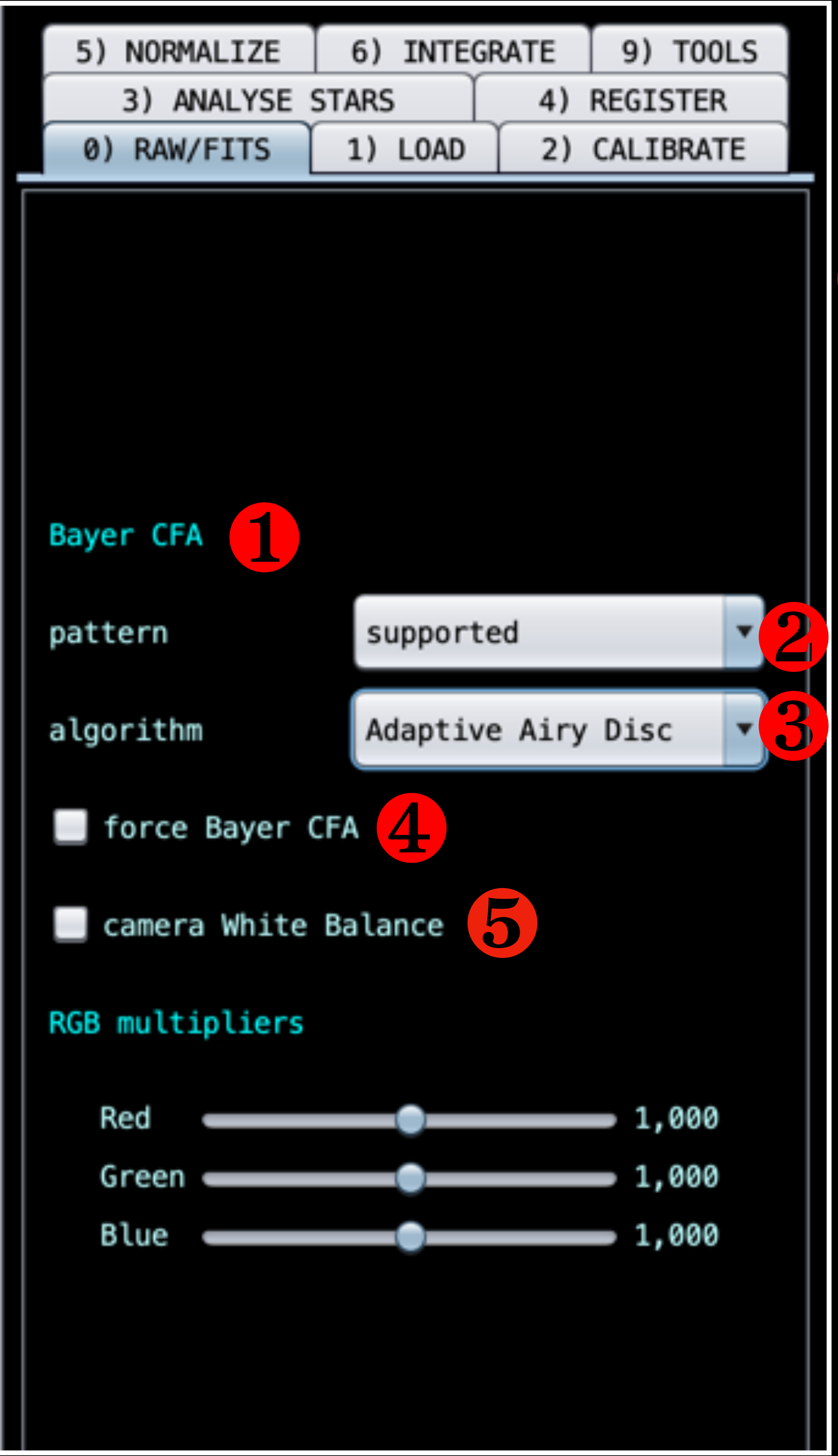


1

Bayer Color Filter Array (or CFA) image processing settings for DSLR RAW & FITS data

2

Choose the bayer pattern of your sensor for correct debayering of your data

- supported
For DLSR images, the correct pattern is automatically chosen. For FITS images, this depends on an appropriate FITS keyword in the fits header. If the keyword is missing, APP will use the default setting of RGGB.

- RGGB, GBRG, BGGR, GRBG
Choose any of these four patterns if the **supported** setting isn't correct. You can check the difference by setting a different pattern and then reloading the image with the image viewer mode options.

4

Enable this if the frames are monochrome Bayer CFA frames and APP doesn't detect this should be interpreted as such.

5

Enable to use the camera White Balance of DSLR RAW images

3

Choose the debayer algorithm

These algorithms have a big influence on your data which will translate into the quality of your integration results. For RGB data, the **Adaptive Airy Disc** (AAD) Algorithm is highly recommended. If you have used a narrowband filter with your One Shot Colour (OSC) camera, choose the appropriate algorithm for the filter that was used.

- no interpolation
This will show you the monochrome raw CFA data without interpolation.

- Bilinear
This is the most simple debayer algorithm. It gives artefacts, blurs details and injects a lot of chromatic noise in your data when compared to the other options.

- Adaptive Edge
This is an advanced debayer algorithm for processing of normal photography images. It gives the least artefacts along lines in your images, since it adapts to contrast edges.

- Adaptive Airy Disc
This is a very advanced debayer algorithm for processing of astrophotography images and therefor the default algorithm in APP. This algorithm is developed by Mabula, especially for astrophotography. It performs better than any other algorithm (like AHD, VNG). It gives the best resolution, least artefacts, least chromatic noise and the best colors, especially after background and star color calibration of your integration result.

- Super Pixel
Super Pixel modus will not debayer your data, instead each 2x2 CFA block on your sensor will be replaced by a 1x1 block. The R,G,B values will be the values that were recorded by your sensor in the original 2x2 block. For green, this means the average of the 2 green cfa pixels. Super Pixel modus will downscale your data by a factor of 2, so only use this setting if you have a clear reason to start with downscaling your data. A reason could be to make a downscaled integration as a preview of your data. Please realise, that the super pixel modus will have a big degrading effect on the registration quality of your results, since star lokation calculations will have much bigger uncertainty. Super Pixel modus immediately throws away valuable information.

- Hydrogen Alpha, Beta, Sulfur II, Oxygen III, Nitrogen II
You can directly debayer narrowband data, shot with your One Shot Colour (OSC) camera. If you choose any of these algorithms, you will directly have the monochrome narrowband exposure. There is no need to seperate the channels first. This method is superior to any other workflow used in processing narrowband data shot with an OSC camera. It preserves resolution, which also has the benefit that registration will not suffer. In other workflows, the integration result usually is upscaled again by using drizzle integration which will inject a lot of noise.

