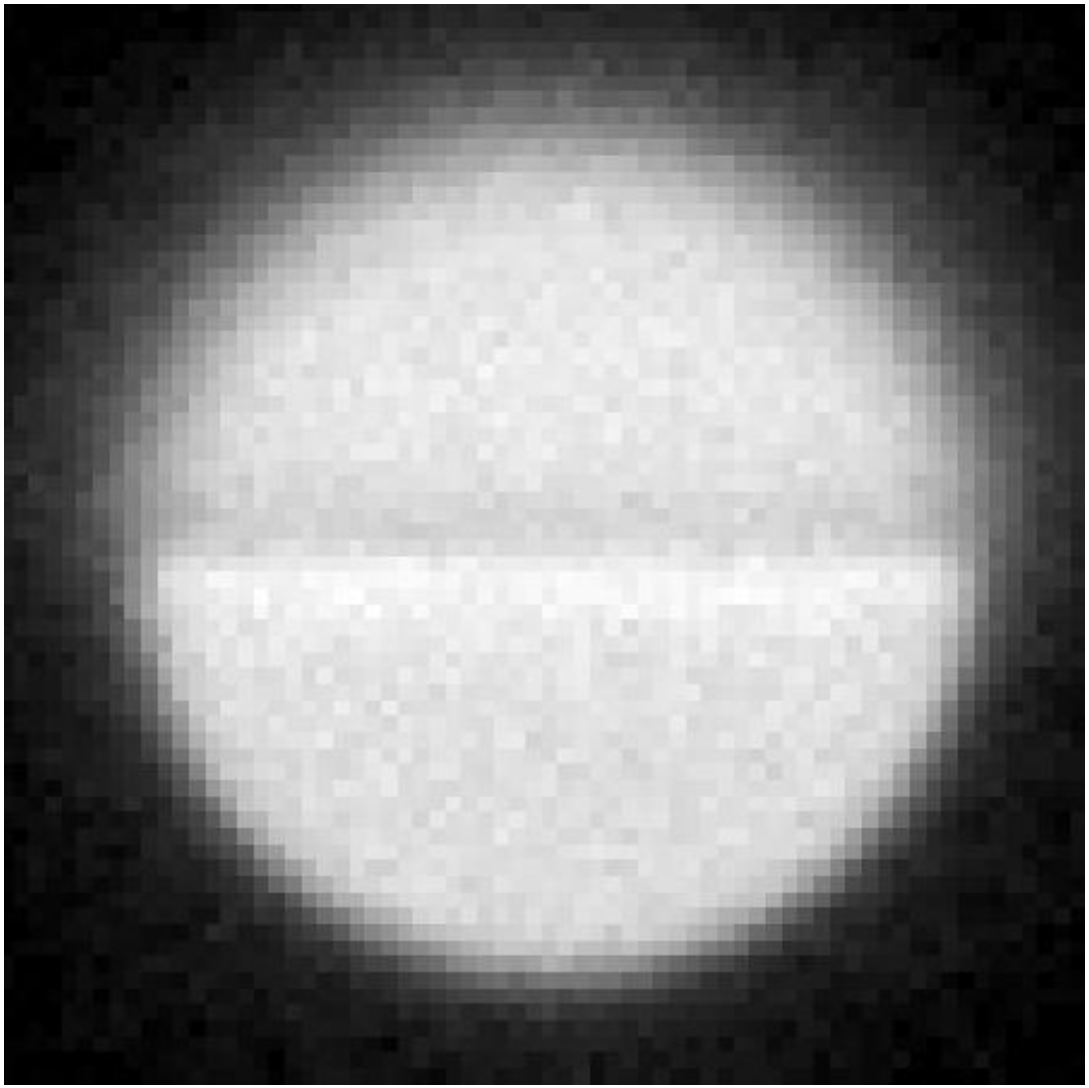
Monte Carlo simulation *noise* removal

By Wavelet thresholding

MC simulation

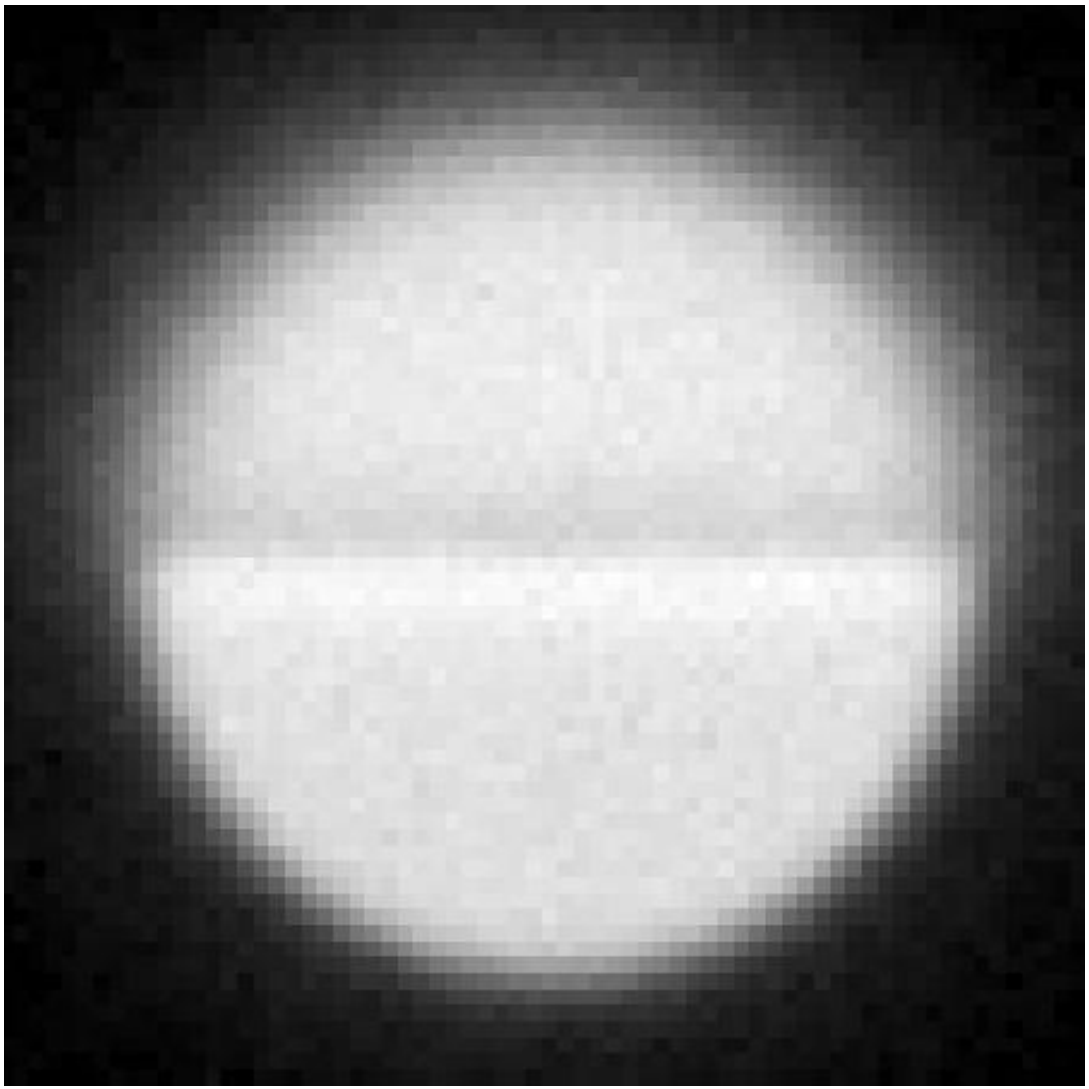
- The MC method is known to be the ultimate dose distribution computational technique
- The simulation generates 2, 4,8, 16M electrons beam dose distributions, with a proportionally increasing computational cost
- The algorithm is fully parallel : adding 2 independent 2Me- distributions results in a 4Me- distribution
- However, the process introduces an inherent noise in the resulting dose distributions, and a very long run (100,150,200Me- beams?) of the MC simulation is necessary to get rid of it

Grayscale displays of 2D dose distribution maps



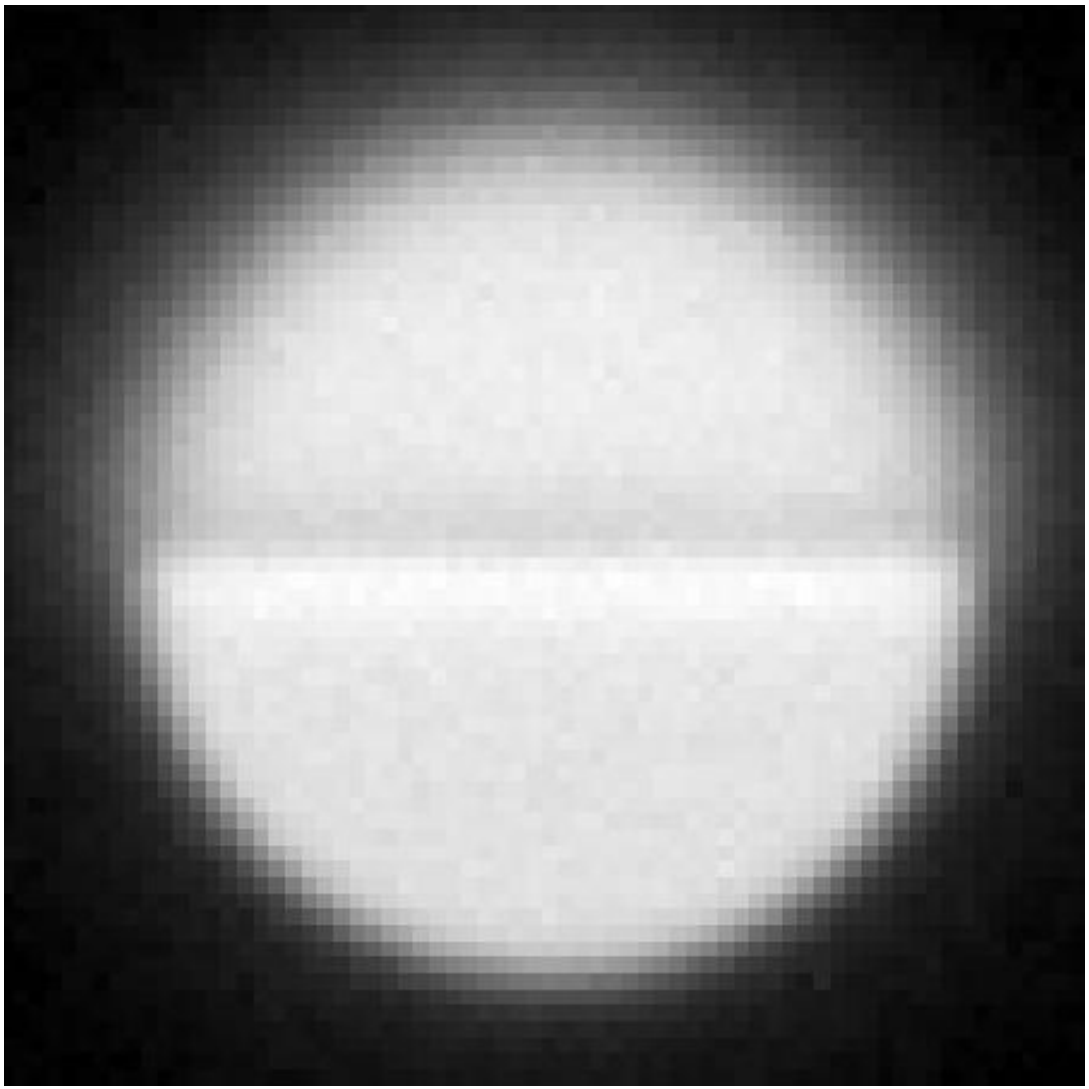
2Me- beam dose radiation

Grayscale displays of 2D dose distribution maps



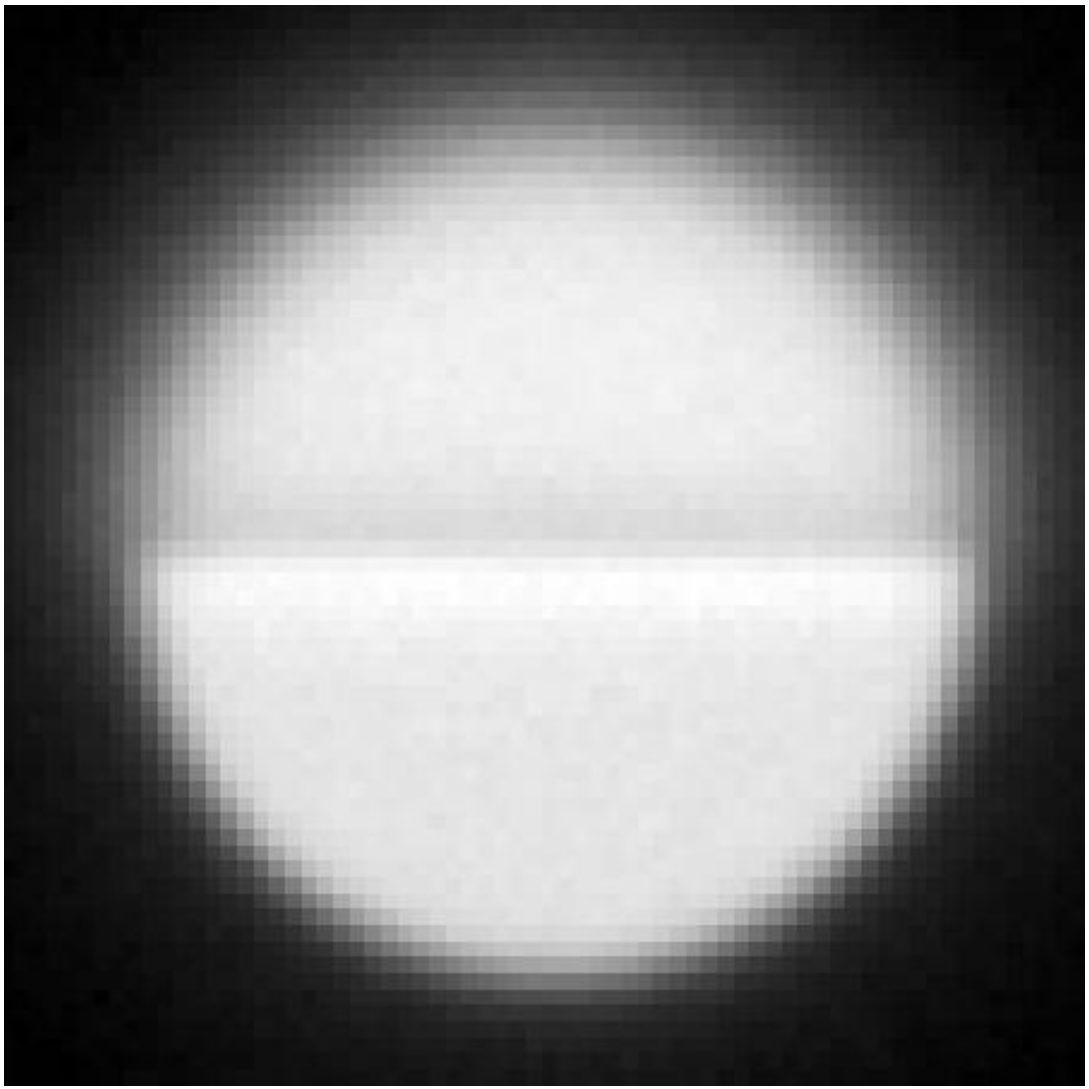
4Me- beam dose radiation

Grayscale displays of 2D dose distribution maps



8Me- beam dose radiation

Grayscale displays of 2D dose distribution maps



16Me- beam dose radiation

Idea

A post-processing step (de-noising) could help in reducing the statistical fluctuations in the MC dose distribution, and thus save an important computation time

Hypothesis on the noise

- De-noising is more efficient if the noise present in the dose distribution is additive and if its variance is independent from the dose value
- We would like to apply a variance stabilizing transformation (reversible) before de-noising
- Previous works suggested that the square root transformation is quite appropriate

De-noising process

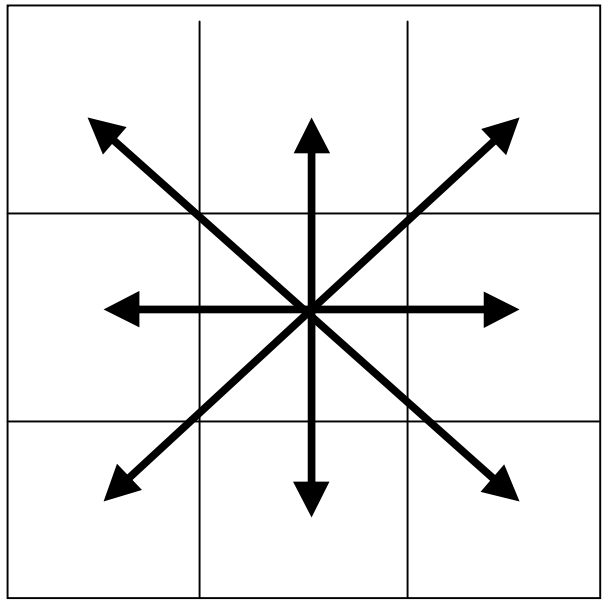
- We start with a 2D dose distribution array ***p***
- We apply the square root variance stabilizing transformation : ***p'* = sqrt(*p*)**
- The (9-7) wavelet transform is applied to ***p'***: ***c* = Omega(*p'*)**
- Wavelet Thresholding : given a percentage ***T***, every wavelet coefficient such that $|c_{i,j}| \leq T \max(c)$ is set to **0**

De-noising process

- The inverse (9-7) wavelet transform is applied to the resulting array $\boldsymbol{c}$
- The new array $\boldsymbol{g}$ values are truncated if necessary to fit in the [0,max] interval of the original dose distribution $\boldsymbol{p'}$
- We finally get the de-noised dose distribution $\boldsymbol{g'}$ by reversing the square root transform : $\boldsymbol{g' = g^2}$

Spin Cycle technique

- Wavelet transforms are very sensitive to a shift on the original data
- To reduce these artifacts, one can apply the previous de-noising method to the 9 2D maps resulting from shifts in all directions, shift back and average the de-noised maps



How to determine the thresholding percentage?

- We dispose of long runs of the MC simulation = what the MC simulation converges to

- One might want for a de-noised dose distribution to be the closest to the “limit” of the simulation, according to the euclidian distance

How to determine the thresholding percentage?

MEAN SQUARE ERROR

$$D_{mse}(\mathbf{p}-\hat{\mathbf{p}}) = \frac{\|\mathbf{p}-\hat{\mathbf{p}}\|^2}{N} = \frac{1}{N}\sum_i \sum_j (p_{i,j}-\hat{p}_{i,j})^2$$

N is the total number of pixels

PSNR

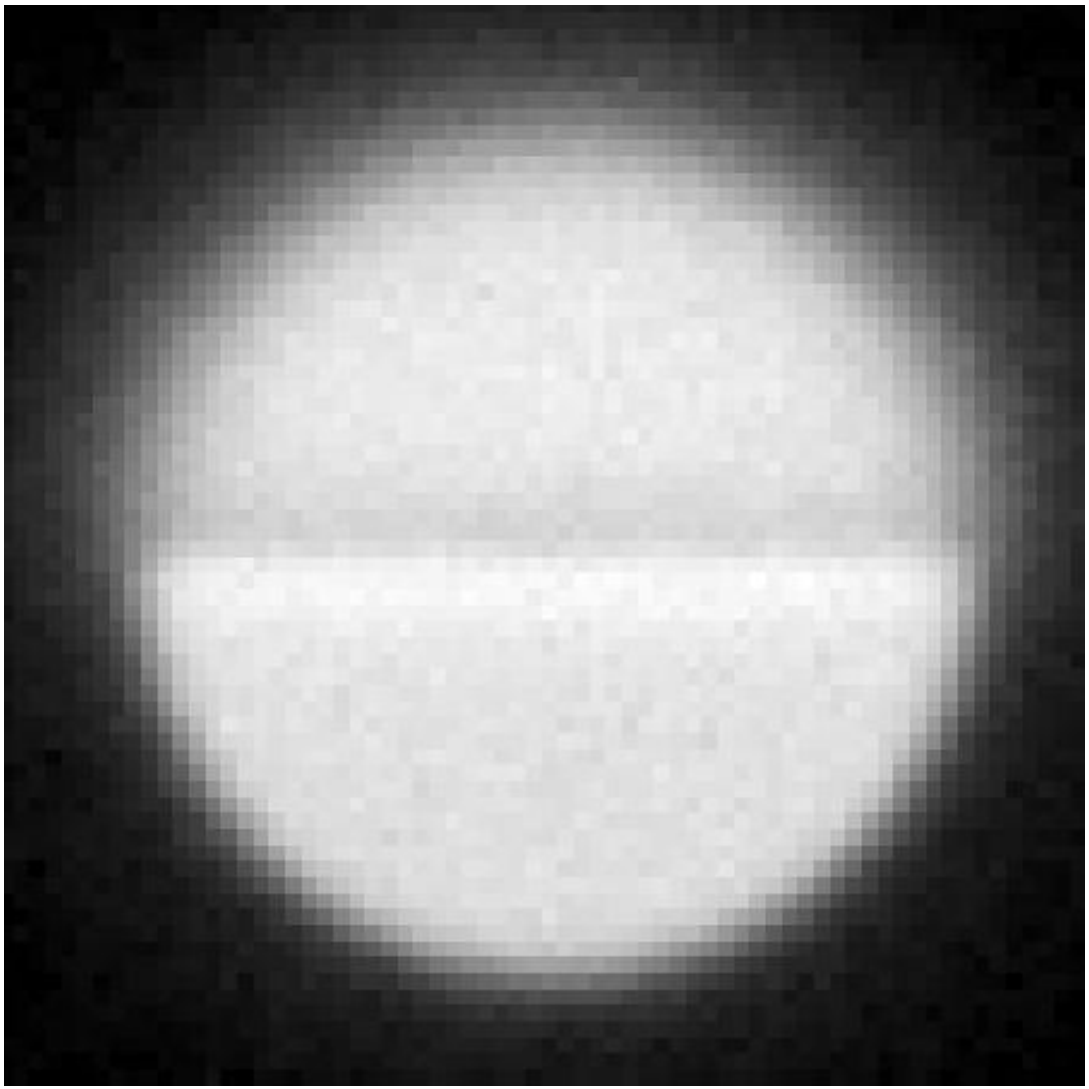
$$PSNR(\mathbf{p}-\hat{\mathbf{p}}) = 10\log_{10}\left(\frac{max^2}{D_{mse}(\mathbf{p}-\hat{\mathbf{p}})}\right) \text{ dB}$$

max is the maximum dose value

How to determine the thresholding percentage?

