User Manual

MaGuS

Map **Gu**ided **S**caffolding

Version 1.0

last update : sep 2015

Maintained by
Carole Dossat (magus@genoscope.cns.fr)

CEA-Genoscope, 2 rue Gaston Cremieux, 91000 Evry, France

Table des matières

1 Introduction

1.1 Map-Guided Scaffolding

The standalone package of MaGus (Map-Guided Scaffolding) is an automated modular pipeline wich is a powerful reference-free evaluator of quality assembly and an efficient map-guided scaffolder to improve assembly. It consists of a collection of Perl objects and libraries.

MaGuS performs a new scaffolding of one assembly, based on evidences provided by the anchoring of this assembly on a genome map and validated by high-throughput sequencing data.

1.2 Requirements

perl version 5.10.1 or higher installed

samtools version 0.1.19

SGA version 0.10.13

R version 3.0.2

exonerate version 2.4.0.20140227.git

getseq Genoscope tool

1.3 Installation

Download the current tarball archive from http ://www.genoscope.cns.fr/magus/download

```
$$ wget http://www.genoscope.cns.fr/magus/download/magus_latest.tar.gz
```

Untar/unzip the archive

```
$$ tar -zxvf magus_latest.tar.gz
$$ cd magus
```

Modify if needed the Perl interpreter that have been set to /usr/bin/perl

NB : MaGuS does NOT support character _ on fasta header files.

1.4 How to cite MaGuS

MaGuS

Madoui M.A., Dossat C., d'Agata L., van Oeveren J., van der Vossen E. and Aury J.M. : **MaGuS : A hybrid strategy to improve assembly by a genome Map-Guide Scaffolding.**

