
Computational Genomics

Project 2 - status

Team:
Agata Kaczmarek, Władysław Olejnik, Mateusz Stączek

GitHub: https://github.com/kaczmareka/CompGen_project2

Chosen approach

- Data
- Graph representation of interactions
- Arrowhead - ground truth
- Evaluation metrics
- Results

Datasets

- Requirements:
 - Small, below 0.5GB each
 - When loaded to Juicebox, should have easily visible target squares
 - From Hi-C experiments
- First:
 - <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE226216>
 - GSE226216_HMEC_Res10_20_40_100_500kb.hic
 - human mammary epithelial cells (HMECs)
- Other files:
 - GSE226216_Huh1_Inter_Intra_Res10_20_40_100_500kb
 - GSE226216_SNU449_Inter_Intra_Res10_20_40_100_500kb
 - Both are hepatocellular carcinoma cell lines

The screenshot shows the NCBI GEO Accession Display page for GSE226216. The page header includes the NCBI logo and the GEO logo (Gene Expression Omnibus). The main content area displays the accession number GSE226216 and provides a link to the data. The page also includes a search bar and navigation links.

NCBI > GEO > **Accession Display** [?](#)

GEO help: Mouse over screen elements for information.

Scope: Format: Amount: GEO accession:

Series GSE226216 [Query DataSets for GSE226216](#)

Status	Public on Jun 22, 2023
Title	Cell Line-Specific Features of 3D Chromatin Organization in Carcinoma [Hi-C]
Organism	Homo sapiens
Experiment type	Other
Summary	Liver cancer, particularly hepatocellular carcinoma (HCC), poses a significant health burden worldwide. The study of 3D chromatin organization in liver cancer is essential for understanding the disease's pathogenesis and identifying potential therapeutic targets.

Graph representation of interactions

- Interactions embedded as graph.
- Louvain Community Detection Algorithm
- To enhance performance of algorithm, we tried to erase interactions closest to main diagonal.
- We tried to find best parameters:
 - Available normalization algorithms
 - Best resolution parameter for LCDA
 - Leave diagonal as it is / erase main diagonal / erase main diagonal along with upper and lower diagonals

Arrowhead

- Algorithm which aims at finding contact domains.
- Highly automatic - you need to run three lines of code, including downloading the data and package.
- Output file contains information e.g. about:
 - Chromosome with the domain
 - Coordinates of the corner point of the domain
- The output from the algorithm is not perfect - as humans we would put some of the TADs in different places.
- Link to original repository: [Arrowhead · aidenlab/juicer Wiki · GitHub](#)

