Supporting information

For “Efficient, automated, and robust pollen analysis using deep learning” by Ola Olsson, Melanie Karlsson, Anna S. Persson, Henrik G. Smith, Vidula Varadarajan, Johanna Yourstone & Martin Stjernman

This document contains additional information regarding

- the species and pollen types used
- the image extraction algorithm
- throughput times
- separation of pollen from non-pollen objects
- confusion matrix of the leave-one-out experiment at the pollen type level
- effects of staining intensity
- optimization of the value of exponent *x* in the adjustment of scores based on sample frequencies.

Species and pollen type list

Below is a table with all species used in CNN trainings and classifications, their grouping into pollen types, the number of samples per species, and the number of pollen grain images in the reference library. Each sample is from a different flower, and were in most cases taken at different localities and days. The aim was to get at least 1000 pollen grains per species from at least two different samples. In some cases, however, fewer grains could be extracted from the images. Also, in some cases we only had a single sample per species, and we then pooled species in a genus (e.g. *Potentilla sp.*) and treated them as one.

Pollen types that consist of a single species are named after that species. If they consist of several species from a single genus or family, they are named after that genus or family. Some types are collections of species from several genera or families, and are named after a typical species or genus in the group.

The species marked with * were only included when investigating the effect of having "empty classes" in the trained model, i.e. species not actually present in the bumblebee samples analysed.

Table S1.

	Species	Pollen type	Number of samples	Number of sorted pollen grains
	Acer campestre	*Acer*	4	1173
	Acer platanoides	*Acer*	5	1828
	Acer pseudoplatanus	*Acer*	5	1395
	Aesculus hippocastanum	*Aesculus hippocastanum*	6	3515
*	*Achillea millefolium*	Asteraceae	2	1000
*	*Arctium tomentosum*	Asteraceae	2	869
*	*Carduus crispus*	Asteraceae	2	588
*	*Cichorium intybus*	Asteraceae	2	1000
*	*Cirsium arvense*	Asteraceae	6	1066
*	*Cirsium vulgare*	Asteraceae	3	1573
*	*Leucanthemum vulgare*	Asteraceae	3	1000
*	*Tripleurospermum inodorum*	Asteraceae	2	1006
	Alliaria petiolata	Brassicaceae	5	1562
	Barbarea vulgaris	Brassicaceae	5	1864
	Brassica napus	Brassicaceae	7	1896
	Capsella bursa-pastoris	Brassicaceae	3	1119
	Cardamine pratensis	Brassicaceae	5	1646
	Hesperis matronalis	Brassicaceae	4	1195
	Thlaspi arvense	Brassicaceae	2	1009
	Centaurea cyanus	*Centaurea cyanus*	5	2772
	Cytisus scoparius	*Cytisus scoparius*	7	2240
*	*Echium vulgare*	*Echium vulgare*	3	1401
*	*Fraxinus excelsior*	*Fraxinus excelsior*	3	1314
	Glechoma hederacea	*Glechoma hederacea*	6	2523
*	*Heracleum sp.*	*Heracleum*	2	1000
*	*Jasione montana*	*Jasione montana*	4	1420
	Laburnum sp.	*Laburnum*	6	2266
	Lamiastrum galeobdolon	*Lamium*-group	6	1805

