Weekly meeting

Miscellaneous updates

Jérémie Decock

CEA Saclay - Irfu/SAp

September 28, 2016

Benchmark

Definition

Benchmark function

The error function $\mathcal{E}$ is given by:

$$\mathcal{E}(\hat{\mathbf{s}}, \mathbf{s}^*) = (\mathcal{E}_1(\hat{\mathbf{s}}, \mathbf{s}^*), \mathcal{E}_2(\hat{\mathbf{s}}, \mathbf{s}^*))^T$$

$$\mathcal{E}_1(\hat{\mathbf{s}}, \mathbf{s}^*) = \text{mean}\left(\text{abs}\left(\frac{\hat{\mathbf{s}}}{\sum_i \hat{\mathbf{s}}_i} - \frac{\mathbf{s}^*}{\sum_i \mathbf{s}^*{}_i}\right)\right)$$

$$\mathcal{E}_2(\hat{\mathbf{s}}, \mathbf{s}^*) = \frac{\text{abs}\left(\sum_i \hat{\mathbf{s}}_i - \sum_i \mathbf{s}^*{}_i\right)}{\sum_i \mathbf{s}^*{}_i}$$

Where:

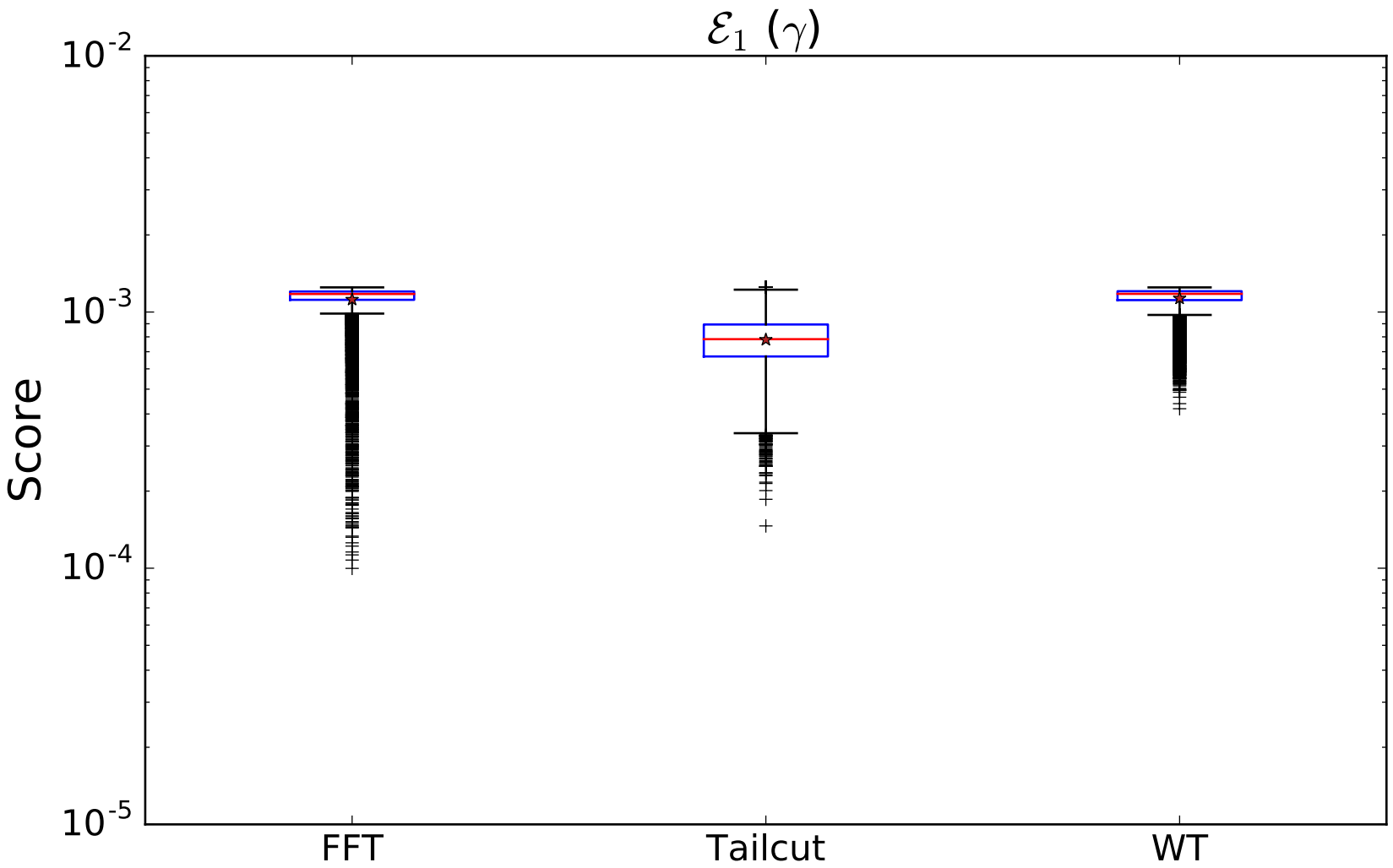
- $\hat{\mathbf{s}}$ is the output image (the "cleaned" image) $\in \mathbb{R}^d$
- $\mathbf{s}^*$ is the reference image (the "clean" image) $\in \mathbb{R}^d$
- i is the index of a PMT (i.e. of a pixel) within an image

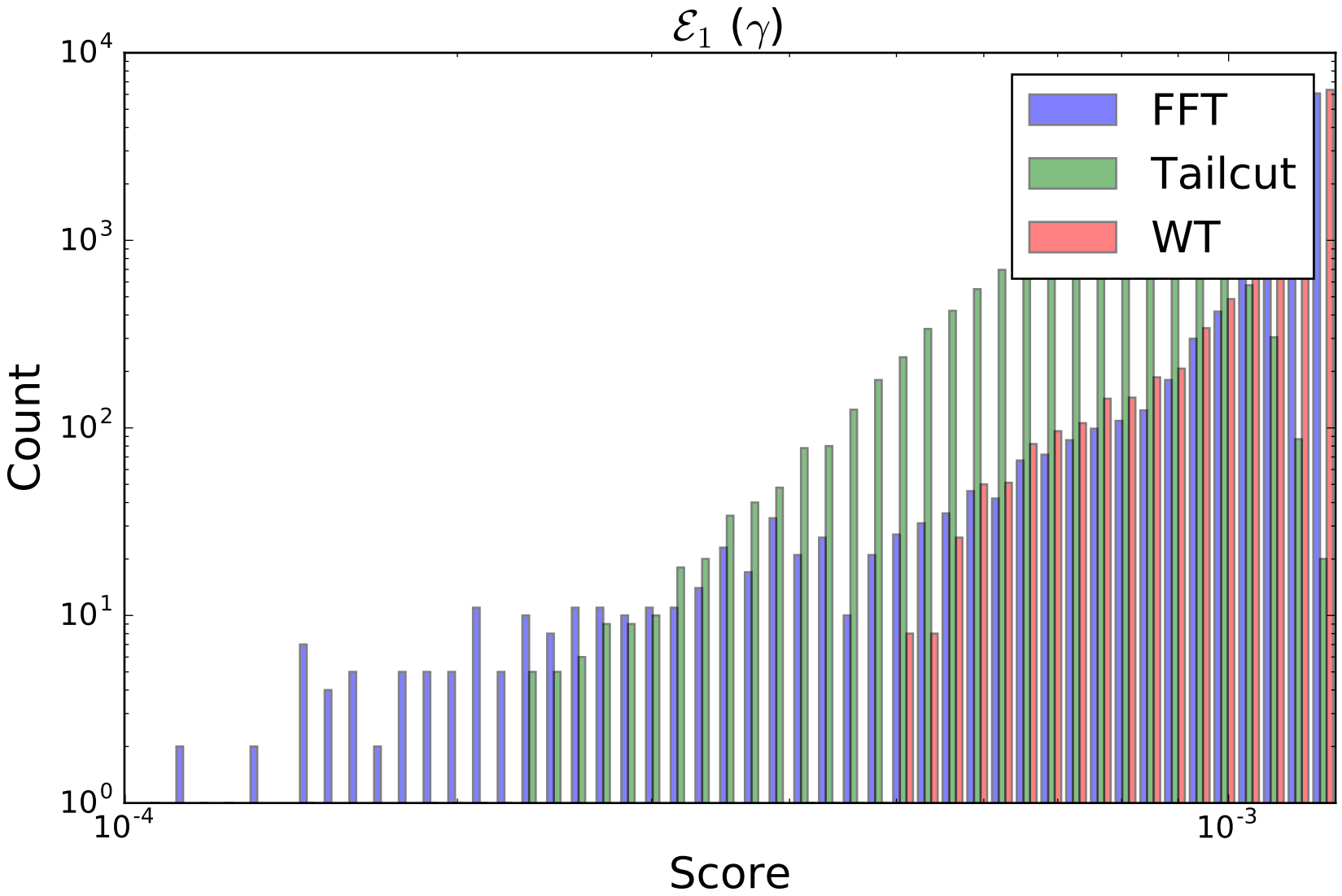
Gammas

- ASTRI mini-array (calibrated data)
- Telescopes 1 to 33 only (ASTRI)
- Polychromatic event set
- Input files: sapcta:/dsm/manip/cta/DATA/astri_mini_array/fits/gamma/
- Source file: sapcta:/dsm/manip/cta/DATA/astri_mini_array/gamma/run1001.simtel.gz
- Num samples: 14899 images

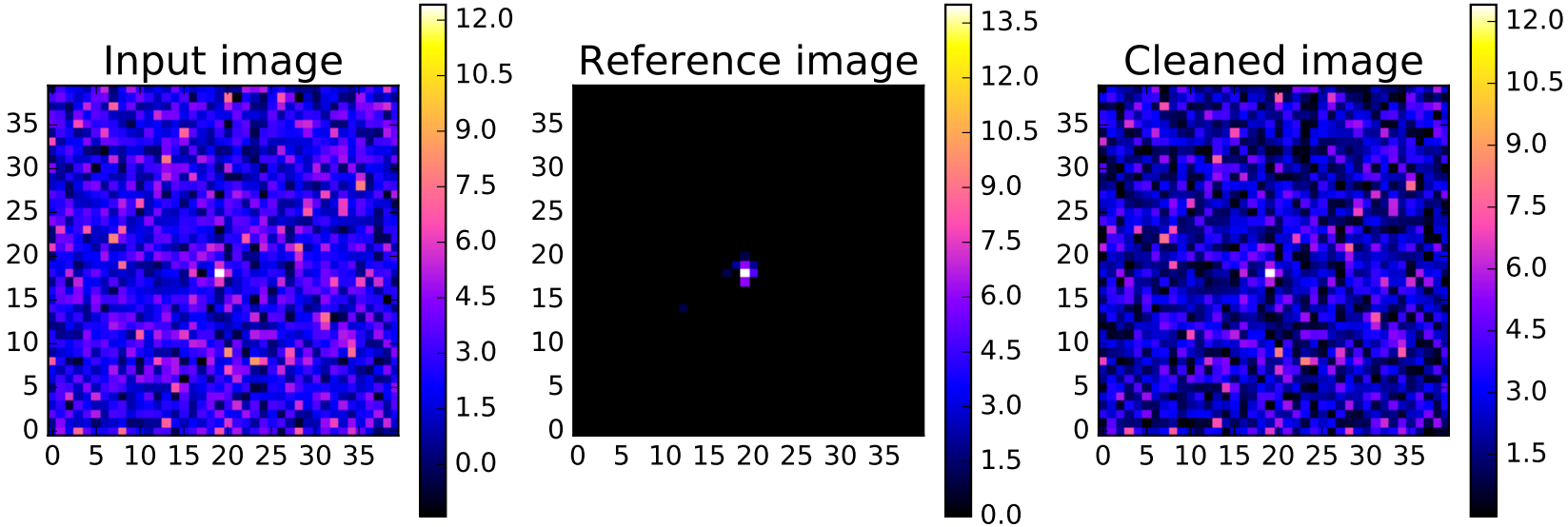
Cleaning algorithms:

- Tailcut: JD's implementation
- FFT: Numpy implementation
- Wavelets: Cosmostat Sparce2D (mr_transform) b-Spline wavelet transform

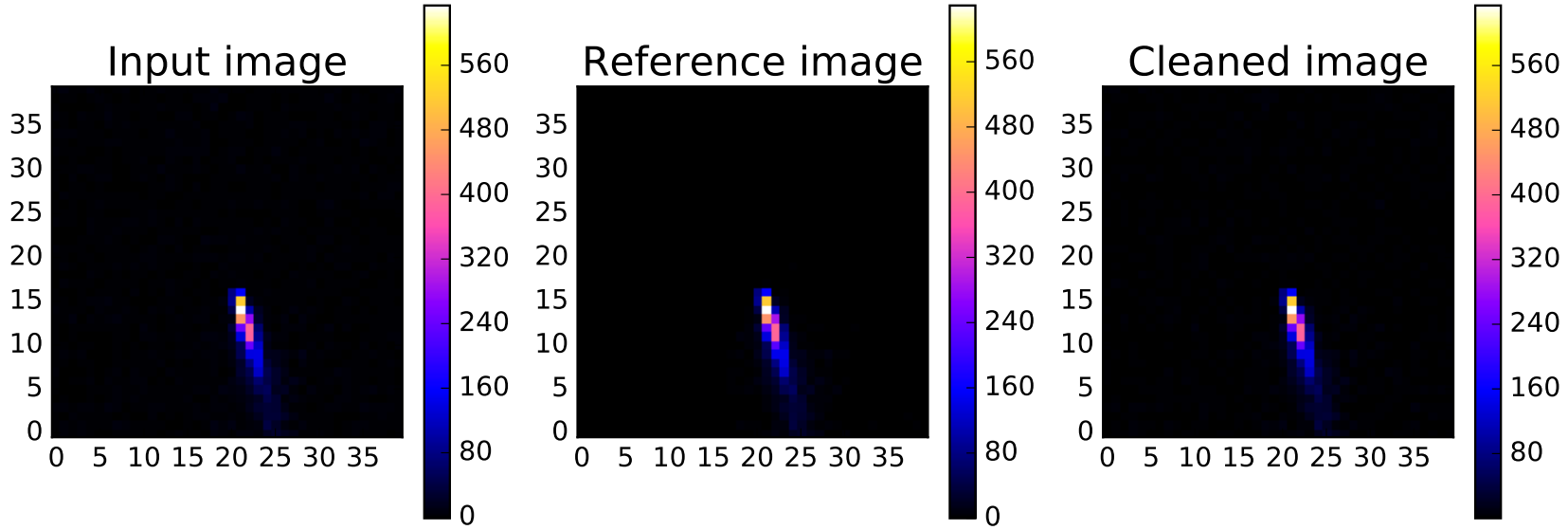

$\mathcal{E}_1$ (γ)
Score
10^{-2}
10^{-3}
10^{-4}
10^{-5}
FFT
Tailcut
WT


$\mathcal{E}_1$ (γ)
FFT
Tailcut
WT
Count
Score
10^0
10^1
10^2
10^3
10^4
10^{-4}
10^{-3}

