Title

Few-Shot Learning Enables Population-Scale Analysis of Leaf Traits in Populus trichocarpa.

Permalink

https://escholarship.org/uc/item/97s7q3vh

Authors

Lagergren, John
Pavicic, Mirko
Chhetri, Hari
et al.

Peer reviewed

RESEARCH ARTICLE

Few-Shot Learning Enables Population-Scale Analysis of Leaf Traits in *Populus trichocarpa*

John Lagergren[1*], Mirko Pavicic[1], Hari B. Chhetri[1], Larry M. York[1], Doug Hyatt[1], David Kainer[1], Erica M. Rutter[2], Kevin Flores[3], Jack Bailey-Bale[4], Marie Klein[4], Gail Taylor[4], Daniel Jacobson[1*], and Jared Streich[1*]

[1]Biosciences Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA. [2]Department of Applied Mathematics, University of California, Merced, CA, USA. [3]Department of Mathematics, North Carolina State University, Raleigh, NC, USA. [4]Department of Plant Sciences, University of California, Davis, CA, USA.

*Address correspondence to: lagergrenjh@ornl.gov (J.L.); jacobsonda@ornl.gov (D.J.); streich.jared@ gmail.com (J.S.)

Plant phenotyping is typically a time-consuming and expensive endeavor, requiring large groups of researchers to meticulously measure biologically relevant plant traits, and is the main bottleneck in understanding plant adaptation and the genetic architecture underlying complex traits at population scale. In this work, we address these challenges by leveraging few-shot learning with convolutional neural networks to segment the leaf body and visible venation of 2,906 *Populus trichocarpa* leaf images obtained in the field. In contrast to previous methods, our approach (a) does not require experimental or image preprocessing, (b) uses the raw RGB images at full resolution, and (c) requires very few samples for training (e.g., just 8 images for vein segmentation). Traits relating to leaf morphology and vein topology are extracted from the resulting segmentations using traditional open-source image-processing tools, validated using real-world physical measurements, and used to conduct a genome-wide association study to identify genes controlling the traits. In this way, the current work is designed to provide the plant phenotyping community with (a) methods for fast and accurate image-based feature extraction that require minimal training data and (b) a new population-scale dataset, including 68 different leaf phenotypes, for domain scientists and machine learning researchers. All of the few-shot learning code, data, and results are made publicly available.

Introduction

Image-based plant phenotyping is a method by which scientists use image data to characterize and categorize plants within and across species. This process typically involves the use of tools, instrumentation, and domain expertise to (a) measure information from individual or groups of samples in the greenhouse, field, and/or nature, (b) be applied across scales, ranging from cell microscopy to satellite imagery, and (c) allow researchers to extract complex morphological and topological features that would otherwise be impossible to measure by hand. One of the main challenges of image-based phenotyping is identification of the relevant biological structures (foreground) from the background. In some cases, imaging methods can be modified to highlight these objects, such as using back lights or relying on fluorescence of those objects; however, in many situations, the contrast between the relevant object and the background is low. When contrast is high, simple grayscale or color-based thresholding can be used, but in more complex color imagery, plant phenomics has focused on machine learning approaches.

Deep learning has revolutionized computer vision as a powerful and efficient way to extract features from image-based data [1–3]. An important strength of this approach is that deep learning models can learn an invariance to heterogeneous background effects that allows them to generalize to new samples outside of the training set. However, such approaches can be laborious and expensive to adopt, because users must generally annotate hundreds or thousands of images to provide sufficient training data. In plant biology for example, to reliably associate plant traits with genes at population scale requires large amounts of observations that span hundreds or thousands of genotypes. Such associations provide deeper understanding of the genetic architectures and underlying mechanisms that govern complex processes that control the growth, acclimation, response, and composition of plants, with important implications for sustainable agriculture and bioenergy [4,5]. Thus, there exists a need to develop methods for fast and accurate image-based plant phenotyping that alleviate the data annotation bottleneck.

In contrast to traditional deep learning approaches, which can require large amounts of training samples to reach sufficient prediction accuracy [1,2], few-shot learning is an emerging subset of machine/deep learning that attempts to maximize predictive accuracy while using only a small number of labeled samples for training. Multiple approaches exist to solve this problem, including data augmentation, metric learning, external memory, and parameter optimization [6]. This work utilizes a combination of data augmentation (i.e., applying random spatial and color augmentations to images during training) and iterative algorithms, which have been previously demonstrated for biomedical image analysis, e.g., semantic segmentation of cells and retinal vasculature [7–10]. The goal of this study is to extend these methods to image-based plant phenotyping by leveraging convolutional neural networks (CNNs) to segment the body and visible vein architecture of poplar (*Populus trichocarpa*) leaves from high-resolution scans obtained in the field. In particular, few-shot learning is utilized in this work because it divides a small number of large images into a large number of small image tiles. In this way, the complex task of whole-image segmentation is broken down into smaller easier decision rules, which enables accurate segmentation using very few labeled images for training. Note that use of the term "few-shot learning" to describe the methods in this work is distinct from the typical definition and here refers to maximizing predictive outcomes using very few labeled images following terminology from [6].

P. trichocarpa (also called black cottonwood, western balsam-poplar, or California poplar) is a model system for studying the genetic architecture of complex traits and climate adaptation in woody plants. Spanning from central California, USA, to northern British Columbia, Canada, it harbors tremendous geographic, climatic, phenotypic, and genetic diversity. Further, *P. trichocarpa* has a fully sequenced genome, genome annotation, abundant transcriptomes, resequencing, and phenotypic data. Importantly, rapid biomass growth, clonal propagation, and the ability to grow in marginal lands with low agricultural input make it an ideal crop for sustainable bioenergy applications [11–18]. As a result, research and commercial groups have invested heavily in the development of *P. trichocarpa* as a high-impact species for forest products and biofuel production [16,19–21]. To this end, leaves play a key role in biomass production since they are the primary organs responsible for sunlight absorption and carbon fixation, the food source of vascular plant systems. Further, vein architecture supports the mechanical structure of the leaf and governs the distribution of water and other nutrients, which has important implications for the physiology, biomechanics, and structure of a plant [22]. Thus, capturing accurate leaf traits and relating them to the genetic components that control them may provide insights toward improved tree biomass and composition.

In plant phenotyping, segmentation of individual leaves and their venation has seen sparse attention. In general, existing approaches use (a) experimental methods to chemically clear the leaf lamina and stain the veins to highlight the venation against the background [23,24]; (b) image preprocessing by grayscaling, aggregating specific color channels, or spatial rescaling [23–27]; (c) global filters and morphological operations (e.g., Odd Gabor filters, Hessian matrices, vesselness filters, and region merging) to obtain binary segmentations [23,25–28]; (d) ensembles of scales and models to make aggregate predictions [28,24]; and (e) require hundreds of manually annotated

training samples to produce accurate segmentation models [24]. However, these commonly encountered steps can bottleneck the scalability and accuracy of image-based plant phenotyping at population scale. For example, approach (a) adds additional experimental time, effort, materials, expenses, and hazards to data acquisition compared to capturing just raw images; (b) destroys fine-grained image details across spatial and color dimensions; (c) may be overly simplistic and generate large amounts of effort in segmentation postprocessing; (d) uses complex workflows which may be difficult to automate at scale; and (e) can be infeasible for smaller research groups with limited time and budgets. These challenges may help explain why leaf and vein segmentation has not received as much attention compared to crop- or field-level phenotyping for plant stress, shoot morphology, and plant/organ counting [29].

This work presents few-shot learning methods based on CNNs to segment the body and visible vein architecture of *P. trichocarpa* leaves. Leaf segmentation is formulated as a tracing task, in which a CNN iteratively traces the boundary of a leaf to produce a single contiguous leaf segmentation. Previous studies have shown that alternative CNN-based segmentation methods (e.g., U-Net [3]) that do not include spatial priors that encourage object contiguity may result in biologically unrealistic segmentation masks (i.e., contain holes or other artifacts caused by individual or groups of pixels that do not exceed a global probability threshold) [7]. In contrast, boundary tracing addresses this challenge by only segmenting a single contiguous region, thereby ensuring accurate downstream extraction of morphological features. Alternatively, vein segmentation is formulated as a region growing task, in which a CNN iteratively adds neighboring pixels to a growing region of interest corresponding to the visible vein architecture. Similar to the tracing approach, the vein segmentation ensures biologically realistic morphological features by including pixels in the segmentation only if a neighboring pixel was previously classified. Each method is fully automated (i.e., requires no human supervision or initialization) and segments images orders of magnitude faster compared to manual annotation. Additionally, each method is compared against a variant of the state-of-the-art image segmentation model, U-Net [3].

The current work is designed to provide the plant phenotyping community with (a) methods for fast and accurate image-based feature extraction with minimal training data and (b) a new population-scale dataset for domain scientists and machine learning researchers. In particular, the methods developed here are applied to raw RGB images with no experimental/image preprocessing, use individual CNN models that learn the complex relationships between pixels for accurate leaf and vein segmentation, and require very few training samples to generalize and make accurate predictions at population scale. The segmentations are used to extract biologically realistic features that are validated using real-world physical measurements and applied downstream using broad-sense clonal heritability estimates and a genome-wide association study (GWAS).

Materials and Methods

CNNs are used to segment the body and visible vein architecture of *P. trichocarpa* leaves from high-resolution scans. The resulting segmentations are combined with open-source tools for image processing and genomic analysis to expand the application of these methods to a wider scientific audience. All deep

learning methods are implemented in Python (version 3.7.8) using the PyTorch deep learning library (version 1.11.0) [30] and are made publicly available at [31]. Feature extraction is completed using Fiji (version 2.9.0) [32] and RhizoVision Explorer (RVE, version 2.0.3) [33,34]. Genomic analysis is conducted in R (version 4.2.0) using the GAPIT3 software package (version 3) [35]. All of the images, manual segmentations, model predictions, extracted features, and underlying genome sequences are made publicly available at [36]. Finally, a copy of both the code and the data is available at [37].

Data collection

The leaf scans considered in this work were collected during a field campaign in August, 2021, from the 10-acre poplar plantation at the University of California, Davis, which maintains a common garden of poplar trees that can be grown on low-quality, poor, and marginal land [4,38]. The plantation is composed of 3 blocks. Each block is partitioned into rows and positions that uniquely identify the corresponding genotypes and contain approximately 1,500 *P. trichocarpa* trees per block. For practical reasons, leaf samples were collected from the entire first block (1,322 viable samples) and partially from the second block (131 samples) totaling 1,453 trees.

Leaves were sampled from a branch at approximately breast height (i.e., ~1.37 m) from the south-facing side of each tree. Leaves were chosen by selecting the first fully mature leaf counting from the top of each branch. Each leaf was also paired with a barcode label that encoded the treatment, block, row, and position of the tree, which uniquely identified the corresponding genotype and allowed the user to record the sample ID during data capture. This helped expedite the phenotyping process and reduce human error. Selected leaves were scanned in the field as they were sampled from each tree using a USB-powered Epson Perfection model V39 [39]. The top and bottom of each leaf was scanned with a resolution of 300 dots per inch (DPI). To account for heterogeneous leaf shapes (e.g., leaves with nontrivial 3-dimensional [3D] characteristics like "waviness"), a weight was used on the scanner lid to compress each leaf to the glass of the scanner in order to reduce image artifacts like blurring. Additionally, between rows of trees (there are approximately 30 trees per row), the scanner glass and background were cleaned to reduce the buildup of dust and other debris.

During data capture, the scanner suffered a hardware failure where one of the pixels of the scanner began to malfunction and caused a vertical white line to gradually appear near the center of each subsequent scan. This artifact affected approximately 100 leaf scans. To mitigate the malfunction, leaves were moved to the edge of the scanner away from the malfunctioning pixel, affecting 62 leaf scans. A new scanner was acquired and used for the remainder of the field campaign (2,634 leaf scans). Despite the hardware failure, these data acquisition steps resulted in 2,906 RGB leaf scans (i.e., top and bottom of 1,453 samples), each with a dimension of 3,510 × 2,550 pixels.

In addition to image-based measurements, petiole length and diameter were measured manually for each leaf. Using a similar procedure to leaf imaging, barcode scanners were used to record the sample ID, followed by length/diameter measurements using USB-powered SPI 17-600-8 electronic calipers [40]. The manual measurements for petiole length and width are used to validate image-based measurements.

