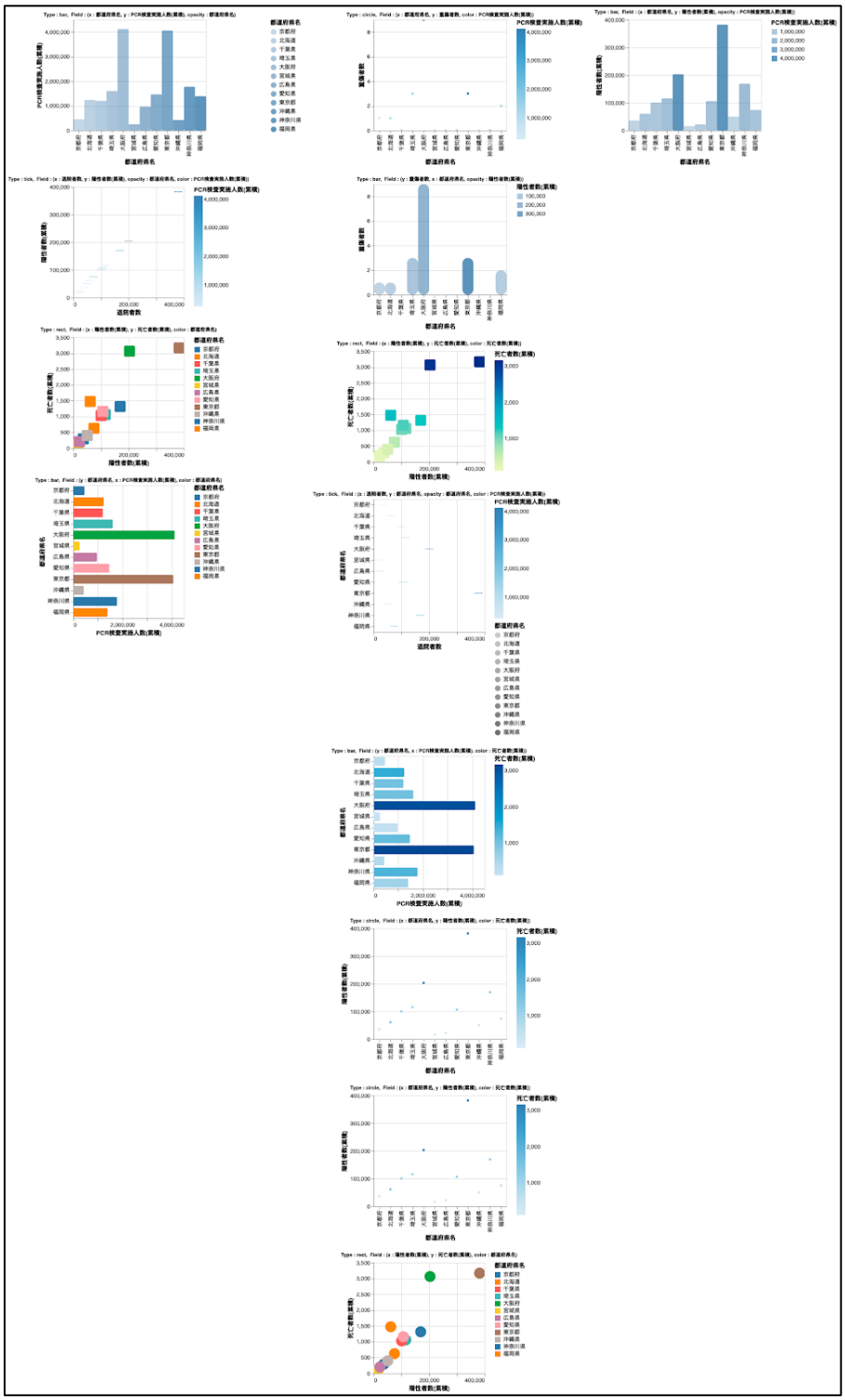


A01

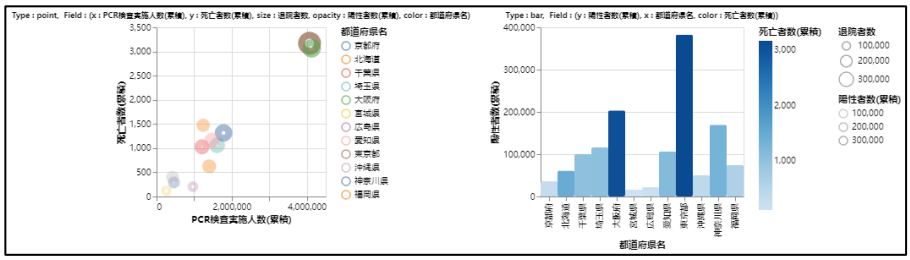


A04

図 4.10　A グループの生成結果(データを重視した可視化)

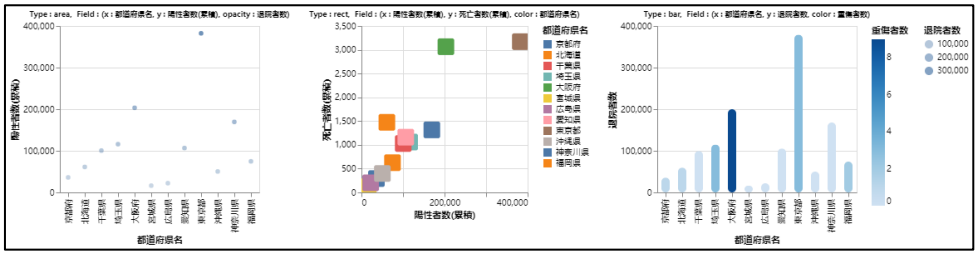


B01

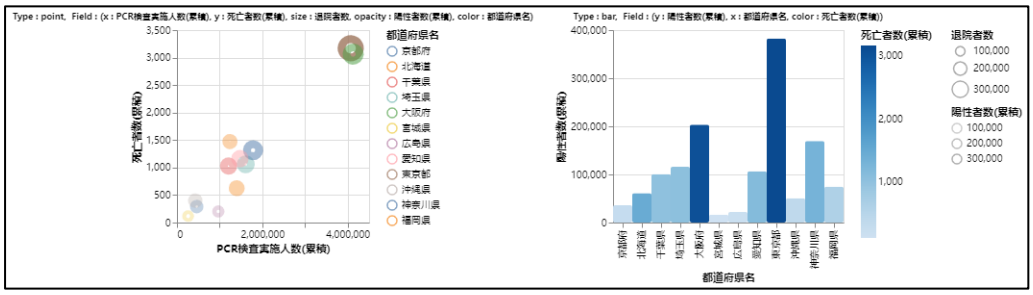


B05

図 4.11　B グループの生成結果(種類を重視した可視化)

