deBGR: An Efficient and Near-Exact Representation of the Weighted de Bruijn Graph

Prashant Pandey\textsuperscript{1}, Michael A. Bender\textsuperscript{1}, Rob Johnson\textsuperscript{12}, and Rob Patro\textsuperscript{1}

\textsuperscript{1}Stony Brook University, NY USA, \textsuperscript{2}VMWare Inc USA
de Bruijn graphs are ubiquitous

A de Bruijn graph is the data representation at the heart of a lot of sequence analyses.
de Bruijn graph (dBG)

A **read** is a string of bases over the DNA alphabet A, C, T, and G.

A **k-mer** is a substring of length k. Here, k is 5.
de Bruijn graph (dBG)

Read: ….CAAAAA…. 

Prefix (k-1)-mer  

Suffix (k-1)-mer

An edge is a k-mer connecting its two k-1 substrings.
de Bruijn graph (dBG)

Read 1: ….CAAAAAT….  
Read 2: ….CAAAAC….
Weighted de Bruijn graphs

- Topology-only de Bruijn graphs are not adequate for transcriptome assembly.

- Abundance information of each k-mer is critical for transcriptome assembly.
Weighted de Bruijn graphs

- Topology-only de Bruijn graphs are not adequate for transcriptome assembly.
- Abundance information of each k-mer is critical for transcriptome assembly.

Weighted de Bruijn graphs pose an extra obligation and opportunity.
A weighted de Bruijn graph associates each edge (k-mer) its abundance in the underlying dataset.
Measuring dBG representation

de Bruijn graphs store only k-mers, memory usage scales with the number of unique k-mers.

- **Human genome (few Billion k-mers):** >100 GB
- **Soil metagenomes (few Million species):** Few TBs

Beefy server machines are needed to perform weighted de Bruijn graph analysis.
WdBG as a multiset

<table>
<thead>
<tr>
<th>MultiSet</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TCCG, 2</td>
<td></td>
</tr>
<tr>
<td>CCGC, 9</td>
<td></td>
</tr>
<tr>
<td>CCGA, 6</td>
<td></td>
</tr>
<tr>
<td>CGCT, 5</td>
<td></td>
</tr>
<tr>
<td>AGCT, 2</td>
<td></td>
</tr>
</tbody>
</table>
Past work on Probabilistic dBG representation

- **Pell et al. 2012**: Represented dBG using a Bloom filter.
Past work on Probabilistic dBG representation

• **Pell et al. 2012**: Represented dBG using a Bloom filter.

• **Pellow et al. 2016**: Showed how to exploit redundancy in k-mers to reduce the false-positive rate of the Bloom filter without increasing the space.
Past work on Probabilistic dBG representation

- **Pell et al. 2012**: Represented dBG using a Bloom filter.

- **Pellow et al. 2016**: Showed how to exploit redundancy in k-mers to reduce the false-positive rate of the Bloom filter without increasing the space.

- **Chikhi and Rikz 2013 and Salikhov et al. 2013**: They showed how to convert a probabilistic representation into an exact one using a small and exact auxiliary data structure.
A counting filter is a lossy representation of a *multiset*. Operations: inserts, count, and delete. Generalizes AMQs. False positives $\approx$ over-counts. Counting quotient filter [Pandey et al. 2017]
Probabilistic weighted de Bruijn graph

[Counting quotient filter]

TCCG, 4
CCGC, 9
CCGA, 6
CGCT, 5
AGCT, 4
GAGC, 2
CGAG, 1
GAGT, 1
AAAA, 1

[Abundance error]

[Topological errors]
This paper: deBGR

• An **exact representation** of the weighted de Bruijn graph.

• An algorithm that uses counts in the approximate representation in an AMQ to iteratively **self-correct approximation errors**.

• It corrects both kinds of errors, **abundance and topological errors** and supports **membership queries**.

• It **supports deletion** of k-mers from the structure.

• It takes 18-28% more space than the approximate representation and has **no errors**.
A weighted de Bruijn graph invariant

Read 1: ….CAAAAAT….  
Read 2: ….CAAAAAC…..

