Barcode Layout Optimization in Spatial Transcriptomics: Theory and Experiments

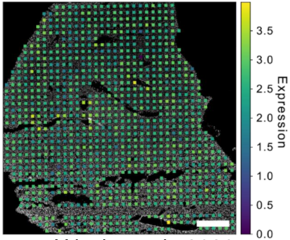
Frederik L. Jatzkowski, Antonia Schmidt, Robert Mank, Steffen Schüler, Matthias Müller-Hannemann

Martin Luther University Halle-Wittenberg

March 01, 2024

Introduction

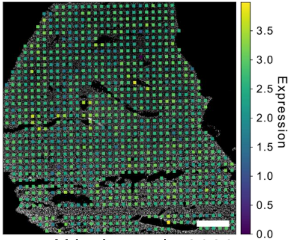
- **Application**: Spatial transcriptomics with high-resolution microarrays of DNA-barcodes



Wirth et al. 2023

Introduction

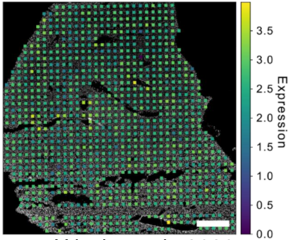
- **Application**: Spatial transcriptomics with high-resolution microarrays of DNA-barcodes
- **Problem**: High error rates during barcode synthesis by photolithography $(10 - 20\%$ per base)



Wirth et al. 2023

Introduction

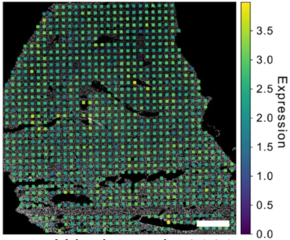
- **Application**: Spatial transcriptomics with high-resolution microarrays of DNA-barcodes
- **Problem**: High error rates during barcode synthesis by photolithography ($10 - 20\%$ per base)
- **Goal**: Reduce errors during synthesis



Wirth et al. 2023

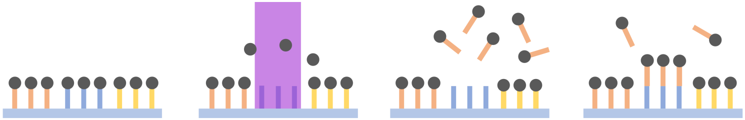
Introduction

- **Application**: Spatial transcriptomics with high-resolution microarrays of DNA-barcodes
- **Problem**: High error rates during barcode synthesis by photolithography ($10-20\%$ per base)
- **Goal**: Reduce errors during synthesis
- **Approach**: Barcode selection and placement

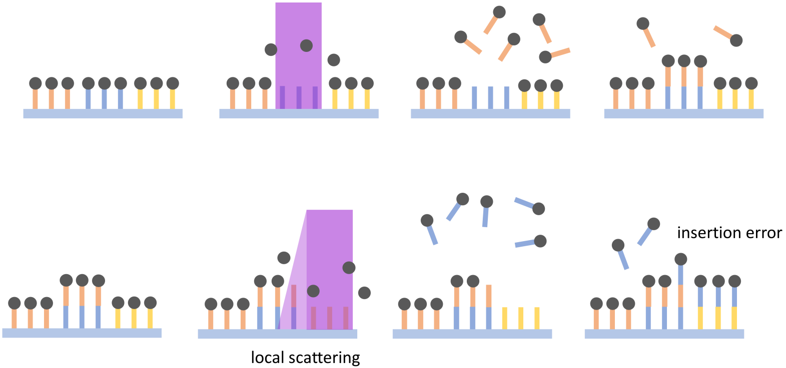


Wirth et al. 2023

Barcode Synthesis



Barcode Synthesis



BARCODE-LAYOUT Problem

Objective: Minimize situations, in which only one of two adjacent barcodes is illuminated.

BARCODE-LAYOUT Problem

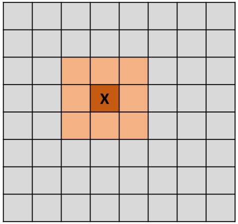
Objective: Minimize situations, in which only one of two adjacent barcodes is illuminated.

