# Processing
### Follow this workflow to get more out of your deep-sky astrophotos.
## with PixInsight

Ron Brecher

**"One size fits all"** is a rarely used term in astrophotography. Astro-imagers shoot all types of deep-sky objects in a variety of ways: wide field or high magnification, color or monochrome, natural or representative-color narrowband. We often use different telescopes, cameras, software, and processing techniques to achieve beautiful results. And although I've used a variety of equipment over the years, I've settled on one program to work up all of my deep-sky images from start to finish: *PixInsight* (**pixinsight.com**).

*PixInsight* is perhaps the only software that can handle everything from basic image calibration to advanced processing. It reads and writes virtually every image file format and has effective tools for just about anything you'd want to make your images stand out from the crowd. But it is a complex program that can be intimidating to a beginner.

I found that adopting a standard workflow helped me to become familiar with this powerful software and to streamline my own processing routine to get the best results from those rare clear nights. Here's how I do it.

## Processing Flow

To begin, there are several ways to access *PixInsight*'s processes and scripts. The easiest is using the pull-down menu along the top of the screen. You can find all processes under Process > All Processes. To its right, the Script menu contains a standard set of scripts, organized in groups (Utilities, Image Analysis, Batch Processing, and more). You can download other scripts, or record your own to further streamline your own workflow.

My typical processing routine involves 5 main steps:

1. Reducing my data to a single master for each filter.
2. Linear processing of the color and luminance (unfiltered monochrome) stacked separately. This includes combining the red, green, and blue files into a color image and enhancing it with Hydrogen-alpha (Hα) data, if shot.
3. Nonlinear stretching.
4. Combining the luminance and color data.
5. Final tweaks to the photo to prepare it for publishing or printing.

Following this general outline has helped raise the quality of the images I produce and also keeps me from skipping important processes. With a few minor modifications, this workflow can be used for image files generated by any type of camera, including DSLR and one-shot color CCD images.

## Data Reduction

A night of CCD imaging can generate dozens of files, not to mention many calibration frames — the bias frames, dark frames, and flat-field images used to clean up the "light" (target) exposures. Data reduction is the process of making single "master" images from the many raw files I record of my chosen target. It involves selecting the best frames and then calibrating, aligning, and combining them into a final (though still unprocessed) result. Because I use a monochrome camera with color and narrowband filters, I often shoot through 3 to 5 filters to produce a color image (red, green, and blue, and sometimes luminance and Hα).

The first tool in my image-processing workflow is the Blink process, which lets me inspect each image and discard any that appear to be of poor quality (images with clouds, poor tracking, and so on). *PixInsight*'s SubframeSelector script (Script > Batch Processing > SubframeSelector) can automatically evaluate images and sort them into "approved" or "rejected" groups using quality criteria you can modify. When evaluating image quality, I use this opportunity to identify the best one to use as a reference when aligning images.

With my best frames in hand, I then apply the Batch-Preprocessing (BPP) script. BPP is how I gather my calibration frames and generate "masters" from each group, and then apply these to my images. This script also performs additional functions, such as debayering one-
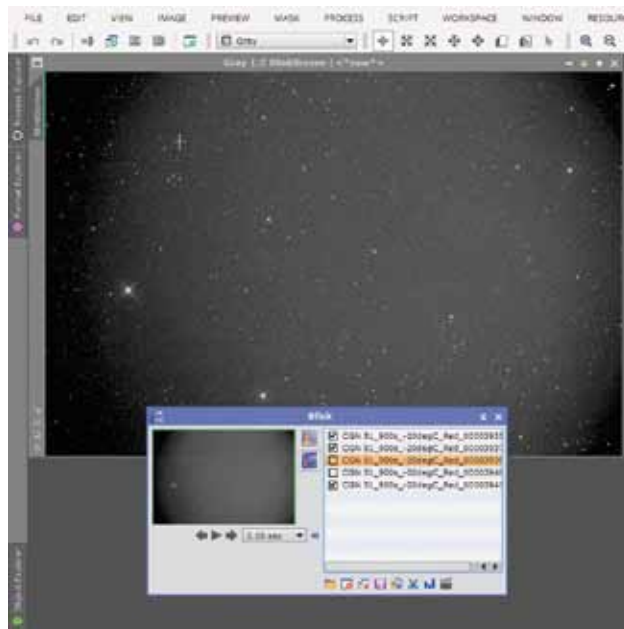