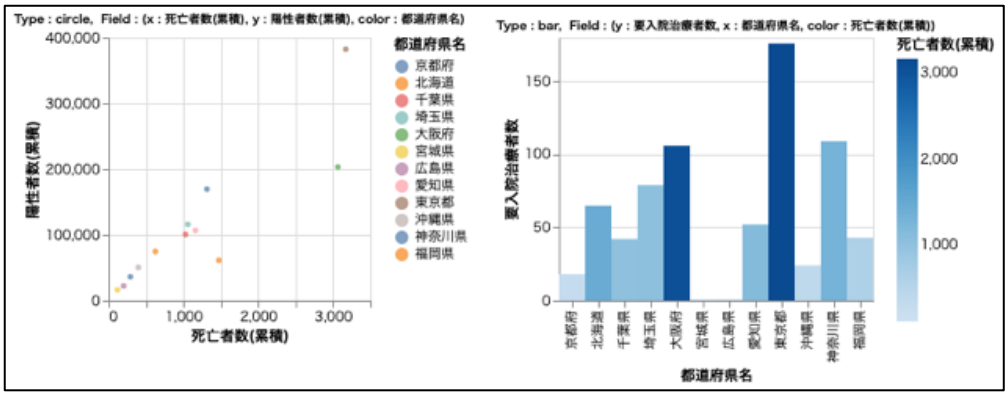


B02



B03

図 4.12　B グループの生成結果(データを重視した可視化)



B04

図 4.13　B グループの生成結果(種類を重視した可視化)

4.2.2. 実験結果の分析

最後に選んだ可視化について，満足度の分布を図 4.14，読み取りやすさの分布を図 4.1 5，次世代個体の生成時のユーザ評価の反映の度合いの分布を図 4.1 6 にそれぞれ示す.