- similarity measured by `synthesis distance`

BARCODE-LAYOUT Problem

Objective: Minimize situations, in which only one of two adjacent barcodes is illuminated.

- similarity measured by `synthesis distance`
- cost = sum of distances between neighboring barcodes

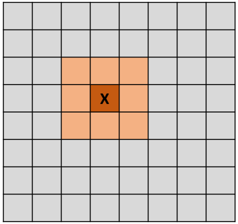


BARCODE-LAYOUT Problem

Objective: Minimize situations, in which only one of two adjacent barcodes is illuminated.

- similarity measured by `synthesis distance`
- cost = sum of distances between neighboring barcodes

Task:



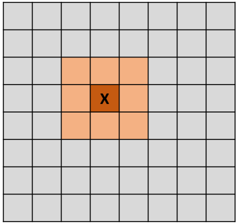
BARCODE-LAYOUT Problem

Objective: Minimize situations, in which only one of two adjacent barcodes is illuminated.

- similarity measured by `synthesis distance`
- cost = sum of distances between neighboring barcodes

Task:

- Select barcodes from a large set of candidates



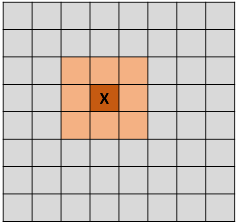
BARCODE-LAYOUT Problem

Objective: Minimize situations, in which only one of two adjacent barcodes is illuminated.

- similarity measured by `synthesis distance`
- cost = sum of distances between neighboring barcodes

Task:

- Select barcodes from a large set of candidates
- Assign barcodes to locations



Related Work

Border Length Minimization Problem

- Discussed by Hannenhalli et al. 2002, Kahng et al. 2004 and Carvalho Jr. and Rahmann 2008
- Main difference: no selection of probes
- Application: diagnostics

Related Work

Border Length Minimization Problem

- Discussed by Hannenhalli et al. 2002, Kahng et al. 2004 and Carvalho Jr. and Rahmann 2008
- Main difference: no selection of probes
- Application: diagnostics

Quadratic Assignment Problem

- Assign n facilities to n locations minimizing a distance-dependent cost function

Contribution

Contribution

- Proof for maxSNP-hardness $\Rightarrow$ hard to approximate

Contribution

- Proof for maxSNP-hardness $\Rightarrow$ hard to approximate
- Study of lower bounds for layout cost

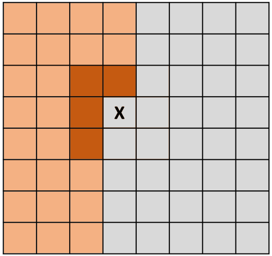
Contribution

- Proof for maxSNP-hardness $\Rightarrow$ hard to approximate
- Study of lower bounds for layout cost
- Development and assessment of heuristics

Layout Generating

Greedy Algorithms

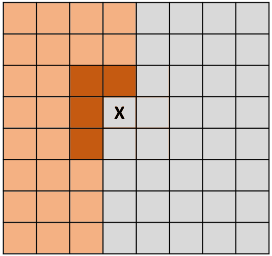
- Iteratively fill positions by choosing the best remaining candidate



Layout Generating

Greedy Algorithms

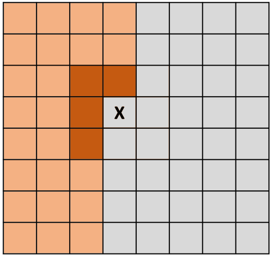
- Iteratively fill positions by choosing the best remaining candidate
- Previously used with limited lookahead (Kahng et al. 2004)



Layout Generating

Greedy Algorithms

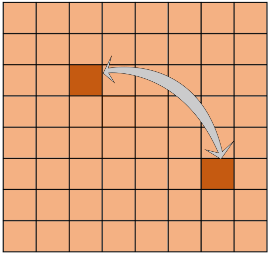
- Iteratively fill positions by choosing the best remaining candidate
- Previously used with limited lookahead (Kahng et al. 2004)
- GPU parallelization ⇒ all remaining barcodes can be searched



