Algorithms in Bioinformatics

Spring 2021 - BIOINFOM222-1 / CHEMCM160B-1 / CHEMCM260B-1 / COM SCICM122-1 / COM SCICM222-1 - ESKIN / HE

Search forums...

Discussion forum

Paper Response: Velvet

Settings

◀ Paper Response: Bowtie

Display replies in nested form

Paper Response: Velvet
by HE, ROSEMARY - Tuesday, 11 May 2021, 1:48 PM PDT

Please read the velvet paper posted under week 7 and post 1 question under this forum by Wednesday the 19th, and 2 responses by Friday the 21st.

Permalink Reply

Re: Paper Response: Velvet
by LONG, GABRIELLA - Tuesday, 11 May 2021, 3:34 PM PDT

With Velvet it seems that most of the memory and time are used during the aligning of the reads and/or the initial construction of the de Bruijin graph. However, I would have thought that repeatedly removing errors and re-simplifying the graph would have taken more time and memory requirements, but it doesn't seem that way according to the "Complexity and scaling issues" section of the paper. I know the paper mentioned that the tour bus algorithm is around O(n log n) since it was based on Dijkstra's algorithm, but is there an overall time complexity for the error removal process in Velvet?

Permalink Show parent Reply

Re: Paper Response: Velvet
by YU, DANNING - Wednesday, 12 May 2021, 12:14 PM PDT

I agree that removing errors and simplifying the graph would be quite time consuming, but I think the time spent on it is low compared to initial graph construction because there is a low rate of errors in the reads (they tested with 1% errors and 1 SNP per 500 bp). Most of the graph is probably either correct or has a low rate of errors that's not enough to meet the consensus threshold.

Permalink Show parent Reply

Re: Paper Response: Velvet
by CHEN, CAROLINE - Tuesday, 11 May 2021, 6:24 PM PDT

In the paper, they mention that the N50 for the human BAC graph is 1958 for all nodes, 2041 bp for nodes > 100bp, and 3266 bp for nodes > 1000 bp, while the graph of Streptococcus suis has a N50 of 8564 for all nodes, and then is relatively similar (8742 bp for nodes > 100 bp, and 8871 for nodes > 1000 bp). They mention that the Alu repeats in the human genome limit the N50 which makes sense, but how come the relative increase in N50 in 100 bp nodes vs 1000 bp nodes is not as pronounced in Streptococcus suis as compared to the human BAC?

Permalink Show parent Reply

Re: Paper Response: Velvet
by PIMPLASKAR, ADITYA - Tuesday, 11 May 2021, 8:43 PM PDT

QUESTION
The Tour Bus bubble removal method merges the longer sequence into the smaller one. How does this affect the assembly if the longer sequence gets too long? Also, does inserting the modified component of the longer sequence have any bearing on intermediate kmers in the shorter sequence?

Permalink Show parent Reply

Re: Paper Response: Velvet
by MISHRA, SRIPATH - Wednesday, 12 May 2021, 11:11 AM PDT

For the first question: on page 823 of the pdf under the removing bubbles with the tour bus algorithm subsection the author expresses the usage of length thresholds. I wasn't able to find any other signaling information. My current understanding is that the bus tour algorithm has a max length threshold which prevents the algorithm to negatively impact the assembly due to long sequence.

Permalink Show parent Reply

Re: Paper Response: Velvet
by JIA, SICHENG - Wednesday, 12 May 2021, 11:19 AM PDT

I believe that the two paths are merged via local sequence alignment, so only a smaller portion of the larger sequence will be inserted into the smaller sequence. Since the merge also merges the node information of both paths, the coverage information should indicate the quality of an inserted node.

Permalink Show parent Reply

Re: Paper Response: Velvet
by NATESAN, GUNALAN - Tuesday, 11 May 2021, 9:14 PM PDT

The paper says that shotgun sequencing is random. However, is there some structure behind the shotgun sequencing that may allow for sequencing data more easily? In other words, How random are the reads from shotgun sequencing? Is there some usable information (in the entropy sense) regarding the distribution?

