# Tassel\_to\_J pipeline

## Disclaimer

This pipeline was developed specifically for the production of a coconut high density linkage map and is not intended for distribution. It should be considered as a demonstration of the procedures used in the above project and as a work in progress.

## Description

This pipeline aims to producing a data file usable for Joinmap starting from the VCF file produced by Tassel. It comes in the form of five R markdown files (one per step) and, in addition of a number of output files, each part produces a report. Both R software and R Studio need to be installed. These are free software and can be downloaded at the following sites.

<https://cran.r-project.org/>

<https://www.rstudio.com/>

In principle, the R markdown files need *not* be modified. They receive instructions from two R scripts which do nothing but assign variables. The first one is tassel\_to\_J\_parameters.R, which simply says which parameter file to use and where it is. The other one is the parameter file itself eg. first\_round.R, which assigns values to variables, which serve as parameters. The meaning of the variables is given as comments, as well as suggested values. Neither the names nor the order of the variables should be changed, but the values of the variables depend on the job being executed. The parameters may represent

* the path to an existing file is or where a new file has to be created,
* the name of a file,
* the value of a parameter used in calculations,
* the desired coding system,
* free comments to be included at the beginning of the report. They may include Latex code.

Report problems and suggestions to [luc.baudouin@cirad.fr](mailto:luc.baudouin@cirad.fr)

## Overview

The situation considered here is the following: a mapping population (in the present version, only BC1 populations are supported) was genotyped by sequencing (GBS) and a genomic scaffold library is available. The sequences generated by GBS are aligned onto scaffolds and SNPs are called by Tassel or any suitable software. The final goal is to produce a linkage map where the order of markers on scaffolds is preserved. This can be achieved by using Joinmap to calculate recombination rates and LOD scores, followed by Scaffhunter, which, unlike Joinmap, considers the position of loci on scaffolds. In between, the data need to be filtered in order to limit the number of missing values. This is the role of Tassel\_to\_J (figure 1). It consists in five steps.



Figure 1 Place of Tassel\_to\_J in the production of a linkage map using GBS



Figure 2 The five steps of the Tassel\_to\_J pipeline

## Preliminary steps

### Input file

Before using Tassel\_to\_J, the VCF file needs to be converted into a tabular text file. Columns are separated by tabulations. The first row is interpreted as column names.

There must be three columns to the right for the markers name, the scaffolds and the position of the marker on the scaffold. The column names can be changed, but this must be reflected in the parameter file. Then come the genotype data under two possible formats (see below). The last two columns are for the P values and the χ2 scores.

### genotype formats

In the columns devoted to genotypic data, the first row is a genotype identifier. The rest of the columns represents the two bases scored at each SNP. They may be homozygous (A/A, C/C, G/G or T/T) or heterozygous (A/C, A/G, A/T, C/A etc.). Note that ‘A/C’ and ‘C/A’ are interchangeable (i.e. phase is not taken into account). Finally, missing data are represented by ‘./.’. Optionally, additional data ( eg. coverage) can be added after a colon ‘:’ These data are ignored. See examples in tables 3-1 and 3-2 at the end of the document.

## Executing tassel\_to\_J

It’s a good practice to begin by creating an R studio project (possibly with version control). Each time you open the project, the working folder will be the same. If you place all your working files in descending folders, you can use relative paths. Executing the pipeline starts with creating a parameter file. The safest is to copy an existing parameter file as template and changing the variable assignments wherever needed. The provided parameter files have a lot of comment lines providing guidelines. The other lines are assignment commands in R. The user can also add his/her own comments.

Parameters can be of different types.

* character. Use quotes eg. "titi"
* numeric eg. 25
* logical. Either TRUE or FALSE

Two parameters are used in all steps:

* draw\_plot: logical. Should normally be TRUE. However, in some situation (involving an apple computer and R executed on an UNIX station), attempting to draw plots provoked an error.
* comment0: character. This comment will be printed at the beginning of all reports. It may contain Latex code. See comments in the template for more details.