**QUALITY CONTROL** The Blink process lets you quickly inspect your images and reject any with clouds or poor tracking.

shot color images or correcting hot and cold pixels with the CosmeticCorrection processes (both found in the Light tab). Once everything is loaded and ready, I choose where to save my calibrated files in the Output Directory section and click the Run button in the bottom right.

When BPP is complete, I'll open all the calibrated images taken with all my filters and align them all using the ImageRegistration > StarAlignment process. Be sure to select the reference image you identified earlier as your alignment target.

Now it's time to combine all my sub-exposures using the ImageIntegration process (found at Process > ImageIntegration > ImageIntegration), selecting one single-filter group at a time. *PixInsight* includes four integration processes, but ImageIntegration is the workhorse. I choose the best combination method for each filter's
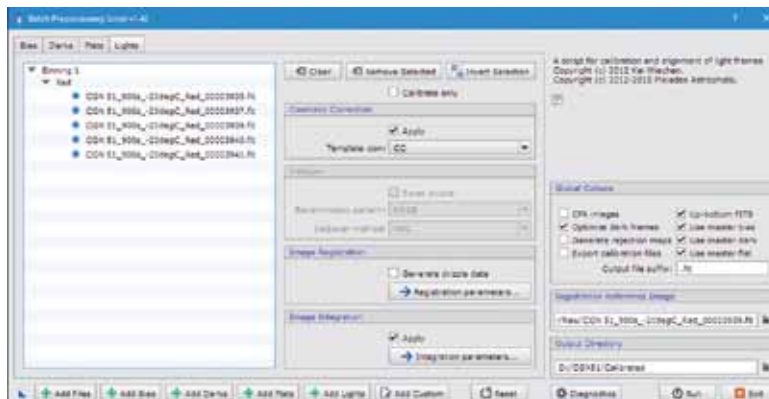


**DATA REDUCTION** Image calibration can be automated in *PixInsight* using the BatchPreprocessing script.
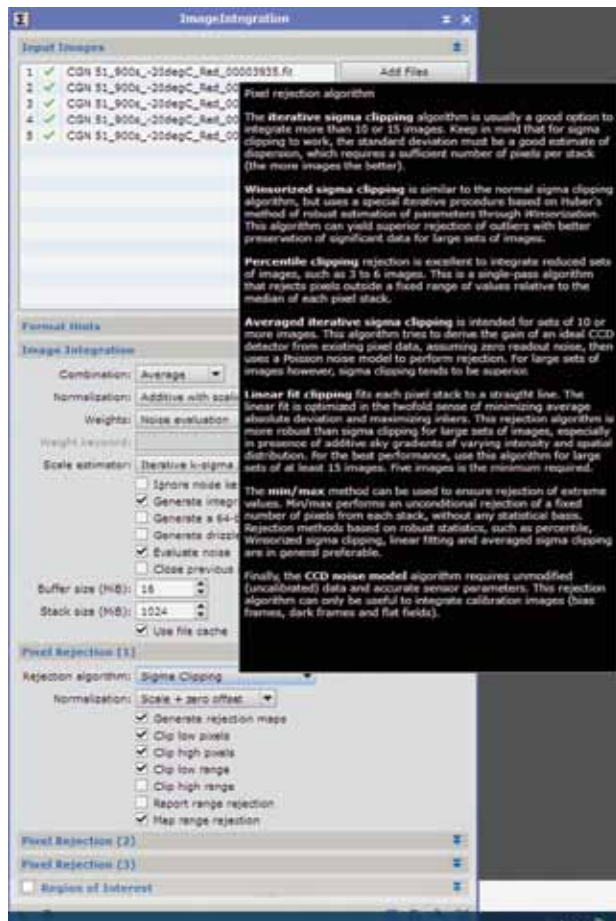
**STACKING** The ImageIntegration process generates a master frame from a group of calibrated and aligned images. Hovering over a selection (in this case the Rejection algorithm) displays helpful tips about the options available.

data set, depending on how many sub-exposures I'm combining. The program will open a description of each method when the cursor is hovered over each option for a few seconds. Once I've loaded the tool with the images to be combined, I apply the process by clicking the blue circle at the bottom left of the window.