- Ha-OIII color, mono, extract Ha, extract OIII
These debayer algorithms are especially for so-called Multiple Narrowband filters (STC Astro DUO, OPT TRIAD Tri-Band Narrowband filters). These filters enable you to capture narrowband data from Hydrogen Alpha and from Oxygen III at the same time with a One Shot Color camera. Since you will capture data, shot with these filters, with your OSC camera in the Red, Green and Blue channels, there are different ways to process the data. You can simply interpret the data directly as it is recorded in RGB (**Ha-OIII color**). You can interpret it as luminance data, so monochrome (**Ha-OIII mono**). You can even directly extract the Hydrogen alpha (**Ha-OIII extract Ha**) and Oxygen III (**Ha-OIII extract OIII**) data directly from these exposures, which will give you much more control in post-processing.

The **Ha-OIII color** algorithm is an improvement of processing data shot with these kind of filters, over the default **Adaptive Airy Disc (AAD)** debayer algortihm. The OIII data quality and sharpness is the same, since the AAD algorithm is perfect for the OIII data that has data in both the green and blue channels. For the Hydrogen alpha data however, the AAD algorithm is not suitable. The HA-OIII color algorithm **will treat the Hydrogen alpha data totally separate from the OIII data and will therefore give a much better reconstruction of the Hydrogen alpha data versus the AAD algorithm.** The noise in your H-alpha data will be much lower so in integrations of several light frames, the Signal To Noise Ratio will be much higher as a consequence.

1

Push this button to add frames that don't need to be calibrated and/or integrated now.

4

Enables auto-detection of Master calibration frames and Integrations that were created earlier with APP.

With this enabled, all master calibration frames (MasterBias, MasterDark, MasterDarkFlat, MasterFlat & BadPixelMap) and Integrations can be loaded with any of the

Light, Flat, Dark, DarkFlat, Bias

or other load buttons. APP will recognize that these are Master frames from the MetaData in the file header, and will use them as such.

So if you load a previously created MasterBias, it is automatically recognized and loaded as a MasterBias. If you load an old Integration, it will be recognized and loaded as an Integration.

If you disable this option, a previously created MasterDark can be loaded as a MasterDarkFlat now by using the MasterDarkFlat load button. And an old Integration can be loaded as a new Light frame to be included into a new Integration.

Integration Output Maps (outlier rejection, Drizzle/MBB weight, normalization) are always detected automatically.

6

Push this button to add Light frames.

Removes unselected Light frames

Select/deselect all Light frames

7

Push this button to add Bias frames.

Push this button to add DarkFlat frames.

Push this button to add Dark frames.

Push this button to add Flat frames.

2

Enables Multi-Session processing.

Enable this if you need to calibrate, register, normalize and/or integrate data that was shot in different imaging sessions.

If you load light and/or calibration frames, you will be asked to which session the data belongs.

All data will be registered and normalized to the (automatically) chosen reference frame.

In 6) INTEGRATE, the separate sessions will be automatically integrated and will all have the same bitdepth and image dimensions.

The integrated sessions can be loaded directly into the 9) RGB Combine tool.

5

The object name will be used to give the final light frame integration(s)/stack(s) a suitable file name.

8

Push this button to add MasterFlat frames.

If you load a single Flat frame as a MasterFlat, it will be directly used as a MasterFlat.

Push this button to add MasterDark frames.

If you load a single Dark frame as a MasterDark, it will be directly used as a MasterDark.

Push this button to add MasterDarkFlat frames.

If you load a single DarkFlat frame as a MasterDarkFlat, it will be directly used as a MasterDarkFlat.

Push this button to add MasterBias frames.

If you load a single Bias frame as a MasterBias, it will be directly used as a MasterBias.

Push this button to add Bad Pixel Maps (BPMs).

Please be aware, you can only load BPMs that were previously created in Astro Pixel Processor.

3

Enables Multi-Channel/Filter processing.

Enable this if you need to calibrate, register, normalize and/or integrate data that was shot with different filters.

If you load light and/or calibration frames, you will be asked to which channel/filter the data belongs.

All data will be registered and normalized to the (automatically) chosen reference frame.

In 6) INTEGRATE, the separate channels/filters will be automatically integrated and will all have the same bitdepth and image dimensions.

The integrated channels/filters can be loaded directly into the 9) RGB Combine tool.

9

Toggle sorting of all frames for the chosen frame characteristic

best to worst:
- the best frame for the chosen frame characteristic will be at the top, the worst at the bottom

worst to best:
- the worst frame for the chosen frame characteristic will be at the top, the best at the bottom

So the actual chosen frame characteristic determines how the data is sorted.

For instance, for **SNR** (Signal to Noise Ratio) this means, that the **highest values are shown at the top** of the frame list panel. For **noise** however, this means that the **lowest values are shown at the top.**

By clicking this button the sort order will be reversed.

1

Choose an integration method for the MasterBias frame integration.

median – Use **median** with less than 20 frames, this will remove outliers as well, so you can disable the use of an outlier rejection filter or relax it's settings.

average – If you have more than 20 frames, use **average** integration combined with outlier rejection, this will be superior in most cases in terms of noise reduction in your integration results.

maximum – Use **maximum** to show all outliers/artefacts (with too high ADU values) in the data like hot pixels, cosmic rays etc...

automatic – With the **automatic** integration method, APP will automatically set either median or average integration, and it will enable an outlier rejection filter with suitable kappa low and high values, all depending on the number of frames in the integration/stack.

3

Choose the **kappa low** value for the outlier rejection filter.

This value is responsible for **outlier rejection on the low side** of the pixel distribution in the individual pixelstacks.

In almost all cases, we will have very little outliers on the low side, so you can **safely set the kappa low to a value of 4-8 kappa.**

Please realize, the lower the kappa value, the more you will remove including good data. **If you set it too low, it will remove all outliers and also a lot of good data, thereby reducing the Signal to Noise Ratio (SNR) / Quality of your results**

4

Choose the **kappa high** value for the outlier rejection filter.

This value is responsible for **outlier rejection on the high side** of the pixel distribution in the individual pixelstacks.

In almost all cases, we will have the most outliers on the high side, so you want to **set the kappa high value to a value that is clearly lower than the kappa low value.**

A reasonable value for kappa high is between 2.2 and 4.

Please realize, the lower the kappa value, the more you will remove including good data. **If you set it too low (lower than 2 is usually way too low), it will remove all outliers and also a lot of good data, thereby reducing the Signal to Noise Ratio (SNR) / Quality of your results.**

2

Choose an **outlier rejection** filter.

Such a filter will **remove bad pixels, satellite/airplane stripes, cosmic rays, etcetera...** in your stack/integration result. Thus it will remove signal that is only present in a small fraction of your frames per pixelstack.