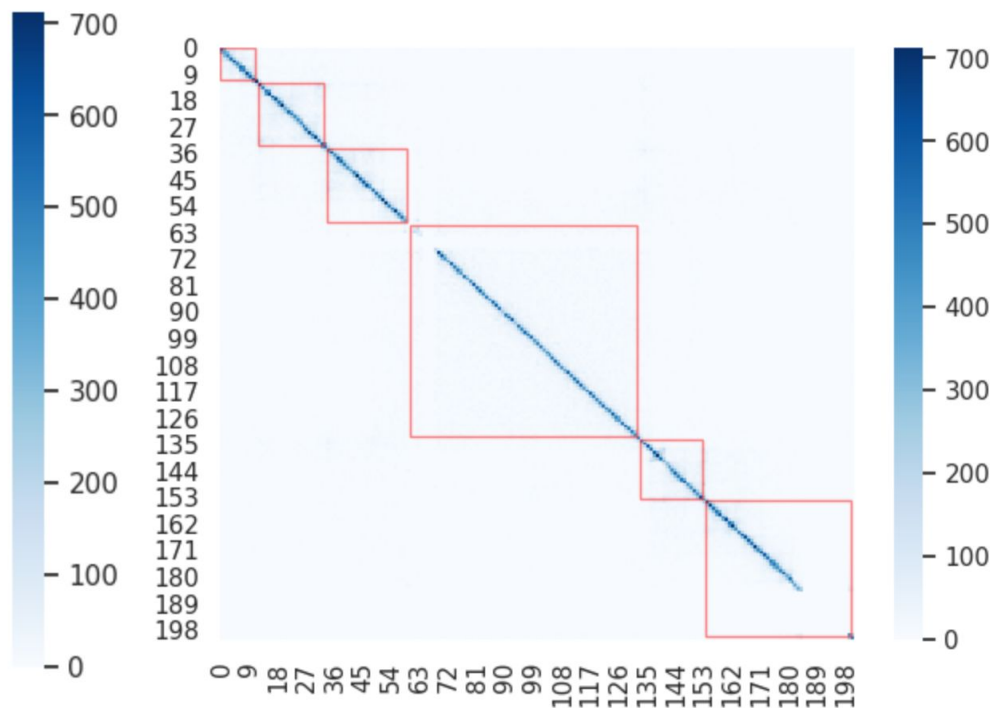
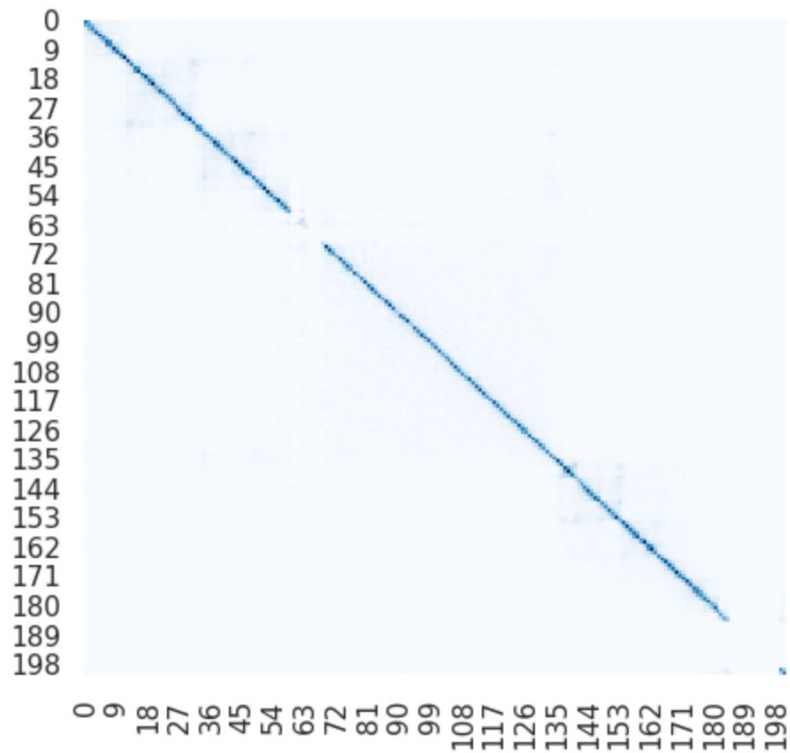
First results for first file only