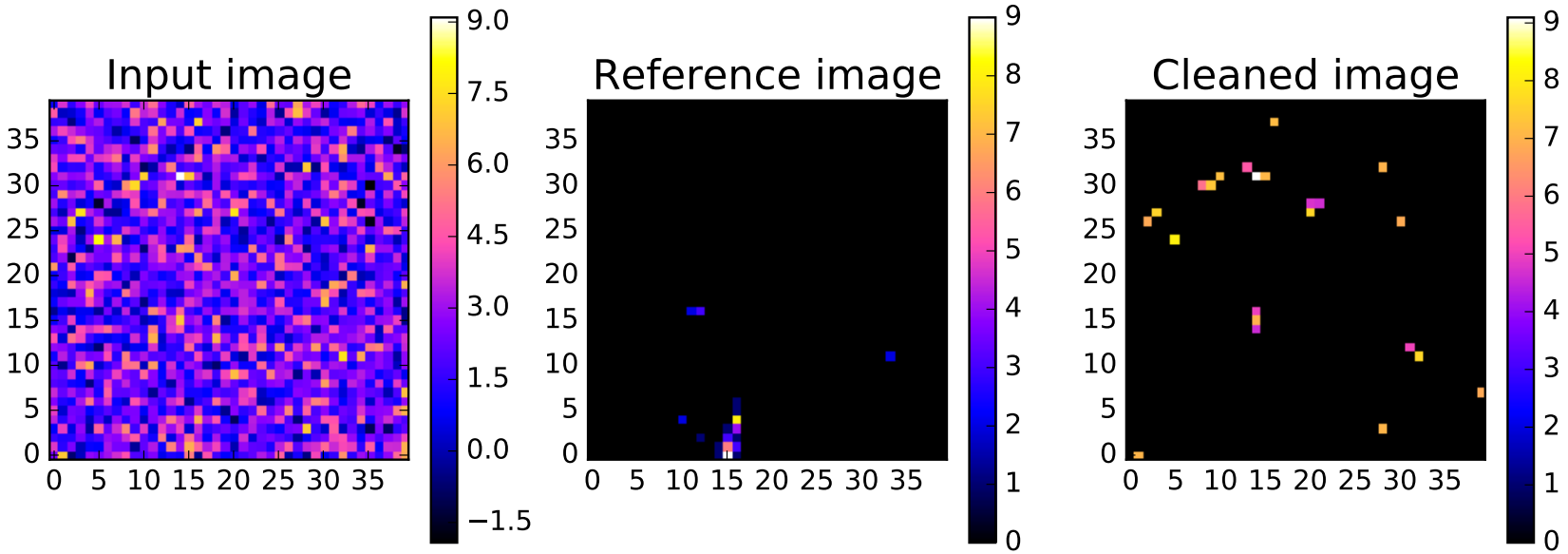
Bad FFT example



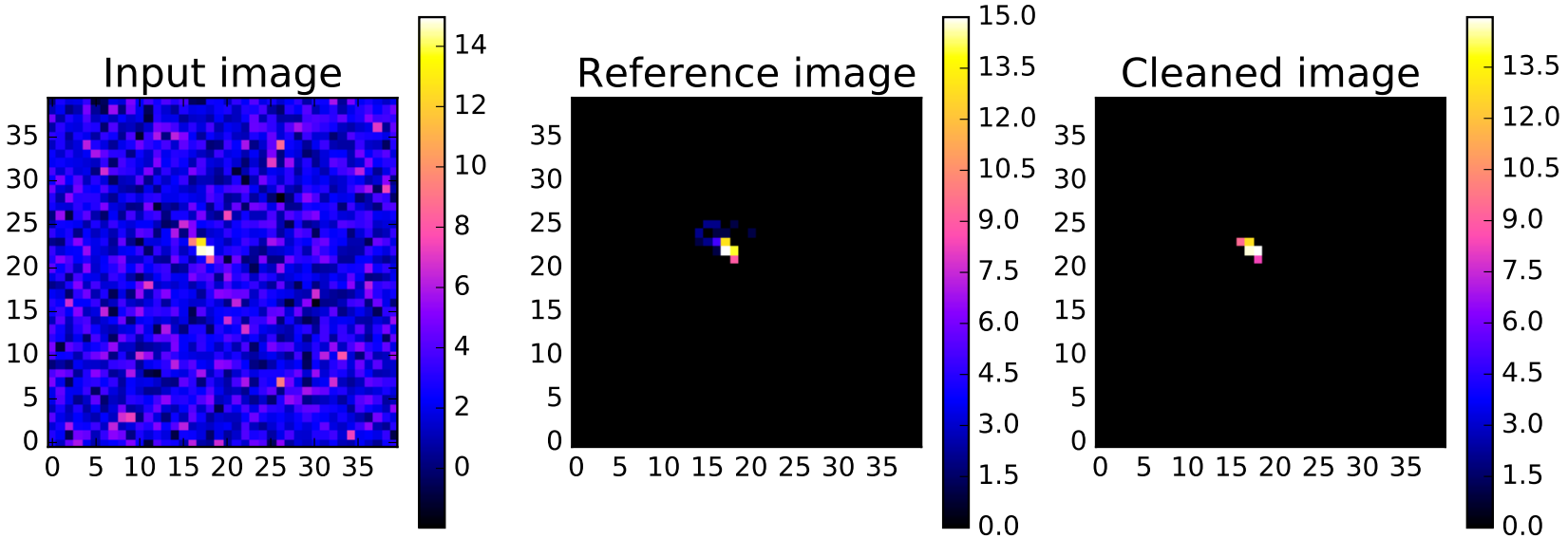
Good FFT example



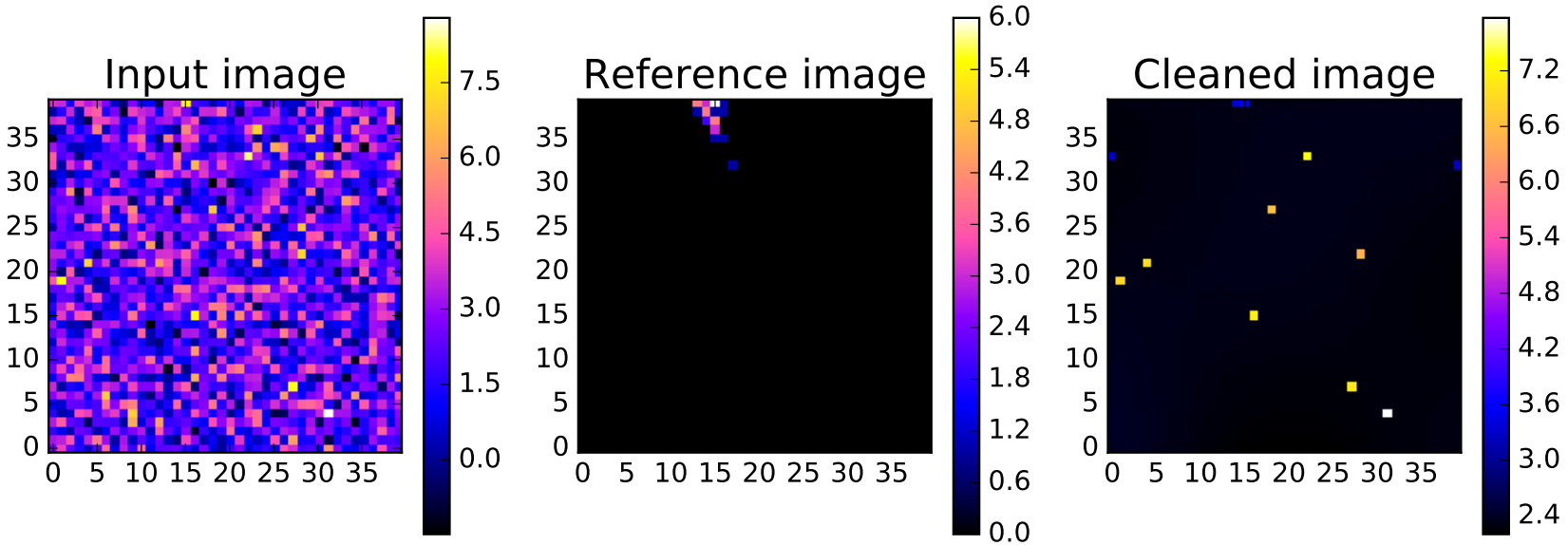
Bad Tailcut example



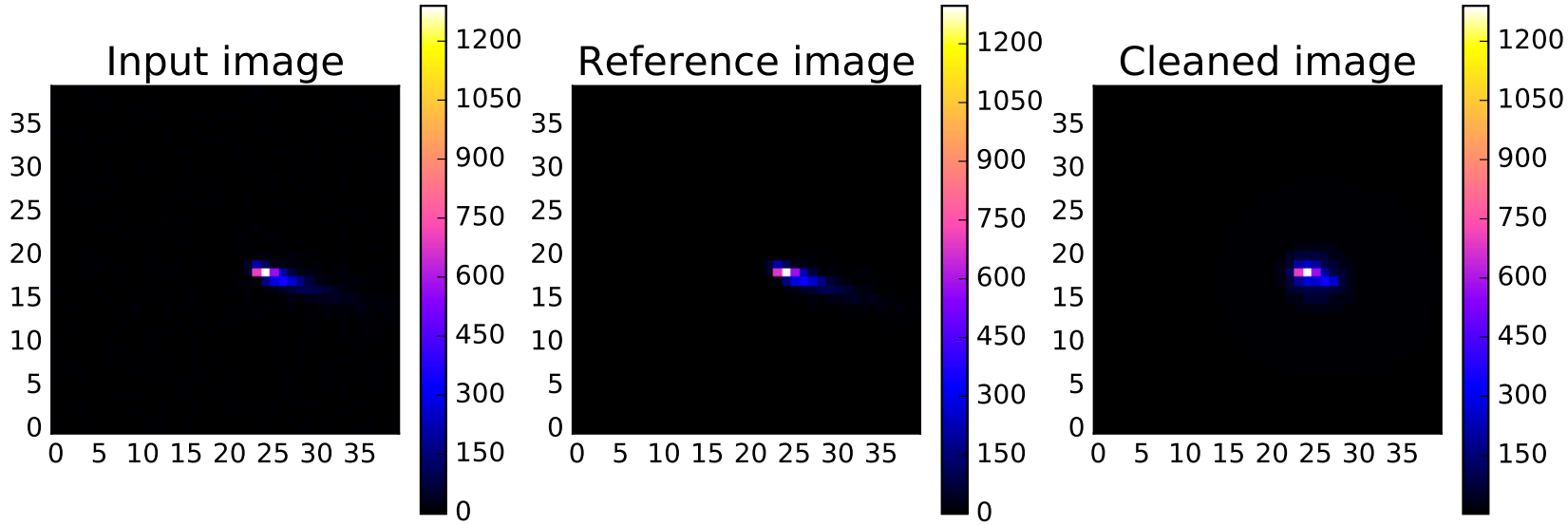
Good Tailcut example

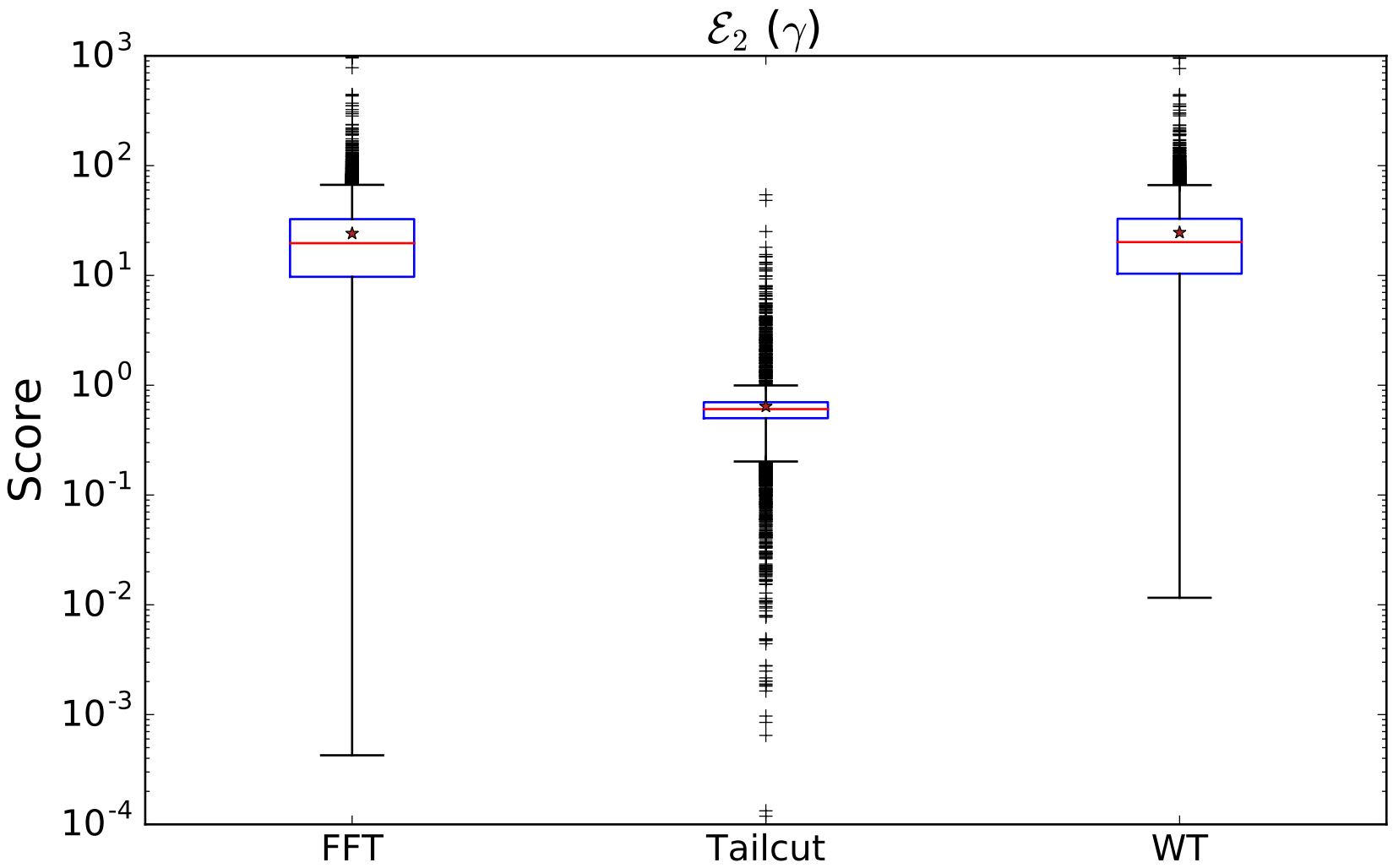


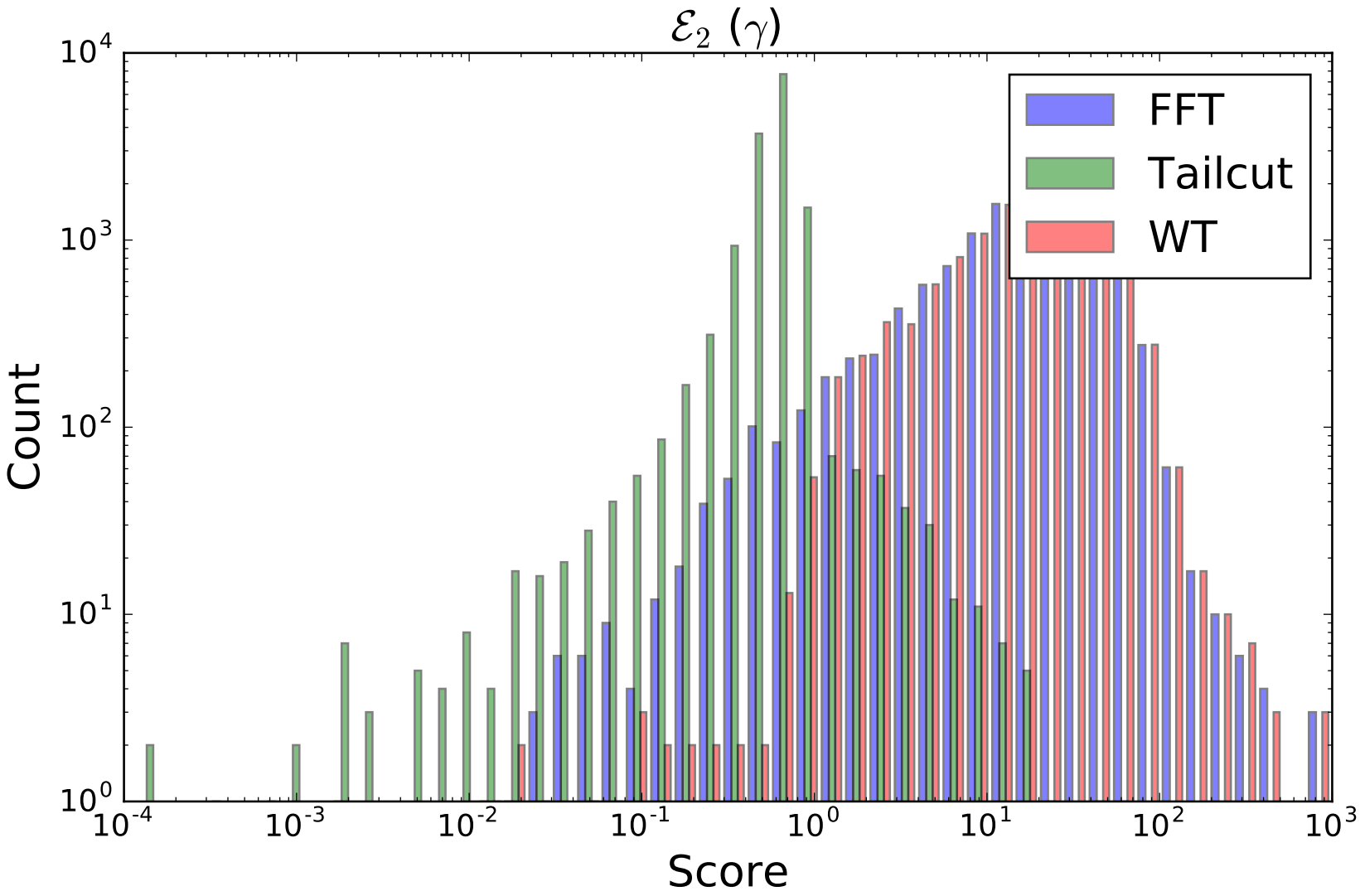
Bad WT example



