

## Supporting Information

# A high-throughput imaging system to quantitatively analyze the growth dynamics of plant seedlings

Yongfan Men<sup>¶1,2</sup>, Qiang Yu<sup>¶1,3</sup>, Zitian Chen<sup>1,4</sup>, Jianbin Wang,<sup>2,4</sup>  
Yanyi Huang<sup>\*1,2,4</sup>, and Hongwei Guo<sup>\*1,3</sup>

<sup>1</sup> Biodynamic Optical Imaging Center, College of Life Sciences, <sup>2</sup> College of Engineering, <sup>3</sup> State Key Laboratory of Protein and Plant Gene Research, College of Life Sciences, and <sup>4</sup> College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China

¶ These authors made the equal contribution to this work.

\*Correspondence should be addressed to H. G. (hongweig@pku.edu.cn) or Y. H. (yanyi@pku.edu.cn).

## 1. Supporting Figure



**Fig. S1** Three horizontal (TSM25-1A, Zolix Instrument, Beijing, China), rotational (RSM82-1A, Zolix Instrument, Beijing, China), and vertical (TSMV5-1A, Zolix Instrument, Beijing, China) stages that hold the sample chambers are shown in the figure as (a), (b), and (c), respectively. The stages are constructed vertically, and fixed on the automated translational stage by aluminum frame.

## 2. Supporting Movies

**Movie S1.** The calculation process illustrated by the first approach described in the paper. In the movie, the algorithm is able to proceed even when the cotyledon is touching the hypocotyl. The seedling is divided into geometrically different regions. In each region, different rules are applied to determine the middle points. The whole middle line determination is divided into 4 stages. In the first

stage, the middle points in the hypocotyl below the red line are determined with a large step size. The yellow line shows the calculation direction. The grey circle highlights the middle point centering but is not used in the calculation. In the second stage, the step size is decreased. This operation ensures the calculation can run smoothly into the third stage. In the third stage, the step size is further decreased, to accommodate the high curvature of the hook. In the last stage, the points in the second stage are amended, and the middle line calculation is finished.

**Movie S2.** The tracking of the seedling terminus when cotyledon is growing and changing its orientation. The red box is the patch that is used to track the endpoint. The blue and green circles show the starting and ending point of the middle line, respectively. The red line shows the whole middle line. The length of the middle line is shown in each frame.

**Movie 3.** The calculation of the root of the seedling. The blue and green circles show the starting and ending point of the middle line, respectively. The red line shows the whole middle line. The length of the middle line is shown in each frame. The terminus of the middle line is determined from the intersection between the middle line and the edge of the root tip.

### 3. Supporting Algorithms.

Note: The source code is highly integrated with the hardware system, thus cannot be simply proceeded on other machines without further modifications. Here we present our key algorithms using pseudo code with explanations below. The readers are welcome to contact with the authors for comments or questions in detail.

Certain scripts, including the image processing procedures, are omitted as they are neither essential for applying the idea nor mentioned in the paper.

```
% Part 1:
% Pseudocode: central-line calculation.
% -----
% basic data structure is a "point" ('p' in scripts below):
% point.c=[x,y] - the x,y coordinate of the point
% point.v - the tangent direction of the point
% point.rl and point.rr - left and right radius length of the point
% point.vl and point.vr - left and right radius direction of the point
% -----
% 'img', a hypocotyl binary picture, is required for applying this scripts.

% define parameters
pace_min=0.1;
pace_max=0.7;
times_try_max=10;
radius_tolerance=1/100;
```

```
% scripts starts
pace=(pace_min+pace_max)/2;
i=1;
terminatebox=centralline_premeasurement(img);
% find termination box (locate around the cotyledon)
p(1)=centralline_initialize(img,ybottom);
% initialize the first point of hypocotyl
is_terminal=0;
while ~is_terminal
    times_try=0;
    i=i+1;
while ~is_centroid
    times_try=times_try+1;
if times_try>times_try_max;
    is_terminal=1;
% warnings, loggings and break.
break;
end
    p(i).c = NextPoint(p(i-1).c, p(i-1).v, p(i-1).r*pace);
% predict the next point position, (aligning?)the current point's
% tangent and moving pace is determined by last point's radius and
% pace factor.
    [p(i).v,p(i).rl,p(i).rr,p(i).vl,p(i).vr]=point_radius(img,
p(i));
    is_centroid = abs(p(i).rl-p(i).rr)/(p(i).rl+p(i).rr) <
radius_tolerance;
% ---- here, an error detection function is omitted ---- %
if is_centroid
% if current point fits the centroid good.
    pace = (pace_max+pace)/2;
% increase the pace factor for faster calculation.
    p(i).r = (p(i).rl+p(i).rr)/2;
else
% else, do the "centroid mending algorithm" mentioned in paper.
    A=p(i).rl;
    B=p(i).rr;
    Theta=ReformDirection(p(i).vr-p(i).vl);
% ^- the three quantitative values needed for "centroid mending algorithm"
    T=A/cos(Theta);
    X=T*(B-A)/(A+T);
% calculation of the mending offset along 'vr' or right vector
    ptemp=NextPoint(p(i).c,p(i).vr,X);
% an approximate mended point.
```

```
p(i-1).v=AbsDirection(GetDirection(p(i-1).c,ptemp),p(i-1).v);
% Changing the current point's tangent.
    pace=(pace_min+pace)/2;
% pace decreased for better precision.
end
end
% repeat until current point fits centroid good. usually takes
% 1-3 times for trying.
% -----Some error detection code is omitted ----- %
% ---Detected errors may also cause termination-----%
    is_terminal = is_terminal || is_locate_in(p(i).c,terminatebox);
end

% Part 2:
% Pseudocode: terminus determination
% a centralline is a sequence of central point ( data structure point or
'p' )
% a centrallinelist is a sequence of centralline
% terminus determination can be applied after a correct centralline list
% is achieved.
% The sequence of binary images is assigned to img{n}

index_terminus(1)=terminal_point_initialize(centrallinelist{1});
% In current method, the point has most curvature is detected.

for i=1:numel(centrallinelist)-1
% generate target image around the index_terminus of current centralline
% generate a list of positions to scan. Scanlist is concentrated around
% the next centralline's terminus which is predicted from information
% provided by current centralline.

tar=generate_target(img(i),centrallinelist{i}(index_terminus(i)));
    scanlist=generate_scanlist(centrallinelist{i},index_terminus(i));

% caculating the Compare matrix, by the method
    C_rough=CompareMatrix(tar,img(i+1),scanlist);
    [~,min_index]=min(C_rough);
    min_scan=scanlist(min_index);

% refine the matching result:
    scanlist=generate_scanlist_refine(min_scan);
    C_fine=CompareMatrix(tar,img(i+1),scanlist);
```

```
[~,min_index]=min(C_fine); % 2D minimum value
min_scan=scanlist(min_index);
% ~90% time cost is saved by doing rough matching before refining.

% coordinate [min_x,min_y] may not fit with

index_terminus(i+1)=find_nearest_point(centrallinelist(i+1),min_scan)
;
end

% 'scan' is consist (consistent?) with a [x,y]. Cell 'scanlist' is consist
by a group of 'scan'.
function c=CompareMatrix(tar,field,scanlist)
[y,x]=size(tar);
for i=1:numel(scanlist)
    scan=scanlist{i};
    C=xor(tar,field(scan(2):scan(2)+y-1,scan(1):scan(1)+x-1));
    c(i)=sum(sum(C));
end
```