Analyzed for multiple metrics and
settings of our algorithm

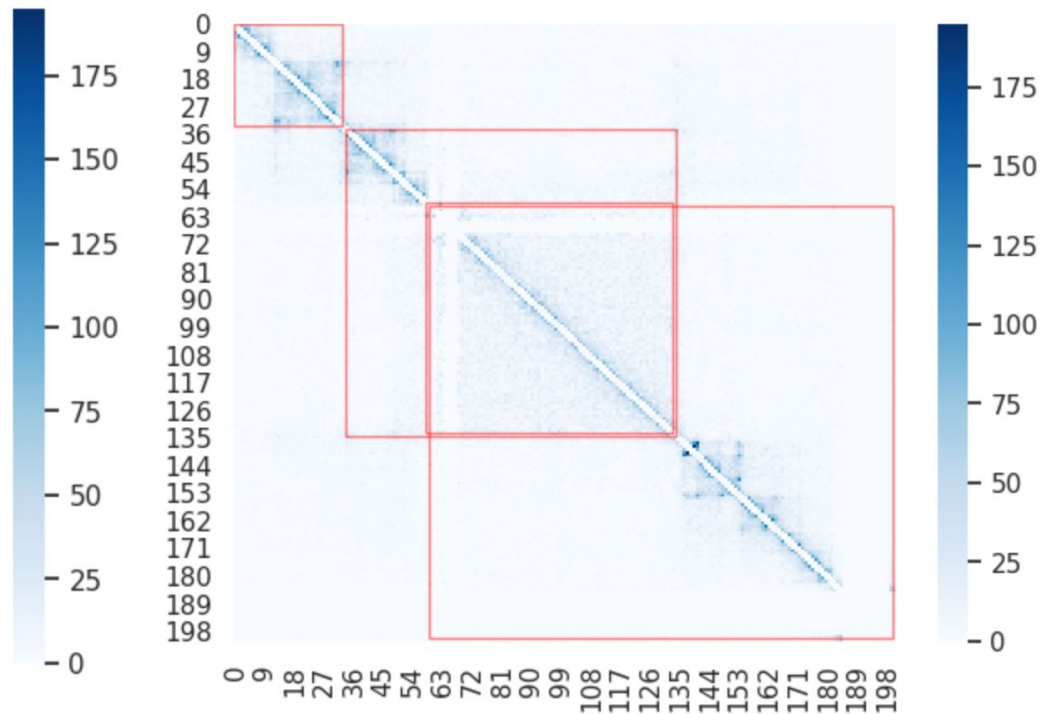
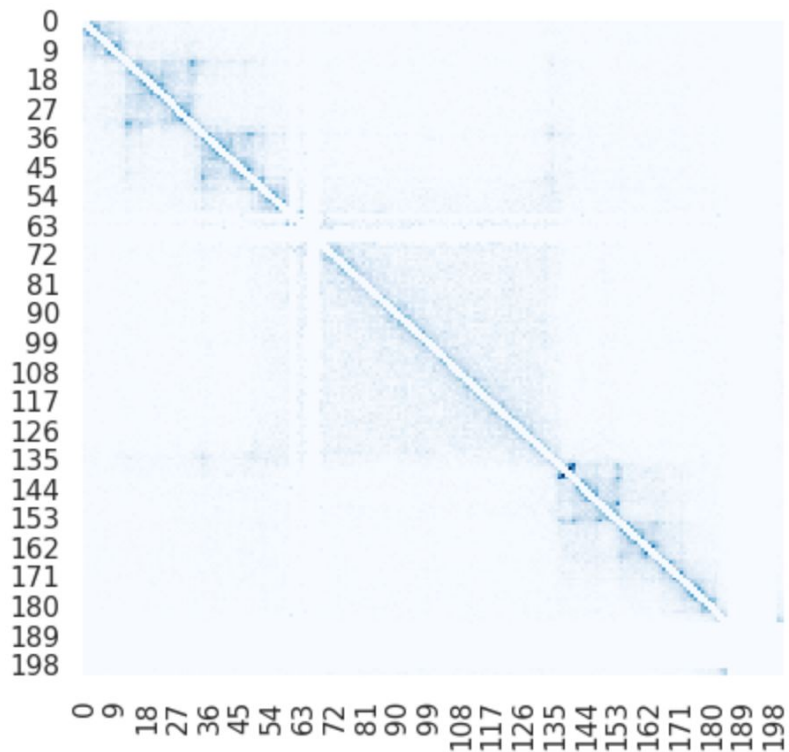
Evaluation metrics

- The ground truth for our measures is the results we got from Arrowhead.
- Confusion matrix and calculated:
 - Accuracy
 - Precision
 - Recall
 - F1
 - Balanced Accuracy
- Histogram of detected TADs sizes - comparison between our method and Arrowhead.