Before running ‘tassel\_to\_J’, you will have to tell it which parameter file to use. This is done by including an R ‘source’ command in tassel\_to\_J.parameter. There should be only one source command in the file, but you can keep track of previous jobs by adding a ‘#’ character at the beginning of the line (making it a comment) and adding a new ‘source’ command.

### Step 1

#### Description

Imports one or more input files and creates two types of files: One dedicated to genotype data and one to information about markers. Using multiple input files may be useful with large data sets. It allows parallel treatment and it may save memory.

#### Parameters

* comment1: character. An optional comment which will be printed at the beginning of the report produced by this step.
* where\_in: character. A valid path to the folder containing the initial data file(s)
* file.in: character. The name(s) of input file(s). To use several files, they should be entered as a vector. The syntax is c("file1", "file2", "file3").
* where\_out\_1: character. A valid path to the folder where the files produced at this step will be stored.n
* file.value: character. Name of the genotype file created at this step. Vector length must be the same as for file.in.
* file.ident: character. Name of the marker information file created at this step. Vector length must be the same as for file.in.
* P.value.min: numeric. P-value cut-off value. Markers whose P-value is lower are discarded. A reasonable value could be 1/(2\* number of rows). Use negative values to cancel the filter.
* type: character. Cross type (currently, only "BC1" is supported)
* parent: a two-element vector such as c("parent1","parent2"). Two values are required (In BC1, the heterozygous parent comes first).

The following two logical parameters (either TRUE or FALSE) state whether the parental genotypes are included in the present data set. At least one of them must have the value ‘TRUE’.

* has.hetero: for the heterozygous parent
* has.homo: for the homozygous parent
* others: character. Vector containing the names of genotypes which should not be included in the analysis eg. controls or aberrant individuals. if equal to NULL,all genotypes will be analysed.

The following three parameters are the names of the first three columns of the input file

* marker\_column: character
* scaffold\_column: character
* position\_column: character

### step 2

#### Description

At this stage, the genotypes are assigned to the parents. Only loci whose segregation is compatible with the mating plan are retained. Statistics are computed on the disjunction and the results are written into the locus info file. The columns of the genotype table are reordered. First the parents, then the progenies in order of increasing number of missing data. All genotyping files and all locus information file are merged at this step.

#### Parameters

Input files are the output files of step 1

* comment2: character. An optional comment which will be printed at the beginning of the report produced by this step.
* where\_out\_2: character. A valid path to the folder where the files produced at this step will be stored.
* file.value.2: character. Name of the genotype file created at this step (only one value).
* file.ident.2: character. Name of the marker information file created at this step (only one value).
* where\_functions: character. A valid path to the folder where the R functions are located. Normally, it should be "functions".

The following parameters represent thresholds used for filtering individuals and markers.

* cut.ind: numeric. Individuals having less than 100\*cut.ind % good data are discarded. This first test should be liberal (eg. 0.5) because more tests will come later.
* Markers which do not meet the following criteria are flagged for elimination at a later stage.
* balanced: numeric. Minimal frequency of the minority allele.
* few\_good\_data: numeric. Minimal proportion of valid data (excluding missing and unexpected alleles).
* many\_wrong: numeric. Maximum proportion of unexpected alleles.

### Step 3

#### Description

Codes the genotype file in a form usable by mapping software. Parents are removed from the output file.

#### Parameters

Input files are the output files of step 2

* comment3: character. An optional comment which will be printed at the beginning of the report produced by this step.
* where\_out\_3: character. A valid path to the folder where the files produced at this step will be stored.
* file.value.3: character. Name of the coded genotype file created at this step (only one value).
* file.ident.3: character. Name of the marker information file created at this step (only one value).
* genotype\_code: character. A four letter code to describe the genotypes (preferred for Joinmap: "ABHU" or "ABH-").
* first letter: a homozygous parental genotype
* second letter: the other homozygous parental genotype (not used in BC1)
* third letter: the heterozygous genotype
* fourth letter: unknown genotype

### Step 4

#### Description

The purpose of this step is to automatize the selection of useful markers. There are two main issues: redundant markers and outliers ie. markers that segregate independently from the other loci of the same scaffold (might represent chimeric scaffolds or incorrect assignment of marker to scaffold). At the end of the report, a figure and a table allow removing individuals (at step 5) in order to optimize the data matrix.

