GNN FOR GENOME RECOVERY

...metagenomics and deep learning

Metagenomics

- Study DNA sequencing of microbial communities
- Genome - Collection of all its genetic information, represented in the form of a sequence of DNA base (ATCG)
- Genomes are long
  - 13 million base pairs
- Technological limitations limit complete sequencing of genomes
- Read: Random sequence from sample
  - CGATCTTA
- State-of-the-art read: max 2-30k bases
  - longer --> more error
- String reconstruction

String reconstruction

- Set of large reads (thanks to high through-put sequencing technologies)
- Try to reconstruct the original string (genome) from these reads
    - Example/Analogy: A pile of (same) newspaper bundled together
    - It exploded and we try to reconstruct the newspaper content from the bits
- Idea: Overlapping bits can be pieced together (somewhat like jigsaw)
- Contigs: Overlapping reads forming long contigous sequences
- Relatively longer subsequence of the genome
- Still need to piece them together
- NP-complete

Approach

- ASSUMPTION: Reads from only a single Genome
- Divide each contig into fixed size (k-mer)
    - CGATCTTA: CGA, GAT, ATC, TCT, CTT, TTA (k=3)
- Consider k-mers as nodes
    - Group all same k-mers into one node
- add edge if (k-1) bases overlap
- De-Bruijin Graph
- Every Eulerian walk gives a possible genome sequence

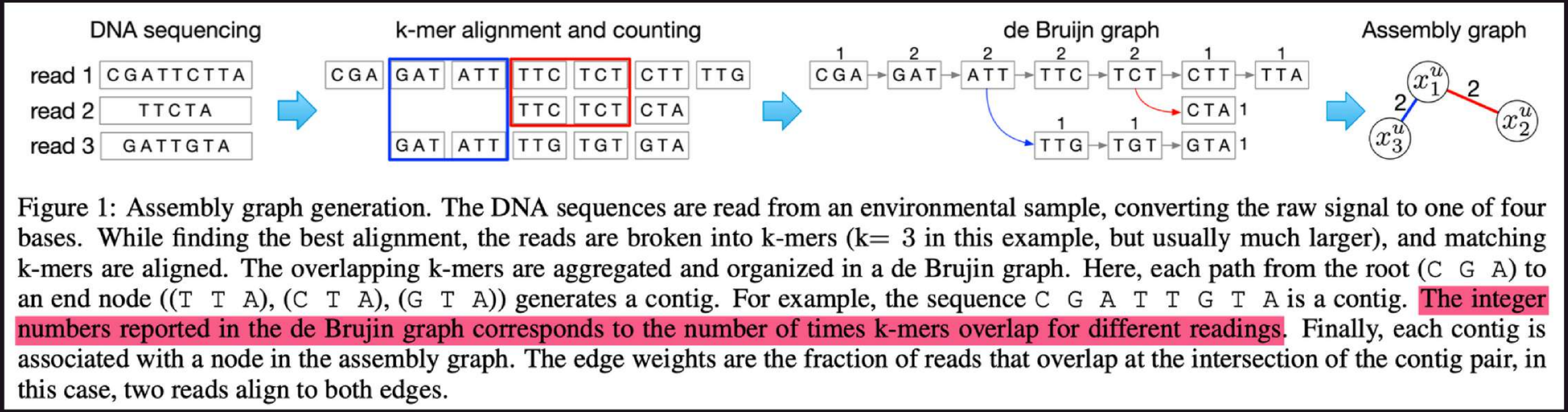


Figure 1: Assembly graph generation. The DNA sequences are read from an environmental sample, converting the raw signal to one of four bases. While finding the best alignment, the reads are broken into k-mers (k= 3 in this example, but usually much larger), and matching k-mers are aligned. The overlapping k-mers are aggregated and organized in a de Brujin graph. Here, each path from the root (C G A) to an end node ((T T A), (C T A), (G T A)) generates a contig. For example, the sequence C G A T T G T A is a contig. The integer numbers reported in the de Brujin graph corresponds to the number of times k-mers overlap for different readings. Finally, each contig is associated with a node in the assembly graph. The edge weights are the fraction of reads that overlap at the intersection of the contig pair, in this case, two reads align to both edges.

- Figure from paper
- Assembly graphs: just a graph depicting correlation between contigs

Binning contigs

- Binning: Cluster the similar contigs together
- Single Copy Genes (SCGs): occurs only once in the full genome
- Important info!
  - since two contigs with the same SCG must belong to different genomes and should therefore appear in different clusters/bins
- Aim: to partition contigs into bins that contain a single copy of all the genes in the set of SCGs.