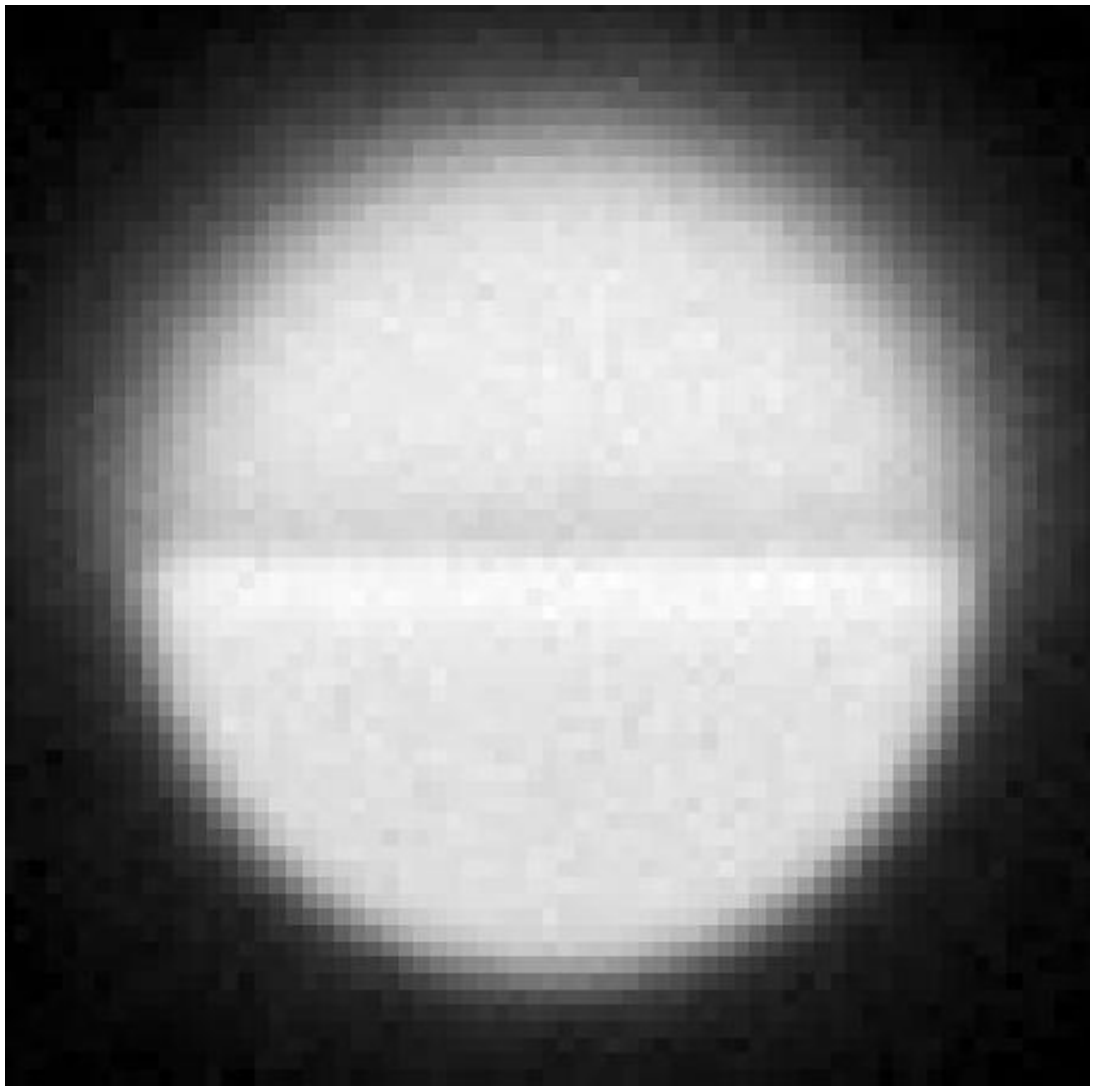
- The higher the **PSNR** is, the closer the de-noised file to the limit of the MC simulation gets (according to the $\| \|_2$ norm)
- The PSNR(thresholding percentage) function usually presents a maximum