#### Parameters

Input files are the output files of step 3

* comment4: character. An optional comment which will be printed at the beginning of the report produced by this step.
* where\_out\_4: character. A valid path to the folder where the files produced at this step will be stored.
* reduced\_data: character. name of the genotype file created at this step
* reduced\_info: character. name of the marker information file created at this step
* indiv\_select: character. Name of the file that lists all individuals to be considered for removal at the last step.

Filtering thresholds

* t\_outlier: numeric. Threshold used (with ‘dist’) to select outliers. Suggested value: a little less than N/2-sqrt(N) where N is the number of descendents.
* l\_gap: numeric. Threshold used (with ‘f2’) to identify large gaps. Suggested value: 0.05
* majority: numeric. threshold used to validate the most frequent allele for imputation. Suggested value: 0.8
* stop\_remove: numeric. Number of individuals to include in the last figure and table of the report. Used to identify individuals removed at step 5.
* max\_miss\_loc: numeric. Minimal acceptable proportion of missing data. Suggested value: 0.2

### Step 5

#### Description

Last step: Examine last figure of previous step and chose the number of individuals to eliminate. This step takes one data and one information file. Another file set with less individual and less loci (but also less missing data) is produced.

#### Parameters

Input files are the output files of step 4

comment5: character. An optional comment which will be printed at the beginning of the report produced by this step.

* where\_out\_5: character. A valid path to the folder where the files produced at this step will be stored.
* file.value.5: character. Name of the genotype file created at this step (only one value).
* file.ident.5: character. Name of the marker information file created at this step (only one value).
* removed: numeric. The number of individuals to be removed. Should be determined based on the inspection of the last figure and table of the step 4 report.

## Notes relating to R Studio

Tassel\_to\_J will not work properly is the working directory is not located where the five .RMD files are. Go to the “*files*” tab, find the right folder and click on “*More|Set As Working Folder*”. A good idea is to create a project using use version control etc. (optional).

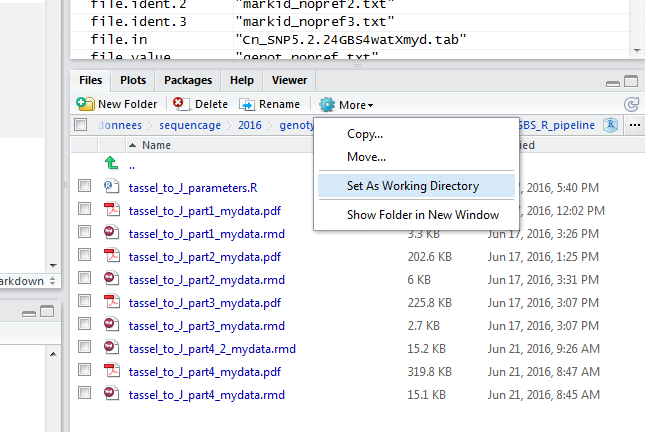


Figure 3 Set RStudio Working Directory

All .rmd scripts should be run in the proposed After opening a script (file/open file…), execute it by clicking on “knit pdf”.

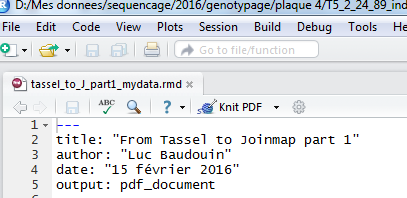


Figure 4 RStudio Knit PDF

## Appendix 1: File organization.

The relative paths to the folders are provided. The folder where the pipeline is loaded is indicated by a dot (.). Make sure all files required for execution is present

