

PICBREEDER

ANALYZE IMAGE DNA

APPLET VERSION 1.0
USER INSTRUCTIONS

Jeremiah T. Folsom-Kovarik

INTRODUCTION

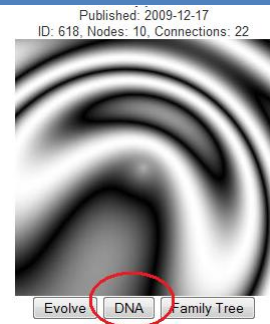
Picbreeder is a collaborative image evolution website that showcases NEAT, Neuroevolution of Augmenting Topologies. NEAT and picbreeder are projects of Dr. Kenneth Stanley and students in his research group EPLex at the University of Central Florida.

In picbreeder, users interactively evolve Compositional Pattern Producing Networks (CPPNs). CPPNs are similar to neural networks that are especially suited to output, among other things, original pictures. The CPPN of an image is something like the DNA of a living thing in the real world. The Analyze Image DNA applet lets users view the DNA, or networks, that created their favorite pictures on the website. Users can also manipulate the networks to see how each change affects the pictures they produce, and learn more about how CPPNs work.

The applet was created by Jeremiah T. Folsom-Kovarik, a graduate student working under Dr. Stanley.

STARTING THE ANALYSIS TOOL

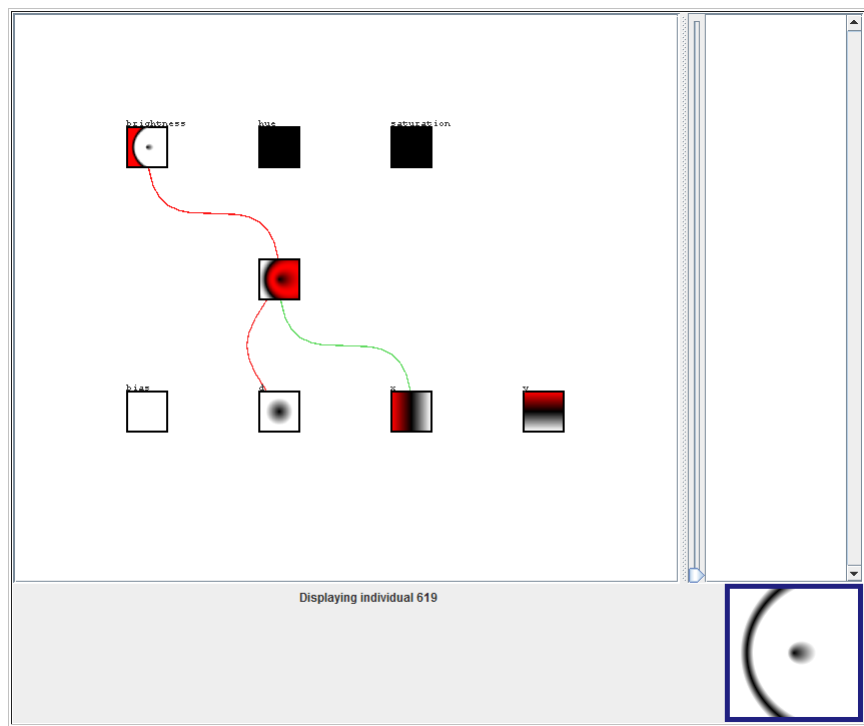
To access the analysis tool, click the “DNA” button below any picture on the picbreeder website. The tool will start and load the network for that picture.



UNDERSTANDING THE NETWORK

Here is a picture of a simple picbreeder image viewed in the analysis tool. The main part of the tool, the pane at the top left, is a graphical representation of the network underlying the image. At the bottom right is the output of the network, that is, the picture itself.

Picbreeder networks are made up of nodes and links between nodes. In the analysis viewer, nodes are shown as squares with pictures on them and links are shown as colored lines connecting the nodes. Each node in a network applies a simple function to process some data, and then the links pass data



from the output of one node to the input of another. The data that goes into the network is the location of an image pixel. The network applies all its internal functions to that data, and finally outputs what color to make the pixel at that location. Then the network starts over with another pixel, until it has colored every pixel in the output image. With the image analysis tool, users can trace how all the simple nodes in the network work together to create a complex output.