Example: from mild to severe de-noising



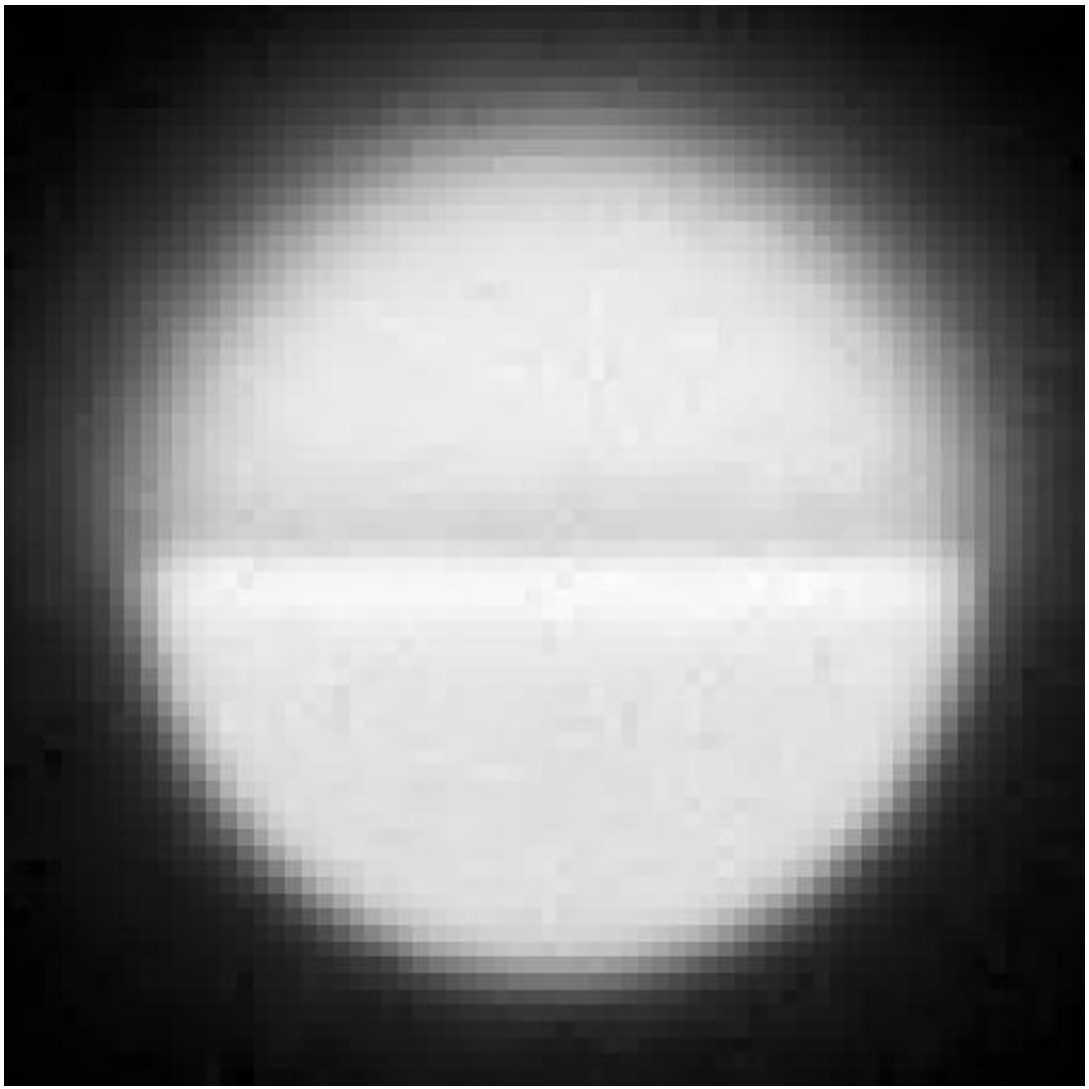
4Me- dose distribution: original

Example: from mild to severe de-noising



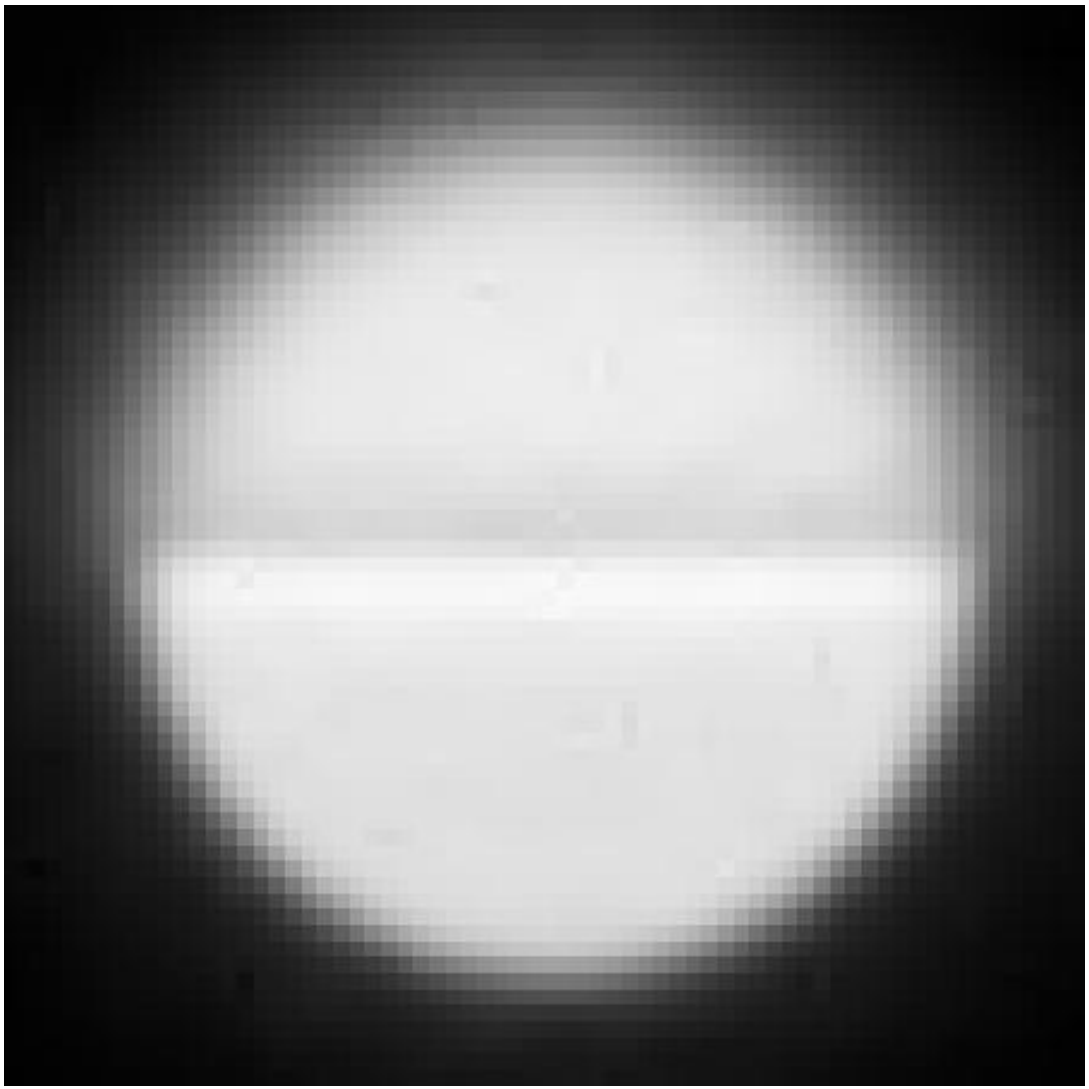
4Me- percentage: 0.05%

Example: from mild to severe de-noising



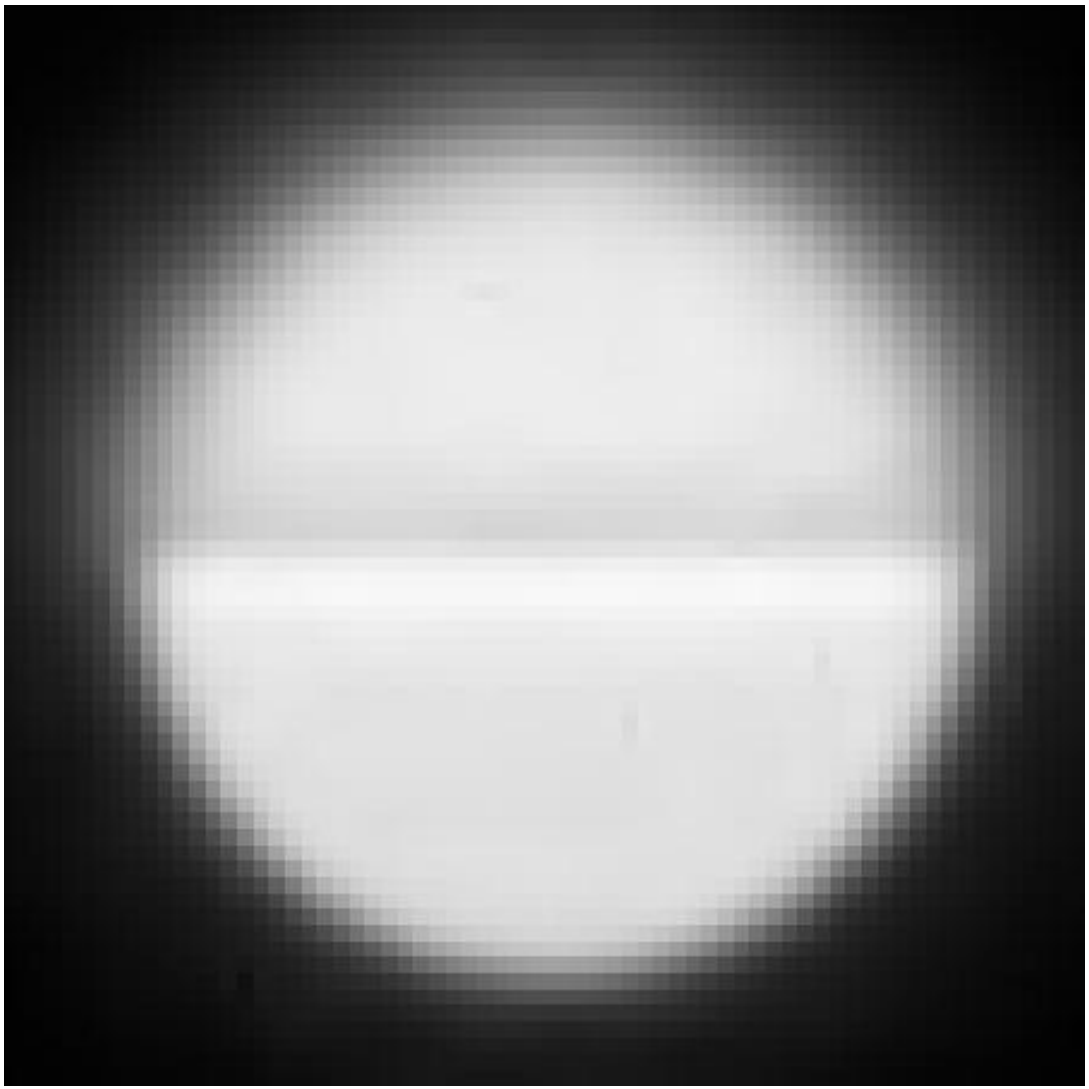
4Me- percentage: 0.1%

Example: from mild to severe de-noising



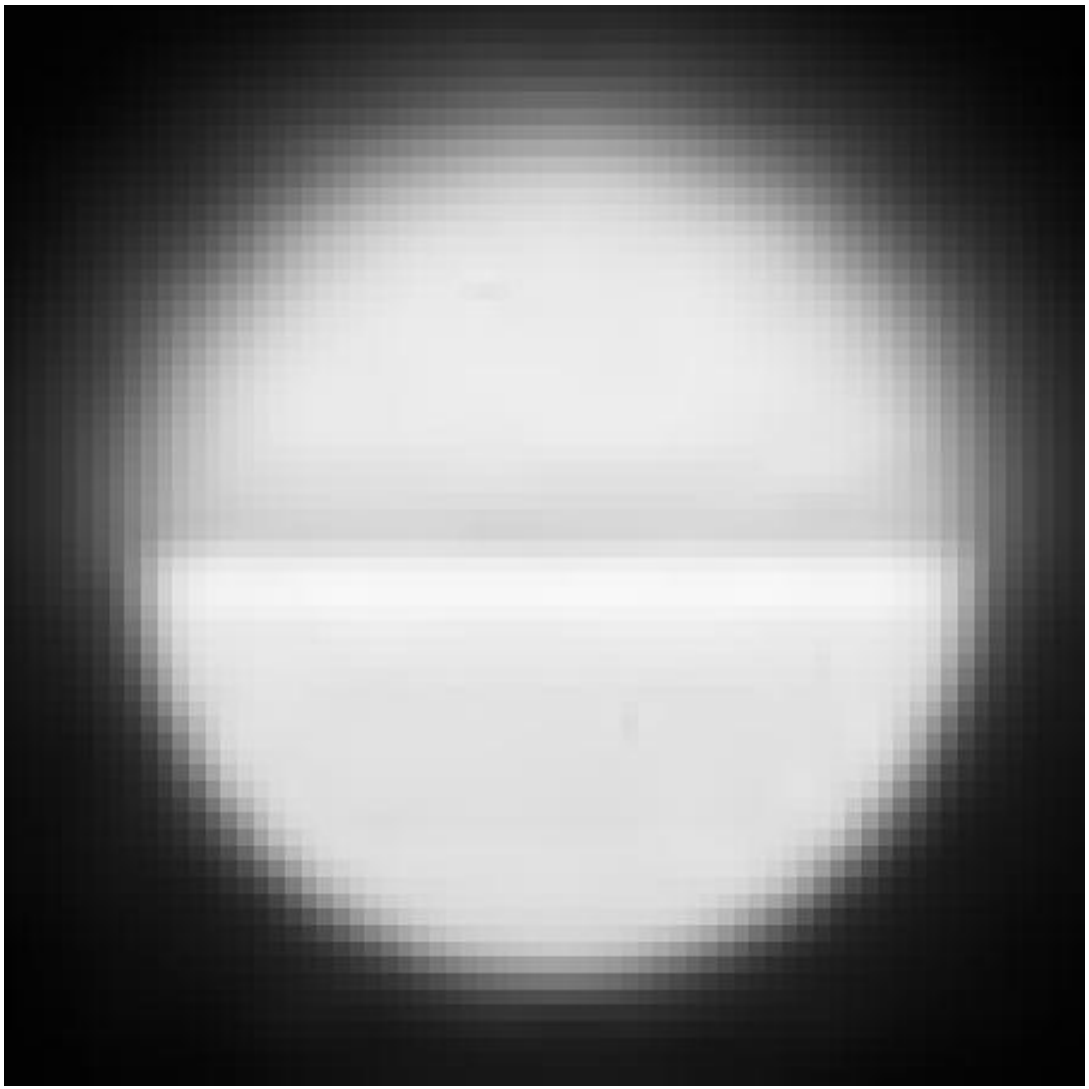
4Me- percentage: 0.15%

Example: from mild to severe de-noising



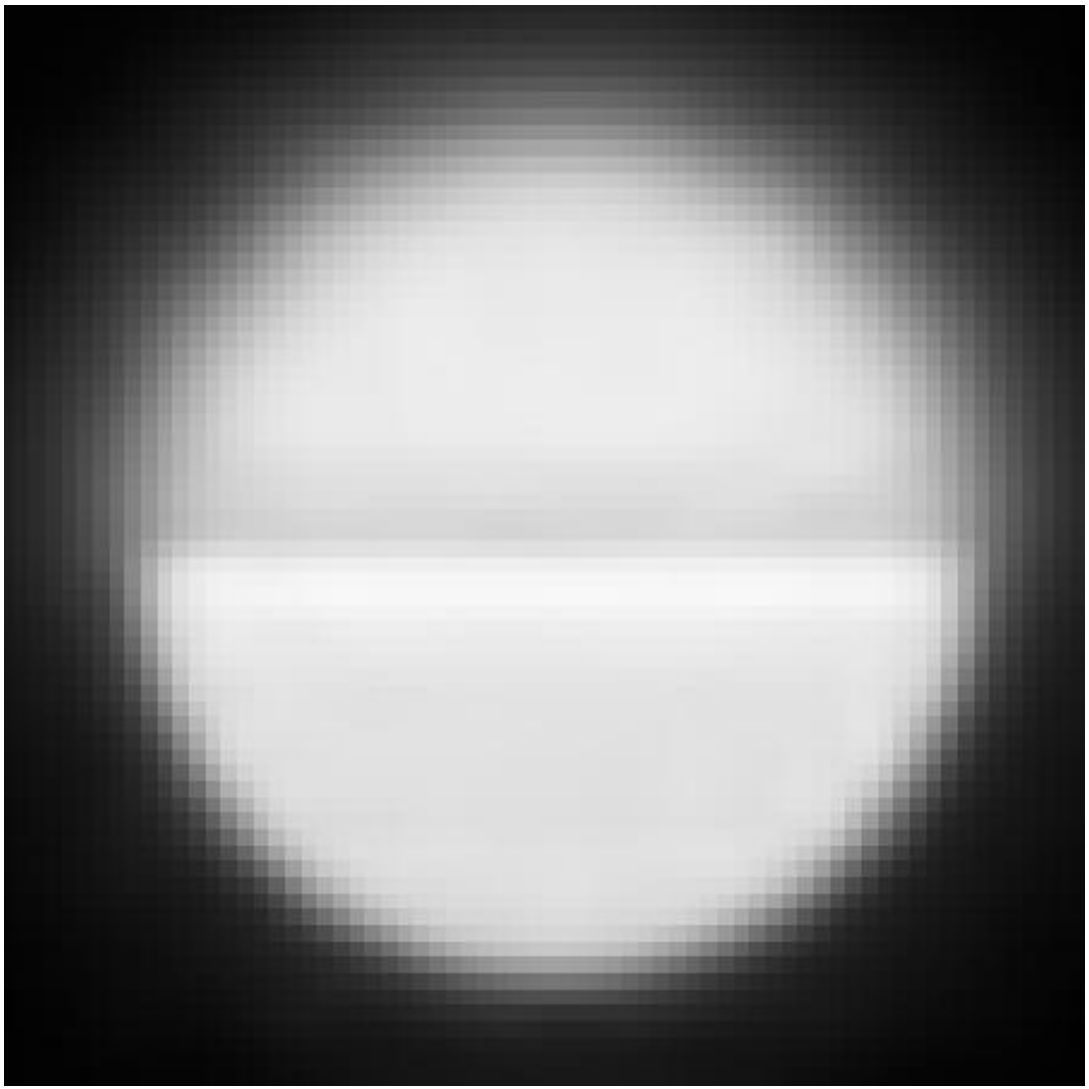
Best PSNR=47dB; percentage: 0.18%

Example: from mild to severe de-noising



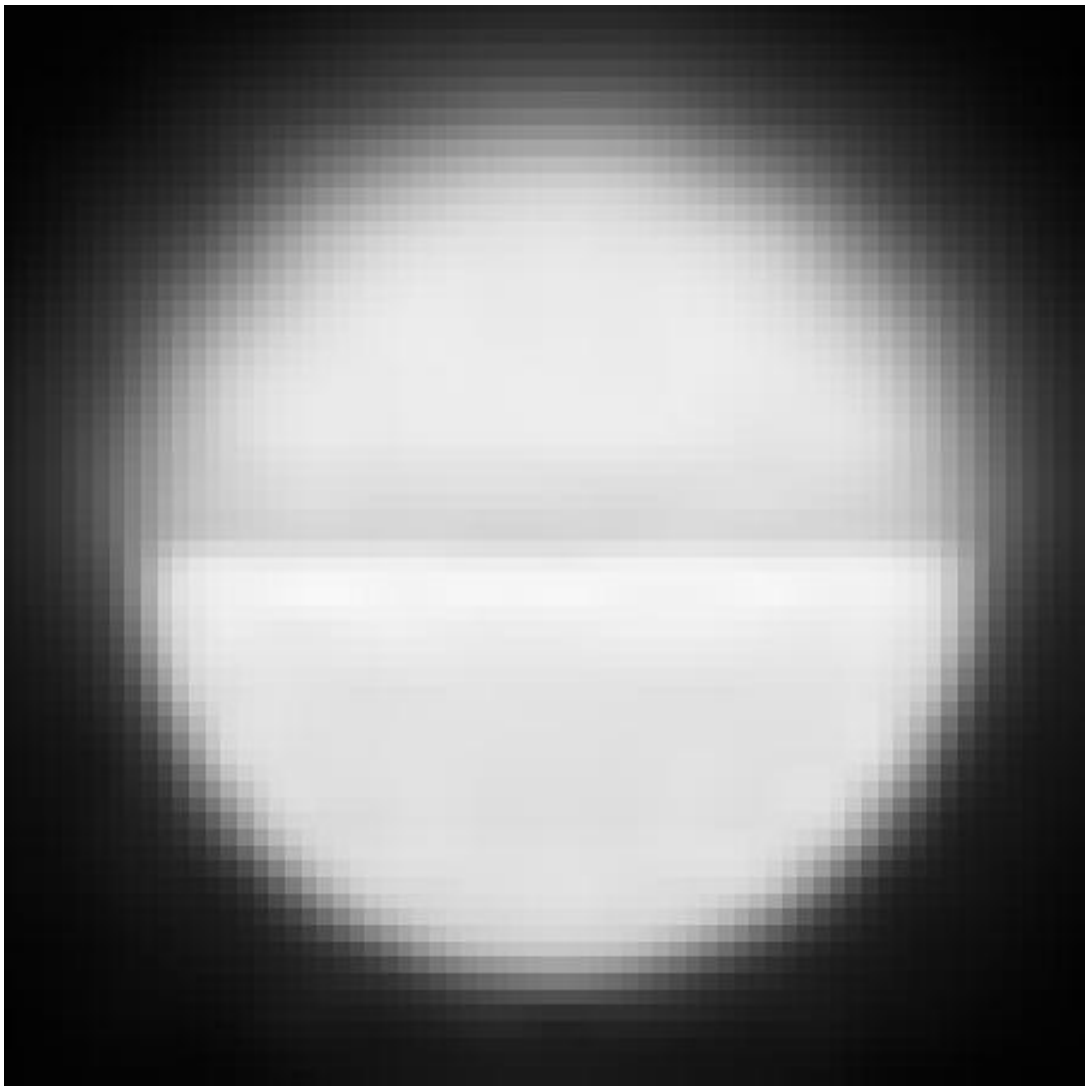