Table 1 File organisation for Tassel\_to\_J R pipeline

|  |  |  |
| --- | --- | --- |
| Folder | File name |  |
| Files used for project management (not used) | | |
| . |  |  |
|  | *Curation-GBS-data.Rproj* |  |
|  | *LICENSE* |  |
|  | *README.md* |  |
| Files required for execution | | |
| . |  |  |
|  | tassel\_to\_J\_parameters.R |  |
|  | first\_round.R |  |
|  | tassel\_to\_J\_part1.rmd |  |
|  | tassel\_to\_J\_part2.rmd |  |
|  | tassel\_to\_J\_part3.rmd |  |
|  | tassel\_to\_J\_part4.rmd |  |
| functions |  |  |
|  | cleanup\_data2.R |  |
|  | convert\_data.R |  |
|  | correctGenotype.R |  |
|  | distance.R |  |
|  | do\_correct.R |  |
|  | impute.R |  |
|  | multiple\_check.R |  |
|  | multiple\_check\_all.R |  |
|  | pair\_check.R |  |
|  | pair\_check\_all.R |  |
|  | posit\_loc.R |  |
|  | reassign\_chosen.R |  |
|  | scaffoldRule.R |  |
|  | select\_ind.R |  |
|  | subset\_data.R |  |
| data/set1 |  |  |
|  | first\_round.tab |  |
| Files created during execution | | |
| . |  |  |
|  | tassel\_to\_J\_part5.rmd |  |
|  | tassel\_to\_J\_part1.pdf |  |
|  | tassel\_to\_J\_part2.pdf |  |
|  | tassel\_to\_J\_part3.pdf |  |
|  | tassel\_to\_J\_part4.pdf |  |
|  | tassel\_to\_J\_part5.pdf |  |
| data/set1/step1 |  |  |
|  | genot\_first\_round.txt |  |
|  | markid\_first\_round.txt |  |
| data/set1/step2 |  |  |
|  | genot\_first\_round\_2.txt |  |
|  | markid\_first\_round\_2.txt |  |
| data/set1/step3 |  |  |
|  | genot\_first\_round\_3.txt |  |
|  | markid\_first\_round\_3.txt |  |
| data/set1/step4 |  |  |
|  | indiv\_select.txt |  |
|  | newinfo.txt |  |
|  | reduced\_data.txt |  |
|  | reduced\_info.txt |  |
|  | stat\_ind.txt |  |
|  | stat\_loc.txt |  |
| data/set1/step5 |  |  |
|  | data\_joinmap.txt |  |
|  | info\_loc.txt |  |

## Appendix 2: Files created by the pipeline

This appendix describes all files produced by Tassel\_to\_J. All files are text files, delimited by tabulations, which can be open indifferently with a text editor or by Excel. These files are located in different folders (‘step 1’, ‘step 2’ etc.). File names and paths can be changed in the parameter file. We recommend using this faculty with discretion.

* it is a good idea to include a “job name” in the file names.
* It is also a good idea to include a “version number” if you run part of the workflow again with modified parameters (otherwise you will overwrite your previous work).
* It is definitely a bad idea to rename one of these file from the file explorer!
* Keep in mind that the input files of step *n* are the output files of step *n*-1.

### Step 1

#### Genotype file

example genot\_first\_round.txt

Description:

* first column : marker name
* first row : genotype name
* genotypes in the form T/T, A/G, etc.

#### Marker identification file

example markid\_first\_round.txt

Description:

* first row : header
* columnns
  + marker: marker name
  + scaffold: scaffold name
  + position: position of the marker on the scaffold
  + P.value: P-value of the allelic distribution with a binomial model.
  + ChiSquare: χ2 score for of the allelic distribution with a binomial model.

### Step 2

### Genotype file

example genot\_first\_round\_2.txt

Description : similar to the one of step 1 except for the order of the columns:

* marker names
* F1 hybrid parent
* Recurrent parent
* Progenies, ranked by decreasing percentage of valid data. Individuals with less than parameter (adjustable parameter) good data are discarded.