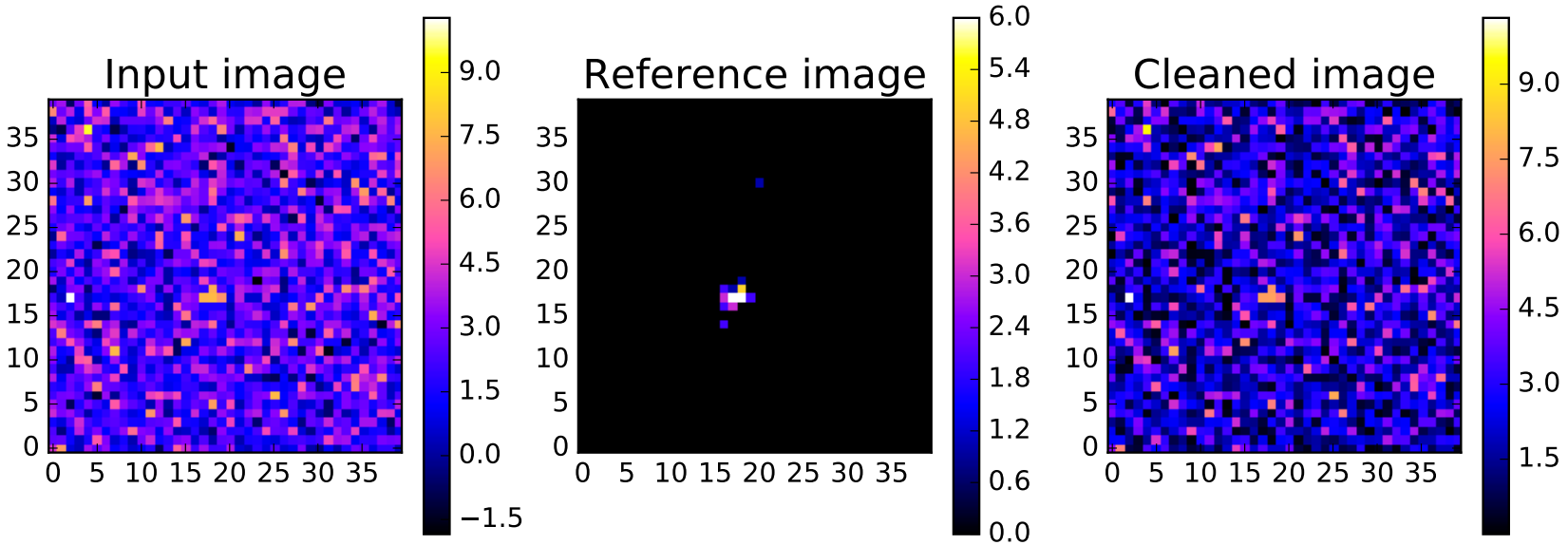
Good WT example



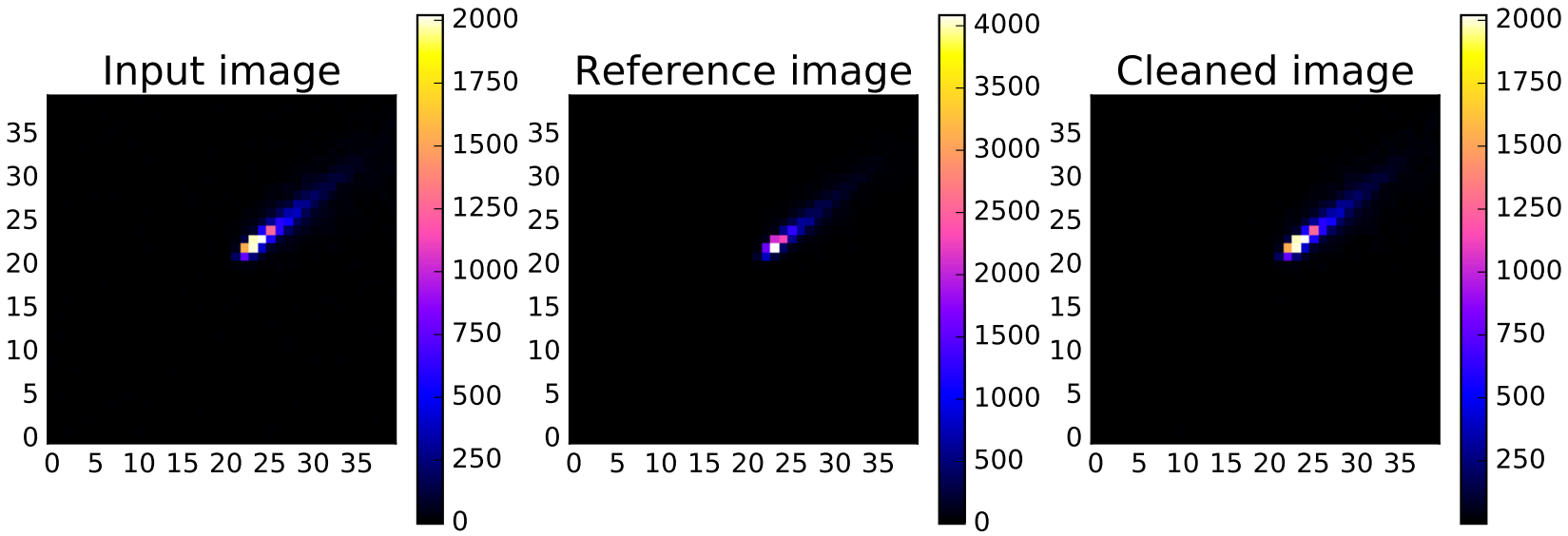

$\mathcal{E}_2$ (γ)
Score
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
FFT
Tailcut
WT


$\mathcal{E}_2$ (γ)
FFT
Tailcut
WT
Count
Score
10^0
10^1
10^2
10^3
10^4
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^0
10^1
10^2
10^3