Which filter do I need to use? – If you have less than 20 frames, **winsorized rejection** is the preferred filter. If you have more than 20 frames, **adaptive rejection** is the preferred filter.

Winsorized/MAD/adaptive rejection is preferred over **sigma rejection** in almost any case.

no rejection – no outlier rejection is applied.

sigma rejection – this is an outlier rejection filter which uses the **Weighted Median** as the central value and the **Weighted Standard Deviation** (square root of the variance) relative to the central value for outlier rejection.

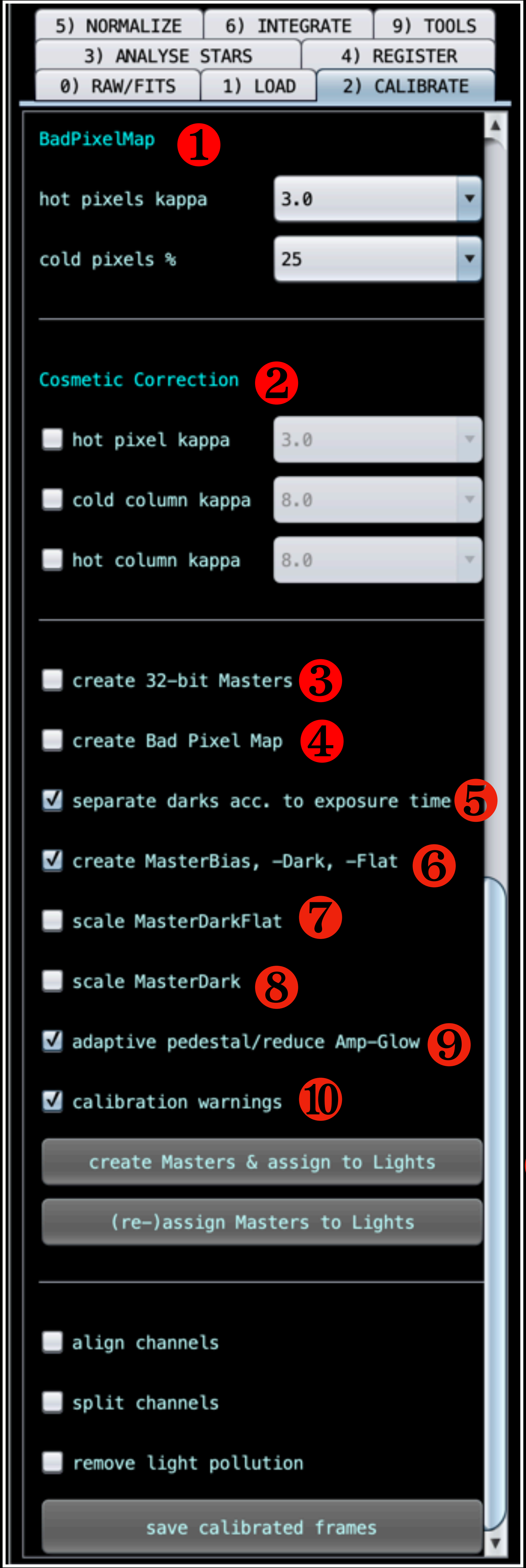
winsorized rejection – uses a **Winsorized Weighted Median** and **Winsorized Weighted Standard Deviation** (square root of the variance) to perform outlier rejection.

MAD rejection – this is an outlier rejection filter which uses the **Weighted Median** as the central value and the **Weighted Median Absolute Deviation** (corrected to correspond to a standard deviation) relative to the central value for outlier rejection.

adaptive rejection – this is a more advanced outlier rejection filter in which **sigma, winsorized and MAD rejection** are combined to further improve the outlier detection in the pixel stacks. This filter should work as the best filter if you integrate more than 20 frames.

Be carefull with outlier rejection! setting the kappa values too low, will remove the outliers, but also a lot of good data, so it will hurt the Signal to Noise Ratio significantly.

Because of the potential harmfull effects of outlier rejection, it's always best to try to get rid of most outliers before integration, for instance: **remove all bad pixels with a Bad Pixel Map (BPM)** to begin with. A BPM will never harm your data and you can get rid of a lot of unwanted noise to start with.



1

This kappa value times the gaussian noise in the darks or master dark is used to determine whether a pixel is hot

This percentage of the median of the master flat, is used to determine whether a pixel is cold or defect.

2

Cosmetic Correction

Set the hot pixel correction kappa value

A lower value will correct more hot pixels, a higher value will correct less.

3

Enable to create 32bits masters for calibration.

Disable to create masters in the same bitdepth as the indiivdual calibration frames.

4

Create a Bad Pixel Map or BPM

The first thing you should do in data calibration is to try to create a very good Bad Pixel Map of your sensor.

A good Bad Pixel Map will correct all bad pixels of your sensor. **Bad pixels are pixels that don't behave linear.**

We have two types of bad pixels, cold and hot:

cold pixels - these pixels are either defect or respond really bad to incoming photons (and/or the dark current of your sensor). These will be detected using flat frames. For the detection of the cold pixels, only a couple of flats are needed to extract them robustly and add them to the bad pixel map.

hot pixels - these pixels respond very strongly on incoming photons (and/or to the dark current of your sensor). These will be detected using dark frames. For the detection of the hot pixels, you need to supply at least 20 darks and more is always better in this case.

For detection of bad columns (hot & cold) you will need to supply both darks and flats.

The BPM is a Master Calibration frame that you only need to create once per year or even a couple of years.

In the BPM, the cold pixels have the value of 0, the hot pixels will have the value of 255. The remaining pixels are considered linear and have the value of 127.

5

Separate individual darks according to exposure time

If you load dark frames with different exposure times, enable this setting to create separate MasterDarks per used exposure time.

If you disable this setting, then all darks of different exposure times are bundled together to create one masterdark.This could be useful to create a robust masterdark which you can use together with dark scaling.

Please realise that darks of different ISO/gain values are always treated separately, so different MasterDarks will be created if you load darks with different ISO/gain values.

Finally, darks that are assigned to different channels/filters or sessions, will off course be treated separately.

6

Create the MasterBias, MasterDark, MasterDarkFlat and/or MasterFlat

If you are creating a Bad Pixel Map, by enabling this selectbox, the MasterBias, MasterDark and/or MasterFlat will be created as well, if the required frames are loaded. However, if you disable this selectbox, only the Bad Pixel Map will be created.

If you aren't creating a Bad Pixel Map, this setting will be enabled by default.

7

Enables MasterDarkFlat frame scaling in Flat calibration.

Dark scaling requires a MasterBias to be loaded as well.

Dark frame scaling will correct the MasterDarkFlat to minimize noise in the calibration result.