Permalink Show parent Reply

Re: Paper Response: Velvet
by WANG, YE - Wednesday, 12 May 2021, 8:47 AM PDT

It is a very good question. Actually the distribution of reads couldn't been completely random. The bias could be introduced during DNA fragmentation (which is less pronounced in sonication but much more evident in enzyme fragmentation), PCR amplification (the amplification efficiency varies for different sequence due to their GC content), and multiple experimental procedures. However, the randomness of reads distribution is also the basis for many bioinformatic applications like CNV calling, I think this problem could be investigated by the benchmark of NGS on long, aritifical synthesized sequence (to ensure the fidelity of reference sequence)

Permalink Show parent Reply

Re: Paper Response: Velvet
by TURNBULL, STEVEN - Wednesday, 12 May 2021, 1:08 AM PDT

Questions:
Are there any advantages to using SSAKE and VSAKE compared to Velvet? (besides the slight memory difference mentioned in the paper)
How is Velvet's Space Complexity worse? I'm sure I'm missing something pretty obvious here but from my understanding Velvet would have less nodes in the de brujin graph and overall aren't we storing the same data? I'm not too familiar with the other to algorithms' method of storage so please let me know if it's something pretty obvious.

Permalink Show parent Reply

Re: Paper Response: Velvet
by CHEN, CAROLINE - Wednesday, 12 May 2021, 5:43 PM PDT

I am also not familiar with the other algorithms' methods of storage, but I found this 2018 paper (linked below) that compares seven assemblers including SSAKE and Velvet using paired end and single end reads in prokaryotic and eukaryotic datasets. They found that in prokaryotic single end datasets, SSAKE used the highest amount of memory, and in the eukaryotic single end data set Velvet consumed the least memory (while SSAKE (and Perga) consumed the highest memory).
In prokaryotic paired end datasets Velvet used more memory than SSAKE, and in eukaryotic paired end datasets Velvet consumed the highest amount of memory, while SSAKE used the least amount of memory.
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5826002/

Permalink Show parent Reply

Re: Paper Response: Velvet
by WU, RAYWEN - Wednesday, 12 May 2021, 4:53 AM PDT

Question:
As we know De bruijn graph is used for sequence assembly, same as de novo. however, de novo is based on the de bruijn's algorithm; what are the key features that make de novo to be able to eliminate the errors and resolve the repeats? Briefly describe how these features work.

Permalink Show parent Reply

Re: Paper Response: Velvet
by WANG, YE - Wednesday, 12 May 2021, 8:35 AM PDT

Some graph features could be ascribed with useful information, such as the unique paired path between A and B in figure 5 that are separated by repetetive regions is correspond to the most simplified path without repeats.

Permalink Show parent Reply

Re: Paper Response: Velvet
by WOO, JUSTIN - Wednesday, 12 May 2021, 2:28 PM PDT

De novo is a class of algorithms that reassembles short reads into a longer sequence without the reference genome. De Bruijn graphs are one implementation of this class that optimize for global optimality. However, there are also greedy de novo resequencing algorithms to target local optimality.

Permalink Show parent Reply

Re: Paper Response: Velvet
by WANG, YE - Wednesday, 12 May 2021, 8:24 AM PDT

Question: I am not clear about the concept "shortest path" in Tour Bus algorithm when removing bubbles. The paper said "The path that reached the end
node first in the search, "shortest" according to the metric, is used as the consensus path because of its higher coverage." Does it means B-C has smaller length than B'-C' ?(but two sequences with SNP should have same length), or B-C is just arbitrarily selected to be firstly travesed?

Permalink Show parent Reply

Re: Paper Response: Velvet
by WANG, SHAODIAN - Wednesday, 12 May 2021, 10:27 AM PDT

Response:
In the paper, "distance" is defined as "length of s(B) divided by the multiplicity of the arc leading from A to B". Multiplicity refers to the number of times that arc appears. Intuitively, this distance metric favors "higher coverage, more reliable, paths" when

comparing paths of the same length.
Since this distance definition is used for the variant of Dijkstra's algorithm in Tour Bus, B-C in the example is traversed before B'-C' likely because it is "shorter", i.e. has greater multiplicity.