Lamium album	*Lamium*-group	5	1605
Lamium hybridum	*Lamium*-group	5	724
Lamium purpureum	*Lamium*-group	8	1315
Knautia arvensis	*Lonicera*-group	7	622
Kolkwitzia amabilis	*Lonicera*-group	4	1203
Lonicera caprifolium	*Lonicera*-group	4	684
Lonicera periclymenum	*Lonicera*-group	4	1331
Lonicera tatarica	*Lonicera*-group	4	989
Lonicera xylosteum	*Lonicera*-group	6	1922
Valeriana dioica	*Lonicera*-group	2	1027
Lupinus polyphyllus	*Lupinus polyphyllus*	8	3135
Chelidonium majus	*Papaver*-group	4	1000
Papaver dubium	*Papaver*-group	2	1000
Papaver rhoeas	*Papaver*-group	5	1714
Phacelia tanacetifolia	*Phacelia tanacetifolia*	3	3670
Pinaceae	Pinaceae	4	1006
Fragaria vesca	*Potentilla*-group	4	1825
Geum rivale	*Potentilla*-group	4	1559
Geum urbanum	*Potentilla*-group	3	1081
Potentilla sp.	*Potentilla*-group	8	4429
Amelanchier sp.	*Prunus*-group	5	1369
Cotoneaster sp.	*Prunus*-group	6	1127
Crataegus laevigata	*Prunus*-group	3	1612
Crataegus monogyna	*Prunus*-group	6	1373
Malus domestica	*Prunus*-group	5	2000
Prunus avium	*Prunus*-group	5	2066
Prunus domestica	*Prunus*-group	3	876
Prunus laurocerasus	*Prunus*-group	3	1194
Prunus padus	*Prunus*-group	4	1522
Prunus spinosa	*Prunus*-group	5	1847
Pyrus communis	*Prunus*-group	4	1626
Sorbus aucuparia	*Prunus*-group	2	1510
Ribes nigrum	*Rosa*-group	3	1095
Ribes rubrum	*Rosa* -group	2	1044
Rosa helenae	*Rosa* -group	3	1312
Rosa multiflora	*Rosa* -group	2	1137

	Rosa rugosa	*Rosa* -group	2	1000
	Rubus fruticosus	*Rosa* -group	4	1956
	Rubus idaeus	*Rosa* -group	2	1000
	Anchusa officinalis	*Pulmonaria*-group	5	1567
	Quercus robur	*Quercus*	4	1187
	Quercus rubra	*Quercus*	2	1025
*	*Ficaria verna*	Ranunculaceae	4	1348
*	*Ranunculus acris*	Ranunculaceae	5	1241
*	*Ranunculus bulbosus*	Ranunculaceae	2	1052
*	*Ranunculus repens*	Ranunculaceae	2	1015
	Rhododendron sp.	*Rhododendron*	5	1415
	Robinia pseudoacacia	*Robinia pseudoacacia*	2	1027
	Salix alba	*Salix*	2	513
	Salix caprea	*Salix*	5	1783
	Salix cinerea	*Salix*	5	1282
	Salix euxina	*Salix*	4	1248
	Salix repens	*Salix*	2	1298
	Salix viminalis	*Salix*	3	1117
	Salix x fragilis	*Salix*	2	1026
	Sambucus nigra	*Sambucus nigra*	5	1371
	Symphytum officinale	*Symphytum*	3	1364
	Symphytum x uplandicum	*Symphytum*	3	1106
	Bellis perennis	*Taraxacum*-group	2	1000
	Jacobaea vulgaris	*Taraxacum*-group	3	1000
	Petasites hybridus	*Taraxacum*-group	2	1029
	Pilosella officinarum	*Taraxacum*-group	3	1192
	Senecio leucanthemifolius	*Taraxacum*-group	2	1035
	Senecio vulgaris	*Taraxacum*-group	2	778
	Taraxacum sp.	*Taraxacum*-group	5	1626
	Tussilago farfara	*Taraxacum*-group	4	1309
	Tilia cordata	*Tilia cordata*	4	1132
	Trifolium pratense	*Trifolium pratense*	10	2395
	Trifolium repens	*Trifolium repens*	4	1384
	Viola arvensis	*Viola arvensis*-group	5	716
	Viola tricolor	*Viola arvensis*-group	5	1047