8

Enables MasterDark frame scaling in Light calibration.

Dark scaling requires a MasterBias to be loaded as well.

Dark frame scaling will correct the MasterDark to minimize noise in the calibration result.

9

The **adaptive data pedestal** will guarantee that your data isn't harmfully clipped on the black point in calibration.

This setting is only applicable in case of dark frame calibration.

This setting will make sure that statistics of pixelstacks in integration aren't skewed to the upside from missing statistics on the lowside.

A case where this can happen is with strong Amplifier Glow on the sensor (Amp-Glow), resulting in residual **Amp-Glow** after data calibration.

For most camera's and datasets, this setting will have no effect, but in cases where severe black clipping in calibration does occur, for whatever reason, this setting will prevent it and will ensure precise calibration and no-skewed statistics in image integration.

Enabling this setting will reduce calibration speed a little bit.

10

Enable this to have APP warn you about a possible suboptimal calibration path.

Warnings will occur if you:

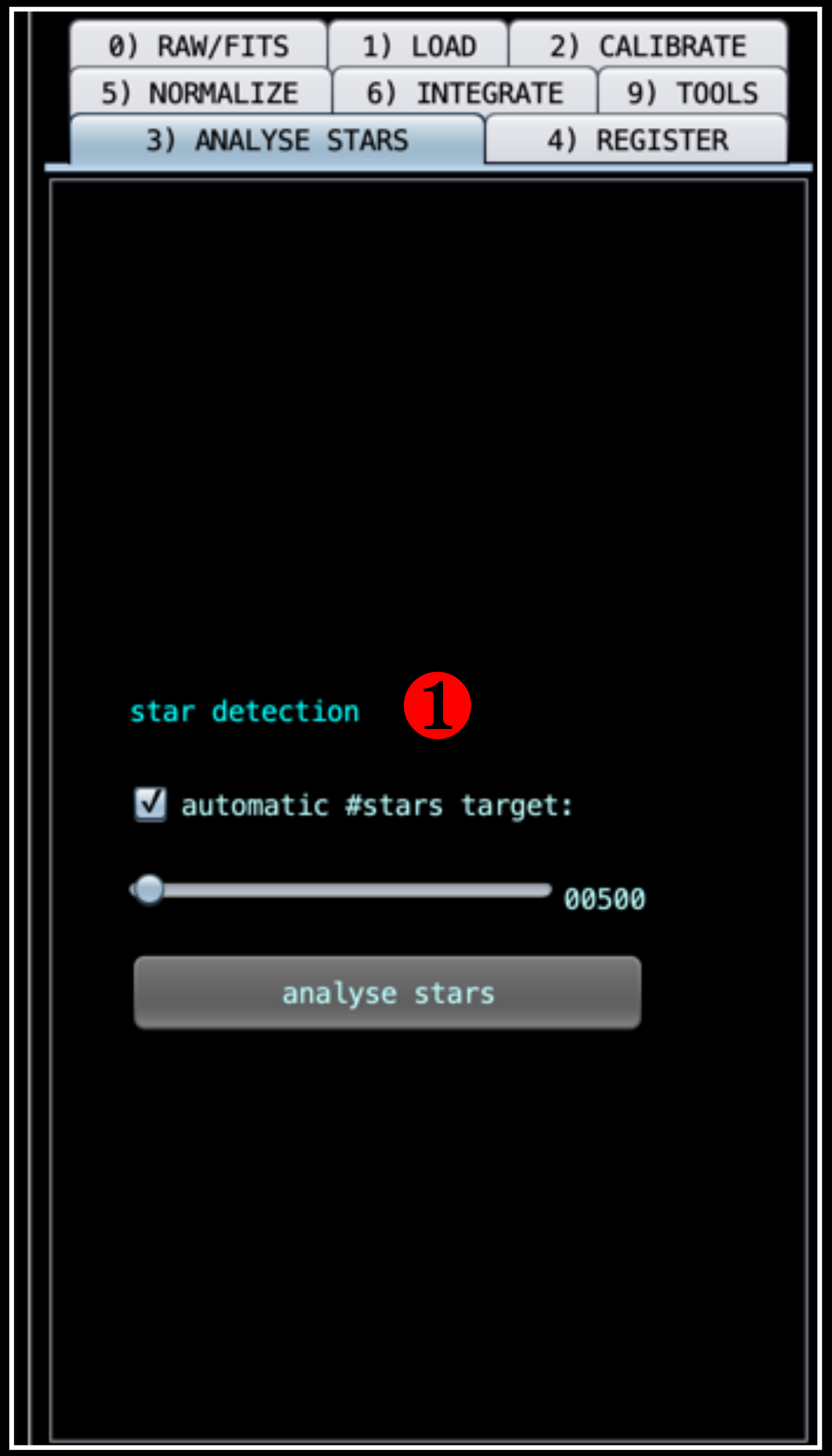
- match a MasterBias or MasterDark/DarkFlat to a Light/Flat frame that doesn't match on ISO or gain value.

- match a MasterDark/DarkFlat to a Light/Flat frame that doesn't match on exposure time if **dark frame scaling is disabled** or **not possible due to lack of a MasterBias.**

- if the Flats can't be calibrated with either a MasterBias or a MasterDarkFlat or both. In this case, Flat-Field correction can't work correctly.

- if the Lights can't be calibrated with either a MasterBias or a MasterDark or both. In this case, Flat-Field correction can't work correctly.

- if you match a Bad Pixel Map to a Light/Flat frame that doens't have the same instrument/camera name in it's metadata/file header.