4Me- percentage: 0.2%

Example: from mild to severe de-noising



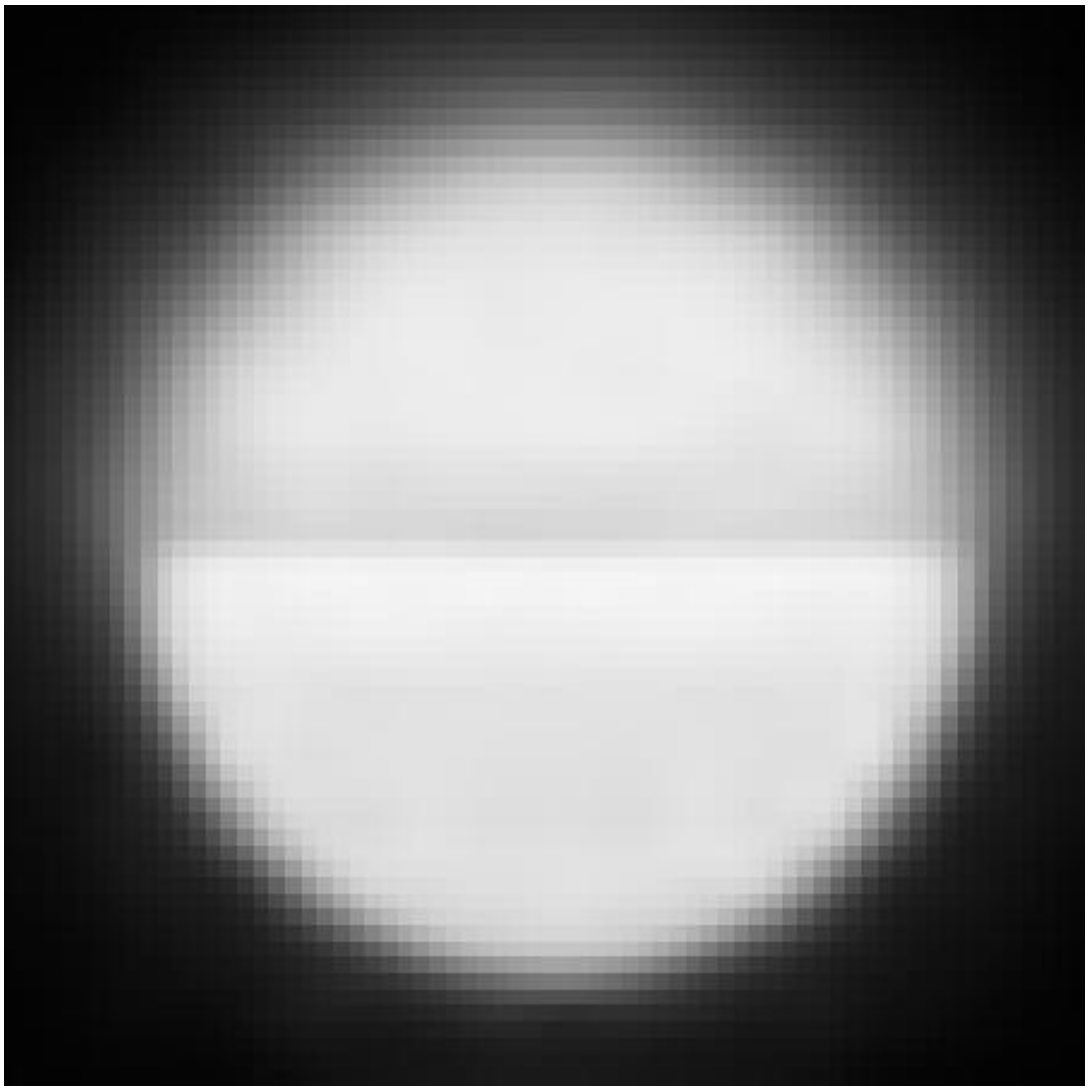
4Me- percentage: 0.25%

Example: from mild to severe de-noising



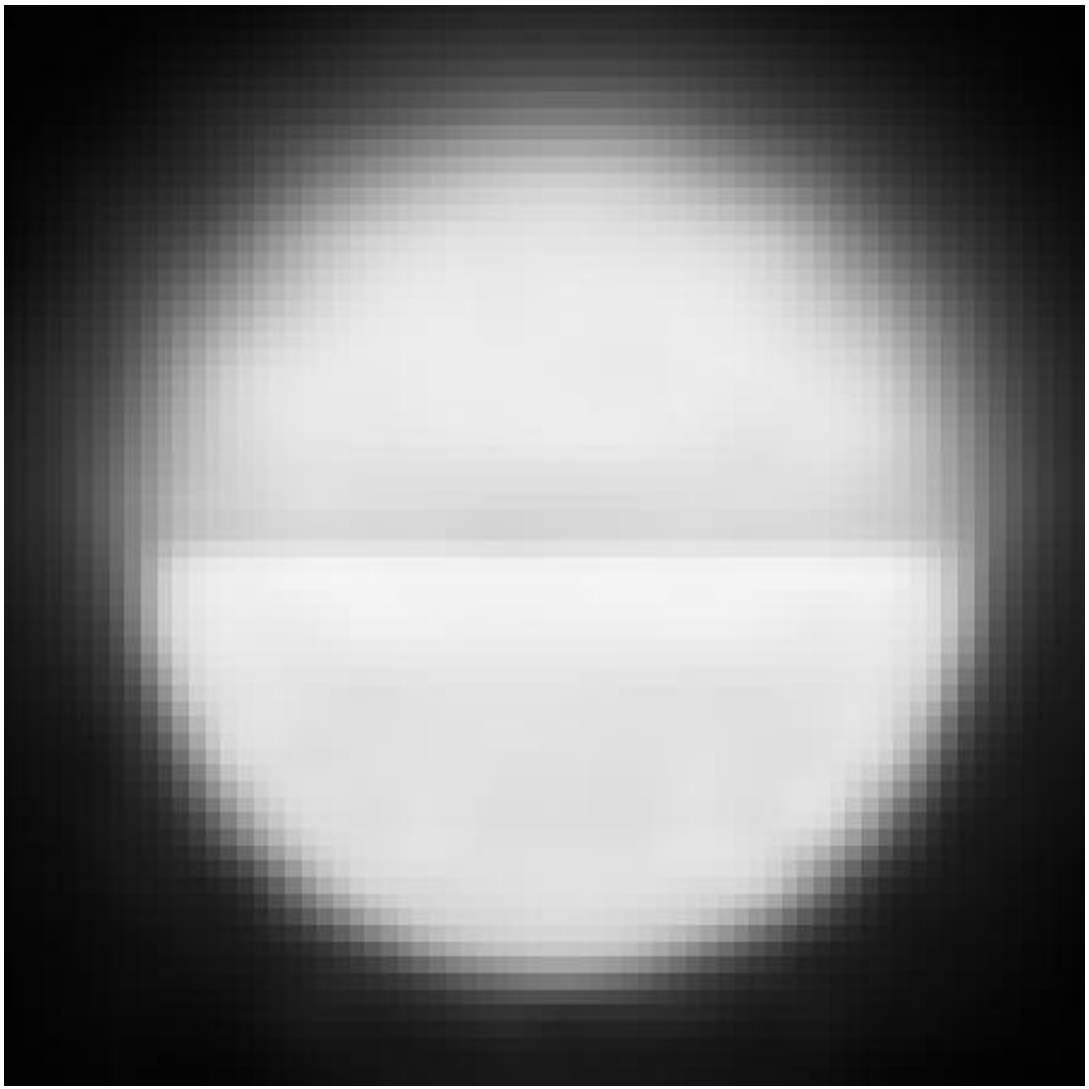
4Me- percentage: 0.3%

Example: from mild to severe de-noising



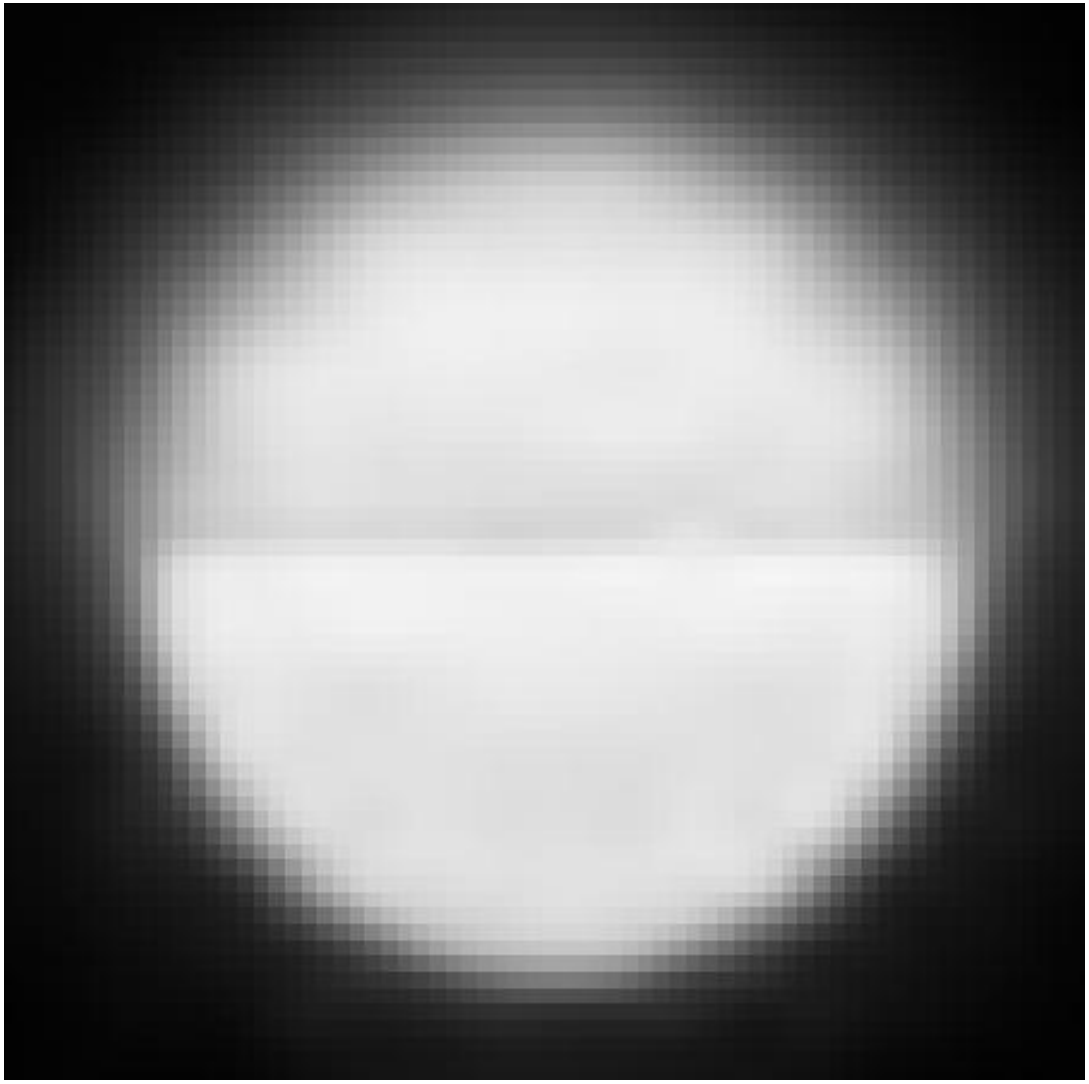
4Me- percentage: 0.35%

Example: from mild to severe de-noising



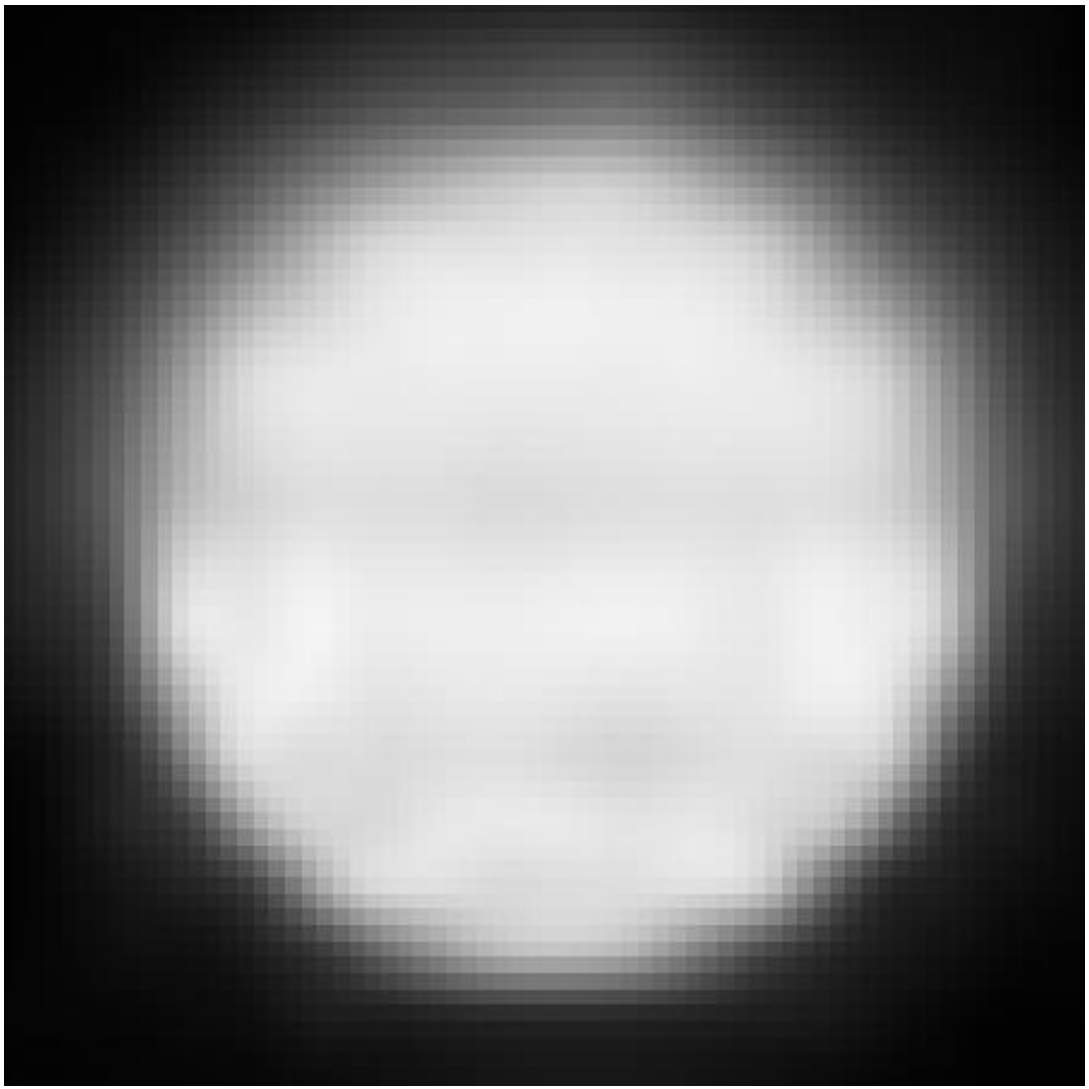
4Me- percentage: 0.4%

Example: from mild to severe de-noising



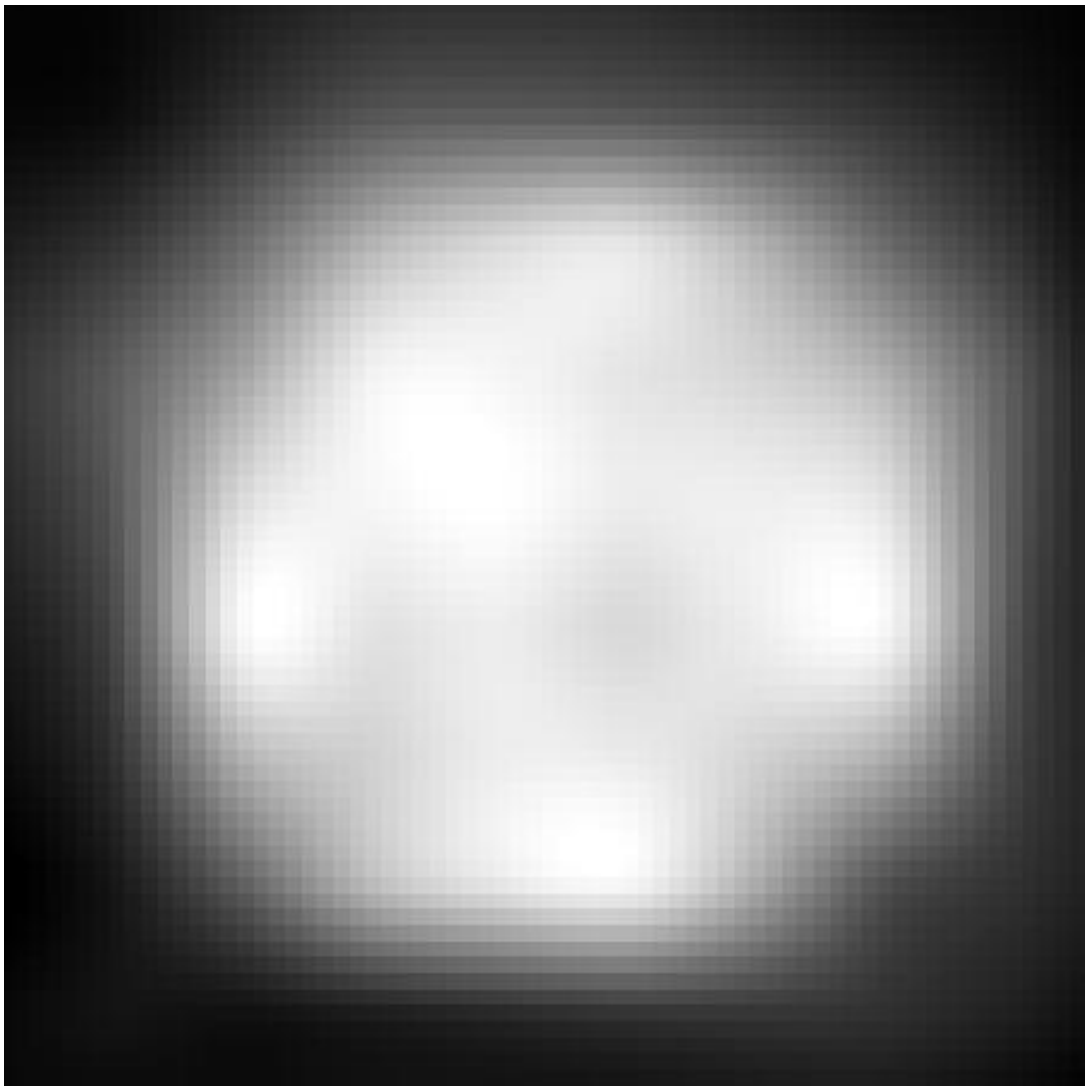
4Me- percentage: 0.5%

Example: from mild to severe de-noising