Image extraction

The CNN algorithms need small images of fixed size (e.g. 224 × 224 pixels), with a single object in each. To extract such images, with (ideally) a single pollen grain in each, we developed an algorithm in MATLAB, based on edge detection and morphology, and followed by watershed analysis (Fig. S1). The algorithm is based on a sequence of operations using functions from MATLAB's Image Processing Toolbox. First, the image is converted to grey scale (Fig. S1 A), and then converted to a gradient mask using the edge function (method "Sobel"; Fig. S1 B). The mask is dilated using linear structuring elements (MATLAB functions strel and imdilate; Fig. S1 C), and then filled (function imfill; Fig. S1 D), and eroded (function imerode; Fig. S1 E), to create a binary mask covering all objects in the image. Our algorithm then connects pixels in the mask with value 1 into "objects" of sizes between 500 and 7.105 pixels (using function bwareafilt; Fig. S1 F), which typically includes all individual pollen grains, many aggregations of grains, and some debris in the images. To separate individual pollen grains occurring in aggregations, the algorithm first calculates the distance from any pixel inside an object to the edge (function bwdist, Fig. S1 G), and then identifies individual pollen grains through a watershed analysis of the resulting distance matrix (function watershed; Fig. S1 H and I). This process correctly extracted more than 95% of all pollen grains in most images, as well as some non-pollen objects. Finally, the centroids and major axis lengths of all objects are calculated using the function regionprops. Bounding boxes, based on centroids and major axis length are shown in Fig. S1 I, and these are used to extract and save individual images of each object after rescaling to the appropriate size (in our case 224 × 224 or 299 × 299 pixels depending on CNN model). Thus, the process locates and extracts all objects in the sample images, and saves individual images of each object, which can then easily be handled and sorted by standard file management software. The algorithm is fast and it takes ca 1 min to process one sample image (ca 24 000 × 24 000 pixels), and extract and save all objects in it (typically up to 10 000) using current hardware (Intel i9 with 64 GB RAM; Nvidia Geforce RTX 2060 Super with 8 GB GRAM).