Permalink Show parent Reply

Re: Paper Response: Velvet
by JIA, SICHENG - Wednesday, 12 May 2021, 11:14 AM PDT

I agree that the shortest path metric seems counterintuitive for merging. It makes more sense to merge the shorter path to the longer path to obtain a higher coverage overall. However the paper is right, the shorter path will have the higher cover, and by merging to the shortest path, this will implicitly conduct a majority vote. In the example, I believe that B-C and B'-C' are traversed simultaneously, and because B'-C' is longer when it reaches D, B-C will already have visited that node, thus leading to a merge.

Permalink Show parent Reply

Re: Paper Response: Velvet
by MISHRA, SRIPATH - Wednesday, 12 May 2021, 10:25 AM PDT

Question: The paper mentions ideal number of nodes/graph and other ideal things. How was ideal calculated in the first place?

Permalink Show parent Reply

Re: Paper Response: Velvet
by WANG, SHAODIAN - Wednesday, 12 May 2021, 11:02 AM PDT

Response:
From the paper, the ideal graph is "a de Bruijn graph from the known finished sequence of the BAC", which is "equivalent to an error-free, gap-free assembly". The results from Velvet are then compared with the results from running Tour Bus on this ideal graph.

Permalink Show parent Reply

Re: Paper Response: Velvet
by WU, RAYWEN - Wednesday, 12 May 2021, 7:53 PM PDT

Response:
The idea of ideal is that the de Bruijn graph has no potential mistakes with that graph exists; we can therefore implement that graph and go through the Tour Bus algorithm. After running the Tour Bus, it would increases the sensitivity and specificity if the correction as well as the quality and Velvet is not really affected by the variations of the reads. Moreover, the ideal things include shorter reads could be covered more than 90%, and no miss-assembly is created.

Permalink Show parent Reply

Re: Paper Response: Velvet
by JIA, SICHENG - Wednesday, 12 May 2021, 11:07 AM PDT

The paper mentions that Velvet is sensitive to user parameters such as k for the k-mer length. Is there a way (beside experimentally) to estimate the optimal k?

Permalink Show parent Reply

Re: Paper Response: Velvet
by YU, DANNING - Wednesday, 12 May 2021, 12:09 PM PDT

The paper gives a formula at the bottom of page 827 to estimate k in terms of coverage and read length, with the idea that we want the expected number of times a given k-mer is observed to be approximately 10-15 (presumably so that their error correct algorithms have enough consensus to work well). Since we know coverage from the number of reads, a starting value of k can be estimated.

Permalink Show parent Reply

Re: Paper Response: Velvet
by MCQUILLEN, CADEN - Wednesday, 12 May 2021, 5:02 PM PDT

Response: I would guess that like when we were building de bruijn graphs, shorter k-mers would give more branching than longer k-mers so I would think that background information like that would help eliminate a range of k-mer sizes without having to experimentally test them.

Permalink Show parent Reply

Re: Paper Response: Velvet
by WU, RAYWEN - Wednesday, 12 May 2021, 8:06 PM PDT

Response:
As the paper mentions, to find an optimal k-mer, we need to take genome, the coverage and the length of the reads. Another method other than the experimental actions, we can estimate the number N of times a unique k-mers in the genome that has the length of L that are broke into m reads, and each read has length of l. And we can treat these information as to solve for the coverage to get the optimal k-mer as desired.

Permalink Show parent Reply

Re: Paper Response: Velvet
by WANG, SHAODIAN - Wednesday, 12 May 2021, 11:21 AM PDT

Question:
In the "Removing erroneous connections" subsection of the paper, it is mentioned that connections are removed "with a basic coverage cutoff", that is "set by the user, based on plots of node coverage after the removal of bubbles". I wonder if there is a heuristic for determining the cutoff. It seems from Table 1 and Table 2 that around half of the nodes from Tour Bus are removed in this step.