Layout Improving

Local search (2-OPT)

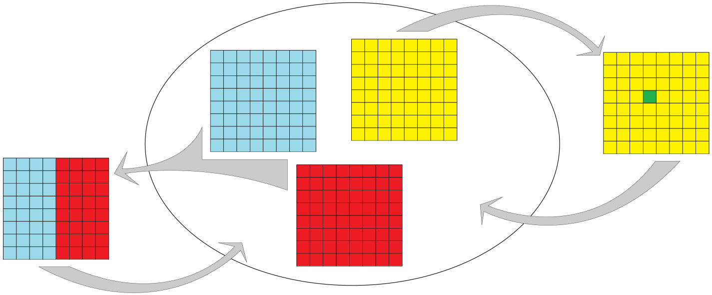
- Swap barcodes if it improves layout cost
- Stop when local minimum is reached



Layout Improving Algorithms

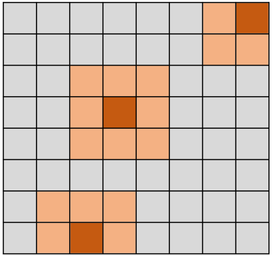
Genetic Algorithm (GA)

- Population of 1024 layouts
- Simulates natural evolution



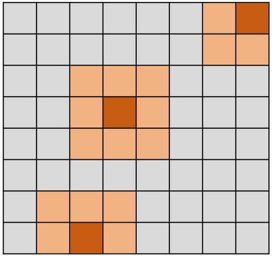
Lower Bounds for Layout Cost

1. Bound from Kahng et al. 2004 (modified):
    - Each barcode selects 8 closest neighbors
    - Discard surplus of neighborhood relationships



Lower Bounds for Layout Cost

1. Bound from Kahng et al. 2004 (modified):
   - Each barcode selects 8 closest neighbors
   - Discard surplus of neighborhood relationships
2. LP relaxation of Integer Linear Programming formulation
3. Gilmore Lawler Bound
4. Bound based on *b*-matching



Comparison of the algorithms

- 768×1024 array
- $768,432$ barcodes
- Gain: improvement over a random layout
- Gap: optimality gap to the lower bound

Algorithm	Gain (%)	Gap (%)

Comparison of the algorithms

- 768×1024 array
- $768,432$ barcodes
- Gain: improvement over a random layout
- Gap: optimality gap to the lower bound

Algorithm	Gain (%)	Gap (%)
Greedy (lookahead)	29.90	49.60
Greedy (unlimited)	37.16	34.16

Comparison of the algorithms

- 768×1024 array
- $768,432$ barcodes
- Gain: improvement over a random layout
- Gap: optimality gap to the lower bound

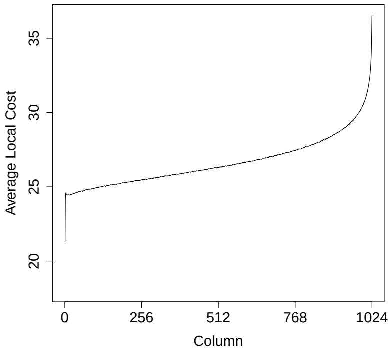
Algorithm	Gain (%)	Gap (%)
Greedy (lookahead)	29.90	49.60
Greedy (unlimited)	37.16	34.16
Random + 2-OPT	28.56	52.51
Greedy + 2-OPT	**37.18**	**34.12**