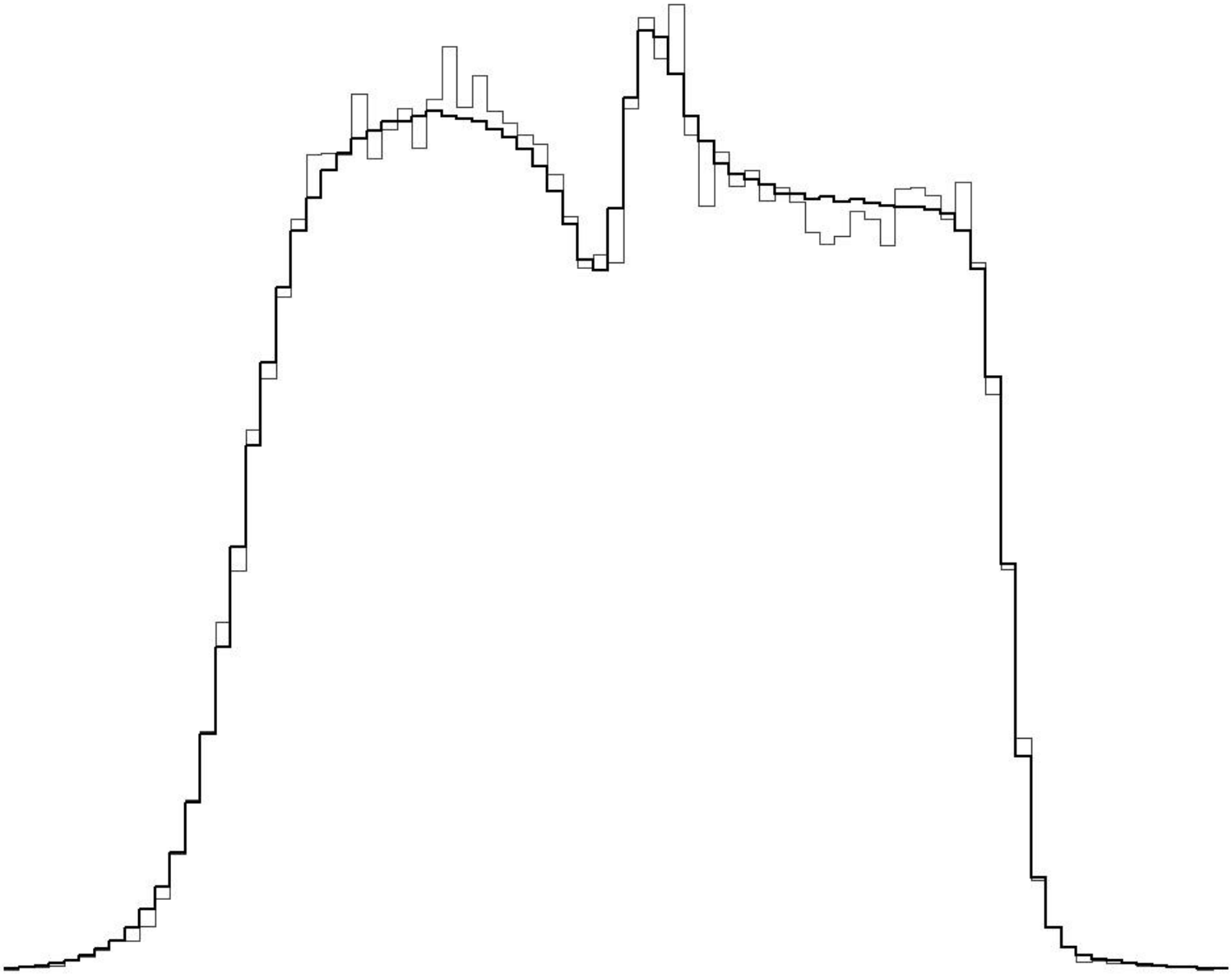


4Me- percentage: 1.0%

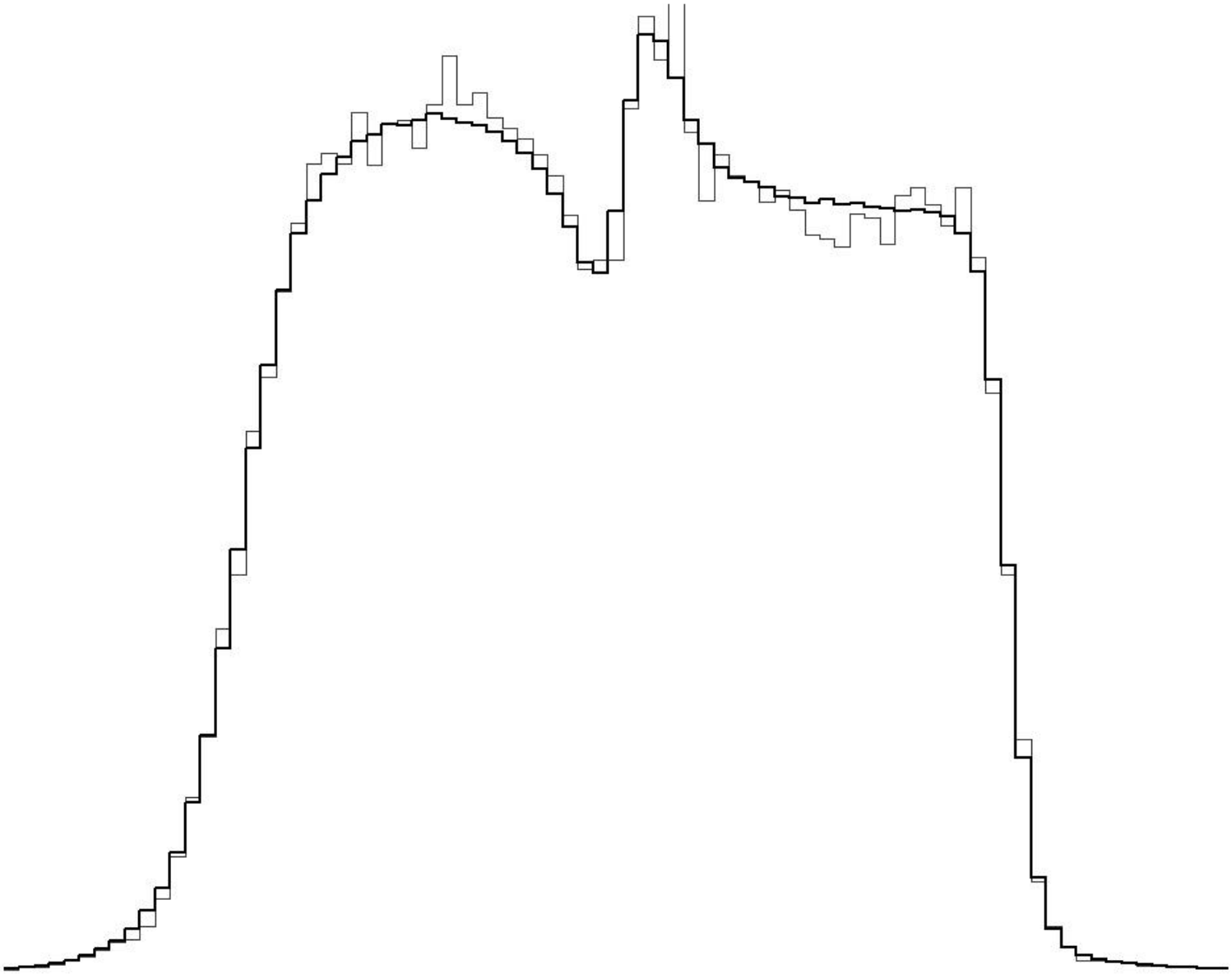
Example: from mild to severe de-noising



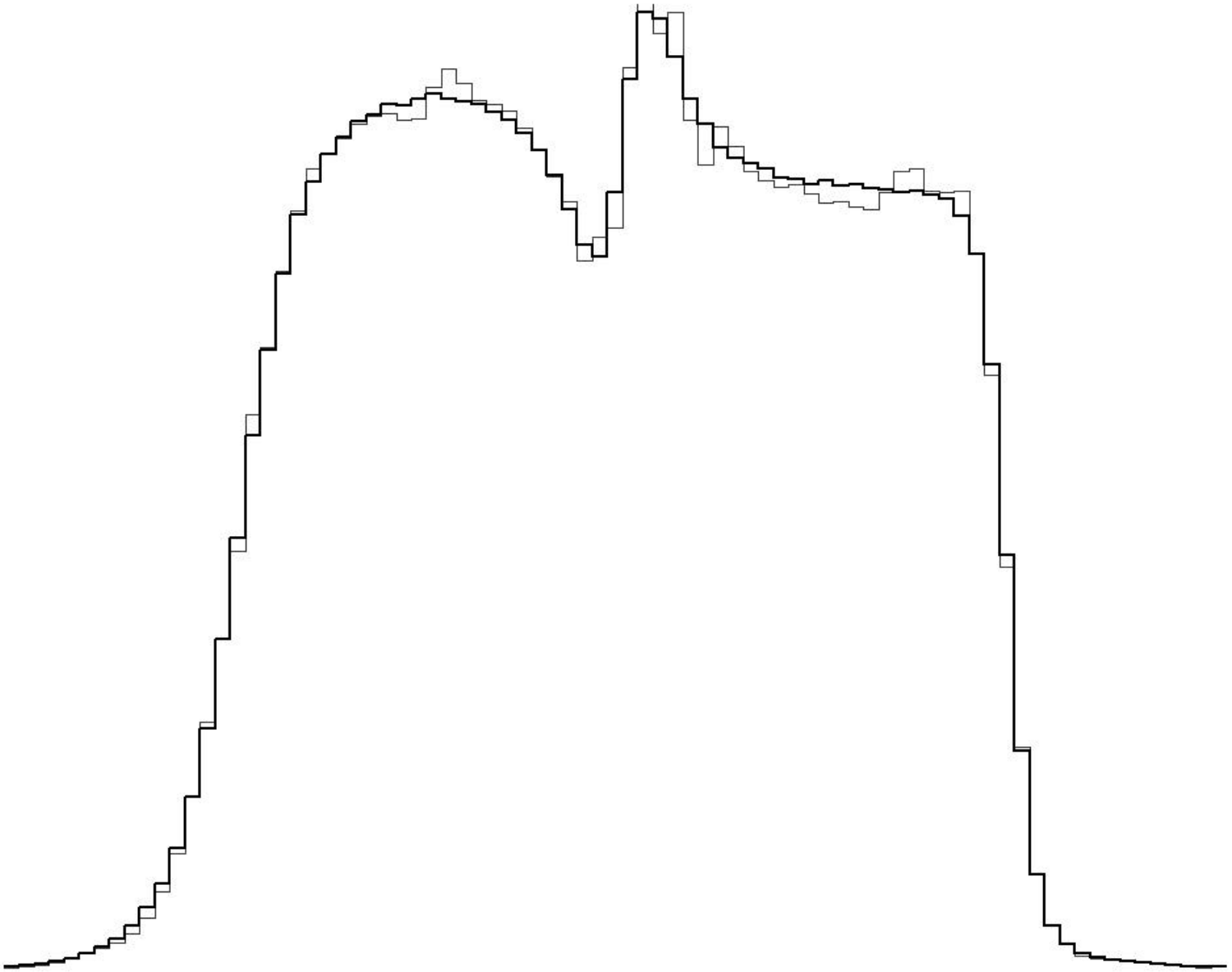
4Me- percentage: 5.0%



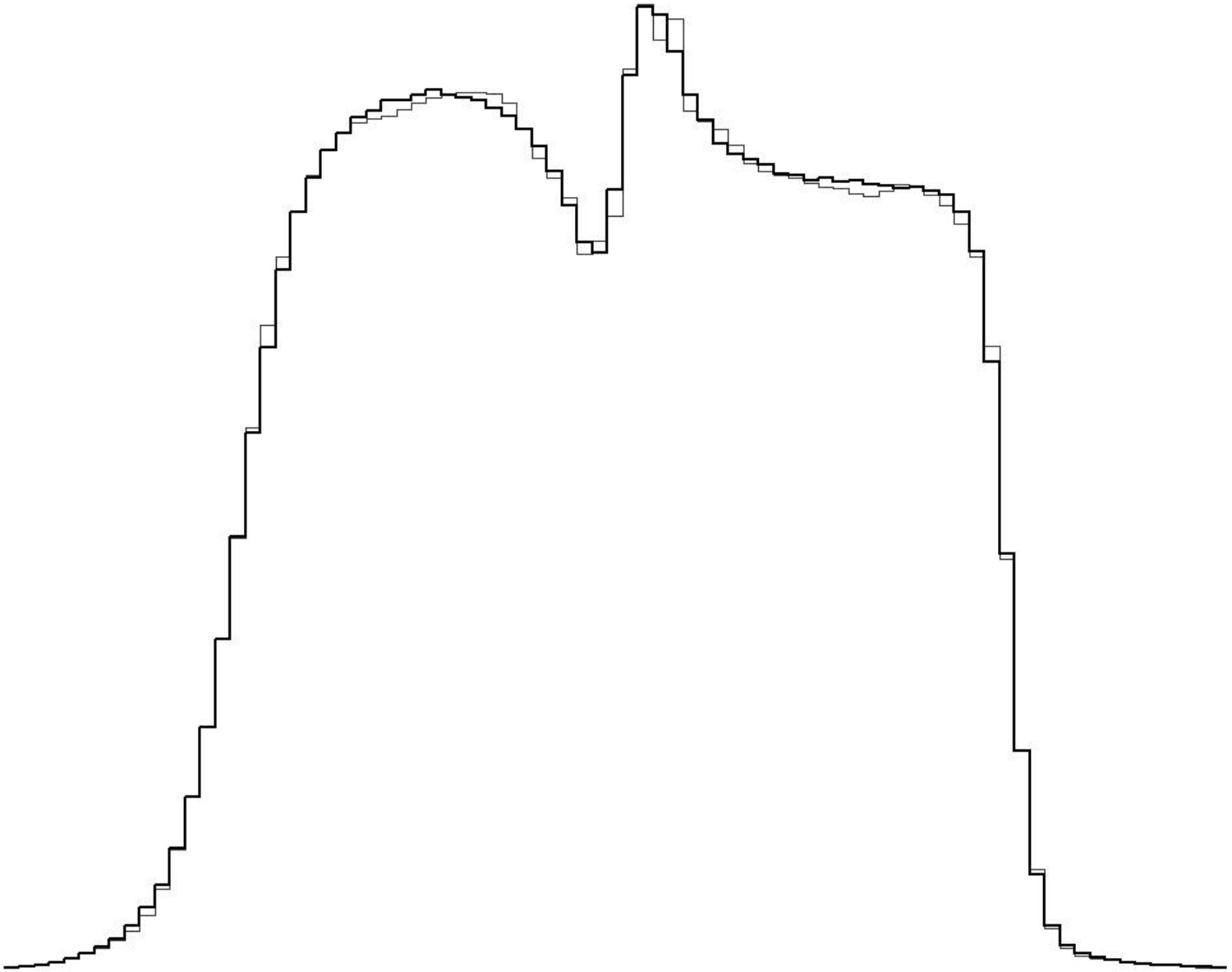
4Me- dose distribution: original



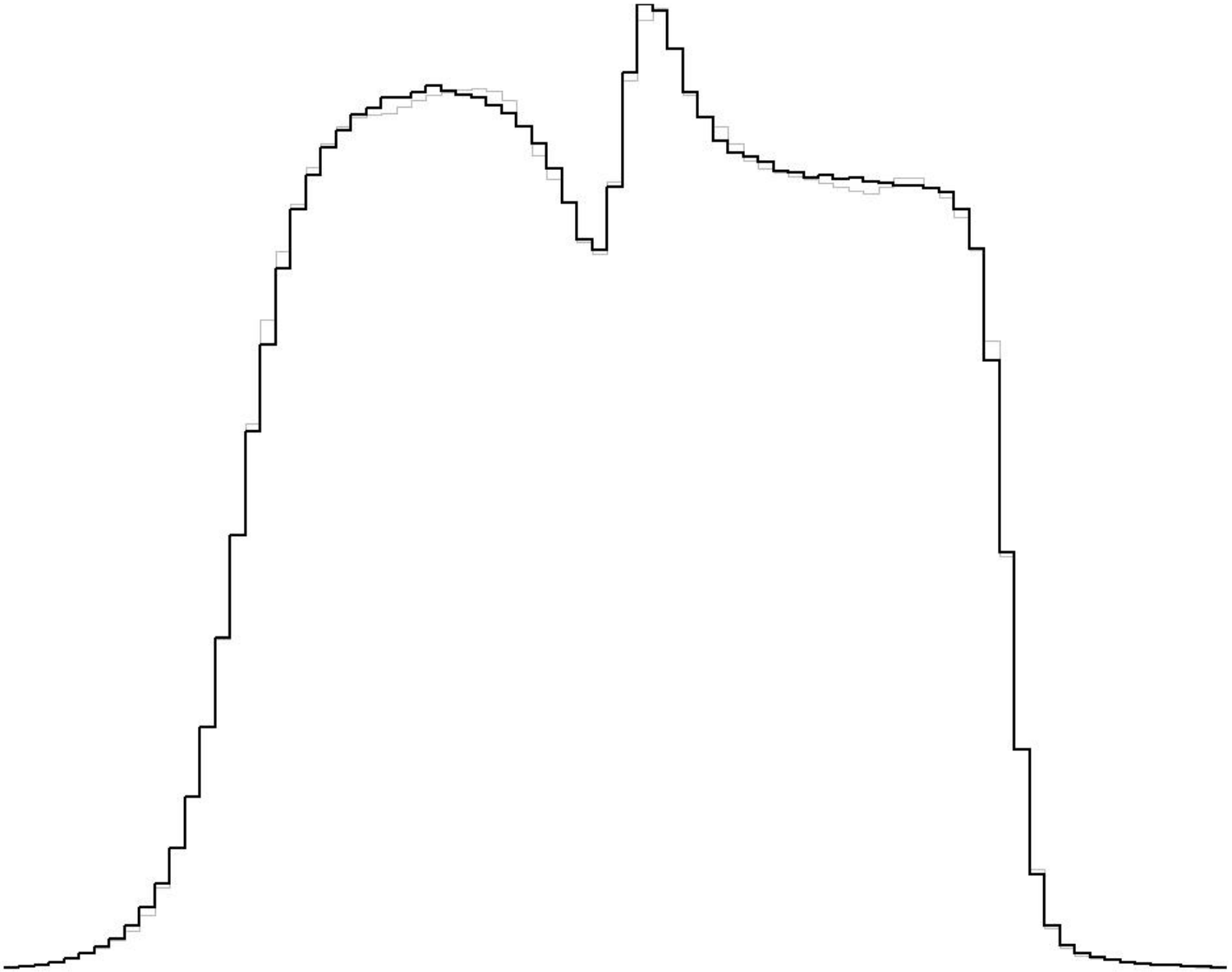
4Me- percentage: 0.05%



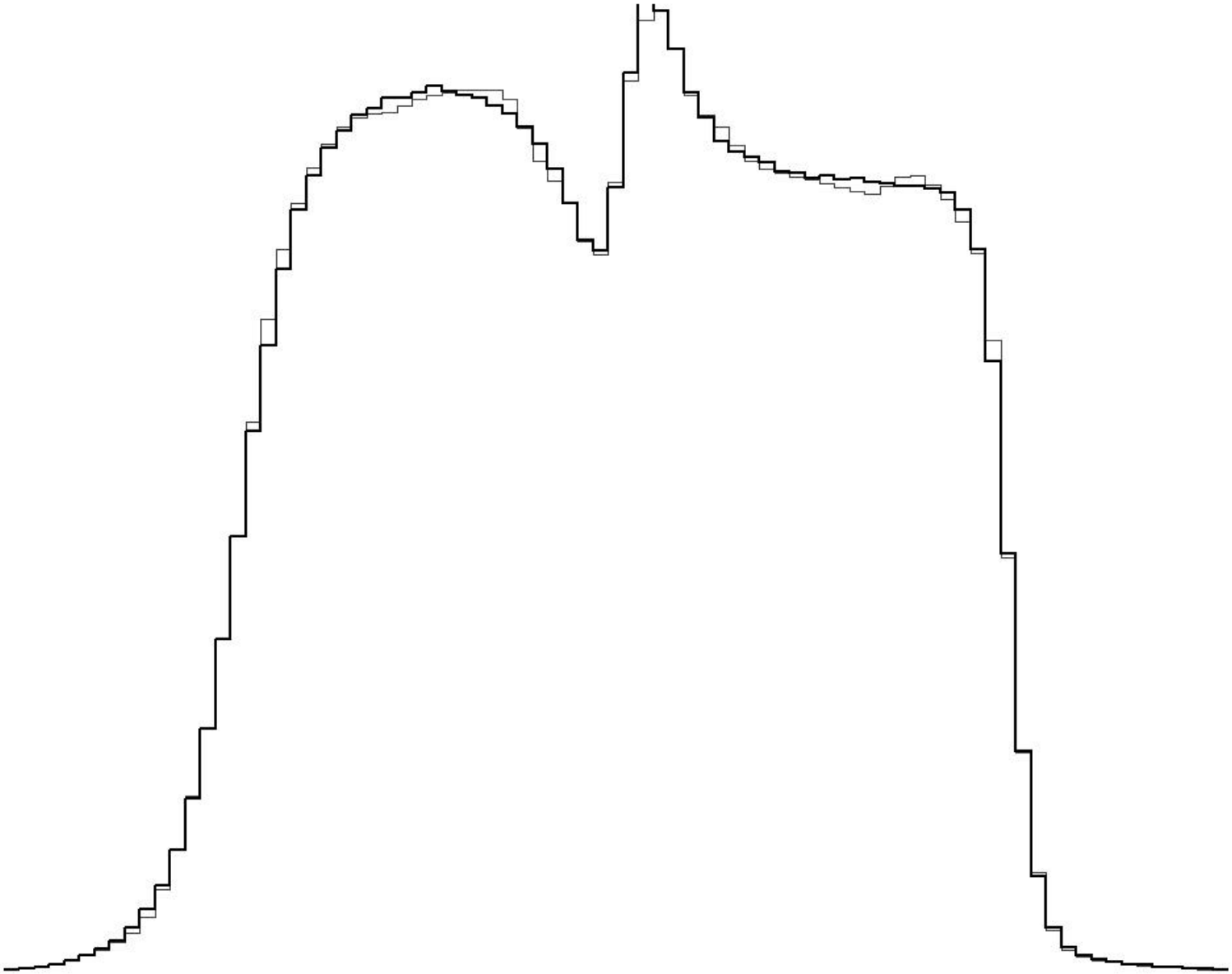
4Me- percentage: 0.1%



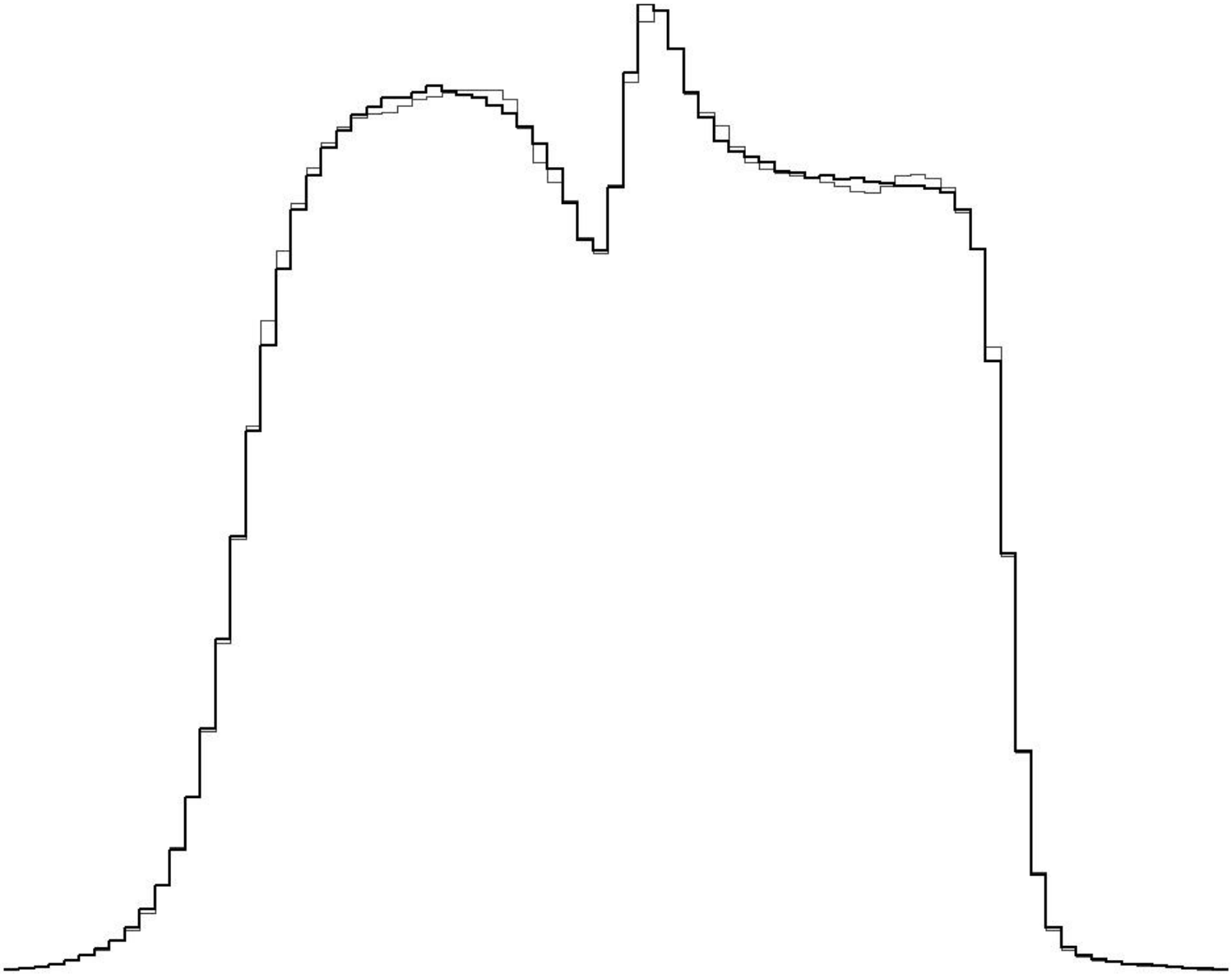
4Me- percentage: 0.15%



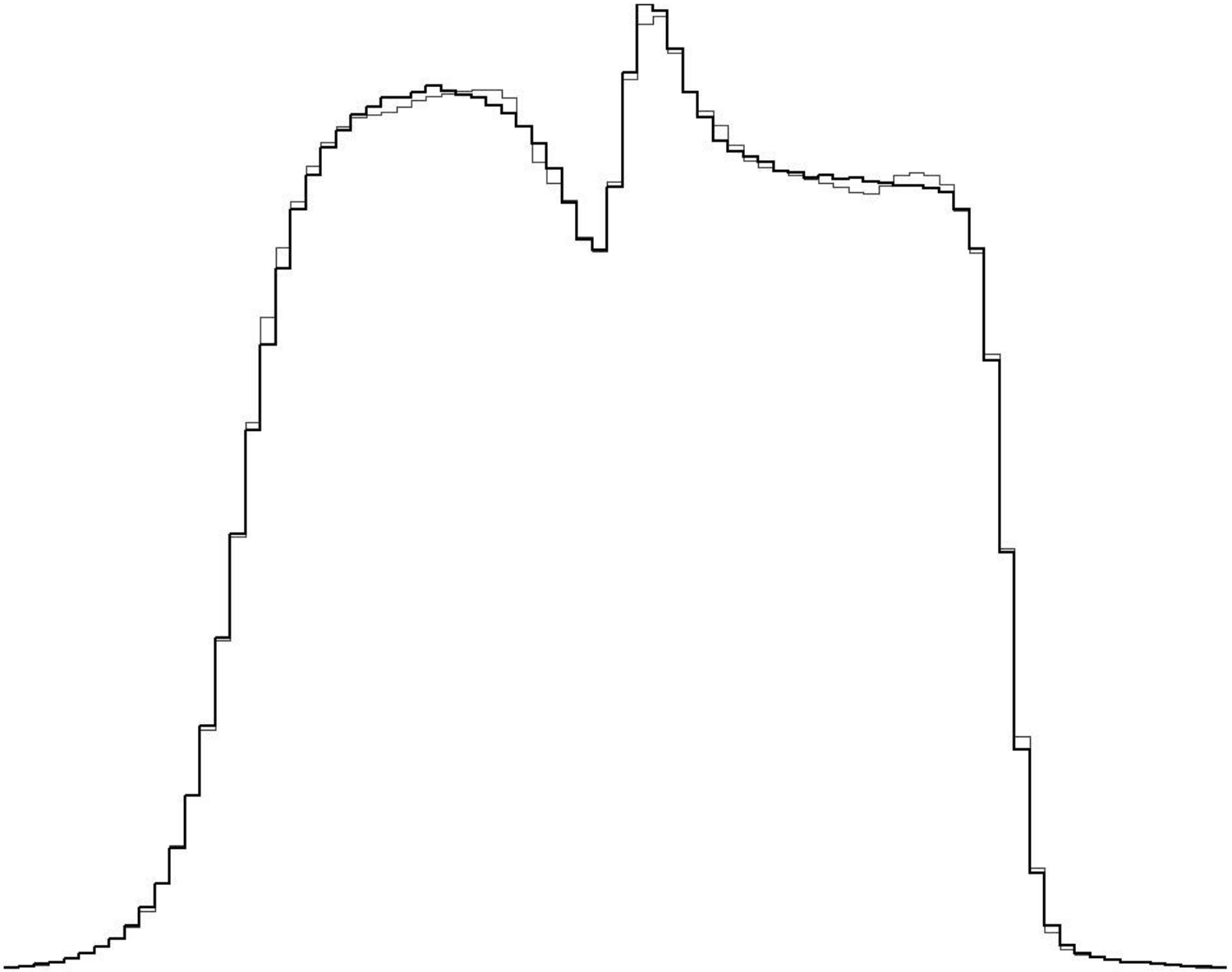
4Me- percentage: 0.18%



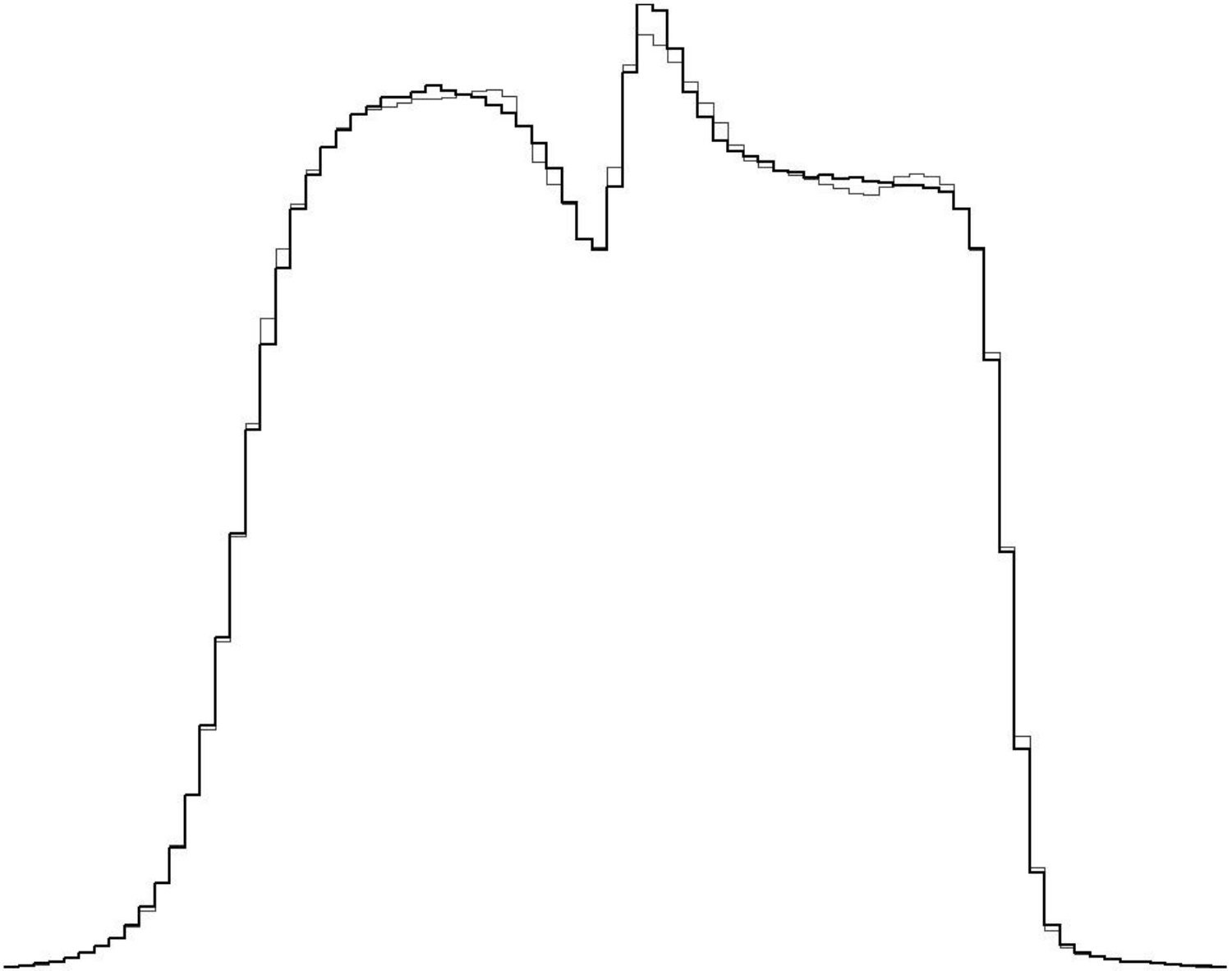
4Me- percentage: 0.2%



