Review of: "A Novel Variable Neighborhood Search Approach for Cell Clustering for Spatial Transcriptomics"

Simon de Givry

Potential competing interests: No potential competing interests to declare.

The paper proposes the application of a well-known metaheuristic, variable neighborhood search, to single cell clustering. The minimization criterion combines a distance between gene expressions with a spatial distance between cells. Different embeddings are evaluated to measure the gene expression distance.

A background on the main classical clustering methods is given. Although interesting, it lacks an introduction to the type of clustering problem solved to understand how these clustering methods are adequate for the problem. Then different existing embedding methods are presented. Again, it is unclear if these methods are dedicated to the problem or not.

The notion of centroid is exploited but not introduced (when speaking about k-means/k-medoids).

The problem formulation is very close to the p-median. I don't see what is specific to cell clustering apart from a linear combination of spatial distance and gene expression similarity. Tuning the alpha hyperparameter might be complex. I expect some form of normalization for the two contributions. Looking at their Pareto front might be interesting too.

The justification of using VNS is missing. Why use this metaheuristic?

A comparison with a basic greedy heuristic (selecting the best centroid at each iteration until k_max selections), and a simple lower bound computed by taking the sum of second min values per gene, would help to evaluate the remaining optimality gap.

Overall, the technical part of the paper is sound, and the experimental results are convincing, showing a clear improvement compared to existing clustering methods with respect to clustering similarity to ground truth solutions.

Minor corrections:
p.5 line 188 Title Section "Implementation" should be replaced by "Methods"

What is the cosine similarity of two embeddings?

p.5 line 209 Eq (3) : replace q by y
Eq (2) is incorrect: it says only one i and j should be swapped.

Which value was used for max_iter?

Why in Table 2 is the VNS PCA time much longer than in Table 1?

I had to modify the default command suggested in the README; I could not get GraphST or other embeddings except X_pca:

```
python implementation/vns.py -adata_path SS200000128TR_E2.h5ad -k 33 -max_iter 10 -p 12 -m 15 -alpha 1 -seed 1
```

It results in out-of-memory on my laptop!
It took more than 50GB to run on the large field mouse brain hemisphere E dataset with the following result of 19125.7 (far from the reported solution of 9550.0 in Table 1)

Output:

```
Number of clusters: 33
Number of cells: 38811
max_iter, parameter, m, alpha, s, seed = [10, 12, 15, 1, 10, 1]
Best solution:
[3630,328,33812,3751,37710,34437,6667,7963,18252,18502,34176,36587,25559,16906,19028,28821,15541,19642,22890,25378,37882,29104,6406,16972,34770,7429,6366,6142,1938,11456,71,36448,3351]
Best objective function: 19125.722361663233
Elapsed time: 83.8853 seconds
```

For the same dataset, selecting manually a single best centroid (K=1) results in an overall score of 24878.77. The sum of n-K second minima gives a lower bound of 8433.62.