In a few moments, the data reduction is complete, producing a single stacked result. I'll save this file and then repeat the process for each filter group.

## Linear Image Processing

After data reduction is complete, I'll open up each of my stacked results. These images will appear very dark, with only a handful of stars visible. This is because most deep-sky objects are much fainter than starlight, so they get recorded at the shadow region of the histogram. We can "stretch" the display in *PixInsight* using the Screen-TransferFunction process to better display the shadow regions, which doesn't modify the actual data.

Before performing any additional processing, I'll inspect the images carefully for any leftover hot and cold

pixels. If only a few spots need cleaning up, I'll use the clone stamp tool found at Process > Painting > Clone-Stamp; if I find many, I'll apply the CosmeticCorrection process again. Next, I'll crop out any non-overlapping edges in my images using the DynamicCrop process. This tool crops each image exactly the same, to remove areas that could interfere with later processing steps.

My next action is to address any light-pollution gradients in the images. Gradients can severely limit the quality of the final picture, particularly in a color image. Fortunately, there are two excellent tools in *PixInsight* to fix these problems: AutomaticBackgroundExtraction (ABE) and DynamicBackgroundExtraction (DBE). Both work well on individual files or a color image. (See *S&T*: Sep. 2014, p. 68, for a detailed article on using these powerful functions.)

Now I'll combine my individual red, green, and blue images into a color result using Process > ChannelManagement > ChannelCombination. This tool should automatically assign your filtered images to their respective channels, but if not, they can be selected manually. Click the blue "Apply Global" circle at the bottom left of the window and in a moment your color-combined image will appear. Be sure to save the result.

Often the color image will appear to have a strong bias toward one color. This can be corrected using the ColorCalibration process, but in difficult cases, applying the BackgroundNeutralization process first can help. Both often work well without any changes to their default settings.

At this point, I'll add any Hα data to the color image if available. Many experienced imagers struggle with blending Hα into a color image. Fortunately, *PixInsight* has an excellent script for the job. The NBRGBCombination script (under Script > Utilities) is very easy to use and produces a rich, natural-looking red color in emission nebulae without affecting star colors. With the
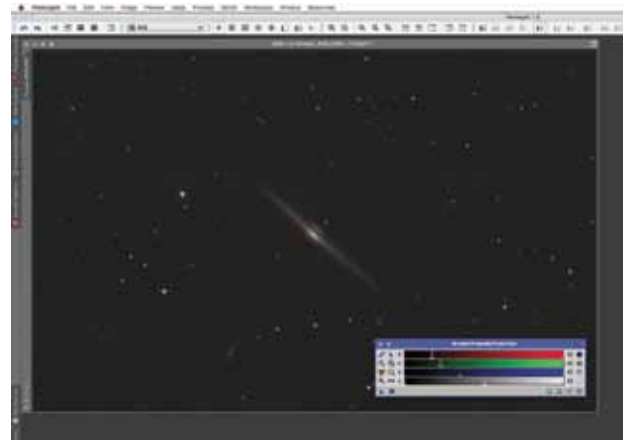


**SCREEN STRETCH** Use the ScreenTransferFunction tool to display an image while continuing to work with it in its unstretched (linear) state.

script open, I'll select my color image in the RGB Source Image section and the Hα image in the Narrowband for R channel (eg Ha). Because Hα also emits at blue wavelengths, adding some hydrogen-alpha signal to the blue channel can make for a more natural-looking result, though reduce the Scale setting in the blue channel to about 0.02 and click Apply. It's usually necessary to re-balance the color on the blended image.

