Online Appendix

Image processing

Counting the number of spots on leaves manually is a very time consuming and error-prone task. Moreover we did not find a reasonable manual approach to determine a leaf's area occupied by holes. Thus, we have automated both processes by implementing an image analysis tool in C++, using the *ImageMagick Magick++ API* 1.

Photographs of each *Malva sylvestris* leaf were taken on a light board beside a black 100-mm^2 marker. Automated image processing was then used to compute leaf area, the proportion occupied by *Puccinia malvacearum* pustules, as well as the area of herbivore damage for each leaf. We applied a connected-components analysis to determine leaf (green component), pustules (black components within the leaf) and holes (white components within the leaf) in the images. In a connected-components analysis, continuous areas of similar colour are annotated (Samet & Tamminen 1986). For some leaves with severe insect damage the original leaf shape was estimated according to symmetric properties of the opposing leaf margin.

In the first step the picture's green channel is extracted. This results in a grey-scale image representing the levels of green in the original picture. As the leaf is (mostly) green, this allows a good separation of holes and spots from the leaf itself while unimportant details like leaf veins are made blue. Furthermore, to avoid artefacts we de-speckled the image.

Next the image's background is detected. We use the top-leftmost and lower-rightmost pixels in the image to represent the background colour and apply a fuzzy flood-fill algorithm from both pixels. Using predefined thresholds for black and white, we reduce the image information to three states: (*a*) background, (*b*) spot (all pixels below the black threshold), and (*c*) hole (all pixels above the white threshold) with (*d*) leaf (all remaining pixels). An edge detection algorithm determines the two largest objects (composed from pixels of categories *b*, *c* and/or *d*) surrounded by background. The largest one is supposed to be the leaf. The second one is supposed to be the marker, which is used as size control.

To determine the number of spots, we look for connected components among the pixels of category *b* on the leaf. Thus, all neighbouring black pixels are considered to belong to the same hole. Spots with an area smaller than 0.02% of the image area are omitted and converted to category *d*, as they most likely represent artefacts. Apart from spots, shadows and overlapping leaf parts may also appear black in the image and might be counted as spots. To avoid this we have implemented an adaptive size limiter. These false holes are usually considerably bigger than the true holes and their number is largely limited. For up to the 10 largest holes, we checked if the following condition is fulfilled:

$$\frac{s_{n-1}}{s_n} < 0.55,$$

where S_n represents the spot size in an ascending ordered list. If the condition is not fulfilled, we discard every spot bigger than S_{n-1} and convert them to category *d*. In this way false holes are automatically detected in an adaptive way.

Finally, we calculate the number of pixels in categories *b*, *c* and *d* to derive our measurements. All three categories are thereby ascribed to the leaf area. Category *b* pixels are also counted as spots and category *c* pixels for holes. Thus, the fraction of holes would be the number of category *b* pixels divided by the number of category *b*, *c* and *d* pixels and so on.

The presented algorithm is not capable of estimating the original shape of a leaf if holes touch the background. The hole would be regarded be background. To circumvent this, a thin green line was added to those pictures manually. It follows the assumed shape of the leaf where it is interrupted by holes. In this way the program will correctly identify the area as a hole.

Furthermore we manually adjusted the black threshold for some images with odd exposure conditions and adjusted the spot-size-limit parameter for leaves with very unequal spot sizes. All image annotations were manually checked for plausibility. The implemented tool is distributed under the open-source licence *GPLv3* and can be downloaded at http://www.toxido.de/bioinf/SpotCounter/.

Reference

Samet H, Tamminen MA 1986. A general approach to connected component labeling of images. Computer Science Technical Report TR-1649. College Park, MD, University of Maryland.