

1. 背景

同一のソースコード集合に対してであっても、ツールの特性やパラメータによって検出されるコードクローンは大きく変化する

- デフォルトパラメータと調整したパラメータでは、再現率は変化せずに適合率が3倍以上向上した[1]
- ある1つのツールで検出されたコードクローンの多くは他のツールでは検出されない[2]

➡ 1つのコードクローン検出結果だけでは正確な分析が困難

様々なツール・パラメータで検出を行い結果の共通部分や差異を知ることが重要

3. 提案手法

Step 1: クローンペアのマッピング[2]

重複度と包含度から、異なる検出結果間で似たクローンペア同士をマッピング

Step 2: ミスマッチ率の計算

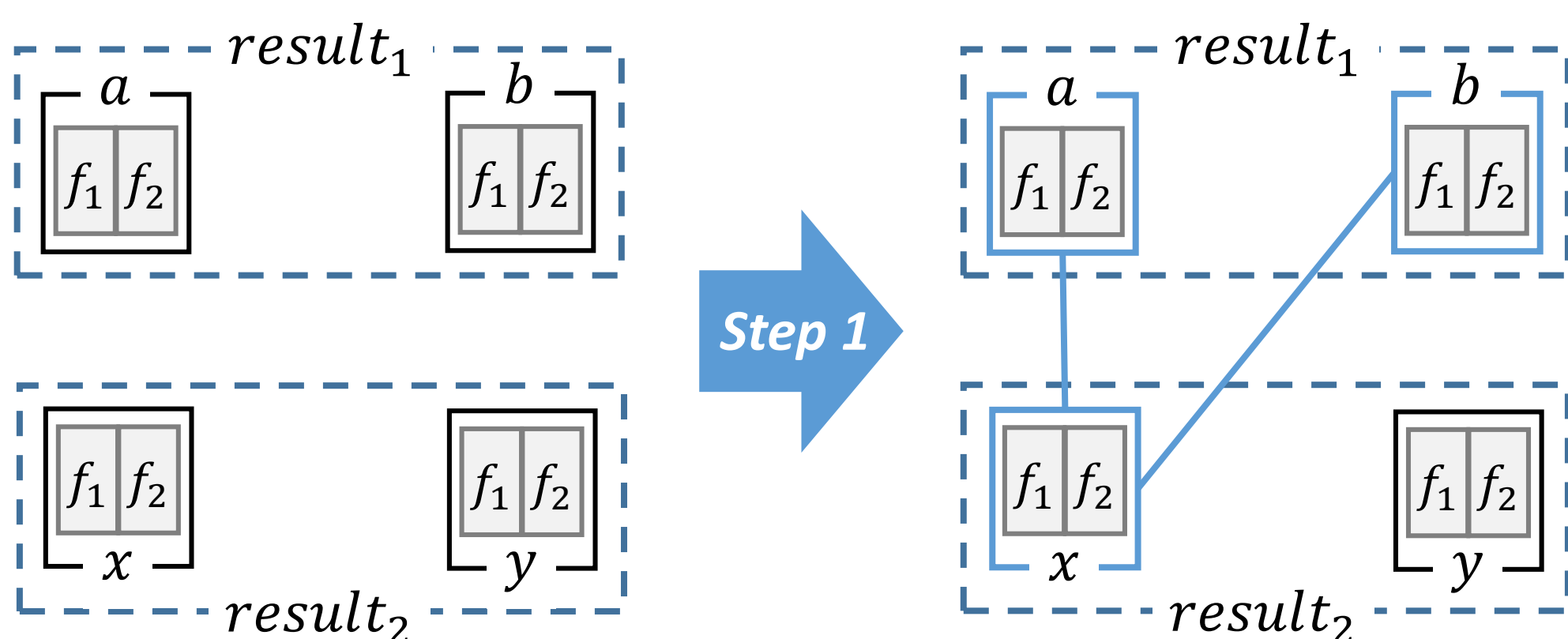
ミスマッチ率: ある検出結果を基準として他の検出結果のクローンペアとマッピングされていない割合