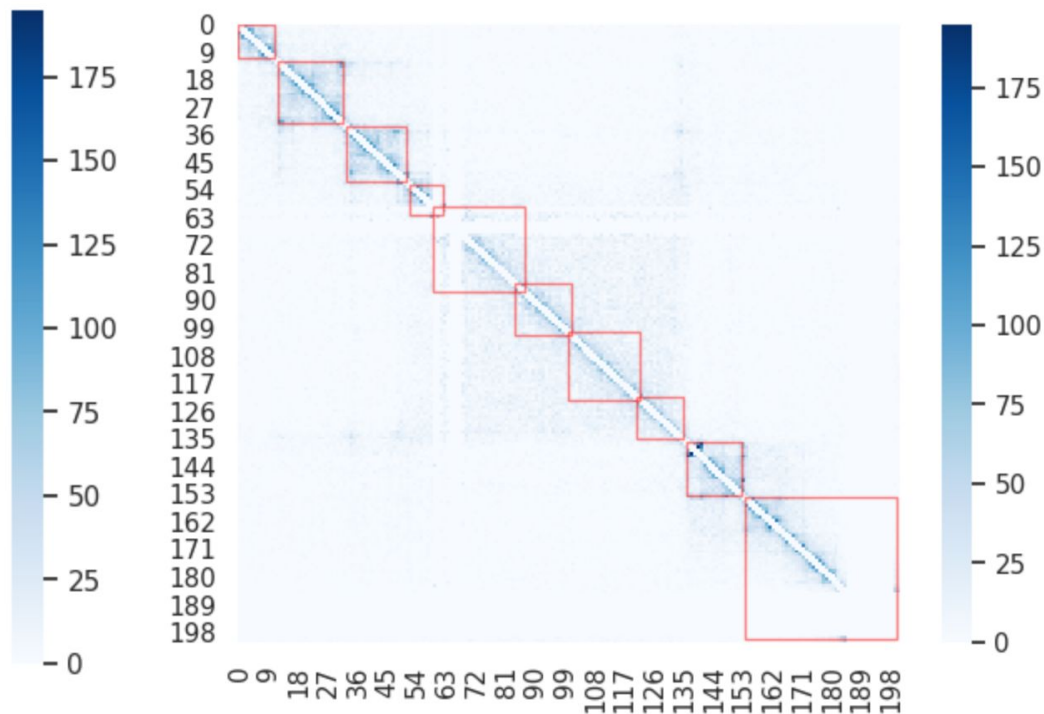
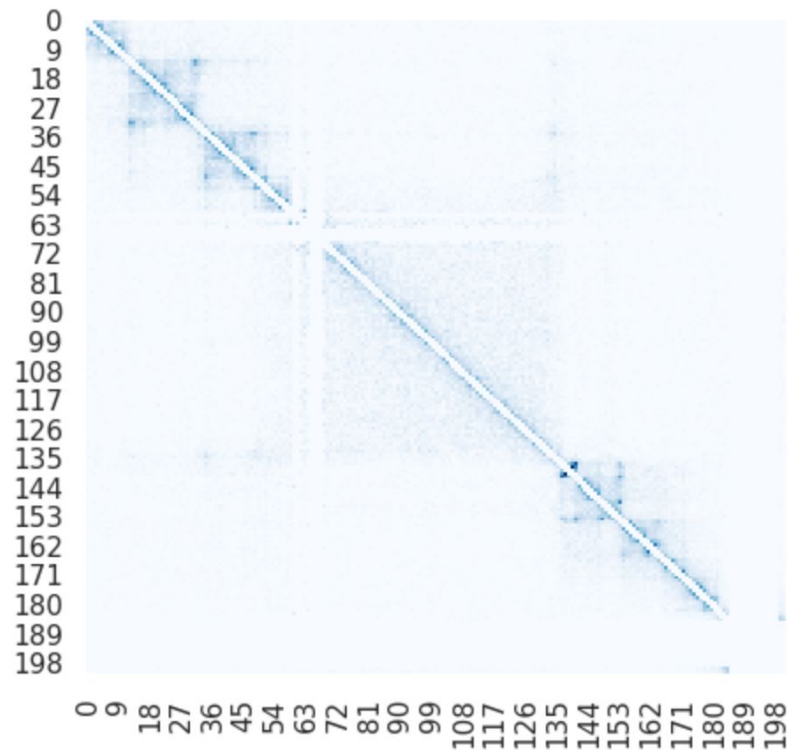
Example - normalization | w/o erasing diagonal | res=1