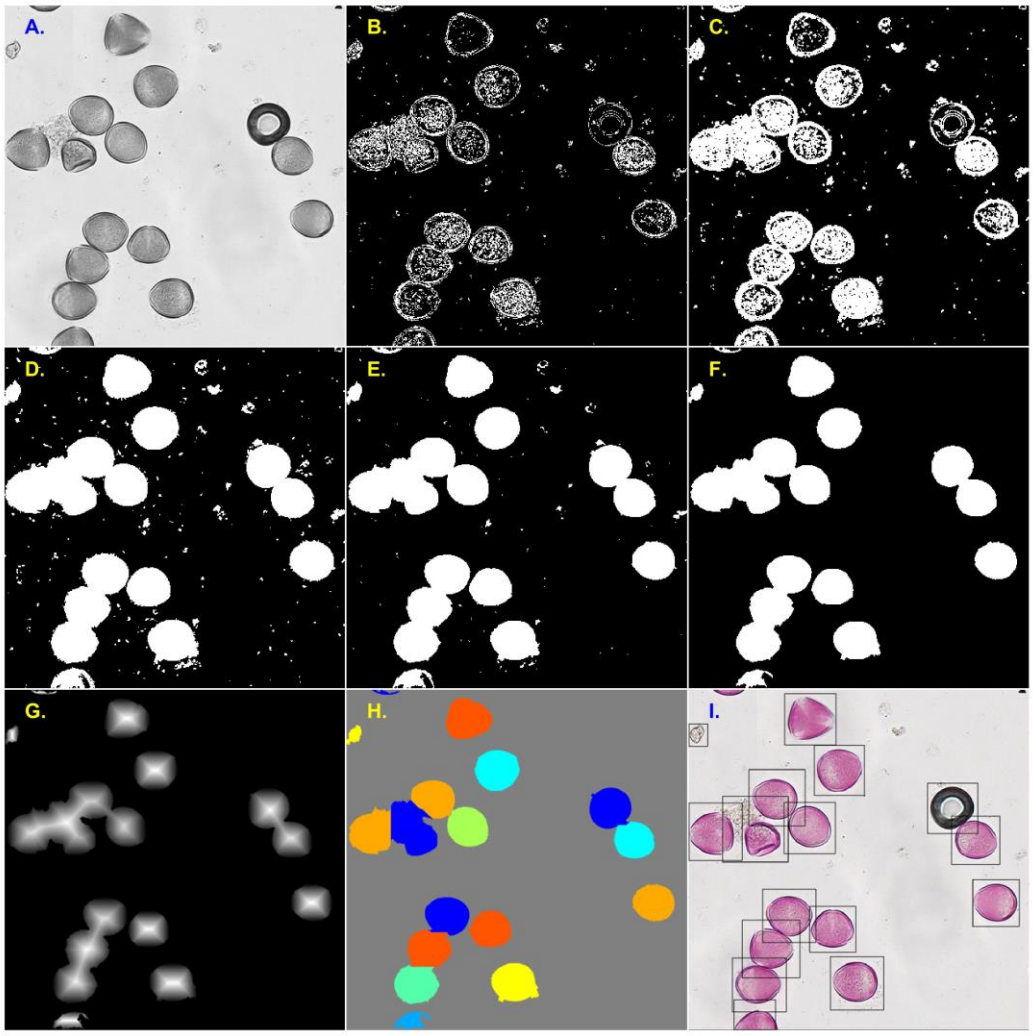

Figure S1.

Throughput times

Times spent per the different steps involved in the pollen analysis process. All times are approximate, but meant to represent net-times including documentation and error handling (e.g. if scanning of a slide needs to be repeated because it failed). Image stacking takes the most machine time, but is fully automated, and can run in batches overnight.

Table S2.

	Manual time	**Machine time**	**per sample unit**	**per pollen units**
Sample preparation				
Preparation of gel	ca 1 h		1000 samples	
Field sampling, reference	<5 min		1 sample	100+ pollen grains
Slide preparation, reference	4-5 min		1 sample	100+ pollen grains
Field sampling, bumblebee	1-5 min		1 sample	≤10 000 pollen grains
Slide preparation, bumblebee	6-7 min		1 sample	≤10 000 pollen grains
Scanning of slides	5-10 min	25-30 min	5 slides	
Image stacking	<<1 min	15 min	1 image	
CNN training				
Reference pollen sorting	10-25 min		1 sample	100-400 pollen grains
Model training, ResNet-18	ca 5 min	30-45 min	1 model	29-83 pollen types
Model training, GoogLeNet	ca 5 min	40 min	1 model	29 pollen types
Model training, Xception	ca 5 min	3:45 hours	1 model	29 pollen types
Image extraction and data analysis				
Object finding	<<1 min	ca 1 min	1 image	≤10 000 pollen grains
Model classification	<<1 min	<1 min	1 sample	≤10 000 pollen grains

Separating pollen from non-pollen objects

We ran a CNN-model with the 28 pollen types as defined in the main paper, plus damaged pollen, and four categories of non-pollen objects (Fig. S2). The model was trained on 2700 pollen per type, and a splitting experiment was run using 300 pollen per type. The pollen types were treated individually, but to simplify the illustration shown as a single class ("Pollen") in the confusion matrix (Fig. S3). The CNN-model could successfully separate most real pollen from the non-pollen types.

Fig. S2. Examples of damaged pollen and four types of non-pollen objects.