Obtaining accurate high-quality ground truth data is important for deep learning applications in general, but it is crucial for few-shot learning, since a model must learn features from a small number of training samples that generalize well to the broader population. To this end, training data was generated for leaf body segmentation using the top and bottom scans of 25 leaves (50 images in total), which were randomly selected and manually traced. Manual segmentation was completed using the open-source graphics editor, GNU Image Manipulation Program [41], taking between 15 and 30 min per image, depending on the size and serration of the leaf. Similarly for vein segmentation, GNU Image Manipulation Program was used to manually draw all visible leaf venation for 8 leaf-bottom scans, taking between 4 and 8 h per image, depending on the vein density. Note that only leaf-bottom venation is considered in this work. Leaf-top venation will be considered in future work.

Due to the large amount of manual effort required for vein segmentation, the training dataset was constructed using iterative dataset refinement, in which images were individually added to the training set based on manual inspection of model performance across the set of all images. For example, compressing samples against the scanner glass caused some leaves to fold on themselves, which produced dark lines that were falsely identified as veins. Thus, an image with multiple examples of such folds was manually segmented and added to the training set so that the model learned an invariance to such artifacts. This process was repeated similarly for other leaf characteristics (e.g., dead, diseased, and nutrient-deficient leaf tissue), including a scan exhibiting the hardware failure discussed above, until the model converged to acceptable performance across the population. This strategy resulted in a total of 8 images (6 for training, 2 for validation) mentioned previously. Note that in practice, the number of images may vary depending on the application and image quality, but it is important (particularly for few-shot learning) that the training dataset is fine-tuned to the point that the model is able to generalize.

Leaf segmentation

Segmentation of the leaf body is formulated as an object tracing task based on [7,8], in which a CNN is used to iteratively trace along the contour of a leaf. These methods have been shown to reach state-of-the-art accuracy in biomedical image segmentation using a fraction of the training data required by other approaches [8]. In this framework, a CNN inputs a small image tile centered somewhere on the edge of an object and outputs a predicted trace (i.e., set of pixel displacements) along the object boundary from the center to the edge of the tile. The iteration proceeds by generating new image tiles along the predicted contours, continuing the trace until reaching the starting location, thereby closing the loop and finishing the segmentation. An important benefit of this approach is that it breaks the complex task of whole-leaf segmentation into multiple smaller, easier decision rules and requires only a small number of images to train an accurate model. See Fig. 1 for a diagram of the leaf tracing algorithm and Movie S1 for a video of the iteration.

The leaf tracing CNN inputs 256 × 256 × 4 image tiles, which include 3 color channels (RGB) and an overlay of the previously traced path as an additional channel. The tile size is chosen large enough to provide the model with sufficient context to trace through areas where the leaf contour may be obscured (e.g., in damaged/diseased areas or near the petiole). The RGB
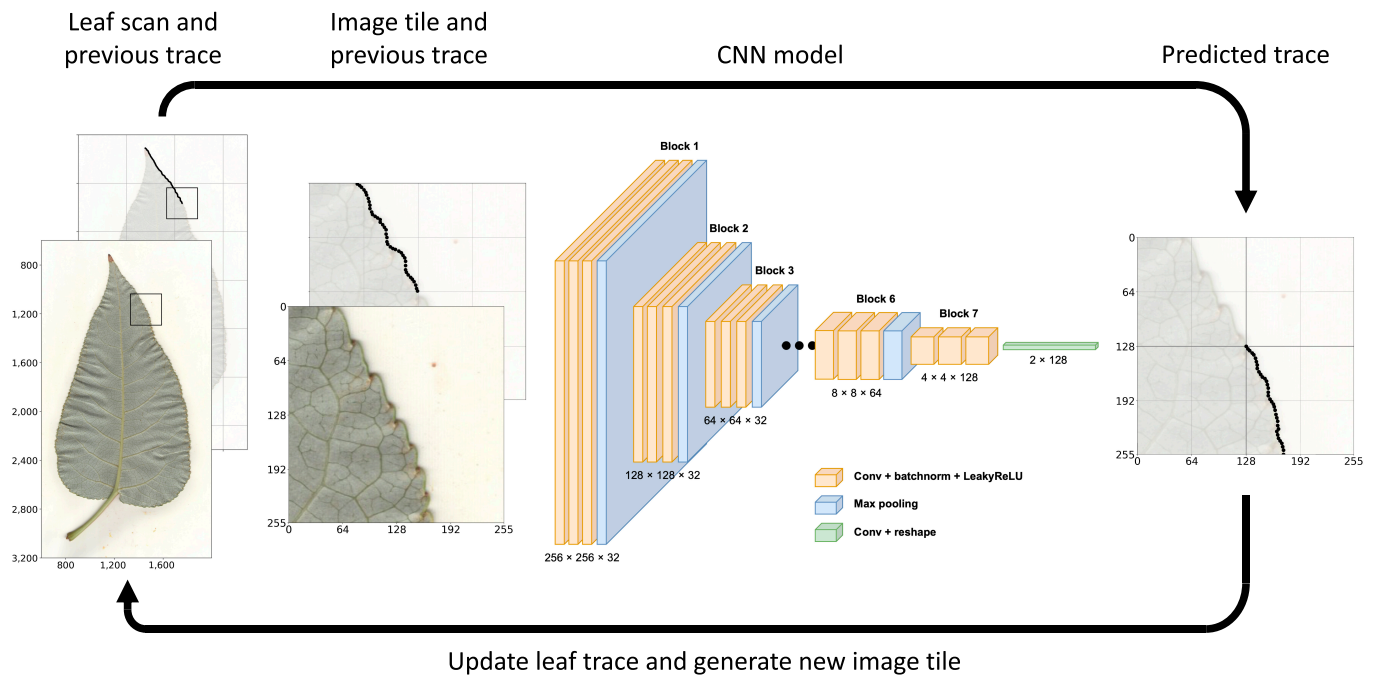
Fig. 1. Leaf tracing algorithm. An image tile and a small segment of the previously traced path are input to a CNN that predicts the next steps of the trace. The predictions are added to the leaf contour and used to generate an image tile at the new location. This iteration continues until the trace reaches the starting location of the contour. Left: RGB leaf scan and input tile (front) with previously traced pixels (back). Center: The leaf tracing CNN which transforms the 256 × 256 input tile into a 2 × 128 set of pixel displacements for trace prediction. Right: Predicted pixel displacements that are used to update the trace and generate image tiles in the next iteration.

values are normalized to $[0, 1]$ for computational stability. The additional channel is a binary image composed of ones along pixels of the previously traced path and zeros otherwise and thus provides the network with a direction to continue the trace. Each image tile is centered at a pixel on the contour of a leaf, by which the 50 manually traced samples are used to generate more than 300,000 individual tiles for training. Further, heavy image augmentation is used so that the CNN learns an invariance to heterogeneous leaf shapes and conditions. In particular, random continuous rotations, horizontal and vertical flips, displacement jitter, and color augmentation (hue, saturation, brightness, and contrast) are combined to increase image diversity during training.

The leaf tracing CNN outputs $2 \times N$ trace predictions, which encode N horizontal and vertical pixel displacements along the leaf contour relative to the center pixel of the input tile. Training data is generated by evenly sampling pixels from the center to the edge of the tile along the contour of the leaf. Distance is then measured between the predicted trace and the ground truth contour using mean squared error, $\mathcal{L}_{\text{MSE}}$, as an objective function. Importantly, the quality of the predicted trace degrades near the edges of image tiles since the CNN does not have context beyond the boundaries of the input. However, it is still important for the model to predict the trace from the center pixel to the edge of the image tile so that the predicted trace can "skip" over obscured segments of the contour [8]. To account for these effects, weighted mean squared error is used to weight predictions closer to the center pixel more heavily than predictions near the edge. The objective function is given by

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \omega_i \, \| \, y_i - \text{CNN}(x)_i \, \|_2^2 \,, \qquad (1A)$$

$$\omega_i = 1 + \frac{1 - \tanh(\alpha i + \beta)}{2}, \qquad (1B)$$

where $x \in \mathbb{R}^{256 \times 256 \times 4}$ is the input image tile, $y \in \mathbb{R}^{2 \times N}$ is the set of ground truth row/column coordinates with y_i indicating the row and column position of the ith pixel, $\omega \in \mathbb{R}^N$ is the weight vector, the number of pixel displacements is $N = 128$, and $\alpha = 8/N$ and $\beta = -4$ are chosen such that the hyperbolic tangent function (which defines ω) gradually decreases the error weight from 2 to 1 along the predicted contour. In this way, the objective function weights pixels near the center of the tile approximately twice compared to pixels near the edge.

The model architecture follows standard practices for CNNs [2,42]. In particular, the CNN uses blocks of three 3×3 convolution layers with zero-padding and a max pooling layer. Each convolution layer includes batch normalization to stabilize training [43] and a "LeakyReLU" activation function for nonlinearity [44]. Additionally, residual connections are applied between convolution layers for easier optimization and better prediction accuracy [2]. In total, the leaf tracing CNN includes 6 blocks with max pooling and the final block without, which transforms the spatial image dimensionality from 256×256 to 4×4. Then, a final 4×4 convolution layer reduces the outputs to a vector of length 256, which is reshaped into 2×128 for trace prediction. Note that the final convolution is linear (i.e., it does not include a nonlinear activation function) so that the trace predictions can reach the edges of the input tile in any direction. To prevent overfitting, images are randomly split into 80% training (i.e., 40 images totaling ~250K image tiles) and 20% validation (i.e., 10 images totaling ~50K image tiles) sets. The model is then trained for 1,000 epochs with a batch size of 256 and the Adam optimizer [45] with default parameters. Further, early stopping with 20 epochs (i.e., training is

stopped if the validation error does not improve within 20 epochs) is used to guarantee the convergence of the model.

Once the leaf tracing CNN is trained, it is used to iteratively trace the contour of each leaf image in the dataset. The tracing algorithm is initialized using automatic thresholding to obtain a rough segmentation of the leaf, which provides both a starting location and trace direction. An image tile centered at the top of the rough segmentation (i.e., at the tip of the leaf) is initially fed to the CNN, which outputs the initial trace prediction from the center to the edge of the image tile. The first 32 pixel predictions along the edge of the leaf are added to the trace, and a new image tile is drawn centered at the new location. This iteration continues until the predicted trace falls within 10 pixels of the previously traced contour, after which a line is drawn from the prediction to the contour to close the loop. To eliminate errors from the trace initialization, the tracing algorithm uses 10 "burn-in" iterations before storing traced pixels for the final segmentation. The leaf body segmentation is obtained by classifying all interior pixels as foreground and exterior pixels as background. Note that the trace direction is randomized during training, so that the tracing CNN can segment leaves in either clockwise or counterclockwise directions. In practice, the tracing algorithm does not require human supervision to start or stop the iteration and takes $\mathcal{O}(1)$ s per image on a single graphics processing unit (GPU) of an NVIDIA DGX Station A100.

Vein segmentation

Segmentation of the leaf venation is formulated as a region growing task based on [9,10], in which a CNN is used to iteratively expand a region of interest (i.e., visible veins of a leaf). The convolutional region growing method (also called flood filling networks [9]) has been shown to reach state-of-the-art segmentation accuracy while preserving biologically realistic morphological features [10]. However, rather than tracing the boundary of an object with a 1-dimensional line, the vein growing CNN iteratively grows a segmentation in all directions (e.g., 2-dimensional [2D] in [10] and 3D in [9]) by classifying which pixels/voxels should be included in or rejected from the region. In particular, a CNN inputs small image tiles centered on pixels of interest and predicts classifications of the center pixel and its adjacent neighbors. Neighboring pixels that are added to the region become the seeds for new image tiles in the next iteration. This process continues until no new pixels are added to the region, thereby finishing the segmentation. Similar to the leaf tracing framework, the region growing approach breaks the complex task of vein segmentation into many smaller decision rules and can produce high-accuracy segmentations using fewer than 10 images for training. See Fig. 2 for a diagram of the vein growing algorithm and Movie S2 for a video of the iteration.