With my color data assembled, it's time to address the luminance channel. I often shoot an unfiltered luminance image, which has the benefit of recording faint details with a very high signal-to-noise ratio compared to the color data. This can then be added to the color picture to produce a very smooth, detailed result. The strong signal recorded in my luminance data can then be deconvolved to reduce blurring due to atmospheric turbulence. Deconvolution is a complex mathematical process that reverses some of the degradation and enhances small-scale details in the image. Getting it right takes a fair amount of experimentation (and will be detailed in an upcoming issue).

## Noise Reduction

My next step is to address noise in the image using the MultiscaleLinearTransform (MLT) process. This tool can
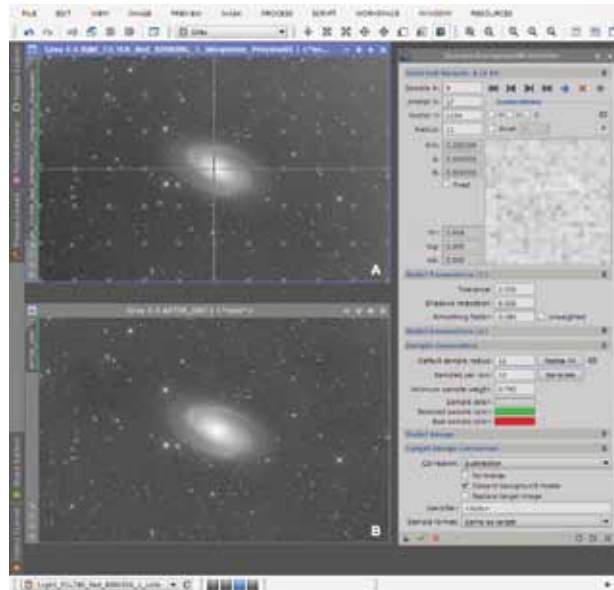


**GRADIENT KILLER** Samples are placed on an image of M81 to remove a light-pollution gradient using the DynamicBackgroundExtraction process.