2 Quick start

2.1 The MaGuS pipeline

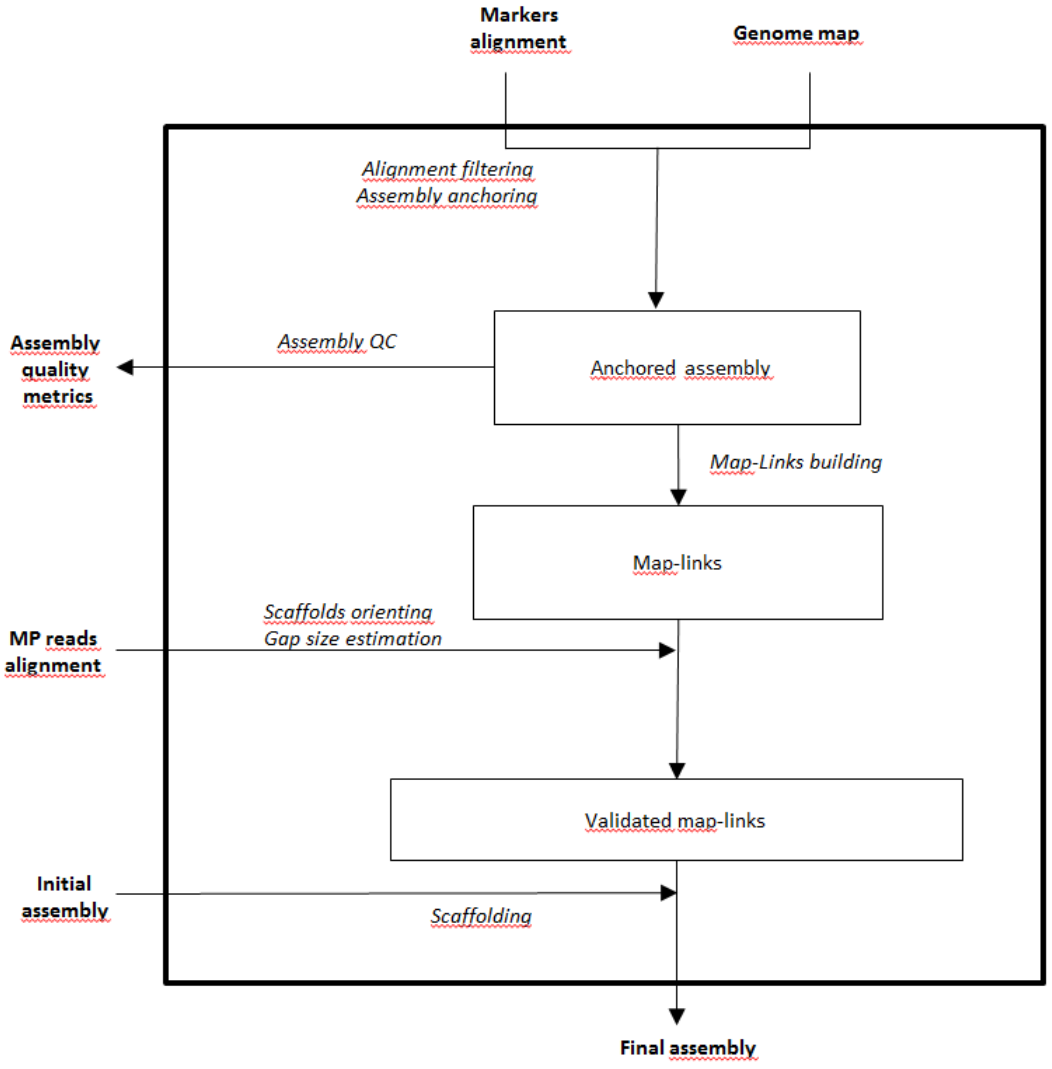

FIGURE *1 – MaGuS pipeline*

2.2 module all

This command runs all steps automatically. It is the simplest way for using MaGuS. You must provide all files needed by each module. See the example box to have a complete command line.

Help

```
magus all -h
```

Inputs

tagsWgp.out : file
tags provided in Keygene format.

tags.bam : file
tags alignments on draft assembly previously build in bam format. (You can use BWA for example).

mp1.bam : file
paired reads alignements on draft assembly previously build in bam format. (You can use BWA for example).

5350 : integer
mean size of paired reads bank in bp.

1000 : integer
standard deviation of the paired reads bank.

76 : integer
reads lenght.

assembly.fa : file
fasta file of draft assembly previously build.

119667750 : integer
genome size in bp.

Example

```
    magus all -wgp tagsWgp.out -tags tags.bam
-reads mp1.bam,5350,1000,76  -reads mp2.bam,5350,1000,76
-scaff assembly.fa -prefix arabido -genome 119667750
```

You can run this commande line with files provided on directory **Test**.

2.3 module wgp2map

Tags mapped to the draft assembly without multiple locations are selected in a BAM file.

Help

```
magus wgp2map -h
```

Inputs

-wgp [BAM file] : WGP tags mapped to the draft assembly *(required)*

-tags [Keygene format file] : physical WGP map provided by Keygene *(required)*

-prefix : prefix of the outputs *(optionnal, default magus)*

-sam : path for samtools *(optionnal, default is $PATH)*

Outputs

prefix_ordered_tags.txt : [tabular separated format] ordered tags on scaffolds

col1 : scaffold id
col2 : position of the tag on the scaffold
col3 : tag id
col4 : rank of the tag
col5 : contigBac id

prefix_anchorage.txt : [tabular separated format] scaffold anchoring file

col1 : contigBac id
col2 : scaffold id
col3 : minimum rank
col4 : maximum rank
col5 : number of tags

Example

```
magus wgp2map -wgp tagsWgp.out -tags tags.bam -prefix arabido
```

You can run this commande line with files provided on directory **Test** if samtools is in $PATH, else you have to use -sam.

2.4 module map2qc

This module calculates quality metrics.

TA : amount of tags mapped on the draft assembly. It informs about completeness of the assembly. (The highest is the best).

TC : number of tags adjacent in the map and in the assembly. It represents how the assembly fits to the map. (The highest is the best).

AnX : the length of the anchored segment for which X% of the anchored assembly contains anchored segments with a length over the AnX.

AnAX : the length of the anchored segment for which X% of the assembly contains anchored segments with a length over the AnAX.

AnGX : the length of the anchored segment for which X% of the genome contains anchored segments with a length over the AnGX.

MaGuS provides these metrics for three X values (50,75 and 90) and a graph with all values from 1 to 100.