The vein growing CNN inputs $128 \times 128 \times 3$ RGB image tiles (also normalized to $[0, 1]$) centered on pixels in the interior of a leaf. The tile size is chosen to be smaller than the leaf tracing tiles (a) since vein classification does not require as much context and (b) for computational efficiency, since many more image tiles are used in this framework. However, the tile size is still large enough so that the model can accurately predict vein pixels in areas of uncertainty (e.g., blurry patches and diseased/dead tissue). To construct a training set, image tiles are drawn for each vein pixel, by which the 8 manually segmented images
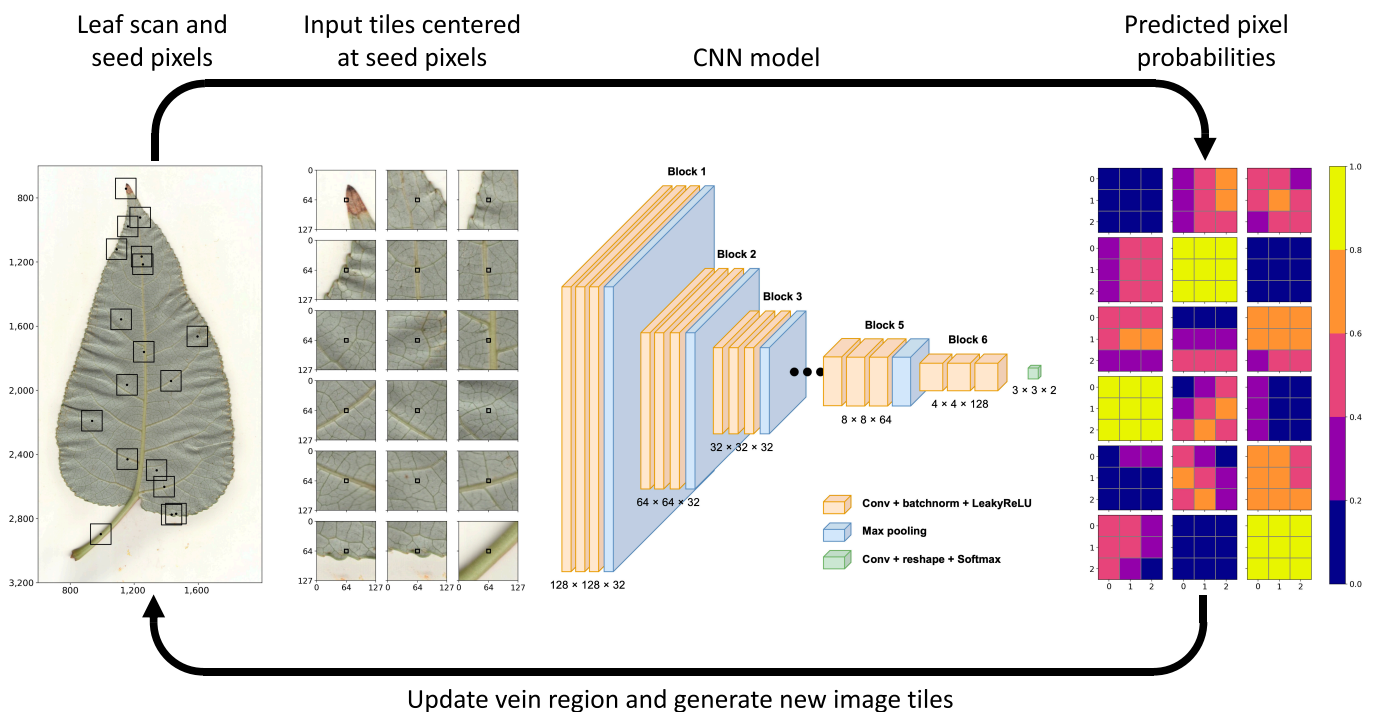

Fig. 2. Vein growing algorithm. Image tiles centered on pixels of interest are input to a CNN which predicts the classification of the center pixel and its neighbors. Neighboring pixels with high probability are added to the vein region and used as seed pixels in the next iteration. The iteration continues until no new pixels are added to the vein region. Left: RGB leaf scan and input tiles with center pixels highlighted in black. Center: The vein growing CNN that transforms the 128×128 input tile into a 3×3 matrix of vein probabilities. Right: Predicted pixel probabilities that are used to update the region and generate new image tiles in the next iteration.

generate more than 1,000,000 positive samples (i.e., samples centered on leaf veins rather than leaf lamina). Further, to account for heterogeneous backgrounds and image artifacts, up to 10 times as many background pixels are sampled from the interior of each leaf. The image tiles are augmented during training using a combination of random continuous rotations, horizontal and vertical flips, color augmentation, and Gaussian blur.

The vein growing CNN outputs $3 \times 3 \times 2$ predictions of the center pixel and its neighbors, in which the 2 prediction channels represent probabilities that a pixel belongs to the foreground (vein) or background (lamina). To measure error between predicted pixel probabilities and their ground truth classifications, Focal Loss ($\mathcal{L}_{FL}$), an extension of standard binary cross-entropy (BCE), is used as an objective function [46]. In particular, Focal Loss seamlessly accounts for the class imbalance between positive and negative samples (i.e., there are many more background pixels than vein pixels) and allows the model to focus on more difficult examples where veins are obscured. The objective function is given by

$$\mathcal{L}_{FL} = \begin{cases} -\alpha \ (1-p)^\gamma \ \log(p) & \text{if } y=1 \\ -(1-\alpha) \ p^\gamma \ \log(1-p) & \text{otherwise,} \end{cases} \quad (2)$$

where $p = CNN(x)$ are the pixel probabilities, $x \in \mathbb{R}^{128 \times 128 \times 3}$ is the input image tile, $y \in \{0,1\}$ are the ground truth pixel classes, and $\alpha = 0.25$ and $\gamma = 2.0$ are the default hyperparameters of the Focal Loss (FL) function [46]. Vein growing CNNs are trained using both FL as well as BCE and compared in the results section.

The model architecture and training strategy are nearly identical to the leaf tracing framework. Since the input tiles for vein segmentation are half the dimension of the inputs for leaf tracing, the first block of 3×3 convolutional layers and max pooling is removed from the architecture described in the leaf segmentation section. Thus, the vein growing CNN transforms the spatial image dimensionality from 128×128 to 4×4, after which a final 4×4 convolution layer reduces the outputs to a vector of length 18, which is reshaped into $3 \times 3 \times 2$ for vein classification. Note that, unlike the leaf tracing CNN, the final convolution includes a Softmax activation function, which constrains the outputs to between 0 and 1 and motivates the probabilistic interpretation for the objective function. The model is trained with the Adam optimizer for 1,000 epochs with a batch size of 1,024 and early stopping of 20. Note that a larger batch size is used here compared to the leaf tracer since the inputs are smaller and thus more can be included in each batch. Finally, 6 images (totaling ~6.7M image tiles) are used for training and 2 (totaling ~2.5M image tiles) for validation.

Once the vein growing CNN is trained, it is used in a recursive framework in which the CNN decides whether new pixels should be added to the vein segmentation. The algorithm is initialized by randomly sampling 10,000 seed pixels inside the leaf body (using the leaf segmentations). For each seed pixel, image tiles are generated and fed to the model, which then classifies the seed pixel and its neighbors. Neighboring pixels that are classified as leaf veins are used as seeds in the next iteration. Once a seed pixel has been considered, it is removed from the sample set for future iterations. This process is then repeated, continuously adding pixels to the segmentation, until no new pixels are positively classified. Note that a pixel can receive multiple classifications as its neighbors become seeds during the iterations. To account for this, the final vein

segmentation is determined by thresholding the average probability of each pixel. The optimal probability threshold is chosen by minimizing the number of connected components in the segmentation mask across a range of threshold values. In other words, the optimal threshold is the one that maximizes vein connectivity in the segmentation mask. Like the leaf tracing framework, the vein growing algorithm does not require human supervision at inference time and completes accurate vein segmentations in $\mathcal{O}(10)$ s on a single GPU of an NVIDIA DGX Station A100, which is orders of magnitude faster compared to human annotation.

Baseline comparison

A variant of the state-of-the-art image segmentation model, U-Net [3], is used as a baseline to compare against the leaf tracing and vein growing methods. Unlike the leaf tracing and vein growing CNNs, which consist of only an encoder network, U-Net uses an additional decoder network in an auto-encoder-like model structure. However, instead of the decoder network leveraging information from just the final layer of the encoder, U-Net includes skip connections from the encoder to the decoder at each spatial resolution so that the model may leverage multiple levels of information to accurately segment images. Following the same strategy for the leaf tracing and vein growing methods, U-Net is also trained on smaller image tiles that are then combined during prediction to reconstruct the image-level segmentation mask. See Fig. 3 for a diagram of the model architectures used for leaf and vein segmentation.

Separate U-Net models are trained for leaf and vein segmentation. For leaf segmentation, following a similar procedure outlined in the leaf segmentation section, U-Net inputs $256 \times 256 \times 3$ RGB image tiles that are $[0,1]$ normalized and outputs corresponding 256×256 probability maps that predict pixels belonging to the leaf body. However, unlike the leaf tracing inputs, U-Net inputs are not constrained to be centered along the leaf boundary and are instead sampled from anywhere in the leaf. For vein segmentation, following the vein segmentation section, U-Net inputs $128 \times 128 \times 3$ RGB image tiles that are $[0,1]$ normalized and outputs corresponding 128×128 probability maps that predict pixels belonging to the venation. Similarly, these input tiles are sampled from anywhere in the leaf. Each U-Net is trained using the same image augmentation techniques that are described in the leaf and vein segmentation sections.

The baseline models used in this work are variants of the original U-Net architecture described in [3] and are adapted from the CNN models described in the leaf and vein segmentation sections. In particular, each U-Net model uses an identical encoder to the leaf tracing and vein growing CNNs. However, rather than applying a final convolution layer to the encoder outputs, a decoder network is applied instead. The decoders use the same number of layers as the encoders, but with layer widths in reverse order (i.e., layer width increases with depth in the encoder and decreases with depth in the decoder). Skip connections are drawn from the activation maps preceding each pooling layer in the encoder and concatenated with the corresponding upsampled activation maps in the decoder. Upsampling is achieved using 2D transpose convolutions. The outputs of the last layer of the decoder are fed to a final convolution layer with a Sigmoid activation, which
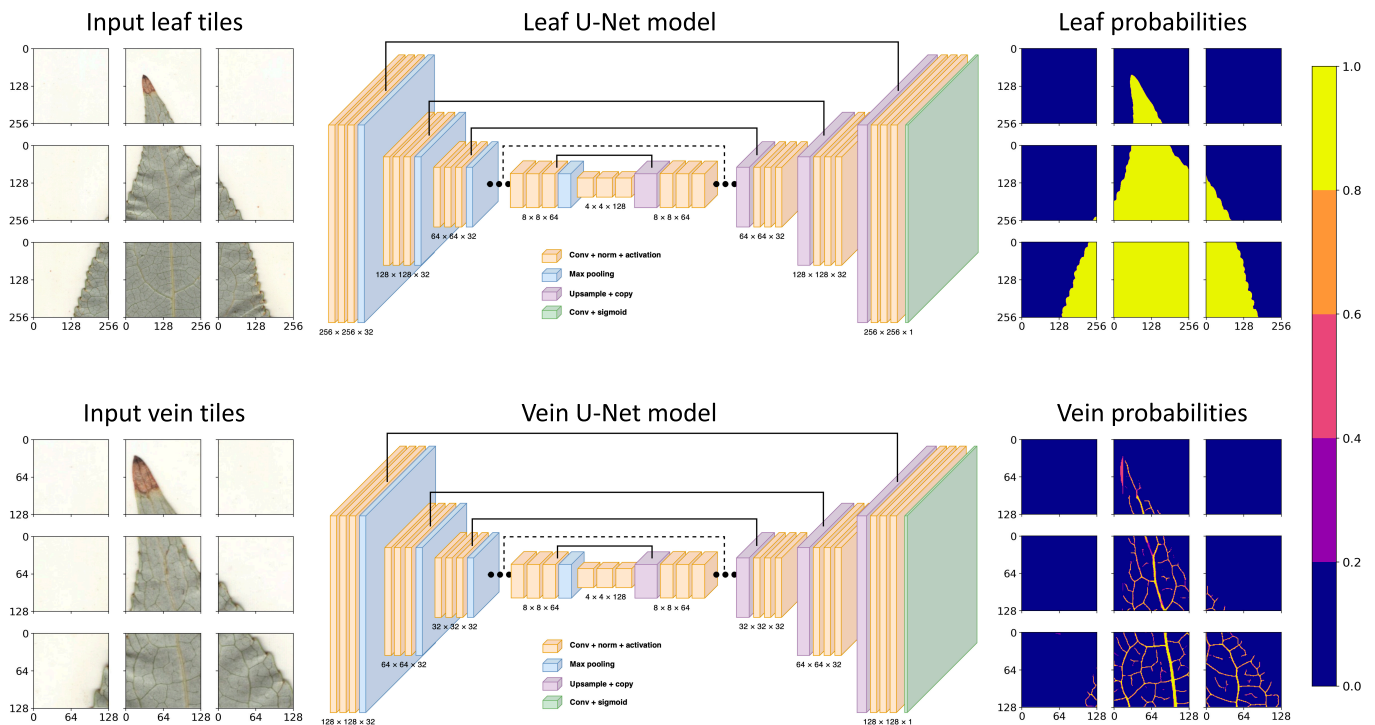
Fig. 3. Baseline models. U-Net is used as a baseline for comparison, in which large images are broken into smaller image tiles for prediction. Note that input tiles are sampled such that they overlap during prediction to account for edge effects. Top: RGB input tiles (256×256×3) are fed to a U-Net model that predicts a 256×256 matrix of leaf pixel probabilities. Bottom: RGB input tiles (128×128×3) are fed to a U-Net model that predicts a 128×128 matrix of vein pixel probabilities.

transforms the outputs into a 1-channel probability map used for pixel classification.