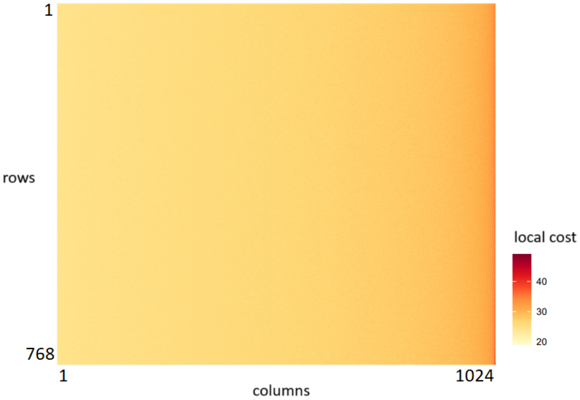
Comparison of the algorithms

- 768×1024 array
- $768,432$ barcodes
- Gain: improvement over a random layout
- Gap: optimality gap to the lower bound

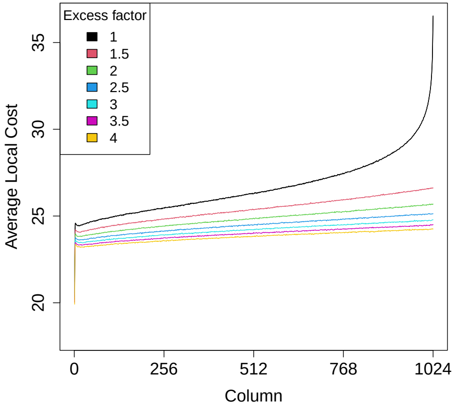
Algorithm	Gain (%)	Gap (%)
Greedy (lookahead)	29.90	49.60
Greedy (unlimited)	37.16	34.16
Random + 2-OPT	28.56	52.51
Greedy + 2-OPT	**37.18**	**34.12**
Random + GA	20.13	70.50
Greedy + GA	35.70	37.31

Local Costs in Greedy Results

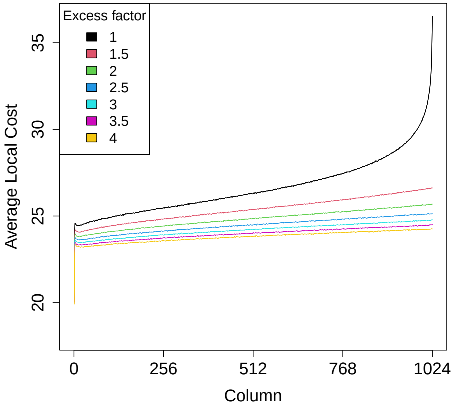




Providing Excess Barcodes

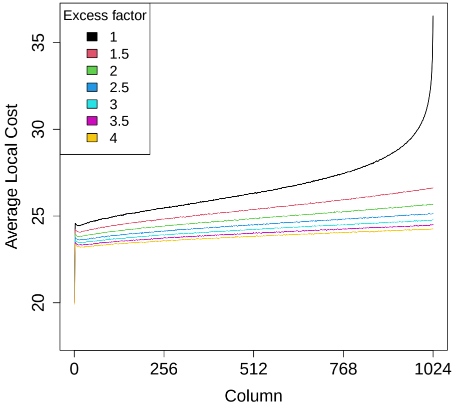


Providing Excess Barcodes



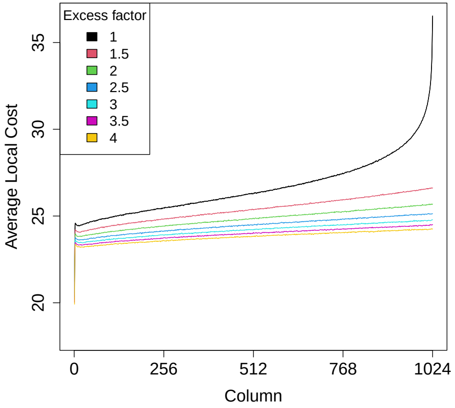
- Steep deterioration disappears

Providing Excess Barcodes



- Steep deterioration disappears
- Gain increased from 37.18% to 43.42%

Providing Excess Barcodes



- Steep deterioration disappears
- Gain increased from 37.18% to 43.42%
- Runtime increased linearly

Conclusions

Conclusions

- BARCODE-LAYOUT is hard to optimize

Conclusions

- BARCODE-LAYOUT is hard to optimize
- Parallel greedy algorithm shows promising results