Help

```
magus map2qc -h
```

Inputs

-scaff : fasta assembly *(required)*

-genome : genome size in db *(required)*

-order : [tabular separated format] ordered tags on scaffolds *(required, default prefix_ ordered_ tags)*
col1 : scaffold id
col2 : position of the tag on the scaffold
col3 : tag id
col4 : tag rank
col5 : contigBac id

-prefix : prefix of the outputs *(optionnal, default magus)*

-R : path for R *(optionnal, default is $PATH)*

-flen : path for fastalength *(optionnal, default is $PATH)*

Outputs

prefix_An.csv : flat file with metrics about AnX

prefix_AnA.csv : flat file with metrics about AnA

prefix_AnG.csv : flat file with metrics about AnG

prefix_quality_metrics.txt : flat file with quality metrics about the assemblies

prefix_quality_metrics.png : graphes of quality metrics

It generates somes graphes to represent these metrics.

2.5 module map2links

This module creates *map-links*, i.e links between adjacent scaffolds anchored on the genome map.

Help

```
magus map2links -h
```

Inputs

-anchor : [tabular separated format] anchored scaffold file *(required, default prefix_ anchorage.txt)*
col1 : contigBac id
col2 : scaffold id
col3 : minimum rank
col4 : maximum rank
col5 : number of tags

-prefix : prefix of the outputs *(optionnal, default magus)*

Outputs

arabido_map_links.txt : map-links file. Links are represented by two ids separated by an underscore. col1 : scaffoldId1_scaffoldId2

Example

```
magus map2links -prefix arabido
```

2.6 module pairs2links

This module validates *map-links* with paired-end reads and estimates gap size by using median size, standard deviation of the library fragment.

Help

```
magus pairs2links -h
```

Inputs

-reads : bam file name, library fragment size mean, library fragment size standard deviation, reads size *(required)*

-scaff : assembly file in fasta format *(required)*

-links : links are represented by two ids separated by an underscore *(required, default prefix_map_links.txt)* col1 : scaffoldId1_scaffoldId2

-prefix : prefix of the outputs *(optionnal, default magus)*

-sam : path for samtools *(optionnal, default is $PATH)*

-flen : path for fastalength *(optionnal, default is $PATH)*

Outputs

prefix_validated_map_links.de : validate links in de format. See details for this format on the web.

prefix_validated_map_links.log : assembly metrics (text file).

prefix_unvalidated_map_links.txt : unvalited links list (two ids separated by an underscore). col1 : scaffoldId1_scaffoldId2

Example

```
magus pairs2links
-reads mp1.bam,5350,1000,76 -reads mp2.bam,5350,1000,76
-scaff assembly.fa -prefix arabido
```

2.7 module links2scaf

This module builds the final scaffolds.

Help

```
magus links2scaf -h
```

Inputs

-connect : file with the links in de format *(optionnal, default prefix_validated_map_links.de)*

-scaff : fasta assembly *(required)*

-prefix : prefix of the outputs *(optionnal, default magus)*

-sga : path for sga *(optionnal, default is $PATH)*

-getseq : path for getseq *(optionnal, default is $PATH)*

Outputs

prefix_sga.scaf : output from SGA program

prefix_sga_scaffold.log : log about SGA run

prefix_scaffolds.fa : fasta file provided by SGA program

prefix_sga_scaffold2fasta.log : fasta file provided by SGA program

prefix_sga_scaf_lost.txt : list of scaffolds lost by SGA

prefix_lost_scaffolds.fa : scaffolds lost by SGA in fasta file

prefix_all_scaffolds.fa : final assembly in fasta file

Example

```
    magus all -wgp tagsWgp.out -tags tags.bam
-reads mp1.bam,5350,1000,76 -reads mp2.bam,5350,1000,76
-scaff assembly.fa -prefix arabido -genome 119667750
```

3 References

samtools

Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. : **The Sequence Alignment/Map format and SAMtools.** *Bioinformatics* 2009, **25**(16) :2078-2079

SGA

Simpson J.T.,Durbin R. : **Efficient de novo assembly of large genomes using compressed data structures** *Genome research* 2012, **22**(3) :549-556

fastalength

It comes from the exonerate software availaible under the LGPL license.
See **http ://www.ebi.ac.uk/ guy/exonerate/** for details.

R

See **cran.r-project.org** for details.

getseq

It is a Genoscope tool, provided with ExtraSeq and FileTools libraries.

bam format

See **https ://samtools.github.io/hts-specs/SAMv1.pdf** for details.

de format

Simpson J.T.,Durbin R. : **Efficient de novo assembly of large genomes using compressed data structures** *Genome research* 2012, **22**(3) :549-556