Earmark graph approach to *de novo* genome assembly

M. Dvorkin, A. Kulikov

June 1, 2011

Abstract

A common approach to assembling a genome from short reads is constructing the de Bruijn graph on all k-mers from the given set of reads and finding a traversal of edges in this graph. We propose a new approach that allows to decrease the graph size without losing the essential information from the input data. Instead of using all the k-mers from a read we take only a few of them (and call them earmarked). Besides an obvious advantage of requiring less memory and time for constructing, the resulting earmark graph has several other advantages over the de Bruijn graph. We discuss them in the paper and also present some experimental results.

Contents

1 Introduction **1**

2 Earmark graph **4**

3 Earmark graph construction **5**
3.1 Earmarks selection procedure 5
3.2 Tip extension procedure 5

4 Practical results **7**

5 Discussion and further ideas **7**
5.1 Advantages of earmark graph over de Bruijn graph 7
5.2 A more careful earmarks selection 7

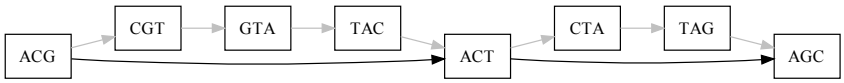
1 Introduction

A common approach to assembling a genome from short reads is constructing the de Bruijn graph [**?**] on all k-mers from the given set of reads and finding a traversal of edges in this graph. An example of the de Bruijn graph for $k = 3$ and a set of all reads of length 6 of a (circular) genome $g =$ `ATGCATTGCACTGCA` is given in Fig. 1a (edges multiplicities are not shown). The genome spells an Eulerian cycle in the de Bruijn graph.

explain why do we mark hash-values but nor k-mers

The size of the de Bruijn graph for most genomes is huge making it difficult to process. We propose a new approach that allows to decrease the graph size without losing the essential information from the input data. Instead of using all the k-mers from a read we take only some fraction of them (and call them *earmarked*). Namely, instead of representing each read as a sequence of edges between its consecutive k-mers (in which case a read of length r defines $r - k - 1$ edges) we represent it as just one (or a few, in general) edge between some of its k-mers. For example, for reads `ACGTACT` and `TACTAGC` and $k = 3$ instead of all gray edges in the figure below we will have only two black edges joining earmarked k-mers `ACG`, `ACT`, and `AGC`.

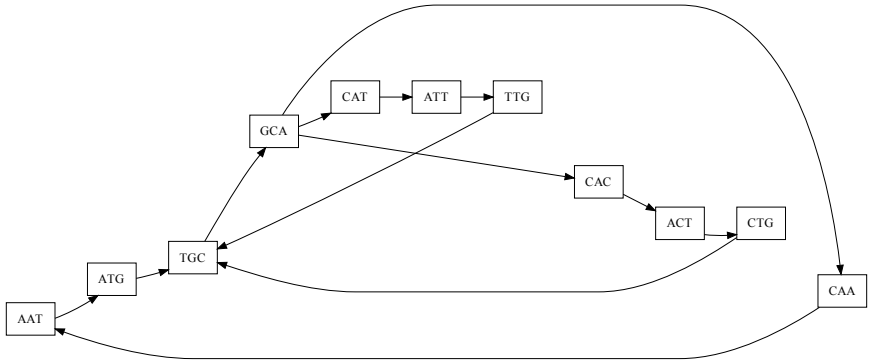
redraw pictues in tikz in the final version