Outlook

Conclusions

- BARCODE-LAYOUT is hard to optimize
- Parallel greedy algorithm shows promising results

Outlook

- Experiments with alternative barcode sets

Conclusions

- BARCODE-LAYOUT is hard to optimize
- Parallel greedy algorithm shows promising results

Outlook

- Experiments with alternative barcode sets
- Improved error models

Conclusions

- `BARCODE-LAYOUT` is hard to optimize
- Parallel greedy algorithm shows promising results

Outlook

- Experiments with alternative barcode sets
- Improved error models
- Experimental validation of error reduction

Lower Bounds

LP relaxation

- Lawler's linearization(Lawler 1963)
- $\mathcal{O}(wh|B|^2)$ variables

ILP formulation

$$\min \quad 2 \cdot \left(\sum_{x=1}^{w} \sum_{y=1}^{h} \sum_{b \in B} \sum_{b' \in B, b' \neq b} \sum_{(x',y') \in N(x,y)} y_{(xyb),(x'y'b')} \cdot d(b, b') \right)$$

s. t.

$$\sum_{x=1}^{w} \sum_{y=1}^{h} x_{xyb} = 1 \qquad \forall b \in B$$

$$\sum_{b \in B} x_{xyb} = 1 \qquad \forall x \in \{1, ..., w\}, \forall y \in \{1, ..., h\}$$

$$x_{xyb} x_{x'y'b'} - 2 \cdot y_{(xyb)(x'y'b')} \geq 0 \qquad \forall b \neq b' \in B, (x', y') \in N(x, y)$$

$$\sum_{x=1}^{w} \sum_{y=1}^{h} \sum_{b \in B} \sum_{(x',y') \in N(x,y)} \sum_{b' \in B, b \neq b'} y_{(xyb),(x'y'b')} = m \qquad m \text{ edge count in a grid}$$

$$x_{xyb} \in \{0, 1\}, y_{(xyb),(x'y'b')} \in \{0, 1\} \qquad \forall 1 \leq x \leq w, 1 \leq y \leq h, b \neq b' \in B, (x'y,') \in N(x, y)$$

Gilmore Lawler Bound

Gilmore Lawler Bound

- For each barcode, choose the best 3, 5, and 8 neighbors (parallelized computation)
- For each barcode, decide if it is best to be placed in a corner, border or middle position (ILP)
- Take into account the required corner, border and middle positions

Gilmore Lawler Bound: ILP Formulation

$$
\begin{aligned}
\text{min} \quad & 2 \cdot \sum_{b \in B} \left(l_{b,3nb} \cdot x_{b,3nb} + l_{b,5nb} \cdot x_{b,5nb} + l_{b,8nb} \cdot x_{b,8nb} \right) \\
\text{s. t.} \quad & \sum_{b \in B} x_{b,3nb} = 4 \\
& \sum_{b \in B} x_{b,5nb} = 2 \cdot (dimX - 2) + 2 \cdot (dimY - 2) \\
& \sum_{b \in B} x_{b,8nb} = (dimX - 2) \cdot (dimY - 2) \\
& x_{b,3nb} + x_{b,5nb} + x_{b,8nb} = 1 \qquad b \in B \\
& x_{b,3nb}, x_{b,5nb}, x_{b,8nb} \in \{0,1\}, b \in B
\end{aligned}
$$

b-matching bound

b-matching bound

- Consider a complete undirected graph over the barcode set
- Choose the required number of edges such that each node has degree $3 \leq deg \leq 8$

b-matching bound (ILP Formulation)

$$\min 2 \cdot \sum_{b \in B} \sum_{c \neq b \in B} d_{synth}(b, c) \cdot x_{bc}$$

$$\text{s. t.} \sum_{b \in B} \sum_{c \neq b \in B} x_{bc} = m$$

$$\sum_{c \neq b \in B} x_{bc} \leq 8 \qquad b \in B$$

$$\sum_{c \neq b \in B} x_{bc} \geq 3 \qquad b \in B$$

$$x_{b,c} \in \{0, 1\}, b \neq c \in B$$

Experiment for lower bounds

Table: Lower bounds for different array sizes and barcode sets of size $w \cdot h$. Entries with NA stand for instances which we could not solve within reasonable time and memory.

