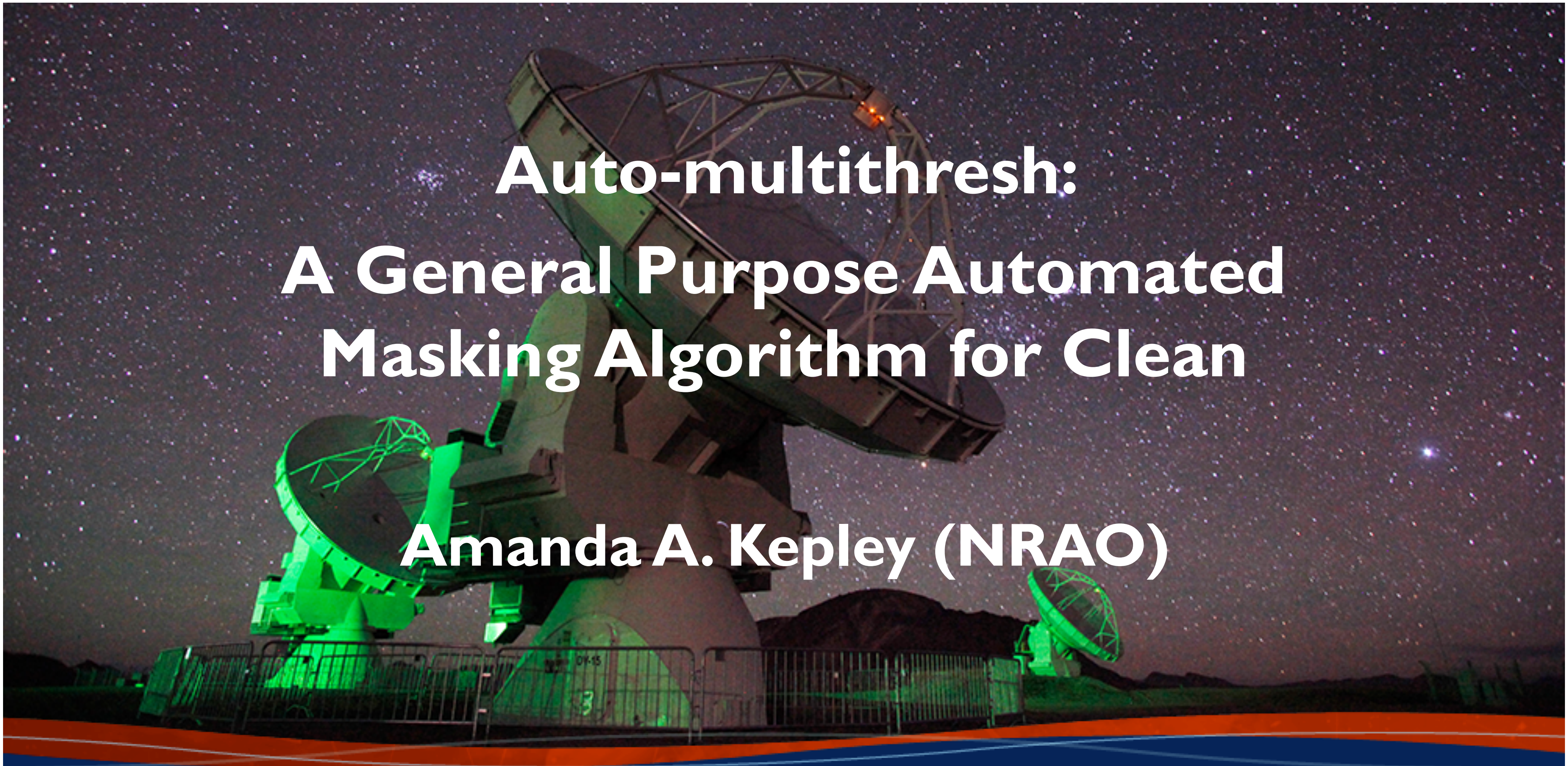


Auto-multithresh: A General Purpose Automated Masking Algorithm for Clean

Amanda A. Kepley (NRAO)

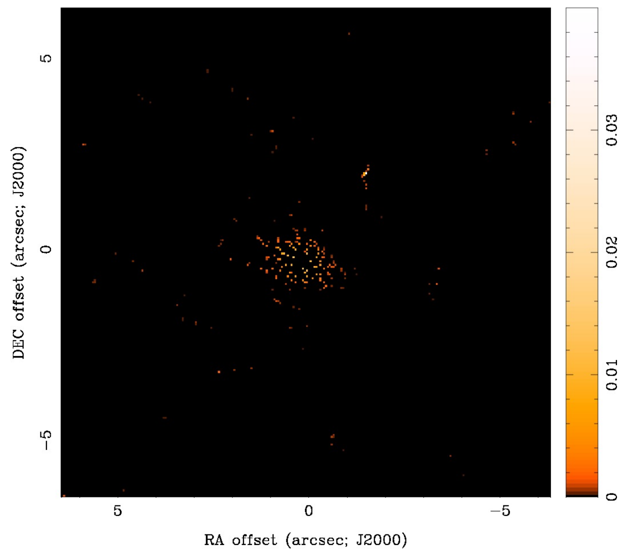
Takahiro Tsutsumi, **Ilsang Yoon**, **Remy Indebetouw**, **Crystal Brogan**, Brian Mason, Jennifer Donovan Meyer (NRAO)

With vital assists from Urvashi Rau, Steve Myers, and Claire Chandler (NRAO).

Clean is used to remove artifacts from discrete sampling of an interferometer.

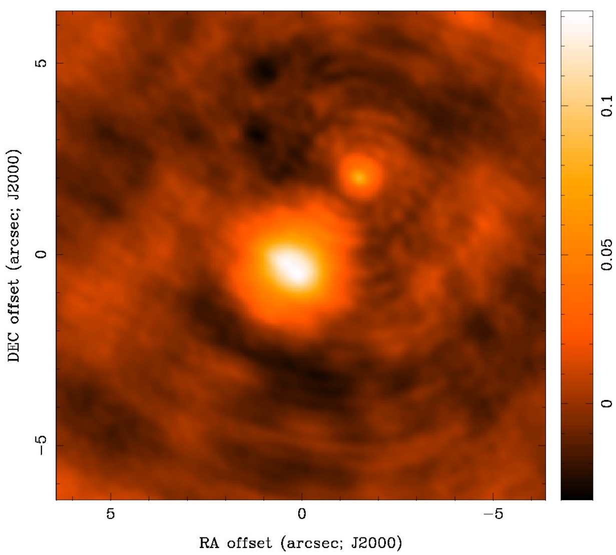
Example courtesy of D. Wilner

Model



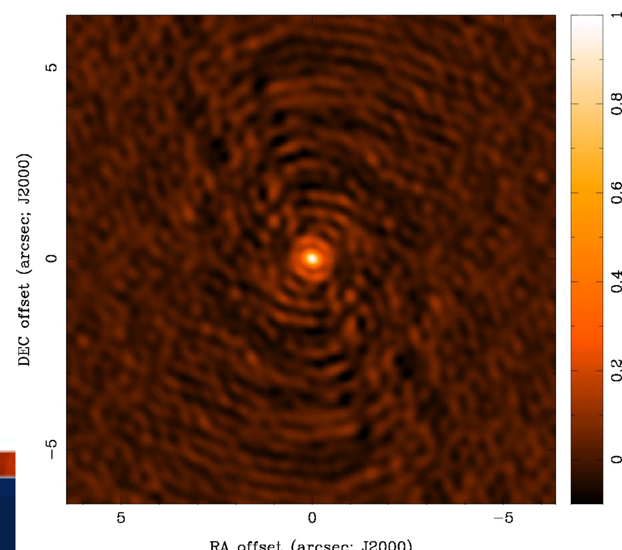
Initial interferometer image

("dirty" image)



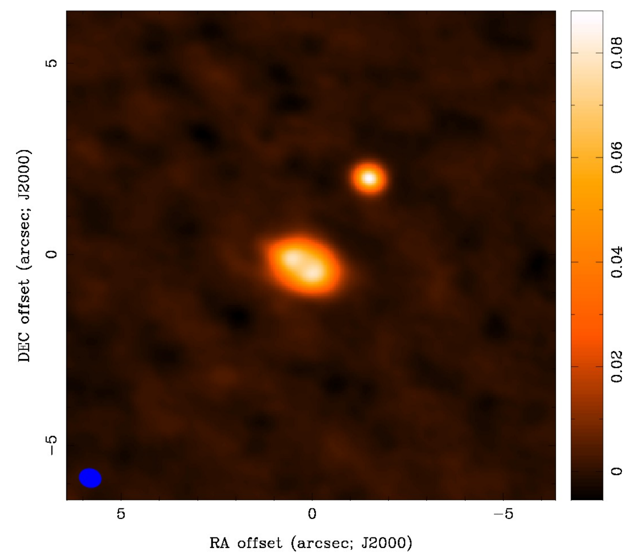
Shows grating artifacts from discrete sampling of the array.

PSF



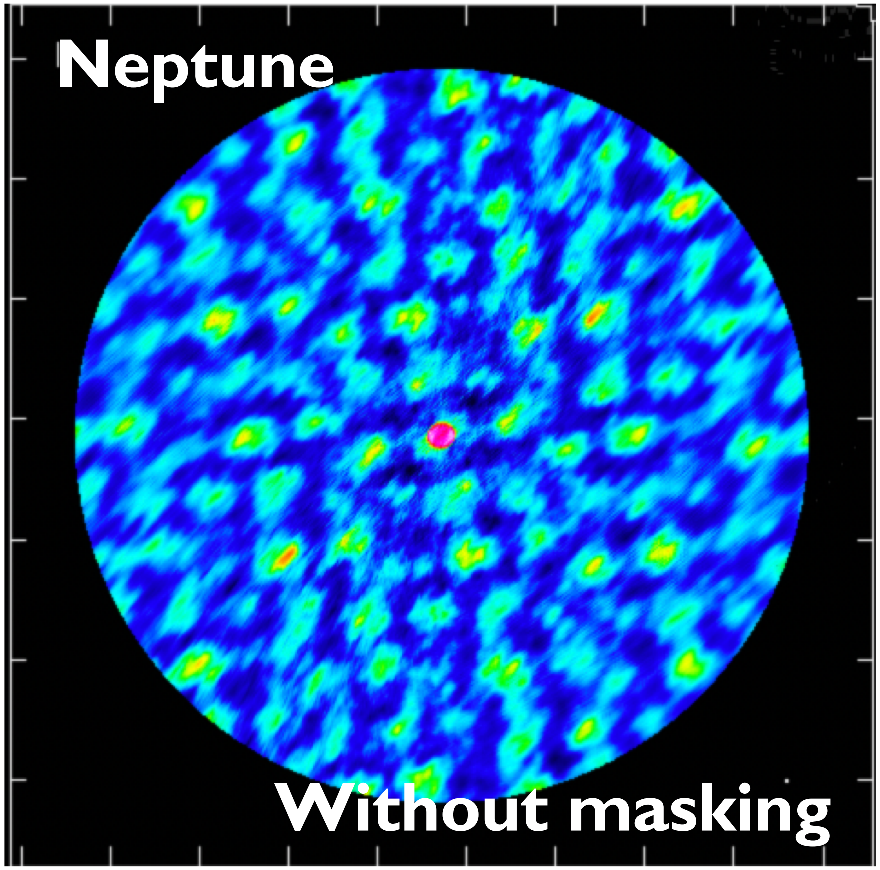
Final interferometer image

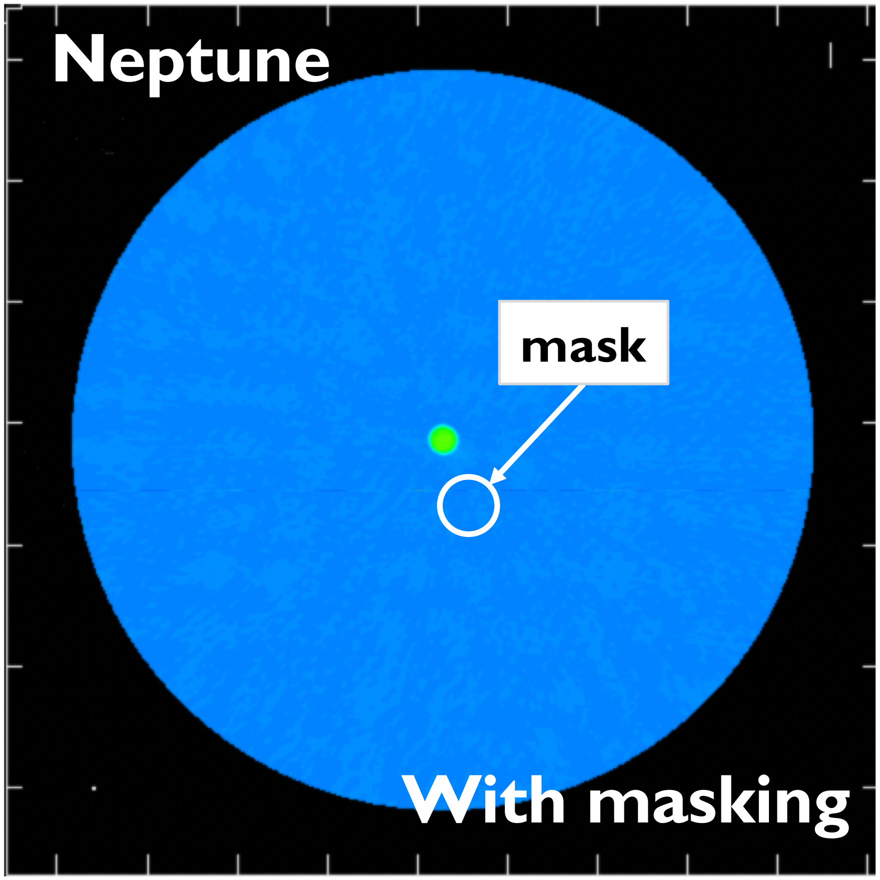
("clean" image)



Model convolved with a Gaussian beam plus the residuals.

Typically the user restricts what features are modeled because it is under-constrained.





Same color scale!

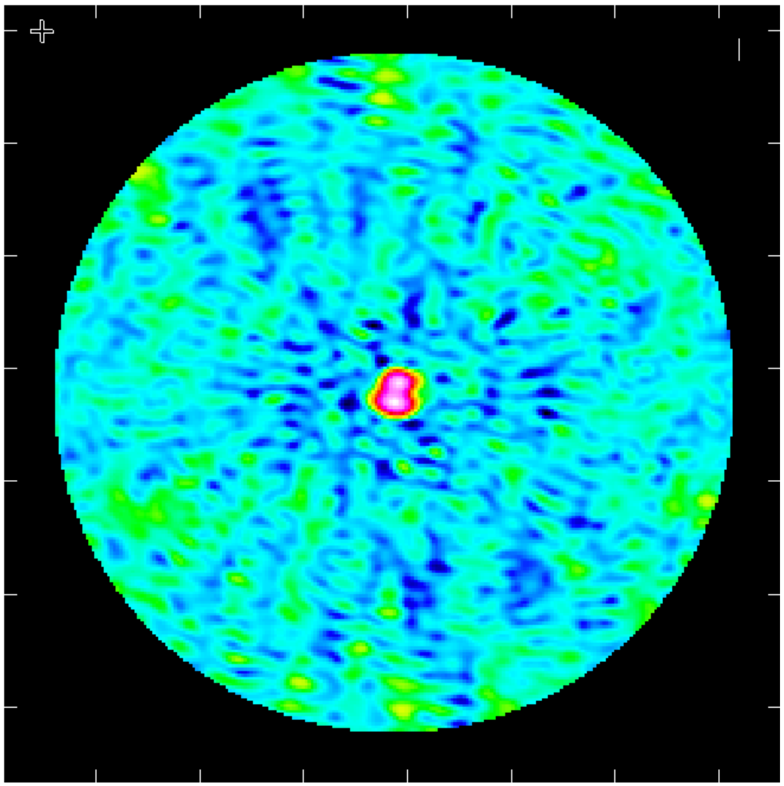
Not using masks can also lead to distortions of the noise in the image (clean bias) or resulting source structure.

The ALMA imaging pipeline needed to automatically mask images during clean.