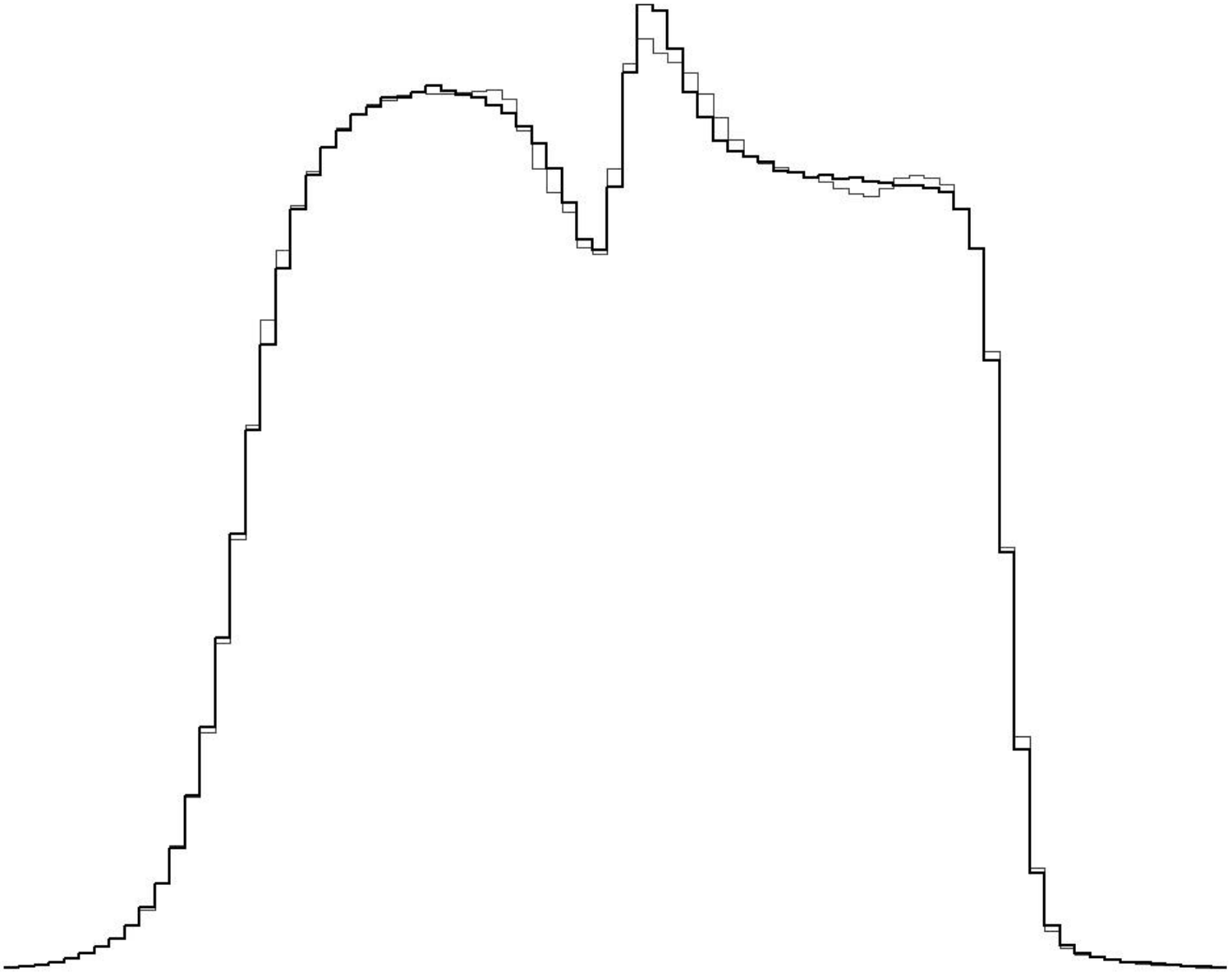
4Me- percentage: 0.25%



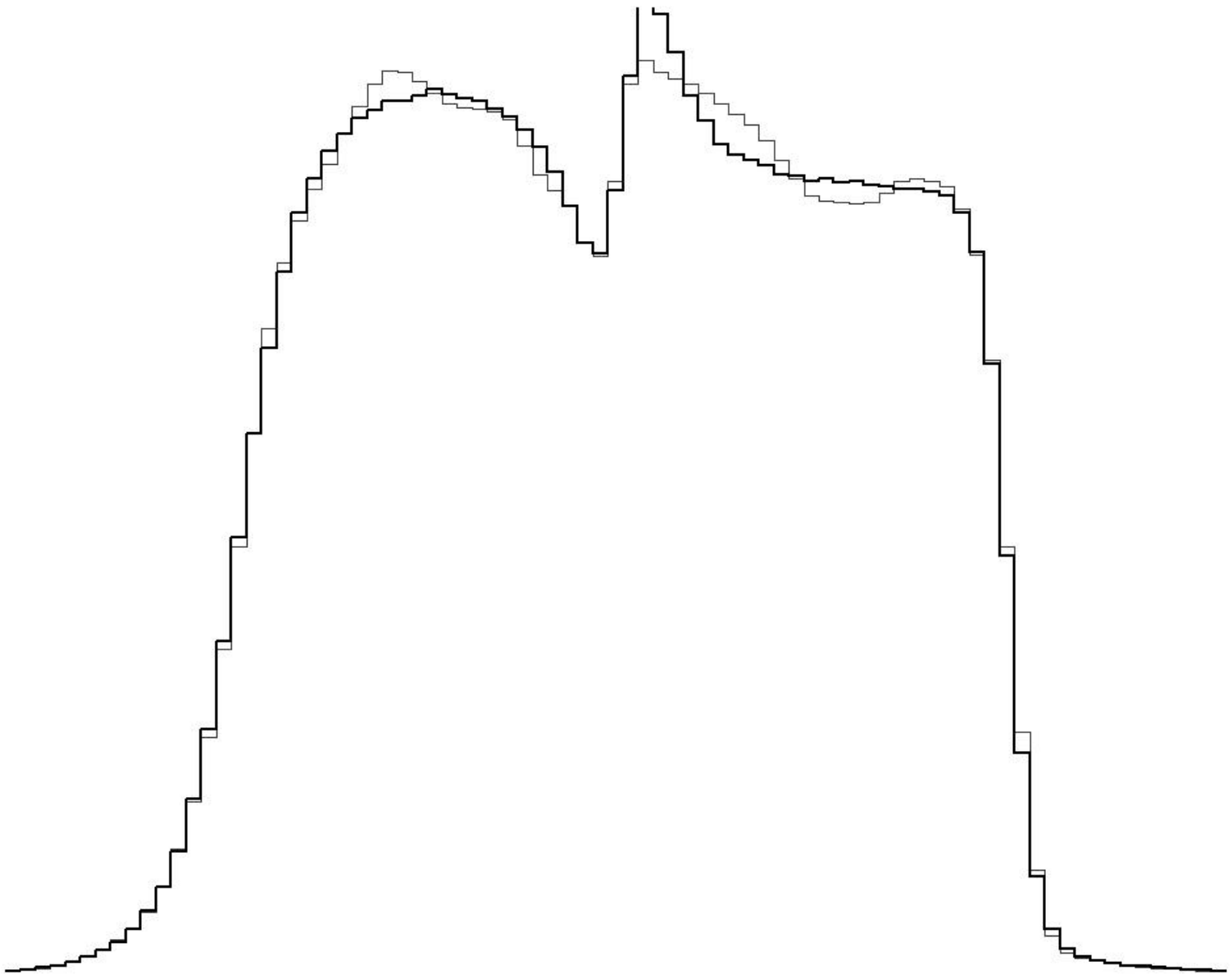
4Me- percentage: 0.3%



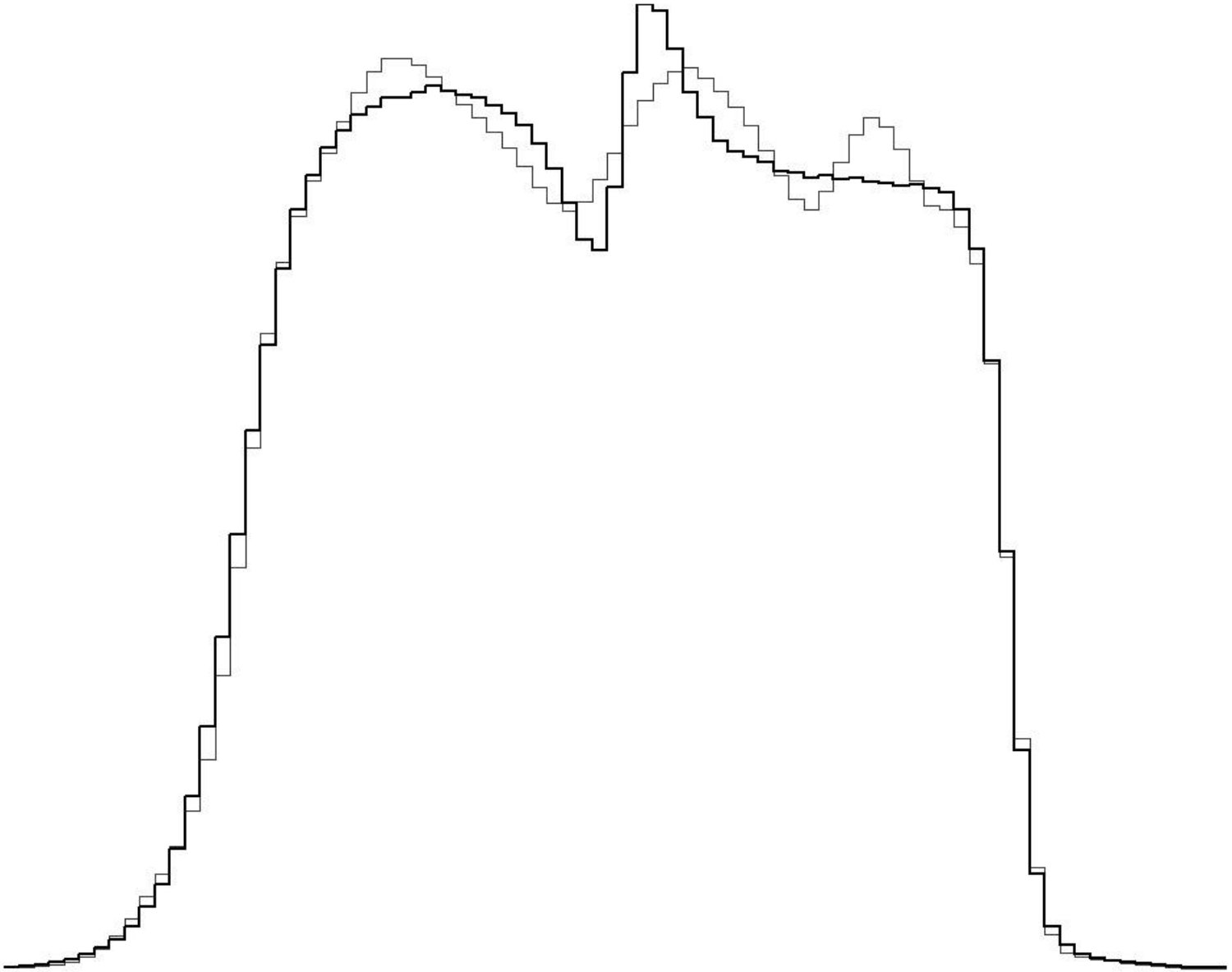
4Me- percentage: 0.35%



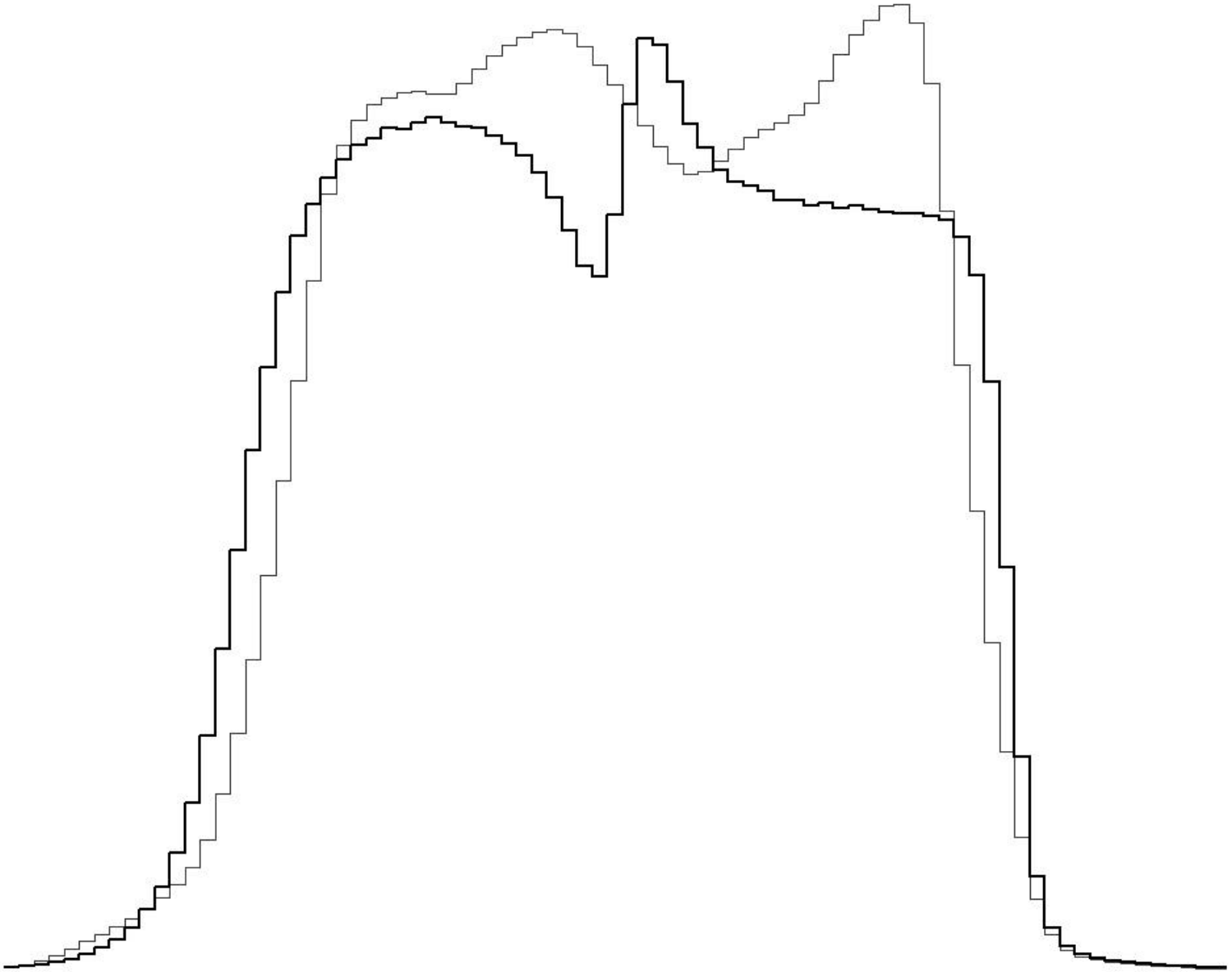
4Me- percentage: 0.4%



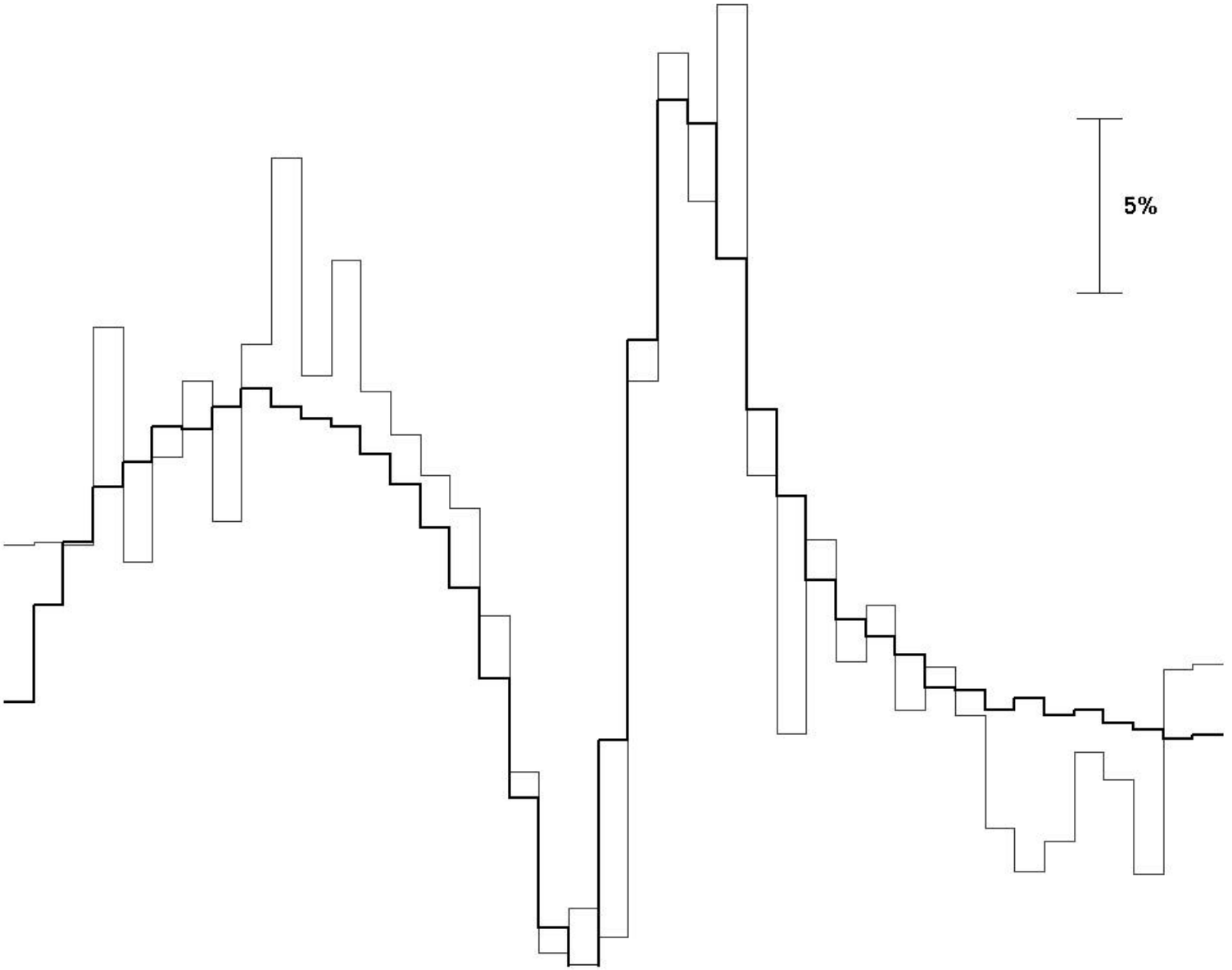
4Me- percentage: 0.5%



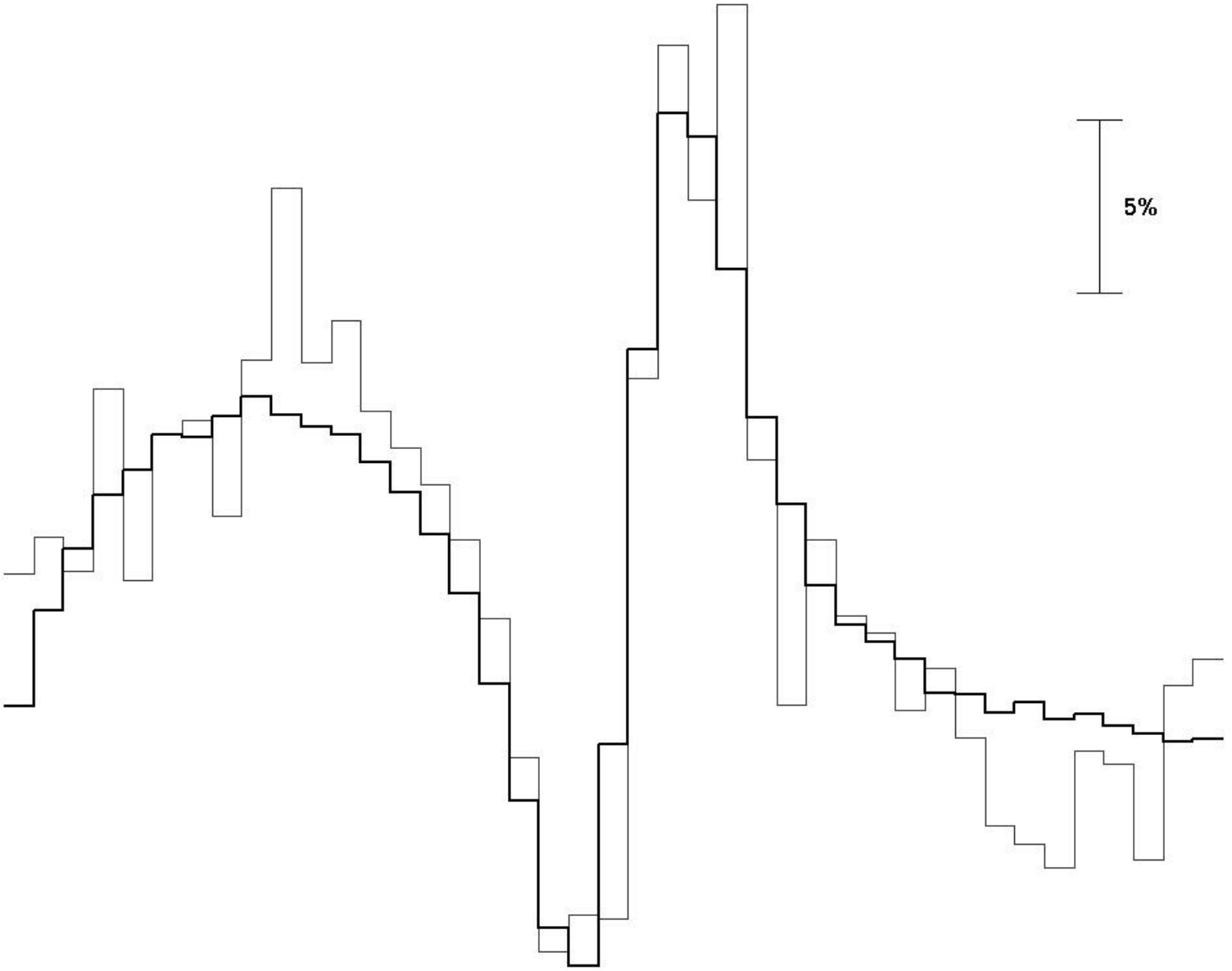
4Me- percentage: 1.0%



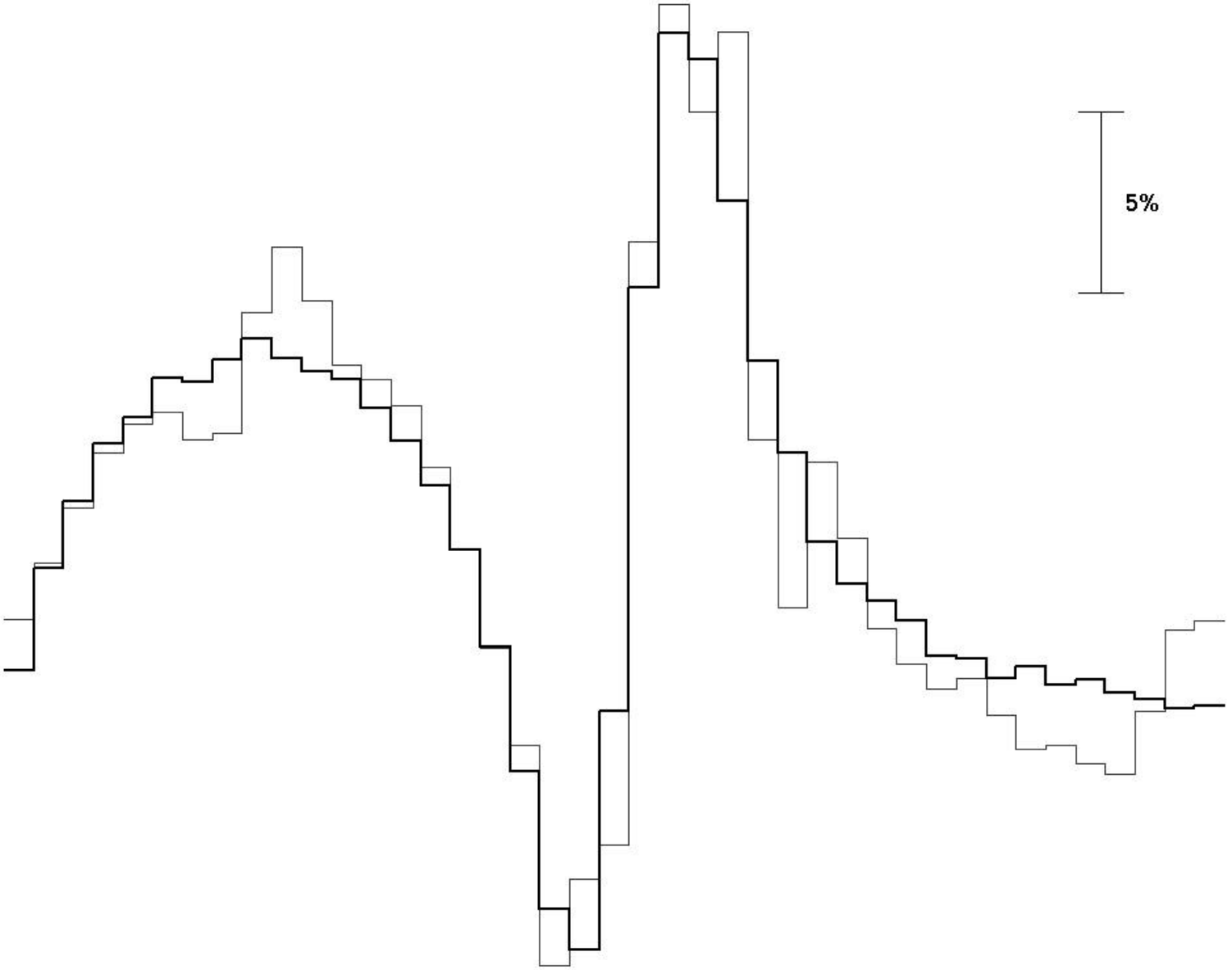
4Me- percentage: 5.0%



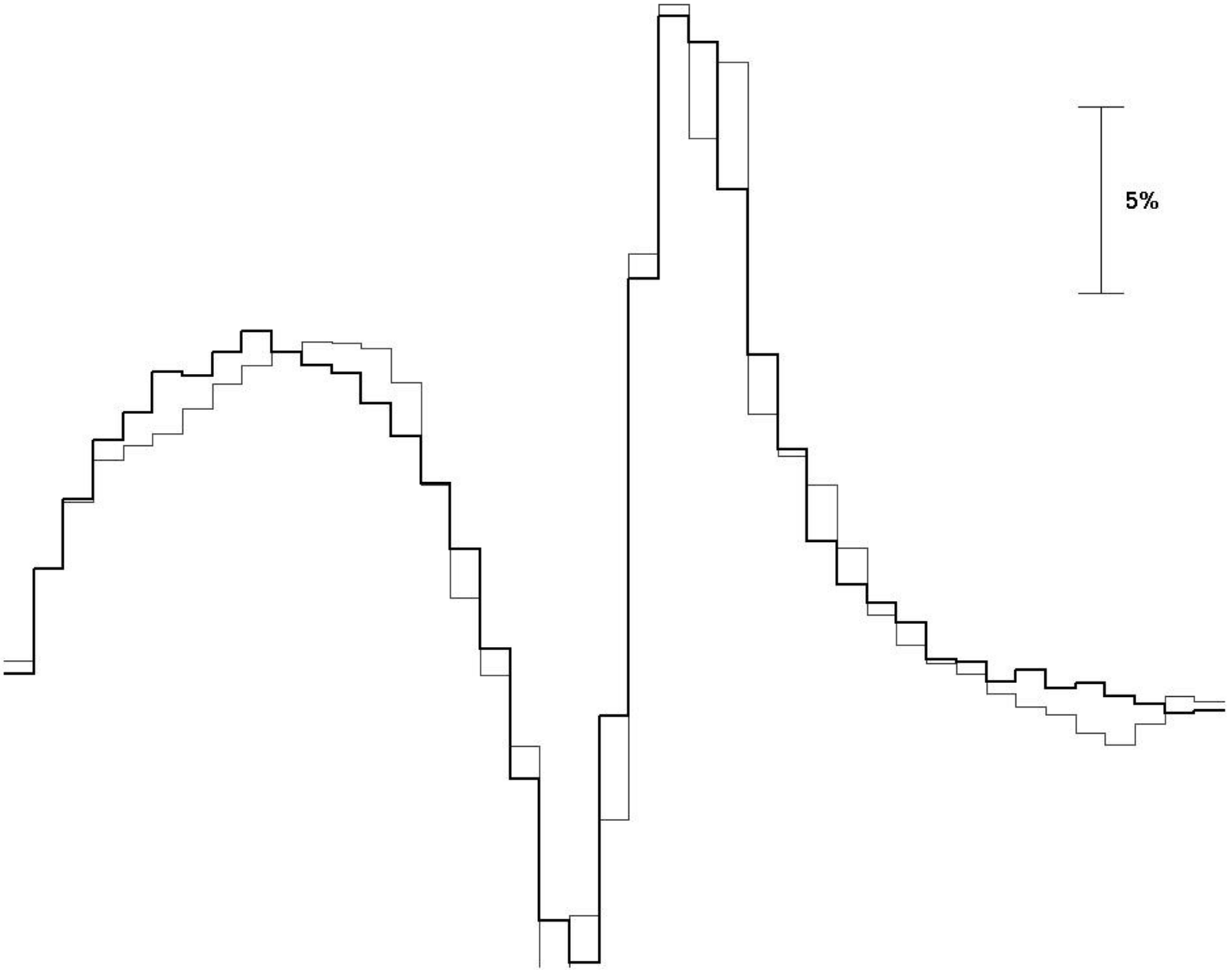
4Me- percentage: original



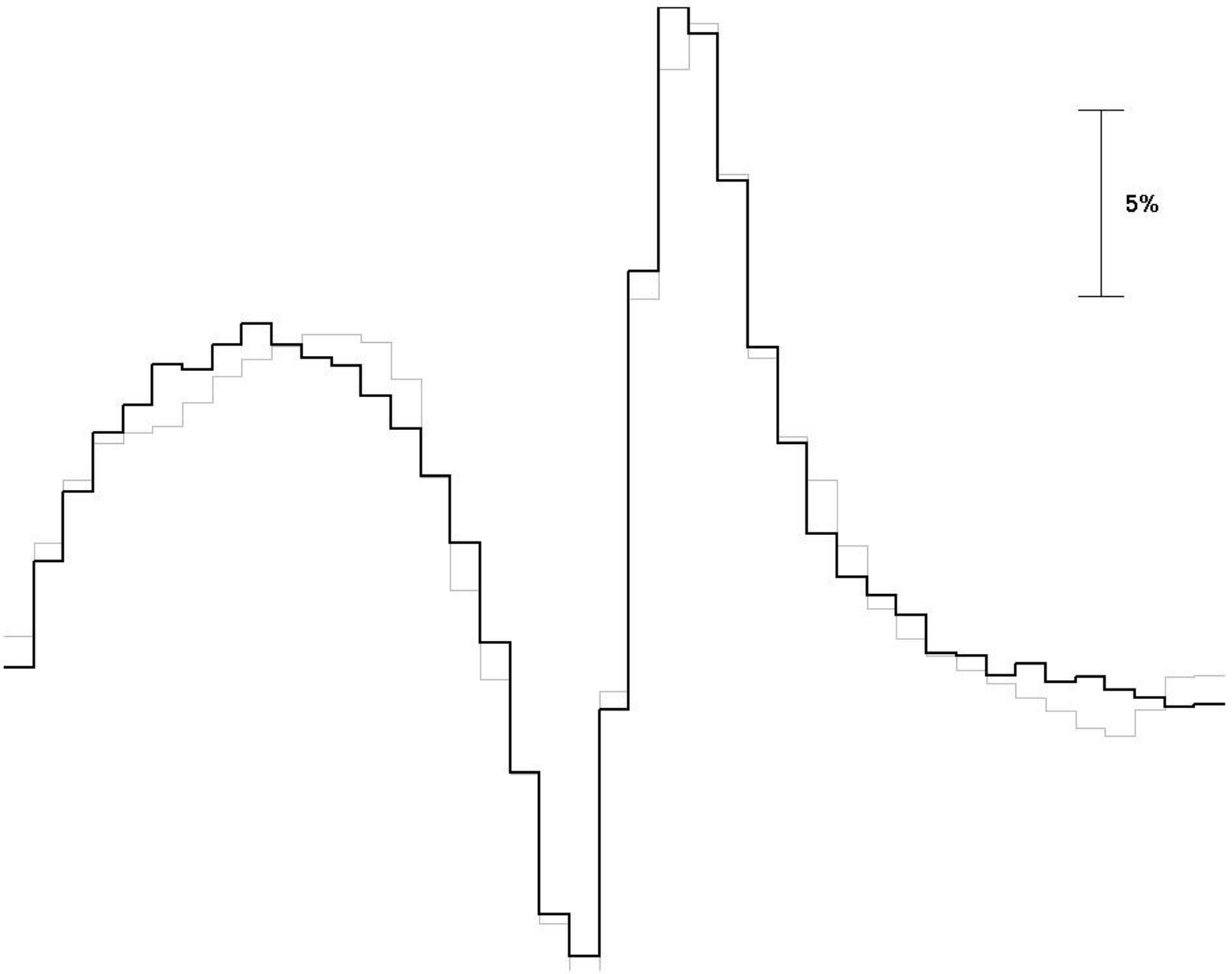
4Me- percentage: 0.05%



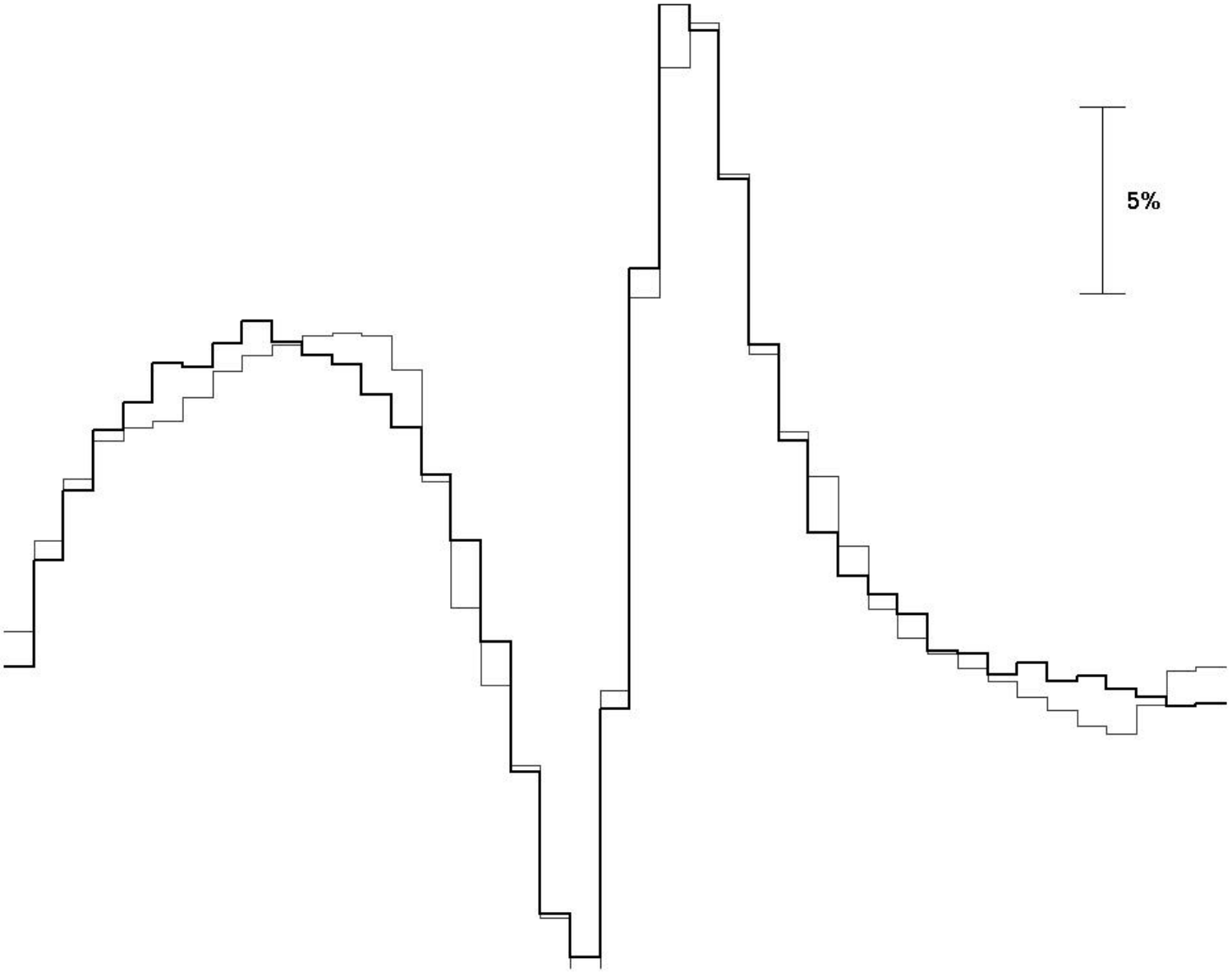