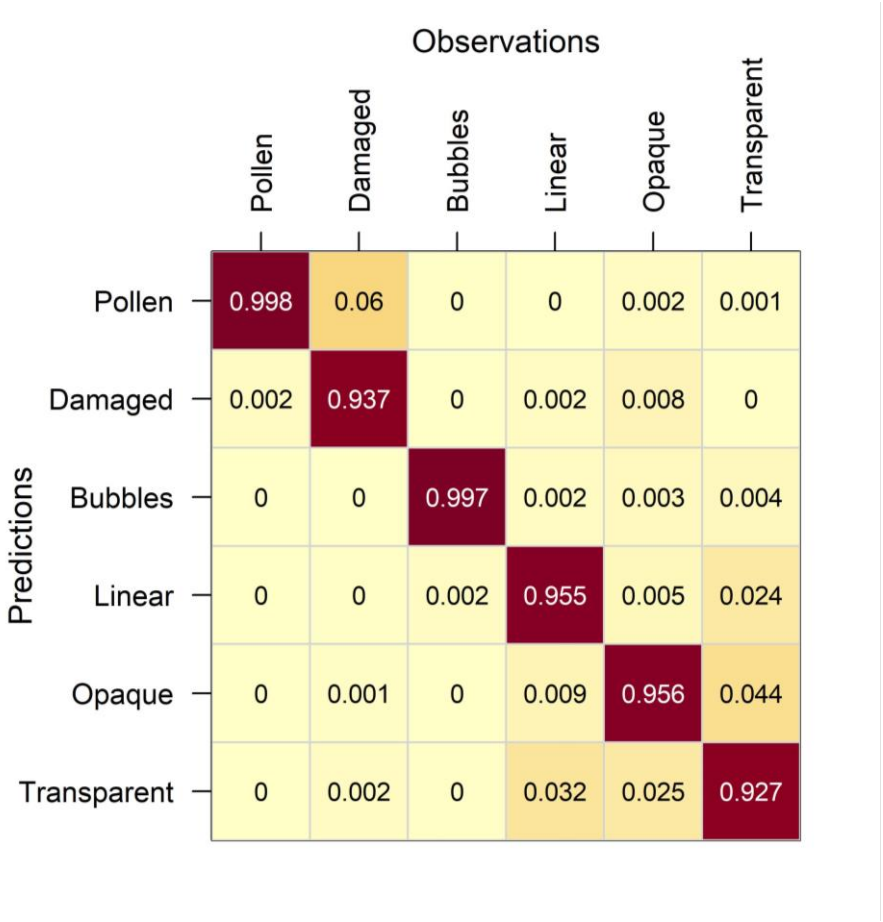

Fig. S3. Confusion matrix of pollen (combination of 28 classes), damaged pollen, and four types of non-pollen objects. Values shown are the summed classification scores, within each class. A value displayed as 0 is <0.0005.

Confusion matrix of pollen types

Confusion matrix (Fig. S2) resulting from the leave-one-out experiment at the pollen type level, i.e. the 28 pollen types in Table S1. Columns show true pollen types and rows show predicted pollen types. Colours indicate the frequency in each cell according to the inset legend.