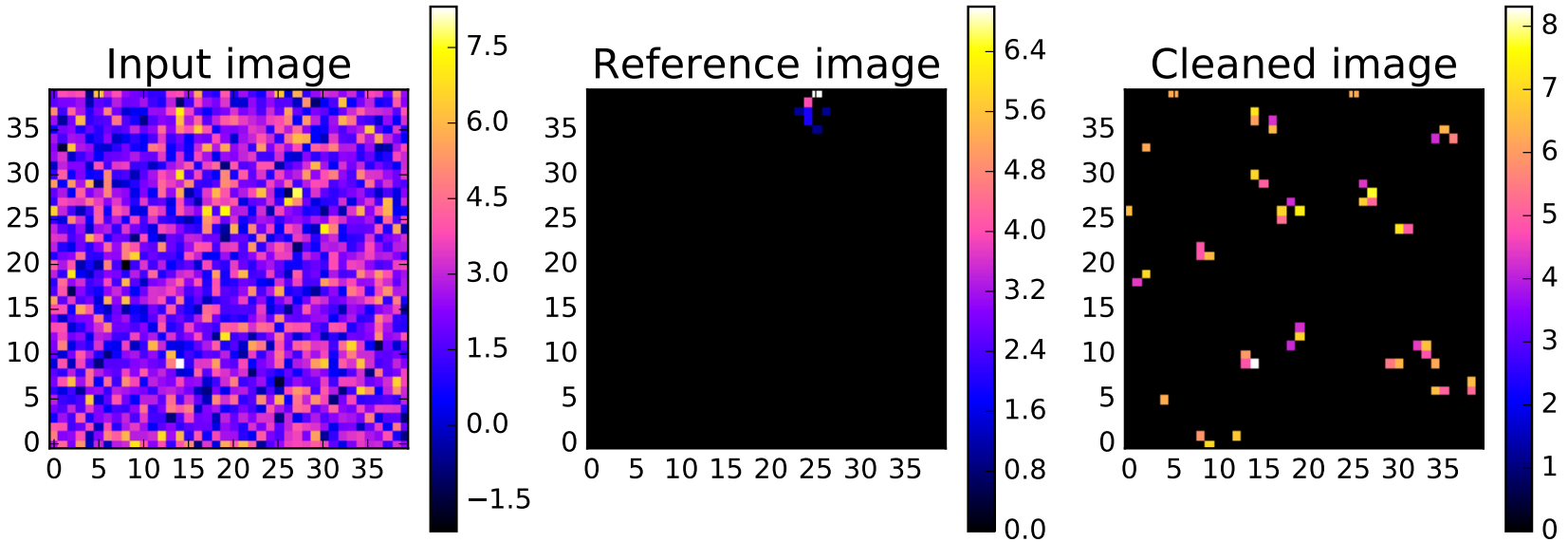
Bad FFT example



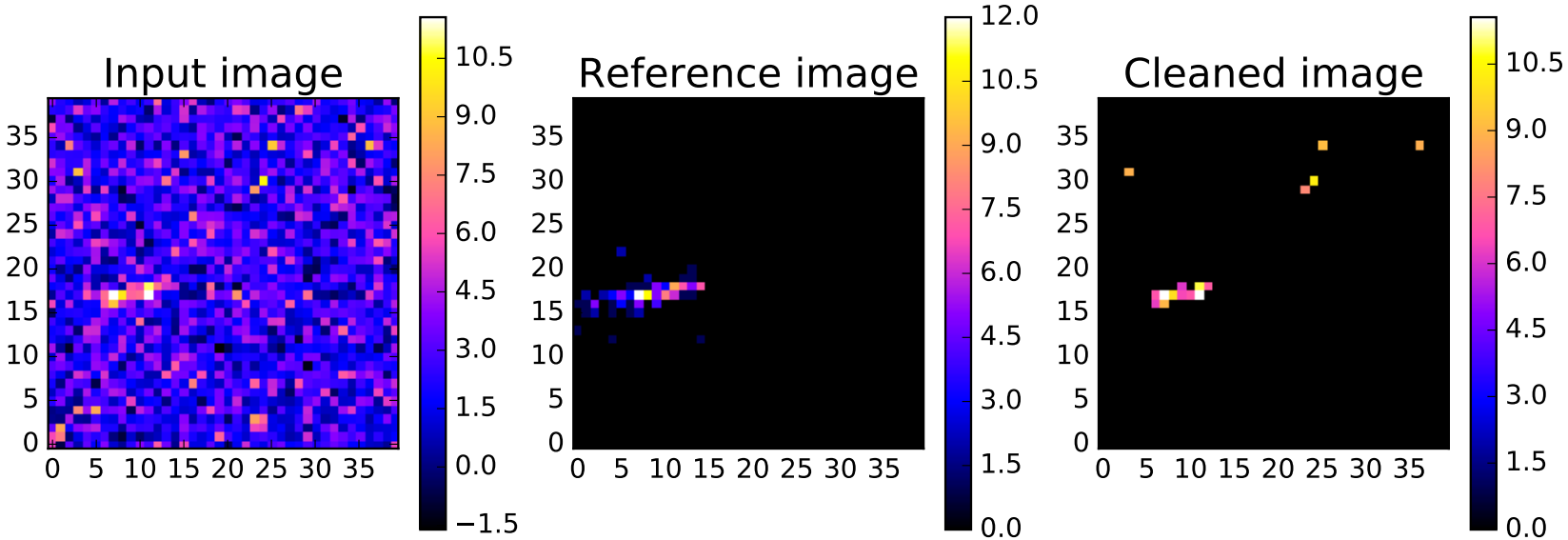
Good FFT example



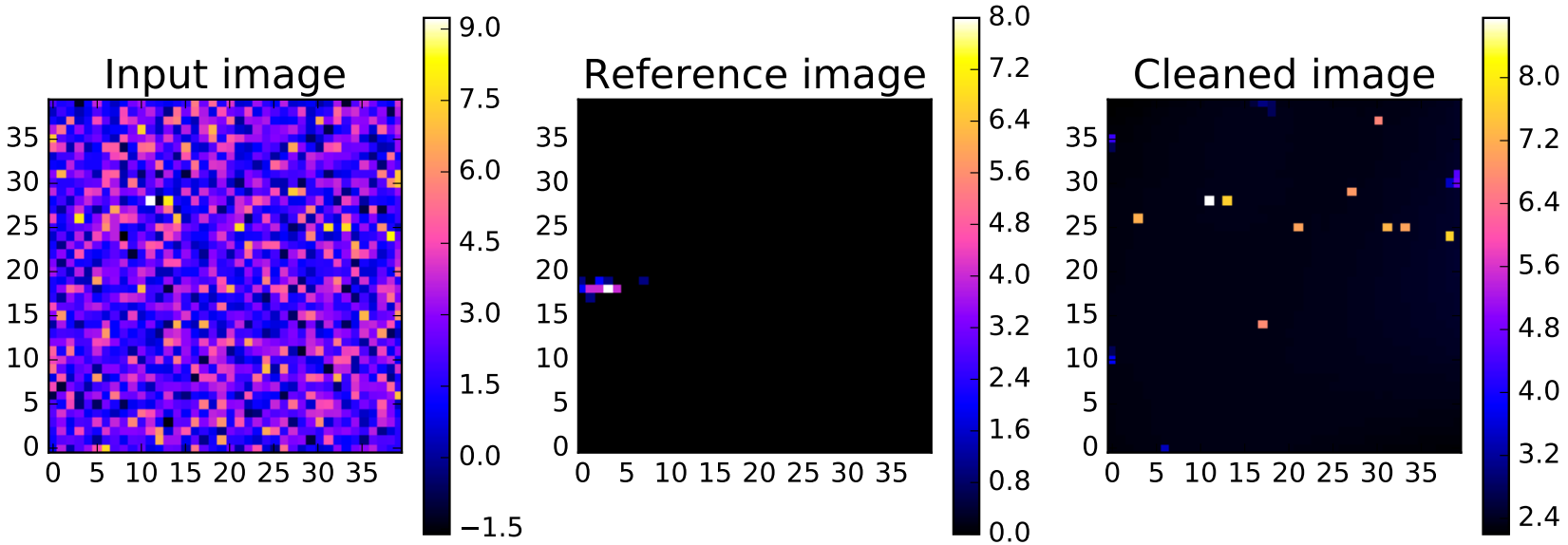
Bad Tailcut example



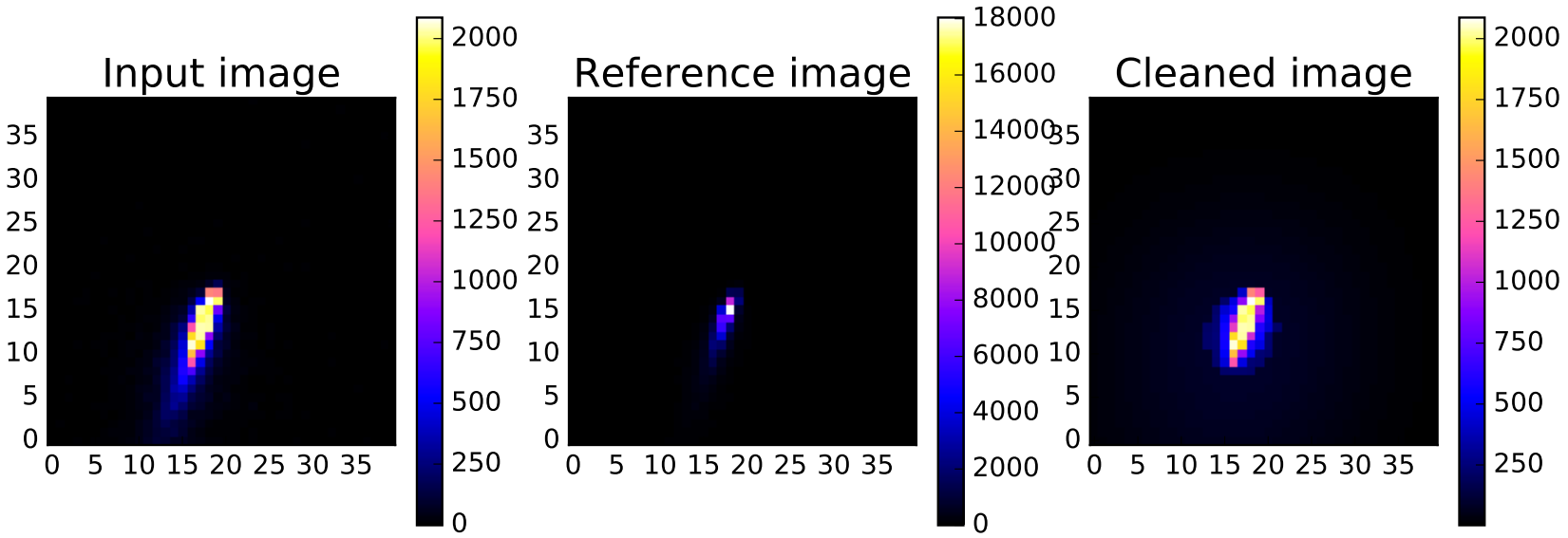
Good Tailcut example

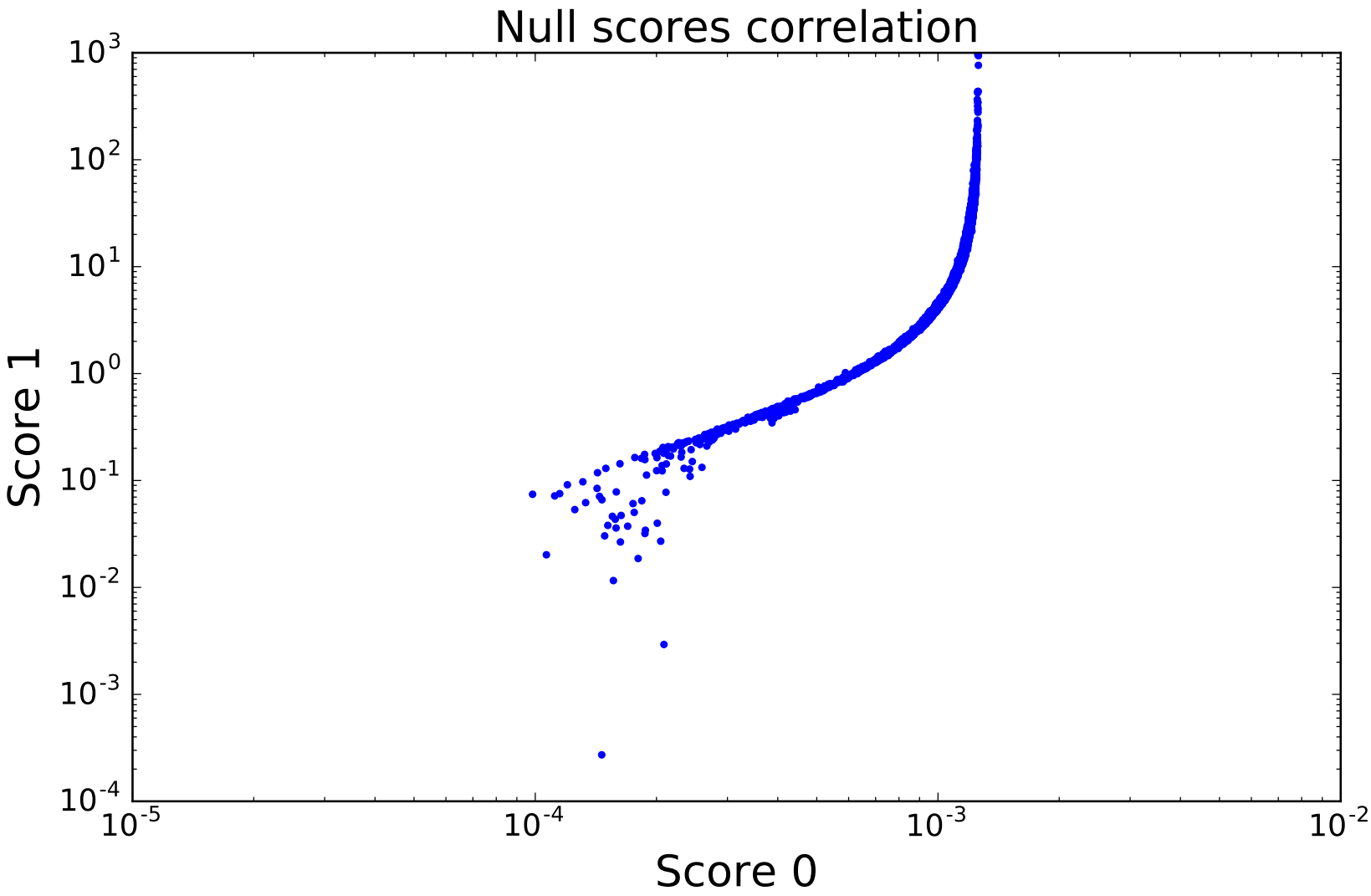


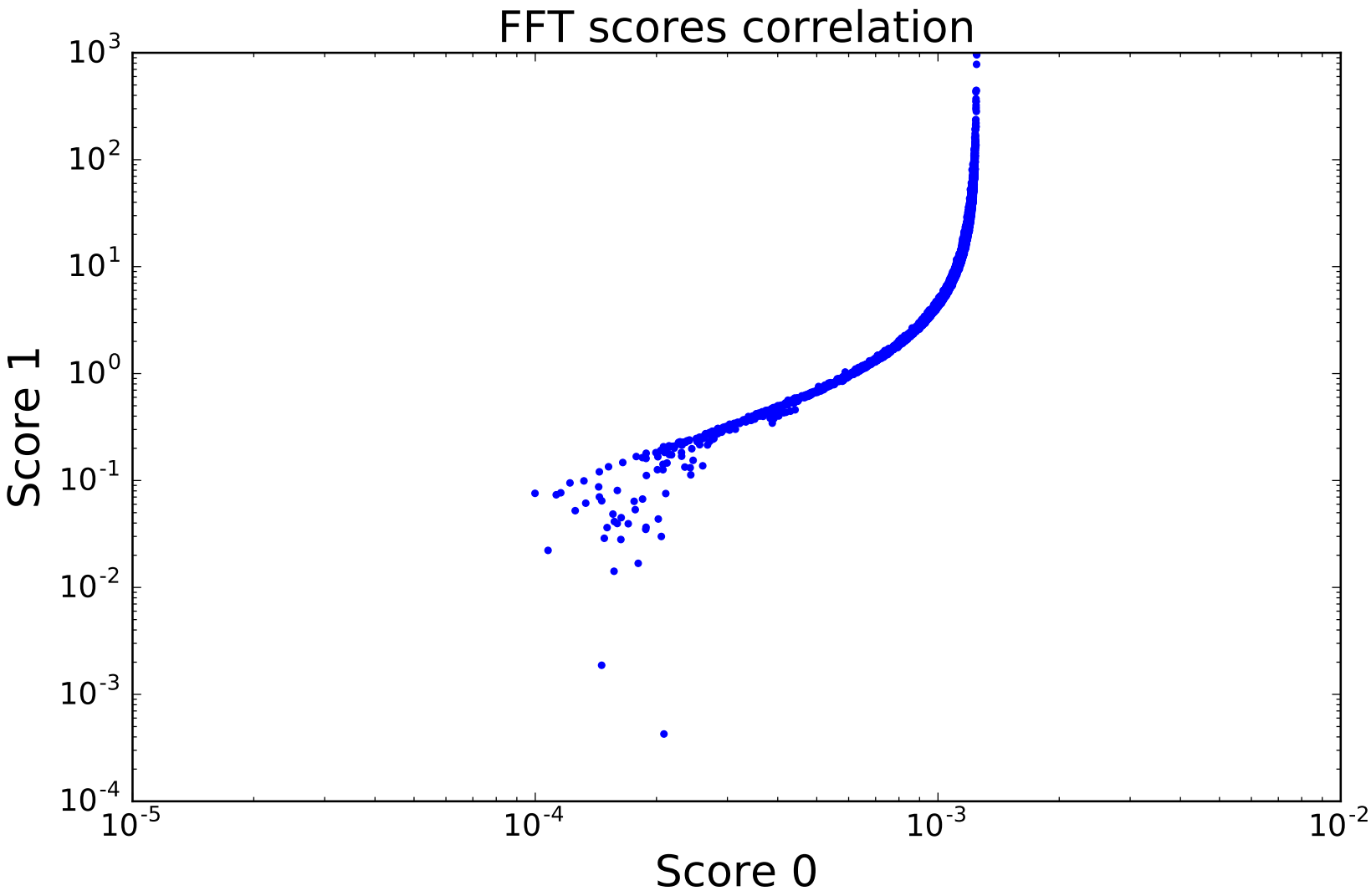
Bad WT example

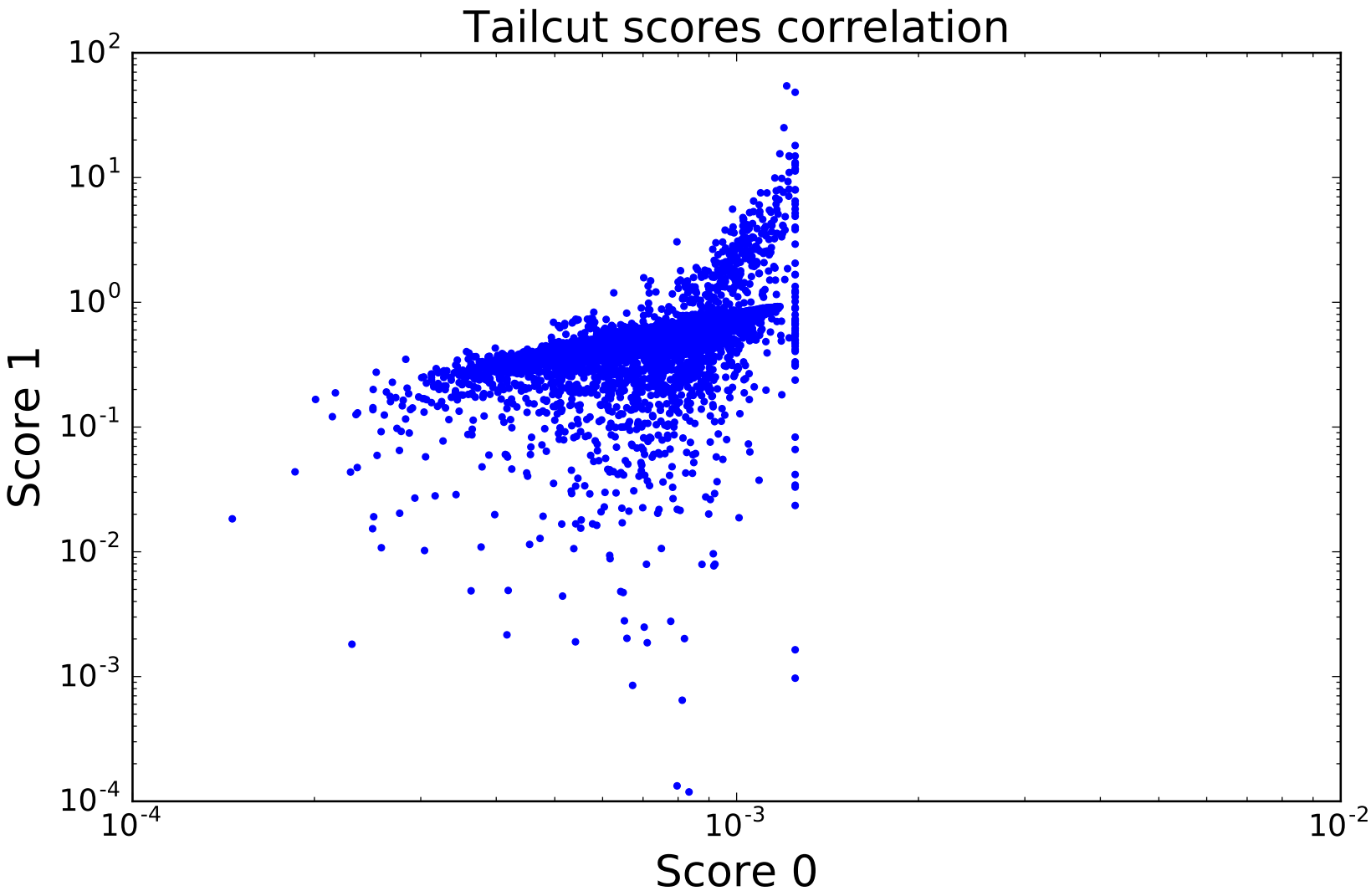


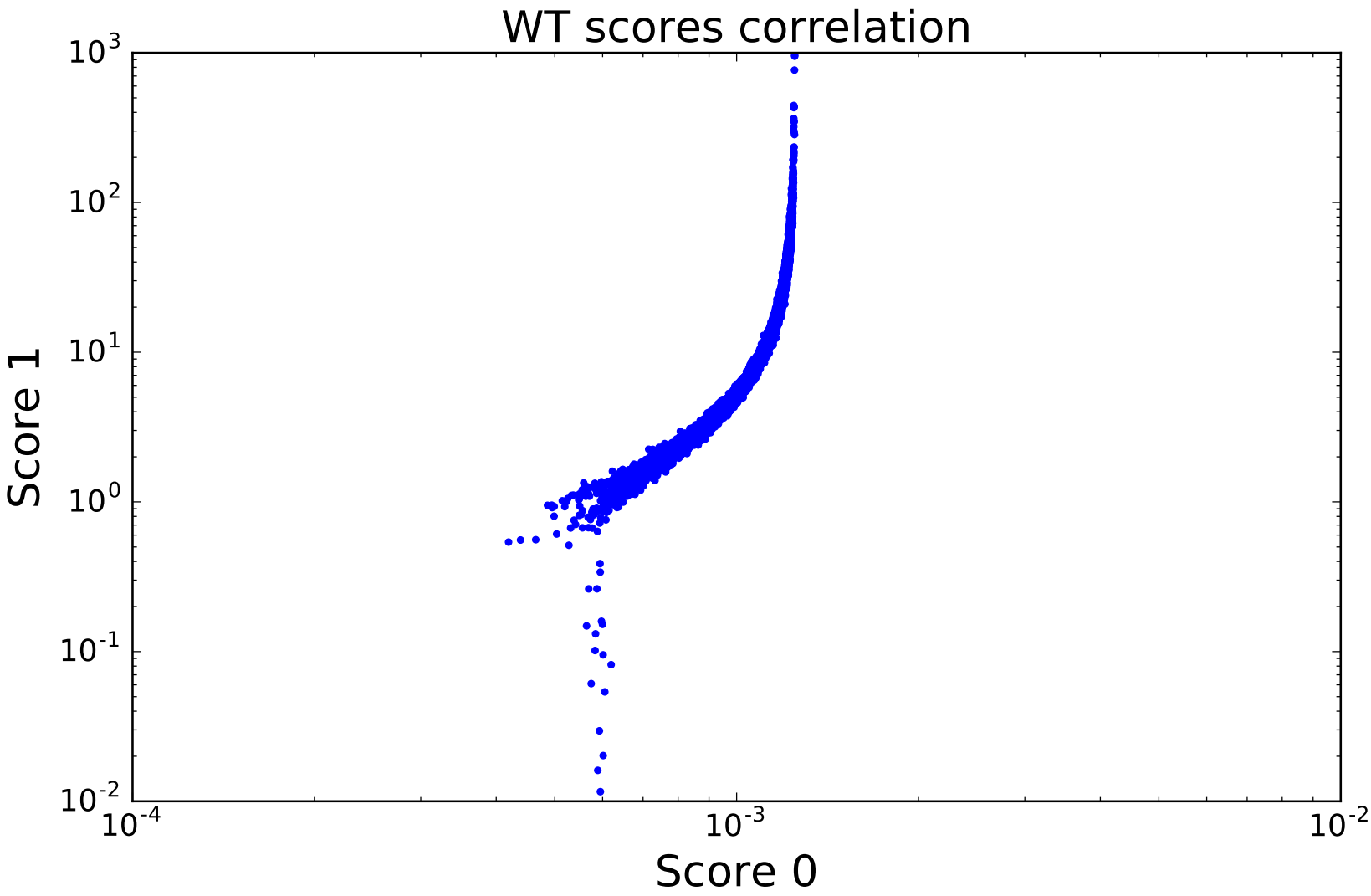
Good WT example




Null scores correlation
Score 1
Score 0
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-4}
10^{-3}
10^{-2}


FFT scores correlation
Score 1
Score 0
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-4}