**ADDING NARROWBAND** *PixInsight* makes it easy to enhance a color image with hydrogen-alpha data using the NBRGBCombination script.
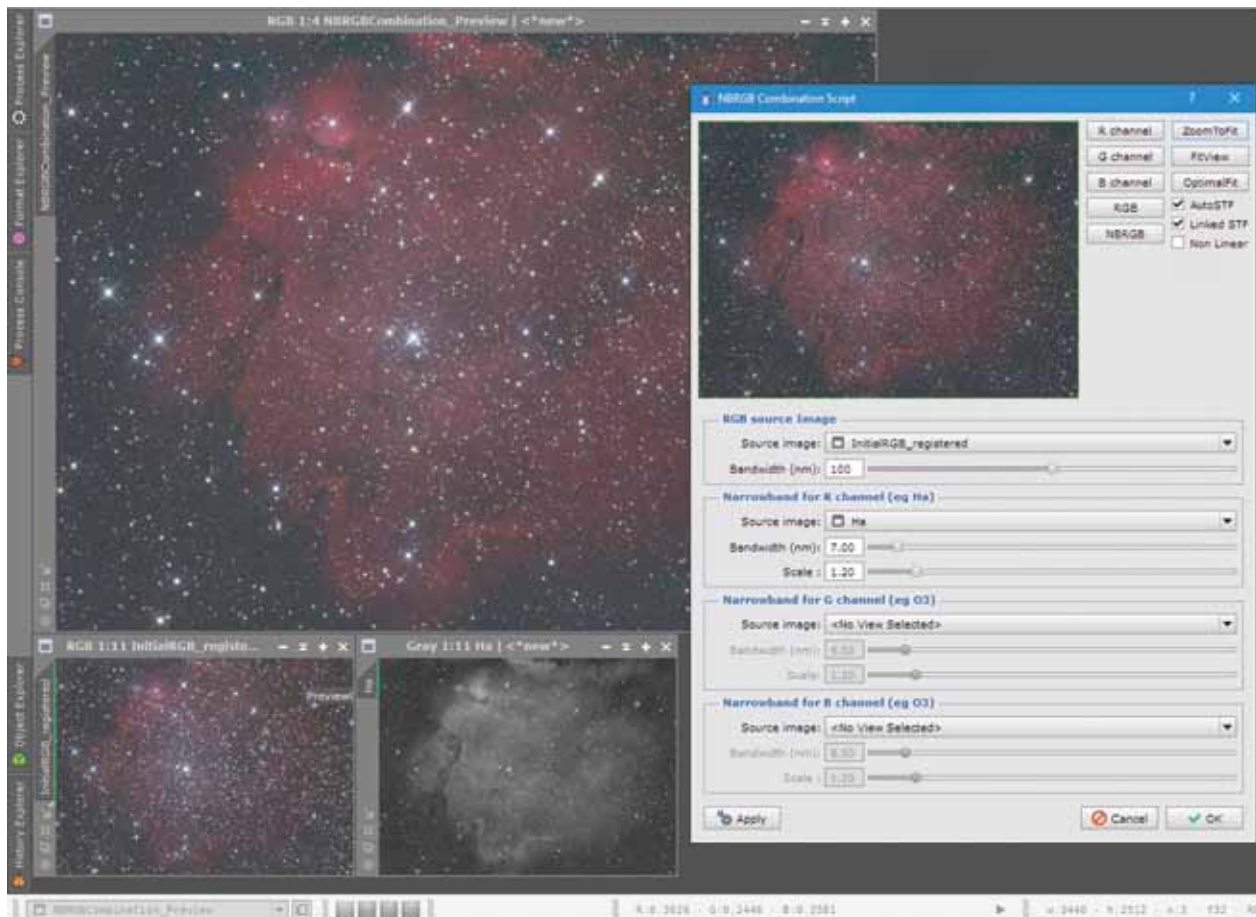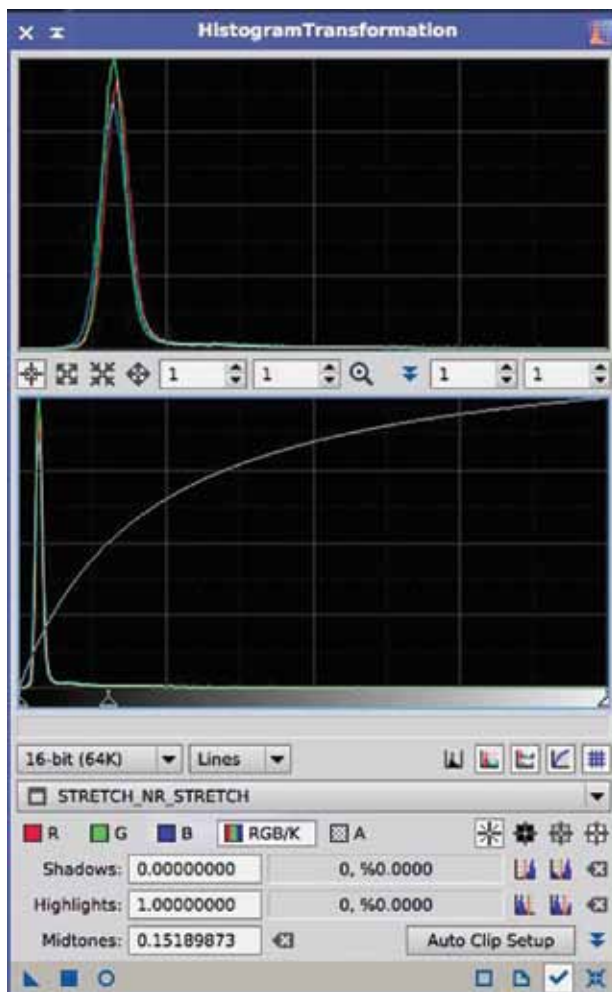
**GOING NONLINEAR** Once you've cleaned up and deconvolved the image, it needs to be permanently stretched using the HistogramTransformation process. Move the middle caret toward the left to bring out the mid-range of the data, and then move the left caret toward the right to darken the background.