1

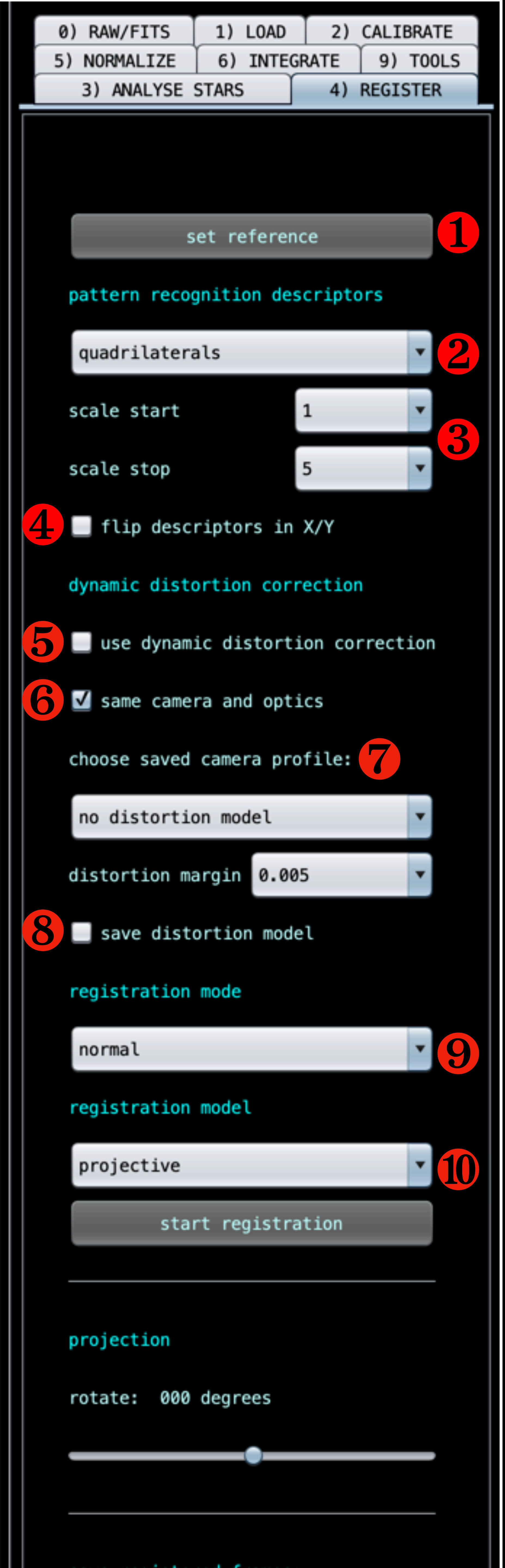
Enable/disable **automatic #stars target**

This setting will set an upper limit to the amount of detected stars in your images.

This setting is very usefull to limit the amount of stars detected in your images and will therefore make registration more robust and faster. Only the brightest stars will survive this limitation which is vital for robust registration as well.

For regular (non-mosaic) integrations, a number of 500-1000 stars per frame is desired. For mosaics, it's better to increase the desired amount of detected stars to at least 2500, especially if your mosaic frames have little overlap, i.e. less than 10%.

Only disable this setting if registration fails and you have already tried to increase the #stars target to a higher value.



1. The frame with the highest quality is used as the reference frame for registration unless you specify otherwise.

2. Choose the registration descriptors for pattern recognition between the frames.

   Quadrilaterals are the default, because they are the fastest and most robust in most situations.

   Choose only triangles when there are very little star pairs found between the frames to be registered, because it will be very slow and unreliable in the case of high star pair counts.

   triangles – 6 to 250 star pairs between 2 frames.

   quadrilaterals – 250 – 5000 star pairs between 2 frames.

   pentagons – more than 5000 star pairs between 2 frames.

3. Set the scale start.

   This is the scale of the patterns that are created for pattern recognition.

   For **normal** registration mode, scale start at 1 and scale stop at 5 is fine.

   For **mosaic mode**, you will need to increase the scale stop usually to at least 10. This will correspond roughly with having a 10% overlap between the mosaic panels. If you have a bigger overlap between the panels, for instance 15%, than the scale stop can be relaxed to 7 or 8.

   Only increase the scale stop if some frame(s) couldn't be registered which do have overlap and be aware that increasing the scale stop will slow down the registration in the case of high star counts in the frames.

   For mosaics that are composed regularly, like a 4x4 mosaic, where all panels overlap for about 10-20%, then the scale start can be set higher, usually a value of 5 is fine.

4. If some of the frames are only flipped horizontally (or only vertically) with respect to the reference frame, enable this.

   Usually, this setting only needs to be enabled when you register frames of different optical configurations.

5. If the frames to be registered have some degree of optical distortion, please enable this to correct for this while registering.

   The registration RMS values will indicate if you need to have this enabled.

6. If all the frames to be registered are taken with the same camera and optics and aren't touched/warped already by a registration module from APP or any other application, enable this. Otherwise disable it.

7. Choose the optical distortion model, use dynamic distortion correction (DDC) if you haven't saved the distortion profile of this camera and lens/telescope yet.

   Choose the distortion margin.

   The default value of 0.05 is fine for most datasets. Only set a higher value in the case of strong optical distortion and **if registration fails.**

   Be aware, that setting this value higher will slow-down registration, setting it lower will speed-up registration.

8. Enable this to save the optical distortion model.

   This is only possible if **dynamic distortion correction** is enabled, **same camera and optics** is enabled and the **calibrated projective** registration model is used.

9. Choose a registration mode.

   Choose **normal** for regular registration with a selected reference frame. In this case all frames must overlap with the reference frame.

   If you have frames that don't have overlap with your reference frames, you will need to choose the **mosaic** mode.

10. Choose a registration model.

    This is the coordinate transformation type that is needed to register the images.

    With a **projective** model, the parameters are calculated using regular projective transformations, also called homographies, without any restrictions on the parameters.

    With a **calibrated projective** registration model, the parameters are again calculated using projective transformations, but now with restrictions on the parameters.

    The calibrated projective model is required if your Field Of View (FOV) of your data is larger than 120 degrees, especially in the mosaic mode!

    The restrictions on the parameters are such that the homography is now calculated from a camera matrix and a rotation matrix. These 2 matrices are found to be able to compose the homography.

    The 3x3 camera matrix consists of the focal length (in number of pixels of your camera's sensor) in the x- and y-directions and the principal point (the location where the optical axis hits your sensor).

    The 3x3 rotation matrix is a regular 3D rotation matrix composed of 3 rotations around the x-,y- and z-axis.

    So, with the **calibrated projective** model, each homography will be described by a camera matrix + rotation matrix. In this formulation, no actual coordinate translation (move in x- and y-direction) is needed as part of the registration parameters, since the stars in our images are assumed to be at infinity in this case.

    Please note, if you register your data with the calibrated projective model, you will get additional options to alter the projection of your data on the field of view.Specifically:

    - different projection methods (Rectilinear, Equirectangular, Mercator)
    - move the Center Of Projection (COP) horizontally or vertically.

1

Choose the normalization mode.

If all light frames are of the same image scale and have nearly identical field of view, choose the **regular** mode. Otherwise choose the **advanced** mode.

For mosaic integrations, usually, the **advanced** mode gives far better normalization results.

2

Choose the normalization method

For allmost all datasets, the preferred method is **multiply-scale.**

The methods without scale, will only normalize for the lokation (the background) of the data. The methods with scale, will normalize the data for both lokation and scale/dispersion of the data.

Normalizing with scale/dispersion correction, will therefore work much better, giving you better Signal to Noise Ratio's in the resulting integrations of your light frames.

If you combine data of **different ISOs/gains, exposure times, filters and or sessions it will be essential to normalize for both location and scale using multiply-scale.**

Multiply-scale normalization will also additively **correct** for an **adaptive and thus artifical data pedestal** if it is used in calibrating your data. Add-scale will not do this.