- Large data volumes produced by ALMA and other modern interferometers require automated imaging pipelines.
- **Goal:** easily understandable algorithm for the ALMA imaging pipeline that closely mimics what an experienced manual imager would do and works on a large number of cases.
- **Note:** This is a conservative goal because the pipeline is largely conservative.
- The algorithm was developed and tested against the ALMA benchmark suite.
- It is current in production as part of the ALMA Imaging Pipeline starting with Cycle 5 (October 2017)

See T. Tsutsumi's poster (P12.15) for more info on implementation

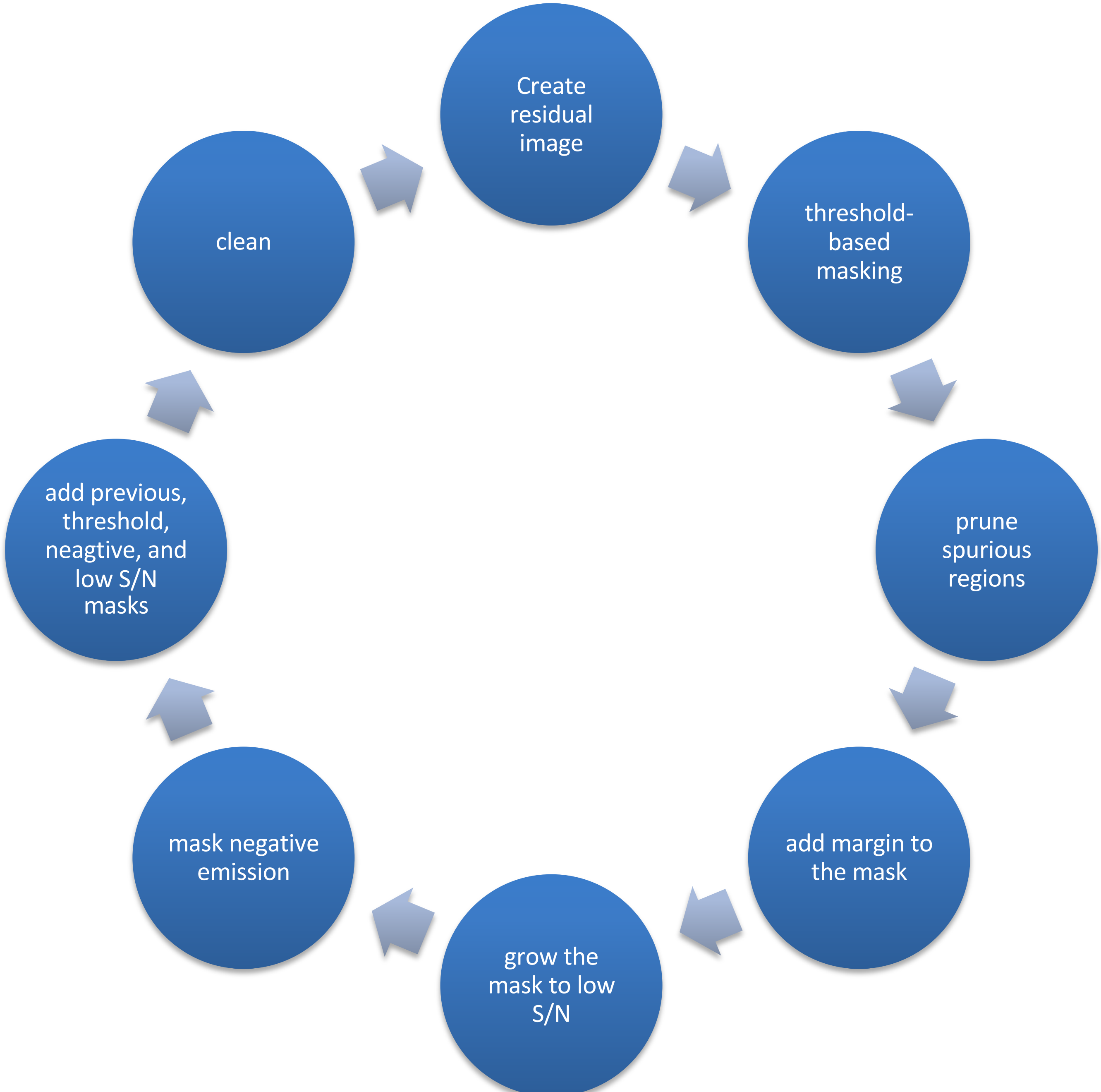
All input parameters are specified relative to the fundamental properties of the image.

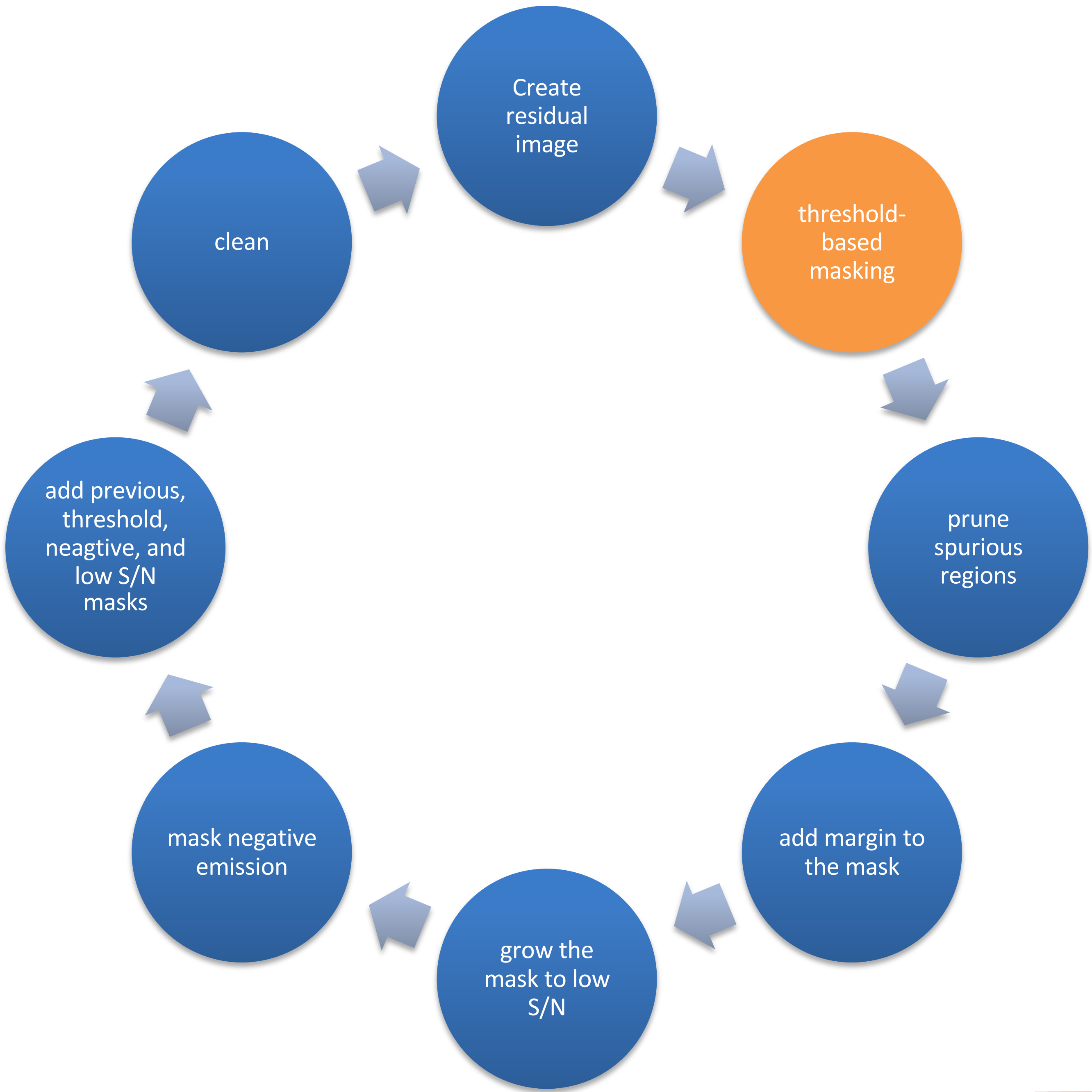


- resolution (i.e., beam)
- noise
- sidelobe level

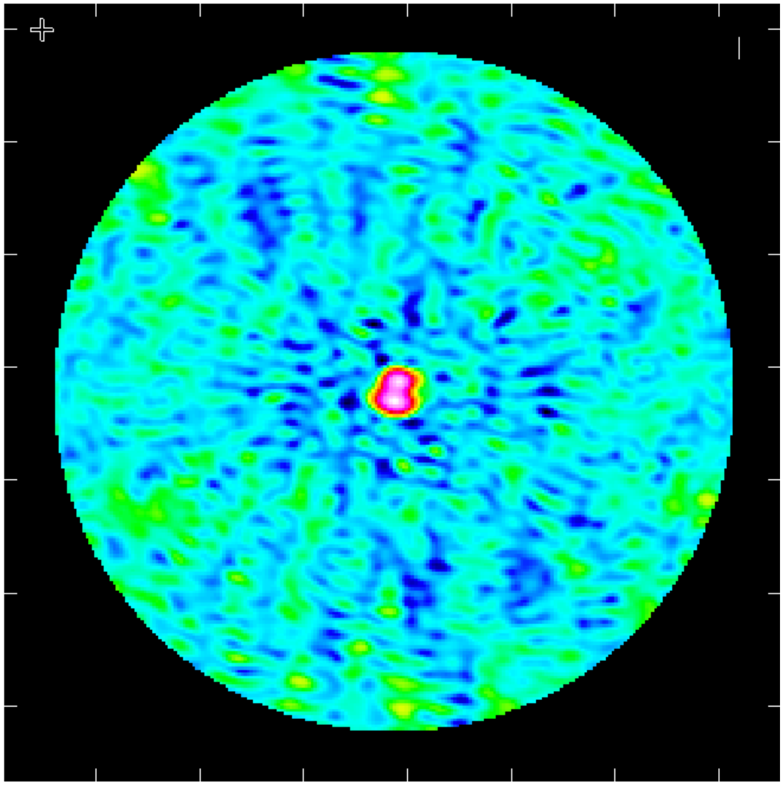
All steps are done once per major cycle and independently per channel.

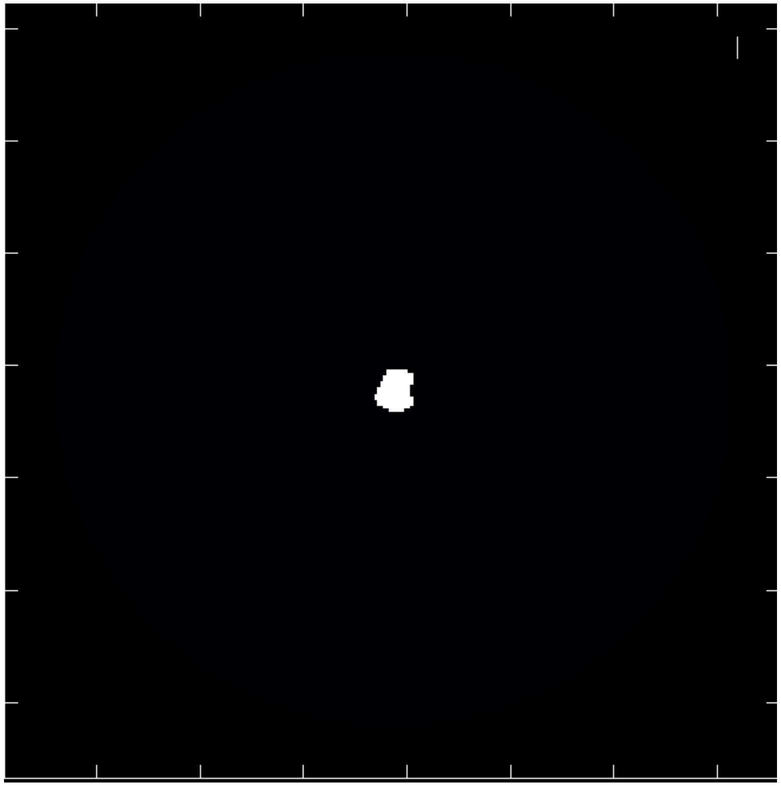
- The mask is updated every time a new residual image is created at the start of a minor cycle (=deconvolution).
- All steps to the algorithm are done on a per-channel basis.
- Continuum = 1 channel cube.


Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, neagtive, and low S/N masks
clean


Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, neagtive, and low S/N masks
clean

Mask above the n times the sidelobe level or m times the noise level, whichever is greater.



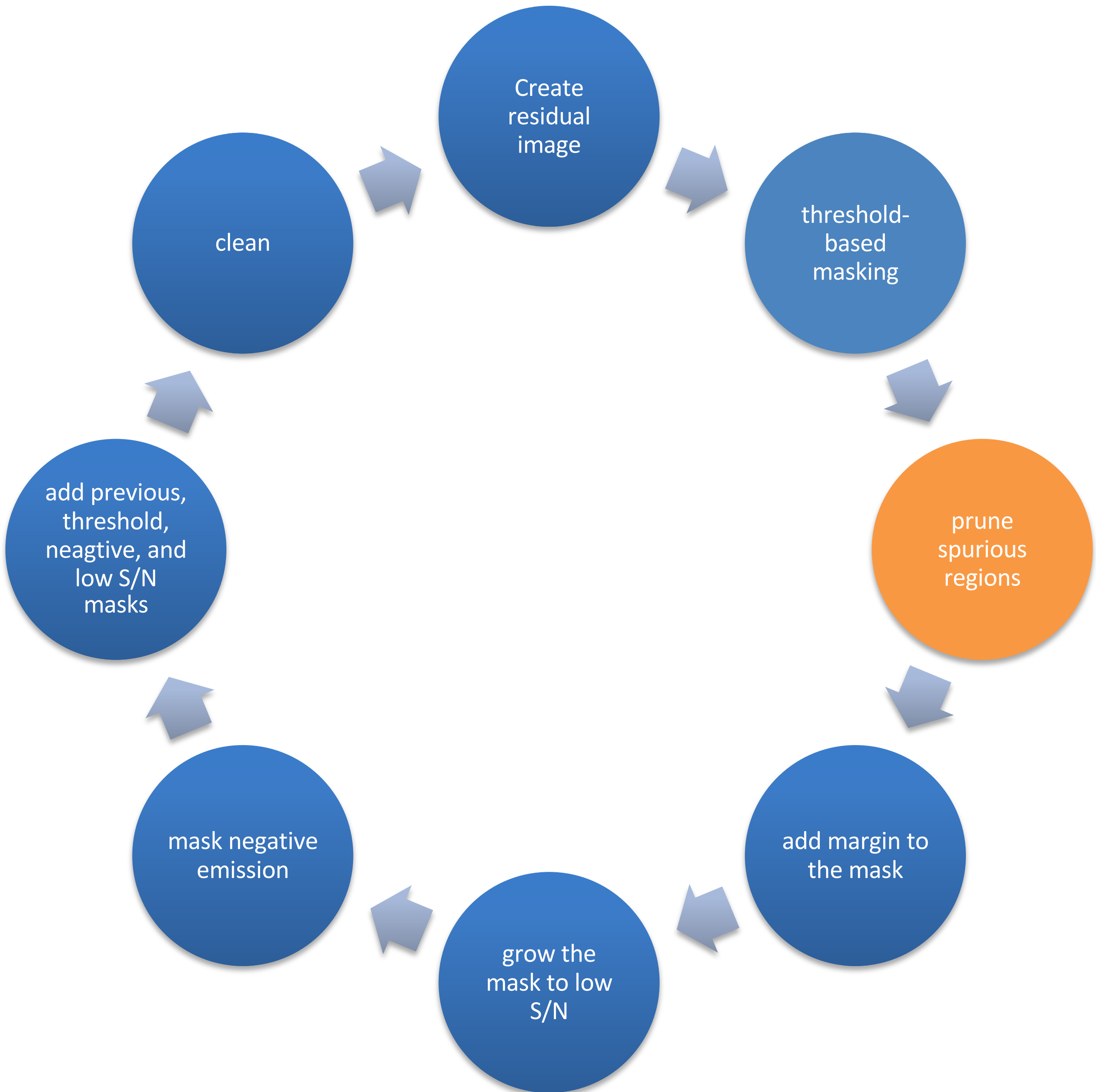


rms in residual = mad(residual) * 1.4826

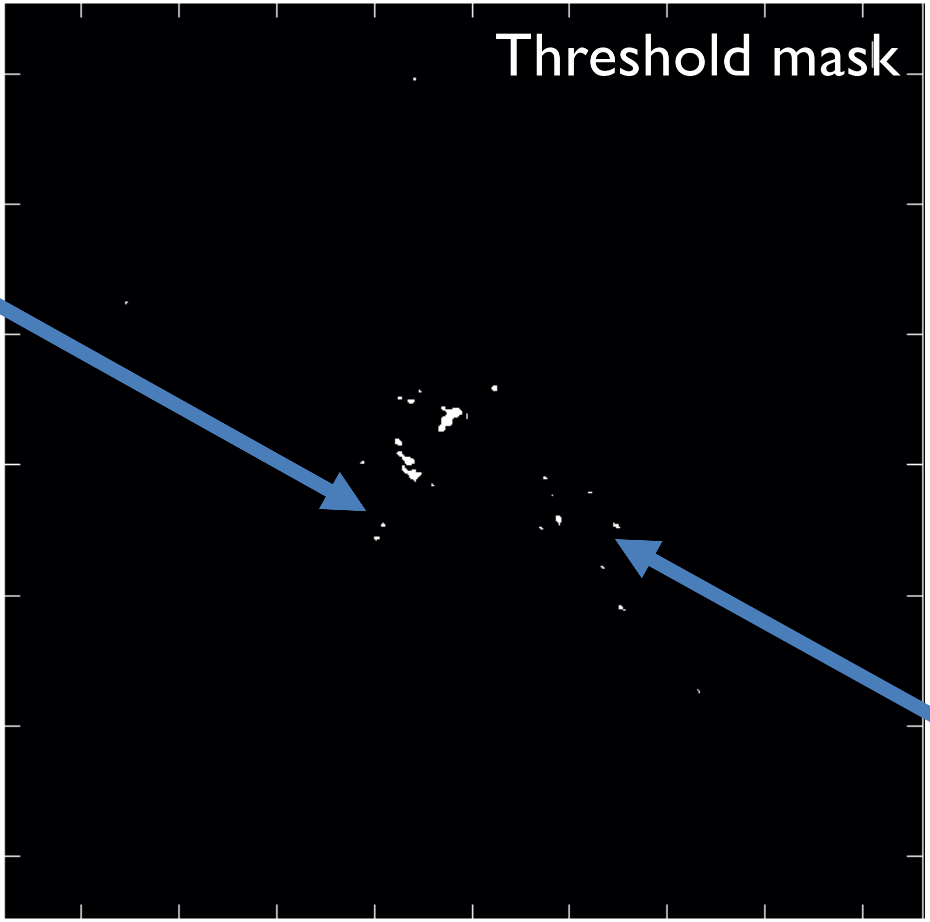
SidelobeThresholdValue = **sidelobeThreshold** * sidelobeLevel * peak in residual

NoiseThresholdValue = **noiseThreshold** * rms in residual

maskThreshold = max(SidelobeThresholdValue,NoiseThresholdValue)


Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, neagtive, and low S/N masks
clean

Prune the spurious regions smaller than the some fraction of the beam.