Permalink Show parent Reply

Re: Paper Response: Velvet
by LAM, RYAN - Wednesday, 12 May 2021, 11:20 PM PDT

I imagine that the user should take into account the expected coverage value according to the sequencing technology used, in order to choose a coverage cutoff value that is sufficiently low enough to only remove truly erroneous nodes.

Permalink Show parent Reply

Re: Paper Response: Velvet
by YU, DANNING - Wednesday, 12 May 2021, 12:07 PM PDT

In the breadcrumb algorithm section, it mentions how the algorithm does not work well for long repeats, such as the Alu repeat, since they create long overlaps and ambiguous loops. How are such areas typically sequenced or assembled, given that long repeats tend to cause issues with most assembly algorithms?

Permalink Show parent Reply

Re: Paper Response: Velvet
by KURODI, ADITYA - Thursday, 13 May 2021, 5:10 PM PDT

This is not at all substantiated with any real system, but I remember reading about interpolating the gaps between read pairs in order to form an even longer sequence. So perhaps via experimentation it would be possible to observe markers in the reference genome that come before and after a long repeat like the ALU sequence, and then crop around it with many paired reads of various gap lengths. Then in joining the paired reads, we also envelop and ultimately sequence the repeat as well.

Permalink Show parent Reply

Re: Paper Response: Velvet
by FENG, BENJAMIN - Wednesday, 12 May 2021, 12:56 PM PDT

The paper mentions that Alu's were problems when it came to creating the contigs. Are there any algorithms that are being worked on

to make sequencing these regions easier?

Permalink Show parent Reply

Re: Paper Response: Velvet
by WOO, JUSTIN - Wednesday, 12 May 2021, 2:22 PM PDT

ALUs cause trouble because they are long, repeated sequences, making it difficult to determine their correct location in the genome. There is a new pipeline called TypeTE from an April 2020 paper in Nucleic Acids Research by Goubert, Thomas, et al. that is specifically optimized for ALUs using local reassembly to reconstruct alleles.

Permalink Show parent Reply

Re: Paper Response: Velvet
by LEE, HANNAH - Wednesday, 12 May 2021, 1:59 PM PDT

Q: The tour bus algorithm seems to be useful in that it allows velvet to dynamically modify the path as it corrects the graph. However, it seems that merging two paths would be quite complex/time consuming as all the underlying graph structures must be remapped—not sure if I misinterpreted but what are the main advantages of tour bus and are there other methods that have more advantages in this regard?

Permalink Show parent Reply

Re: Paper Response: Velvet
by LONG, GABRIELLA - Wednesday, 12 May 2021, 3:07 PM PDT

Response: From my understanding, which could very well be wrong, it seems like one of the main advantages of the tour bus algorithm is its preservation of important low-coverage nodes. So while it may not be the fastest or cheapest algorithm, it is a decent choice since it keeps the complexity down by merging similar nodes together yet is able to preserve the important low-coverage nodes by turning them into longer nodes with average coverage. Unfortunately I'm not aware of other methods that may be able to solve the shortcomings of the tour bus algorithm, so I would also be interested if someone knows of any.

Permalink Show parent Reply

Re: Paper Response: Velvet
by WOO, JUSTIN - Wednesday, 12 May 2021, 2:34 PM PDT

Velvet uses De Bruijn graphs to effectively assemble short read sequences. How does Velvet compare to existing whole-genome shotgun sequencing algorithms for longer reads?

Permalink Show parent Reply

Re: Paper Response: Velvet
by YU, ALEX - Wednesday, 12 May 2021, 3:38 PM PDT

The paper states that Velvet focuses on removing three types of structures (tips, bubbles, and erroneous connections). Are there any other major topological features that might be sources of errors, and if so, what are the approaches to dealing with them?