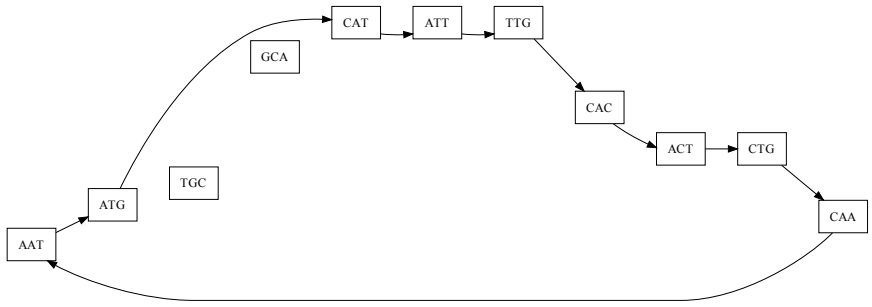
We call the resulting graph built on earmarked k-mers *an earmark graph*. It is a special case of the A-Bruijn graph introduced by Pevzner, Tang, and Tesler [**?**]. An example of the earmark graph for the same genome $g =$ `ATGCATTGCACTGCA` is shown in Fig. 1b. Here, all 3-mers are earmarked except for `TGC` and `GCA`. Besides an obvious advantage of requiring less memory and time for constructing, the resulting earmark graph has several other advantages over the de Bruijn graph. We discuss them after all the necessary definitions.

Our approach is inspired by the work by Roberts et al. [**?**] on sequence comparison. The problem they studied is a pairwise comparison of a set of strings. One of the approaches to this problem is the seed-and-extend approach. One first stores the set of all possible k-mers from a given set of strings. Then, only pairs of strings that both have the same k-mer as a

Figure 1: (a) The de Bruijn graph built on 3-mers of all 6-reads of a toy genome `ATGCATTGCACTGCA`. (b) The earmark graph build on the same set of reads with all 3-mers earmarked except for `TGC` and `GCA`.



a



b

substring are compared. This allows to avoid comparing all pairs of input strings. However the database of all k-mers may be really huge. To reduce the storage requirements Roberts et al. propose to select not all the k-mers, but only those that are minimal in an input string with respect to a certain order. Such k-mers are called *minimizers*. Our earmarked k-mers are close in spirit to these minimizers.

2 Earmark graph

Let $0 < k < r$ be integers and $\mathcal{R} \subseteq \{A, C, G, T\}^r$ be a set of reads of length r. For a read R and indices $1 \le i \le j \le r$, let $R[i, j]$ be a substring of R starting at position i and ending at position j. A *de Bruijn graph* $\mathrm{DG}_k(\mathcal{R})$ is defined as follows:

- the set of all substrings of length k (or just k-mers) of the given reads is the set of vertices;
- two k-mers A and B are joined by a directed edge if there is a read $R \in \mathcal{R}$ and an index $1 \le i \le r - k - 1$ such that $A = R[i, i + k - 1]$ and $B = R[i + 1, i + k]$ (this, in particular, implies that the suffix of A of length $k - 1$ equals the prefix of B of the same length).

we should take care about edges multiplicity here if we would like to speak about an Eulerian cycle in this graph

Note that each read $R \in \mathcal{R}$ is represented by a path of length $r - k - 1$ on its k-mers in $\mathrm{DG}_k(\mathcal{R})$.

tell that a genome corresponds to a cycle in this graph?..

An earmark graph can be viewed as the de Bruijn graph with some paths contracted into a single edge. Let $\mathcal{E} \subseteq \{A, C, G, T\}^k$ be a set of k-mers. Below we refer to it as a *set of earmarks* or as a *set of earmarked k-mers*. An *earmark graph* $\mathrm{EG}_k(\mathcal{R}, \mathcal{E})$ is defined as follows:

- $\mathcal{E}$ is the set of vertices;
- two k-mers A and B are joined by a directed edge if there is a read $R \in \mathcal{R}$ and indices $1 \le i < j \le r - k - 1$ such that $A = R[i, i + k - 1]$ and $B = R[j, j + k - 1]$ and for any $i < t < j$, $R[t, t + k - 1] \notin \mathcal{E}$ (i.e., B is the next earmarked k-mer after A in R); the weight of this edge is set to $j - i$.

Hence, in the earmark graph each read is represented as a path (of length at most $r-k-1$) on its earmarked k-mers. When constructing an earmark graph

it is reasonable to ensure that each read contains at least two earmarked k-mers (so that the length of the corresponding path is at least one). It is easy to see that in the special case when we all the possible k-mers are earmarked the earmark graph coincides with the de Bruijn graph.