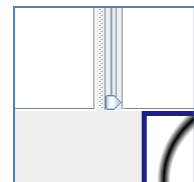
Let's look at the network being analyzed in the example. In the analysis tool, the inputs of the network are always shown at the bottom, and the outputs are at or near the top. The inputs are four nodes labeled "bias," "d," "x," and "y." The small pictures on those four nodes show what their values will be as they pass over different pixels. The four input nodes have the same functions in every network:

- Bias — the network needs an input that is always the same. Traditionally this is called its bias, and the bias node looks like a blank white square because it always outputs 1, the highest value, at every pixel.
- D — this node tells the network how far each pixel is from the center. This distance can run from zero when a pixel is close to the center, up to 1 when the pixel is far from the center. By outputting the distance, the d node produces circles of pixels that are equal distances from the center of the image.
- X — this node tells the network the x-coordinate of each pixel. Its picture shows vertical lines where all the pixels that have the same location on the x-axis are colored alike. Pixels on the right have a value of 1, shading to 0 in the center of the image, and then to -1 on the left. Negative values appear in red.
- Y — this node tells the network the y-coordinate of each pixel. Its picture shows horizontal lines where all the pixels that have the same location on the y-axis are colored alike. Pixels on the bottom have a value of 1, shading to 0 in the center of the image, and then to -1 on the top. Negative values appear in red.

Connected to the tops, or outputs, of these four nodes are green and red lines. Each line in turn connects to the bottom, or input, of another node in the network. The lines represent links that carry data from one node's output to another node's input.

When they transmit data between nodes, links also multiply the data by some constant real number. This multiplier is called the link's weight. A link's weight changes how much the data from its source will affect the destination node — links with weights nearer to zero will have weaker or more subtle effects at the destination. In the analysis tool line colors indicate link weights, with positive weights drawn in green and negative weights in red. The strongest links in the network have bright colors, while the weak links with weights close to zero are almost gray.

Networks can have many links, so a control exists to hide the links that impact the network less. Use the slider at the right edge of the network view to hide weaker connections and make the stronger connections easier to pick out. The hidden connections still affect the network.



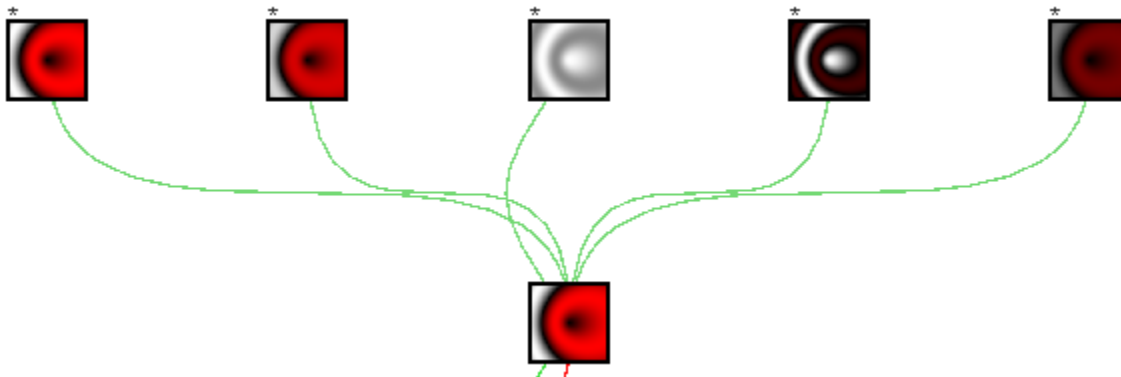
Now let's look at the nodes in the rows above the network input nodes. Nodes that accept incoming links, process the data, and pass it on are called hidden nodes. Hidden nodes are still visible in the viewer — the term simply refers to nodes that are neither network inputs nor network outputs. In the example network, one hidden node appears in the middle of the network with no label. Its inputs combine data from the d and x network input nodes, and its output feeds the network output node that controls brightness.

In addition to accepting incoming links, hidden nodes are also unlike input nodes in that each one applies a simple mathematical function to its inputs. The value of the function on the input determines what the node outputs, which creates a pattern drawn inside the square that represents it in network view. Like the input nodes, these patterns are represented with white for positive values, red for negative values, and black near zero.

To determine a hidden node's value when the network is looking at a particular pixel, the node first adds together all its inputs from elsewhere in the network. Then, it applies one of the following functions:

- Identity — pass on the sum of the node's inputs as its output.
- Sine — take the sine of the inputs. This function can transform a steadily increasing pattern in the input into a repeating pattern in the node's output.
- Cosine — take the cosine of the inputs. This function can also create a repeating pattern.
- Gaussian — apply a zero-centered Gaussian function to the inputs. This function makes inputs near zero become near one in the node's output, while inputs with high absolute values become near zero.
- Sigmoid — apply a sigmoid or logistic function to the inputs. This function "squashes" inputs with extreme values so that the node's output falls between one and minus one. It changes large values by a lot without making much change in values that are near zero.

The effect of applying each function listed above is shown in the figure below. Each of the five nodes marked with an asterisk applies one of the five functions to the data coming from the node at the bottom of the figure, in the same order as the list above. All the links have a weight of one, so they do not change the data flow.