isolate small-, medium-, and large-scale noise kernels, allowing noise reduction to be targeted only in the areas where it's needed. I always apply MLT to both my color and luminance images, usually with the same settings for both. For each Layer from 1 to 4, I adjust the Threshold and Amount sliders to get the desired result. Try a Threshold of 3 and Amount of 0.5 for Layer 1, and lower values for each subsequent layer up to Layer 4 (noise reduction cannot be applied to the R, or residual, layer). The built-in masking feature should be activated to protect stars and other high-signal areas. The default mask settings usually work well.

Now it's time to stretch the image and make it nonlinear. There are a few different ways to do this in *PixInsight*, but I prefer the HistogramTransformation process. Clicking the open blue circle at the bottom left of the tool's window activates the Real-Time Preview mode. Before making any adjustments, I reset the Screen TransferFunction so the Real-Time Preview displays an accurate preview of the results of stretching. Next, I'll slide the middle caret
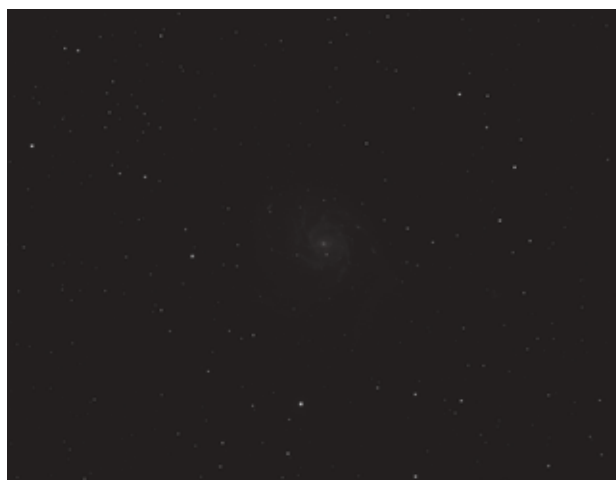
toward the left. This boosts the midtones of the image, though it also increases the background levels. I'll fix this by moving the left caret toward the right, which controls the black point in the image. To preserve significant data, I never adjust the white point, as doing that tends to bloat and overexpose the stars. Similarly, I only move the black point enough to darken the background without clipping out any faint details in the image.

## Working with Nonlinear Images

With my image stretched after basic cleanup, it's time to be a little more artistic. This is where I can sharpen the luminance image, perform additional noise reduction, add it to the color data, and adjust the contrast and saturation. I prefer to work mostly on the Luminance channel and then blend it into the color image near the end.

At this point I want to selectively process regions the image. This is where masks come in. Masks are extremely important when processing nonlinear images in *PixInsight*. These are black-and-white images that I

overlay on the picture I'm working on, allowing processes to modify only the parts of the image I choose.

The program offers two main ways to make masks: RangeMask is used for selecting a range of brightnesses and can make masks that include stars and extended objects like galaxies. As its name implies, the StarMask process generates masks that isolate the stars. Both of these tools are intuitive to use, and tool tips are available to help you set each slider. In some cases, the luminance channel of a color image can make an excellent mask. Luminance can be extracted from a tricolor image using the ChannelExtraction command in Lab mode. A copy of a greyscale image can also make a good mask. Any image can be copied by dragging the view identifier tab at upper left of the image window to an empty spot on the *PixInsight* workspace. More complex masks can be created from simple masks using PixelMath, and there are some mask-related scripts, like ColorMask, that are sometimes helpful. Once I've made a mask, I'll use the selections in the Mask menu on the main toolbar to select, invert, or remove it.

Although I performed some noise reduction before stretching, an additional application is often necessary at this stage. I prefer to use the TGVDenoise process on nonlinear files. I tend to apply it only to the darkest areas of the luminance image, using a mask to protect stars and other structures. After applying noise reduction, the image can be stretched again by moving the black and mid-point carets — again taking care not to clip any significant data.