Both de Bruijn and earmark graphs are built on a set of reads that can be viewed as a set of reads from an unknown genome. A natural way to build these two graphs on a genome itself is to use the set of all its reads of a particular length. Namely, for a genome $S \in \{A, C, G, T\}^*$, define $\Gamma_r(S) \subseteq \{A, C, G, T\}^r$ as a set of all r-mers appearing in S. Then $\mathrm{DG}_k(S)$ and $\mathrm{EG}_k(S, \mathcal{E})$ are just $\mathrm{DG}_k(\Gamma_r(S))$ and $\mathrm{EG}_k(\Gamma_r(S), \mathcal{E})$, respectively.

3 Earmark graph construction

is it ok to give just one sentence about graph simplification here?

In this section, we give a high-level description of an earmark graph construction procedure. When the graph is constructed we simplify it using methods similar to the ones used in EULER and Velvet.

We first construct an initial set of earmarks and then extend it to reduce the number of tips in the resulting graph.

3.1 Earmarks selection procedure

To construct an initial set of earmarks $\mathcal{E}$, we select from each given read several k-mers that are minimial with respect to some ordering. Namely, let h be a hash-function on the set of all possible k-mers and t be a number. We then select initial earmarks as follows.

1. For each read, select t minimum hash values of its k-mers and add them to the global set of earmarked hash values.
2. If $t = 1$, earmark the second smallest hash value for each read with only one k-mer with earmarked hash value. This guarantees that each read has at least two k-mers with earmarked hash values.
3. Add to $\mathcal{E}$ all k-mers that have their hash values earmarked.

3.2 Tip extension procedure

Assume that we are given a set of reads $\mathcal{R}$ of an unknown genome S. It is easy to see that parts of the genome S that are not covered by any read

from $\mathcal{R}$ create vertices of in- or our-degree one in $\mathrm{DG}_k(\mathcal{R})$. Such vertices are called *tips*. At the same time, the earmark graph may contain tips not only because of coverage gaps, but also tips resulting from a bad choice of the set of earmarks. To give an example, consider a genome substring $ABCDE$, where B and D are earmarked k-mers. If none of the reads in the input data contain the whole substring BCD, then vertices corresponding to B and D will be tips in the earmark graph. Meanwhile reads containing ABC and CDE may be present in the input data (and hence the whole considered part is covered by reads), so these vertices must not be tips.

The tip extension procedure is suggested to handle this issue. It consists of three steps that are repeated consequently until no possible tip extensions are present. (Each steps demands reading the entire input data. It is possible to reduce the procedure to less than three steps, but much more memory will be used in that case).

1. By processing all the reads, find out for each earmarked k-mer, whether we have seen any earmarked k-mer to the left of it and to the right from it. Those earmarked k-mers that do not have either left or right "neighbours" are called left and right tips, respectively.

2. By processing all the reads, find the collection of all the non-earmarked k-mers that are present (at least once) in a read to the left of a left tip, or to the right of the right tip. Call them possible tip extensions.

3. By processing all the reads, for each possible tip extension, record whether there was any earmarked k-mer to the left of it and to the right of it.

Now that this information is collected, each tip is being extended with a possible tip extension using the following rules (in the order preference).

1. Check if any of its possible tip extensions was already selected as an earmark (as a tip extension for some previously processed tip, using rules 2 and 3 below). If so, continue to the next tip.

2. Check if any of its possible tip extensions has both left and right neighbours. We select this extension as an earmark and thus eliminate at least one undesirable tip.

3. Select as an earmark the possible tip extension that is most distant (in base pairs) from the tip being proccesed. Doing this, we do not eliminate a tip but we extend it as far as possible. This helps us not to lose information near the actual gaps (or the ends of actual scaffolds).

probably it is reasonable to give an example here?

If no new earmarked k-mers were introduced after this procedure, then stop, otherwise repeat the entire procedure.

4 Practical results

to be written

5 Discussion and further ideas

to be written

5.1 Advantages of earmark graph over de Bruijn graph

should it be here?

5.2 A more careful earmarks selection

paired read info?

clean up the references