To compare the models directly, each U-Net is trained using the same training and validation sets as the leaf tracing (i.e., 40 training images totaling ~250K image tiles and 10 validation images totaling ~50K image tiles) and vein growing (i.e., 6 training images totaling ~6.7M image tiles and 2 validation images totaling ~2.5M image tiles) models. The models are trained for 1,000 epochs with early stopping set to 20 iterations. However, due to memory constraints (i.e., U-Net uses approximately twice the memory compared to the leaf tracing and vein growing CNNs), half of the batch size is used for the respective segmentation tasks (i.e., 128 for leaf segmentation and 512 for vein segmentation). Following the leaf and vein segmentation sections, U-Net is trained using BCE for leaf segmentation and BCE and FL for vein segmentation, which are compared in the results section.

For whole-image segmentation of the leaf body and vein architecture, each image is broken into uniform grid of small overlapping image tiles. For leaf tracing, 256×256 tiles are sampled with a step size that is half of the window size (i.e., 128). Similarly for vein segmentation, 128×128 tiles are sampled with a step size of 64 pixels. In this way, each pixel is considered from multiple positions in the image, thereby reducing poor predictability near the edges where the network has less context. After all of the image tiles have been predicted by U-Net, the average probability is computed for each pixel and then thresholded to produce the final segmentation mask. For leaf segmentation, the probability threshold is set to 0.5. For vein segmentation, the optimal probability threshold is chosen to minimize the number of connected components in the vein structure in order to maximize vein connectivity. Like the leaf

tracing and vein growing methods, the U-Net prediction algorithm requires no human interaction and completes segmentations in $\mathcal{O}(1)$ s for leaf segmentation and $\mathcal{O}(10)$ s for vein segmentation on a single GPU of an NVIDIA DGX Station A100. Note that for vein segmentation, though both methods segment images in the same order of magnitude of seconds, U-Net is consistently 2 to 3 times faster owing to the smaller number of image tiles the model needs to predict (e.g., 20 s rather than 60 s per image).

Feature extraction

Given the binary segmentation maps from the leaf and vein segmentation sections, traditional open-source image-processing tools are used to extract biologically meaningful traits from the leaf body, vein architecture, and petiole. This is possible since the segmentation methods effectively remove background artifacts and highlight the salient information in leaf scans. In this work, Fiji [32] is used to extract leaf-level traits, RVE [33,34] is used for vein traits (e.g., length and thickness), and a custom implementation is used for petiole traits (length and width). RVE is chosen for vein traits in particular since it is designed to analyze root systems, which are composed of vessel-like structures with tips, branch points, redundant connections, etc., and makes it applicable to studying vein architectures, which share many of the same characteristics. Further, since the scan resolution is known (i.e., 300 DPI), features extracted from Fiji and RVE are easily converted from pixel coordinates to standard units (e.g., centimeter).

Fiji is applied to the leaf segmentations to extract 23 image-based traits related to whole-leaf morphology. Some morphological descriptors include area (square centimeter), perimeter

(centimeter), circularity (unitless), and solidity (unitless). Color features are also derived by relating the segmentations back to the original scanned images, including average red, green, blue, hue, saturation, and brightness values corresponding to leaf pixels. Feature extraction in Fiji is scripted and applied in "batch mode" to the full set of leaf segmentations. A detailed description of each of leaf-level trait is provided in (Table S1).

RVE is used to extract 27 features from the vein segmentations. Note that only vein pixels inside the leaf segmentation were used for vein architecture traits (i.e., the petiole is not considered here). The software parameters are set to 300 DPI and "whole mode" for image-level traits. Vein diameter ranges are used classify veins into ranges: (a) less than 0.25 mm, (b) between 0.25 and 0.80 mm, and (c) above 0.80 mm, in an attempt to correspond to third-, second-, and first-order veins, respectively. Extracted traits include those supplied by default (e.g., average vein diameter [millimeter], length [millimeter], and area [square millimeter]), with some traits being repeated across the vein diameter ranges. Following [22], additional venation traits are also derived that measure proportions between vein length/area to leaf morphology. See Table S2 for the full list of vein traits and their descriptions.

Petiole segmentations are derived by considering the largest connected component of vein pixels outside of the leaf segmentation. Then, to compute petiole length and width, Fiji is used to compute the best-fit rotated rectangle around the petiole mask. The height of the bounding rectangle is sufficient to estimate petiole length. However, rectangle width is not used to estimate petiole width since (a) petiole width changes along the length of the petiole (i.e., it tends to be wider near the ends and thinner near the midpoint) and (b) the caliper measurements for petiole width were taken near the center of the petiole. Thus, petiole width is estimated by computing the average diameter over the center 20% of the segmentation. Finally, Fiji is used to estimate similar traits for the petiole compared to the leaf body (e.g., area and perimeter), and RVE was used to estimate petiole volume. See Table S3 for the full list of 18 petiole traits and their descriptions.

The feature extraction process yields 68 traits related to leaf, vein, and petiole morphology that can be used for genomic analysis. To validate image-based features with real-world measurements, petiole length and width are compared against caliper measurements that were recorded manually during image capture. To consider the results from a biological perspective, (a) broad-sense clonal heritability is computed for each recorded trait and (b) a GWAS is performed for the vein density trait (i.e., the ratio of vein area to leaf area). Vein density is chosen since it utilizes both the leaf and vein segmentations and since the ratio between lamina and venation must balance sunlight intake and carbon fixation with the transport of sugars and other nutrients to sink organs, all of which are essential processes for biomass production.

Validation

To validate the leaf and vein segmentations, following [7,10], the Jaccard index (intersection over union) is used to measure segmentation accuracy for images in the validation sets (i.e., 10 for leaf segmentation and 2 for vein segmentation). This metric measures similarity between semantic segmentations by computing the ratio between the set intersection (all true-positive pixels) and the set union (all true-positive, false-positive,

and false-negative pixels), where scores near one indicate high accuracy and near zero indicate low accuracy. These values are compared between the leaf tracing CNN, vein growing CNN, and baseline U-Net models.

To account for cases in which set similarity metrics like the Jaccard index may be misleading, an additional perspective is taken to measure the biological accuracy of the vein segmentations. In particular, the leaf vasculature is made up of a single connected network of veins. In other words, a perfect segmentation mask of the vasculature should, in theory, contain a single connected component. In practice, this is exceedingly difficult to obtain since veins are only partially observable from the surface of the leaf lamina (i.e., veins appear and disappear throughout the leaf surface), meaning that even the ground truth segmentations include multiple connected components. Despite this observation, an accurate vein segmentation must not only overlap with the ground truth mask (i.e., maximize the Jaccard index) but must also be biologically realistic (i.e., minimize the number of connected components). Thus, in addition to computing the Jaccard index for vein segmentations, the number of connected components are also computed and compared across the vein growing CNN and U-Net baseline models.

Note that since validation error was monitored during training, conclusions drawn from segmentation accuracy for validation images may be affected by data leakage (i.e., create an overoptimistic interpretation of the model). To account for this, the predicted digital measurements across the population are further validated using real-world physical measurements. To this end, calipers were used to measure petiole length and width during data collection. These values are compared against the corresponding features extracted from the vein segmentations. To measure the agreement between digital and manual values quantitatively, the coefficient of determination (R^2) is computed for each trait.

Genomic analysis

To preprocess the vein density trait for GWAS, outliers are removed using median absolute deviation (MAD), where any measurement with MAD > 6 is removed. To account for geospatial variation across the plantation, thin-plate spline (TPS) correction is applied using the fields software package in R [47], in which the row and position of each tree are used as coordinates for the TPS models. To extract the genetic component of each sample, best linear unbiased predictors (BLUPs) are computed for the TPS-corrected values using the lme4 software package in R [48], which fits genotypes as random effects for each trait. In addition, to assess the repeatability of each measurement and the genetic control of the vein density trait, broad-sense heritability (H^2) was estimated using the TPS-corrected values of the clonal replicates (131 replicated samples) from the blocks considered in this work. Heritability is computed by

$$H^2 = \frac{\sigma_G^2}{\sigma_G^2 + \sigma_E^2}, \qquad (3)$$

where σ_G^2 is the genotypic variance due to clonal differences and σ_E^2 represents environmental variance.

For genomic analysis, a total of 1,492 *P. trichocarpa* accessions were previously sequenced using the Illumina genetic analyzer

with paired-end sequencing technology at the Department of Energy Joint Genome Institute [49]. The sequences are aligned to the v4 reference genome using the Burrows-Wheeler Alignment tool, BWA-MEM [50], and variant calling is performed using the GATK (version 4.0) Haplotype caller [51]. Starting with more than 22 million single-nucleotide polymorphisms (SNPs) obtained by the GATK Variant Quality Score Recalibration method at tranche 99, 847,066 SNPs across 1,419 genotypes were retained for population-scale genomic analysis after applying the following filters. Seventy-three individuals were removed due to having excessive genomic relatedness or having greater than 10% missing SNP data. SNPs were removed if they had greater than 15% missing genotypes, or minor allele frequency less than 0.05, or Hardy Weinberg Equilibrium chi-square test P value $< 10^{-50}$. SNPs were further pruned using a linkage disequilibrium coefficient of determination threshold of $R^2 \geq 0.7$.

The data preprocessing steps above yield 847,066 SNPs for 1,419 unrelated genotypes that are used for GWAS analysis of the vein density trait. Association between the SNPs and the phenotypic vector was tested using a multilocus GWAS method, Bayesian-information and Linkage-disequilibrium Iteratively Nested Keyway (BLINK), from the GAPIT3 software package in R [52], that uses 2 fixed-effect models (FEM) iteratively. The first FEM tests for the association of all genetic markers independently to generate a set of pseudo quantitative trait nucleotides (QTNs) that are then used in the second FEM to optimize the selection of pseudo QTNs. Only those QTNs that are significant and not in linkage disequilibrium are used as covariates in the association test. The first FEM is given by

$$y_i = S_{i1}b_1 + S_{i2}b_2 + \cdots + S_{ik}b_k + S_{ij}d_j + e_i, \qquad (4)$$

where y_i is the phenotypic value of the ith individual, $S_{i1}, \ldots, S_{ik}$ are the genotypes of the k QTNs, $b_1, \ldots, b_k$ are the corresponding effects of the QTNs, S_{ij} is the genotype of the ith individual and j^{th} SNP, d_j is the jth SNP effect, and e_i is the residual. The second FEM is used to optimize the QTNs for use as covariates in the first FEM and is given by

$$y_i = S_{i1}b_1 + S_{i2}b_2 + \ldots + S_ib_k + e_i, \qquad (5)$$

with a similar interpretation to Eq. 4. Note that Eq. 5 is essentially a reduced version of Eq. 4, in which the SNP term that tests for the association with the phenotypic vector is removed. The model optimization is performed with Bayesian information criterion.