3

Choose the method of calculation for the scale(dispersion) of the light frames.

MAD = Median Absolute Deviation

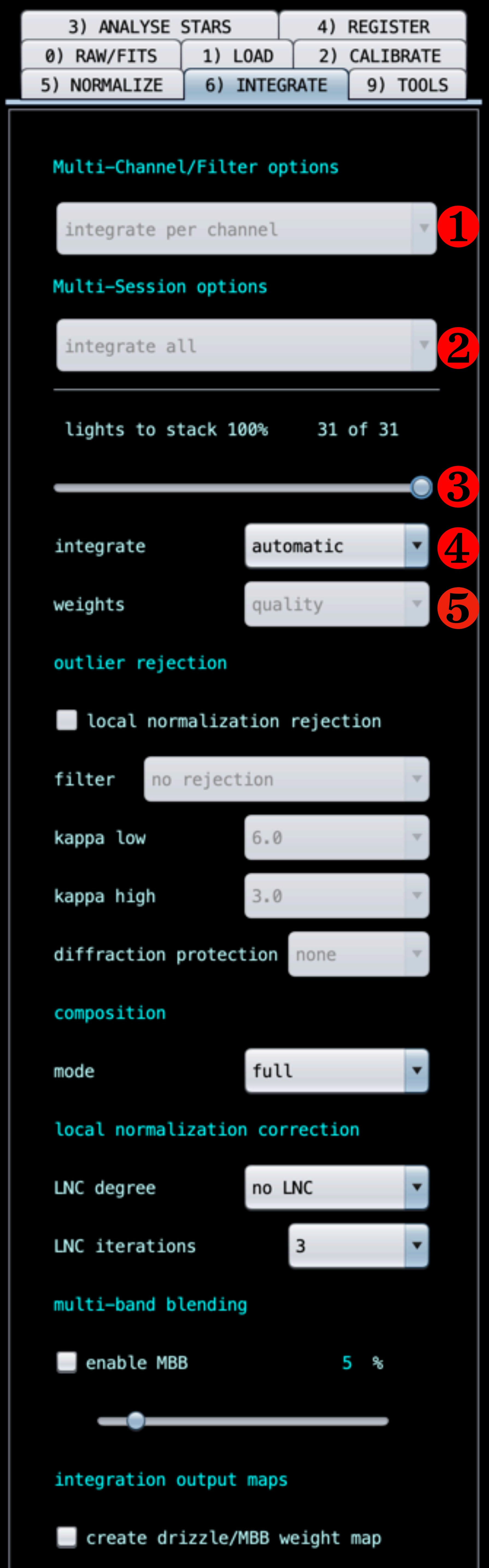
BWMV = Biweight Midvariance

BWMV is the default because it is the most robust for most datasets. Both MAD & BWMV scale parameters are calculated relative to the lokation of the data. The lokation (the background of the sky or roughly the peak of the histogram) is calculated using robust statistics with several outlier rejection iterations.

4

If you want to neutralize the background in your integration, turn this on.

Be aware that background neutralization can surpress nebulosity, so always use the **calibrate background** tool to have the nebulosity fully restored.



1

By default, all channels/filters are integrated separately.

You can also choose to integrate all data of all channels/filters into 1 integration, or do both.

This option is only applicable in Multi-Channel/Filter mode.

2

By default, all sessions are integrated into 1 integration.

You can also choose to integrate all sessions separately, or do both.

This option is only applicable in Multi-Session mode.

3

adjust the percentage of lights that you want to integrate in decreasing order of the quality score of the lights. The frame with the highest score is used as the reference frame for registration unless you specify otherwise.

4

Choose an integration method for the light frame integration.

median – Use **median** with less than 20 frames, this will remove outliers as well, so you can disable the use of an outlier rejection filter or relax it's settings.

average – If you have more than 20 frames, use **average** integration combined with outlier rejection, this will be superior in most cases in terms of noise reduction in your integration results.

maximum – Use **maximum** to create a star trails image or to show all outliers/artefacts (with too high ADU values) in the data like hot pixels, satellites, airplanes, cosmic rays etc...

automatic – With the **automatic** integration method, APP will automatically set either median or average integration, set suitable integration weights (either equal or quality weights) and it will enable an outlier rejection filter with suitable kappa low and high values, all depending on the number of frames in the integration/stack.

5

Set the weight of the frames for stacking.

This setting can have clear impact on the integration result, so you will need to choose the weights that best match what you are trying to accomplish.

equal – this will give you a result which will be subtoptimal in SNR, and suboptimal in sharpness, because all frames are weighted equally. Bad frames will have a big impact on the integration result.

exposure – this setting can be chosen if you combine exposures of different exposure times. The frames with longest exposures will have higher weights which scale as the square root of the exposure. This setting will then give more weight to the frames with higher SNR usually, resulting in an integration with a better SNR, but also less sharp in most cases.

quality – this will use the quality score shown in the **quality score** column in the bottom frame list panel. The quality parameter is based on noise, star density, star size and star shape. This usually gives the best integration result for noise and sharpness combined.

SNR – the weights are now based on the Signal to Noise Ratio of the frames. This is a really dangerous method. Any deviating gradients between the frames will make the SNR metric totally unreliable. From all the settings, this is the least attractive one so it's not recommended. Bad frames with guiding errors, or which are out of focus, or have some clouds, or shot with bad transparancy, could give higher SNR values, strongly reducing the integration result.

noise – use this if you aim for the lowest noise in the end result. The weights calculated by APP are based on the noise values after having applied normalization to the frames, so these are normalized noise values. This is important, because normalization for scale/dispersion will impact the initial noise values of the frames.

star shape – use this if you want to have the smallest and roundest stars in your integration result. It will give weights to the frames based on the star shape of the stars in the frame (star shape means both roundness and size). Frames that have stars that are not round are punished a lot, so these frames will have little weight. This is a very nice integration setting if you have some frames without perfect guiding but still want those frames to help reduce the noise in the data without impacting the star shapes in the integration.

Scale Independent quality parameters.

APP has the parameters star density & relative FWHM which are very helpfull if you combine data of different scales. APP calculates the star size/shape and star density relative to the scale differences between the frames. The scale differences are calculated using the homographies (projective transformations) between the frames.

