Optimal_k - DB-graph inference by accurate sampling Fast and accurate selection of parameters for genome assembly (by sampling)

1 Abstract

Motivation: There is no clear way on how to chose parameters k-mer size and abundance for a De Bruijn based de novo assembler. As *de novo* genome assembly is time consuming for large genomes, it is of importance to chose these parameters well in order to prevent multiple runs. Current software for estimating k only optimize certain features such as maximizing the number of genomic k-mers. There is a need for more clear objectives such as E-size or N50.

Results: We provide a method (optimal_k) to estimate average unitig length, N50 and E-size for all combinations of minimum abundance and k in one run. As unitigs are a foundation of the de Bruijn graph, estimating these quantities provides an understanding of the quality of a DBG based genome assembly as well as a good base for chosing the best combination of k and abundance. The estimations obtained by optimal_k are extremely accurate. [We also note that these estimations also accurately predict the best quality for DBG based assemblers that perform more steps such as tip removals, bubble popping and usage of paried end read information.]

2 Introduction

Mention that there are not many tools for computing optimal parameters at all. And make sure to mention that memry is not the issue. Mention the positives about our methods like speed and clear objective function but make sure to mention that it's memory requiring but thats not a problem if you are going to do the assembly anyway!!

Given an assembly graph G, a path $P = (v_0, v_1, \ldots, v_t)$ in G, where $t \geq 0$, is called a *unitig* if for all $i \in \{1, \ldots, t-1\}$, v_i is a *unary* node, that is, v_i has exactly one in-neighbor and one out-neighbor in G, and v_0 and v_t are not unary. In the contig assembly phase, popular genome assemblers report a unitig decomposition of the assembly graph, after some artifacts have been been dealt with, like tip removal and bubble popping.

3 Methods

The general idea is to provide the user with metrics such as unitigs N50 and E-Size and average number of genomic vertices in a DBG for all possible k-mer sizes and abundances. We implement a FM-index data structure described in cite XX. This allows us to query a k-mer, its in and out neighbors in O() time. We furthermore derive formulas for how much we need to sample in order to reach a given accuracy on all our estimates.

Let R be the set of input reads, and let $\mathcal{K}_k(R)$ be the multiset of all k-mers in the reads, and denote the cardinality of this multiset by $\mathcal{K}_k(R)$. That is, $|\mathcal{K}_k(R)| = n(r-k+1)$, where n denotes the number of reads and r is the read length. Moreover, we denote by $DB_{k,a}(R)$ be the de Bruijn graph built of *order* k and *minimum abundance* a built on the set of reads R. That is, the set of vertices of $DB_{k,a}(R)$ is the set of all k-mers in the reads which occur at least a times in R, and two vertices of $DB_{k,a}(R)$ are connected by an arc if they have a suffix-prefix overlap of length $k-1$. Let $V(DB_{k,a}(R))$ denote the set of vertices of $DB_{k,a}(R)$. For all $v \in V(DB_{k,a}(R))$, let $\alpha(v)$ denote the abundance of the k-mer v. This can be obtained by a query to the index built on the set R.

Talk about reverse complements in dB graphs: a k-mer and its reverse complement are bundled into the same node and the abundances are added up. Say how we deal with this case in practice.

Give pseudo-code of how we get the in/out degrees for all abundances.

Say that one of the main ideas is to do weighted sampling.

3.1 Algorithms

Estimating the number of nodes of a DBG Let $V(DB_{k,a}(R))$ denote the set of vertices of $DB_{k,a}(R)$, and let $\mathbb{I}(x,a)$ be an indicator variable returning 1 if the k-mer x has abundance at least a in R, and 0 otherwise. We can write

$$|V(DB_{k,a}(R))| = \sum_{x \in \mathcal{K}_k(R)} \frac{1}{\alpha(x)} \mathbb{I}(x,a).$$

Since $V(DB_{k,a}(R))$ is a subset of the multiset $\mathcal{K}_k(R)$, we can consider the proportion

$$p_k := \frac{|V(DB_{k,a}(R))|}{|\mathcal{K}_k(R)|} = \frac{\sum_{x \in \mathcal{K}_k(R)} \frac{1}{\alpha(x)} \mathbb{I}(x,a)}{|\mathcal{K}_k(R)|} \in [0,1].$$

We can estimate p_k by sampling a multiset $\{x_1, \ldots, x_n\}$ of n k-mers from $\mathcal{K}_k(R)$, and taking

$$\hat{p_k} = \frac{\sum_{i=1}^{n} \frac{1}{\alpha(x_i)} \mathbb{I}(x_i,a)}{n}. \tag{1}$$

Therefore, we get an estimate of $X := |V(DB_{k,a}(R))|$ as $\hat{X} = \hat{p_k}|\mathcal{K}_k(R)| = \hat{p_k} n(r-k+1)$. By the theory in **??**, we immediately get how many samples we need in order to bound the relative error within a certain confidence interval.