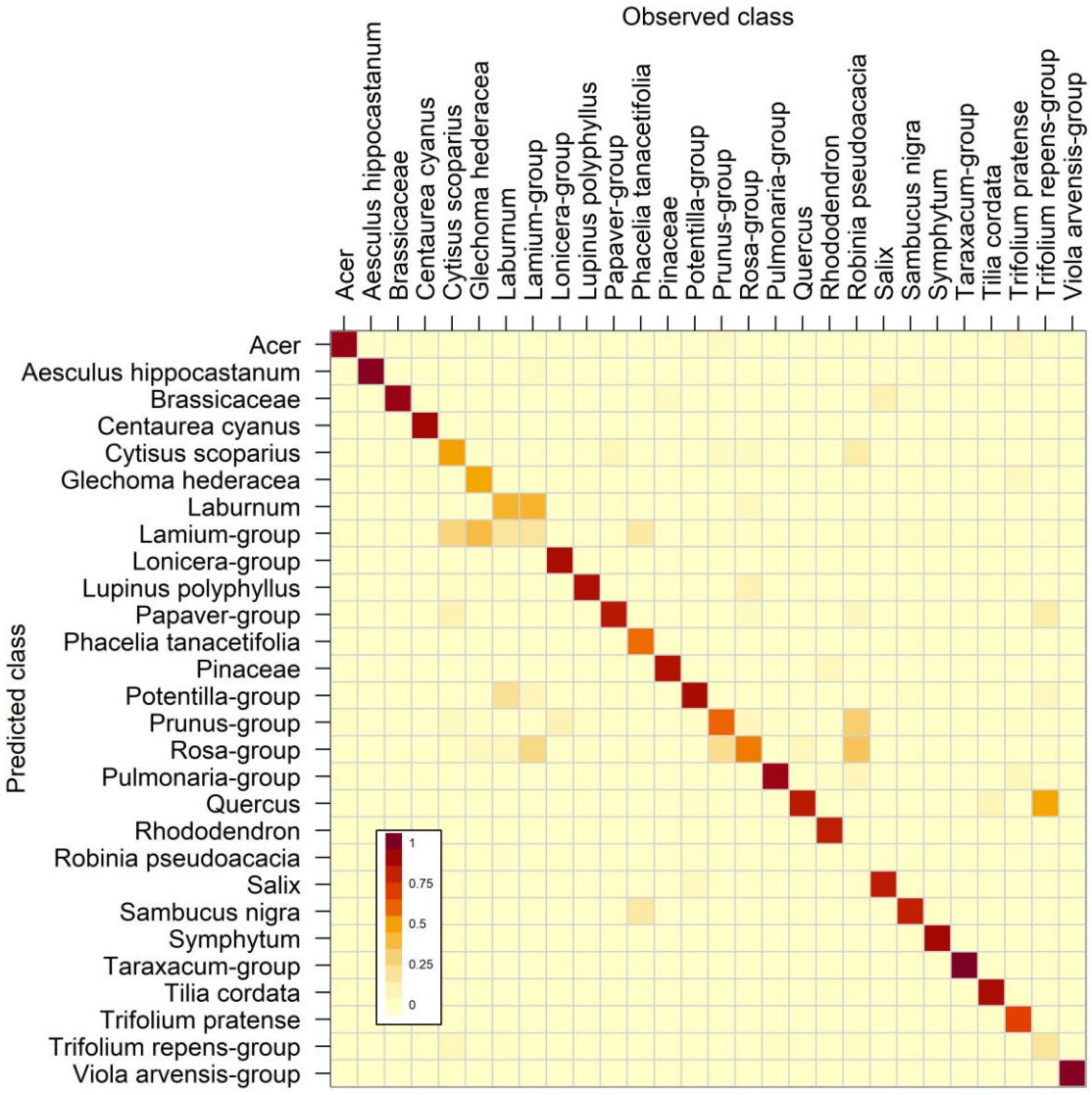

Figure S2.

Effect of staining intensity

The colouring of the fuchsine gel varies strongly with the amount of fuchsine used when preparing it, and different gels may have somewhat different fuchsine concentration and thereby colour, staining the pollen differently. Potentially, variation in staining could affect the possibilities for the CNN to correctly identify the pollen species. To assess to what extent variation in staining affected recall rates of species we measured the grayscale intensity of all pollen images used in the leave-one-out experiment at the pollen type level (Fig. S3). We thus converted the images to grayscale and measured the intensity of pixels in a circle with diameter 75% of the images' width (yellow dashed circles in Fig. S3). For each sample we calculated the mean of the individual pollen grains. Among the 140 samples used in the experiment, the lowest mean intensity was 65.6 and the highest was 192 (mean=113, standard deviation=27.7).

We did not find that image intensity (brightness) affected recall rates of samples within species. In a simple linear model with recall rate as dependent and pollen type and mean intensity per sample as independent variables, pollen type was highly significant ($F_{27,105}$=5.17, P<0.0005), whereas intensity was not ($F_{1,105}$=0.071, P=0.8).

Thus, there is no evidence that staining affects recall rates in our samples, but all our images vary moderately in staining, approximately as represented by the examples in Fig. S3. If staining is much darker features of the pollen grains might disappear, making identification difficult. Similarly, if the staining is too weak, important features might not be apparent.

