**Appendix S6**. MAKER2 workflow (original papers: Cantarel et al., 2008; Holt and Yandell, 2011; Campbell et al., 2013).

The following is a description of our MAKER2 workflow, intended to guide the reader through an analysis of genomic data using the MAKER2 Annotation Pipeline. This example protocol reflects our own experience with installing and using the program on CentOS Linux and thus is not a guaranteed guide to getting MAKER2 up and running on your own machine. The definitive authority for this purpose is the MAKER2 page on the Generic Model Organism Database (GMOD) website (<http://gmod.org/wiki/MAKER>). This website has a list of programs that need to be installed for MAKER2 to work and also has a good tutorial on installing and running MAKER2 (<http://gmod.org/wiki/MAKER_Tutorial>). There is even a video course on YouTube

(<http://www.youtube.com/watch?v=uA96tSSaqLk&feature=youtu.be>).

Everything for MAKER2 will be run from the command prompt in a terminal window. The conventions of this example protocol are that the command prompt (typically some type of login information and the current directory followed by a dollar sign or some other character) will be in blue and the text for the command to be typed will be in black. Below is an example for running the command ‘maker –CTL’.

[paul@wolfelab ~]$ maker –CTL

**1) Installation**

A brief note on installing MAKER2: You may notice after looking at the GMOD website that there is an “Easy Install” option for MAKER2. Don’t use it. In our experience, it did not work and also put the executables for all of the programs in places that we couldn’t find on the computer. It is much better to download each program individually and to put them and install them in a place that is easy to find so that you know where they are. This is very important because you will have to tell MAKER2 where to find everything. This obviously can’t be done if you don’t know where all of your programs are located. We used the desktop to house all of our program folders (this allows you to actually see them). Another good option would be your home folder.

Programs will generally have either a README or INSTALL file in the main folder with instructions on setting up the program. A common method of installation uses three commands: ‘make’, ‘make check’, and ‘make install’. This won’t be the case for all programs but many have this framework.

The last step after installing all of the programs is to tell MAKER2 where to find them. This is done by adding the location of the relevant executables to your $PATH variable. This can be done in two ways. The first way is to export them to the $PATH variable via the command line using a command such as this:

[paul@wolfelab ~]$ export PATH=$PATH:/path/to/program1:/path/to/program2

with the appropriate path information substituted for ‘/path/to/program#’. This would need to be done for every program every time you open a new terminal window. You can specify all the programs in one command, however.

A second way would be to add the paths to the executables by modifying the .bash\_profile in your home folder. This will allow you to permanently add the executables needed by MAKER2 to your search path without having to specify where they are every time you open a new terminal window. Your .bash\_profile can be modified in any text editor, for example in emacs:

[paul@wolfelab ~]$ emacs .bash\_profile

Then, in the text editor just type:

PATH=$PATH:/path/to/program1:/path/to/program2:/path/to/program3:

path/to/program4

**2) Running MAKER2**

With everything properly installed and ready, running MAKER2 is quite straightforward. One easy way to manage MAKER2 runs is to create separate folders for each run you do. You can then run MAKER2 from inside them, and all of the output will be placed in the folder without getting confused with other MAKER2 runs in other folders. For the sake of example, we’ll make a directory called ‘maker\_runs’ and within that we can make folders for all of our runs (‘run1’, ‘run2’, etc.). We can then simply go to the folder where we want to run MAKER2 and type:

[paul@wolfelab ~/maker\_runs/run1]$ maker -CTL

This will generate three control files, ‘maker\_opts.ctl’, ‘maker\_bopts.ctl’, and ‘maker\_exe.ctl’. These files will be filled out for the most part but you will need to add a few things: the location of the genomic contigs to be annotated, the location of the EST and protein libraries, the location of the HMM parameter file for SNAP gene predictions, and the type of BLAST software you are using (ncbi+). The MAKER2 tutorial has a good bit of information about these files and should be consulted for more details. Examples of these files can be also found in Appendix S1. After everything is specified in the control files, MAKER2 can be run by typing ‘maker’ at the command prompt:

[paul@wolfelab ~/maker\_runs/run1]$ maker

Run times for MAKER2 will depend on how much sequence data you have to annotate, as well as how large your EST and protein libraries are. The longest runs for us were during the training of the ab initio gene predictor SNAP, which took around 20 hours to run. We were able to cut down on computation time by repeat masking our sequences prior to annotating them with MAKER2 and then turning off the additional repeat masking done by the program. This resulted in our largest genome (*P*. *centranthifolius* – 4.7 Mbp) taking ~4–5 hours.