Multi-Channel/Filter options

integrate per channel

Multi-Session options

integrate all

lights to stack 100% 31 of 31

integrate automatic

weights quality

outlier rejection

1 local normalization rejection

filter no rejection 2

kappa low 6.0 3

kappa high 3.0 4

diffraction protection none 5

composition

1 **Local normalization rejection** - Enable this if you have varying illumination in your frames for more reliable rejection over the whole filed of view. This can be enabled for all filters. **Local normalization** is an additional technique to further improve the regular outlier rejection filters. Locally, in your integration, all data will be corrected when fed to the outlier rejection filter for more efficient and robust rejection by calculating the local median value of each layer in the stack in a radius of 64 pixels. The pixel values in each pixel stack are corrected using these median values and will therefore correct each pixel for local deviations in intensity over the whole field of view. The pixels that survive the rejection filter will be used uncorrected for the actual integration.

Local normalization makes an outlier rejection filter like linear fit clipping obsolete! It correctly **solves the root problem of local intensity differences between the lights of the stack to be integrated.** The result of enabling **local normalization rejection** is that you will get very uniform, and robust rejection over the entire field of view, which can be easily verified by studying the outlier rejection map(s) if created along side the integration.

2 Choose an **outlier rejection** filter.

Such a filter will **remove bad pixels, satellite/airplane stripes, cosmic rays, etcetera...** in your stack/integration result. Thus it will remove signal that is only present in a small fraction of your frames per pixelstack.

Which filter do I need to use? - If you have less than 20 frames, **winsorized rejection** is the preferred filter. If you have more than 20 frames, **adaptive rejection** is the preferred filter.

Winsorized/MAD/adaptive rejection in combination with diffraction protection (if needed) is preferred over **sigma rejection** in almost any case.

no rejection - no outlier rejection is applied.

sigma rejection - this is an outlier rejection filter which uses the **Weighted Median** as the central value and the **Weighted Standard Deviation** (square root of the variance) relative to the central value for outlier rejection.

winsorized rejection - uses a **Winsorized Weighted Median** and **Winsorized Weighted Standard Deviation** (square root of the variance) to perform outlier rejection.

MAD rejection - this is an outlier rejection filter which uses the **Weighted Median** as the central value and the **Weighted Median Absolute Deviation** (corrected to correspond to a standard deviation) relative to the central value for outlier rejection.

adaptive rejection - this is a more advanced outlier rejection filter in which **sigma, winsorized and MAD rejection** are combined to further improve the outlier detection in the pixel stacks. This filter should work as the best filter if you integrate more than 20 frames.

Local normalization rejection - Enable this if you have varying illumination in your frames for more reliable rejection over the whole filed of view. This can be enabled for all filters. **Local normalization** is an additional technique to further improve the regular outlier rejection filters. Locally, in your integration, all data will be corrected when fed to the outlier rejection filter for more efficient and robust rejection by calculating the local median value of each layer in the stack in a radius of 32 pixels. The pixel values in each pixel stack are corrected using these median values and will therefore correct each pixel for local deviations in intensity over the whole field of view. The pixels that survive the rejection filter will be used uncorrected for the actual integration.

Local normalization makes an outlier rejection filter like linear fit clipping obsolete! It correctly **solves the root problem of local intensity differences between the lights of the stack to be integrated.** The result of enabling **local normalization rejection** is that you will get very uniform, and robust rejection over the entire field of view, which can be easily verified by studying the outlier rejection map(s) if created along side the integration.

diffraction protection - all outlier rejection filters can be used with diffraction protection to prevent artefacts at the star borders. Please read the tooltip for diffraction protection for more information on this feauture.

Be carefull with outlier rejection! setting the kappa values too low, will remove the outliers, but also a lot of good data, so it will hurt the Signal to Noise Ratio significantly.

Because of the potential harmfull effects of outlier rejection, it's always best to try to get rid of most outliers before integration, for instance: **remove all bad pixels with a Bad Pixel Map (BPM)** to begin with. A BPM will never harm your data and you can get rid of a lot of unwanted noise to start with.

3 Choose the **kappa low** value for the outlier rejection filter.

This value is responsible for **outlier rejection on the low side** of the pixel distribution in the individual pixelstacks.

In almost all cases, we will have very little outliers on the low side, so you can **safely set the kappa low to a value of 4-8 kappa.**

Please realize, the lower the kappa value, the more you will remove including good data. **If you set it too low, it will remove all outliers and also a lot of good data, thereby reducing the Signal to Noise Ratio (SNR) / Quality of your results.**

4 Choose the **kappa high** value for the outlier rejection filter.

This value is responsible for **outlier rejection on the high side** of the pixel distribution in the individual pixelstacks.

In almost all cases, we will have the most outliers on the high side, so you want to **set the kappa high value to a value that is clearly lower than the kappa low value.**

A reasonable value for kappa high is between 2.2 and 4.

Please realize, the lower the kappa value, the more you will remove including good data. **If you set it too low (lower than 2 is usually way too low), it will remove all outliers and also a lot of good data, thereby reducing the Signal to Noise Ratio (SNR) / Quality of your results.**

5 **diffraction protection** - this is a new concept with the purpose to prevent rejection artefacts at the star borders.

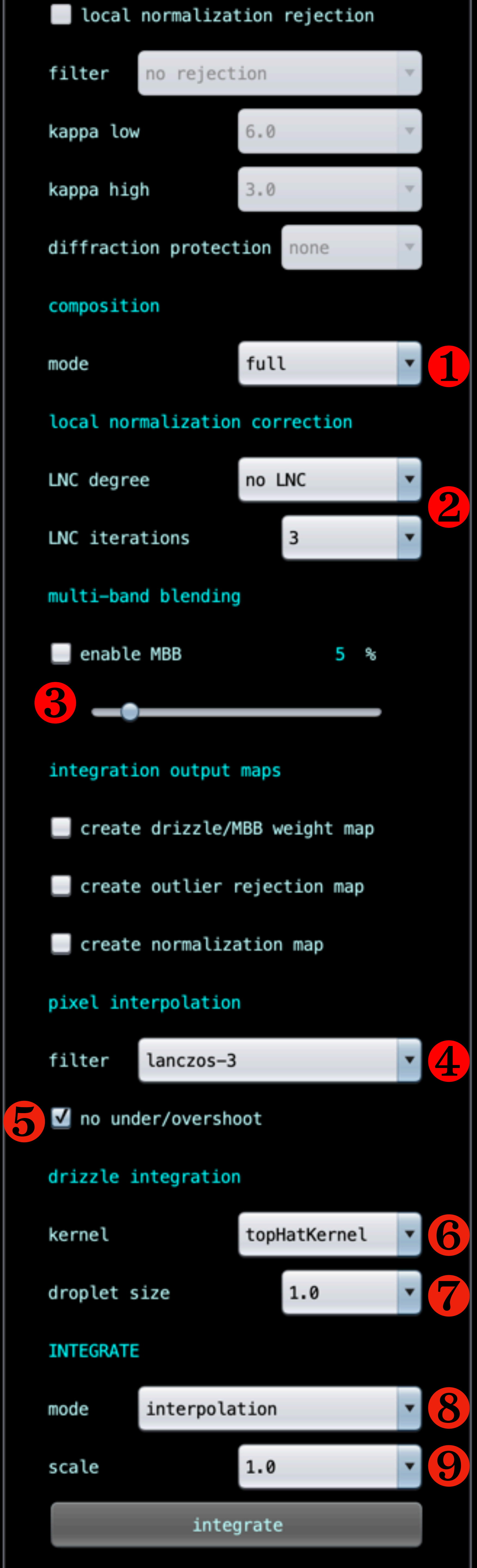
Rejection artefacts will occur if you integrate data with varying star sizes and/or different diffraction patterns. For instance if you want to combine data of separate nights with big differences in atmospheric seeing or if you want to combine data shot with different telescopes like an APO and a Newton telescope.