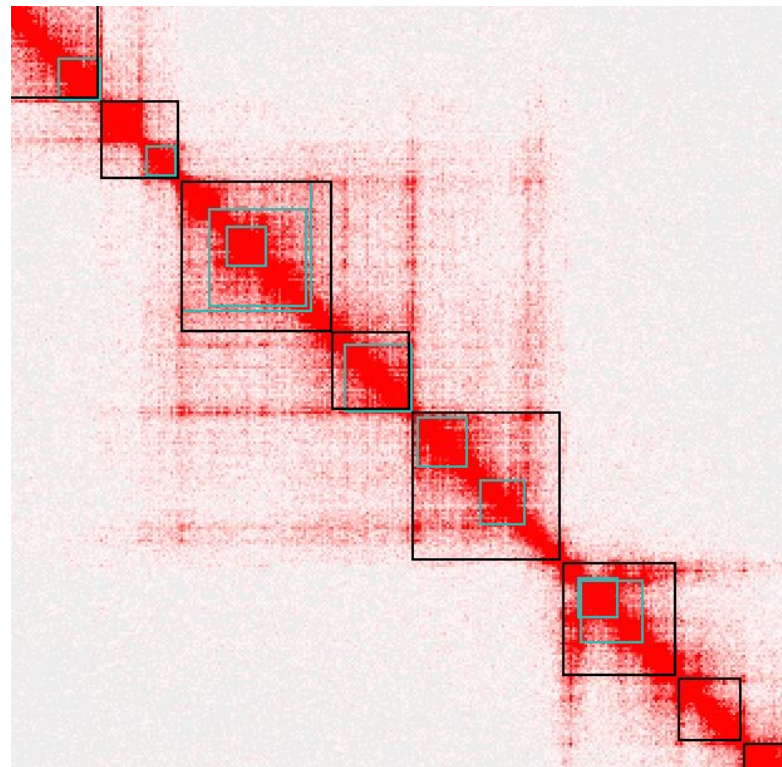
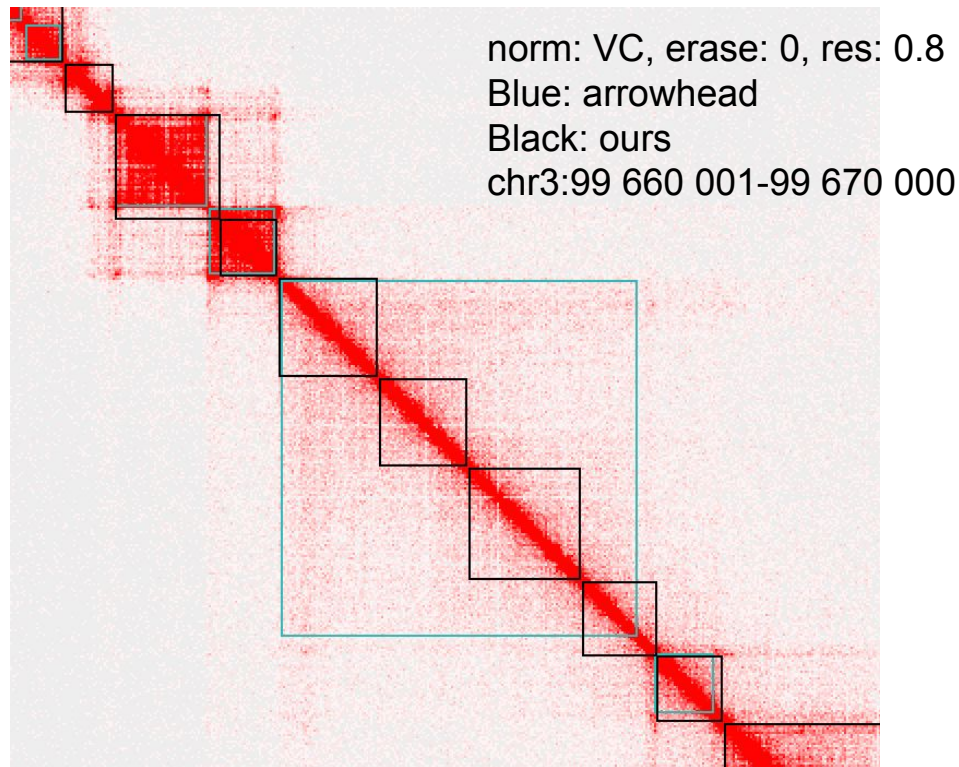
Example - normalization | erasing diagonal | res=1