Results

Segmentation results
The few-shot learning and baseline methods are applied to the total set of images, in which the 2,906 top and bottom scans are used for leaf segmentation, and the 1,453 bottom scans are used for vein architecture. Examples of the resulting model outputs are given in Fig. 4. Note that the leaf in Fig. 4 was not used for model training or validation. Additional segmentation results are visualized in (Fig. S1), which illustrates leaf heterogeneity by varying leaf size and vein density. All of the image data, ground truth annotations, and predicted leaf/vein segmentations are made publicly available at [31,36].

The Jaccard index is used to measure segmentation accuracy for images in the validation sets (i.e., 10 for leaf segmentation and 2 for vein segmentation), where scores near one indicate high accuracy and near zero indicate low accuracy. For leaf

segmentation, the leaf tracing CNN achieves a mean ($\pm$ SD) Jaccard score of 0.9946 ($\pm$0.0016) while the baseline U-Net model achieves a score of 0.9969 ($\pm$0.0009), indicating a high degree of overlap between the predicted and ground truth segmentations and virtually no difference in accuracy between the methods. For vein segmentation, the 2 validation images are used to compare Jaccared scores between the vein growing CNN and U-Net models and BCE and FL objective functions and are given in Table 1. Note that to compare each method equally, the Jaccard scores are computed using a probability threshold of 0.5 for each case. This is in contrast to the other results and visualizations, which use thresholds that minimize the number of connected components in each segmentation mask.

The reduced scores for vein segmentation compared to leaf segmentation is due mainly to (a) human errors in the ground truth segmentation and (b) the complexity of the vein architecture. For example, the model identifies veins that were missed during manual annotation and thus are considered false positives in the Jaccard score. Further, due to thin veins, predicted veins that are off by just 1 pixel can result in large changes in Jaccard score. For a visualization of these phenomena, see (Fig. S2), which illustrates these effects for the validation image with the lowest Jaccard score. Despite the lower Jaccard metric, the vein growing framework achieves recall/sensitivity values (i.e., the probability of detecting a vein pixel) of 0.9219 and 0.8673, respectively, which indicates that the method has a high detection rate, and thus almost completely captures the structure of the visible vein architecture.

In addition to Jaccard index, the number of connected components in each vein segmentation is computed as a proxy for biological accuracy. In other words, since the vein architecture of a leaf is a single fully connected network, a biologically accurate vein segmentation should minimize the number of connected components. To this end, bar plots are constructed for the vein segmentations arising from the vein growing and baseline CNN models and are visualized in Fig. 5. Note that vein objects are counted using custom probability thresholds that minimize the number of connected components in each image for each model as described in the vein segmentation and baseline comparison sections.

Figure 5 demonstrates that the vein growing CNN, while producing more fragmented segmentations compared to the ground truth masks, significantly reduces the number of vein objects compared to U-Net, regardless of objective function. Further, while the objective function does not significantly change the number of connected components for the vein growing CNN, there is a drastic difference between U-Net models. To quantify these observations, the mean number of connected components of each model's outputs is compared using Tukey's honestly significant difference (HSD) test with $\alpha = 0.05$. This test reveals that the 4 models are reduced to 3 statistically significant groups, A: U-Net with BCE, B: U-Net with FL, and C: vein growing CNN with BCE and FL. These findings show that the 3 groups differ statistically significantly in value, with the vein growing CNN producing the smallest number of vein objects. Thus, since the vein growing CNN with FL results in the highest segmentation accuracy see (Table 1) and approximately the same biological accuracy to BCE (see Fig. 5), it is chosen as the optimal model for vein segmentation in this work. In particular, this model is used for feature extraction, validation with physical measurements, and downstream genomic analysis.
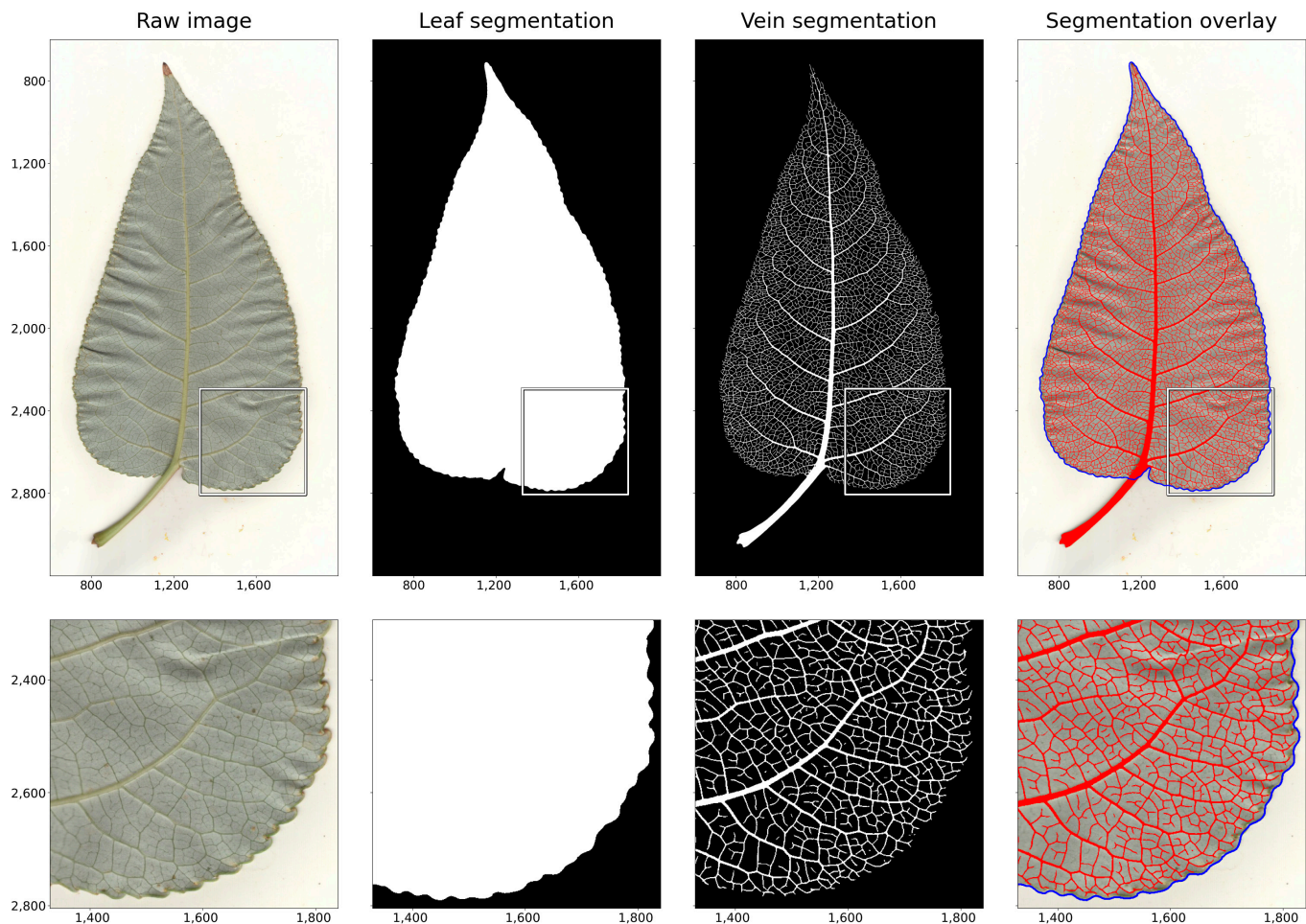
Fig. 4. Leaf and vein segmentations. Results of the leaf and vein segmentation methods on an example leaf outside the training set. The top row shows the full leaf and the bottom row gives a zoomed in view. Left: Example leaf scan chosen from outside the training and validation sets. Center left: Predicted segmentation of the leaf body using the leaf tracing CNN, where pixels inside the traced contour are shown in white and outside the contour in black. Center right: Predicted segmentation of the visible vein architecture using the vein growing CNN with FL, where vein pixels are shown in white and background pixels in black. Right: Example leaf scan with the predicted leaf boundary and vein architecture overlaid in blue and red, respectively. Note that for visualization these images are zoomed in to remove redundant white space from the scanner background.

Table 1. Vein segmentation accuracy. Jaccard index is used to quantify vein segmentation accuracy for the 2 images held out for validation (see the top of each table for the leaf ID). The vein growing CNN (Grower) is compared to the baseline model (U-Net) for both BCE and FL. The model/loss combination resulting in the largest Jaccard score is shown in bold.

C_1_14_18_bot.png		
	BCE	FL
U-Net	0.6312	0.5947
Grower	0.5990	**0.6586**

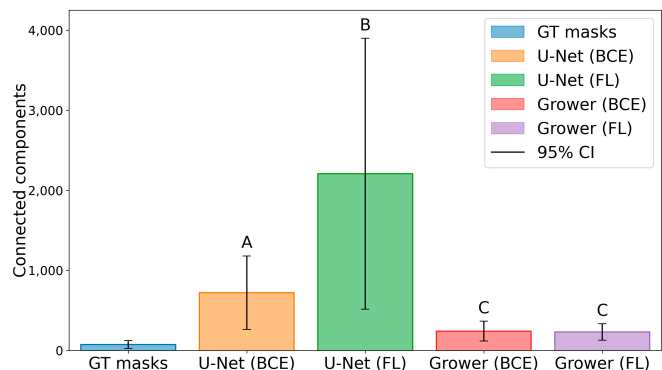| C_1_8_1_bot.png		
	BCE	FL
U-Net	0.5965	0.5633
Grower	0.6047	**0.6381**

Fig. 5. Vein object counts. Biological accuracy of the vein segmentations is measured by counting the number of connected components in the 8 ground truth (GT) and sets of 1,453 predicted masks. The vein growing CNN (Grower) is compared against the baseline model (U-Net) trained using BCE and FL. The vertical axis represents the number of connected components in the vein segmentations. The height of each bar corresponds to the mean object count, with black lines indicating 95% confidence intervals (CI). The results of a Tukey's HSD test groups the means of the 4 models into 3 statistically significant groups: A, B, and C. Note that ground truth masks are not considered in this analysis due to the orders-of-magnitude difference in number of samples.

The predicted digital measurements across the population are further validated using real-world caliper measurements, which are visualized in Fig. 6. In particular, the data are compared against a linear model, which results in $R^2 = 0.96$ for petiole length and $R^2 = 0.77$ for petiole width. This discrepancy between R^2 values is due to several factors. First, manual measurement of petiole length is made from end to end, resulting in larger, more consistent measurements. However, for petiole width, the caliper was placed at the approximate center of the petiole, resulting in greater variation due to subjective positioning of the caliper. Further, since petiole width is typically smaller, measurement errors affect the R^2 value more compared to petiole length. Despite these effects, the digital traits strongly agree with the manual measurements, which further helps validate the accuracy of the segmentations and extracted features.

Genomic analysis results

To consider the segmentation and feature extraction methods from a biological perspective, a GWAS analysis is conducted at population-scale to associate vein density (i.e., the ratio of vein area to leaf area) to the *P. trichocarpa* genome. The broad-sense clonal heritability of the vein density trait is moderately high ($H^2 = 0.65$), which suggests that the trait is under genetic control. The multilocus BLINK method is used to perform GWAS on the vein density trait with 847,066 SNPs in the genome. This analysis identified 12 significant SNPs using a false discovery rate (FDR) P value, $P < 0.05$, and 15 unique SNPs using a less-strict FDR P value, $P < 0.2$. A Manhattan plot, quantile-quantile plot, and the distribution of the TPS-corrected vein density BLUPs are shown in Fig. 7. A total of 30 unique genes that potentially control for the variation in vein density trait are identified for these SNPs based on the nearest flanking genes in both directions of the GWAS hits in the genome. To gain more insight into the function of these genes, the *Arabidopsis thaliana* orthologs are identified based on the protein sequence similarity. The GWAS results are summarized in Table 2. See the discussion section for more information about these genes and their associated physiological plant processes.