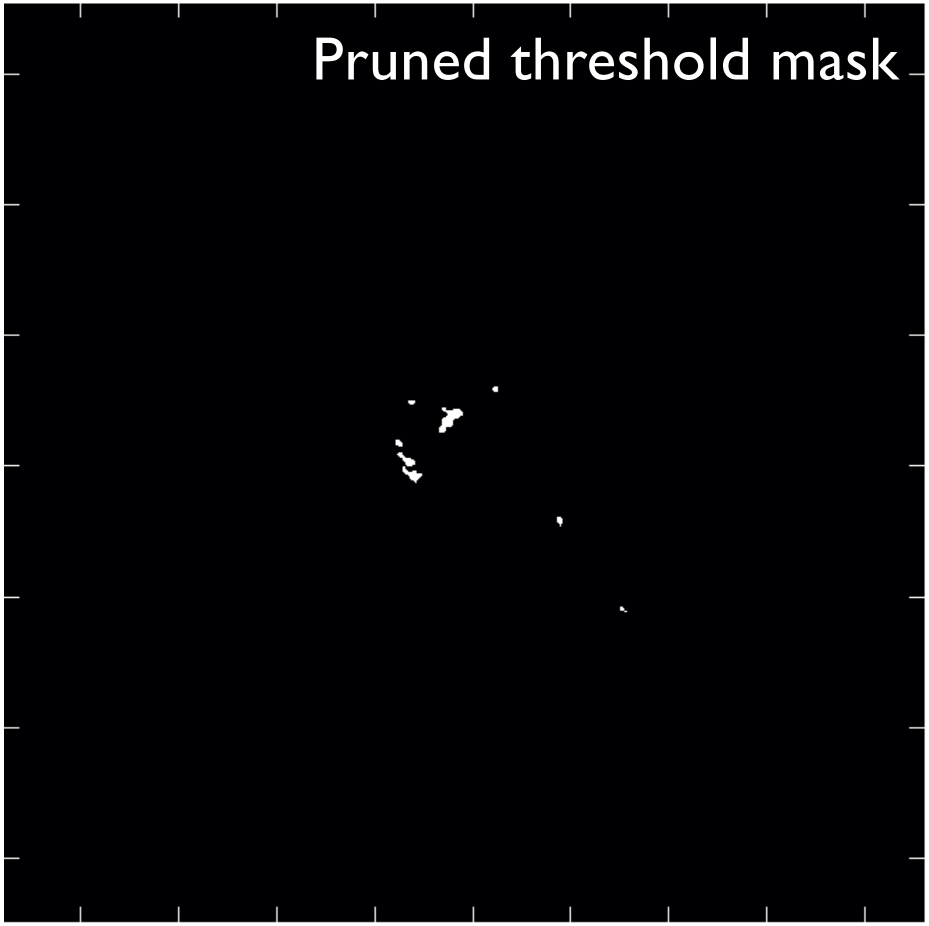


Spurious regions

Spurious regions

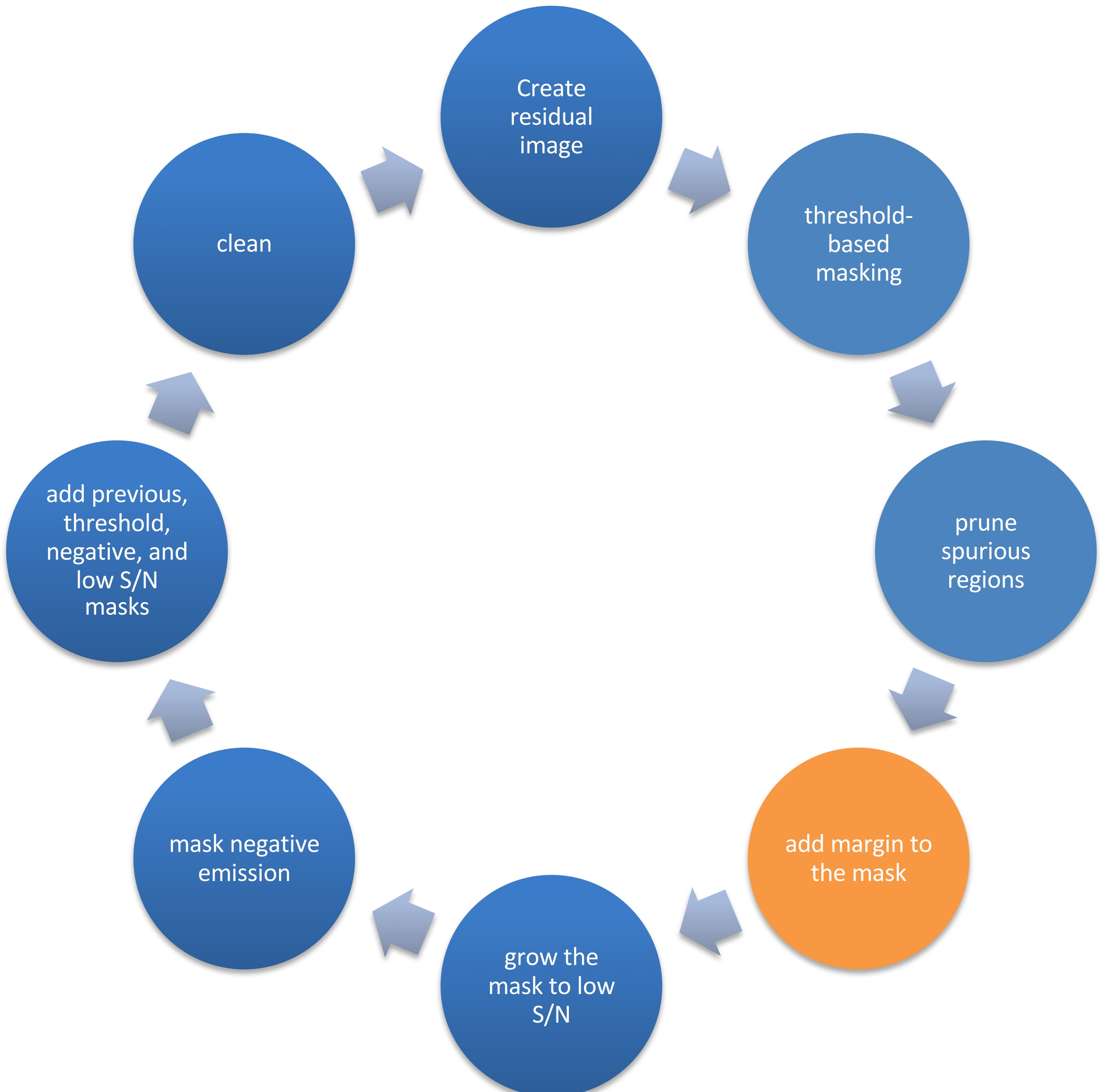
if (# of pixels in a region) < **minBeamFrac** * (pixels in beam),
remove region from mask

Prune the spurious regions smaller than the some fraction of the beam.

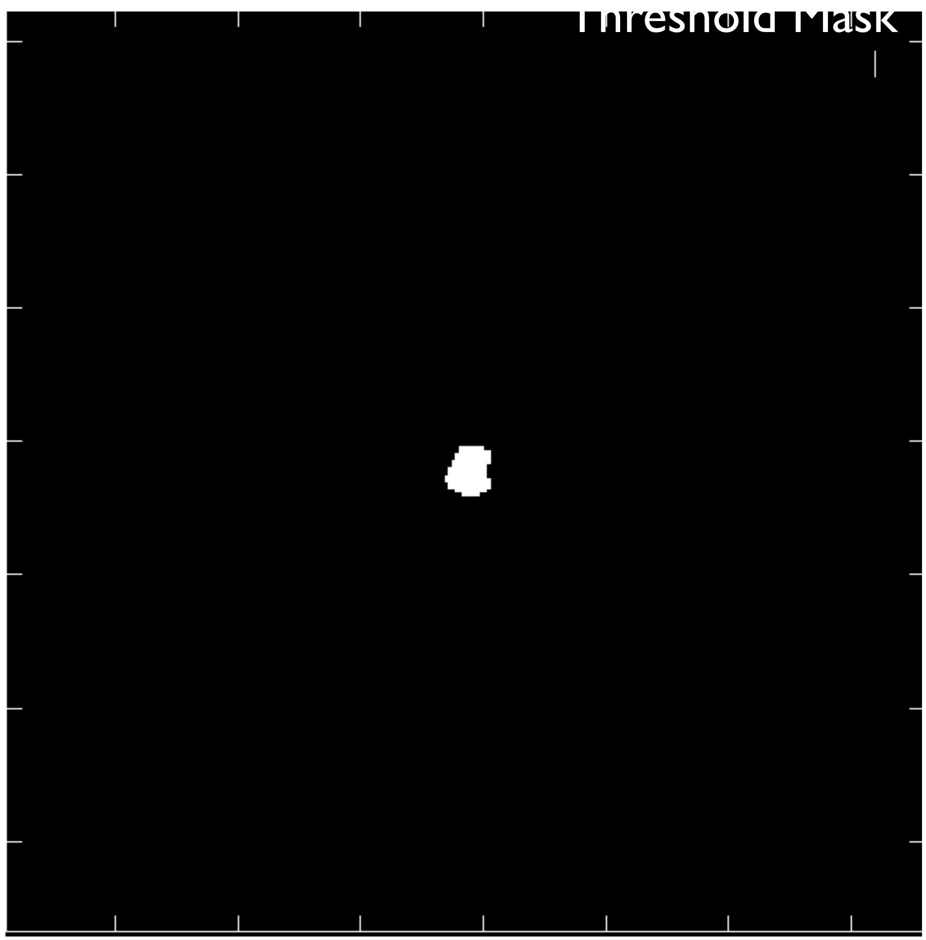


Testing has shown that the number of regions removed is ~equal to false detection rate for Gaussian noise.

if (# of pixels in a region) < **minBeamFrac** * (pixels in beam),
remove region from mask

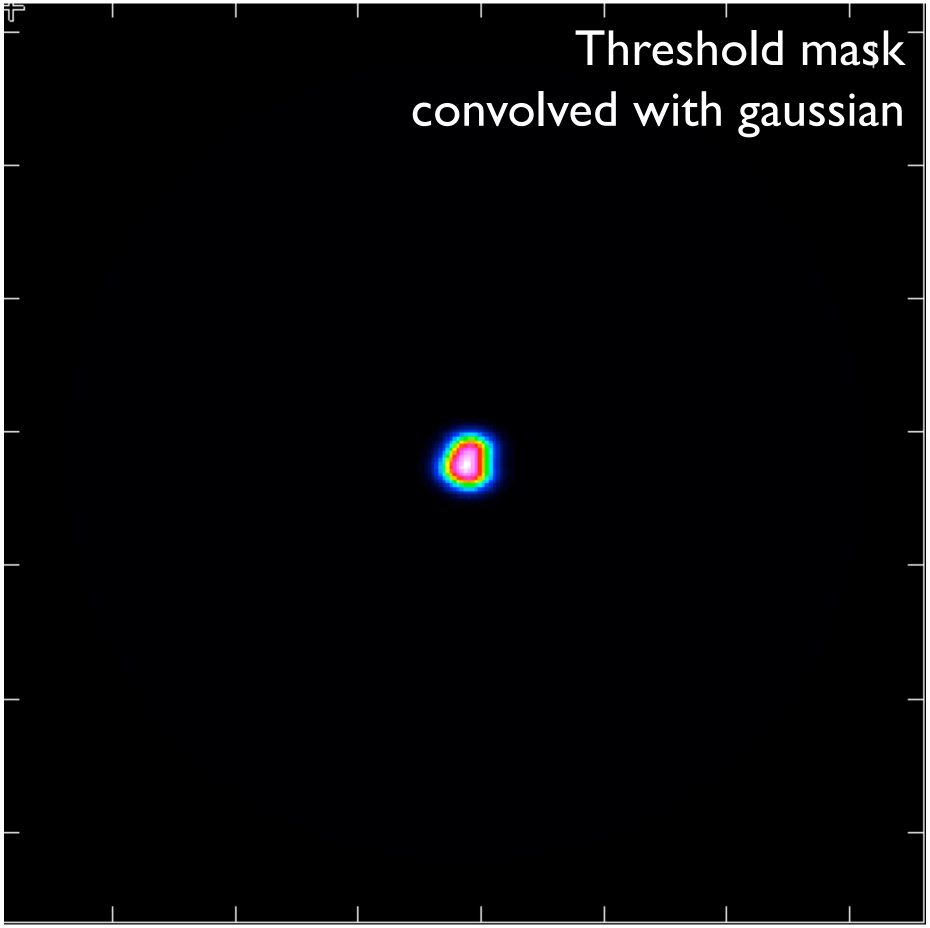

Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, negative, and low S/N masks
clean

Convolve the mask with a Gaussian n times the size of the beam to create a margin.



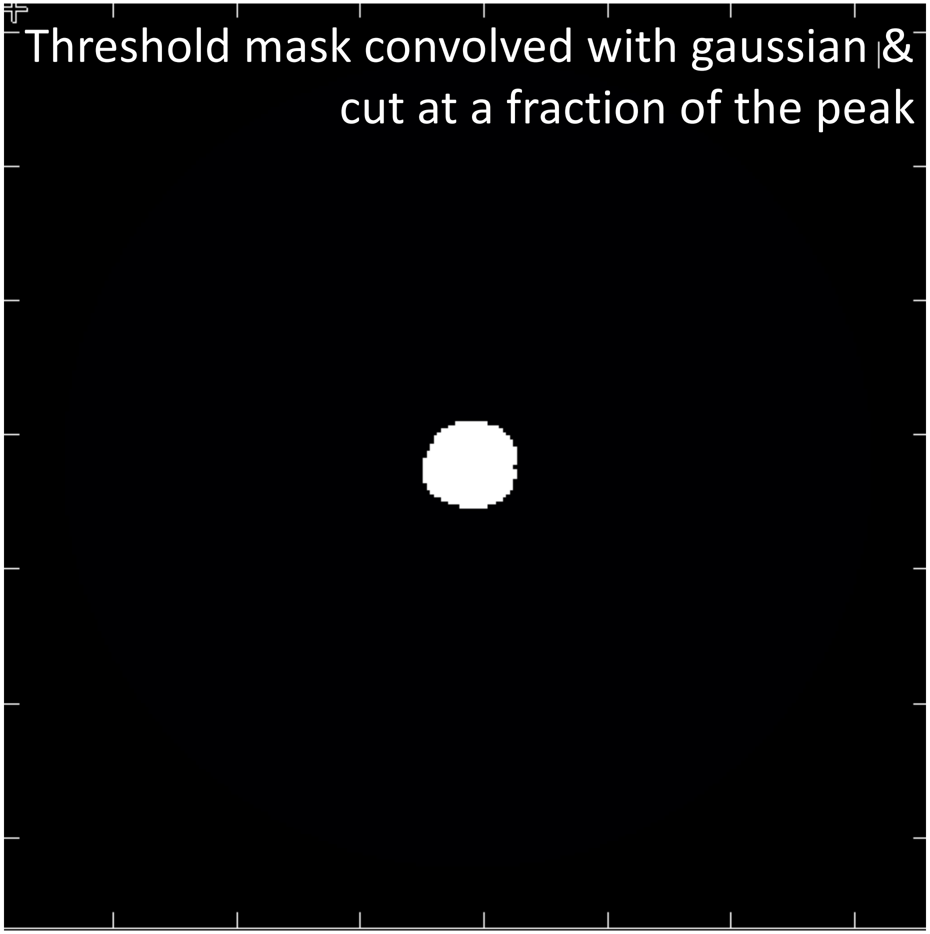
kernel = **smoothFactor** * beam(min,maj)
convolvedMask = convolve(thresholdMask,kernel)
Mask if value > max(convolvedMask)***cutThreshold**

Convolve the mask with a Gaussian n times the size of the beam to create a margin.