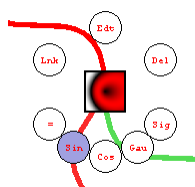
Finally, three network output nodes identified by labels are above the hidden node in the example. Output nodes in picbreeder networks apply functions to their inputs like hidden nodes do, but they are special because their outputs turn the network's calculations into images. They play the following roles in determining each pixel's color:

- Hue — after processing the network, the value at this node determines the hue of the pixel.
- Saturation — the value at this node determines the saturation of the pixel.
- Brightness — the value at this node determines the brightness of the pixel. This node can sometimes appear in the middle of a network, rather than the top row. Because of a design decision, picbreeder networks sometimes have links from the brightness node to other hidden or output nodes, but hue and saturation nodes never have outgoing links.

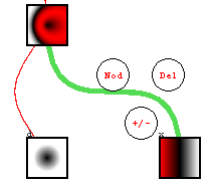
In grayscale images like the one in the example, only the brightness node has an effect and the other two outputs are ignored — they do not take data from anywhere or do any processing. However, notice that even though the brightness node in the example is the only thing that affects the image, that node's output as shown in the network view appears to be different from the network output as shown in the image preview at the bottom right. The brightness node has black, white, and red sections but the preview is only black and white. This is because the red that appears throughout the network view indicates areas where a node's output is negative. After network processing finishes, a post-processing step takes the absolute value of negative outputs and caps all values at a maximum of one.

CHANGING THE NETWORK

The analysis tool lets users make all the same changes to networks as evolution can make. Every change made in the network pane immediately affects the rest of the network and the preview image. This functionality can help users learn about how simple network components work together to create a complex image.



Two context menus control direct changes to the network. Click on any node to bring up the menu associated with that node (left), and click on a link to activate the menu for changing that link (right). The menus do use some confusing mnemonics, but the abbreviations should become clear with the explanation in this section.



First, the node context menu contains the following options:

- **Lnk** — link this node to another node. Clicking this button will cause a button labeled “Tgt” (target) to appear over every node in the network that is eligible to be the destination of a link from the selected node. Some nodes are not eligible destinations because linking to them would create a loop in the network. Also, the “Lnk” option will not appear for the hue and saturation nodes because they are not allowed to have any outgoing links. This action will cause the network to be redrawn, possibly with nodes appearing in different places on the screen.
- **Edt** — edit this node in the interactive node editing pane. See the next section of this document for instructions on using the interactive editing mode. Note that selecting a node for interactive editing can also be a good way to view the pattern it produces close-up.
- **Del** — delete this node from the network immediately. All the links that connect to it or from it will also be deleted. It is not possible to delete input or output nodes.
- **=, Sin, Cos, Gau, Sig** — change the function this node applies to its inputs. The function the node currently applies is highlighted.

Clicking a node also highlights all of its links. This can be useful because when links are highlighted, they may be easier to click. Clicking a link activates the link context menu, which contains the following options:

- **Nod** — create a new node in the network. The selected link will be replaced by a link from its source into the new node and a second link from the new node to the old link’s destination. The two new links will be created with random weights. The new node will be labeled with an asterisk. This action will cause the network to be redrawn, possibly with nodes appearing in different places on the screen.
- **Del** — delete the selected link immediately. If this action leaves a node with no inputs, that node will still exist in the network but will not be updated until a new link gives it some inputs.
- **+/-** — change the weight of the selected link. A text box will appear displaying the link’s current weight. Typing in this box and pressing enter will change the link’s weight. Any numeric value is legal, but evolution only produces links with weights between 3 and -3, so extreme values may have unexpected effects.

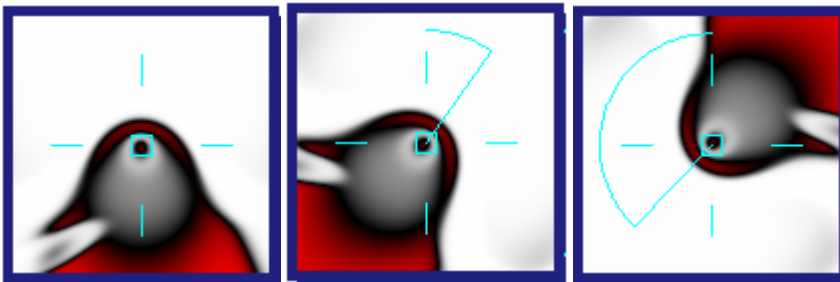
Together, these actions represent all the network changes evolution uses to create the images in the picbreeder library. Try combining these actions to recreate a specific image — it’s very difficult!