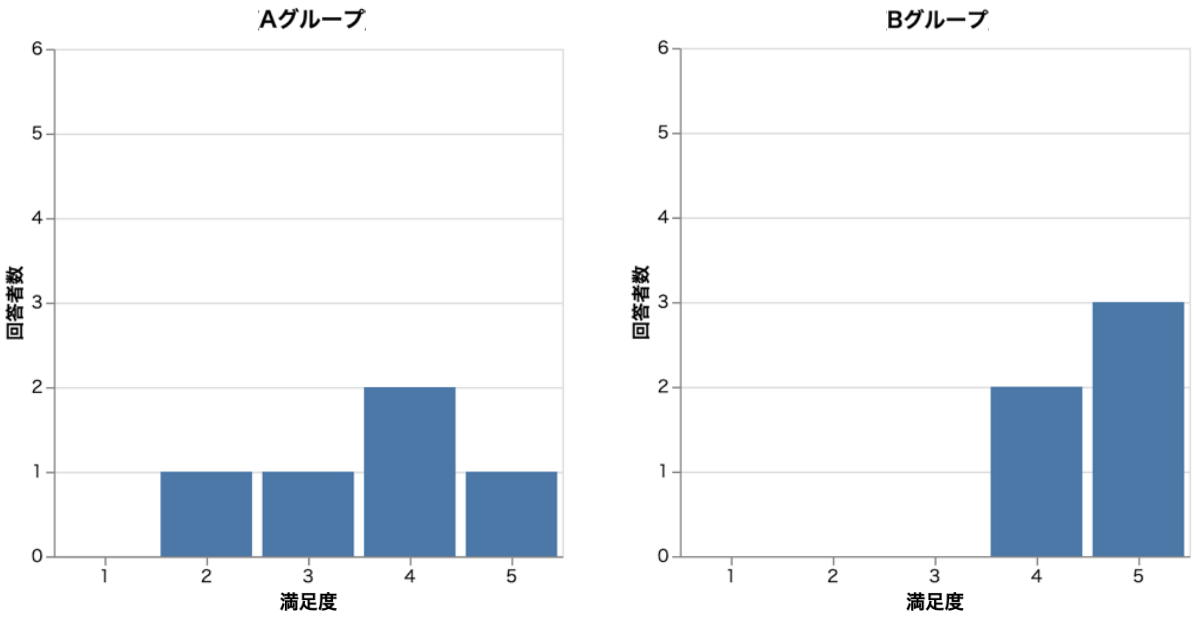


図 4.14　最後に選んだ可視化の満足度の分布

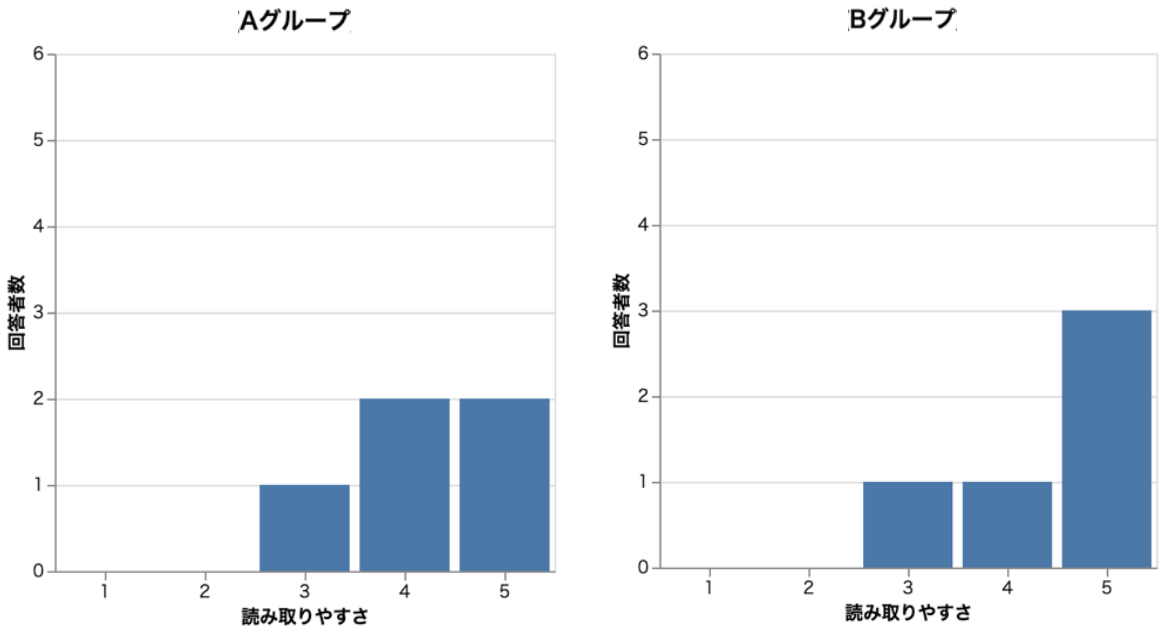


図 4.15　最後に選んだ可視化の読み取りやすさの分布

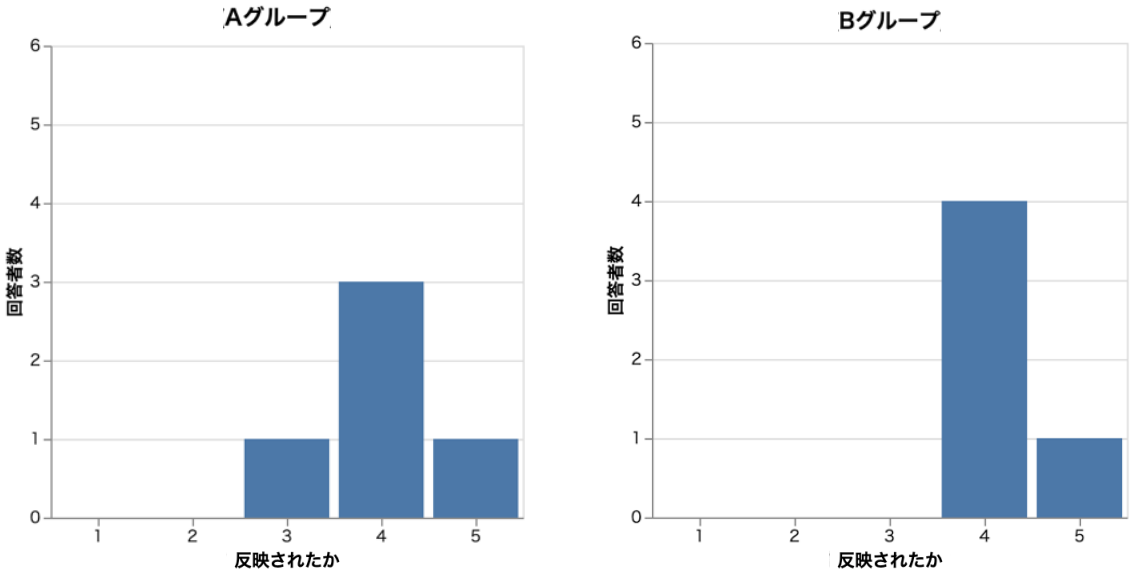


図 4.16　ユーザ評価の反映の度合いの分布

図 4.14 より，B グループの実験協力者は全員満足のいく可視化を生成できていることがわかる．一方で A グループについては，2 を選んだ実験協力者も 1 名存在するが，半数以上は満足している．図 4.15 より，多くの実験協力者が意味を読み取りやすい可視化が生成できたと回答していることがわかる．図 4.16 より，多くの実験協力者が自身の評価が反映されていると感じたことがわかる．また，満足度とユーザの反映の度合いについて，B グループは A グループよりも 4,5 を選んでいる人が多く，読み取りやすさについても，B グループは A グループよりも 5 を選んでいる人が多いことがわかる．これについては，B グループの方が，ユーザ評価のスキップ数が多かったため，次のユーザ評価が行われるまでの間に，GP オペレータの適用回数が多くなり，ユーザの評価がより反映されたことが原因と考える．