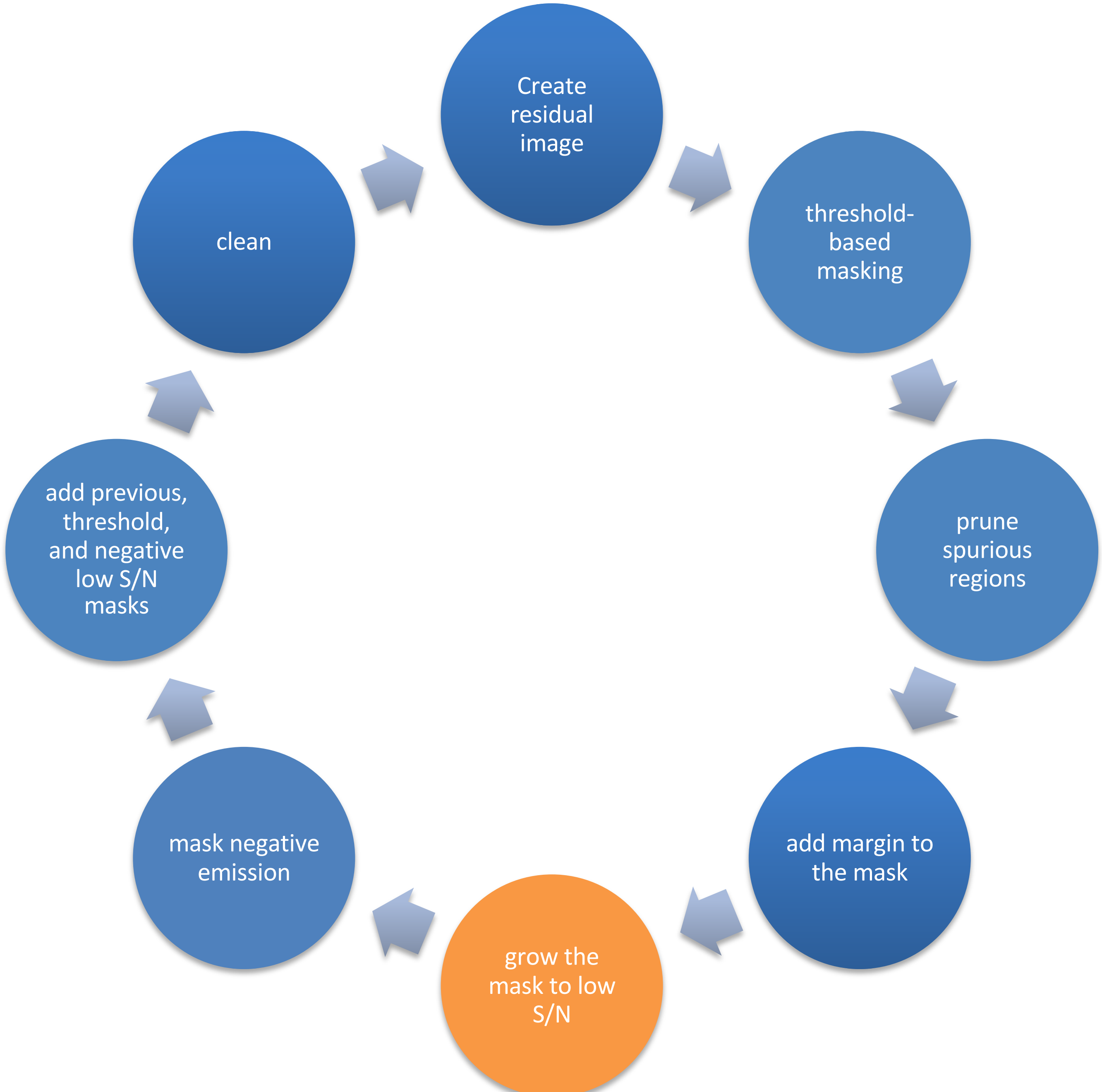


kernel = **smoothFactor** * beam
convolvedMask = convolve(thresholdMask,kernel)
Mask if value > max(convolvedMask)***cutThreshold**

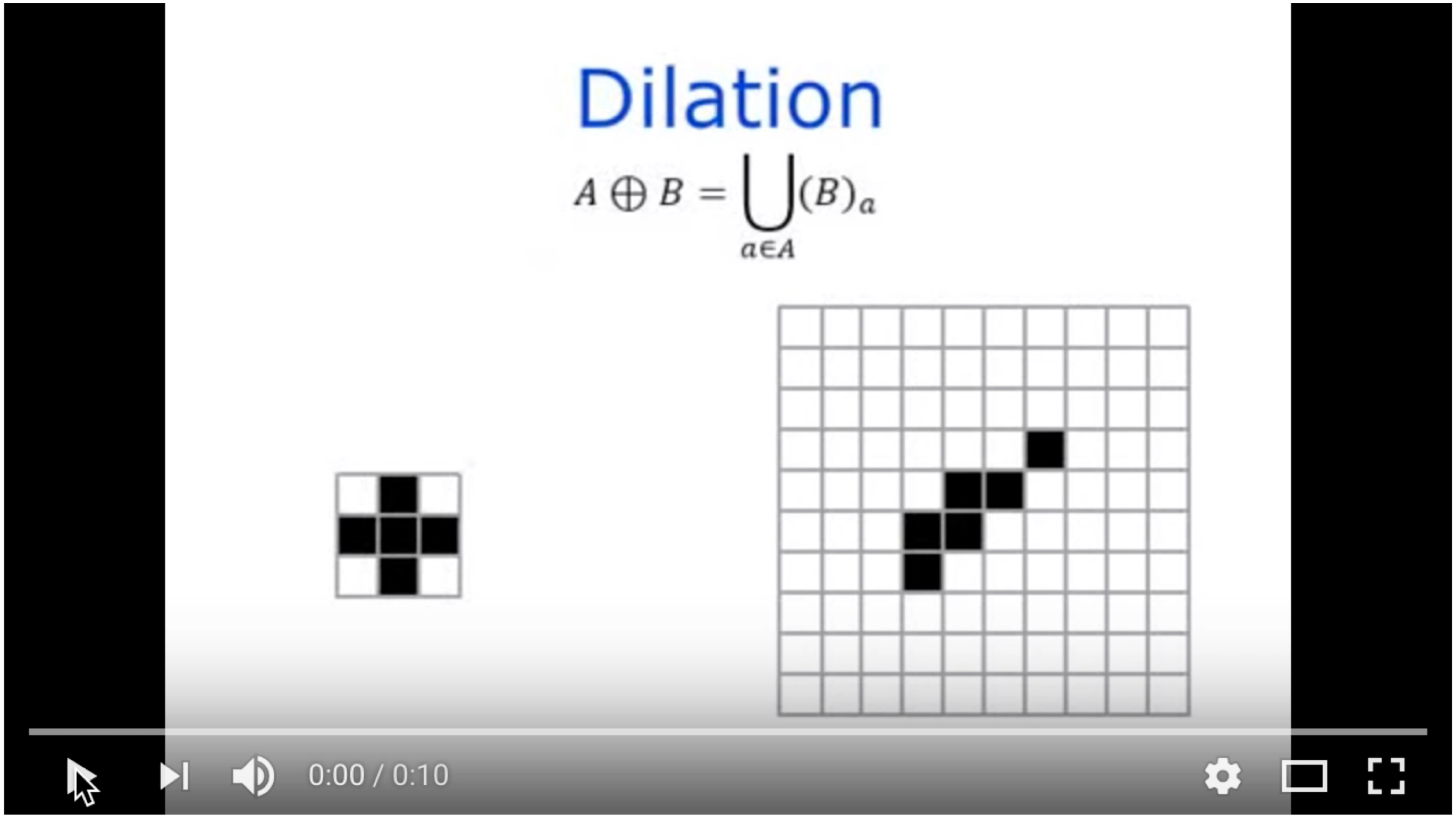
Convolve the mask with a Gaussian n times the size of the beam to create a margin.



kernel = **smoothFactor** * beam
convolvedMask = convolve(thresholdMask,kernel)
Mask if value > max(convolvedMask)***cutThreshold**


Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, and negative low S/N masks
clean

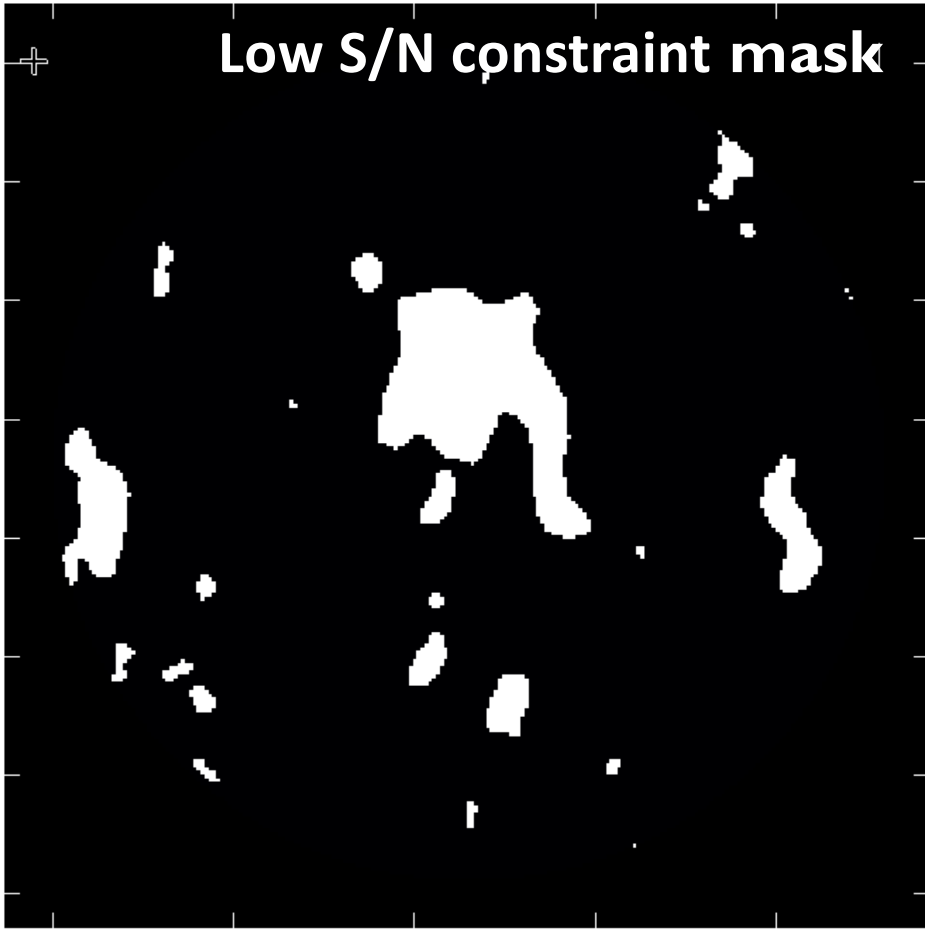
Binary dilation is a convolution-like method to expand high S/N masks into a low S/N regions.



- Same technique used in molecular cloud ID algorithm cprops (Rosolowsky & Leroy 2006)
- Can repeat multiple times
- Can grow one region out to a boundary.

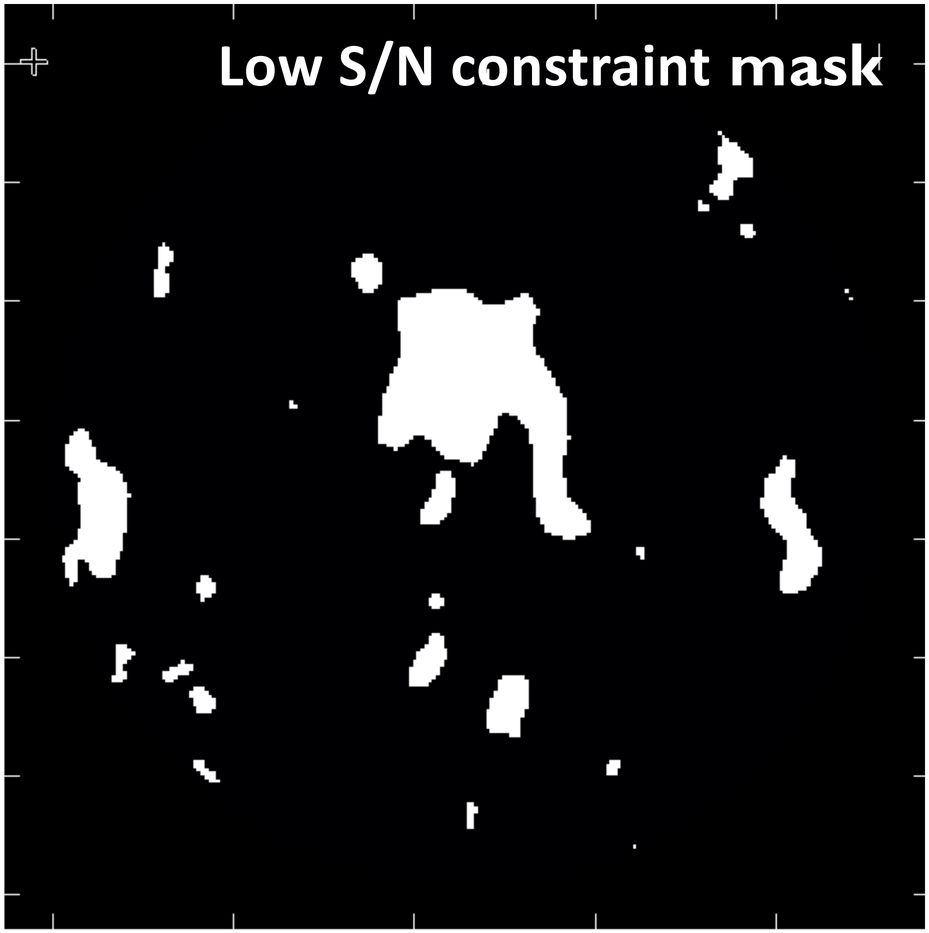
Sofia Praça (https://youtu.be/W5x-z63JFQY)

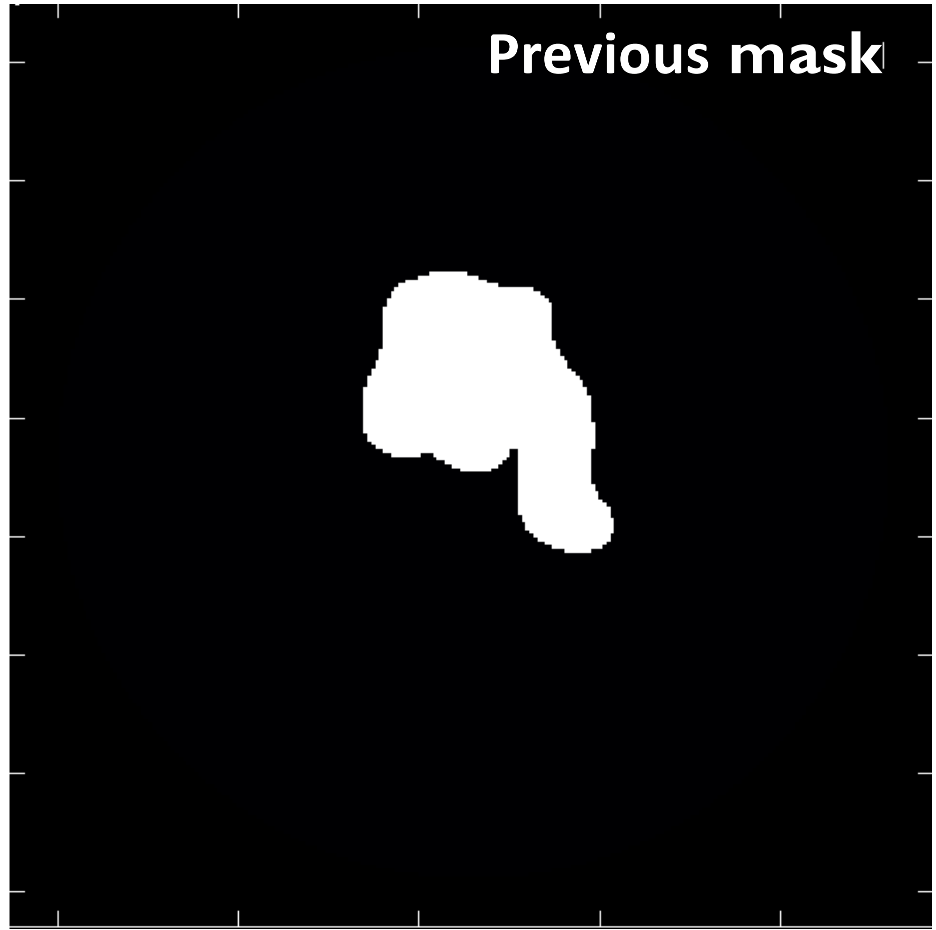
Binary dilation allows us grow the mask down into low S/N.