INTERACTIVE NODE EDITING

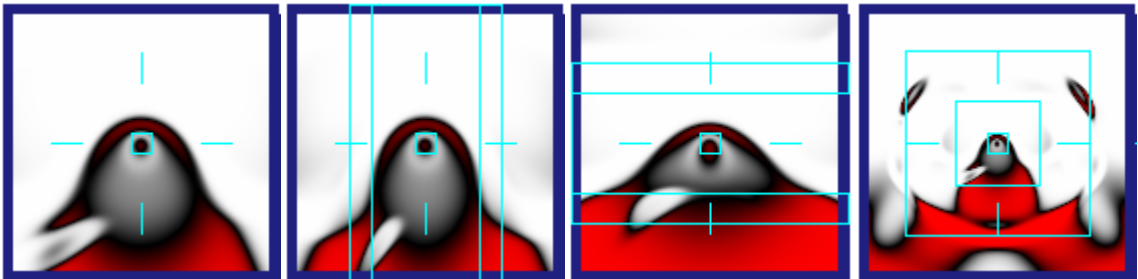
Clicking on a node to activate its context menu and choosing “Edt” will display that node in the right-hand pane of the analysis tool. Here, the node can be examined more closely or edited interactively by applying affine transformations to the value at the node. The changes made in this pane translate into combinations of the link weight change command on the selected node’s inputs, and take effect on the network immediately.

Moving the mouse over a node in the interactive editing pane will display a control that looks like a crosshair. Clicking and dragging on the arms of this control changes the value of the node when the mouse is released. The following changes are available in this pane:

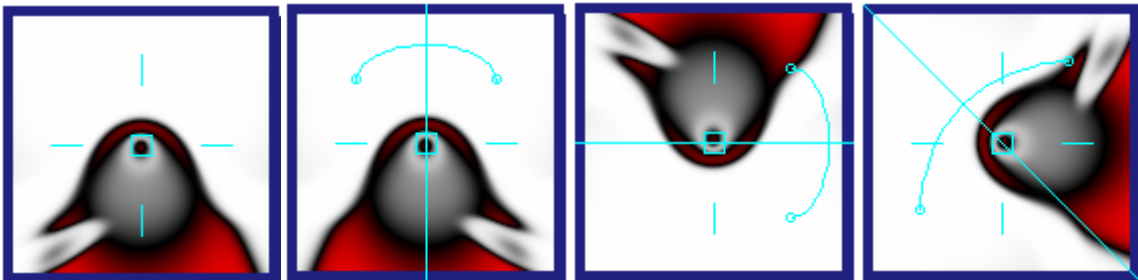
- Rotate — clicking any arm of the crosshair and dragging it in a circular motion, like turning a wheel, will rotate the node.



- Stretch — clicking a horizontal arm and dragging it left or right will stretch or shrink the node in the horizontal direction only. Likewise, dragging a vertical arm up or down will stretch or shrink the node in the vertical direction. Clicking a corner of the image and dragging toward or away from the center will shrink or stretch the node an equal amount in both dimensions.



- Reflect — clicking on any corner of the image and dragging across the crosshair to the other side will reflect the image horizontally, vertically, or diagonally as appropriate. To reflect diagonally, drag from one corner of the image to the diagonally opposite corner.



Changes made in the indirect editing pane are carried out through combinations of link weight change commands in the network, and some nodes do not have links that can be changed by this method. In such cases, it is also not possible to add the required links without changing the nodes' values. For the purpose of editing nodes interactively, indirect links to the required inputs are sufficient. The available changes are as follows:

- Only nodes that take some input from either the x or the y node can be rotated or reflected.
- Nodes that take input from the d node but neither the x nor y nodes can still be stretched, but only in both dimensions at once, not only horizontally or only vertically. Furthermore, nodes that take input from all three of x, y, and d will become distorted when stretching only horizontally or only vertically, because of the d input. Stretching in both directions equally will work correctly.
- Nodes that have no input from x, y, or d can still be displayed in the interactive editing pane, but cannot be changed there.

It can be interesting to change the value of one node and see how the change affects the rest of the network. If the edited node is near the network inputs, one change can affect a lot of other nodes. If it is on separate a network path from some other nodes, the change can affect how the different nodes interact. Sometimes a node with an interesting pattern turns out to be only weakly connected to the rest of the network, so changes don't make much difference at all!

KNOWN LIMITATIONS

In version 1.0 of this tool, the following limitations are known to exist:

- The order of nodes in the network view sometimes does not make sense. Nodes are sometimes drawn in places that make the links more tangled than they need to be.
- Changes made in the interactive node editing view affect all nodes that share inputs with the node being edited. It would be possible to create copies of the node's inputs before editing and change them instead, but this would increase network sizes by up to double each time the user made a change.
- Interactive node editing does not support the affine transformations skew and translation.
- There is no facility to zoom in and out.
- There is no facility to undo changes to the network.
- There is no facility to save or publish networks after changing them.

If an error occurs while using the analysis tool, please use the contact page on the picbreeder website to reach the development team.