Total incoming abundance = Total outgoing abundance
A weighted de Bruijn graph invariant

Total incoming abundance = Total outgoing abundance*

*After accounting for read starts and ends.
WdBG representation in deBGR

Read 1: CAAAAAT
Read 2: CAAAAAC

<table>
<thead>
<tr>
<th>Edge</th>
<th>Abundance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAAAA</td>
<td>2</td>
</tr>
<tr>
<td>AAAAT</td>
<td>1</td>
</tr>
<tr>
<td>AAAAC</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>Start reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAAA</td>
<td>2</td>
</tr>
<tr>
<td>CAAAA</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>End reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAT</td>
<td>1</td>
</tr>
<tr>
<td>AAAC</td>
<td>1</td>
</tr>
</tbody>
</table>

Diagram showing nodes and edges with read abundance and counts.
WdBG representation in deBGR

<table>
<thead>
<tr>
<th>Edge</th>
<th>Abundance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAAAA</td>
<td>2</td>
</tr>
<tr>
<td>AAAAT</td>
<td>2</td>
</tr>
<tr>
<td>AAAAC</td>
<td>1</td>
</tr>
<tr>
<td>CCGTA</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>Start reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAAA</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>End reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>1</td>
</tr>
<tr>
<td>AAAC</td>
<td>1</td>
</tr>
</tbody>
</table>

- **CAAA**
  - Start reads: 2
  - End reads: 0

- **CCGT**
  - Start reads: 0
  - End reads: 0

- **CGTA**
  - Start reads: 0
  - End reads: 0

- **AAAT**
  - Start reads: 0
  - End reads: 1

- **AAAA**
  - Start reads: 0
  - End reads: 0

- **AAAC**
  - Start reads: 0
  - End reads: 1
Error correction

**CCGT**
- Start reads: 0
- End reads: 0

**CGTA**
- Start reads: 0
- End reads: 0

**AAAAT**
- Start reads: 0
- End reads: 1

**AAAAC**
- Start reads: 0
- End reads: 1

**CCGT, 1**

**AAAAT, 2**

**AAAAC, 1**

**CCAAAA, 2**

**AAAA**
- Start reads: 0
- End reads: 0

**AAAT**
- Start reads: 0
- End reads: 1

**AAAC**
- Start reads: 0
- End reads: 1
Error correction

- **CCGT**
  - Start reads: 0
  - End reads: 0

- **AAAA**
  - Start reads: 0
  - End reads: 0

- **AAAT**
  - Start reads: 0
  - End reads: 1

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **AAAAA**
  - Start reads: 0
  - End reads: 0

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **CCGT**
  - Start reads: 0
  - End reads: 0

- **AAAAT**
  - Start reads: 0
  - End reads: 1

- **AAAAC**
  - Start reads: 0
  - End reads: 1

- **CAAA**
  - Start reads: 2
  - End reads: 0

- **CAAAA**
  - Start reads: 2
  - End reads: 0

- **AAAT**
  - Start reads: 0
  - End reads: 1
Error correction

- **CCGT**
  - Start reads: 0
  - End reads: 0

- **CGTA**
  - Start reads: 0
  - End reads: 0

- **AAAT**
  - Start reads: 0
  - End reads: 1

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **AAAA**
  - Start reads: 0
  - End reads: 0

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **AAAAA**
  - Start reads: 2
  - End reads: 0

- **AAAAT**
  - Start reads: 0
  - End reads: 1

- **AAAAC**
  - Start reads: 0
  - End reads: 1

- **AAAA**
  - Start reads: 0
  - End reads: 0
Error correction

- **CCGT**
  - Start reads: 0
  - End reads: 0

- **CGTA**
  - Start reads: 0
  - End reads: 0

- **AAAT**
  - Start reads: 0
  - End reads: 1

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **CAAA**
  - Start reads: 2
  - End reads: 0

- **AAAA**
  - Start reads: 0
  - End reads: 0

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **AAAAT**
  - Start reads: 0
  - End reads: 1