4Me- percentage: 0.1%



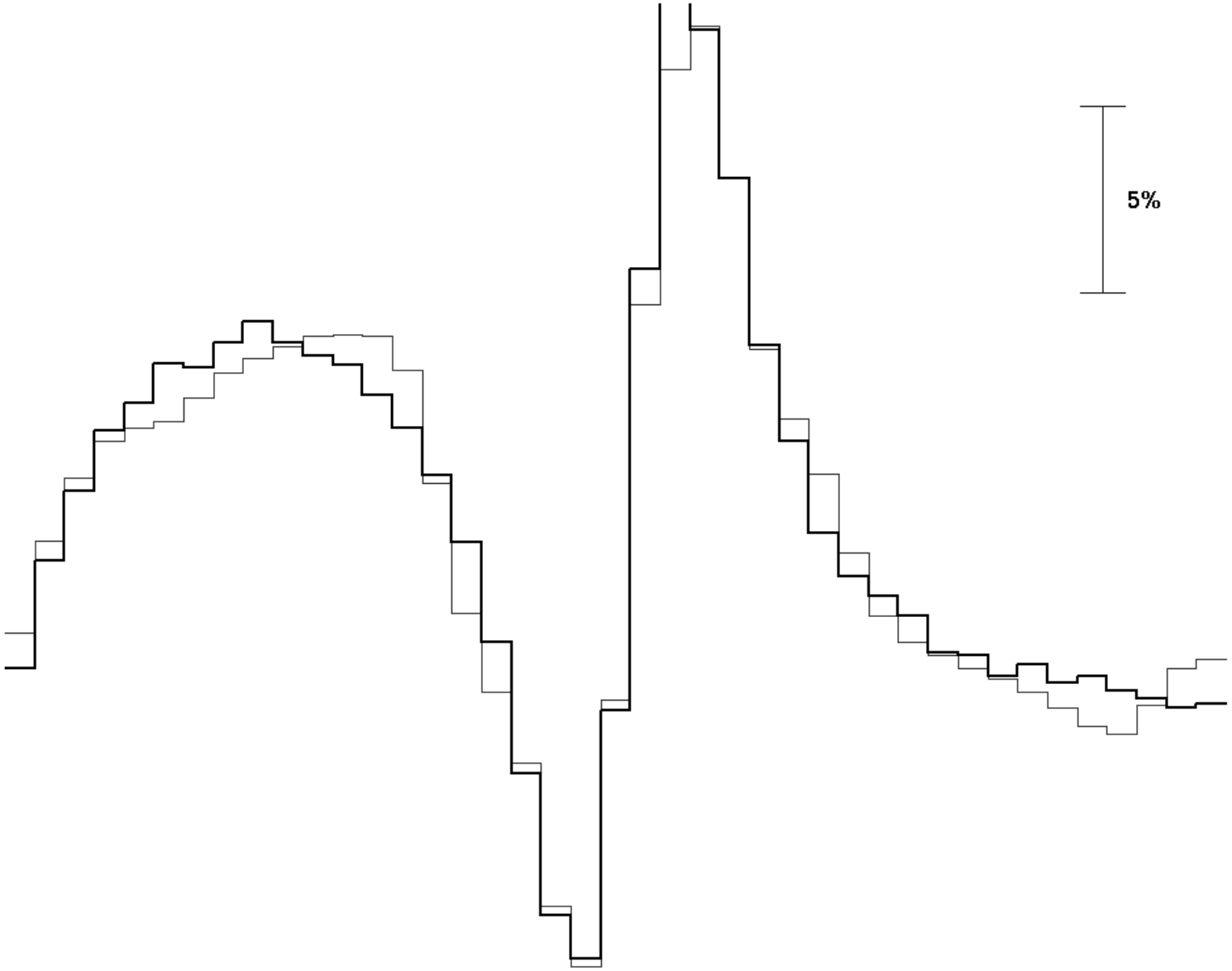
4Me- percentage: 0.15%



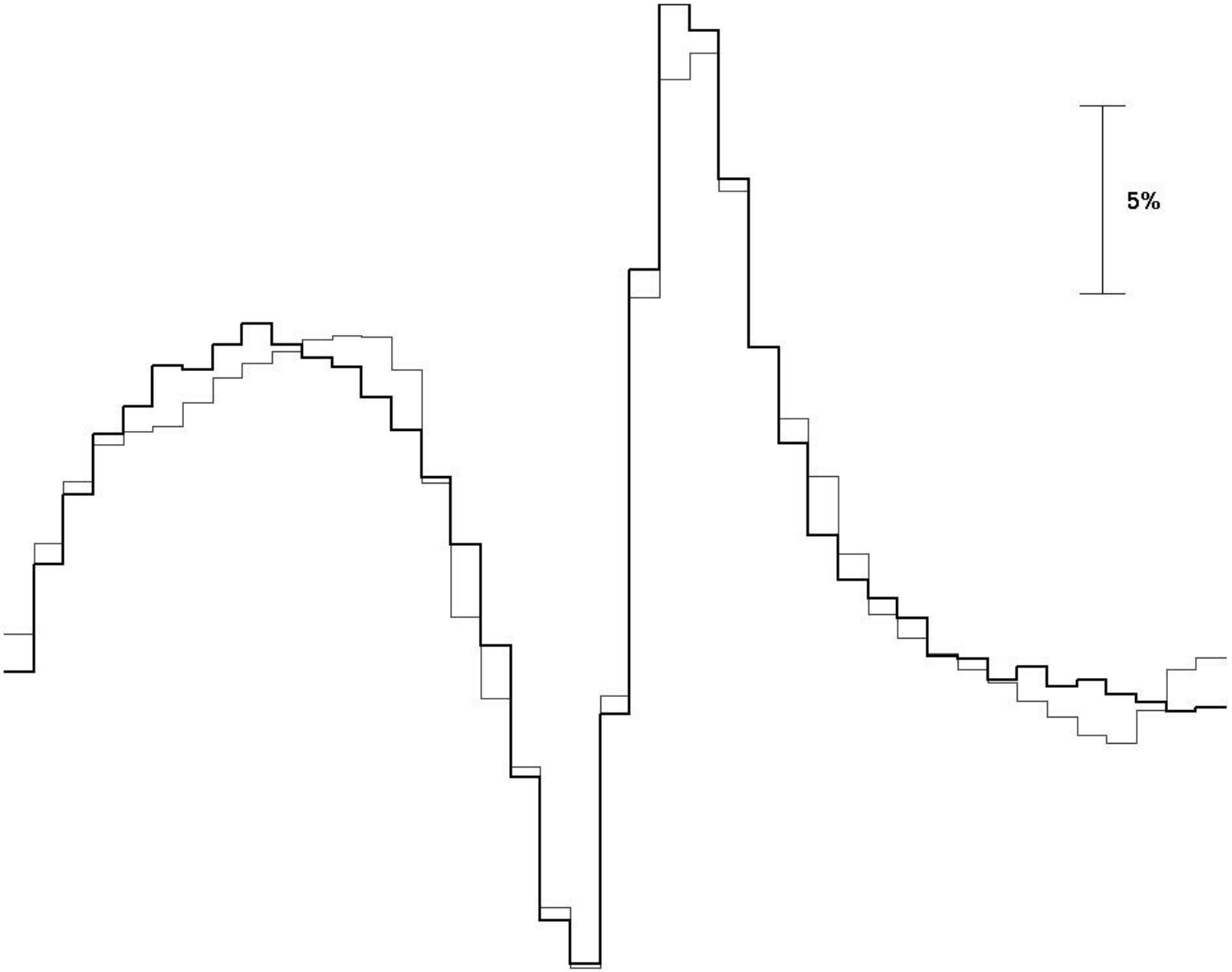
4Me- percentage: 0.18%



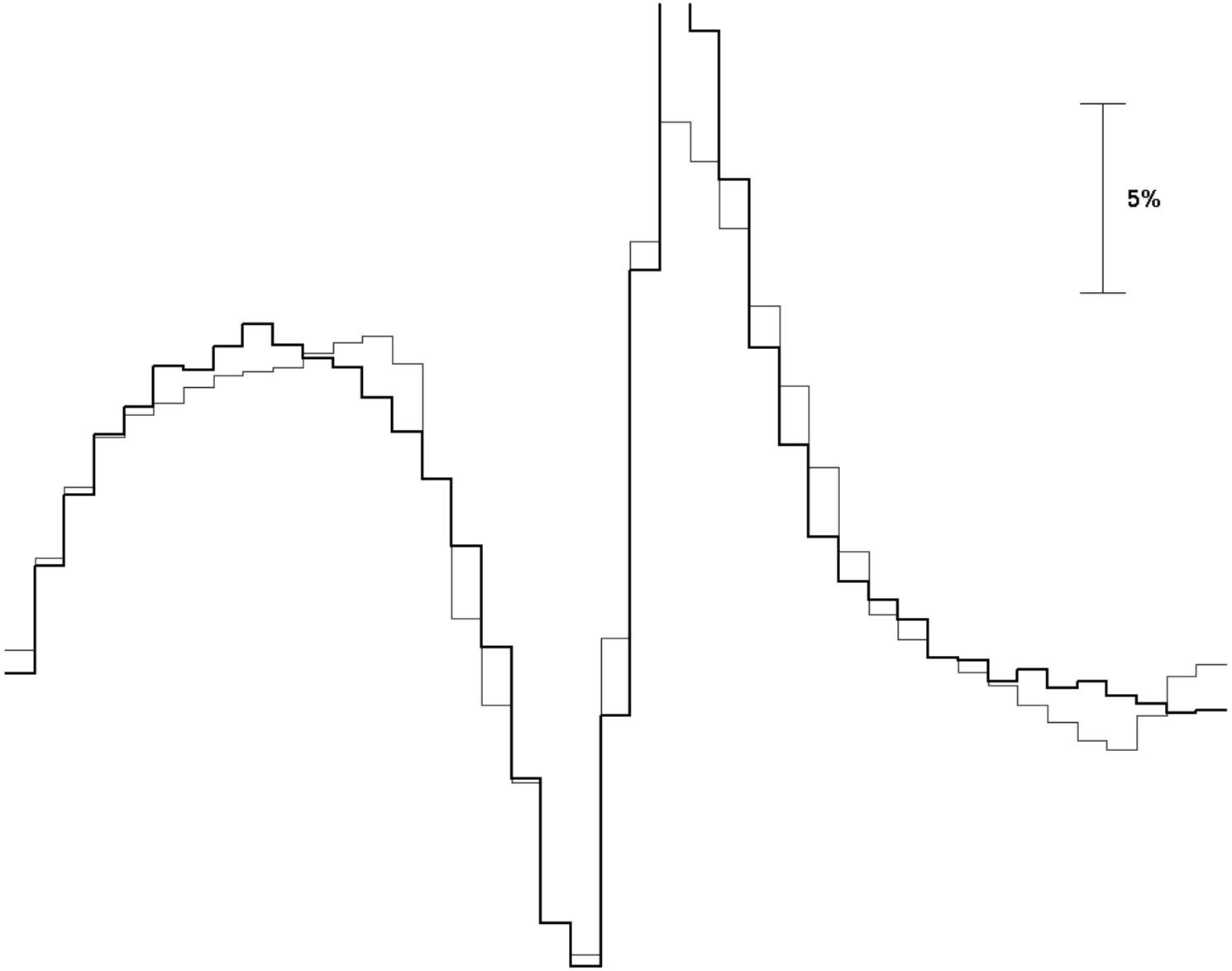
4Me- percentage: 0.2%



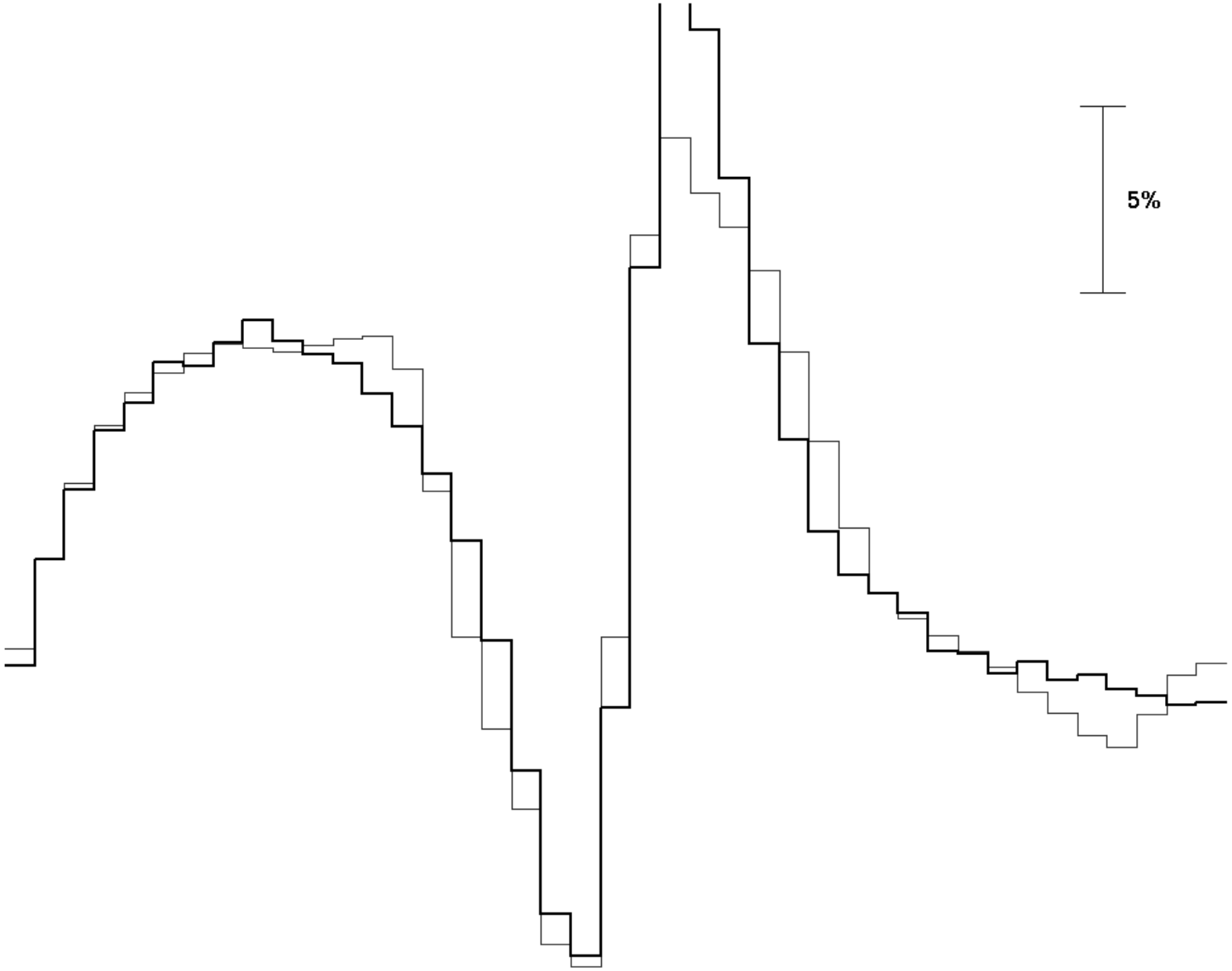
4Me- percentage: 0.25%



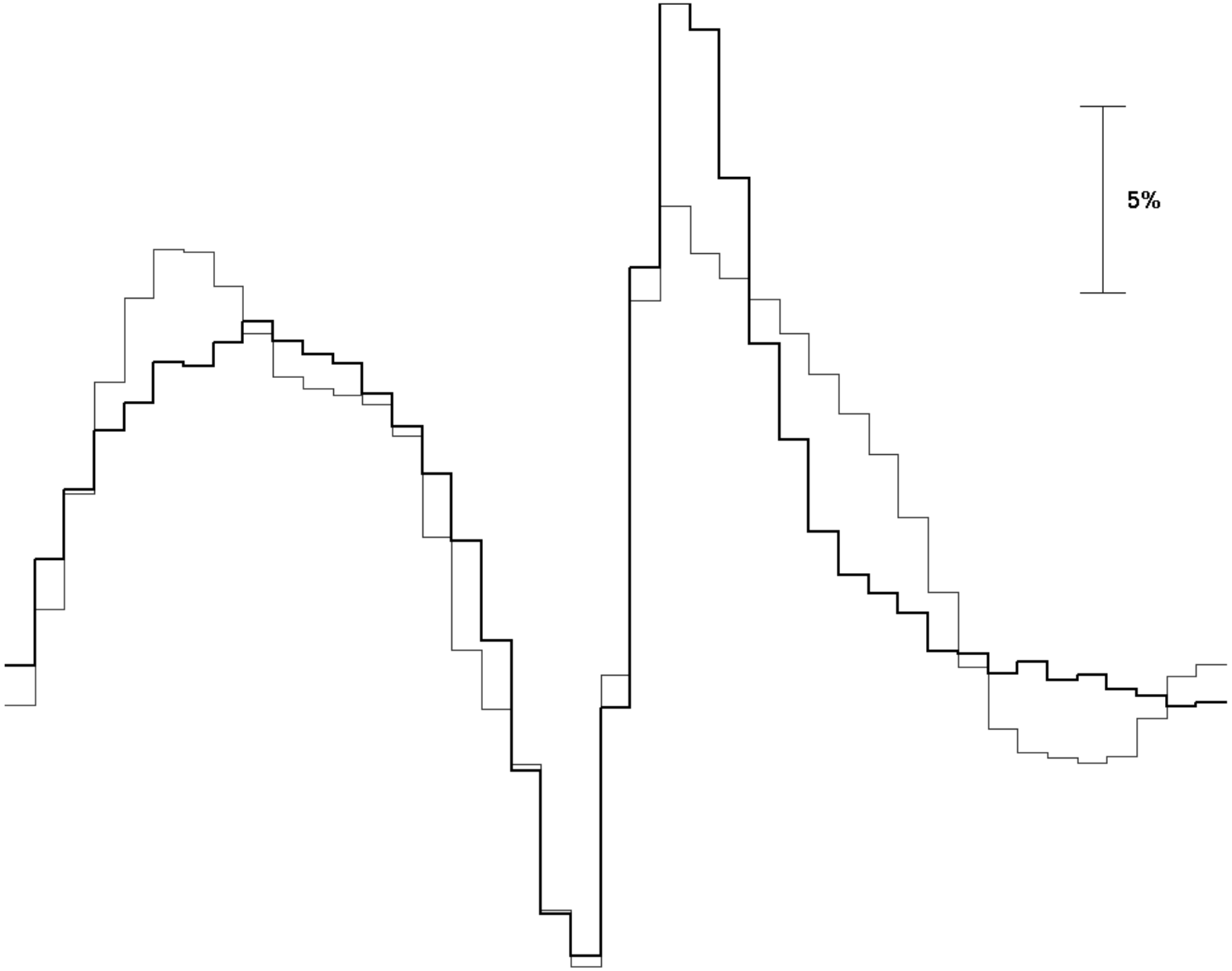
4Me- percentage: 0.3%



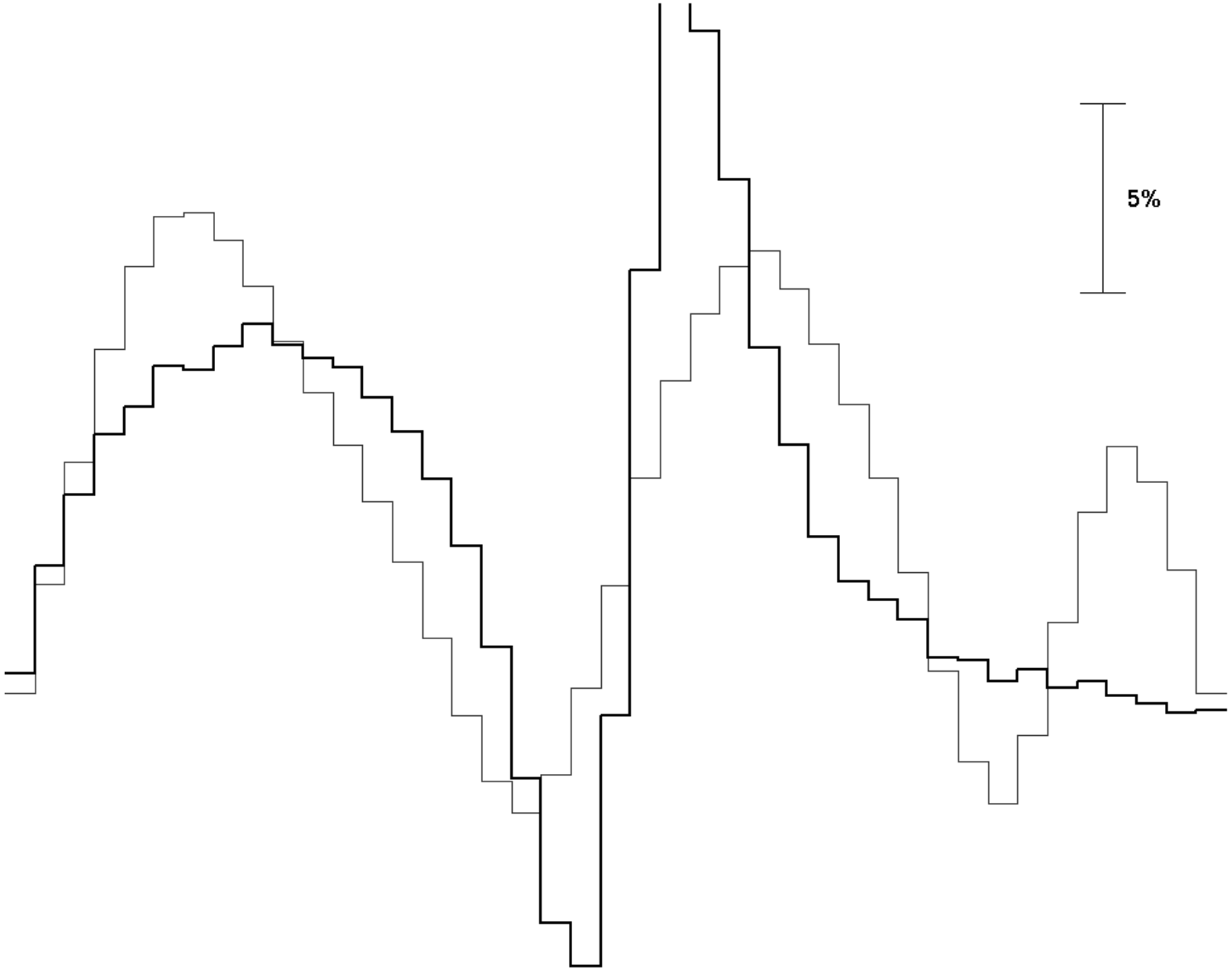
4Me- percentage: 0.35%



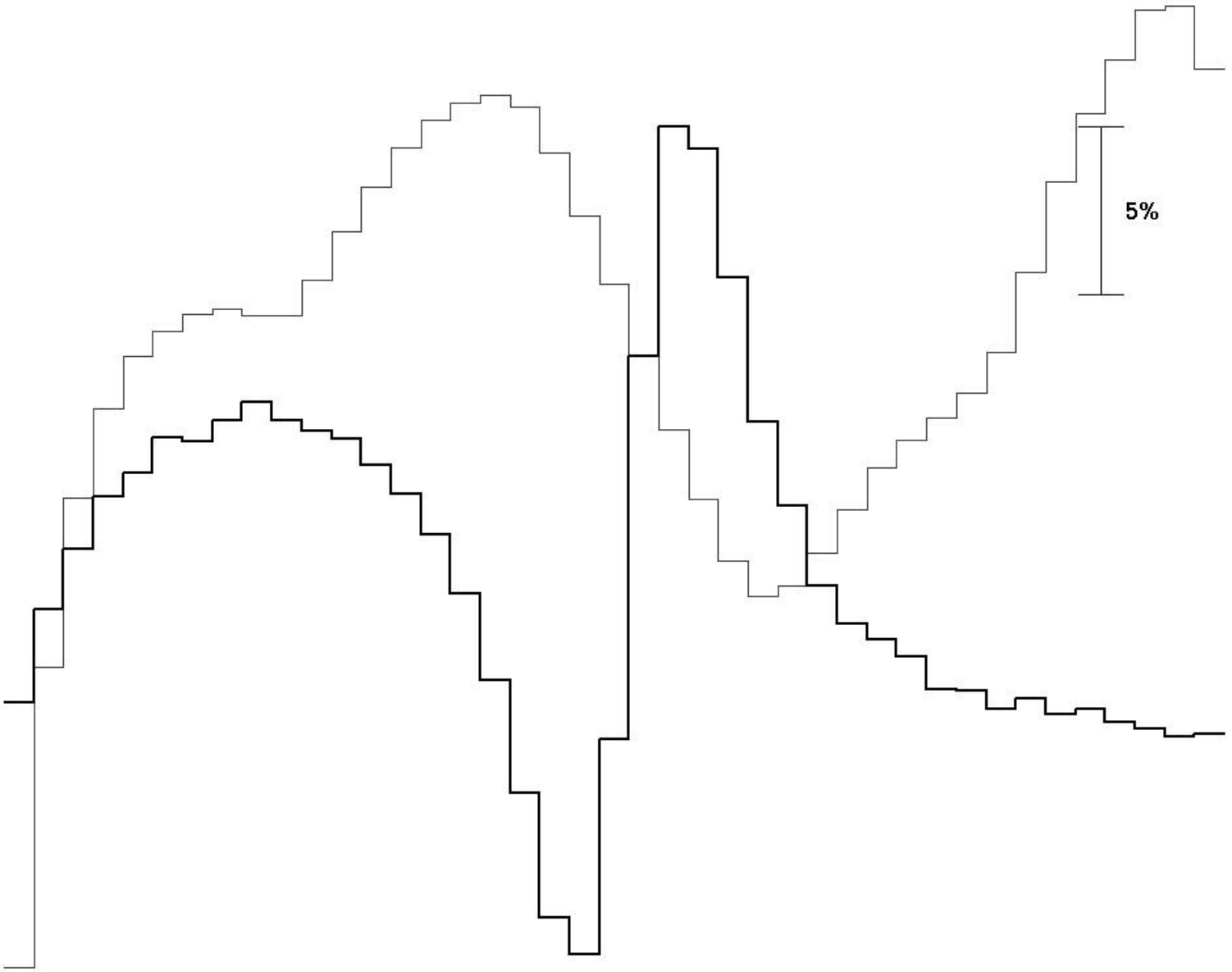
4Me- percentage: 0.4%



4Me- percentage: 0.5%



4Me- percentage: 1.0%



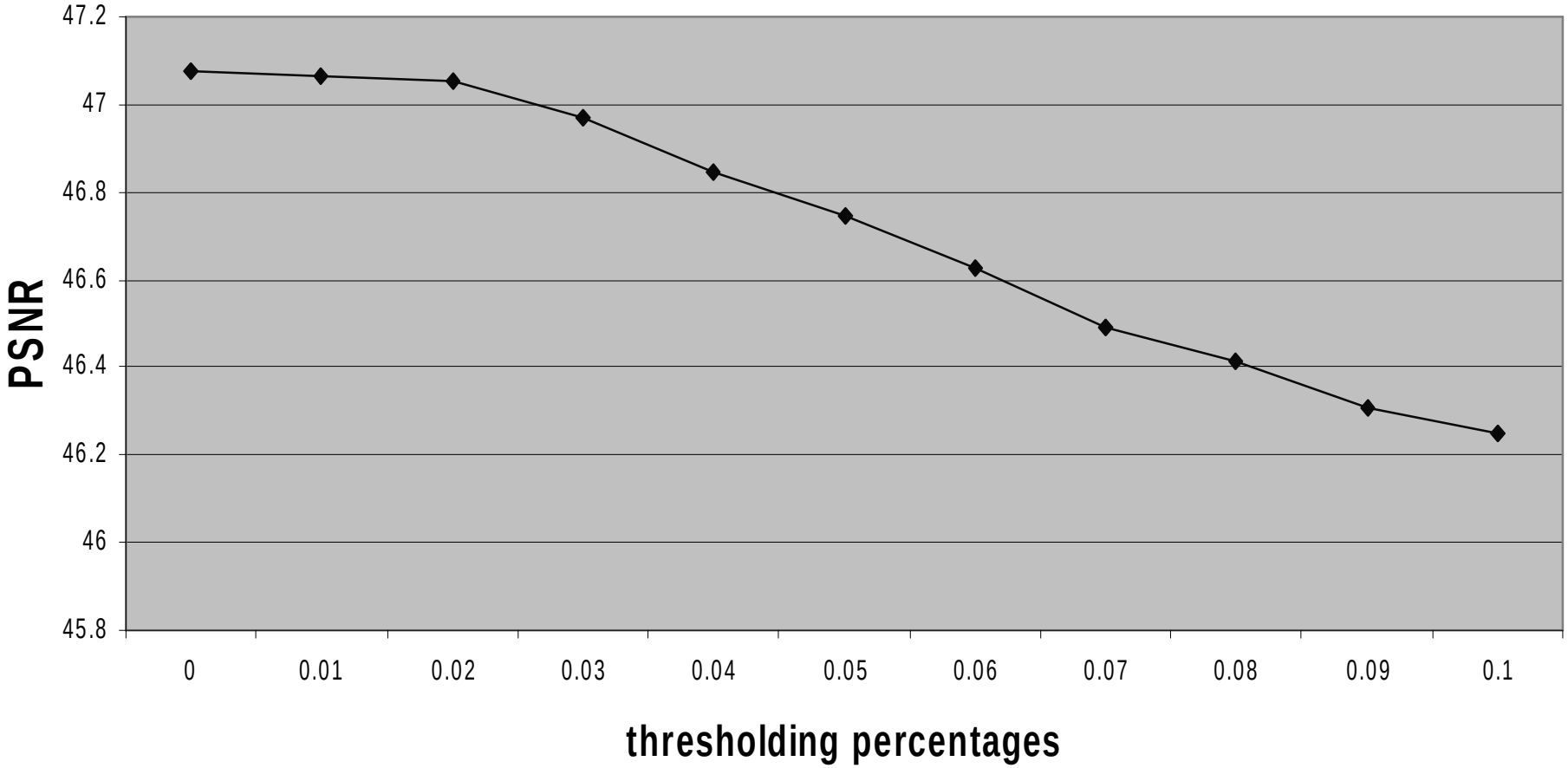
4Me- percentage: 5.0%

Divide and conquer?