10^{-3}
10^{-2}


Tailcut scores correlation
Score 1
Score 0
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-4}
10^{-3}
10^{-2}

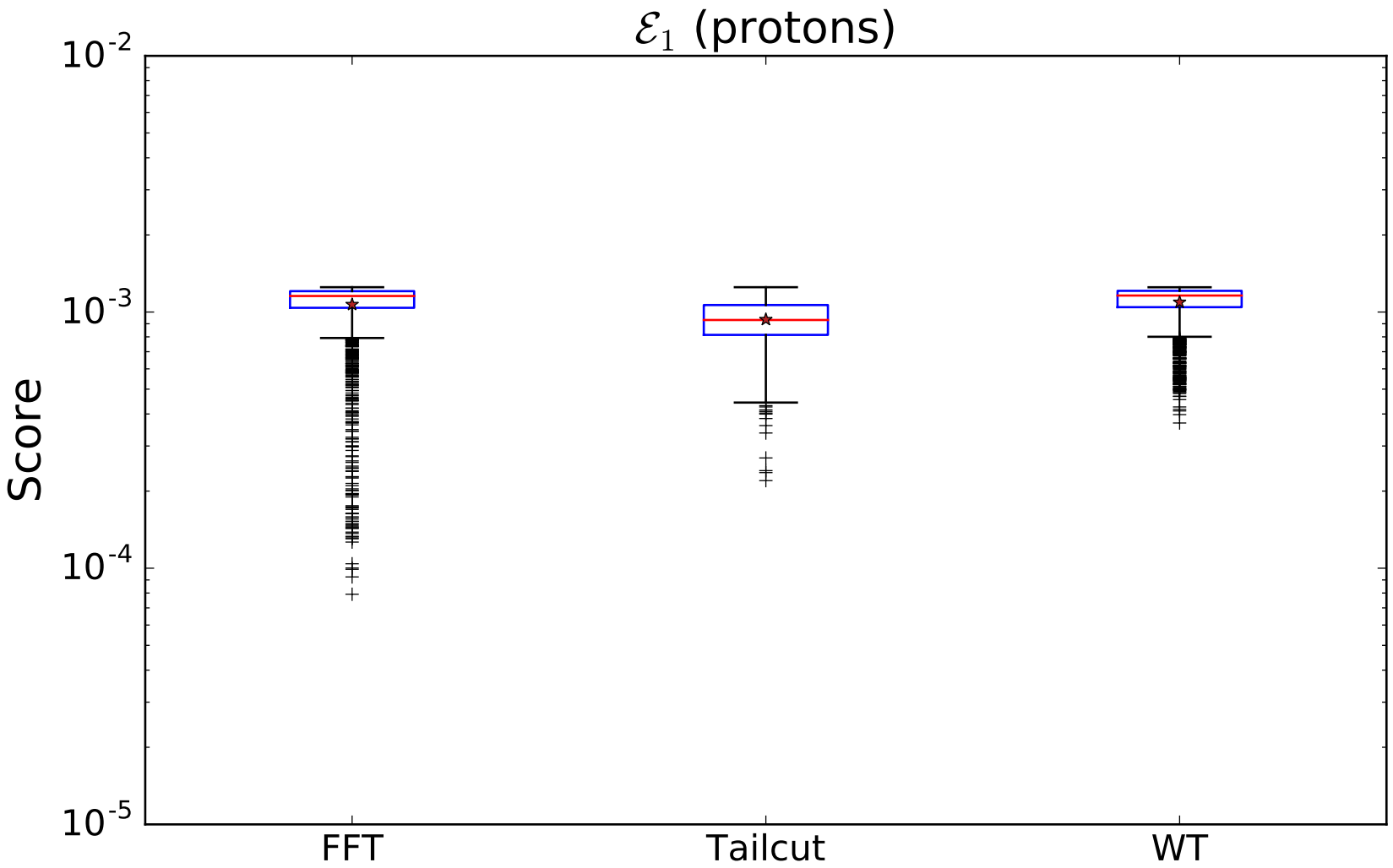

WT scores correlation
Score 1
Score 0
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-4}
10^{-3}
10^{-2}

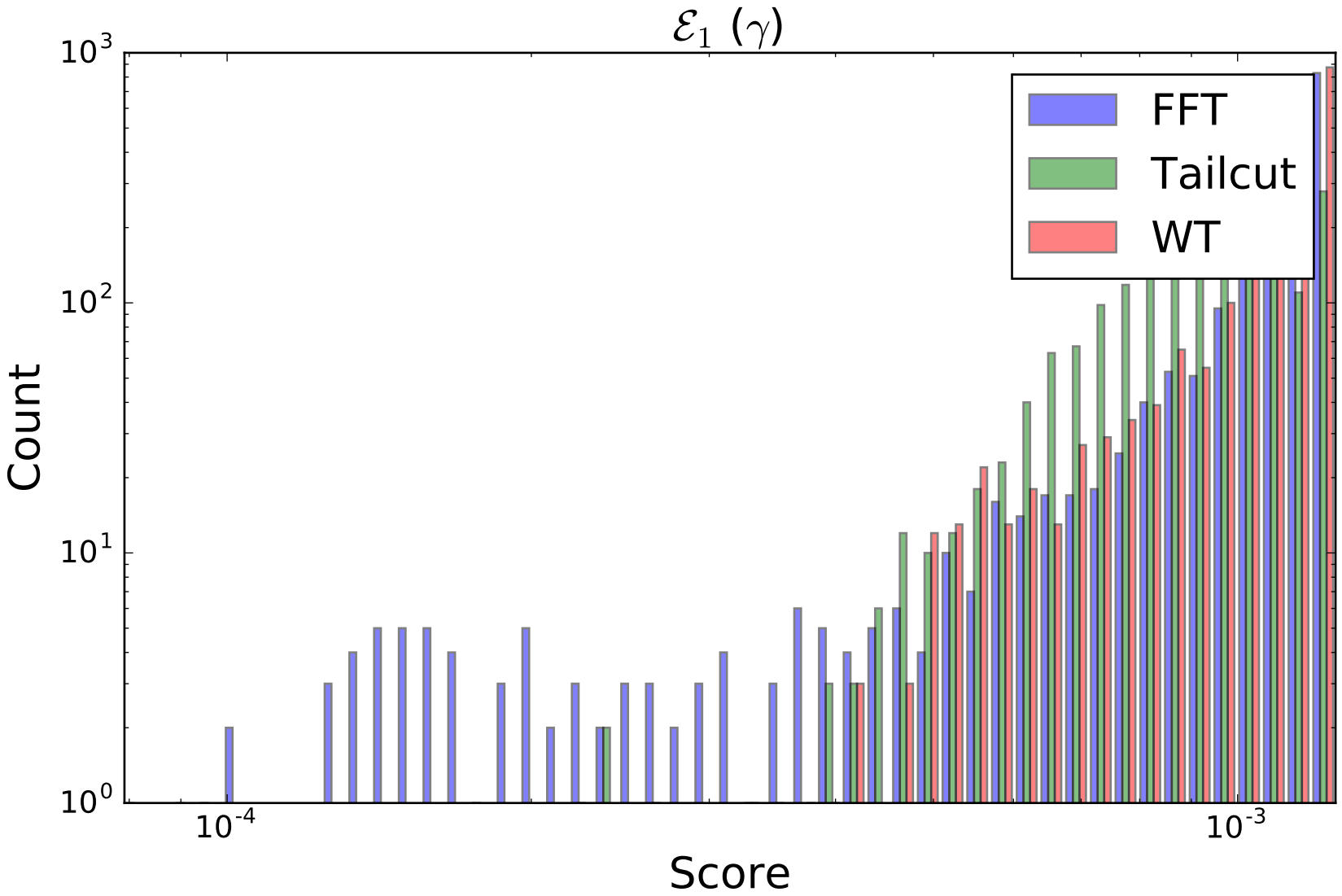
Protons

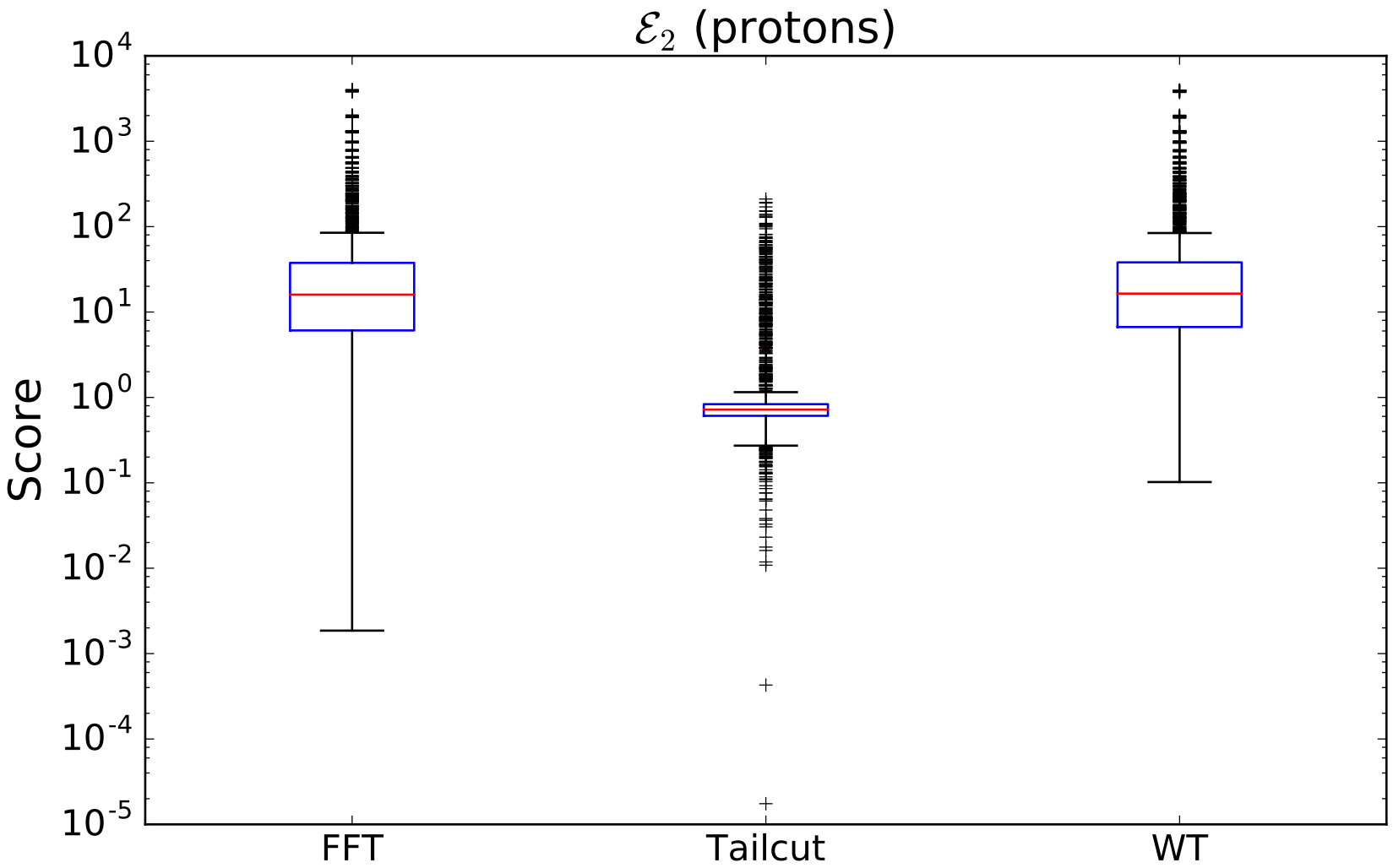
- ASTRI mini-array (calibrated data)
- Telescopes 1 to 33 only (ASTRI)
- Polychromatic event set
- Input files: sapcta:/dsm/manip/cta/DATA/astri_mini_array/fits/proton/
- Source file: sapcta:/dsm/manip/cta/DATA/astri_mini_array/proton/run10001.simtel.gz
- Num samples: 2203 images