lowNoiseThresholdValue = **lowNoiseThreshold** * rms in residual
constraintMaskThreshold = max(sidelobeThresholdValue,lowNoiseThresholdValue)

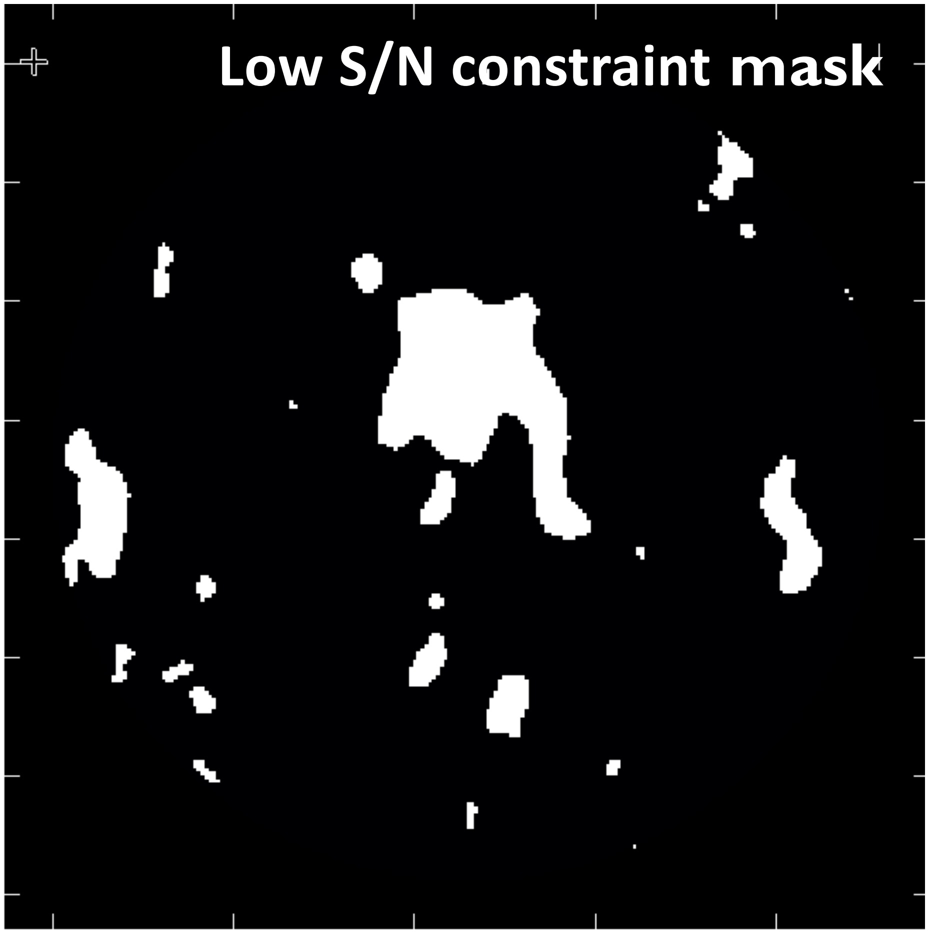
Binary dilation allows us grow the mask down into low S/N.





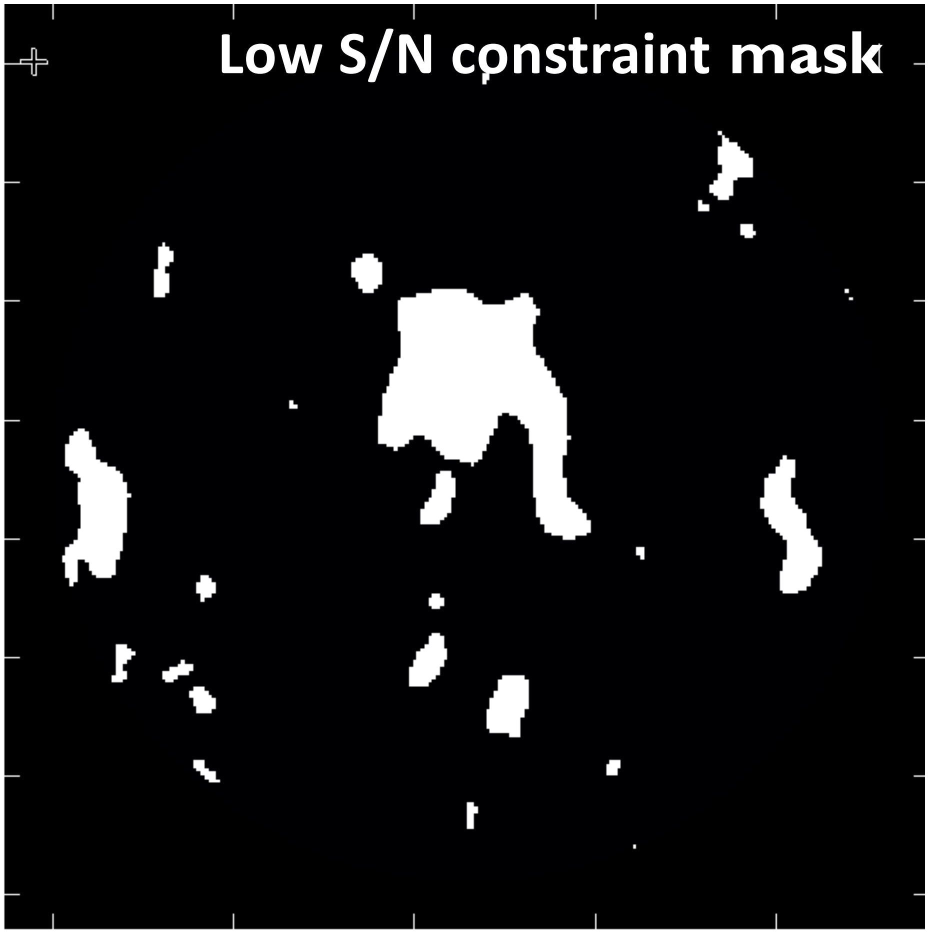
dilatedMask = binaryDilation(previousMask, constraintMask)

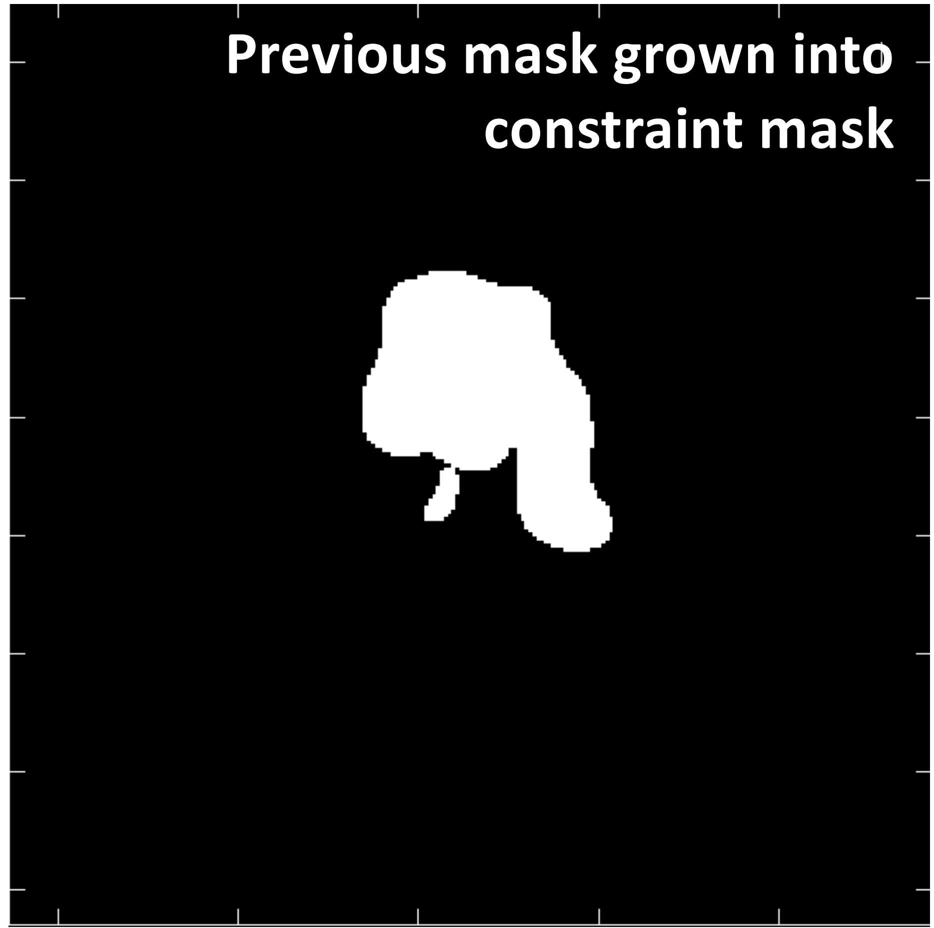
Binary dilation allows us grow the mask down into low S/N.



dilatedMask = binaryDilation(previousMask, constraintMask)

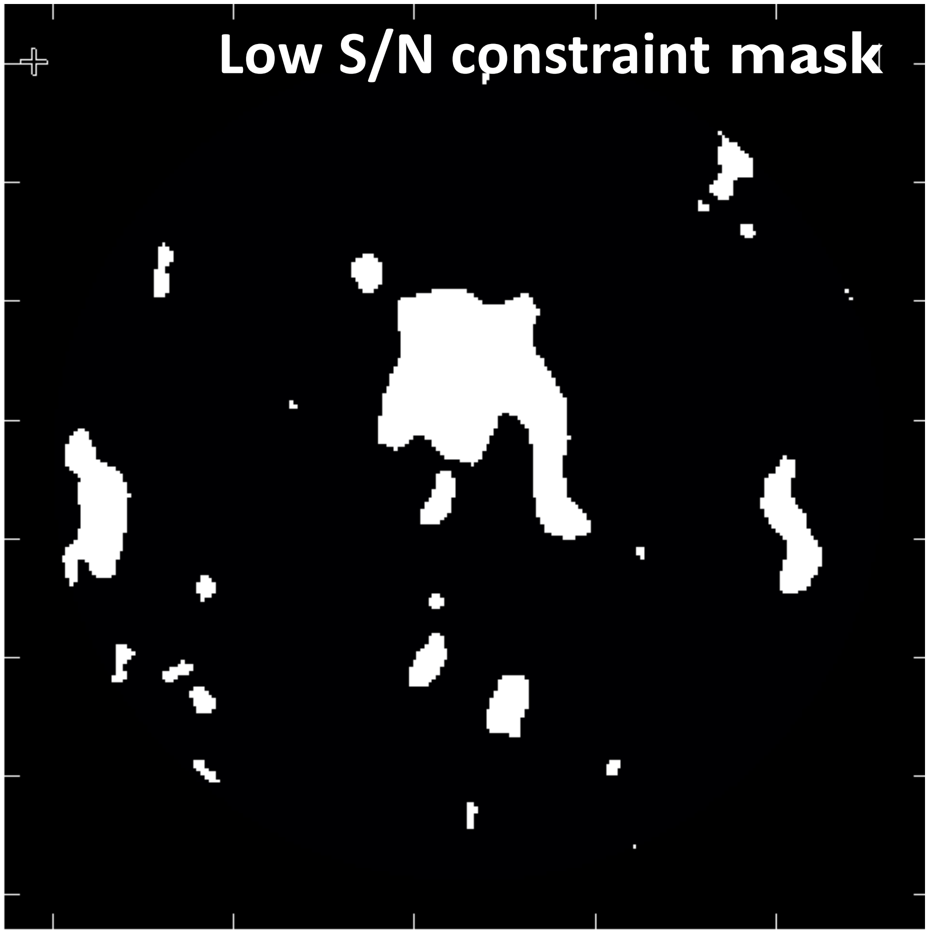
Binary dilation allows us grow the mask down into low S/N.

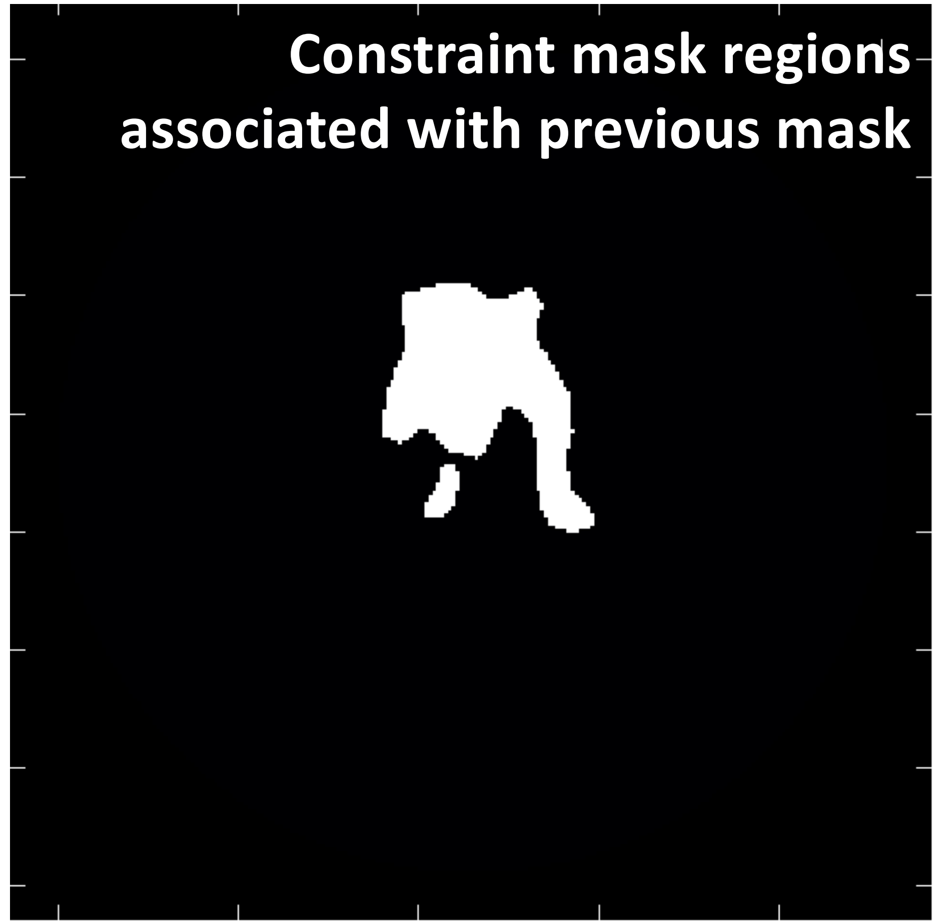




lowNoiseMask = dilatedMask * constraintMask

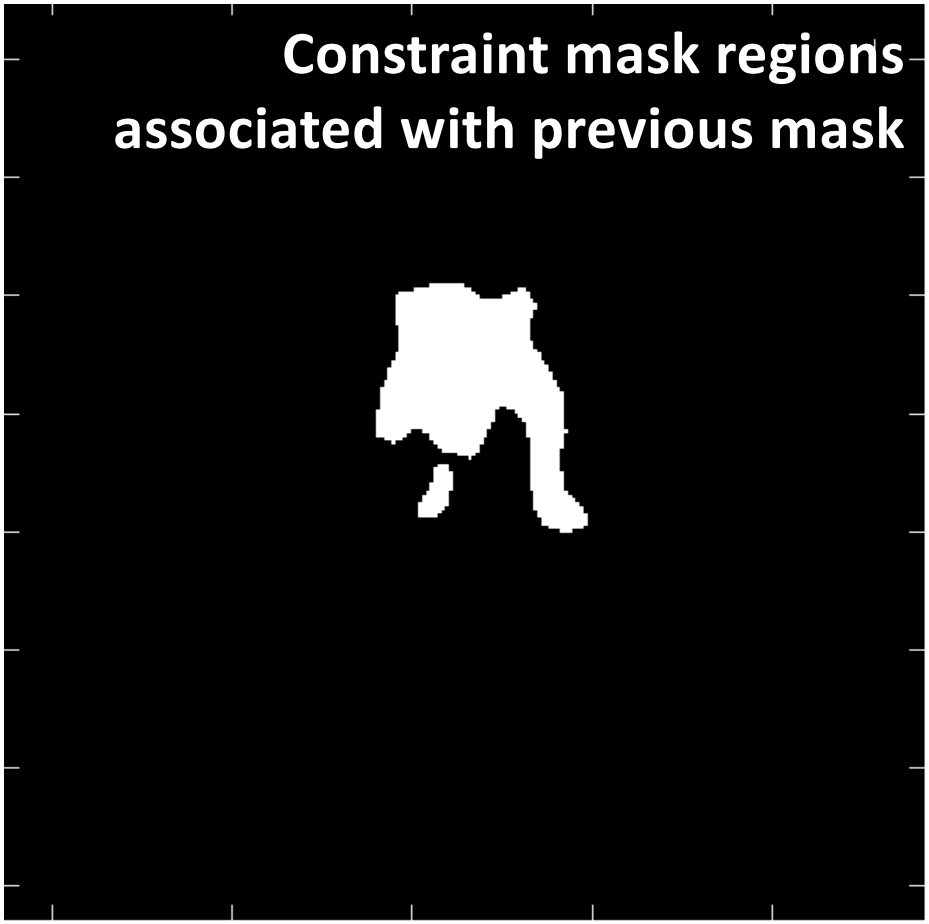
Binary dilation allows us grow the mask down into low S/N.