Say that we can implement this for all abundaces

Estimating the average number of nodes in a DBG Note that we get the number of unitigs in a graph by counting all the start vertices and their outdegree. Let X_s be the number of start nodes in $\mathcal{G}$. We also label all ot in the DBG is given We now divide the set X into Y be the number of internal vertices in the DBG. An internal vertex in a vertex that...

Estimating the E-size Let $U_{k,a}(R)$ denote the set of all unitigs of $DB_{k,a}(R)$. The E-size of $U_{k,a}(R)$ is defined as the expected length of the unitigs of $DB_{k,a}(R)$. Formally,

$$\mathrm{E}_{\text{size}}(U_{k,a}(R)) := \sum_{s \in U_{k,a}(R)} |s| p(s) = \sum_{s \in U_{k,a}(R)} |s| \frac{|s|}{\sum_{s' \in U_{k,a}(R)} |s'|} = \frac{\sum_{s \in U_{k,a}(R)} |s|^2}{\sum_{s \in U_{k,a}(R)} |s|}.$$

In order to derive an estimate for E-size, we will construct a sampling procedure which may sample a unitig more than once. Since the above formula contains each unitig once, we need to normalize it with the expected number of times of sampling each unitig.

Consider the following procedure which produces a multiset W containing the unitigs of $DB_{k,a}(R)$: for all k-mers $x \in \mathcal{K}_k(R)$, if x is a vertex of $DB_{k,a}(R)$, output all the unitigs containing x. Each unitig $w := (v_1, v_2, \ldots, v_t)$ of $DB_{k,a}(R)$ appears $\alpha(w) := \frac{1}{t}\sum_{i=1}^{t} \alpha(v_i)$ times in W. Therefore, we can equivalently express the E-size of the set $U_{k,a}(R)$ as

$$\mathrm{E}_{\text{size}}(U_{k,a}(R)) = \sum_{w \in W} |w| \frac{|w| \frac{1}{\alpha(w)}}{\sum_{w \in W} |w| \frac{1}{\alpha(w)}}.$$

Instead of iterating over all k-mers in $\mathcal{K}_k(R)$, we now sample a multiset of k-mers in $\mathcal{K}_k(R)$, and if they are nodes in the de Bruijn graph, we report the unitigs containing these k-mers. Assume that this produces a multiset $\{w_1, \ldots, w_n\}$ of unitigs of $DB_{k,a}(R)$. We can estimate $\mathrm{E}_{\text{size}}(U_{k,a}(R))$ as

$$\hat{\mathrm{E}}_{\text{size}} := \sum_{i=1}^{n} |w_i| \frac{|w_i| \frac{1}{\alpha(w)}}{\sum_{j=1}^{n} |w_j| \frac{1}{\alpha(w)}}.$$

3.2 Sampling accuracy

In this section we will derive the sample sizes required to get accurate estimations of the quantities that we want to estimate. We recall some basic statistical notions in Sec. 3.2.1 and show how they applies to our quantities in Sec. **??**.

3.2.1 Theory

Sample proportion. Suppose that we have a set partitioned as $A \cup B$, and we need to estimate the proportion $p = |A|/(|A| + |B|)$. Suppose that we sample n elements of $A \cup B$ and for each of them record whether they belong to A or to B, obtaining in this

way an estimate $\hat{p}$ of p. It is a standard result that if we obtain $\hat{p}$ from n samples, the $100(1-\alpha)\%$ confidence interval of $\hat{p}$ is

$$\left[\hat{p} - z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right]$$

where $z_{\frac{\alpha}{2}}$ is the $\alpha/2$ quantile from the normal distribution. For a given relative error ϵ, we want to choose the sample size n such that the $100(1-\alpha)\%$ confidence interval of $\hat{p}$ has a margin of error no more than $E := \epsilon p$. By standard means, we obtain

$$n \geq \left(\frac{z_{\frac{\alpha}{2}}}{E}\right)^2 \hat{p}(1-\hat{p}). \tag{2}$$

Notice that in relation (2) above, both p and $\hat{p}$ are not known at the start of the sampling, when the value of n needs to be chosen. In our case, we choose ...

4 Results and discussion

5 Conclusions