Discussion

In this work, few-shot learning was used for both leaf and vein segmentation. In particular, each method divided a small number of large images into a large number of small image tiles and used the predictions of previous iterations to expand partial segmentations until stopping criteria was reached. In particular, iterative dataset refinement was paired with heavy image augmentation so that the CNN models learned an invariance to potential image artifacts not included in the small number of annotated images. In this way, the complex task of whole-image segmentation was broken down into smaller easier decision rules, thereby maximizing predictive accuracy while minimizing the amount of labeled images needed for model training. This strategy was chosen primarily due to (a) the lack of labeled training images (e.g., only 50 leaf segmentations and 8 vein segmentations) and (b) because each leaf scan is large (3, 510 × 2, 550 pixels) and does not easily fit into a standard deep learning model (e.g., a CNN). To address (a), one could simply annotate more data manually; however, to segment the visible venation in the images considered here (see the data collection section) would require up to 12,000 person-hours. Note that the approaches discussed in this work could be used to automatically label training data for larger deep learning workflows (e.g., U-Net [3] or Mask-RCNN [53]). For (b), large images are typically grayscaled, reshaped, or downsampled to fit within system requirements and hardware limitations [24]. However, this strategy can alter or destroy fine-grained details that may be crucial for accurate prediction. Thus, the methods demonstrated here utilized raw RGB images at full resolution, but
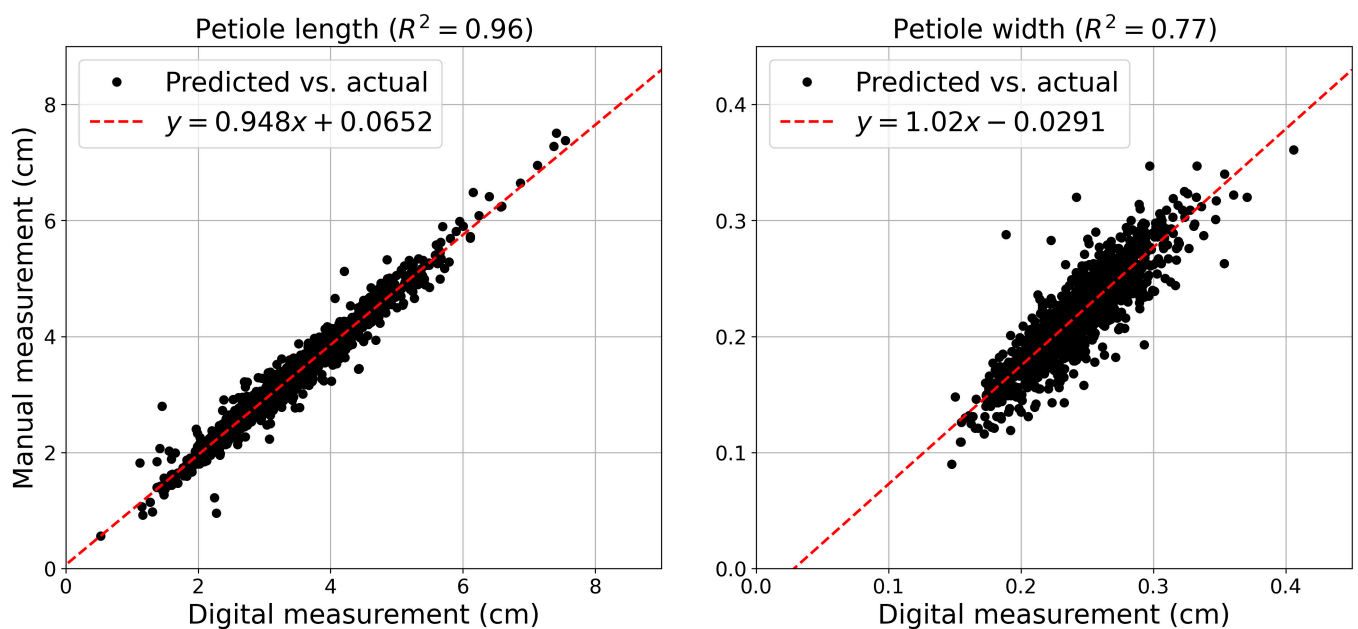

Fig. 6. Petiole measurement validation. Each subplot compares predicted (x-axis) with actual (y-axis) morphological measurements of the petiole. Manual measurements were obtained with calipers during data collection while digital measurements were derived from the leaf and vein segmentations. Left: Petiole length comparison with data shown in black and the best-fit line shown in red. Right: Petiole width comparison with data shown in black and the best-fit line shown in red.
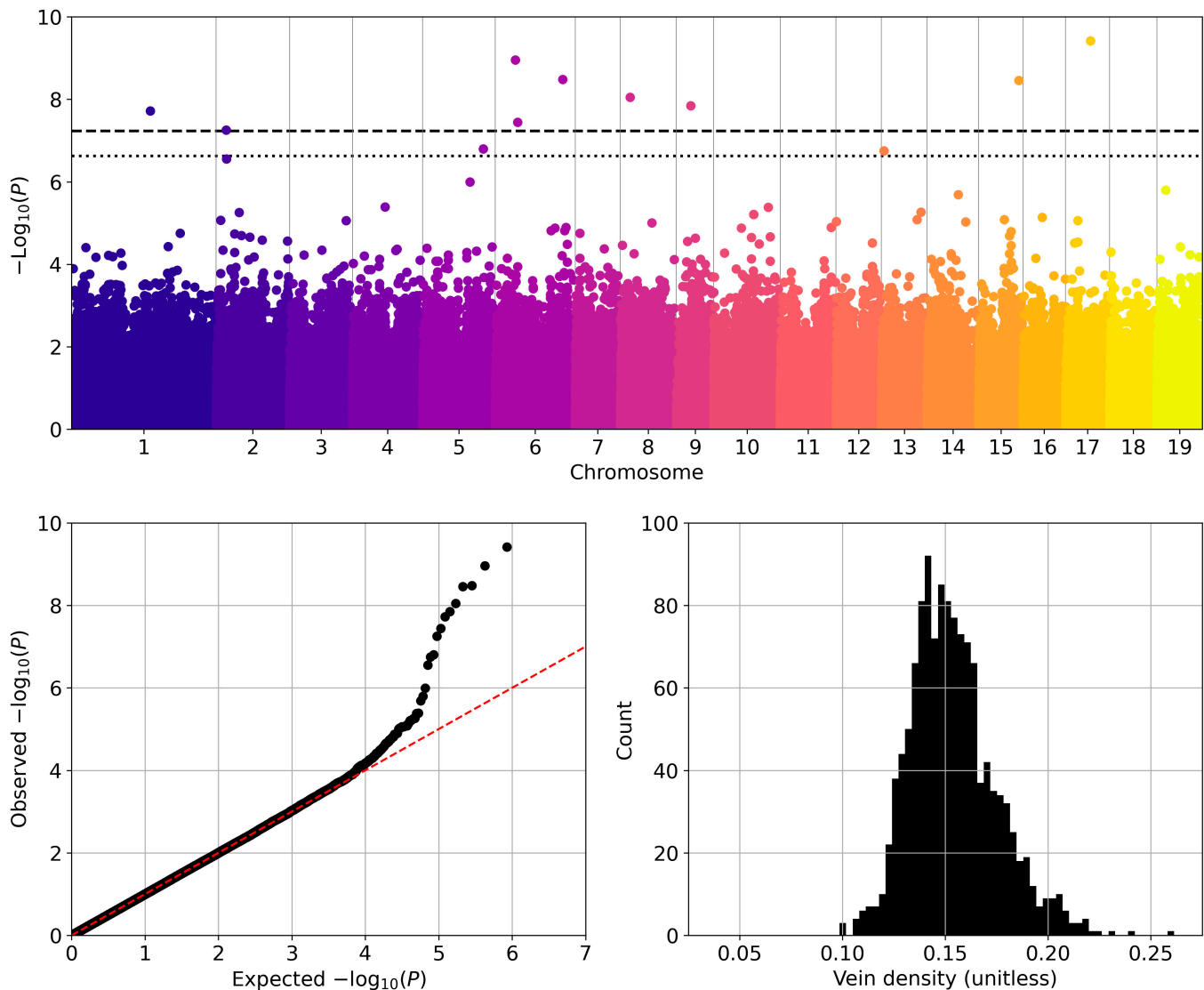
Fig. 7. Population-scale genomic analysis. Top: A Manhattan plot of the GWAS results for leaf vein density using the multilocus BLINK method for 847,066 SNPs across 1,419 *P. trichocarpa* genotypes. The horizontal axis corresponds to genomic positions by chromosome and the vertical axis shows the negative log-base-10 *P* value for each SNP. The dashed horizontal line represents the FDR threshold, *P* < 0.05, and the dotted line represents the FDR threshold, *P* < 0.2. Bottom left: The quantile-quantile plot corresponding to the *P* values shown in the Manhattan plot with the expected values shown by the red dashed line. Bottom right: The distribution of TPS-corrected vein density BLUPs used for GWAS.

Table 2. Identified genes. Top gene models detected by the BLINK GWAS method based on FDR *P* value *P* < 0.2 for vein density in *Populus trichocarpa*. Each row corresponds to different genes and includes the gene ID, chromosome number, SNP position, distance in the genome (positive for upstream and negative for downstream from the SNP position), minor allele frequency (MAF), *P* value, *A. thaliana* ortholog, and ortholog annotation.

Gene ID	Chr. (pos.)	Dist.	MAF	*P* value	Ortholog	Annotation
Potri.017G077200	17 (8,592,229)	−5,329	0.1246	3.24×10^{-4}	AT3G04680	CLP-SIMILAR PROTEIN 3
Potri.017G077300	17 (8,592,229)	6,036	0.1246	3.24×10^{-4}	AT3G01300	PBS1-LIKE 35
Potri.006G090501	6 (6,910,580)	−2,825	0.1938	4.69×10^{-4}	-	-
Potri.006G090600	6 (6,910,580)	1,617	0.1938	4.69×10^{-4}	AT3G53880	ALDO-KETO REDUCTASE FAMILY 4 MEMBER
Potri.006G227300	6 (23,159,363)	−3,829	0.2158	7.47×10^{-4}	AT1G18600	RHOMBOID-LIKE PROTEIN 12
Potri.006G227400	6 (23,159,363)	2,083	0.2158	7.47×10^{-4}	AT3G13784	CELL WALL INVERTASE 5

rather than using entire images as inputs, smaller tiles were sampled from within images to iteratively segment leaf boundaries and visible venation. For example, see (Fig. S1), which illustrates how these tile-based approaches are generally insensitive to changes in object size and characteristics.

Leaf segmentation was formulated as a tracing task, in which a CNN inputs image tiles centered along the boundary of a leaf, and outputs trace predictions that are used to sample new tiles in the next iteration. This methodology performed well in this application since there is only 1 leaf per image and each leaf is fully contained within the image. Note that images with multiple objects have been previously considered [7,8], but additional modifications to the algorithm (e.g., recurrence) may be needed to account for images with cluttered objects and overlapping boundaries. The resulting leaf segmentations were highly accurate and captured the morphology and serration of each leaf. Since each leaf was scanned against a white background, automatic thresholding could be used to obtain a rough segmentation of the leaf. However, this approach captures background artifacts, includes the petiole, and falsely detects shaded regions near the leaf boundaries and petiole. These artifacts could be addressed for individual samples by postprocessing the binary segmentation maps (e.g., binary erosion and dilation), but defining such rules that generalize to all cases across the population is increasingly difficult. In contrast, a strength of the tracing approach is that the tracing CNN can learn an invariance to image artifacts in a data-driven way without human supervision and still leverage the auto-threshold segmentations for trace initialization, which allows the method to be fully automated. Finally, the tracing methodology produced accurate leaf segmentations multiple orders of magnitude faster than human annotation ($\mathcal{O}(1)$ s compared to 15 to 30 min per image), allowing the method to scale up to population-level datasets, especially for computing systems that support parallelization (i.e., tracing multiple images at once).

Due to the complexity of leaf venation, vein segmentation was formulated as a region growing task where a CNN predicts whether to include pixels in a segmentation by inputting image tiles centered at those pixels. Unlike the tracing framework (which segments objects by tracing boundaries in 1 dimension), the region growing approach grows the segmentation directly by continuously adding pixels to the region of interest in 2D (see [9] for 3D). A strength of this approach is that each pixel is considered individually. However, unlike previous methods that conduct an exhaustive prediction over all pixels in an image (which would equate to 8,950,500 pixels per image in this work) [54], pixels were only considered if they exceed a probability threshold, which allowed the model to focus only on pixels of interest. This distinction dramatically increased segmentation speed ($\mathcal{O}(10)$ s compared to 4 to 8 h per image with manual segmentation). In addition, the region growing framework can use a random sample of pixels from anywhere in the image to initialize the iteration. Note that in this work, seed pixels were drawn using the leaf body segmentations from the tracer to reduce the number of redundant white background pixels. Finally, and perhaps most importantly, the model produced accurate segmentations at population scale using just 6 images for training and 2 images for validation. This is in contrast to previous approaches that used CNNs for vein segmentation and required more than 700 ground truth vein segmentations [24]. In particular, this result highlights the importance of iterative dataset refinement, in which images were specifically added to

the training set in order to build an invariance to observed artifacts in the population (e.g., leaf folds that were falsely classified as veins), which has also been noted in similar applications for root segmentation [55].