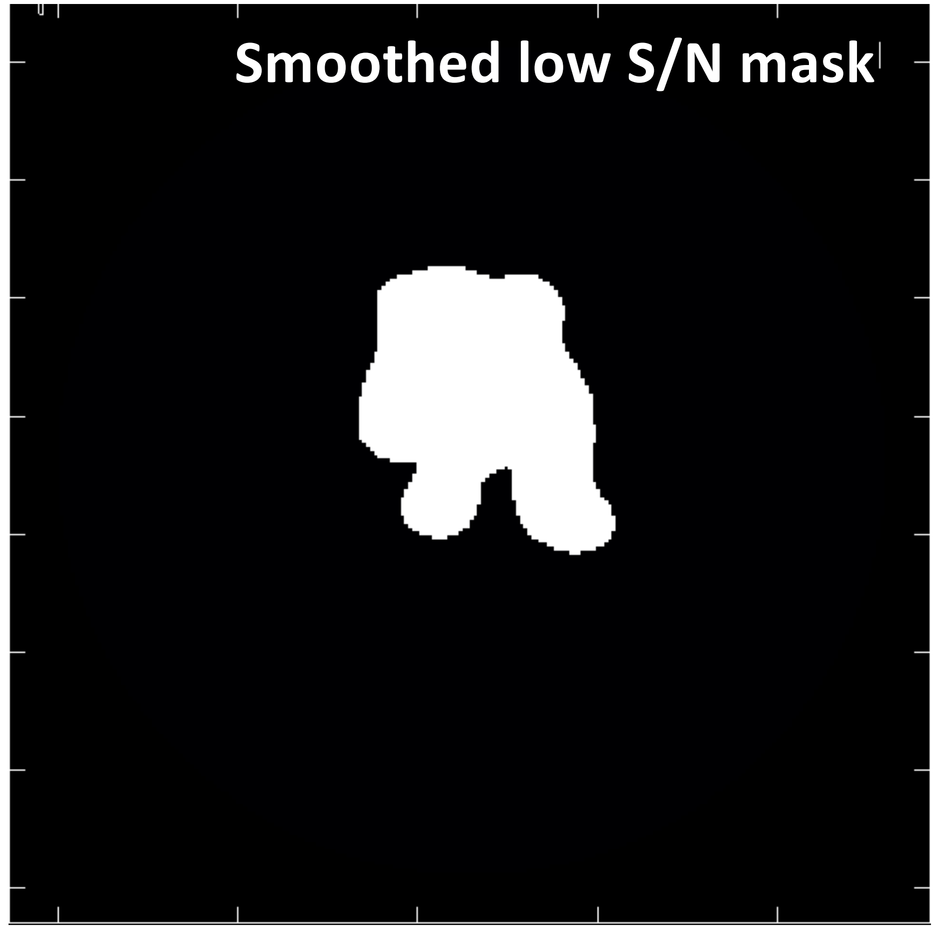




lowNoiseMask = dilatedMask * constraintMask

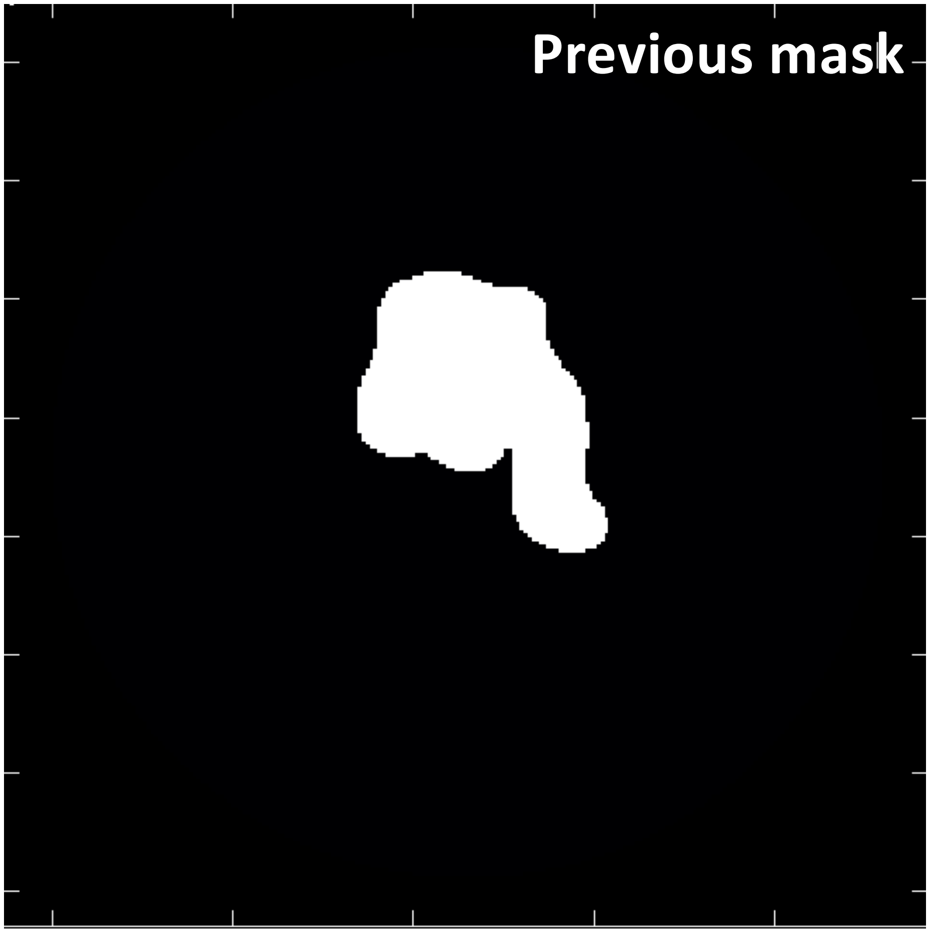
Binary dilation allows us grow the mask down into low S/N.

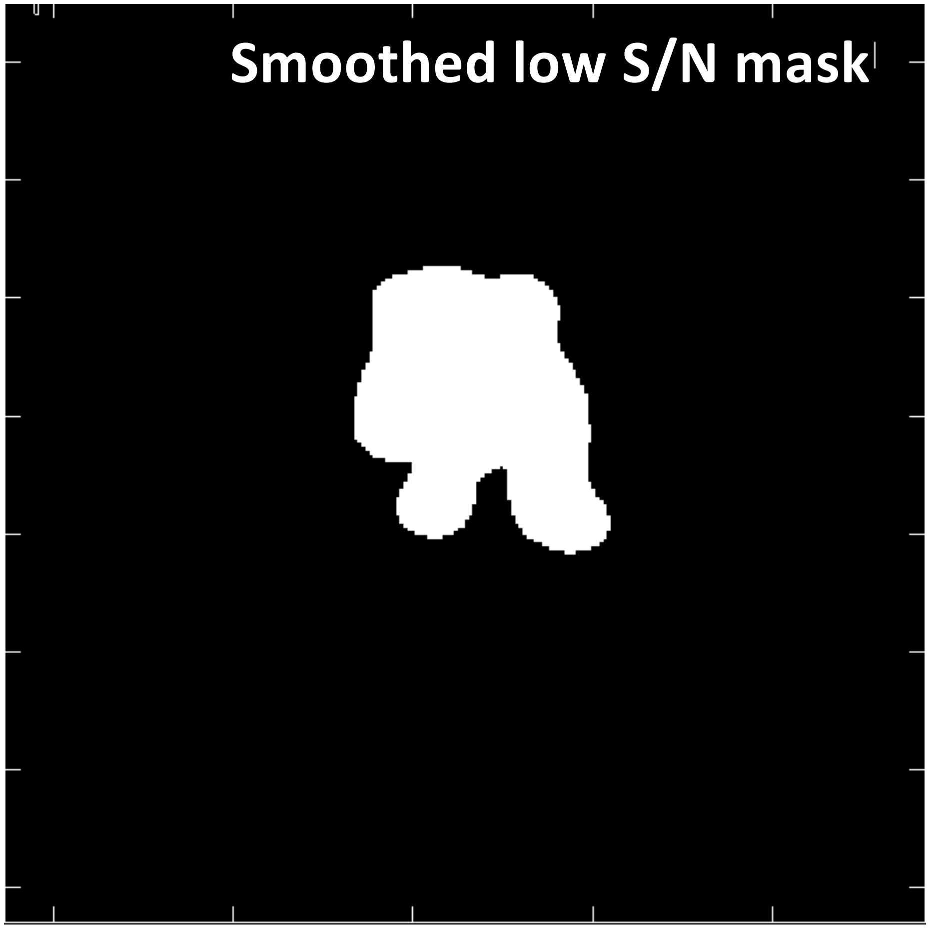


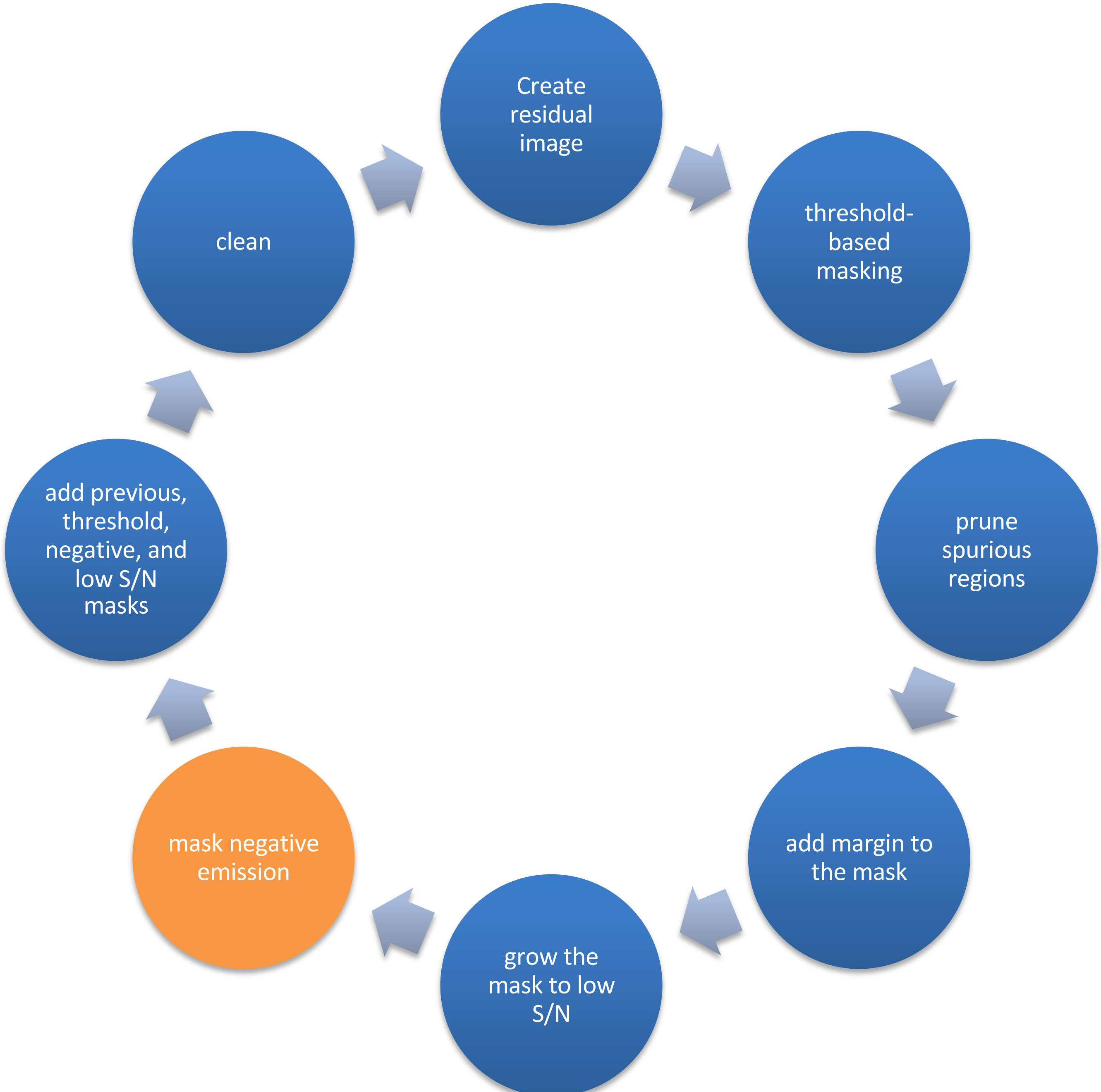


Prune, smooth, and cut lowNoiseMask

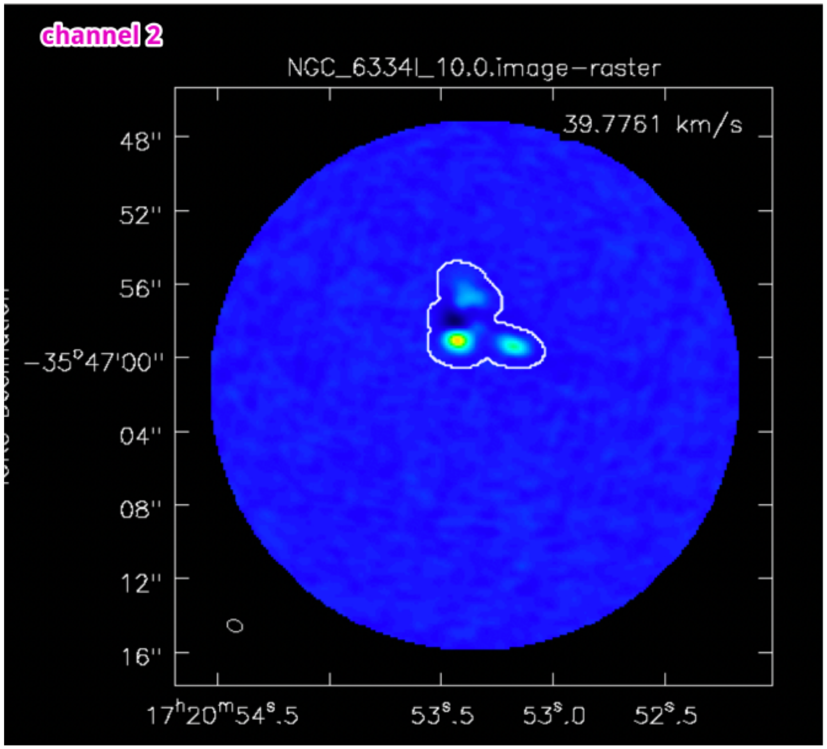
Binary dilation allows us grow the mask down into low S/N.