実験協力者から寄せられた意見としては，「自分好みの可視化がどんどん増えていく様子が面白かった」といった肯定的な意見があった一方で，「同じような可視化を複数含んでしまう個体があるのが気になった」という否定的な意見もあった．否定的な意見については，combination 構造の交叉において，可視化 id が増えた場合，もう片方の親から mark，encoding 構造が追加されることが原因で，同じ可視化を持つ個体が世代個体集団内に増え，それらの間で交叉が行われるようになったと考える．これについては，GTYPE の mark 構造，encoding 構造の類似度を用いて個体内に同一の可視化が含まれていないか確認し，含まれていた場合は交叉箇所を変えて再度やり直すようにすることで，改善可能と考える．

5. おわりに

本研究では，適応度予測を用いた対話型遺伝的プログラミングを用いて，ユーザの意向を反映した可視化を生成する手法を提案した．対話型進化計算の課題である，ユーザ評価にかかる負担を軽減するために，提案手法では，ユーザが評価した個体の GTYPE との類似度を用いて，適応度予測を行い，一部の世代のユーザ評価をスキップすることで，ユーザの評価回数を従来手法よりも減らした．提案手法を実装したプログラムを用いて次世代個体集団を生成する実験を行った結果，可視化の構造と可視化の種類については，ユーザの意向を反映した可視化を生成可能であることが示された．また，ユーザ実験を行った結果，ユーザが満足する可視化が生成可能であることが示された．

今後は，適応度予測の手法の改良，対応可能な可視化の拡張を行うことで，より手軽にユーザの考えを反映した多様な可視化の生成が可能になることが期待できる．前者に関して，提案手法では，部分構造を用いて適応度を予測しているが，各評価の観点ごとに，適応度の予測の手法を変更することで，より適切な予測を行うことができると考える．後者に関して，提案手法では，Vega-Lite コードを対象としたが，生成するコードを Vega や D3.js に変更することで，より多種多様な可視化が生成できると考える．

データの可視化は，分析者に対してデータの特徴をわかりやすく示すことができるため，関心が高まってきている．提案手法のようにユーザの意向を反映したデータの可視化を容易にする技術は，様々なドメインにおけるデータ活用の促進に貢献することが期待できる．

付録 A mark 構造内で設定した項目と encoding 構造内で設定した構造

Vega-Lite では，mark 構造で設定可能な項目と encoding 構造で設定可能な項目は異なっている．描画不可能な可視化が発生しないように設定を行い，今回の実験で設定した mark 構造の項目は表 A.1，設定した encoding の構造は表 A.2 にそれぞれ示した．設定した項目・構造は○で表記し，設定しなかった項目・構造は×で示した．

表 A.1 mark 構造内で設定した項目

可視化の種類	baseline	line	point	orient	interpolate	tension	tooltip	clip	Invalid	width
area	○	○	○	○	○	○	○	○	○	×
bar	○	×	×	×	×	×	○	○	○	○
circle	×	×	×	×	×	×	○	○	○	×
line	×	×	×	×	○	○	○	○	○	×
point	×	×	×	×	×	×	×	×	×	×
rect	○	×	×	×	×	×	○	○	○	○
square	×	×	×	×	×	×	○	○	○	×
text	○	×	×	×	×	×	○	○	○	×
tick	×	×	×	○	×	×	○	○	○	×
trail	×	×	×	○	×	×	○	○	○	×
可視化の種類	height	binSpacing	cornerRadius	size	strokeWidth	strokeCap	strokeDash	shape	angle	order
area	×	×	×	×	×	×	×	×	×	×
bar	○	○	○	×	×	×	×	×	×	×
circle	×	×	×	○	×	×	×	×	×	×
line	×	×	×	×	○	○	○	×	×	×
point	×	×	×	○	○	×	×	○	○	×
rect	○	×	○	×	×	×	×	×	×	×
square	×	×	×	○	×	×	×	×	×	×
text	×	×	×	×	×	×	×	×	×	×
tick	×	×	○	×	×	×	×	×	×	×
trail	×	×	×	×	×	×	×	×	×	○

