PanOriginSV

Group 7

Flowchart

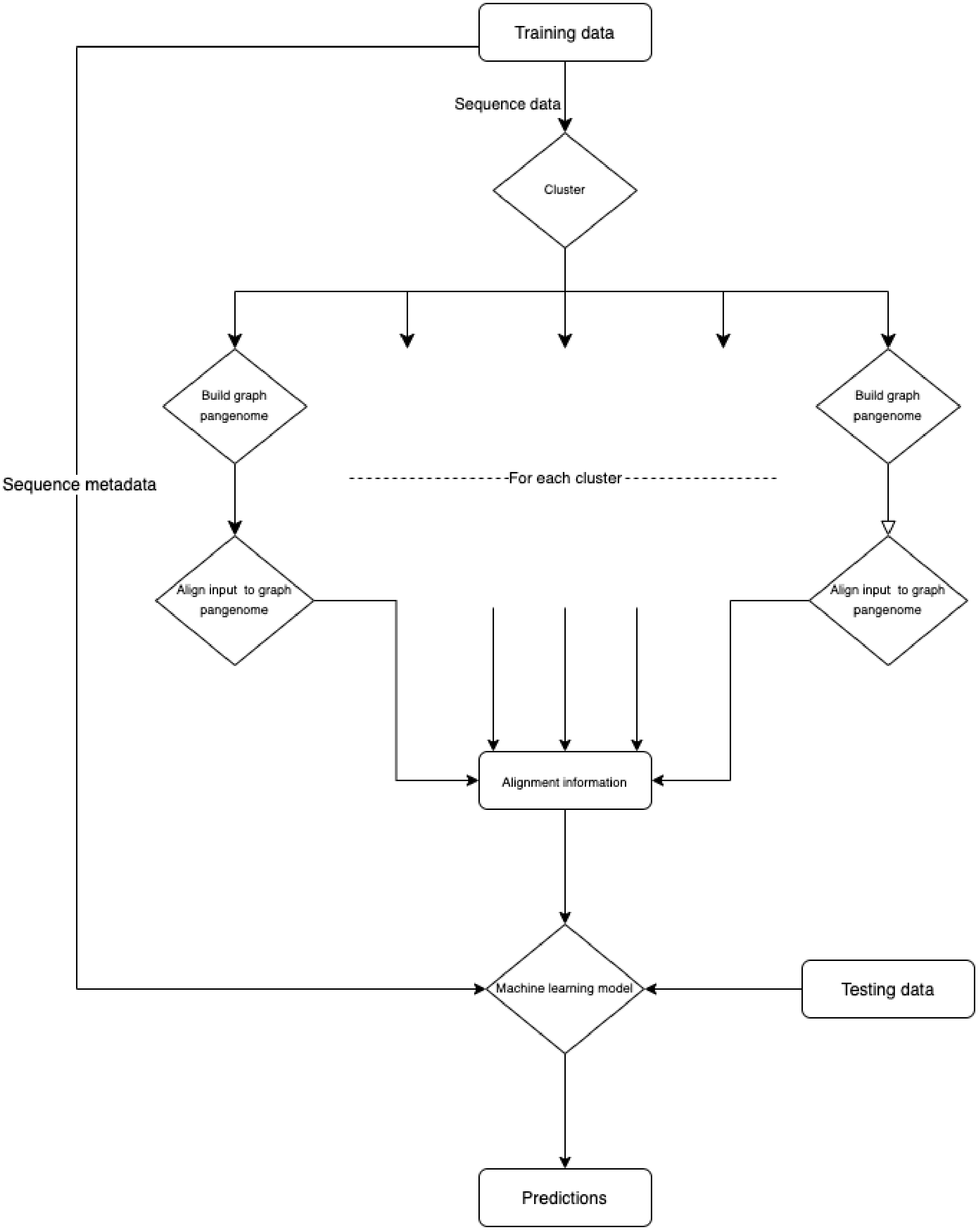
1. Cluster similar training sequences using MMSEQ2

2. For each cluster, create a graph pangenome that incorporates SV information

3. Align the sequences in each cluster to its corresponding pangenome graph
    - Alignment id%, order of alignment, copy number
    - GraphAligner/Minigraph

4. Use Alignment information and metadata as features to train ML models

5. Predict and benchmark on GEAC dataset



Clustering: MMSEQ2

- mmseqs with `--min-seq-id 0.8 -c 0.8 --cov-mode 1 -s 7.0`
- 18000 clusters (originally 60,000 sequences)
  - 11000 of the clusters are singletons
- Top 10 clusters are above 200 sequences each
- Within a cluster, average nucleotide identity (using FastANI) ranges from 80% to 100%

Benchmarking Linear Pangenomes

- For each cluster, we split into training and testing sequences
- Using the training sequences, we construct a linear pangenome using Plaster and train a Random Forest model on the training sequence alignments
- We then align the test sequences to the pangenome and use the Random Forest model to predict the true lab

Cluster	Labs	Sequences	Train accuracy	Test accuracy	Top 5 test accuracy
O3GQU	131	1440	0.55	0.49	0.76
BK5PO	51	421	0.96	0.78	0.95
B6SZW	39	400	0.94	0.74	0.92
C46EW	34	220	0.90	0.73	0.89
O0O60	16	243	0.98	0.96	1.00

Graph Pangenome Construction

- Minigraph fails to create pangenome
- MetaPGN works on a gene-level basis (and we have not been able to successfully run yet)
- PPanGGOLiN also works on a gene-level basis and requires the input genomes to be annotated
- VG tools require an input .vcf file, which means we need to add a structural variant caller to the pipeline