Jul 31, 2020 Version 1

construct a taro linkage map using onemap V.1

DOI
dx.doi.org/10.17504/protocols.io.85xhy7n

Michael Kantar[1], M Renee Bellinger[2], Roshan Paudel[1]

[1]University of Hawaii at Manoa; [2]University of Hawaii at Hilo

M Renee Bellinger
University of Hawaii at Hilo

OPEN ACCESS

DOI: dx.doi.org/10.17504/protocols.io.85xhy7n

Protocol Citation: Michael Kantar, M Renee Bellinger, Roshan Paudel 2020. construct a taro linkage map using onemap. **protocols.io** https://dx.doi.org/10.17504/protocols.io.85xhy7n

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working
This protocol was implemented in the G3 paper "Taro Genome Assembly and Linkage Map Reveal QTLs for Resistance to Taro Leaf Blight". Authors: M Renee Bellinger, Roshan Paudel, Steven Starnes, Lukas Kambic, Michael Kantar, Thomas Wolfgruber, Kurt Lamour, Scott Geib, Sheina Sim, Susan Miyasaka, Martin Helmkampf, Michael Shintaku. Publication Date: August 2020.

Created: November 08, 2019

Last Modified: July 31, 2020

Protocol Integer ID: 29591

Abstract

R script for taro linkage mapping using program onemap.

This script was applied to create a linkage map for a taro mapping population resistant to taro leaf blight.

The genotype file was constructed by calling SNPs using genotyping by sequencing data set using a taro genome reference genome available from NCBI's genbank under Bioproject PRJNA567267.

Attachments

TXT

imputed_loci_0605201...

545KB

Before start

Requires R version >3.5 or >3.6
download this file imputed_loci_06052019.vcf.txt and re-name it imputed_loci_06052019.vcf

Set up your R working environment

1

```
### There are two ways to install onemap

###-------------------###
### Option 1.
####Download and install Rtools in your computer before running this script
###https://cran.r-project.org/bin/windows/Rtools
find_rtools(debug = TRUE)

###Install required packages
install.packages("devtools")
library(devtools)

### The above installation is supposed to run in R version to 3.5 but I had issue with
getting it to install.
devtools::install_github("Cristianetaniguti/onemap")
library("onemap")

###-------------------###
### Option 2. If using R 5.6 this package is compatible with the CRAN version
###
install.packages(onemap)
libarary(onemap)


install.packages("vcfR")
library("vcfR")


#######Set working directory
setwd("/your/working/directory/")

#####Convert the vcf file into onemap format.
## BE SURE THE txt is removed from the end of the vcf file name

vcfR.object ← read.vcfR("imputed_loci_06052019.vcf")
example_out ← onemap_read_vcfR(vcfR.object = vcfR.object,
                        parent1 = "P230.sam.mapped.bam",
                        parent2 = "P255.sam.mapped.bam",
                        cross = "outcross")
```

```
####Get the information on marker genotypes.
plot(example_out)

###Number of markers by segregation type
plot_by_segreg_type(example_out)
###Find redundant markers
bins ← find_bins(example_out, exact = FALSE)
bins
###Create onemap file after removing redundant markers
bins_example ← create_data_bins(example_out, bins)
bins_example
###testing segregation pattern of the markers
segreg_test ← test_segregation(bins_example)
print(segreg_test)
#####to show the markers names with segregation distortion
select_segreg(segreg_test, distorted = TRUE)
###to show the markers numbers with segregation distortion
dist ← select_segreg(segreg_test, distorted = TRUE, numbers = TRUE)
#to show the markers numbers without segregation distortion
no_dist ← select_segreg(segreg_test, distorted = FALSE, numbers = TRUE)
no_dist
####See the segregation test graphically
plot(segreg_test)
##calculate a suggested LOD score
LOD_sug ← suggest_lod(bins_example)
LOD_sug
###Estimating two-point recombination fractions
twopts ← rf_2pts(bins_example, LOD = LOD_sug, max.rf = 0.3)
twopts
###Create the sequences of markers without segregation distortion
mark_no_dist ← make_seq(twopts, c(no_dist))
###Number of linkage groups
lg ← group(mark_no_dist, LOD=LOD_sug, max.rf = 0.3)
print(lg, detailed = FALSE)
###Estimate a genetic map
maps←vector("list", lg$n.groups)
for(i in 1:lg$n.groups)
  maps[[i]]← make_seq(order_seq(input.seq= make_seq(lg,i),twopt.alg =
                                     "rcd"), "force")
##Draw the maps
##Draw the maps
draw_map(maps, names = FALSE, grid = TRUE, cex.mrk = 0.7)
```

```
draw_map2(maps,output="maps.png")

####Name the map and extract it into external file
file.out<-"map_LOD=5.97,max.rf=0.3.csv"
write_map(map1, file.out)
```