- **CAAAA**
  - Start reads: 2
  - End reads: 0
Error correction

- **CCGT**
  - Start reads: 0
  - End reads: 0

- **CGTA**, +0

  - **CCGTA, +0**
  - **CGTA**
    - Start reads: 0
    - End reads: 0

- **AAAT**
  - Start reads: 0
  - End reads: 1

- **AAAA**
  - Start reads: 0
  - End reads: 0

- **AAAC, 1**

  - **AAAT, 2**
  - **AAAAA, 2**
  - **AAAAA, 2**

- **AAAC, 1**

- **AAAT, 2**

**CAAA**
- Start reads: 2
- End reads: 0

**AAAA**
- Start reads: 0
- End reads: 0

**AAAC**
- Start reads: 0
- End reads: 1
Error correction

**CCGT**
Start reads: 0
End reads: 0

**CCGTA, + 0**

**CGTA**
Start reads: 0
End reads: 0

**AAAAT, 2**

**AAAA**
Start reads: 0
End reads: 0

**AAAT**
Start reads: 0
End reads: 1

**AAAAC, 1**

**AAAC**
Start reads: 0
End reads: 1

**CAAAC, 2**

**CAAA**
Start reads: 2
End reads: 0

**CAAAA, 2**

**AAAAA, 2**
Error correction

- **CCGT**
  - Start reads: 0
  - End reads: 0
  - Transition to **CGTA**

- **CGTA**
  - Start reads: 0
  - End reads: 0
  - Transition to **AAAT**

- **AAAT**
  - Start reads: 0
  - End reads: 1

- **CAAA**
  - Start reads: 2
  - End reads: 0
  - Transition to **AAAA**

- **AAAA**
  - Start reads: 0
  - End reads: 0
  - Transition to **AAAC**

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **AAAC**
  - Start reads: 0
  - End reads: 1

Error correction process and transitions between sequences.
Error correction

- **CCGT**
  - Start reads: 0
  - End reads: 0
  - Transition to: CGT

- **CGTA**
  - Start reads: 0
  - End reads: 0
  - Transition to: AAAT

- **AAAT**
  - Start reads: 0
  - End reads: 1
  - Transition from: CGTA

- **CCGTA, +0**

- **AAAA**
  - Start reads: 0
  - End reads: 0
  - Transition to: AAAAC, 1

- **AAAAAC, 1**

- **AAAAC, 1**

- **AAAAT, 2**

- **AAAC**
  - Start reads: 0
  - End reads: 1
  - Transition from: AAAAT, 2

- **AAAAA, 2**

- **CAAA**
  - Start reads: 2
  - End reads: 0
  - Transition to: AAAAA, 2

- **AAAAC, 1**

- **AAAAT, 2**

- **AAAC**
  - Start reads: 0
  - End reads: 1
  - Transition from: AAAAC, 1
Error correction

- **CCGT**
  - Start reads: 0
  - End reads: 0

- **AAAA**
  - Start reads: 0
  - End reads: 0

- **AAAT**
  - Start reads: 0
  - End reads: 1

- **AAAC**
  - Start reads: 0
  - End reads: 1

- **CCGTA**
  - Start reads: 0
  - End reads: 0

- **CGTA**
  - Start reads: 0
  - End reads: 0

- **AAAAC**
  - Start reads: 0
  - End reads: 1

- **AAAT, 2**
  - End reads: 1

- **AAAA, 2**
  - End reads: 2

- **CCGTA, + 0**

- **AAAAA, 2**
  - End reads: 2

- **AAAC, 1**
  - End reads: 1
Error correction algorithm

• We use a standard work queue algorithm.

• We bootstrap with a set $C$ of edges for which we know the abundance is correct.

• We then expand the set $C$ of edges using the weighted de Bruijn graph invariant.

• Please refer to the paper for exact set of rules for error correction.

• Running time: $O(n \cdot \log(n)/\log(1/4\varepsilon))$. 
## Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>#k-mer instances</th>
<th>#Distinct k-mers</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM984609</td>
<td>26 GB</td>
<td>19,662,773,330</td>
<td>1,146,347,598</td>
</tr>
<tr>
<td>GSM981256</td>
<td>22 GB</td>
<td>16,470,774,825</td>
<td>1,118,090,824</td>
</tr>
<tr>
<td>GSM981244</td>
<td>43 GB</td>
<td>37,897,872,977</td>
<td>1,404,643,983</td>
</tr>
<tr>
<td>SRR1284895</td>
<td>33 GB</td>
<td>26,235,129,875</td>
<td>2,079,889,717</td>
</tr>
</tbody>
</table>
Space vs Accuracy

Datasets:
- GSM984609
- GSM981256
- GSM981244
- SRR1284895

Comparing Squeakr, Squeakr (exact), and deBGR.
Space vs Accuracy

Datasets
- GSM984609
- GSM981256
- GSM981244
- SRR1284895

Bits/k-mer
- Squeakr: 16,655,318
- Squeakr (exact): 15,864,754
- deBGR: 12,257,261
- deBGR: 27,200,821
Space vs Accuracy

Datasets

- GSM984609
- GSM981256
- GSM981244
- SRR1284895

Squeakr
- 16,655,318

Squeakr (exact)
- 15,864,754
- 12,257,261

deBGR
- 27,200,821

Number of errors in deBGR: 0!
Conclusion

• Abundance information is important for many data analyses.

• But abundance information is also useful for providing higher de Bruijn graph structural guarantees.

• We show that the abundance information can be used to remove effectively all the errors in an approximate weighted de Bruijn graph representation.

https://github.com/splatlab/debgr