How to use paired-end information for graph decomposition

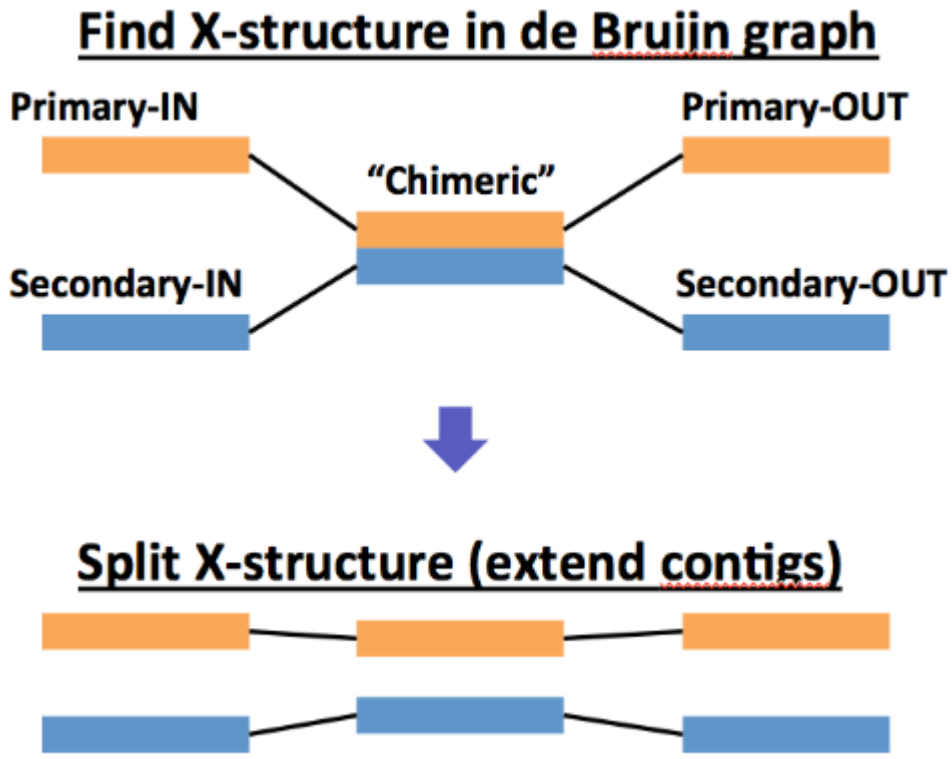
Afiahayati, Sato K, Namiki T, Hachiya T, Tanaka H, Sakakibara Y.

Abstract

Citation: Afiahayati, Sato K, Namiki T, Hachiya T, Tanaka H, Sakakibara Y. How to use paired-end information for graph decomposition. **protocols.io**
dx.doi.org/10.17504/protocols.io.d7p9mm
Published: 14 Jan 2016

Guidelines

In order to decompose a complecated metagenomic de Bruijn graph into simpler subgraphs, MetaVelvet searches the following X-structure, i.e., chimeric nodes have two incoming edges and two outgoing edges.



Then, MetaVelvet split the X-structures when the following conditions are satisfied:

- Primary-IN and Primary-OUT nodes are classified into the same peak.
- Secondary-IN and Secondary-OUT nodes are classified into the same peak.
- Chimeric node has a coverage value mostly equal (within 50% difference by default) to the average between the sum of coverage values of the two origin nodes of incoming edges (Primary-IN and Secondary-IN), and the sum of the two destination nodes of outgoing edges (Primary-OUT and Secondary-OUT).

In order to accurately avoid chimeric contigs/scaffolds caused by subgraph decomposition, we added the following two conditions (>= 1.2.01):

- The number of *consistent paired-end connections* is equal to or larger than a certain value. This cutoff value can be specified by the **-valid_connections** option (default: 1).
- The number of *inconsistent paired-end connections* is equal to or smaller than a certain value. This cutoff value can be specified by the **-noise_connections** option (default: 0).
- For lower compatibility, users can turn off the paired-end conditions by the **-use_connections** option.

Here, we denote paired-end connections between Primary-IN and Primary-OUT or between Secondary-IN and Secondary-OUT as *consistent paired-end connections*. We also denote paired-end connections between Primary-IN and Secondary-OUT or between Secondary-IN and Primary-OUT as *inconsistent paired-end connections*.

Based upon our accuracy evaluation, **-valid_connections 1 -noise_connections 0** is an appropriate setting when very similar species co-exist in an environment (MetaVelvet can efficiently avoid misassemblies). Otherwise, **-valid_connections 0 -noise_connections 0** is a more appropriate setting (MetaVelvet can achieve a greater contig/scaffold N50 while avoiding misassemblies).

Frequently Asked Questions

- Q: meta-velveth is not generated in the new version (>= 1.1.01). Is it problem?
  A: This is not problem. The usage of MetaVelvet is changed when the version 1.1.01 is released, and the new version does not include meta-velveth. Instead, please use velveth, velvetg, and meta-velvetg.

- Q: When only one coverage peak is detected (or manually input), is there any difference between MetaVelvet and Velvet algorithms?
  A: There is no substantial difference. In such cases, meta-velvetg moves to "single-peak mode" and graph splitting functions in meta-velvetg is not called. Instead of graph splitting functions, standard velvet functions are called in such cases.

- Q: What's the difference in working procedures between velvetg, meta-velvetg (<= 0.3.1), and meta-velvetg(>= 1.1.01)?
  A: The following is the working procedure of velvetg, meta-velvetg (<= 0.3.1), and meta-velvetg (>= 1.1.01):

  velvetg & meta-velvetg(<=0.3.1) :
  Load Sequences & Roadmaps file
  -> Generate PreGraph file
  -> Generate Graph or Graph2 file
  -> Generate contigs.fa and LastGraph

  meta-velvetg (>= 1.1.01):
  Load Sequences & Roadmaps & Graph2 file
  -> Generate meta-velvetg.contigs.fa and meta-velvetg.LastGraph

- Q: Is version compatibility between Velvet and MetaVelvet fully tested?
  A: Version compatibility between Velvet-1.0.06 and MetaVelvet-1.1.01 is fully tested.

Troubleshooting

Trouble: When drawing *k*-mer coverage histogram (as in the "Advanced topics 1" section), the following warning messages is appeared:

```
> weighted.hist(data$shot1_cov,data$lgth,breaks=seq(0,200,1))
Warning messages:
1: In min(x, na.rm = na.rm) :
no non-missing arguments to min; returning Inf
2: In max(x, na.rm = na.rm) :
no non-missing arguments to max; returning -Inf
3: In weighted.hist(data$shot1_cov, data$lgth, breaks = seq(0, 200, :
Areas will not relate to frequencies
```

Solution: This warning (error) is caused by "Inf" values in the Graph2 node stats. Accordingly, by removing "Inf" values from the Graph2 stats, the error is resolved:

```
$ head -n 1 meta-velvetg.Graph2-stats.txt \
      > meta-velvetg.Graph2-stats.rmInf.txt
$ perl -ne '{print $_ unless /Inf/;}' \
      meta-velvetg.Graph2-stats.txt \
      >> meta-velvetg.Graph2-stats.rmInf.txt
$ R
> library(plotrix)
> data = read.table("meta-velvetg.Graph2-stats.rmInf.txt", header=TRUE)
> weighted.hist(data$shot1_cov,data$lgth,breaks=seq(0,200,1))
```

Protocol