Permalink Show parent Reply

Re: Paper Response: Velvet
by XING, FRANK - Thursday, 13 May 2021, 12:24 PM PDT

Response: Based on what the paper discusses, it appears that the majority of errors would manifest in the form of tips (errors at edges of reads), bubbles (internal read errors or to nearby tips connecting), or errorneous connections due to cloning errors or distant merging tips. The rest of the differences could be accounted for due to the presence of polymorphism.

Permalink Show parent Reply

Re: Paper Response: Velvet
by XUE, ALBERT - Wednesday, 12 May 2021, 3:54 PM PDT

I think it's interesting how sensitive Velvet is to k. Why would increasing k ever lead to a less optimal answer?

Permalink Show parent Reply

Re: Paper Response: Velvet
by MCQUILLEN, CADEN - Wednesday, 12 May 2021, 5:08 PM PDT

Response: I would think that eventually increasing k too large would lead to memory and possible also time issues when constructing the de bruijn graph. There also is probably another constraint on increasing k that I am not thinking of because if there wasn't then they probably would have stated that using the maximum possible k given your read length gives the best results.

Permalink Show parent Reply

Re: Paper Response: Velvet
by NGUYEN, HANNAH - Wednesday, 12 May 2021, 10:09 PM PDT

I would think increasing k would always make it easier to construct the de Bruijn graph, since there will be fewer nodes and possible paths, and result in fewer errors and repeats in the graph. I believe k is limited by the length of the read, which is limited by the existing sequencing technology.

Permalink Show parent Reply

Re: Paper Response: Velvet
by YU, ALEX - Wednesday, 12 May 2021, 11:18 PM PDT

One possible reason that I thought of is that increasing k might increase the probability that another kmer doesn't overlap with it, meaning that the graph would end up disjoint and you wouldn't be able to assemble the whole sequence.

Permalink Show parent Reply

Re: Paper Response: Velvet
by MISHRA, SRIPATH - Thursday, 13 May 2021, 10:33 AM PDT

Answer: Having a longer K-mer will lead to a decrease in connectivity as the chance of having an overlap between two reads decreases. Longer reads does lead to decrease in ambiguous reads though which is an advantage.

Permalink Show parent Reply

Re: Paper Response: Velvet
by MCQUILLEN, CADEN - Wednesday, 12 May 2021, 4:59 PM PDT

Question: In the paper they mentioned that they remove "tips" which are a chain of nodes that are disconnected at one end. They choose an arbitrary cutoff length of 2k which they say is because "greater than the length in k-mers of an individual very short read". Then then state that any tip longer than that is either genuine or an accumulate of errors that is hard to distinguish from a novel sequence. My question is how would one approach trying to differentiate these two things? Is there some useful biological background that could be used in helping determine if one of these long tips is genuine?

Permalink Show parent Reply

Re: Paper Response: Velvet
by CHEN, CAROLINE - Wednesday, 12 May 2021, 6:27 PM PDT

I'm not sure if there is something biologically intrinsic to the sequence that could tell us whether or not this is more likely an accumulation of errors vs. a novel sequence. I may be wrong but I think the 'minority count' step that it iterates tip removal with the 'length criteria step' could help to differentiate between accumulation of errors and novel sequence since I'd imagine sequencing errors would result in lower coverage nodes and would be discarded as tips are removed in increasing order of multiplicity.

Permalink Show parent Reply

Re: Paper Response: Velvet
by TANDON, YASH - Wednesday, 12 May 2021, 8:23 PM PDT

What are the primary differences and extensions between the de Bruijn algorithm and velvet?

Permalink Show parent Reply

Re: Paper Response: Velvet
by NGUYEN, HANNAH - Wednesday, 12 May 2021, 10:00 PM PDT

The Velvet algorithms are post-assembly extensions, i.e. they improve on existing de Bruijn graphs. Error correction removes tips, bubbles, and erroneous connections to merge sequences together, and the repeat solver separates paths in the de Brujin graph that have overlap.

