

Use Bambus2 scaffolding module

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

Abstract

Bambus2 is a scaffolding module that can be applicable to metagenomics settings. MetaVelvet uses a novel graph splitting algorithm during contigging process, and uses the scaffolding module of Velvet (RockBand and Pebble) during scaffolding process. Alternatively, users can output MetaVelvet contigs and uses Bambus2 scaffolding module instead of using RockBand and Pebble.

Citation: Afiahayati, Sato K, Namiki T, Hachiya T, Tanaka H, Sakakibara Y. Use Bambus2 scaffolding module. **protocols.io**

dx.doi.org/10.17504/protocols.io.d7n9md

Published: 14 Jan 2016

Guidelines

For installation and getting started protocol click [here](#).

Frequently Asked Questions

- **Q: meta-velveth** is not generated in the new version ($\geq 1.1.01$). 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** ($\leq 0.3.1$), and **meta-velvetg** ($\geq 1.1.01$)?
- **A:** The following is the working procedure of **velvetg**, **meta-velvetg** ($\leq 0.3.1$), and **meta-velvetg** ($\geq 1.1.01$):

velvetg & meta-velvetg ($\leq 0.3.1$) :

- 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

Step 1.

To install Bambus2 programs, please see their web site:

 [LINK:](#)

<http://www.cbcb.umd.edu/software/bambus/>

Step 2.

Execute **velveth**, **velvetg** as in the ["Getting started"](#) protocol. Please check that "out-dir/Sequences",

"out-dir/Roadmaps", and "out-dir/Graph2" files are created.

Step 3.

Execute **meta-velvetg** with **-amos_file** and **-scaffolding** options.

```
cmd COMMAND
~$ meta-velvetg out-dir -ins_length 260 \
    -exp_covs 214_122_70_43_25_13.5 \
    -amos_file yes -scaffolding no
```

📌 NOTES

Bonnie Hurwitz 16 Nov 2015

Please check that "out-dir/meta-velvetg.asm.afg" file is created. The file is **meta-velvetg** contigging result in an AMOS format.

Step 4.

Create amos bank.

```
cmd COMMAND
~$ bank-transact -m out-dir/meta-velvetg.asm.afg -b bank-dir -cf
```

Step 5.

Use the paired-end information to construct a collection of contig links.

```
cmd COMMAND
~$ clk -b bank-dir 2> /dev/null
```

Step 6.

Bundle the contig links into a collection of contig edges.

```
cmd COMMAND
~$ Bundler -b bank-dir -t M 2> /dev/null
```

Step 7.

Identify genomic repeats and output them.

```
cmd COMMAND
~$ MarkRepeats -b bank-dir \
    -redundancy num-pairs \
    2> /dev/null \
    | tee repeat-file
```

Step 8.

Order and orient contigs according to repeat and link information.

```
cmd COMMAND
~$ OrientContigs -b bank-dir \
    -redundancy num-pairs \
    -prefix output-prefix \
    -repeats repeat-file \
    -all \
    2> /dev/null
```

Step 9.

Linearize the scaffolds.

```
cmd COMMAND
~$ untangle -e output-prefix.evidence.xml \
    -s output-prefix.out.xml \
    -o output-prefix.untangle.xml \
    2> /dev/null
```

Step 10.

Output contig FastA file.

```
cmd COMMAND
~$ bank2fasta -d -b bank-dir \
    | tee contig-file
```

Step 11.

Output scaffold FastA file.

cmd **COMMAND**

```
~$ printScaff -merge -e output-prefix.evidence.xml \  
    -o output-prefix \  
    -s output-prefix.untangle.xml \  
    -l output-prefix.library \  
    -f scaffold-file \  
    2> /dev/null
```

Step 12.

Then, *scaffold-file* is the final scaffolding results in the FastA format.