- $result_1$ が基準: $mismatch(s_0, s_1) = 0/2 = 0.0$
- $result_2$ が基準: $mismatch(s_0, s_1) = 1/2 = 0.5$

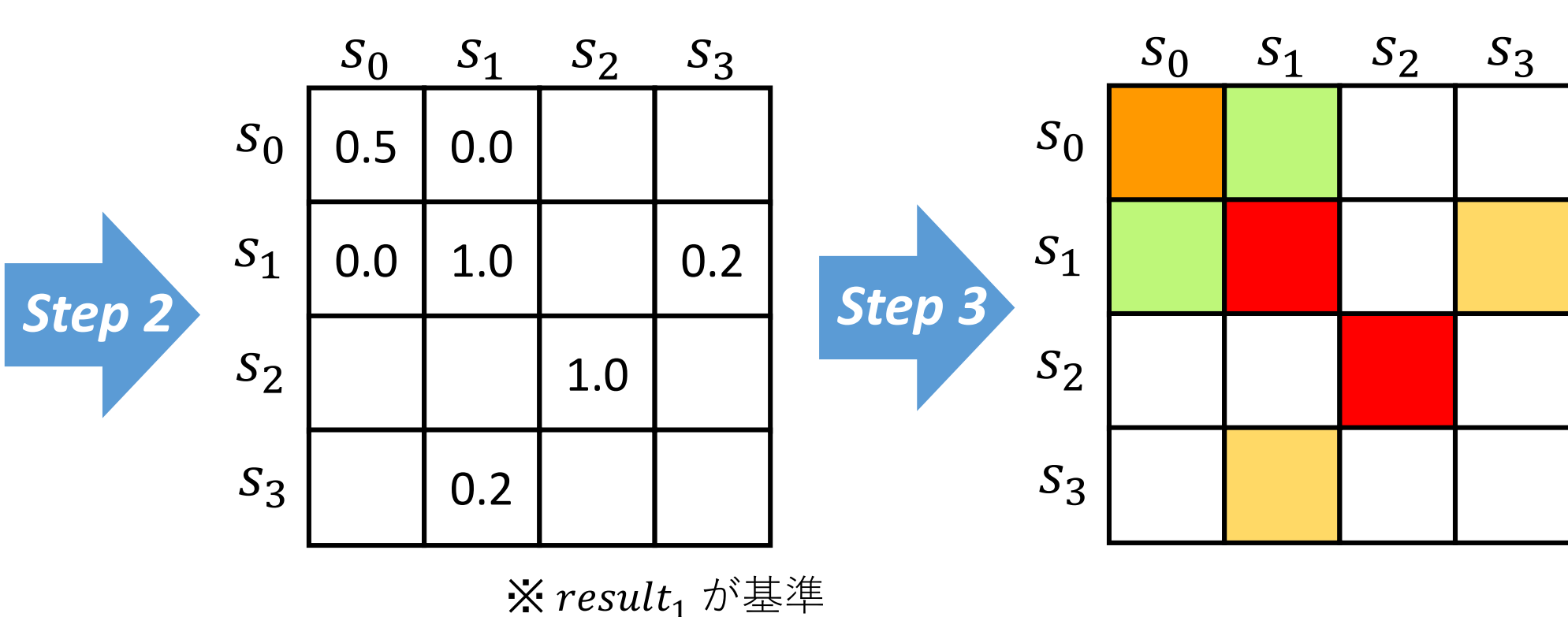
Step 3: 散布図を用いた比較結果の提示

各点にミスマッチ率に応じた色を表示

- 白: クローンペアが存在しない
- 緑: 0%
- 赤: 高い
- 黄: 低い



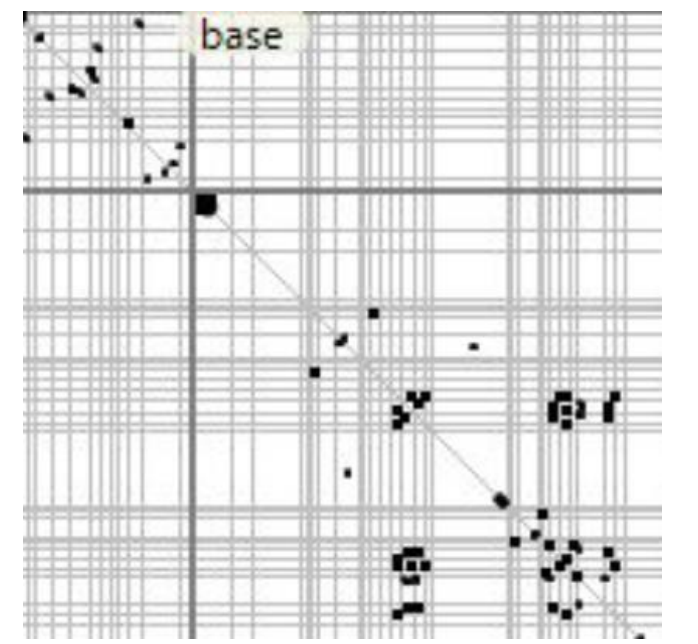
※a, b, x, y はファイル s_0, s_1 間のクローンペア



2. コードクローン検出結果の可視化手法

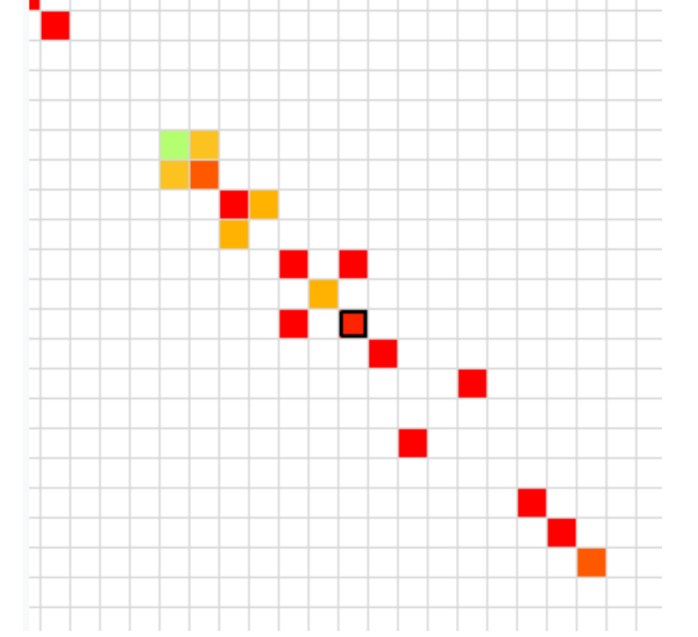
既存手法: クローン散布図

- 両軸に出現順にトークンを並べる
- コードクローンが存在する部分に点を表示
- コードクローンの分布を表す