Leaf and vein segmentation were specified as independent tasks and used separate computational strategies to achieve each goal. Since both approaches were designed for segmentation, a natural question arises concerning whether distinct approaches are necessary, or whether a single method would suffice for both tasks. In principle, the region-growing CNN could be used for leaf segmentation; however, this would be computationally inefficient due to the large number of leaf pixels in the high-resolution scans. Compared to the tracing CNN (which focuses solely on boundary pixels), the region-growing CNN would waste a large amount of computational resources on redundant "interior" pixels, which vastly outnumber boundary pixels. In the reverse case, the tracing CNN could theoretically be applied to vein segmentation. However, since the vein architecture is not homotopic to a circle, the tracing CNN would need to be reinitialized thousands of times inside the leaf to account for all of the "holes" in the vein architecture. Further, overlapping tracer boundaries near 1-pixel-thick veins would create additional challenges in postprocessing the thousands of traced contours. Thus, iterative methods of this kind may only be effective for certain types of segmentation tasks. In particular, images containing multiple objects that overlap and obscure each other may be less suitable for tracing, while high-resolution images with large objects may be computationally infeasible for region growing. Therefore, it is important to account for such nuances when selecting a method for a particular task. For example, in this work, each method was particularly well suited for the segmentation task it was assigned.

The leaf tracing and vein growing CNNs were compared with a variant of the state-of-the-art image segmentation model, U-Net. Almost every aspect of the models and training strategies were kept equal, e.g., identical encoder networks, same datasets, input sizes, augmentation techniques, loss functions, training parameters, thresholding techniques, etc., with only a small number of changes to account for fundamental differences between the models, e.g., output size and batch size. The U-Net models provided multiple strengths. In particular, whereas tracing and region growing approaches are applied separately to leaf and vein segmentation, U-Net is agnostic to these objectives and only differs in the size of the input/output resolution. Further, U-Net reported approximately equal segmentation speed and accuracy compared to the leaf tracing CNN. For vein segmentation, U-Net segmented images in the same order of seconds but consistently 2 to 3 times faster than the vein growing CNN. However, U-Net requires more than twice the memory per sample compared to the methods presented here (296.30 MB compared to 139.91 MB for leaf segmentation and 76.93 MB compared to 36.33 MB for vein segmentation), which may limit its use on smaller compute hardware (e.g., edge-computing devices for field deployment). Additionally, U-Net reported lower Jaccard scores and dramatically more fragmented segmentations compared to the vein growing CNN, which were quantified using Tukey's HSD test. Thus, the vein growing CNN is a more biologically accurate and memory-efficient model for vein segmentation, with important implications for downstream scientific analyses where vein connectivity is important (e.g., leaf hydraulic modeling, gas exchange,

length/branching estimation, biomechanics, etc.). Note that even the ground truth segmentations included multiple connected components, since veins appear and disappear throughout the leaf lamina, which necessitates further development of methods to infer these missing connections in future work.

A utility of the segmentation methods discussed here is that they remove background artifacts and highlight salient information in images (e.g., the leaf body or vein architecture). Using traditional computer vision applications (e.g., Fiji and RVE) to extract digital traits from such segmentations becomes trivial compared to using the raw image data. These advances not only reduce human effort but also expand the number and variety of traits one can extract by including traits that can only be estimated digitally (e.g., vein density, leaf solidity, etc.). Further, custom algorithms can be developed that extract cryptic phenotypes related to leaf morphology and topological information from the vein networks, which may yield new biological insights into the role leaves play in plant physiology—this analysis is left for future work. These methods were also used to estimate measurable traits like petiole width and length, which were used as a source of validation in this work. This study demonstrated that manually estimated petiole length and width strongly correlated with their corresponding digital measurements, suggesting that the segmentation quality and feature extraction methodology accurately predicts biologically relevant features from the raw images. A strategy to consider these digital traits from a biological perspective was to estimate their broad-sense heritability (i.e., the amount of variation in the trait that is controlled by genetics), denoted by H^2. In particular, the H^2 value for vein density was 0.65, suggesting that the trait is under significant genetic control.

As a proof of concept for downstream application of the segmentation methods, a GWAS analysis was performed for the vein density trait at population scale. The top GWAS hit was Potri.017G077200, which is highly expressed in apical bud, dormant bud, and stem [56]. This gene is also expressed in immature/young leaves, suggesting that it plays a role in such tissues [56]. Comparing to *A. thaliana*, *CLP-SIMILAR PROTEIN 3* (*CLPS3*, AT3G04680) is the closest ortholog of Potri.017G077200 and is related to the human cleavage factor polyribonucleotide kinase subunit 1, which forms part of the complex responsible for polyadenylation of 3′ (3 prime) of messenger RNA [57]. CLPS3 also interacts with components of the polyadenylation complex in plants, and it is expressed throughout whole plant development, including leaves and vasculature [58]. Overexpression of *CLPS3* causes aberrant leaf phenotypes, abnormal phyllotaxis, and early flowering [58]. Futher, *CLPS3* increases the expression of *CUP-SHAPED COTYLEDON 1* (*CUC1*), an NAC transcription factor, which together with *CUC2* and *CUC3*, have been found to participate in meristem formation, organ boundary separation, and leaf shape [59–61]. Since Potri.017G077200 is an *CLPS3* ortholog, it could play similar roles in leaf development of *P. trichocarpa*, making it a strong target for genomic selection studies.

Potri.006G227300 is expressed in most plant tissues, but it is highly expressed in apical bud in spring, swelling bud, late dormant bud, as well as young and immature leaves [56]. Its *Arabidopsis* ortholog, *RHOMBOID-LIKE PROTEIN 12* (*RBL12*, AT1G18600), follows a similar expression pattern, being enriched in floral buds [62]. Very little is known about *RBL12*, but it is predicted to be an active transmembrane protease located in the mitochondria [63]. Further, *RBL12* substrates in *A. thaliana*

have not been identified; therefore, its role is yet to be determined [64]. Other genes that were associated with vein density by GWAS may play an indirect role in leaf development. For instance, the Potri.017G077300 ortholog, *PBS1-LIKE 35* (*PBL35*, AT3G01300), participates in shoot apical meristem homeostasis and plant immunity, while the Potri.006G090600 ortholog, *ALDO-KETO REDUCTASE FAMILY 4 MEMBER C11* (*AKR4C11*, AT3G53880), participates in abiotic stress tolerance through detoxification of reactive carbonyls [65–68]. Thus, Potri.017G077300 and Potri.006G090600 may play a role in leaf and vein development through such processes. Further, the Potri.006G227400 ortholog, *CELL WALL INVERTASE 5* (*CWINV5*, AT3G13784), is a cell wall invertase, and members of this family have been found to affect plant development by making hexoses available for transport [62,69].

Conclusions

Few-shot segmentation methods were extended to image-based plant phenotyping, whereby researchers can maximize predictive accuracy while minimizing the amount of training data. These methods were demonstrated for leaf scans of *P. trichocarpa*, where 50 training images were used to train an automated tracing algorithm for whole-leaf segmentation and 8 images were used to train a region growing algorithm to segment the visible vein architecture. The methods were compared against a variant of the U-Net model, which suggested that the models considered here may produce more biologically realistic segmentations (i.e., reduce fragmentation). The leaf and vein segmentations were used to extract biologically relevant morphological and topological traits related to the leaf body, venation, and petiole, which were validated with real-world manual measurements. Broad-sense clonal heritability estimates for each trait were measured, and a population-scale genomic analysis was conducted for vein density, which combined information from both leaf and vein segmentations. The GWAS analysis revealed a set of previously unconsidered SNPs and associated genes with mechanistic associations to multiple physiological processes relating to leaf development and function. Future work will include a deep dive into the relevant biology surrounding the features discussed in this work and will include the extraction of additional cryptic phenotypes relating to leaf morphology and vein topology. In particular, this work will leverage systems biology, network analysis, and climatic data to uncover the mechanistic associations within and across genotypes as they relate to sustainable bioenergy applications (e.g., biomass yield and composition).

In conclusion, this study demonstrated a complete workflow from image acquisition to phenotype extraction. The utility of these methods for biological use cases was further demonstrated by comparing biological accuracy with U-Net-based segmentations and performing GWAS that identified genomic regions and associated genes potentially controlling important plant phenotypes, such as vein density. This enhances current understanding of the genetic architecture of complex traits and may facilitate future quantitative genetics and genotype × environment interaction studies. This further allows researchers to assess how vein traits relate to other physiological processes, such as stomatal conductance, gas exchange, and overall plant productivity with important implications for developing *Populus* as a bioenergy crop. Genes detected from the quantitative genetic analysis can be used in future biotechnology experiments for

optimizing traits targeted for climate resilience, biomass production, and accelerated domestication for agriculture and biofuel production.

Data Availability

In addition to releasing all of the segmentation code on a public GitHub repository [31], we are also releasing all of the images, manual segmentations, model predictions, 68 extracted leaf phenotypes, and a new set of SNPs called against the v4 *P. trichocarpa* genome for 1,419 genotypes on the Oak Ridge National Laboratory Constellation Portal (a public DOI data server) [36]. Further, a copy of both the code and the data is available at [37]. This is, to our knowledge, one of the largest releases of plant genotype and phenotype data in a single manuscript. We hope that this work becomes a valuable community resource and helps reduce barriers commonly associated with high-throughput image-based plant phenotyping and machine learning.

Supplementary Materials

Figs. S1 and S2
Tables S1 to S3
Movies S1 and S2

References

1. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Commun ACM*. 2017;60(6):84–90.

2. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Paper presented at: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, USA.

3. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A, editors. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer; 2015. p. 234–241.

4. Taylor G, Donnison IS, Murphy-Bokern D, Morgante M, Bogeat-Triboulot MB, Bhalerao R, Hertzberg M, Polle A, Harfouche A, Alasia F, et al. Sustainable bioenergy for climate mitigation: Developing drought-tolerant trees and grasses. *Ann Bot*. 2019;124(4):513–520.

5. Grattapaglia D, Silva-Junior OB, Resende RT, Cappa EP, Müller BSF, Tan B, Isik F, Ratcliffe B, el-Kassaby YA. Quantitative genetics and genomics converge to accelerate forest tree breeding. *Front Plant Sci*. 2018;9:Article 1693.

6. Yang J, Guo X, Li Y, Marinello F, Ercisli S, Zhang Z. A survey of few-shot learning in smart agriculture: Developments, applications, and challenges. *Plant Methods*. 2022;18(1): Article 28.

7. Rutter E, Lagergren J, Flores K. Automated object tracing for biomedical image segmentation using a deep convolutional neural network. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018*, Lecture Notes in Computer Science; Cham: Springer; 2018.

8. Rutter E, Lagergren J, Flores K. A convolutional neural network method for boundary optimization enables few-shot learning for biomedical image segmentation. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019*, Lecture Notes in Computer Science; Cham: Springer; 2019.

9. Januszewski M, Kornfeld J, Li PH, Pope A, Blakely T, Lindsey L, Maitin-Shepard J, Tyka M, Denk W, Jain V. High-precision automated reconstruction of neurons with flood-filling networks. *Nat Methods*. 2018;15(8):605–610.

10. Lagergren J, Rutter E, Flores K. Region growing with convolutional neural networks for biomedical image segmentation. ArXiv. 2020. https://doi.org/10.48550/arXiv.2009.11717

11. Tuskan GA, Difazio S, Jansson S, Bohlmann J, Grigoriev I, Hellsten U, Putnam N, Ralph S, Rombauts S, Salamov A, et al. The genome of black cottonwood, populus trichocarpa (Torr. & Gray). *Science*. 2006;313(5793):1596–1604.

12. García-Lorenzo M, Sjödin A, Jansson S, Funk C. Protease gene families in populus and arabidopsis. *BMC Plant Biol*. 2006;6:Article 30.

13. Geraldes A, Pang J, Thiessen N, Cezard T, Moore R, Zhao Y, Tam A, Wang S, Friedmann M, Birol I, et al. Snp discovery in black cottonwood (populus trichocarpa) by population transcriptome resequencing. *Mol Ecol Resour*. 2011; 11 Suppl 1:81–92.