**3) Processing MAKER2 output**

The output from MAKER2 is a large data store with many levels of folders and subfolders, each one holding a GFF3 (\*.gff) file with the annotations for a given contig. There will also be an index file that tells where each contig is located in the data store. In general, you won’t have to try to navigate through all of the folders and subfolders in the data store. MAKER2 comes with a number of utility scripts that can use the index file to process the output. The first script we used combines all of the GFF3 files for all of the contigs into one file. This is done with the ‘gff3\_merge’ utility script. To run the script, specify the data store index using the ‘-d’ flag followed by the name of the data store, then specify the output file using the ‘-o’ flag followed by the name you want for the output file.

[paul@wolfelab ~/maker\_runs/run1]$ gff3\_merge –d datastore.index –o all\_annots.gff

The resulting GFF3 file now has all annotations for every contig and can be used for any downstream applications. One application that we used was to take annotations and gene predictions produced by MAKER2 to train the ab initio gene predictor SNAP. The best instructions for how to complete the training of gene predictors is to read the MAKER2 tutorial as well as the documentation accompanying SNAP. The first step is to convert the GFF3 file to a format that is compatible with SNAP, which is called ZFF format. This can be done using the script ‘maker2zff.pl’. To run the script, simply call it from the command prompt and specify the combined GFF3 file with all of the annotations for your contigs.

[paul@wolfelab ~/maker\_runs/run1]$ maker2zff.pl all\_annots.gff

This creates the files ‘genome.ann’ and ‘genome.dna’. The commands for completing the creation of the HMM parameter file are as follows:

[paul@wolfelab ~/maker\_runs/run1]$ fathom –categorize 1000 genome.ann genome.dna

[paul@wolfelab ~/maker\_runs/run1]$ fathom –export 1000 uni.ann uni.dna

[paul@wolfelab ~/maker\_runs/run1]$ forge export.ann export.dna

[paul@wolfelab ~/maker\_runs/run1]$ hmm-assembler.pl org . > org.hmm

The final command will create the HMM file in the current directory (hence the ‘.’). The name of the organism of interest that you are studying should replace the word ‘org’. The files ‘uni.ann’, ‘uni.dna’, ‘export.ann’, and ‘export.dna’ are generated as you run the commands so you don’t need to worry about making them yourself. You can also extract statistics about the gene predictions and annotations your contigs received using the following command:

[paul@wolfelab ~/maker\_runs/run1]$ fathom genome.ann genome.dna –gene-stats

The HMM file can then be used to generate new gene predictions by running MAKER2 and using the output to make an HMM file. This iterative process is one way to create a reliable and useful gene prediction model for any organism that you study.

With all of your annotations complete, you can now curate them as you see fit. Annotations can be visualized using the Apollo Genome Browser (Lewis et al., 2002), and there are many other programs and utility scripts that can be coupled with MAKER2 to generate databases or to add additional information to your annotations (see GMOD website). We added functional annotations using BLASTX and the Perl script in Appendix S3 (*Hit-desc.pl*), and also used the annotations for phylogenetic marker development.

**LITERATURE CITED**

Campbell, M. S., M. Yaw, C. Holt, J. C. Stein, G. D. Moghe, D. E. Hufnagel, J. Lei, et al. 2013. MAKER-P: A tool kit for the rapid creation, management, and quality control of plant genome annotations. *Plant Physiology* 164: 513–524.

Cantarel, B. L., I. Korf, S. M. Robb, G. Parra, E. Ross, B. Moore, C. Holt, et al. 2008. MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Research* 18: 188–196.

Holt, C., and M. Yandell. 2011. MAKER2: An annotation pipeline and genome-database management tool for second generation genome projects. *BMC Bioinformatics* 12: 491.

Lewis, S. E., S. M. J. Searle, N. Harris, M. Gibson, V. Iyer, J. Richter, C. Wiel, et al. 2002. Apollo: A sequence annotation editor. *Genome* *Biology* 3: research0082- research0082.14.