提案手法の散布図

- 両軸に出現順にファイルを並べる
- コードクローンが存在する部分にミスマッチ率に応じた色を表示
- 2つの検出結果の差を表す



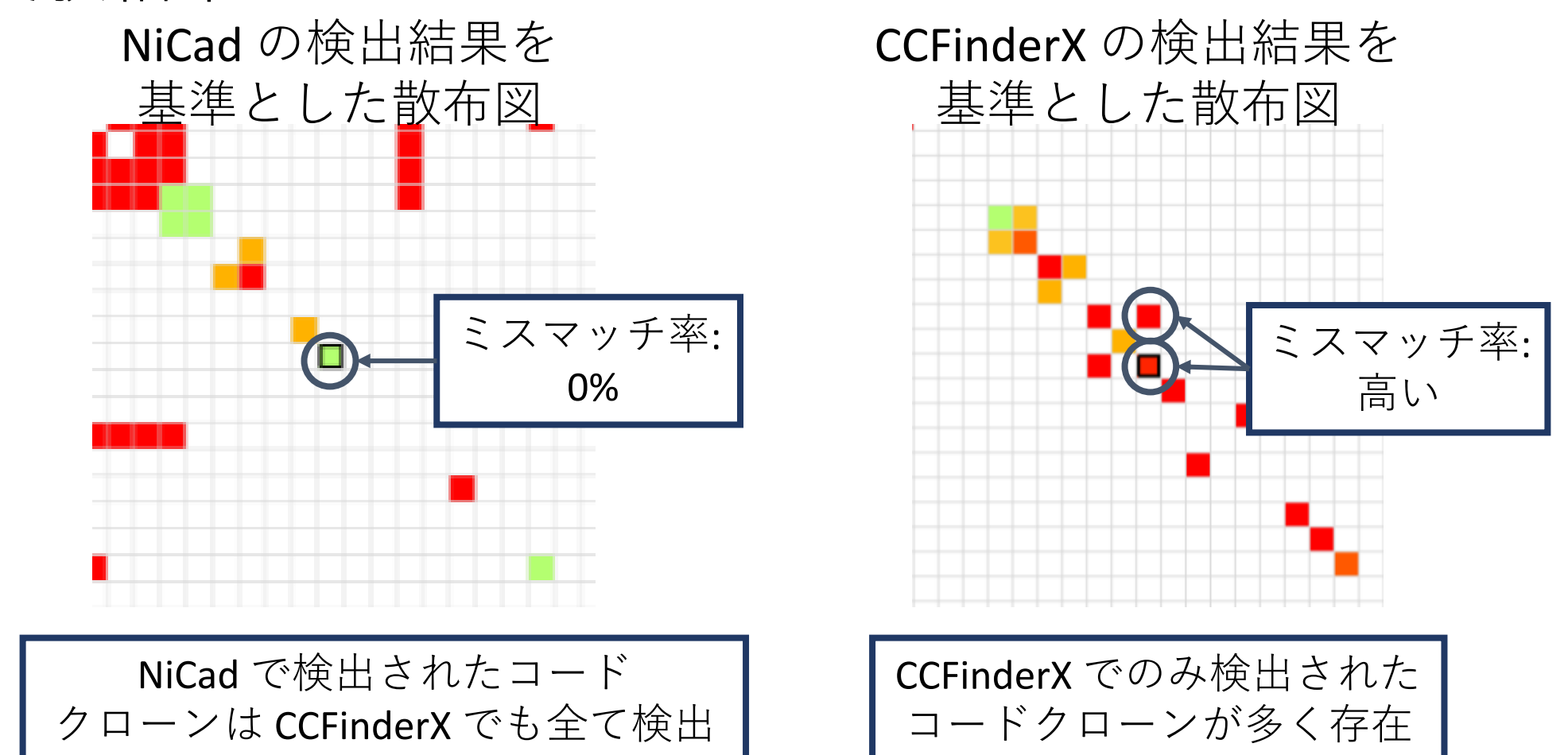
4. 適用実験

実際のソフトウェア(jEdit)において、複数のコードクローン検出結果を比較することで重要なコードクローンを発見する

シナリオ:

- PrinterDialog.java にバグが発見された
- NiCad を使用して同様のバグを含むコード片を探したが見つからなかった
- 更にCCFinderXを使用して検出漏れが無い確認したい

実験結果:



- CCFinderX でのみ検出されたクローンペアを調査 → バグ部分のコードクローンが1つ存在
- 発見されたコードクローンは実際に同時にバグ修正が行われていた

検出結果を比較することで重要なコードクローンを発見できた

5. 今後の課題

片方のコードクローン検出結果にしか含まれないクローンペアのうち重要なものを優先的に開発者に提示

[1] T. Wang, M. Harman, Y. Jia, and J. Krinke. Searching for better configurations: a rigorous approach to clone evaluation. In Proc. ESEC/FSE 2013, pp. 455–465, Saint Petersburg, Russian Federation, 2013.

[2] S. Bellon, R. Koschke, G. Antoniol, J. Krinke and E. Merlo. Comparison and Evaluation of Code Clone Detection Tools. IEEE Transactions on Software Engineering, Vol. 33, No. 9, pp. 577–591, 2007.