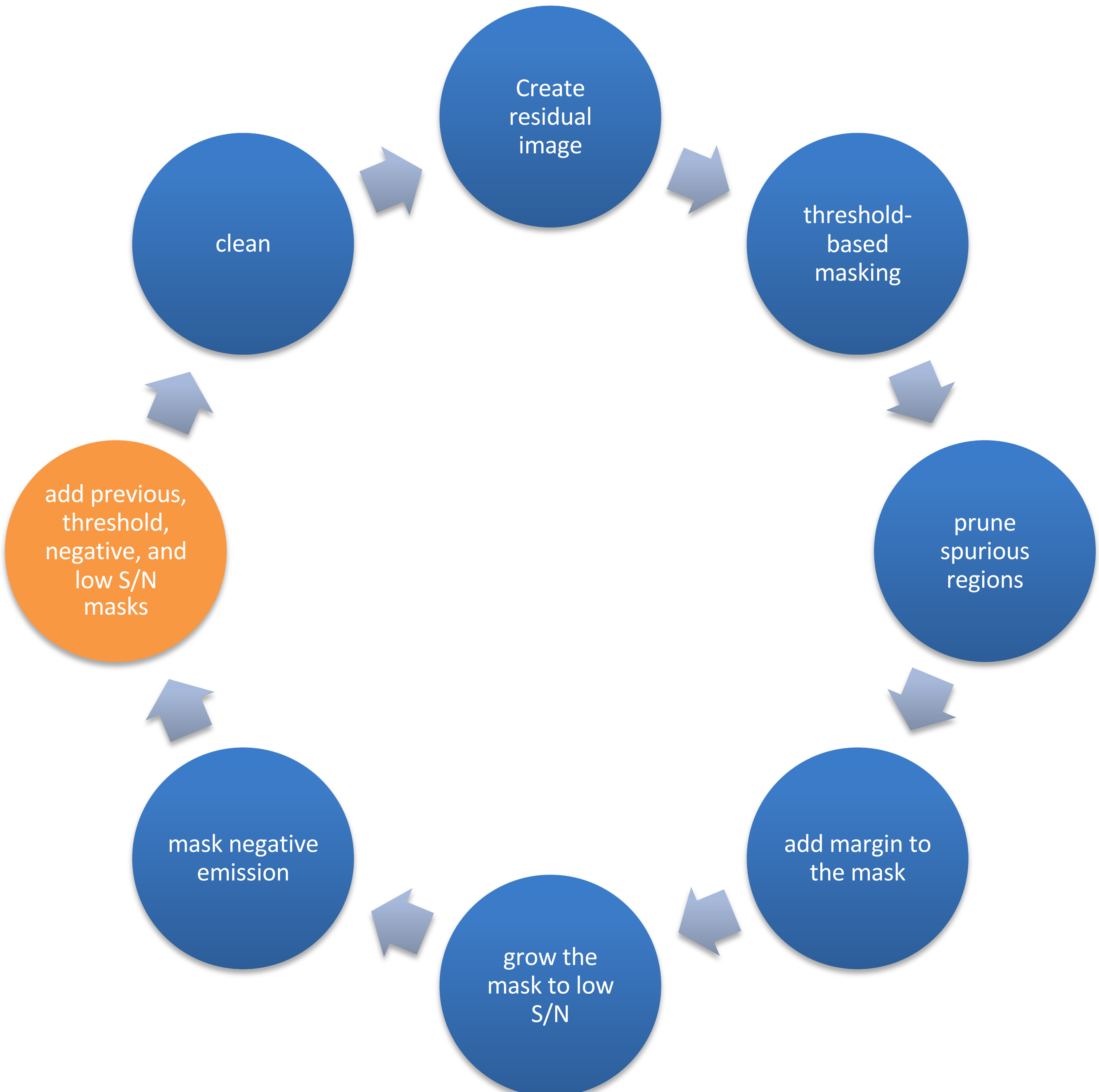


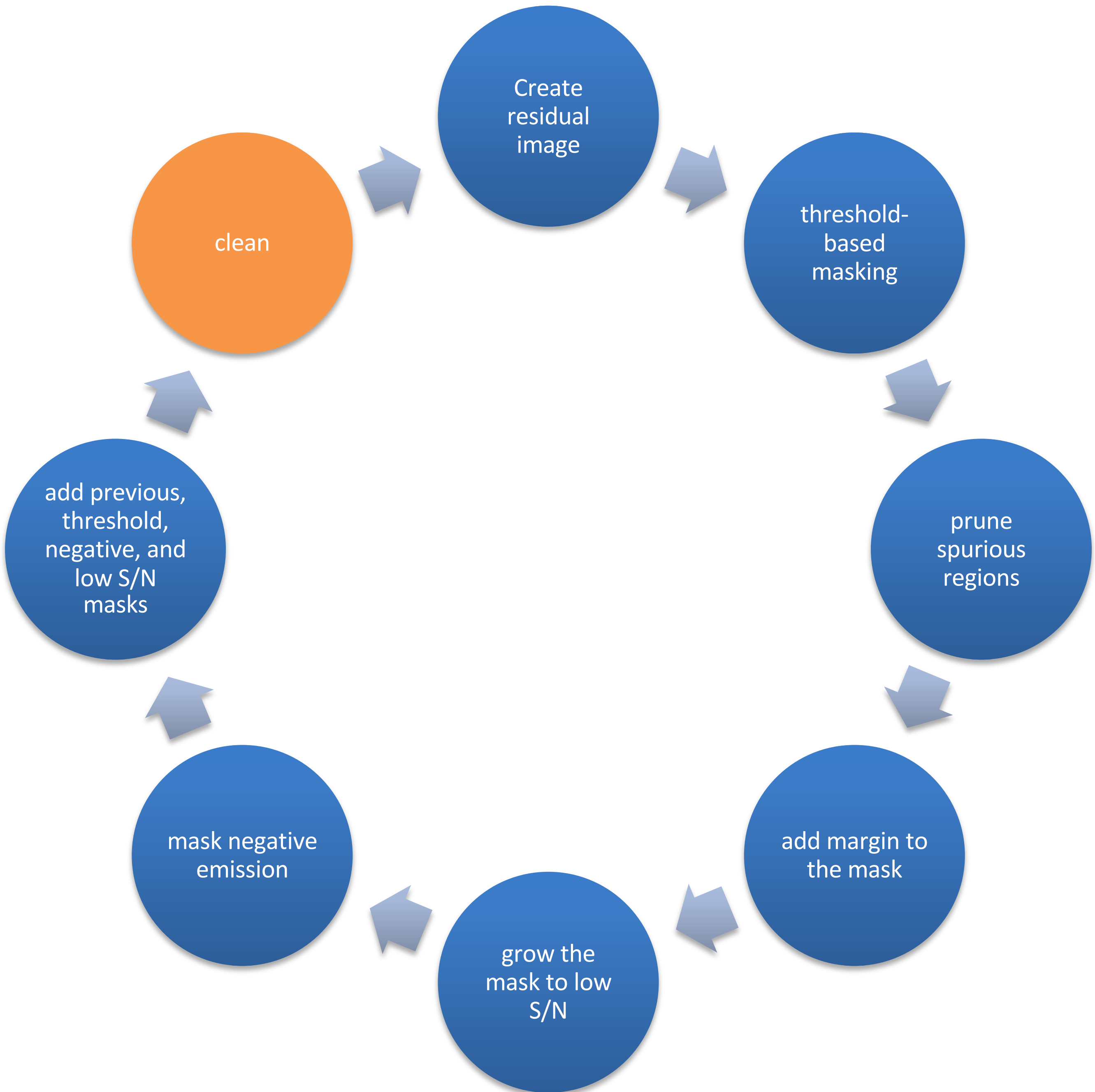

Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, negative, and low S/N masks
clean

Absorption is masked using a simple threshold.

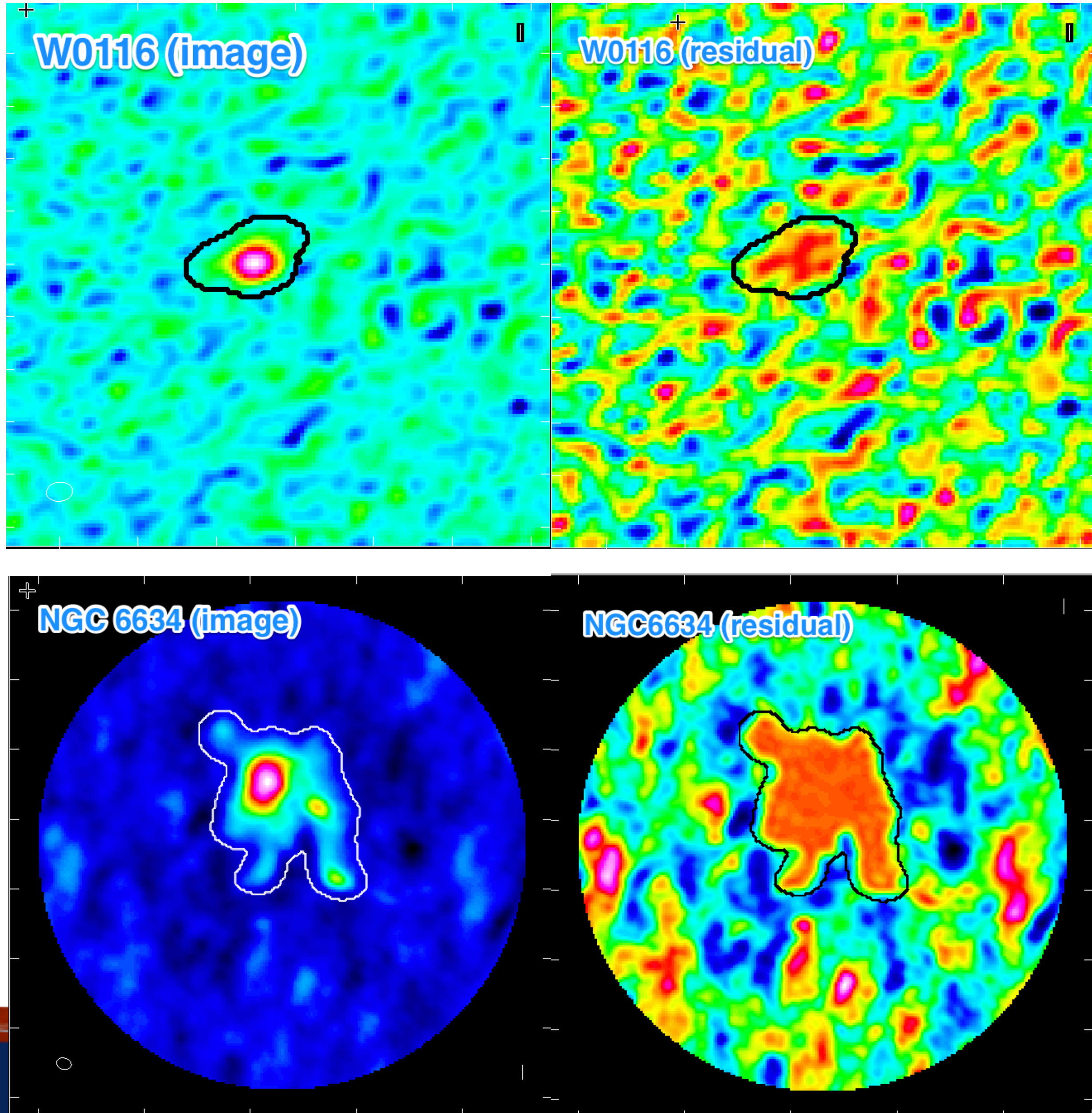


sidelobeThresholdValue = **sidelobeThreshold** * sidelobeLevel * residPeak
negativeThresholdValue = max(**negativeThreshold** * rms, sidelobeThresholdValue)
mask negative pixels with values <= - negativeThresholdValue
Smooth and cut mask as done with positive threshold mask.

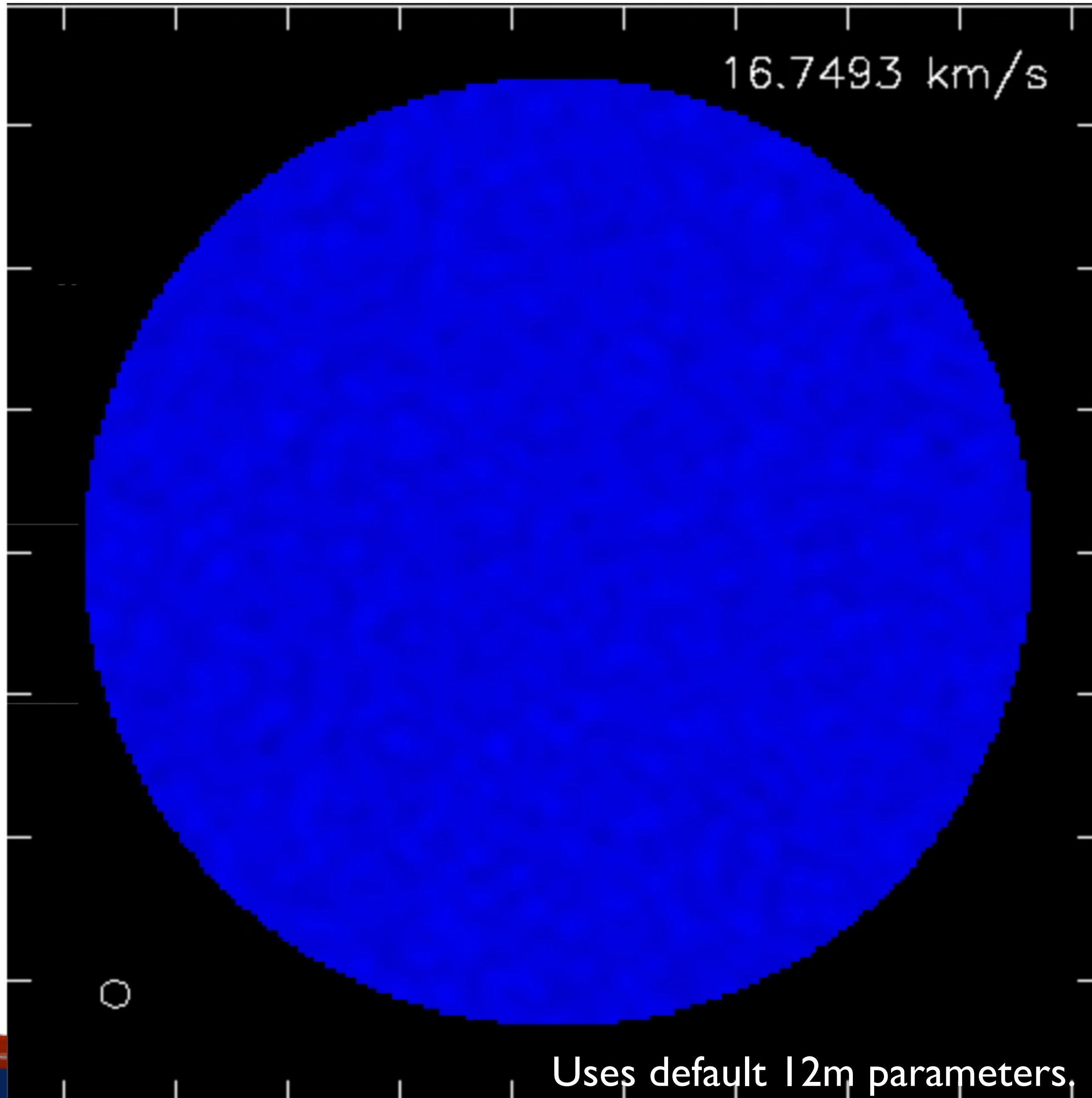

Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, negative, and low S/N masks
clean


Create residual image
threshold-based masking
prune spurious regions
add margin to the mask
grow the mask to low S/N
mask negative emission
add previous, threshold, negative, and low S/N masks
clean

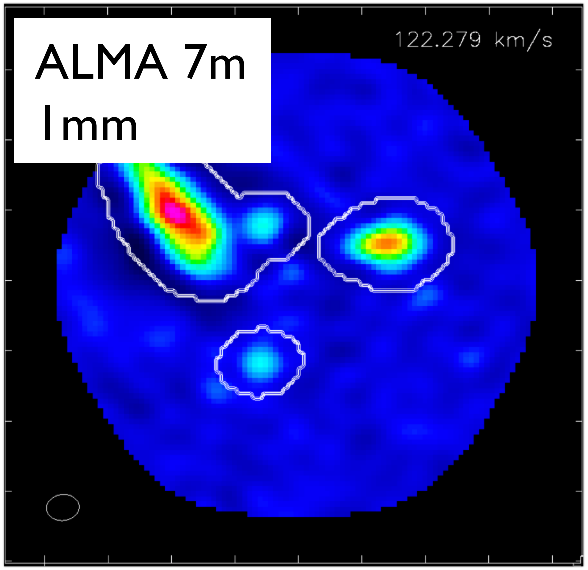
Algorithm can handle both simple and complex emission.

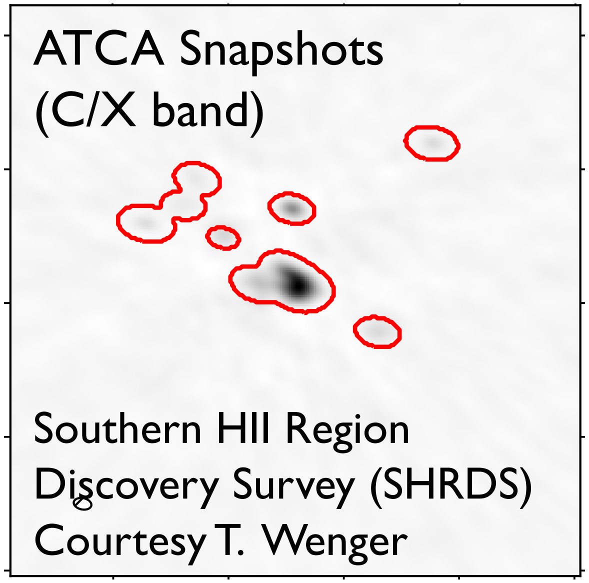


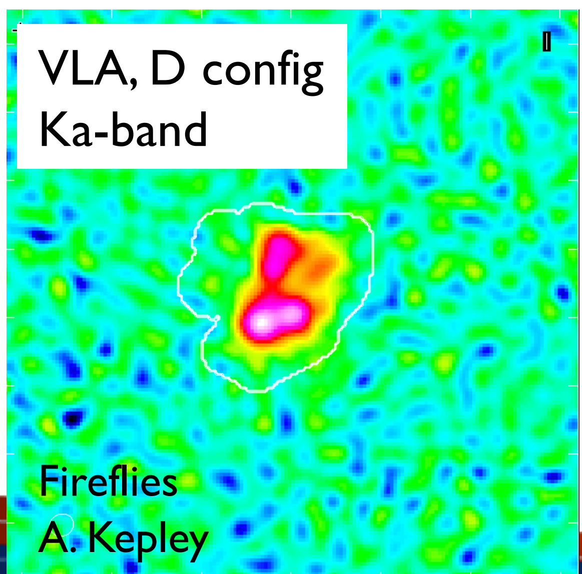
Algorithm can handle cubes.