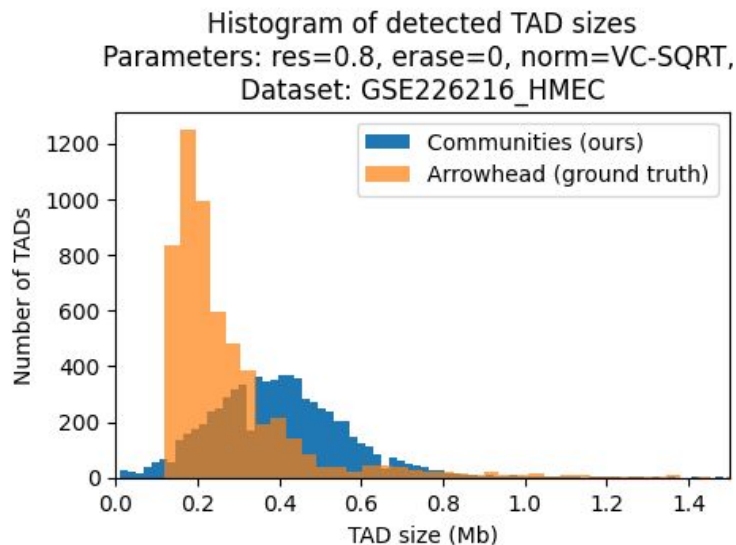
Example - normalization | erasing diagonal | res=2



Example - normalization | w/o erasing diagonal | res=0.8



Results from first dataset



```
confusion_matrix / all_datapoints
```

✓ 0.0s

	Predicted TAD	Predicted non-TAD
Arrowhead TAD	0.489836	0.003116
Arrowhead non-TAD	0.428607	0.078442

```
metrics_df
```

✓ 0.0s

	Accuracy	Precision	Recall	F1	Balanced accuracy
0	0.5683	0.5333	0.9937	0.6941	0.5742

Final results

Analyzed for multiple files

- only specific metrics
- one setting of our algorithm
- added new metric - coverage

Among applied transformations, we used VC_SQRT normalization and erasing values close to the diagonal.

New metrics

Algorithm:

- for each chromosome:
 - create a vector of ones and zeros, with ones where TADs were detected
 - multiply element-wise the vectors from 2 sources: our results and Arrowhead results
 - for each of the source files:
 - for each community:
 - calculate the fraction of ones from this community that are in the product vector
 - add it to the current sum of products
 - divide the total sum of products by the number of communities in the source file