### Marker identification file

example markid\_first\_round\_2.txt

Description : same as in step 1 but with additional columns

* first row : header
* columns
  + marker : marker name
  + scaffold : scaffold name
  + position : position of the marker on the scaffold
  + P.value : P-value of the allelic distribution with a binomial model.
  + ChiSquare : χ2 score for of the allelic distribution with a binomial model.
  + Hyb : Genotype of the F1 hybrid
  + Dwarf : [sic] genotype of the homozygous parent
  + D : number of homozygous genotypes
  + H : number of heterozygous genotypes
  + M : number of missing genotypes
  + wrong : number of unexpected genotypes (given the parent genotypes)
  + total : number of individuals retained at step 2
  + good : sum of ‘D’ and ‘H’
  + f\_good : previous column divided by ‘total’
  + is\_good : TRUE if ‘f\_good’ exceeds parameter ‘few\_good\_data’
  + f\_D : ‘D’ divided by ‘total’
  + balanced : TRUE if ‘min(f\_D, 1-f\_D)’ exceeds parameter ‘balanced’
  + third\_allele : TRUE if ‘wrong’ >0

### Step 3

#### Genotype file

example genot\_first\_round\_3.txt

Description : Same as in step 2 except that the parental genotypes are coded as ‘A’ (same as in recurrent parent) and ‘H’ (same as in F1 hybride parent). All other genotypes are coded ‘U’ (unknown).

#### Marker identification file

example markid\_first\_round\_3.txt

Description : same as in step 2 but only markers which meet the ‘good’ and ‘balanced’ conditions are kept.

### Step 4

#### Statistics about loci

Example: stat\_loc.txt

Description:

* index : row numbering
* loc : locus name
* a, A, b, B, h, H, u, U : number of genotypes presenting the said code at a given locus. (see previous file for details). If the code is in upper case, it is the original data. If it is in lower case, it results from a correction or from an imputation. Note that in the present version, only the BC1 case is considered. Therefore, the ‘b’ and ‘B’ columns should be empty.

#### Statistics about genotypes

Example stat\_ind.txt

Description: Same as before except the statistics are calculated per individual.

#### individuals considered for removal

example indiv\_select.txt

Description: statistics about the individuals with the largest amount of missing data. To improve the quality of the data matrix, we have to compromise between removing individuals and removing markers. Normally, the choice is made based on a table printed at the end of the fourth step.

Some columns appear to be useless…

* index : row numbering
* loc : same as before
* size : Number of genotypes remaining
* ind1: (used in calculations)
* ind2: (used in calculations)
* loc: number of loci which will be used for mapping
* scaff: number of scaffold which will be used for mapping
* remove: identity of the genotype.

Detailed statistics for all loci considered in this step

example newinfo.txt

Description: information generated during the correction and imputation process

* index: row number
* loc: locus name
* scaffold: scaffold name
* locus: duplicates ‘loc’
* posit: position of locus on scaffold.
* target: the closest locus. distances are calculated between the locus and the target
* dist: this distance takes into account missing data and is the expectation of the number of recombinations *under the hypothesis that the loci are independent.* This is the criterion for identifying outlayers.
* f1: probability of recombination under the same hypothesis. it is ‘dist’ divided by the number of individuals.
* f2: recombination rate ignoring missing data. It is the criterion used for identifying redundants.
* signif: Number of genotypes with significant data *before* imputation.
* select: classes of loci
  + chosen: marker selected in a set of redundant markers.
  + redundant, quasi\_redundant, swapped: all discarded because they are redundant. Their genotype is still represented in the final data by a ‘chosen’ locus.
  + single, small, large: loci with unique genotype.
  + outlier: discarded because it is too distant from the other loci of its scaffolds.
* parent: (used in calculations)
* swap: (used in calculations)
* new\_signif: Number of genotypes with significant data *after* imputation.

#### Filtered locus information

example: reduced\_info.txt

description: same structure as above but outliers and (quasi) redundant loci are removed.

#### Filtered genotypic data

example: reduced\_data.txt

Description: same structure as the genotype file in step 3. Only loci in ‘reduced\_info.txt’ are used.