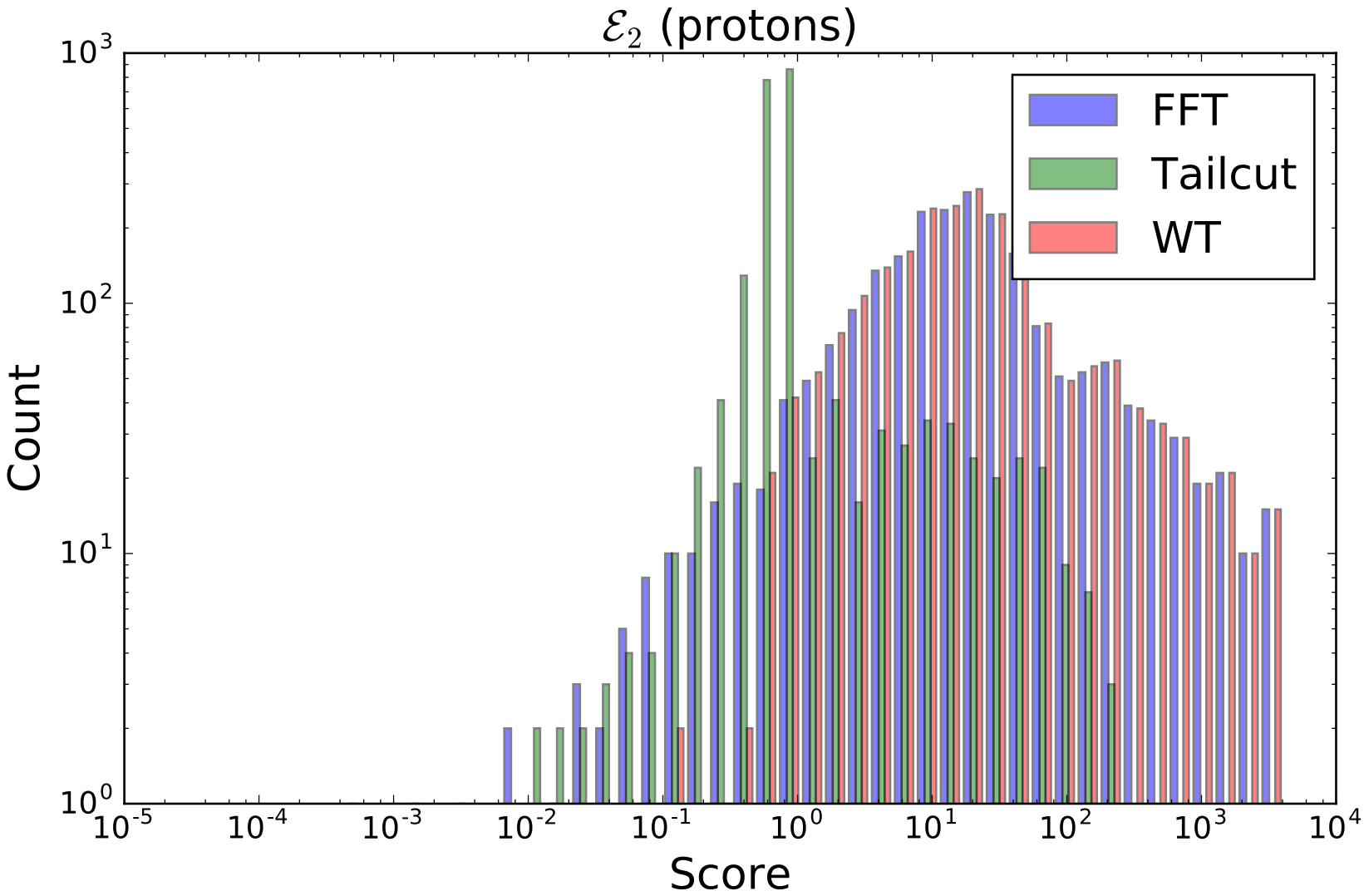
Cleaning algorithms:

- Tailcut: JD's implementation
- FFT: Numpy implementation
- Wavelets: Cosmostat Sparce2D (mr_transform) b-Spline wavelet transform

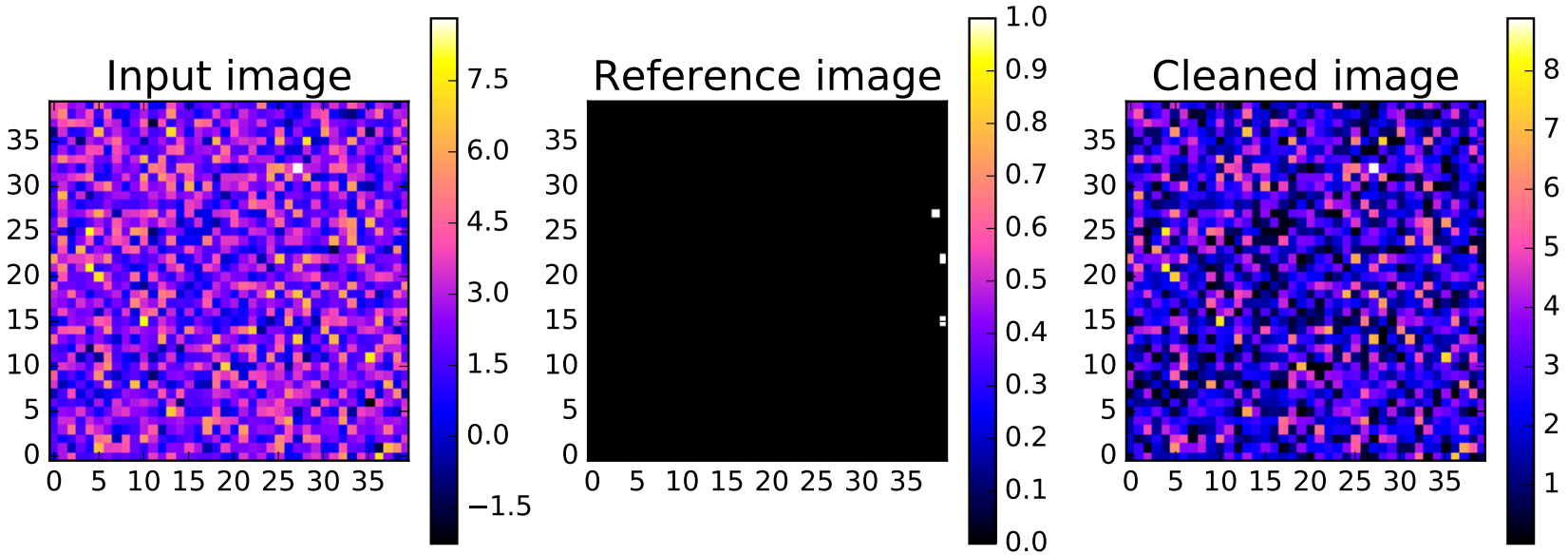

$\mathcal{E}_1$ (protons)
Score
10^{-2}
10^{-3}
10^{-4}
10^{-5}
FFT
Tailcut
WT


$\mathcal{E}_1$ (γ)
FFT
Tailcut
WT
Count
Score
10^3
10^2
10^1
10^0
10^{-4}
10^{-3}

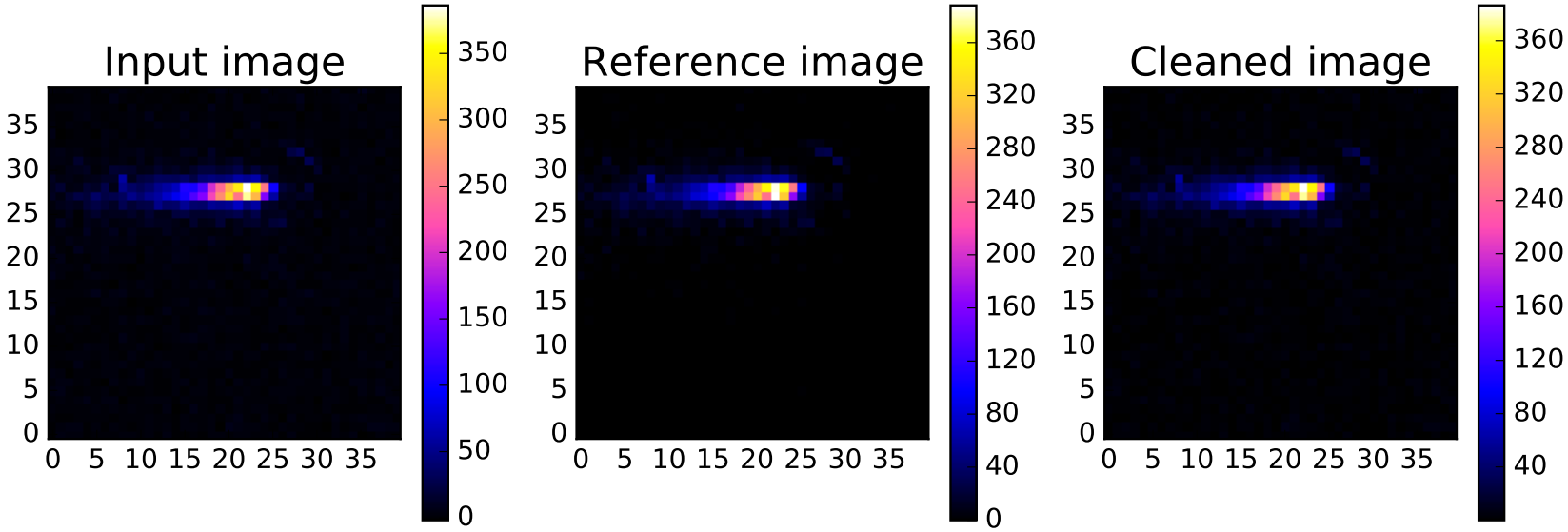

$\mathcal{E}_2$ (protons)
Score
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
FFT
Tailcut
WT


$\mathcal{E}_2$ (protons)
FFT
Tailcut
WT
Count
Score
10^0
10^1
10^2
10^3
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^0
10^1
10^2
10^3
10^4