You can leave this function disabled by setting it to none. Any other setting will enable this feauture on all 4 filters.

If enabled, you can prevent artefacts at star borders by choosing a suitable value. The value correspons to how many pixels are allowed to be rejected in a pixelstack. Realize, that zero pixel values in each pixel stack are not part of pixel stacks. These are removed earlier.

An example, I am going to integrate 20 frames of night 1 and 30 frames of night 2. Normally, each pixel stack will consist of 50 pixels. And out of those 50 pixels, normally only a couple of pixels will be real outliers. It's very reasonable to assume that if more than 10% of the pixels are rejected we are not simply rejecting real outliers. In that case we are probably rejecting pixels at star borders, introducing possible ugly artefacts. In this instance, setting the diffraction protection at 5-10 pixels will now start to prevent a lot of artefacts around star borders.

An extra benefit is that this feature compensates for the well-known artefacts of winsor sigma clipping. So with diffraction protection enabled, winsor sigma clipping can be used perfectly if you want to combine data of different telescopes with different diffraction patterns. Depending on the amount of frames used in the integration, artefacts can become non-existent while achieving very optimal outlier rejection in pixel stacks where it's really needed.



1

Choose the composition of the integration.

full - which is the default, will make a field of view for the integration such that all pixels of all frames that could be registered will be shown in the integration.

reference - will make a field of view that matches the field of view of the reference frame.

crop - will allow you to draw a crop of the reference frame which will be the field of view of the integration. Useful to only integrate a part of the reference frame's field of view.

Click on the **apply mode** button to apply a different composition.

2

Choose the degree of Local Normalization Correction (LNC)

LNC will correct the illumination in your frames in such a manner that the illumination differences of the separate frames in your stack will be matched locally instead of globally.

Normally, normalization is only calculated globally per frame. With LNC, the gradients in your data will be matched per frame while maintaining a solution for all frames at once that is stable and as flat as possible.

If you are combining data shot at 1 single night then LNC will still help, but the improvement will probably be very little and visually hard to see, because locally, the frames already match pretty well.

If you are combining data with deviating gradients (multiple imaging sessions, data from different setups and/or photographers) then LNC should give a very clear improvement.

For regular integrations (i.e. not a mosaic), start with 1st degree and 1 iteration. Then improve from there if needed. First increase iterations to 3. If that doesn't change the outcome, increase the degree. For regular integrations normally 2nd or 4th degree with 3 iterations gives a very nice improvement.

For mosaics, it really depends on the data which degree and the amount of iterations that are needed. The user should experiment, but just like for regular integrations, start with 1st degree and 1 iteration and then improve from there.

3

Choose to enable Multi-Band Blending (MBB).

Multi-Band Blending is a tool to blend images into each other. It will reduce stack artefacts at the borders of regular integrations and will remove seams in mosaics.

In most cases a value of 5-15% is fine. For regular integrations, usually 5-10% is fine. For mosaics, it really depends on the amount of overlap between the mosaic panels. If the amount of overlap is about 10%, setting MBB at 10% will work very nicely.

4

Choose the data interpolation kernel for registration of the frames.

5

Keep this enabled to prevent artefacts around star borders.

6

Choose the drizzle kernel for input sampling.

Always realize, that the choice to use drizzle is a choice between sharpness and noise, because the drizzle algorithm is a big noise injector. And to benefit at all from the drizzle algorithm, you need to meet 2 criteria.

1) your data must be well dithered.
2) your data must be undersampled

If you don't meet both criteria, then it's always better to simply upscale the integration(if you want to upscale) without drizzling using just the **scale** parameter.

A drizzle algorithm needs to calculate what fraction of a pixel of the target pixelgrid is hit by the drizzle droplet. The drizzle kernel, is essentially the shape of the drop of data that hits the pixelgrid.

pointKernel - the shape of the drop is just a point, so the droplet size has no influence in this case. It gives the sharpest and also noisiest drizzle result. It gives the same result as setting the droplet size to virtually 0 for the other kernels.

squareKernel - this drop has a 3D square shape and will give less noise, but could result in some squared star shapes.

topHatKernel - This is a circulair kernel with equal height and the default in APP, because it gives the smoothest results with the least noise in most cases.

gaussKernel - the droplets with this kernel have the shape of a 3D gaussian, it gives sharper and just as smooth results as the tophat kernel, but usually, also a little bit noisier.

7

Set the drizzle drop size.

The smaller the drops, the sharper and noisier the result.

8

Choose the integration mode.

no drizzle - will integrate normally using data interpolation kernels like Lanczos-3.

drizzle - will use drizzle integration, see the tooltip at **drizzle integration kernel** for an explanation on when to use drizzle.

bayer drizzle - will use bayer drizzle integration. This is only applicable if the light frames consist of bayer CFA data. Bayer drizzle is a replacement for using a debayer algorithm like bilinear or Adaptive Airy Disc. In this case, no debayering will be done, only the CFA pixels are used to drizzle to the target pixelgrid. Any setting for the debayer algorithm in 0) RAW/FITS will be ignored in this case when the data is loaded for integration.

A recommended setting for bayer drizzle is to leave the scale at 1.0, set the droplet size to 2-2.5 pixels and use the tophat kernel. This will emulate the result of regular debayering but possibly with a sharper and smoother result. It will be very unlikely however, to improve on the quality and sharpness of the Adaptive Airy Disc algorithm for debayering. Using bayer drizzle can be sharper but will most likely be a lot noisier as well.

9

Set the scale of the integration.

Larger than 1 increases resolution, less than 1 reduces resolution.

You can use this together with drizzle: for instance, set scale to 2 and the droplet size to 0.5. This probably equals drizzle 2x known from other applications.

You can also use this without drizzle, just to upscale or downscale the integration.

For instance, if you are making a big mosaic, it can be useful to integrate the first time to a small scale, so it is finished more quickly and you can verify if registration and normalization are working correctly, before making an integration on the desired scale.