Related works

- Read coverage: measure of how many times a specific base in the genome is covered by reads
- Abundance: Embed this into a vector
- MetaBAT2: Based on an empirical posterior probability
    - uses abundance and k-mer comp. to compute a pairwise distance matrix for all contig pairs, calculated with a k-mer frequency distance probability and abundance distance probability
- MaxBin2: uses an Expectation-Maximization algorithm to estimate the probability of a contig belonging to a particular bin
- Most commonly used

Related works contd.

- VAMB: binner based on a variational autoencoder
    - encodes k-mer composition and abundance features in a low dimensional embedding
    - uses this to improve binning (merely an assistive tool)
- GraphBin: uses assembly graph for label propogation
    - only post-processsing, not full binning process

Papers' idea

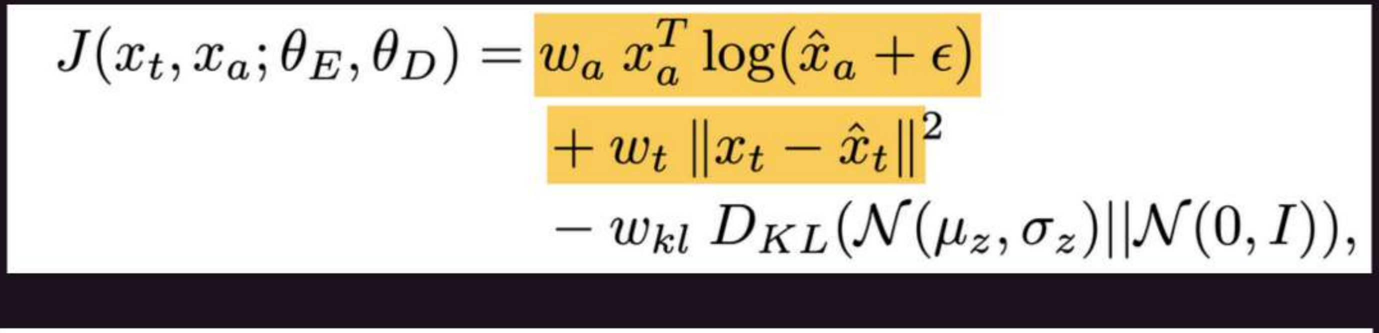
- VAEG-BIN
  - use VAE to encode/learn individual contig representation
  - refine/learning more the representation by feeding the VAE output to GNN input
  - VAE --> local feature, GNN --> Overall structure
- Helps in binning since learning distribution of contigs naturally clusters them

Results

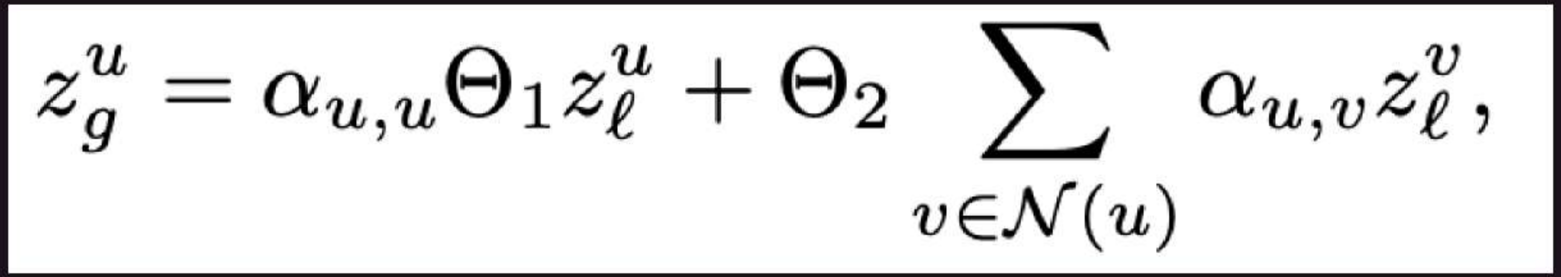
- Compared it with various other state-of-the-art binners
- VAEG-BIN outperforms in simulated and real world dataset
- Conclusion: "leveraging the relational information in the assembly graph, we can significantly increase the number of high-quality genomes recovered during the subsequent binning process as compared to the state-of- the-art baseline methods "