14. Zhang J, Yang Y, Zheng K, Xie M, Feng K, Jawdy SS, Gunter LE, Ranjan P, Singan VR, Engle N, et al. Genome-wide association studies and expression-based quantitative trait loci analyses reveal roles of hct 2 in caffeoylquinic acid biosynthesis and its regulation by defense-responsive transcription factors in populus. *New Phytol*. 2018;220(2):502–516.

15. Slavov GT, DiFazio SP, Martin J, Schackwitz W, Muchero W, Rodgers-Melnick E, Lipphardt MF, Pennacchio CP, Hellsten U, Pennacchio LA, et al. Genome resequencing reveals multiscale geographic structure and extensive linkage disequilibrium in the forest tree populus trichocarpa. *New Phytol*. 2012;196(3):713–725.

16. Evans LM, Slavov GT, Rodgers-Melnick E, Martin J, Ranjan P, Muchero W, Brunner AM, Schackwitz W, Gunter L, Chen JG, et al. Population genomics of populus trichocarpa identifies signatures of selection and adaptive trait associations. *Nat Genet*. 2014;46(10):1089–1096.

17. Chhetri HB, Macaya-Sanz D, Kainer D, Biswal AK, Evans LM, Chen JG, Collins C, Hunt K, Mohanty SS, Rosenstiel T, et al. Multitrait genome-wide association analysis of populus trichocarpa identifies key polymorphisms controlling morphological and physiological traits. *New Phytol*. 2019;223(1):293–309.

18. Chhetri HB, Furches A, Macaya-Sanz D, Walker AR, Kainer D, Jones P, Harman-Ware AE, Tschaplinski TJ, Jacobson D, Tuskan GA, et al. Genome-wide association study of wood anatomical and morphological traits in populus trichocarpa. *Front Plant Sci*. 2020;11:Article 545748.

19. Jansson S, Douglas CJ. Populus: A model system for plant biology. *Annu Rev Plant Biol*. 2007;58:435–458.

20. Rubin EM. Genomics of cellulosic biofuels. *Nature*. 2008;454(7206):841–845.

21. McKown AD, Klápště J, Guy RD, Geraldes A, Porth I, Hannemann J, Friedmann M, Muchero W, Tuskan GA, Ehlting J, et al. Genome-wide association implicates numerous genes underlying ecological trait variation in natural populations of populus trichocarpa. *New Phytol*. 2014;203(2):535–553.

22. Sack L, Scoffoni C. Leaf venation: Structure, function, development, evolution, ecology and applications in the past, present and future. *New Phytol*. 2013;198(4): 983–1000.

23. Bühler J, Rishmawi L, Pflugfelder D, Huber G, Scharr H, Hülskamp M, Koornneef M, Schurr U, Jahnke S. Phenovein–a tool for leaf vein segmentation and analysis. *Plant Physiol*. 2015;169(4):2359–2370.

24. Xu H, Blonder B, Jodra M, Malhi Y, Fricker M. Automated and accurate segmentation of leaf venation networks via deep learning. *New Phytol*. 2021;229(1):631–648.

25. Katyal V, Aviral. Leaf vein segmentation using odd gabor filters and morphological operations. ArXiv. 2012. https://doi.org/10.48550/arXiv.1206.5157

26. Larese MG, Craviotto RM, Arango MR, Gallo C, Granitto PM. Iberoamerican Congress on Pattern Recognition; Springer; 2012.

27. Salima A, Herdiyeni Y, Douady S. Paper presented at: International Conference on Advanced Computer Science and Information Systems (ICACSIS); 2015 Oct 10–11; Depok, Indonesia.

28. Zhu J, Yao J, Yu Q, He W, Xu C, Qin G, Zhu Q, Fan D, Zhu H. A fast and automatic method for leaf vein network extraction and vein density measurement based on object-oriented classification. *Front Plant Sci*. 2020;11:Article 499.

29. Jiang Y, Li C. Convolutional neural networks for image-based high-throughput plant phenotyping: A review. *Plant Phenomics*. 2020;2020:Article 4152816.

30. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. Pytorch: An imperative style, high-performance deep learning library. *Adv Neural Inf Proces Syst*. 2019;32:Article 721.

31. Lagergren J, Pavicic M, Chhetri HB, Larry M, Hyatt D, Kainer D, Rutter EM, Flores K, Bailey-Bale J, Klein M, et al. *Few-shot leaf segmentation*, Tech. rep., Oak Ridge National Lab. (ORNL); United States; 2023.

32. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, et al. Fiji: An open-source platform for biological-image analysis. *Nat Methods*. 2012;9(7):676.

33. Seethepalli A, York L. Rhizovision explorer-interactive software for generalized root image analysis designed for everyone (version 2.0.2). Zenodo. 2020 Oct 16.

34. Seethepalli A, Dhakal K, Griffiths M, Guo H, Freschet GT, York LM. Rhizovision explorer: Open-source software for root image analysis and measurement standardization. *AoB plants*. 2021;13(6):plab056.

35. Wang J, Zhang Z. Gapit version 3: Boosting power and accuracy for genomic association and prediction. *Genom Proteom Bioinform*. 2021;19(4):629–640.

36. Lagergren J, Pavicic M, Chhetri HB, Larry M, Hyatt D, Kainer D, Rutter EM, Flores K, Bailey-Bale J, Klein M, et al. Supporting information for "Few-shot learning enables population-scale analysis of leaf traits in Populus trichocarpa", Tech. rep. Oak Ridge National Lab. (ORNL), Oak Ridge, USA; 2023. https://doi.org/10.13139/ORNLNCCS/1908723

37. Lagergren J, Pavicic M, Chhetri HB, Larry M, Hyatt D, Kainer D, Rutter EM, Flores K, Bailey-Bale J, Klein M, et al. Supporting information for "Few-shot learning enables population-scale analysis of leaf traits in Populus trichocarpa", Tech. rep. Oak Ridge National Lab. (ORNL), Oak Ridge, USA; 2023. https://doi.org/10.5281/zenodo.7908197

38. Bailey-Bale J, Freer-Smith P, Suvocarev K, Chen J, Muchero W, DiFazio S, Tuskan G, Tschaplinski T, Taylor G. Genomic sciences program (GSP). Annual PI Meeting. 2021 Feb 22.

39. Perfection v19/v39 user's guide.

40. Swiss precision instruments.

41. The GIMP. Development team. Gimp.

42. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ArXiv. 2014. https://doi.org/10.48550/arXiv.1409.1556

43. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Paper presented at: International Conference on Machine Learning (PMLR); 2015 Jul 7–9; Lille, France.

44. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. Paper presented at: Proceedings of the 30th International Conference on Machine Learning; 2013 June 16–21; Georgia, USA.

45. Kingma DP, Ba J. Adam: A method for stochastic optimization. ArXiv. 2014. https://doi.org/10.48550/arXiv.1412.6980

46. Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. Paper presented at: Proceedings of the

IEEE International Conference on Computer Vision; 2017 Jun 20–23; Cambridge, USA.

47. Nychka D, Furrer R, Paige J, Sain S, fields: Tools for spatial data. 2021.

48. Bates D, Mächler M, Bolker B, Walker S. Fitting linear mixed-effects models using lme4. *J Stat Softw*. 2015;67(1):1–48.

49. Nordberg H, Cantor M, Dusheyko S, Hua S, Poliakov A, Shabalov I, Smirnova T, Grigoriev IV, Dubchak I. The genome portal of the department of energy joint genome institute: 2014 updates. *Nucleic Acids Res*. 2014;42:D26–D31.

50. Li H. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. ArXiv. 2013. https://doi.org/10.48550/arXiv.1303.3997

51. Van der Auwera GA et al. From fastq data to high-confidence variant calls: The genome analysis toolkit best practices pipeline. *Curr Protoc Bioinformatics*. 2013;43(1110):11.10.1–11.10.33.

52. Huang M, Liu X, Zhou Y, Summers RM, Zhang Z. Blink: A package for the next level of genome-wide association studies with both individuals and markers in the millions. *Gigascience*. 2019;8:giy154.

53. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. Paper presented at: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy.

54. Ciresan DC, Giusti A, Gambardella LM, Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. Paper presented at: Proceedings of the 25th International Conference on Neural Information Processing Systems; 2012 Dec 3–6; NY, USA.

55. Smith AG, Han E, Petersen J, Olsen NAF, Giese C, Athmann M, Dresbøll DB, Thorup-Kristensen K. Rootpainter: Deep learning segmentation of biological images with corrective annotation. *New Phytol*. 2022;236(2):774–791.

56. Sreedasyam A, Plott C, Hossain MS, Lovell JT, Grimwood J, Carlson J, Shu S, Phillips J, Amirebrahimi M, Zane M, et al. Jgi plant gene atlas: An updateable transcriptome resource to improve structural annotations and functional gene descriptions across the plant kingdom. bioRxiv. 2022. https://www.biorxiv.org/content/10.1101/2022.09.30.510380v1

57. de Vries H, Rüegsegger U, Hübner W, Friedlein A, Langen H, Keller W. Human pre-mrna cleavage factor iim contains homologs of yeast proteins and bridges two other cleavage factors. *EMBO J*. 2000;19(21):5895–5904.

58. Xing D, Zhao H, Li QQ. Arabidopsis clp1-similar protein3, an ortholog of human polyadenylation factor clp1, functions in gametophyte, embryo, and postembryonic development. *Plant Physiol*. 2008;148(4):2059–2069.

59. Ki H, Karim MR, Takada S, Ki T, Furutani M, Aida M, Tasaka M. Arabidopsis cup-shaped cotyledon3 regulates postembryonic shoot meristem and organ boundary formation. *Plant Cell*. 2006;18(11):2946–2957.

60. Spinelli SV, Martin AP, Viola IL, Gonzalez DH, Palatnik JF. A mechanistic link between stm and cuc1 during arabidopsis development. *Plant Physiol*. 2011;156(4):1894–1904.

61. Nikovics K, Blein T, Peaucelle A, Ishida T, Morin H, Aida M, Laufs P. The balance between the mir164a and cuc2 genes controls leaf margin serration in arabidopsis. *Plant Cell*. 2006;18(11):2929–2945.

62. Klepikova AV, Kasianov AS, Gerasimov ES, Logacheva MD, Penin AA. A high resolution map of the arabidopsis thaliana developmental transcriptome based on rna-seq profiling. *Plant J*. 2016;88(6):1058–1070.

63. Lemberg MK, Freeman M. Functional and evolutionary implications of enhanced genomic analysis of rhomboid intramembrane proteases. *Genome Res*. 2007;17(11):1634–1646.

64. Kmiec-Wisniewska B, Krumpe K, Urantowka A, Sakamoto W, Pratje E, Janska H. Plant mitochondrial rhomboid, atrbl12, has different substrate specificity from its yeast counterpart. *Plant Mol Biol*. 2008;68(1-2):159–171.

65. Simpson PJ, Tantitadapitak C, Reed AM, Mather OC, Bunce CM, White SA, Ride JP. Characterization of two novel aldo–keto reductases from arabidopsis: Expression patterns, broad substrate specificity, and an open active-site structure suggest a role in toxicant metabolism following stress. *J Mol Biol*. 2009;392(2):465–480.

66. Sengupta D, Naik D, Reddy AR. Plant aldo-keto reductases (akrs) as multi-tasking soldiers involved in diverse plant metabolic processes and stress defense: A structure-function update. *J Plant Physiol*. 2015;179:40–55.

67. Luo X, Wu W, Liang Y, Xu N, Wang Z, Zou H, Liu J. Tyrosine phosphorylation of the lectin receptor-like kinase lore regulates plant immunity. *EMBO J*. 2020;39: Article e102856.

68. Wang W, Hu C, Li X, Zhu Y, Tao L, Cui Y, Deng D, Fan X, Zhang H, Li J, et al. Receptor-like cytoplasmic kinases pbl34/35/36 are required for cle peptide-mediated signaling to maintain shoot apical meristem and root apical meristem homeostasis in arabidopsis. *Plant Cell*. 2022;34(4): 1289–1307.

69. Sherson SM, Alford HL, Forbes SM, Wallace G, Smith SM. Roles of cell-wall invertases and monosaccharide transporters in the growth and development of arabidopsis. *J Exp Bot*. 2003;54(382):525–531.