- Is it more efficient to de-noise smaller dose distributions and add them, or to de-noise directly a bigger dose distribution?
- Experiment: De-noising 2 independent 2Me- dose distributions, adding them together and trying to de-noise the sum to get the best possible PSNR.

Divide and conquer?

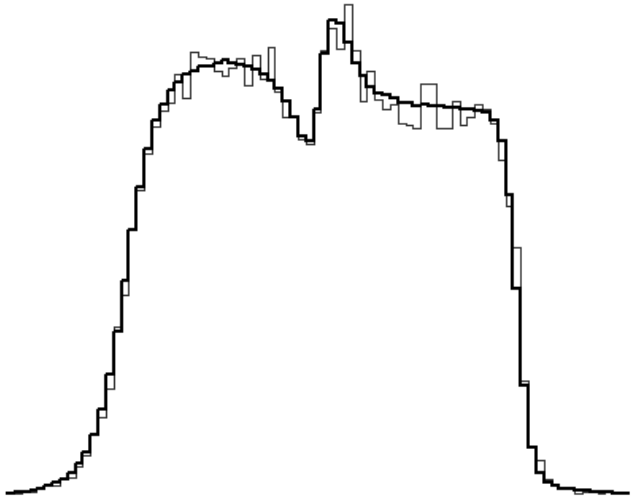
best PSNR after denoising the sum of two 2Me- runs denoised separately with percentages going from 0% to 0.1%



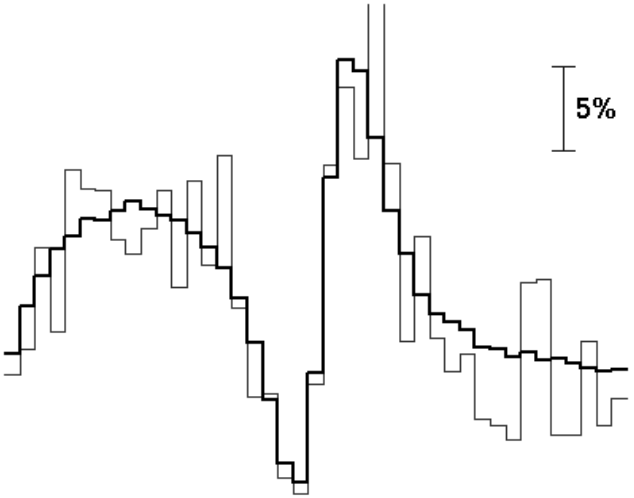
Results :

Statistics on a large number of independent samples showed that the thresholding percentage leading to the best PSNR is a function of the size of the original dose distribution

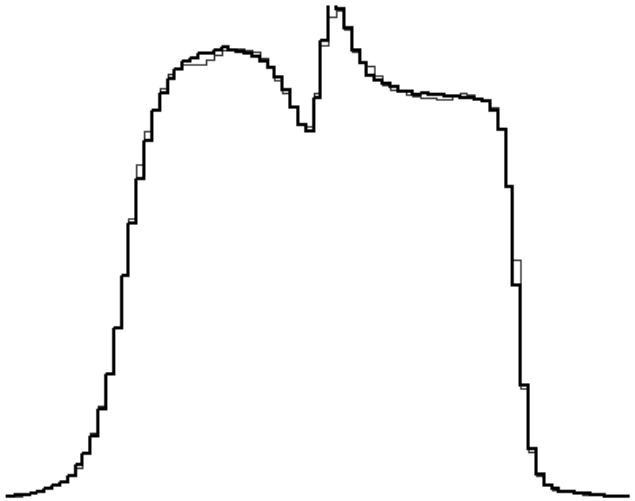
Low density



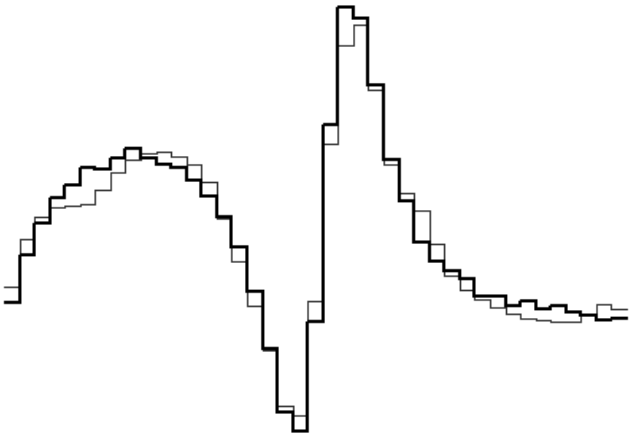
Low density (detail)



Denoised

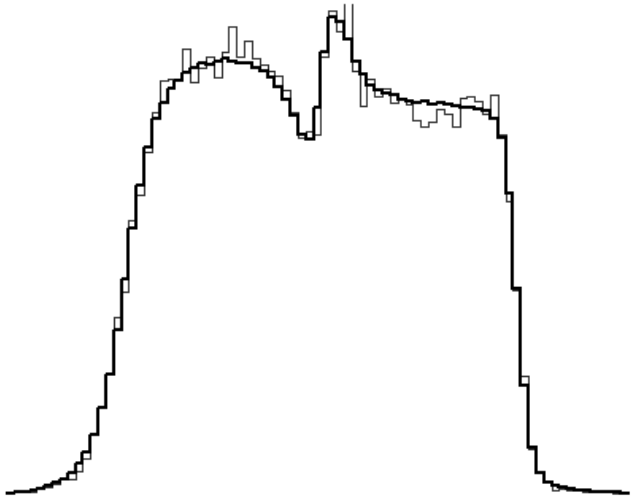


Denoised (detail)

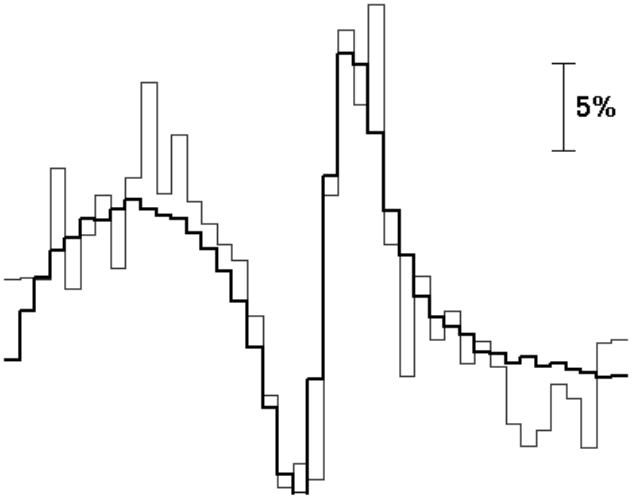


2Me- dose distribution

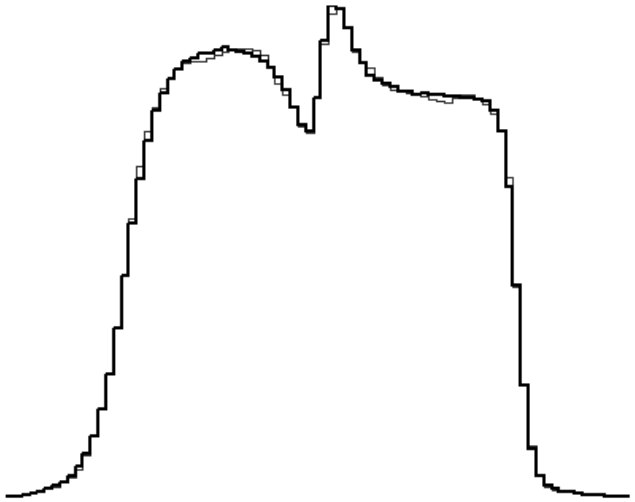
Low density



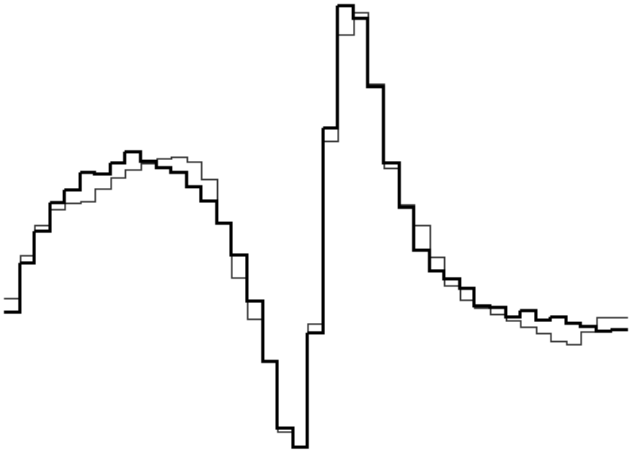
Low density (detail)



Denoised

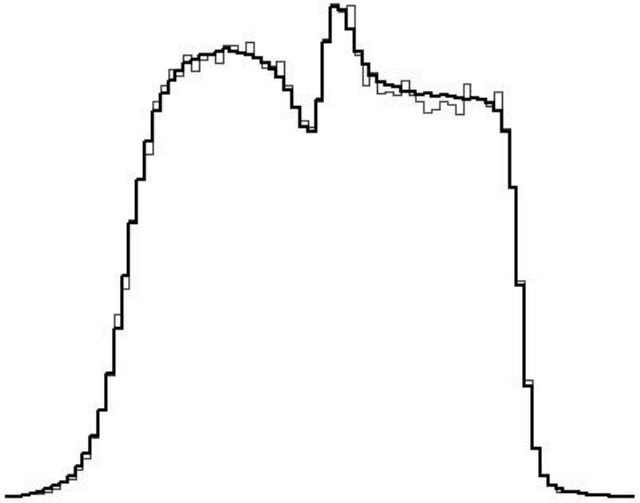


Denoised (detail)

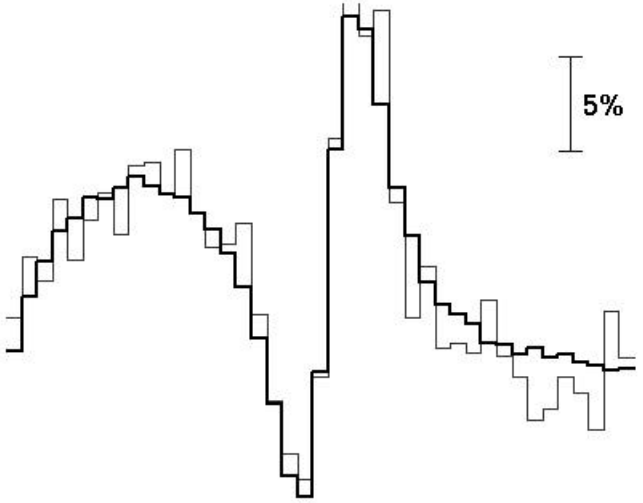


4Me- dose distribution

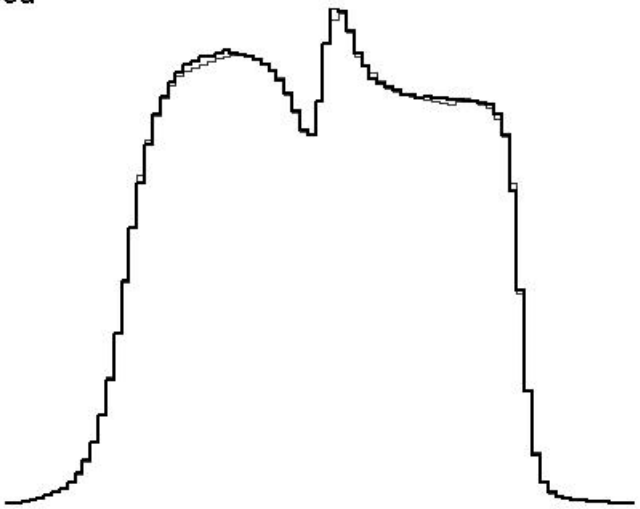
Low density



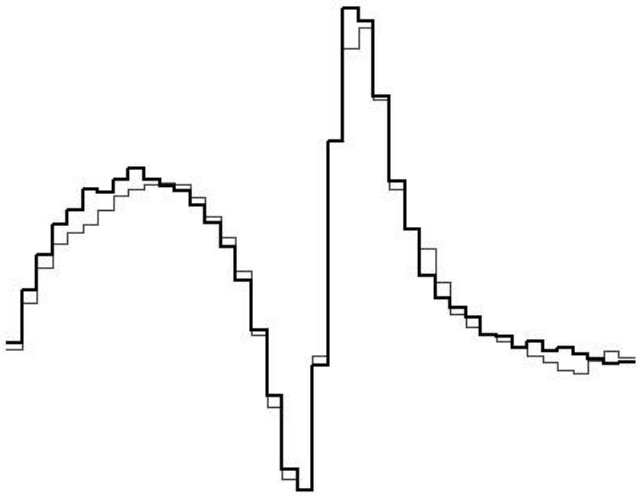
Low density (detail)



Denoised

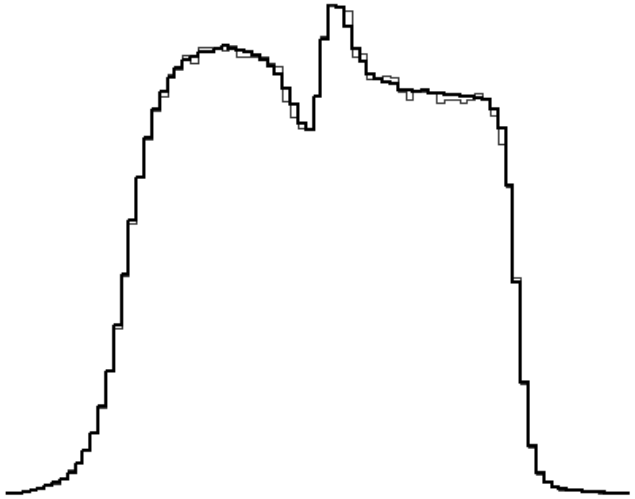


Denoised (detail)

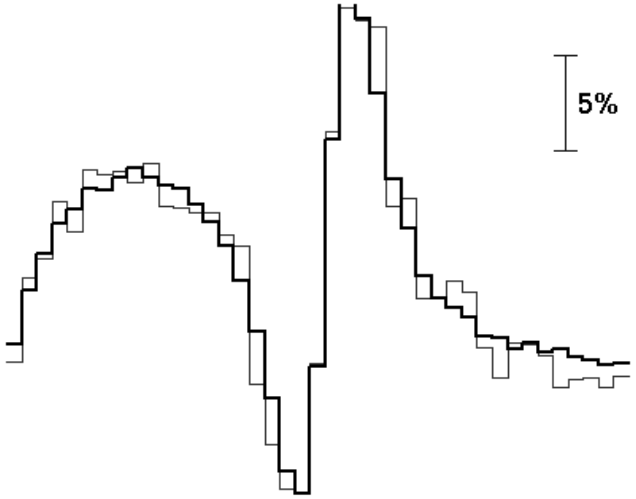


8Me- dose distribution

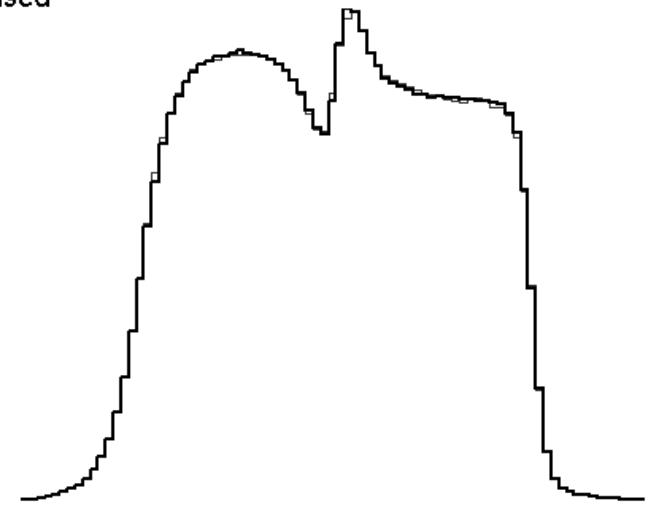
Low density



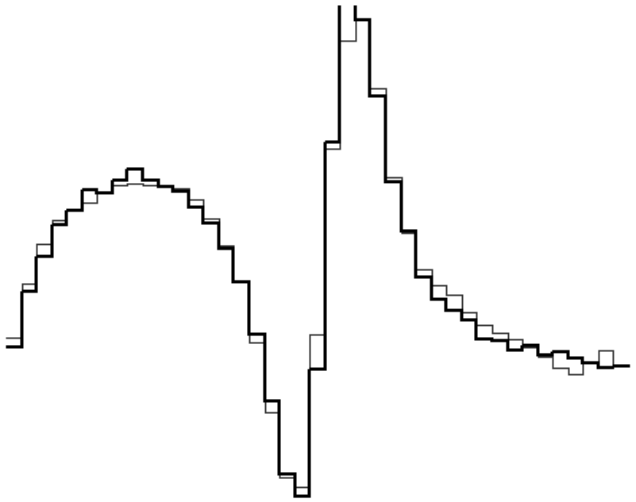
Low density (detail)



Denoised

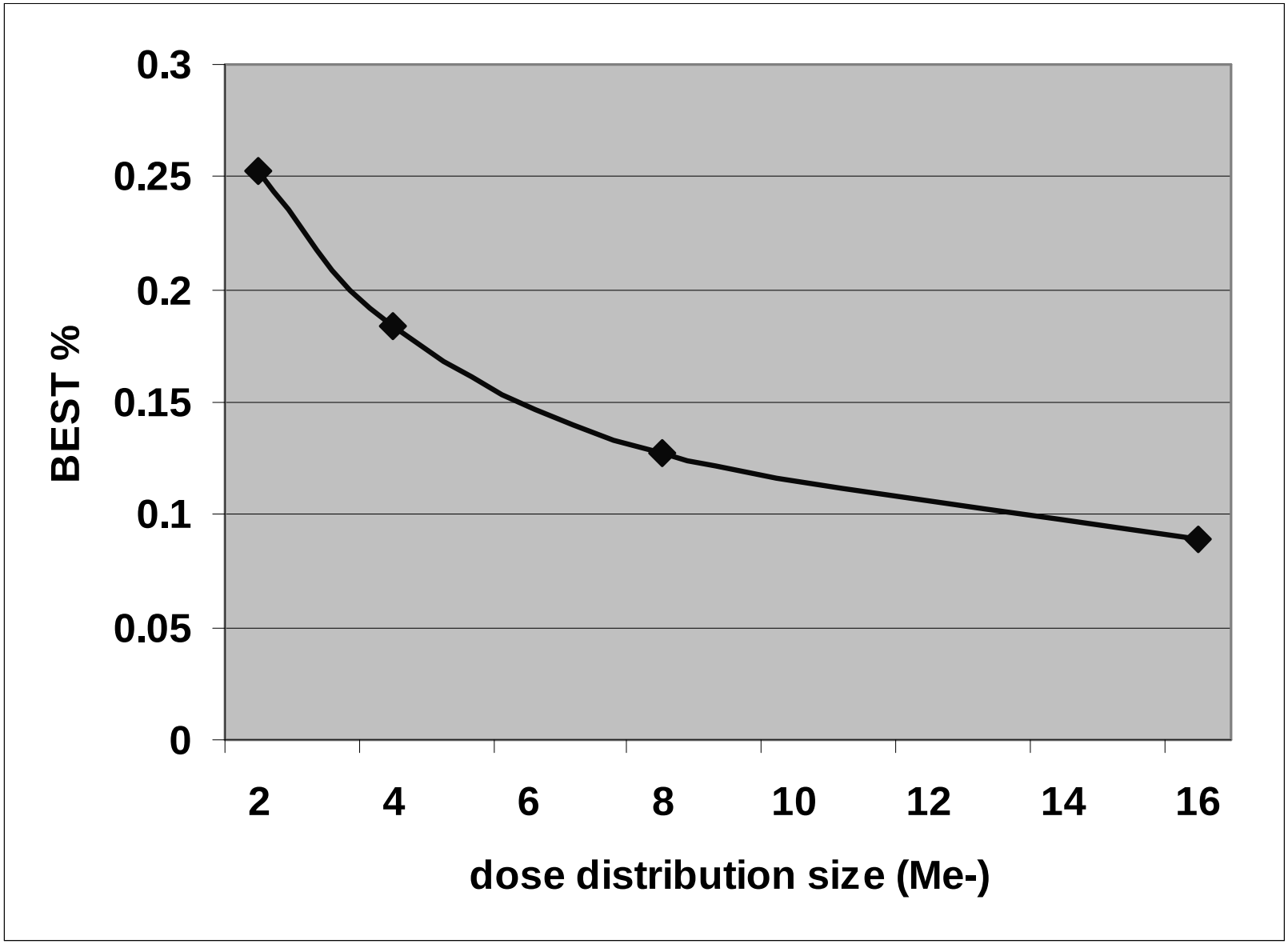


Denoised (detail)



16Me- dose distribution

Best thresholding % function



Best PSNR function