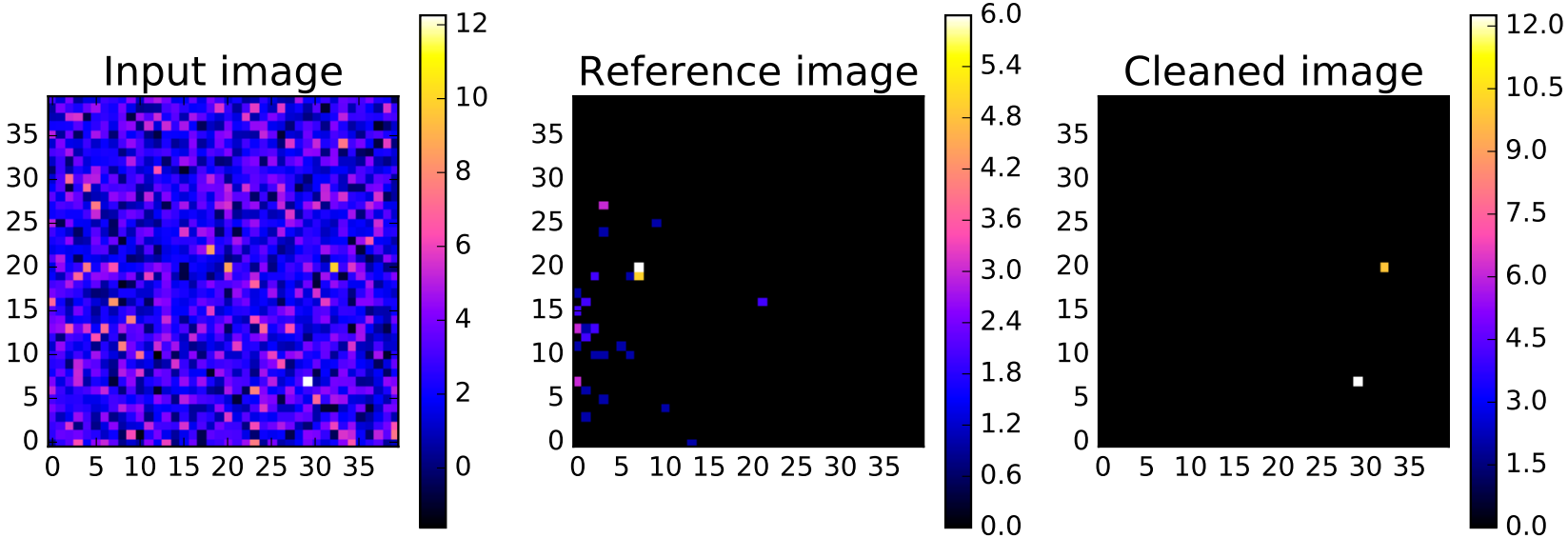
Bad FFT example



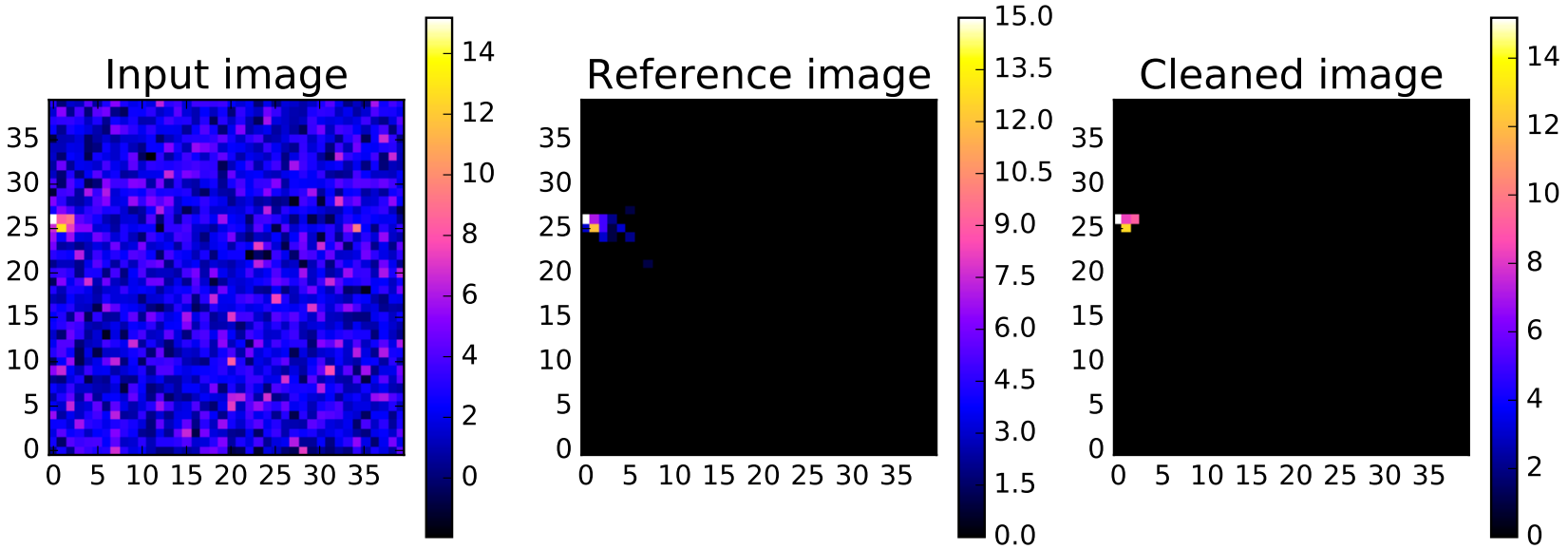
Good FFT example



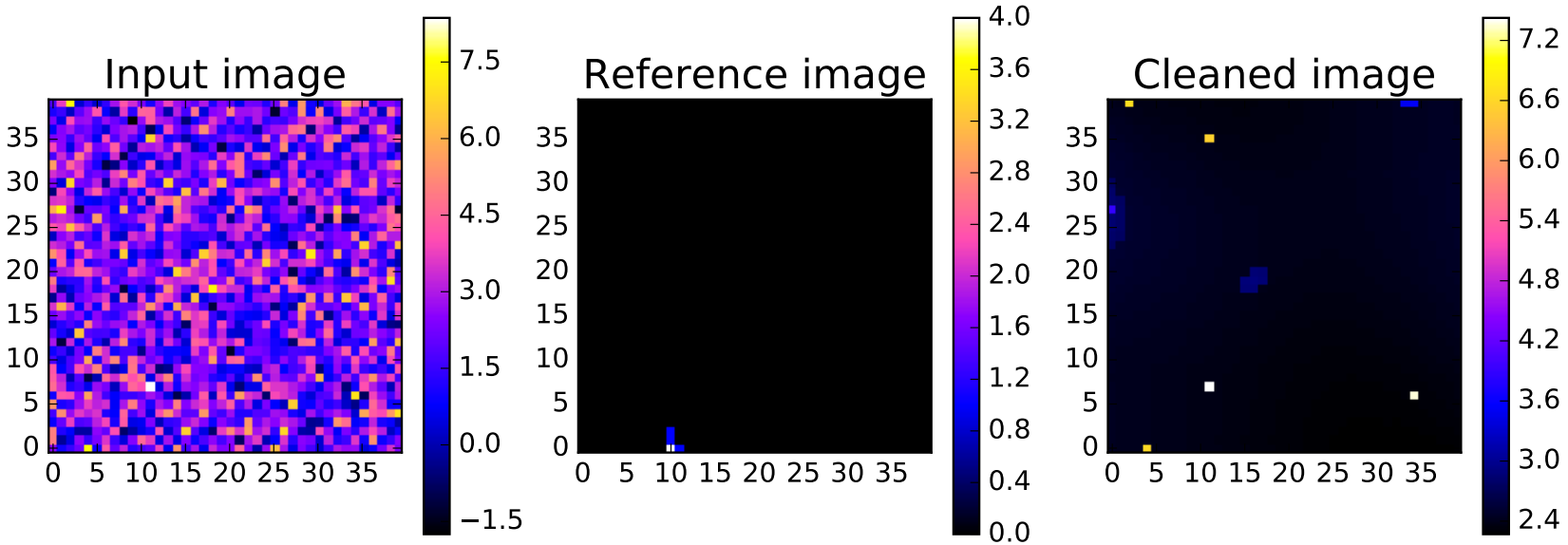
Bad Tailcut example



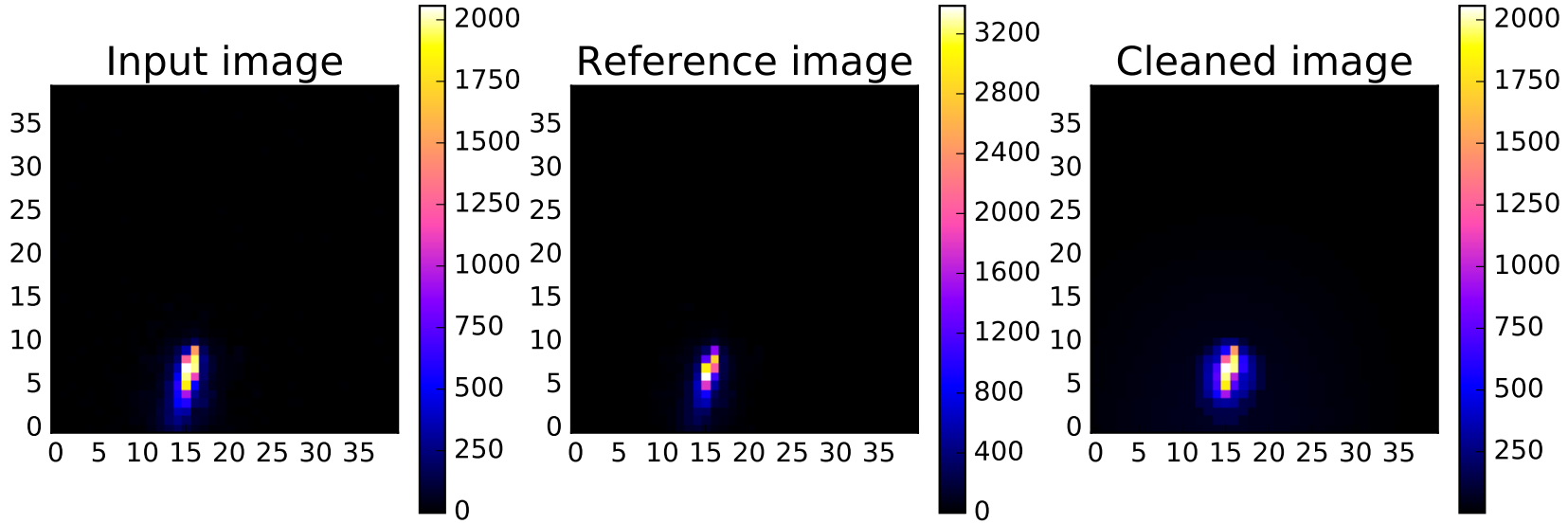
Good Tailcut example

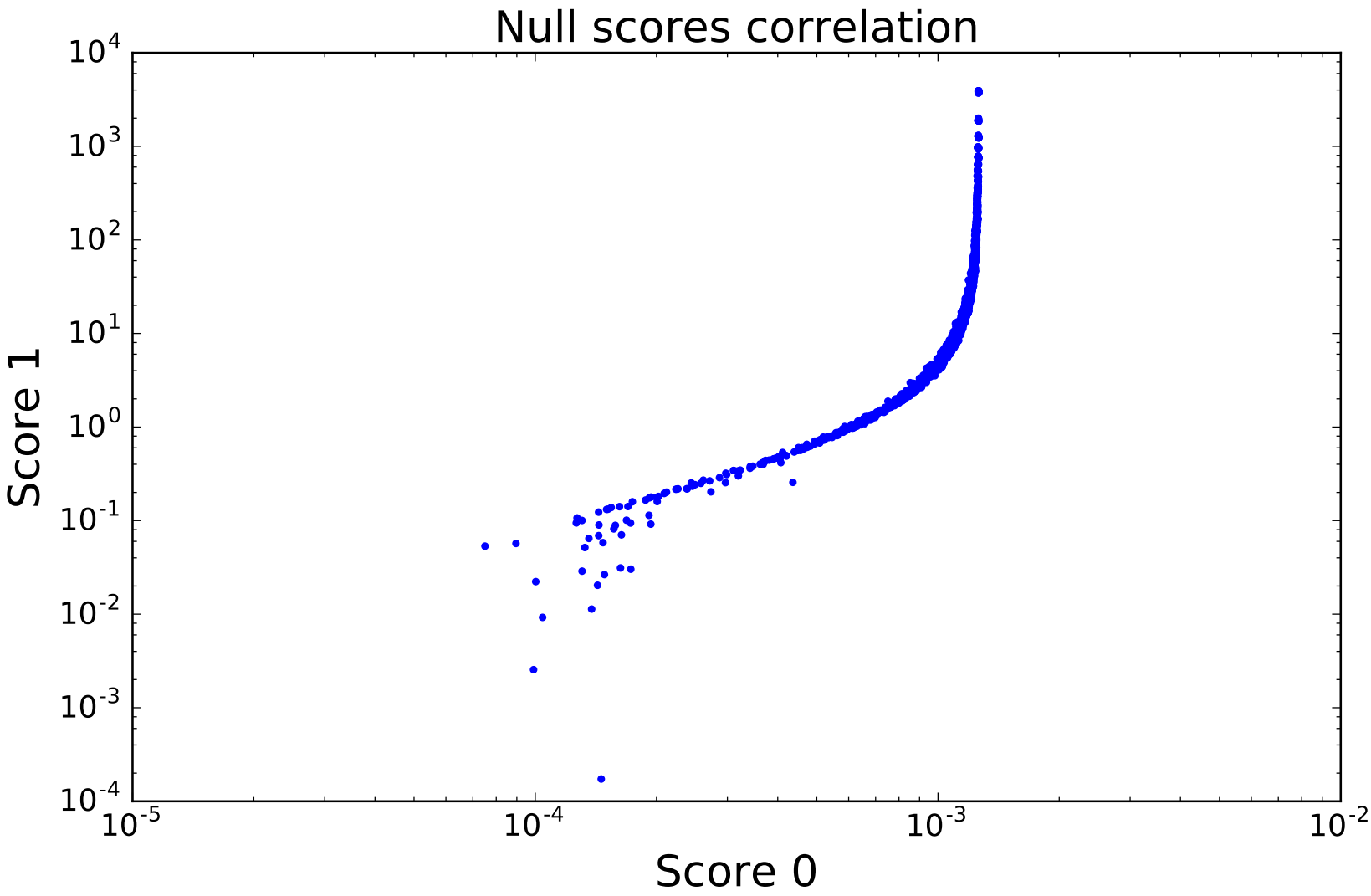


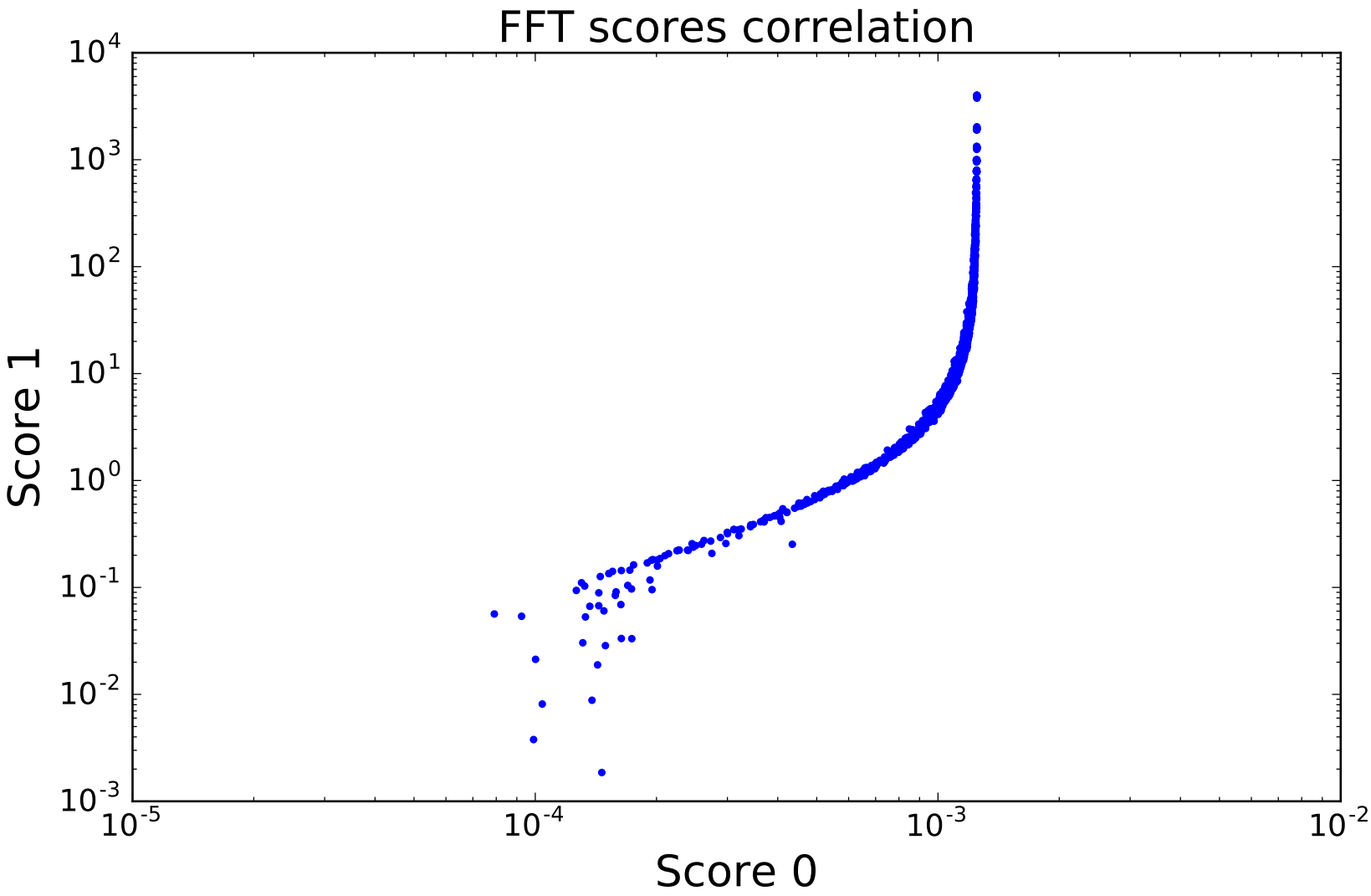
Bad WT example

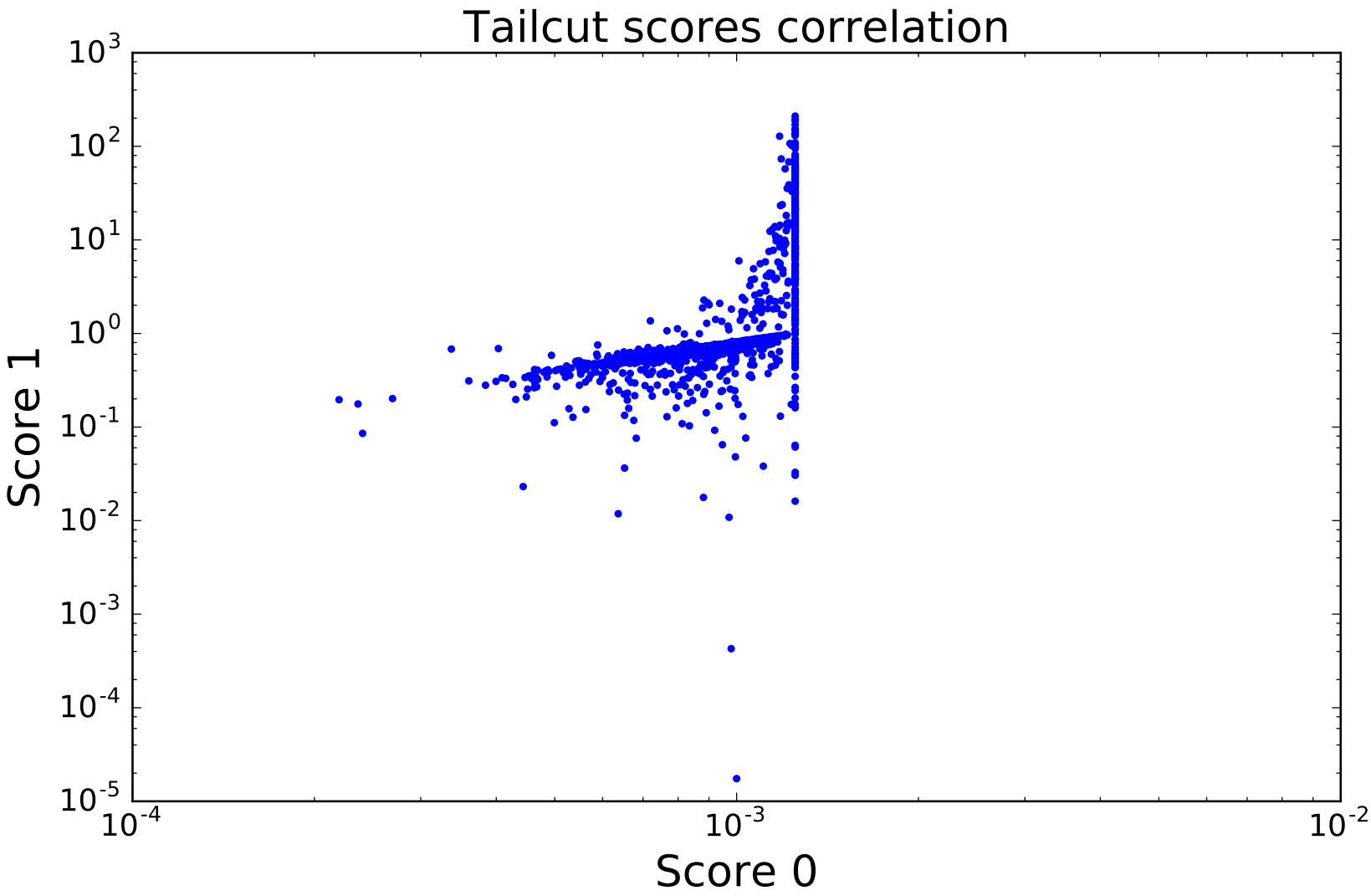


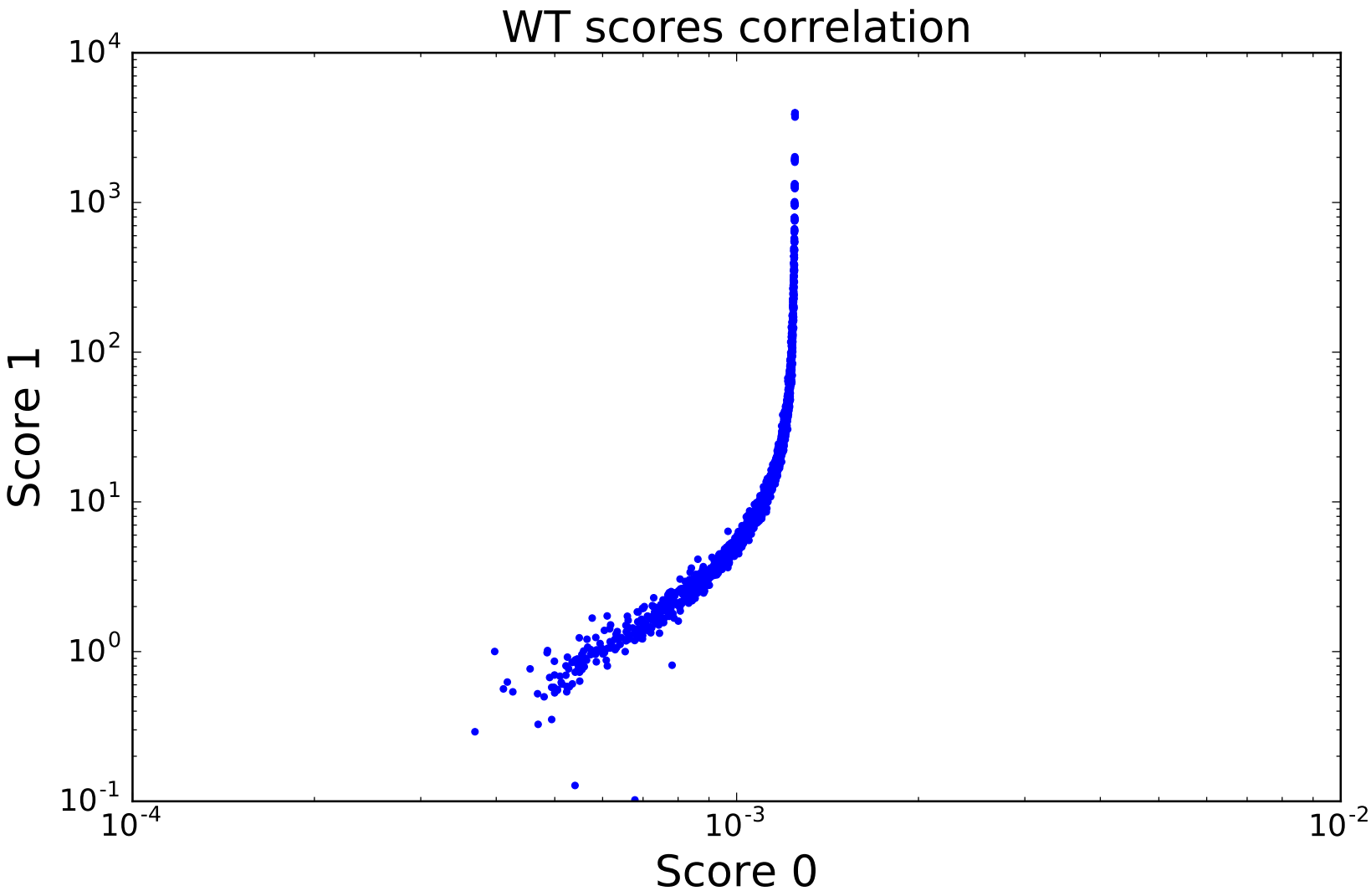
Good WT example




Null scores correlation
Score 1
Score 0
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-4}
10^{-3}
10^{-2}


FFT scores correlation