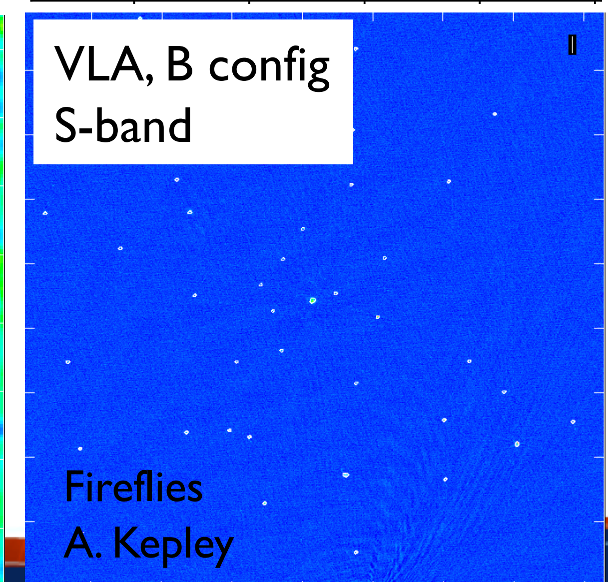


Algorithm also works for ALMA 7m, VLA, and ATCA data sets.



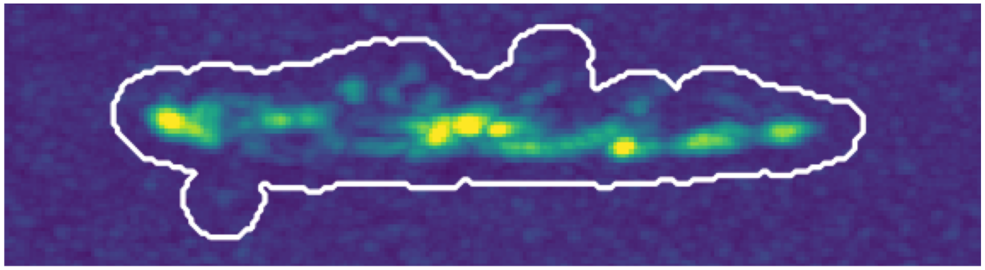






- Algorithm performs well for a variety of data.
- Parameters need to be tuned based on the data set.
- The primary discriminate appears to be the PSF.
- The ALMA pipeline uses three different sets of parameters:
    - 12m long baseline
    - 12m short baseline
    - 7m

Conclusions



- We have developed a general purpose algorithm to automatically mask emission while cleaning to enable the fully automated imaging of interferometer data.
- The algorithm is implemented in CASA 5.1 and up as usemask='auto-multithresh' and in production in the ALMA Cycle 5 and 6 Imaging Pipelines.
- The algorithm works well for a wide variety of data, although the parameters do need to be tuned based on the PSF of the data.
- Paper in preparation (Kepley, Tsutsumi, et al) and automask casaguide available
  - https://casaguides.nrao.edu/index.php/Automasking_Guide
- Happy to take questions about speed and performance during the question section!

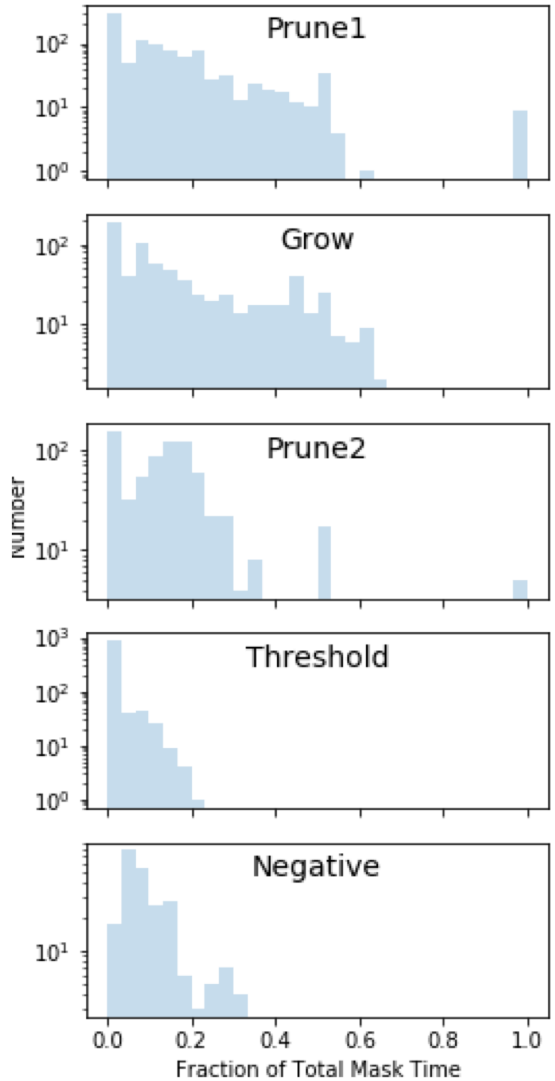
NRAO

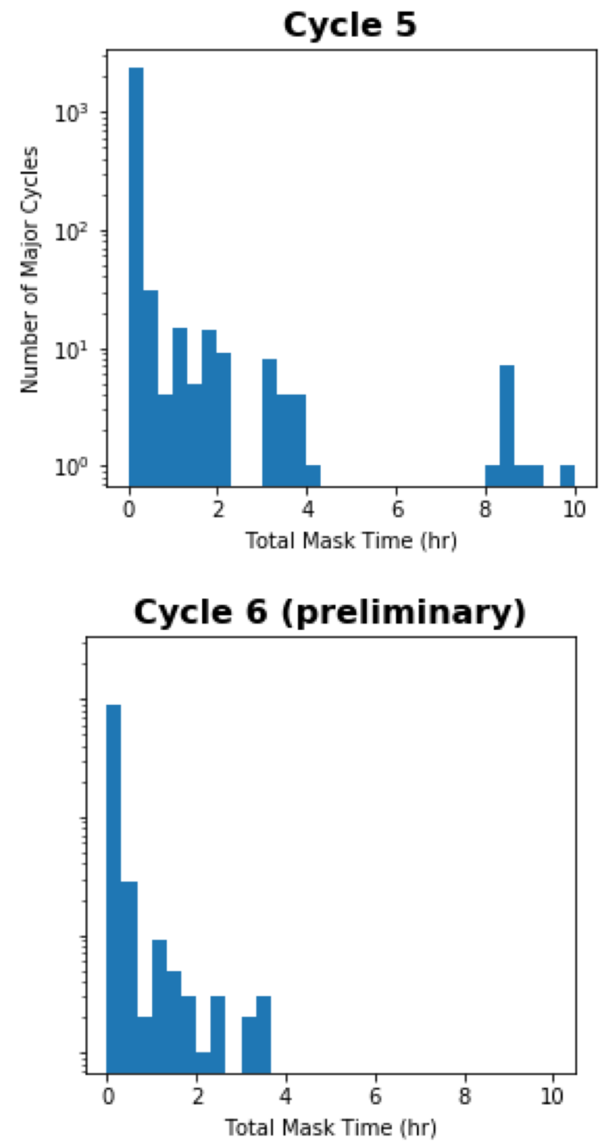
www.nrao.edu
science.nrao.edu
public.nrao.edu

The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

For last year, we have been focusing on speeding up the algorithm.

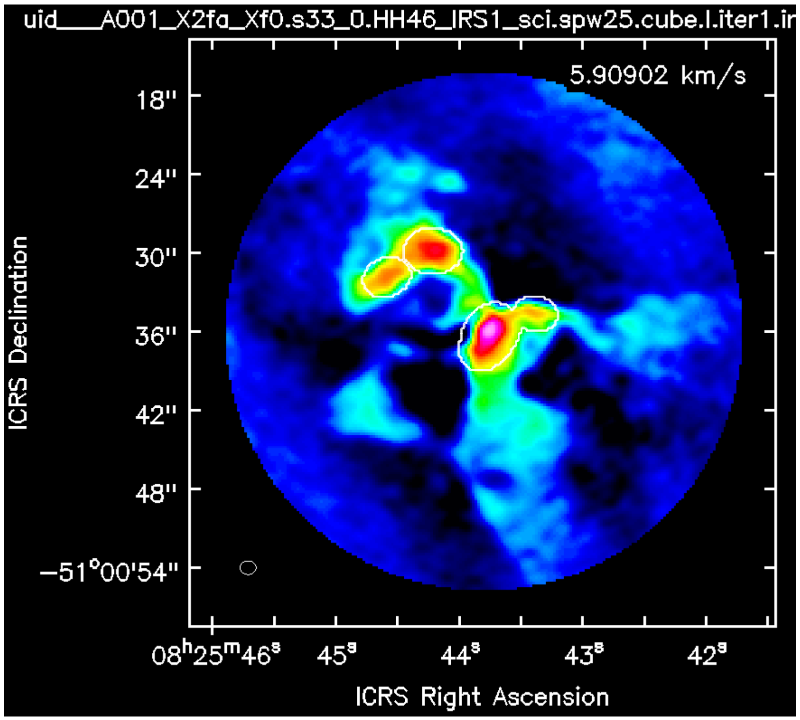
- General purpose algorithm, which introduces additional complexity.
- Most time consuming portions are the prune and grow steps.
- Have been able to reduce masking time by a factor of 2 for Cycle 6.
- Biggest win has been to stop updating the mask based on certain conditions.





Auto-multithresh produces reasonable masks for most data sets with a few exceptions.

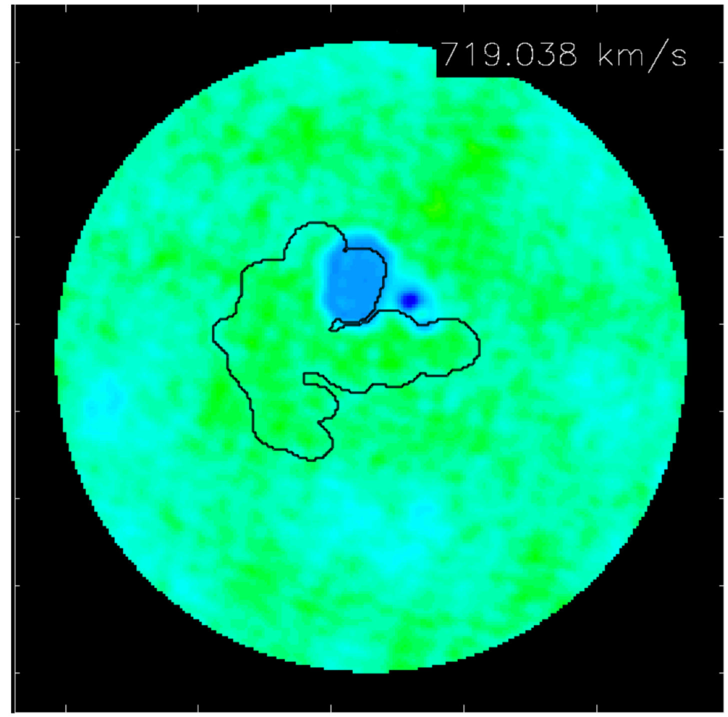
Channels with bright, wide-spread emission



Problem: Currently use simple MAD to estimate noise

Improvement in CASA 5.5: better noise estimates

Low level noise near absorption (i.e., fluff)



Problem: Subtle bugs with negative mask calculation

Improvements: Track positive and negative masks separately (5.3) and <u>take absolute value of peak residual (5.5)</u>

All statistics use the MAD to estimate the noise.*

For a univariate data set $X_1, X_2, \ldots, X_n$, the MAD is defined as the median of the absolute deviations from the data's median:

$$\mathrm{MAD} = \mathrm{median}(\ |X_i - \mathrm{median}(X)|\ ),$$

wikipedia

In practice, it's a robust statistic that allows us to do a good estimate the noise in the presence of signal.

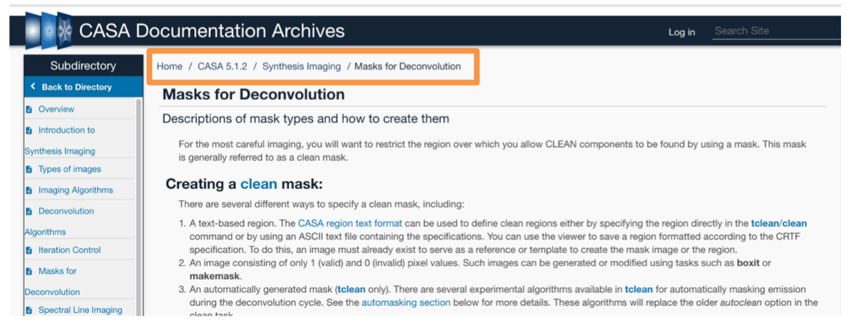
You can rescale to a gaussian error by multiplying via 1.4826.

* A more sophisticated noise estimate is in flight for CASA 5.5. This estimate will be used for the ALMA Cycle 7 Pipeline.

Efforts are also underway to improve user documentation for this feature.

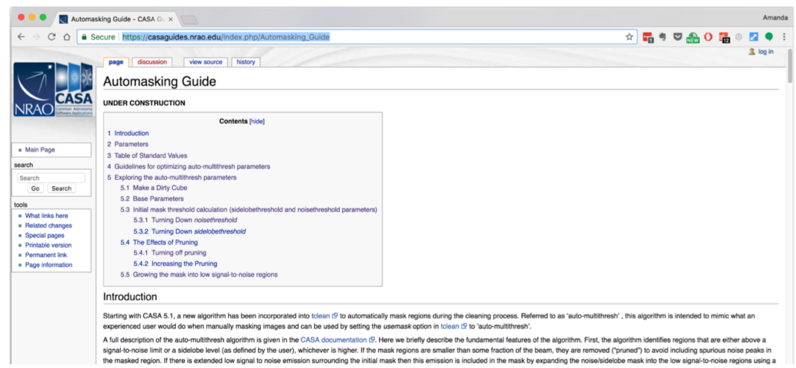
CASA Plone Documentation:

- Description of algorithm and parameters
- Updated with current state of algorithm



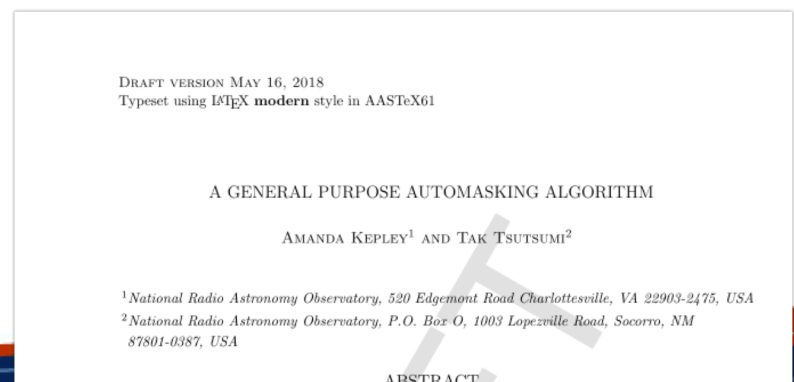
CASA Guide:

- Lead by Andy Lipnicky (NAASC)
- https://casaguides.nrao.edu/index.php/Automasking_Guide



Paper:

- In preparation
- Kepley, Tsutsumi et al.


DRAFT VERSION MAY 16, 2018
Typeset using LaTeX **modern** style in AASTeX61

A GENERAL PURPOSE AUTOMASKING ALGORITHM

AMANDA KEPLEY[1] AND TAK TSUTSUMI[2]

[1]*National Radio Astronomy Observatory, 520 Edgemont Road Charlottesville, VA 22903-2475, USA*
[2]*National Radio Astronomy Observatory, P.O. Box O, 1003 Lopezville Road, Socorro, NM 87801-0387, USA*

ABSTRACT

Path from Implementation to Production

- Prototyped in Python using custom version of tclean task that runs masking code internally [A. Kepley, Fall 2016]
  - https://github.com/aakepley/autobox
- Verified approach on sub-set imaging pipeline benchmark data sets (plus some friends) [A. Kepley, Fall 2016]
- Implemented in tclean as 'auto-multithresh' [T. Tsutsumi, Fall 2016 - Spring 2017]
- Tested tclean implementation [I. Yoon & A. Kepley, Spring 2017]
- Auto-multithresh tested on entire ALMA benchmark suite and parameters tuned to produce the best masking. [A. Kepley and C. Brogan, Summer 2017]
- Auto-multithresh in production in the Cycle 5 pipeline [October 2017 – September 2018]
- Focus on speed improvements for CASA 5.3 and ALMA Cycle 6 [October 2017 - September 2018]
- Improving noise estimate and testing interaction with n-sigma clean [October 2018 to Summer 2019]