method	10×10	15×15	20×20	100×100	1024×768
LP	13,216	30,120	NA	NA	NA
GLB	21,344	48,202	84,988	1,881,900	119,211,464
Kahng	20,964	47,744	84,376	1,884,112	119,215,966
b-matching	21,032	47,852	84,676	1,884,904	NA

Calculation of expected cost of a random layout

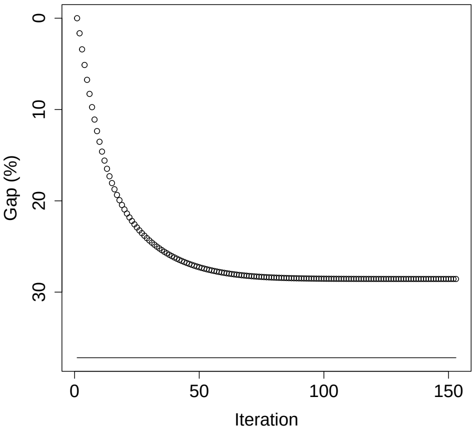
Expected Layout Cost

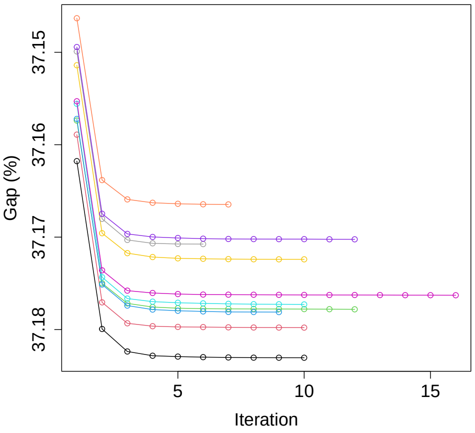
- Calculate the expected synthesis distance $E(d_{synth})$ between two barcodes sampled uniformly at randomly from B
- Expected layout cost for $w \times h$-array $E(cost) = 2m \cdot E(d_{synth})$ with $m := 2 \cdot (w-1) \cdot (h-1) + w \cdot (h-1) + (w-1) \cdot h$

Computed Layout Cost

- For set B with $|B| = 768 \cdot 1024$ and a 1024×768 array
- $E(cost) = 254,498,050$
- Empirically: $254,485,241.6$ with a standard deviation of $14,378.1$

Performance of 2-OPT





Objective value for different barcode set sizes

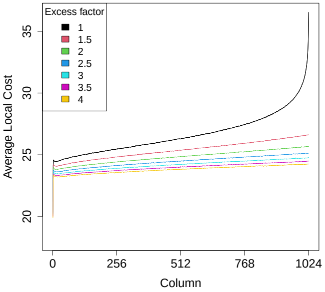


Figure: Average local cost of every column in the layout for each barcode set size used.

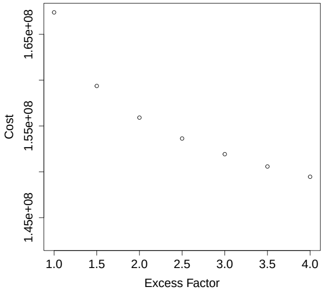


Figure: The average cost value for each barcode set size used.

Objective value for different barcode sets

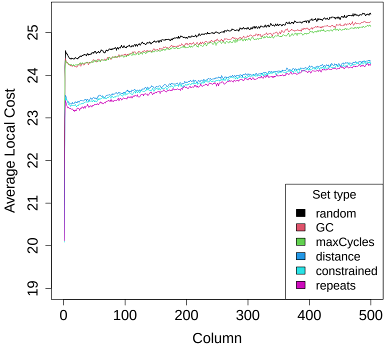


Figure: Average local cost of every column in the layout for each barcode set type used.