## Appendix 3: input file format

Table 2 Data table in the short format

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Marker | scaffold | position | C409 | C646 | C404 | C645 | C524 | More genotypes | P-value | ChiSquare |
| CN\_C69366023M359 | CN\_C69366023 | 359 | T/T | T/T | T/C | T/C | T/T | … | 0.27258268 | 1.2037037 |
| CN\_C69366149M946 | CN\_C69366149 | 946 | G/G | G/A | G/A | G/A | G/A | … | 0.14259232 | 2.14975845 |
| CN\_C69366295M1852 | CN\_C69366295 | 1852 | G/G | G/G | G/G | G/A | ./. | … | 0.61012016 | 0.26 |
| CN\_C69367833M1225 | CN\_C69367833 | 1225 | C/C | C/G | C/G | C/C | ./. | … | 1.57E-05 | 18.6551724 |
| CN\_C69367833M1278 | CN\_C69367833 | 1278 | C/C | C/G | C/G | C/G | C/G | … | 0.40341841 | 0.69811321 |
| CN\_C69370447M1909 | CN\_C69370447 | 1909 | A/C | A/C | A/C | A/A | A/C | … | 0.81004337 | 0.05777778 |
| CN\_C69370447M1980 | CN\_C69370447 | 1980 | T/T | C/T | C/T | C/C | C/T | … | 0.5451535 | 0.36607143 |
| CN\_C69373359M1568 | CN\_C69373359 | 1568 | ./. | C/G | C/G | C/G | ./. | … | 0.1601071 | 1.97321429 |
| CN\_C69373359M1571 | CN\_C69373359 | 1571 | ./. | T/C | T/C | T/C | ./. | … | 0.1601071 | 1.97321429 |
| CN\_C69373549M1553 | CN\_C69373549 | 1553 | ./. | C/C | ./. | C/C | G/G | … | 0.43695039 | 0.60427808 |
| CN\_C69373613M739 | CN\_C69373613 | 739 | A/A | A/A | A/A | A/T | ./. | … | 0.00010825 | 14.987013 |
| More markers | … | … | … | … | … | … | … | … | 0.18530632 | 1.75454546 |

Table 3 Data in the long format

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Marker | scaffold | position | C520 | C636 | C483 | C486 | C401 | More genotypes | P-value | ChiSquare |
| Cn\_s000006M2012 | Cn\_s000006 | 2012 | ./.:4,0 | ./.:4,0 | G/G:6,0 | G/G:12,0 | ./.:1,0 | … | 9.38E-13 | 50.969697 |
| Cn\_s000006M2015 | Cn\_s000006 | 2015 | ./.:4,0 | ./.:1,0 | G/T:3,3 | G/G:12,0 | ./.:1,0 | … | 3.92E-11 | 43.6545455 |
| Cn\_s000006M2023 | Cn\_s000006 | 2023 | ./.:0,4 | ./.:0,1 | T/T:0,6 | T/T:0,12 | ./.:0,1 | … | 2.75E-07 | 26.4181818 |
| Cn\_s000006M2026 | Cn\_s000006 | 2026 | ./.:0,4 | ./.:0,1 | G/G:0,6 | G/G:0,12 | ./.:0,1 | … | 1.72E-07 | 27.3214286 |
| Cn\_s000006M2027 | Cn\_s000006 | 2027 | ./.:4,0 | ./.:1,0 | A/A:6,0 | A/A:12,0 | ./.:1,0 | … | 3.68E-12 | 48.2857143 |
| Cn\_s000006M2040 | Cn\_s000006 | 2040 | ./.:0,4 | ./.:0,1 | T/T:0,6 | T/T:0,12 | ./.:0,1 | … | 1.47E-05 | 18.7818182 |
| Cn\_s000006M2046 | Cn\_s000006 | 2046 | ./.:4,0 | ./.:1,0 | T/T:6,0 | T/T:12,0 | ./.:1,0 | … | 3.22E-13 | 53.0701754 |
| Cn\_s000006M2047 | Cn\_s000006 | 2047 | ./.:4,0 | ./.:1,0 | T/T:6,0 | T/T:12,0 | ./.:1,0 | … | 3.22E-13 | 53.0701754 |
| Cn\_s000006M2048 | Cn\_s000006 | 2048 | ./.:0,4 | ./.:0,1 | A/A:0,6 | A/A:0,12 | ./.:0,1 | … | 1.11E-06 | 23.7272727 |
| Cn\_s000006M2067 | Cn\_s000006 | 2067 | ./.:4,0 | ./.:1,0 | C/C:6,0 | C/C:12,0 | ./.:1,0 | … | 3.22E-13 | 53.0701754 |
| More markers | … | … | … | … | … | … | … | … | 5.35E-13 | 52.0714286 |