Permalink Show parent Reply

Re: Paper Response: Velvet
by HSU, NEIL - Wednesday, 12 May 2021, 10:54 PM PDT

Velvet utilizes a de Bruijn graph method to perform genome assembly. Velvet is essentially a collection of tools developed to simplify the de Bruijn graph, identify and remove errors (finding them as tips, bubbles, and erroneous connections), in addition to using the breadcrumbs method to deal with repeats. Essentially, using a de Bruijn graph as a model for the reads and overlaps between them, Velvet can assemble the genome.

Permalink Show parent Reply

Re: Paper Response: Velvet
by LIAN, ROGER - Thursday, 13 May 2021, 1:40 PM PDT

The Velvet is achieved through the manipulation of de Bruijn graphs for genomic sequence assembly via the removal of errors and the simplification of repeated regions. I think the primary extensions for Velvet is the simplification and error removal.

Permalink Show parent Reply

Re: Paper Response: Velvet
by MURSHED, FARAZ - Wednesday, 12 May 2021, 8:25 PM PDT

Question:
As said in the paper, Velvet assists in assembly by leveraging very short reads in combination with read-pairs. What possible methods and extensions could be employed to leverage longer reads?

Permalink Show parent Reply

Re: Paper Response: Velvet
by HSU, NEIL - Wednesday, 12 May 2021, 10:40 PM PDT

Longer reads make overlaps less ambiguous, allowing for a less tangled graph and if it exceeds the length of repeats, the problem of repeats are easy to deal with. However, as mentioned in Stepik Chapter 3 Epilogue, the problem of longer reads lie in that the most accurate sequencing techniques can only create reads up to 300 base pairs. Additionally, the newer/cheaper sequencing methods generate very short reads; Solexa for example generates 35 bp reads as stated in the paper. This is why Velvet is good in that it utilizes short reads, which are generally hard to deal with, and uses a quick and accurate algorithm to perform genome assembly.

Permalink Show parent Reply

Re: Paper Response: Velvet
by NGUYEN, HANNAH - Wednesday, 12 May 2021, 10:04 PM PDT

How would the BreadCrumb (the repeat resolver) process be affected if the kmers were longer but the overall coverage of the genome was reduced?

Permalink Show parent Reply

Re: Paper Response: Velvet
by HUANG, ETHAN - Thursday, 13 May 2021, 6:44 PM PDT

R:
Breadcrumb's purpose is resolving repeats and extending/connecting contigs to reduce tangling in the DB graph. Increasing the k-mer length should help because longer k-mers reduces tangling. However, reducing the overall coverage, according to the Figure 4, exponentially decreases the resulting contig length.

Permalink Show parent Reply

Re: Paper Response: Velvet
by HSU, NEIL - Wednesday, 12 May 2021, 10:27 PM PDT

In figure 2 when they are explaining how Tour Bus worked, it is mentioned that if the two paths in the bubble are deemed similar, the longer one is merged into the shorter one. Is there a reason why the longer one is merged into the shorter one?

Permalink Show parent Reply

Re: Paper Response: Velvet
by LAM, RYAN - Wednesday, 12 May 2021, 10:58 PM PDT

The paper states that the shorter path is used as the consensus path because it has a higher coverage than the longer path.

Permalink Show parent Reply

Re: Paper Response: Velvet
by LIAN, ROGER - Thursday, 13 May 2021, 3:37 PM PDT

According to the paper, they measure the length between two consecutive nodes A and B is the length of s(B) divided by the multiplicity of the arc leading from A to B. This metric gives priority to higher coverage, more reliable, paths because the path that reached the end node first in the search, “shortest” according to the metric, isused as the consensus path because of its higher coverage.