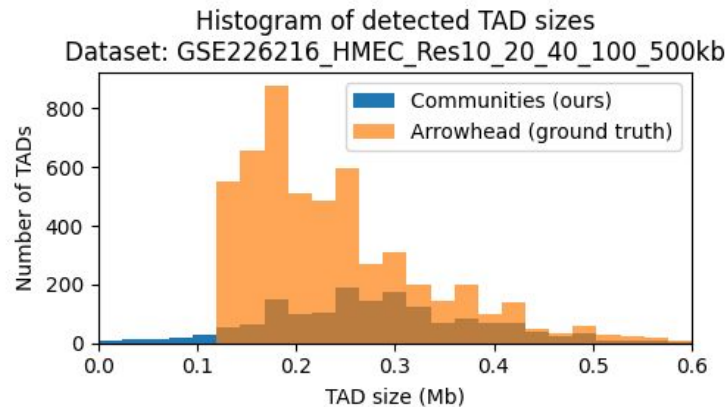
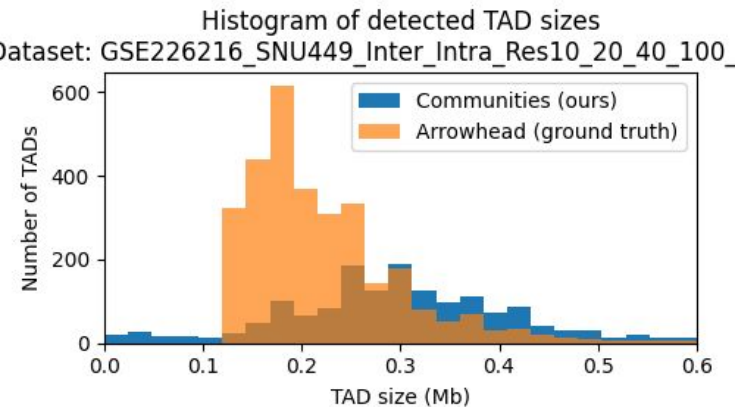
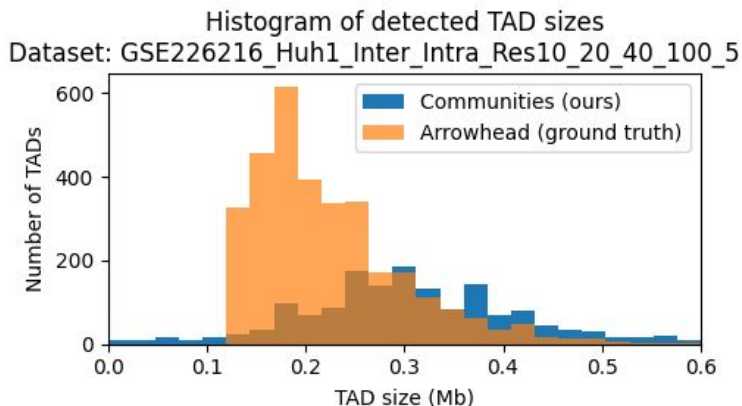
This way we obtain a metric that calculates the average fraction of elements from each community that are also in another results file.

Results

Filename	Balanced accuracy	Jaccard index	Average match for our communities	Average match for arrowhead
GSE226216_HMEC_Res10_20_40_100_500kb	0.4982	0.0352	0.14246313461285748	0.03465270531999266
GSE226216_Huh1_Inter_Intra_Res10_20_40_100_500kb	0.4987	0.0375	0.10710274295095695	0.04005569943021708
GSE226216_SNU449_Inter_Intra_Res10_20_40_100_500kb	0.4988	0.0391	0.11061192731669447	0.042512549879933546

Results presented above are for all 3 files. Unfortunately, in all cases, our algorithm marked much fewer TADs than Arrowhead. Our new metric shows that our results rarely overlap with Arrowhead.

Distributions of TAD sizes



Conclusions

Our implementation:

- marks TADs based on community detection algorithm,
- obtains different results from Arrowhead ground truth.

Based on comparison between our results and Arrowhead on 3 files:

- TADs marked by our algorithm have similar sizes to those marked by Arrowhead, but we found either more or much fewer than correct.

In the future, we could fine tune more parameters, and try preprocessing too.

Thank you!

— Any questions/ comments? —