表 A.2 encoding 構造内で設定した構造

グラフの種類	x	y	color	opacity	shape	size	text
area	○	○	○	○	○	×	×
bar	○	○	○	○	×	○	×
circle	○	○	○	○	×	○	×
line	○	○	○	×	○	×	×
point	○	○	○	○	○	○	×
rect	○	○	○	○	×	×	×
square	○	○	○	○	×	○	×
text	○	○	○	○	×	○	○
tick	○	○	○	○	×	○	×
trail	○	○	○	○	×	○	×

表 A.1 は，各可視化の種類ごとに mark 構造内の設定可能な項目を示したものである．mark 構造内で設定可能な項目は，散布図の点の形状を示す shape，角度を示す angle，大きさを示す size などがあり，可視化の細かい設定を行うことが可能である．

表 A.2 は，各可視化の種類ごとに，encoding 構造内の設定可能な構造を示したものである．encoding 構造内で設定可能な構造は，x 軸，y 軸，color などがあり，データ変数とプロット方法の関係を示している．

参考文献

[1] L. Wilkinson, “The Grammar of Graphics”, Springer Berlin Heidelberg, 2012.
[2] J. Heer, M. Agrawala, “Software Design Patterns for Information Visualization”, IEEE Trans. Visualization & Comp. Graphics, Vol.12, No.5, pp. 853-860, 2006
[3] J. Heer, S. K. Card, J. Landay, “Prefuse: A Toolkit for Interactive Information Visualization”, ACM Human Factors in Computing Systems (CHI), 2005
[4] M. Bostock, J. Heer, “Protovis A Graphical Toolkit for Visualization”, IEEE Trans. Visualization & Comp. Graphics, No.15, Vol.6, pp.1121–1128, 2009.
[5] M. Bostock, V. Ogievetsky, and J. Heer, “D3: Data-Driven Documents,” IEEE Trans. Visualization & Comp. Graphics,” Vol.17, No.12, 2011.
[6] A. Satyanarayan, K. Wongsuphasawat, and J. Heer, “Declarative Interaction Design for Data Visualization”, In Proceedings of the 27th annual ACM symposium on User interface software and technology, 2014.
[7] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer, “Reactive Vega: A streaming dataflow architecture for declarative interactive visualization”, IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2016.
[8] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, “Vega-Lite: A Grammar of Interactive Graphics,” IEEE Transactions on Visualization and Computer Graphics, Vol.23, No.1, 2017.
[9] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, J. Heer, “Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations”, IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2016
[10] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, J. Heer, “Voyager 2: Augmenting Visual Analysis with Partial View Specifications”, ACM Human Factors in Computing Systems (CHI), 2017
[11] K. Z. Hu, M. A. Bakker, S. Li, T. Kraska, and C. A. Hidalgo, “Vizml: A Machine Learning Approach to Visualization Recommendation,” In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, No.128, pp.1–12, 2019.
[12] H. Li, Y. Wang, S. Zhang, Y. Song, and H. Qu, “KG4Vis: A Knowledge Graph-Based Approach for Visualization Recommendation,” IEEE Transactions on Visualization and Computer Graphics, pp.1-11, 2021.
[13] V. Dibia and C. Demiralp, “Data2vis: Automatic Generation of Data Visualizations Using Sequence-to-sequence Recurrent Neural Networks,” IEEE Computer Graphics and Applications,” Vol.39, No.5, pp.33-46, 2019.
[14] A. Narechania, A. Srinivasan, J. Stasko “NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries”, IEEE Computer Graphics and Applications,” Vol.21, No.3, pp.23-50, 2022.
[15] R. Mitra, A. Narechania, A. Endert, J. Stasko, “Facilitating Conversational Interaction in Natural Language Interfaces for Visualization”, IEEE Visualization and Visual Analytics (VIS), pp.6-10, 2022.