Things to covered

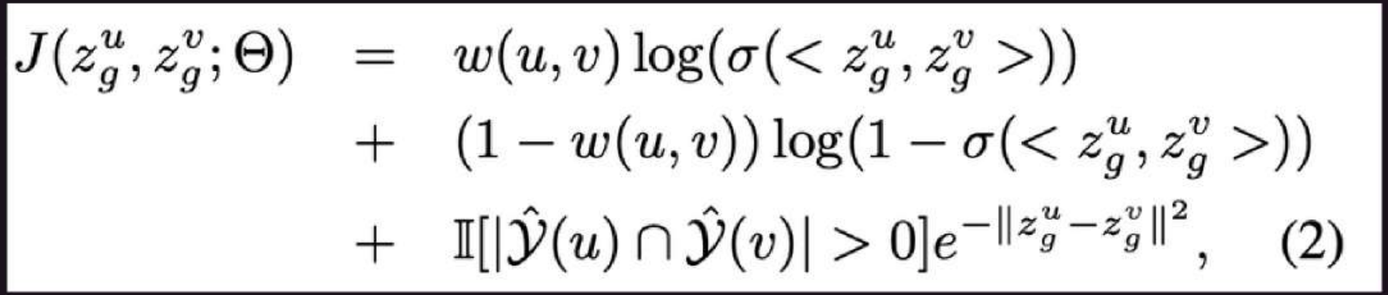
- math



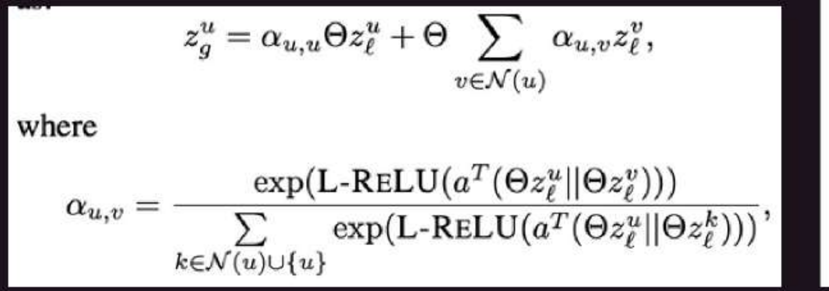
$$\begin{aligned} J(x_t, x_a; \theta_E, \theta_D) = {} & w_a \, x_a^T \log(\hat{x}_a + \epsilon) \\ & + w_t \, \|x_t - \hat{x}_t\|^2 \\ & - w_{kl} \, D_{KL}(\mathcal{N}(\mu_z, \sigma_z) || \mathcal{N}(0, I)), \end{aligned}$$



$$z_g^u = \alpha_{u,u} \Theta_1 z_\ell^u + \Theta_2 \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} z_\ell^v,$$



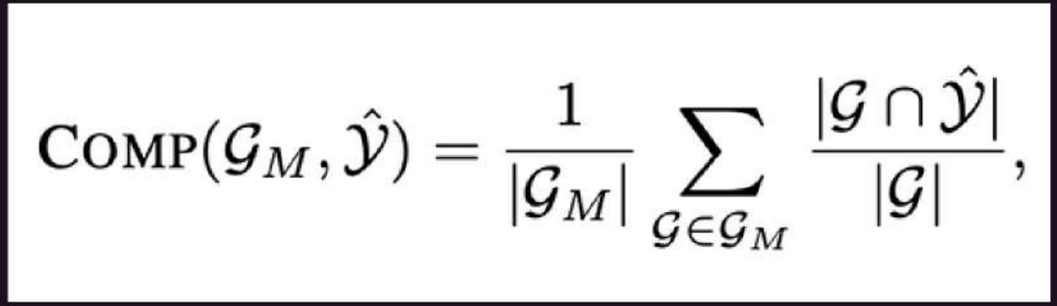
$$\begin{aligned} J(z_g^u, z_g^v; \Theta) \quad = \quad & w(u,v) \log(\sigma(< z_g^u, z_g^v >)) \\ + \quad & (1 - w(u,v)) \log(1 - \sigma(< z_g^u, z_g^v >)) \\ + \quad & \mathbb{I}[|\hat{\mathcal{Y}}(u) \cap \hat{\mathcal{Y}}(v)| > 0] e^{-\|z_g^u - z_g^v\|^2}, \qquad (2) \end{aligned}$$



$$z_g^u = \alpha_{u,u} \Theta z_\ell^u + \Theta \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} z_\ell^v,$$

where

$$\alpha_{u,v} = \frac{\exp(\text{L-ReLU}(a^T(\Theta z_\ell^u || \Theta z_\ell^v)))}{\sum_{k \in \mathcal{N}(u) \cup \{u\}} \exp(\text{L-ReLU}(a^T(\Theta z_\ell^u || \Theta z_\ell^k)))},$$



$$\textsc{Comp}(\mathcal{G}_M, \hat{\mathcal{Y}}) = \frac{1}{|\mathcal{G}_M|} \sum_{\mathcal{G} \in \mathcal{G}_M} \frac{|\mathcal{G} \cap \hat{\mathcal{Y}}|}{|\mathcal{G}|},$$

- implementation/verification of results

Thank you!