Permalink Show parent Reply

Re: Paper Response: Velvet
by LAM, RYAN - Wednesday, 12 May 2021, 11:06 PM PDT

As stated in the Complexity and Scaling Issues section, the construction of the de Bruijn graph is the main computational bottleneck for Velvet. The paper also states that, for whole genome assembly, memory persistent data structures will be needed as they provide virtually unlimited data storage. Functional programming languages commonly have memory persistent data structures, as the languages discourage using mutable data. What would be the pros and cons of implementing Velvet using a functional language as opposed to the current implementation in C?

Permalink Show parent Reply

Re: Paper Response: Velvet
by XING, FRANK - Thursday, 13 May 2021, 12:35 PM PDT

Response: My understanding of functional langugages is that one of the main limitations is the non-existence of classes. Functional languages mainly uses recursion, which works will with the BFS method discussed in the paper, but it would have trouble generating a node class which consists of the needed coverage, sequence identifiers, and arcs.

Permalink Show parent Reply

Re: Paper Response: Velvet
by XING, FRANK - Thursday, 13 May 2021, 12:18 PM PDT

Question: The Tour Bus algorithm mentions the difficulty of merging when there is a Palindrome shape. How would using a chain of markers and mapping the first unmapped minority node to the corresponding major consensus node solve this issue? More

specifically, how would the merged path look like compared to the original Palindrom shape?

Permalink Show parent Reply

Re: Paper Response: Velvet
by LIAN, ROGER - Thursday, 13 May 2021, 1:38 PM PDT

Question: the paper mentioned that the results of Velvet are very sensitive to the parameter k, I'm wondering how expensive it is to get the optimal k for the best result?

Permalink Show parent Reply

Re: Paper Response: Velvet
by YU, ALEX - Thursday, 13 May 2021, 3:04 PM PDT

I might be wrong, but I don't think there is any way to know the optimal value of k without just testing with different values of k and seeing how good the answer is. As the paper says, one might try using several values of k and picking the best result, which is probably not very efficient since you end up doing more computations than you really want to. The paper also offers a heuristic that allows you to calculate the value of k that might give you a good result. Sometimes, it's better to just get a good enough solution rather than try to exhaust all possibilities to get the optimal solution.

Permalink Show parent Reply

Re: Paper Response: Velvet
by HUANG, ETHAN - Thursday, 13 May 2021, 5:24 PM PDT

R:
To get the absolute optimal k, I assume it would be very expensive because we'd need to try more than 1 value of k. As the paper stated, " In practice, given the limited number of possible values for k, it is common to try out various values in parallel then choose the one that produces the highest N50 contig length." This implies testing each of these possible values, how many of which is unknown.

Permalink Show parent Reply

Re: Paper Response: Velvet
by YEH, YU-WEI VINCENT - Thursday, 13 May 2021, 5:33 PM PDT

Response:
I would assume it wouldn't be that expensive since Velvet is quite efficient both memory-wise and time-wise. However, there's no doubt that testing multiple k's does seem like a lot as opposed to knowing the optimal k in the first place. As mentioned in the paper, genome, coverage, the quality, and the length of the reads could all influence the optimal parameter k. Therefore, with that many factors, optimal k varies case by case, so testing different values might be necessary.

Permalink Show parent Reply

Re: Paper Response: Velvet
by YEH, YU-WEI VINCENT - Thursday, 13 May 2021, 5:12 PM PDT

Question:
The paper mentioned that errors made by Velvet are mostly from the repeated region reconstruction done by Breadcrumb, and it seems to me that repeated region remains one of the biggest challenges in genome assembly. Is there any advanced algorithm created since 2008 in which repeats aren't the main sources of error? If so, how does the algorithm conduct corrections differently when compared to Velvet?

Permalink Show parent Reply

Re: Paper Response: Velvet
by HUANG, ETHAN - Thursday, 13 May 2021, 5:20 PM PDT

Q:

Towards the end in the methods section, it is mentioned that tour bus merging 2 paths is based on 3 thresholds, one of which is the "sequences must be at least 80% similar". What metric is used to define similarity, and 80% at that?

Permalink Show parent Reply

◄ Paper Response: Bowtie

◄ Announcements

Jump to...

Discussion 1A ►