[16] 伊庭斉志, “進化論的計算の方法,” 東京大学出版会, 1999.
[17] 北野宏明, “遺伝的アルゴリズム, ” 人工知能学会誌, Vol.7, No.1, pp.26-37, 1992.
[18] 若林真一, 小出哲士, 八田浩一, 中山喜勝, 後藤睦明, 利根直佳,
“交差手法の適応的選択機能を組み込んだ遺伝的アルゴリズムの LSI チップによる実現,” 情報処理学会論文誌 Vol.41, No.6, pp.1882-7764, 2000.
[19] 伊庭斉志, “遺伝的プログラミングと進化論的計算手法, ” 人工知能学会誌, Vol.15, No.2, pp.251-258, 2000.
[20] 高木英行, 畝見達夫, 寺野隆雄, “対話型進化計算法の研究動向, ” 人工知能, Vol.13, No.5, pp.692-703, 1998.
[21] 中川雄太, 吉田香, “対話型進化計算手法を用いたフォント自動生成システム,” 情報処理学会研究報告,” ヒューマンコンピュータインタラクション研究会報告, Vol.2010-HCI-136, No.3, pp.1-8, 2010.
[22] 是永, 基樹, 萩原, 将文, “対話型進化計算法によるインテリアレイアウト支援システム”, 情報処理学会論文誌”, No.41 Vol.11, pp.3152-3160, 2000
[23] 山上遼馬, 柴田祐樹, 高間康史, “対話型遺伝的プログラミングを用いた可視化生成システムの提案”, 第 36 回人工知能学会全国大会, 2022/06
[24] 花木 康, 橋山智訓, 大熊 繁: “適応度の推論による進化的アルゴリズムの計算時間の短縮”, 電気学会論文誌C(電子・情報・システム部門誌), 2004, Vol.124, No.9, pp.1853-1860
[25] 川中 普晴, 吉川 大弘, 三橋 麗子, 伴野 佳史, 篠木 剛, 鶴岡 信治, 染色体の類似度を考慮した評価値推論システム, 電気学会論文誌C(電子・情報・システム部門誌), Vol.123, No.3, pp.568-575, 2003
[26] 田中 雅晴, 溝口 正信,高見 勲: “適応度予測型遺伝的アルゴリズムにおける探索性能の向上” 電気学会論文誌C(電子・情報・システム部門誌), No.124, Vol.9, pp.1853-1860, 2004
[27] 伴野 佳史, 吉川 大弘,川中 普晴, 解空間に適した染色体の類似度導出法に関する一考察, ファジィシステムシンポジウム講演論文集 Vol.18, pp.445-448, 2002/08
[28] 大崎美穂,高木英行: “対話型操作者の負担低減 -評価値予測による提示インタフェースの改善-“, 人工知能学会誌, Vol 13, No.5, pp.41-51 1998.
[29] H. Nakayama, M. Arakawa, and R. Sasaki, “Optimization of Unknown Objective Functions by RBF Networks and Genetic Algorithms,” Trans. of the Institute of Systems, Control and Information Engineers, Vol.13, No.3, pp.152–154, 2000/03
[30] 佐藤嘉洋, 栢菅彩, 有田隆也, “適応度予測に基づく対話型進化計算とその似顔絵生成への応用”, 第 32 回知能システムシンポジウム論文集, pp.199-204, 2005.

学会発表

1. 山上遼馬, 柴田祐樹, 高間康史, “対話型遺伝的プログラミングを用いた可視化生成システムの提案”, 第 36 回人工知能学会全国大会, 2022/06

謝辞

本研究を進めるにあたり，指導教員の高間康史教授から，丁寧なご指導と適切な助言を賜りました．心より感謝いたします．同じく，本研究を進めるにあたり，ご指導と助言をいただいた柴田祐樹助教にも深く感謝いたします．また，修士論文発表会にて，助言をいただいた副査の横山先生と片山先生に深く感謝いたします．お忙しい時期に，実験にご協力いただいた高間研究室の皆様にも，この場を借りてお礼を申し上げます．