I can now combine the luminance with the color image using the LRGBCombination process. The best results are usually obtained when the two images being combined have a similar brightness. LinearFit, found in the ColorCalibration menu, can be applied with default settings to match the brightness of two images. Just specify the source image and drag the triangle (New Instance) icon to the target image.

After my LRGB image is created, I might use the CurvesTransformation process to adjust contrast, brightness, and color saturation. Additional tweaks such as sharpening are performed using UnsharpMask, and I might also employ ExponentialTransormation to bring up faint features. The use of masks is advised for all of these steps, with only the final contrast and saturation tweaks on the unmasked image.

## Sharing Online and in Print

Once my image is ready, I'll upload it to my website — and, of course, send it to *Sky & Telescope*. I'd strongly advise serious imagers to invest in a color-managed workflow so that others see your images the same way you do. It's best to use a hardware monitor calibration device such as the Spyder tools available from **datacolor.com**. With a calibrated monitor and *PixInsight*'s color management and proofing tools enabled and properly configured, my images look as good in print as they do online. All the settings and history of your project, including masks, processes, workspaces, and more, can be saved for reference at a later date (File > Save Project).

A final tip — find some constructive and friendly critics on various online imaging forums. Many there know what it takes to make a compelling astrophoto, and they might give you just the idea you need to make a great picture even better. ✦

*Ron Brecher* *images deep-sky objects from his home observatory in Guelph, Ontario. Visit his website at* **astrodoc.ca***.*
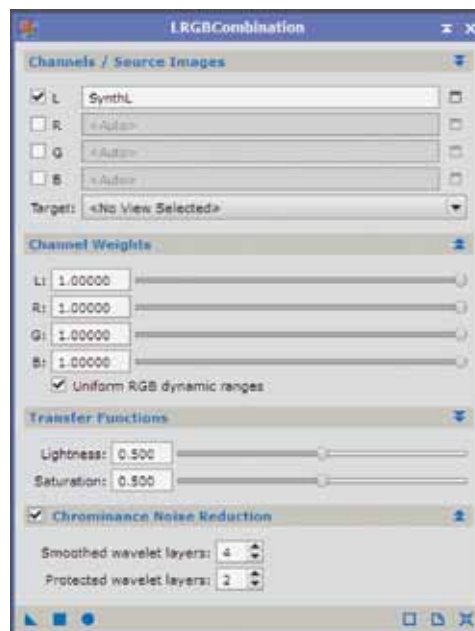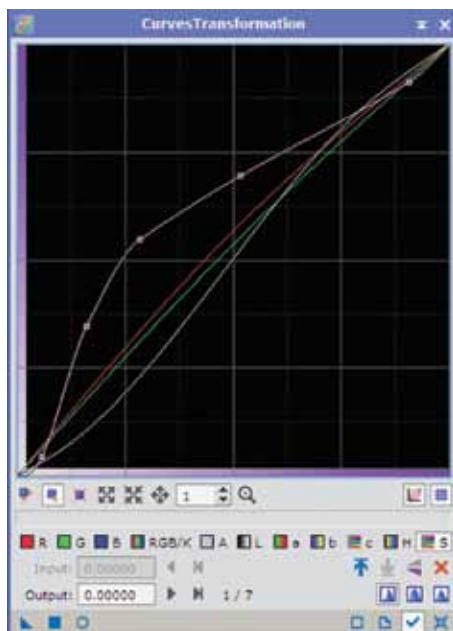
**FINAL CURVES**
*Far left:* The Curves process allows you to simultaneously adjust the brightness and contrast of the image (white), and the saturation (pink), as well as the individual color channels.

**ADDING LUMINANCE**
*Left:* Use the LRGBCombination process to replace the brightness of an RGB image with Luminance data processed separately. The contribution of each to the final result is controlled with the Transfer Functions sliders, and noise reduction can be applied to the color image, all in one process.