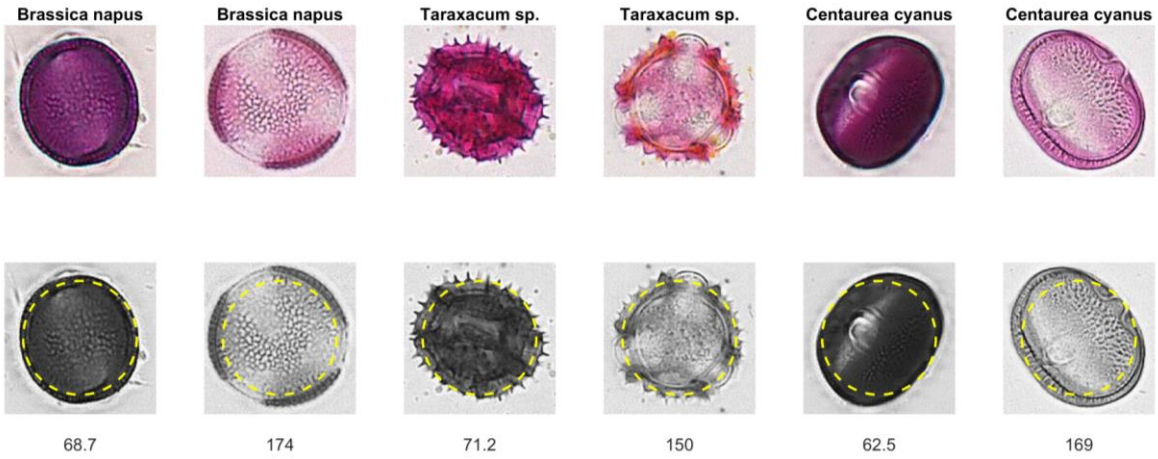

Figure S3. One dark and one bright image of three species. The upper row in full colour and the lower in grayscale. The values below the images are their intensity values within the yellow circles, on a scale from 0 to 255.

Optimizing the value of exponent *x* in the adjustment of scores based on sample frequencies

As described in the main article, the scores from the CNN model for each pollen grain, *i*, in a sample, *k*, can be adjusted by the estimated species frequencies of that sample to reduce the risk of overestimating the number of species in samples where one can expect a low number of species (such as in samples from bumblebees). The extent of this adjustment is controlled by the exponent x in the equation $\mathbf{p}'_i = \mathbf{p}_i \mathbf{f}_k^x$ (i.e the number of times that $\mathbf{f}_k$ is multiplied into the original scores; see the main article). The optimal value of *x* is can be determined through cross-validation against a subset of samples where pollen grains are manually classified under the assumption that this classification is correct. The optimal value of *x* is then the value that minimize deviance of adjusted scores as calculated against the manual classification (see main article for description of deviance).

As an illustration, we performed an optimization of *x* for our bumblebee samples. We tried values of *x* from 0-2.5 to find what value minimized deviance (Fig. S4). We found the optimal choice of *x* to be 1.4. However, as the deviance curve is flat near the minimum, *x*=1 would result in an adjustment almost as good as the optimal and, hence, setting *x* to 1 could be an option in cases where cross validation against manually identified samples is not possible.

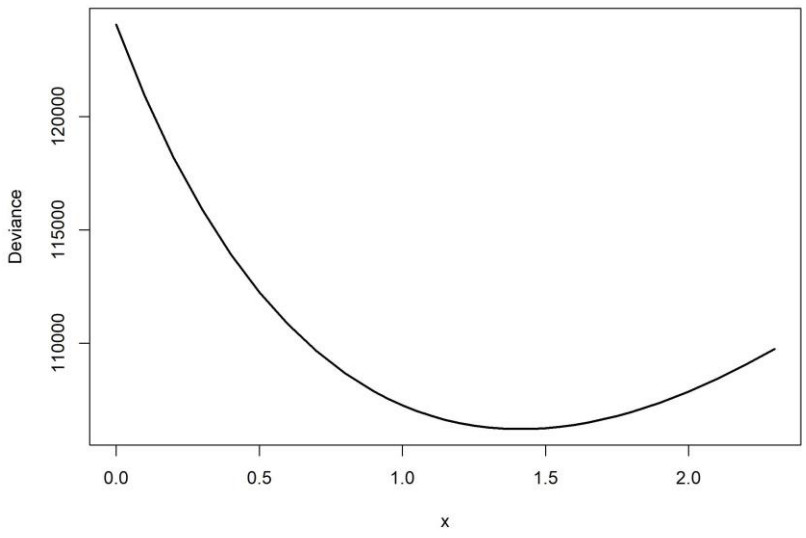

Figure S4.