Score 1
Score 0
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-5}
10^{-4}
10^{-3}
10^{-2}


Tailcut scores correlation
Score 1
Score 0
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-4}
10^{-3}
10^{-2}


WT scores correlation
Score 1
Score 0
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-4}
10^{-3}
10^{-2}

Energy conservation

Definition

Definition

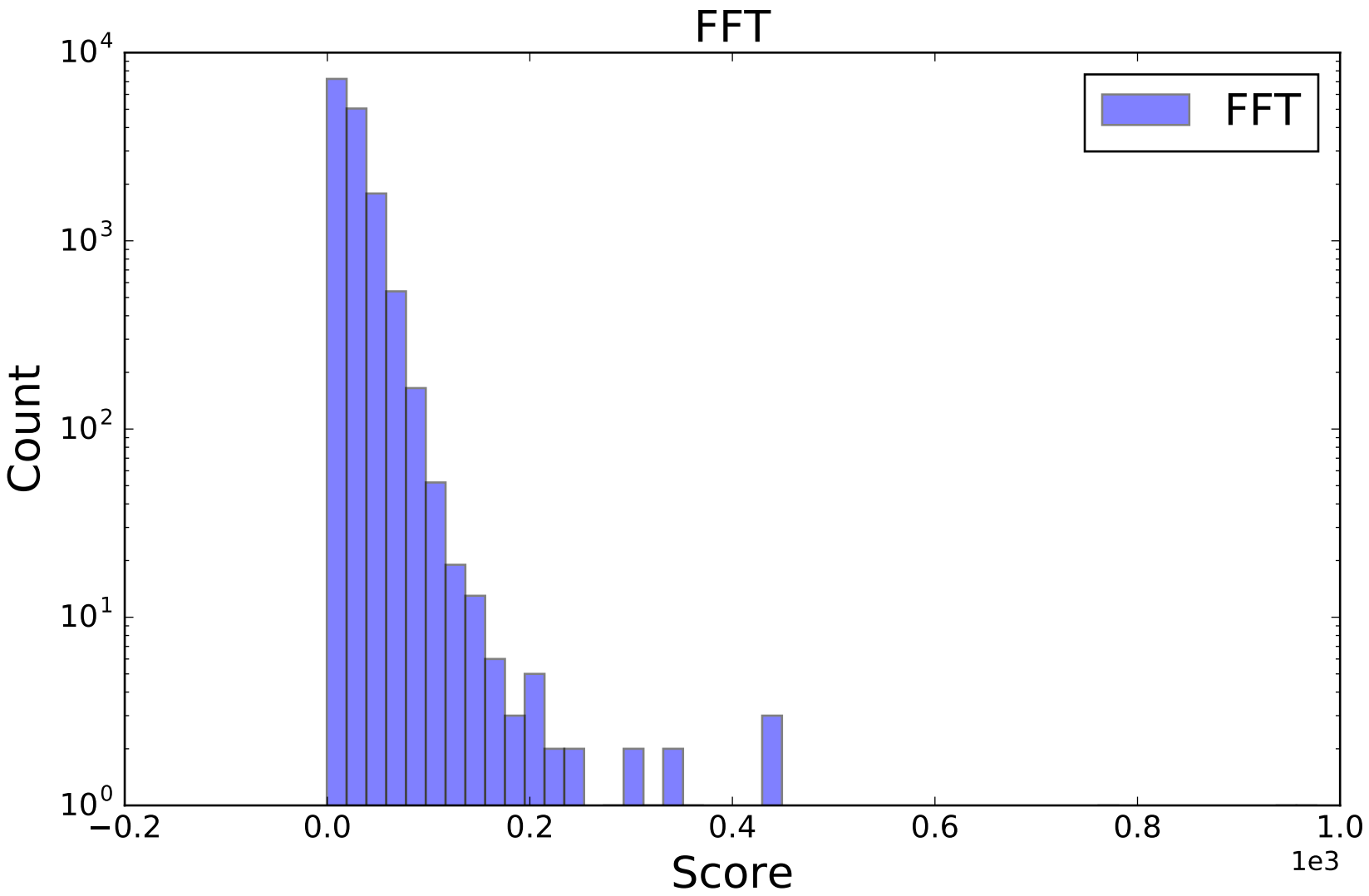
The error on energy is given by:

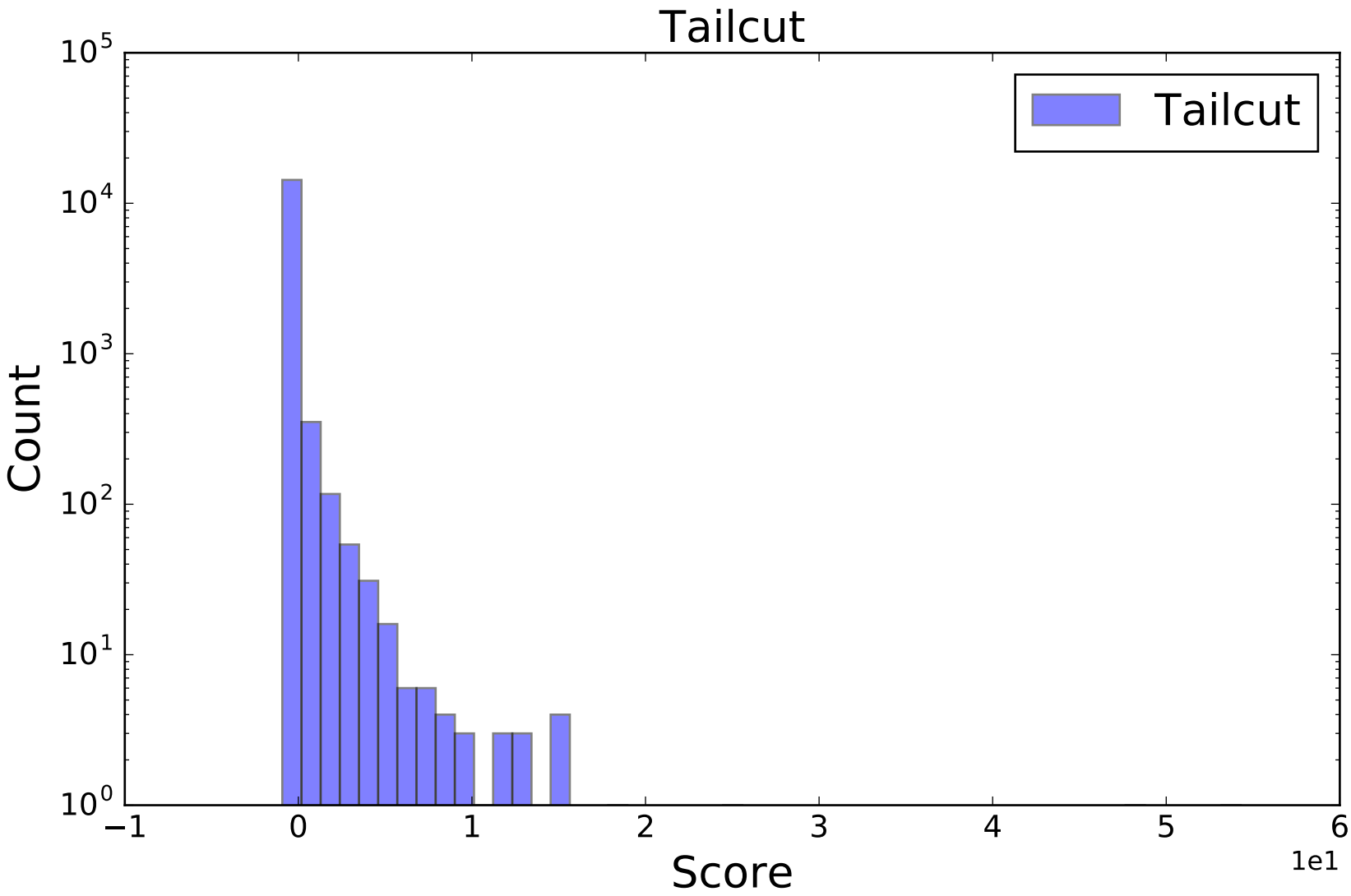
$$\frac{\sum_i \hat{\mathbf{s}}_i - \sum_i \mathbf{s}^*{}_i}{\sum_i \mathbf{s}^*{}_i}$$

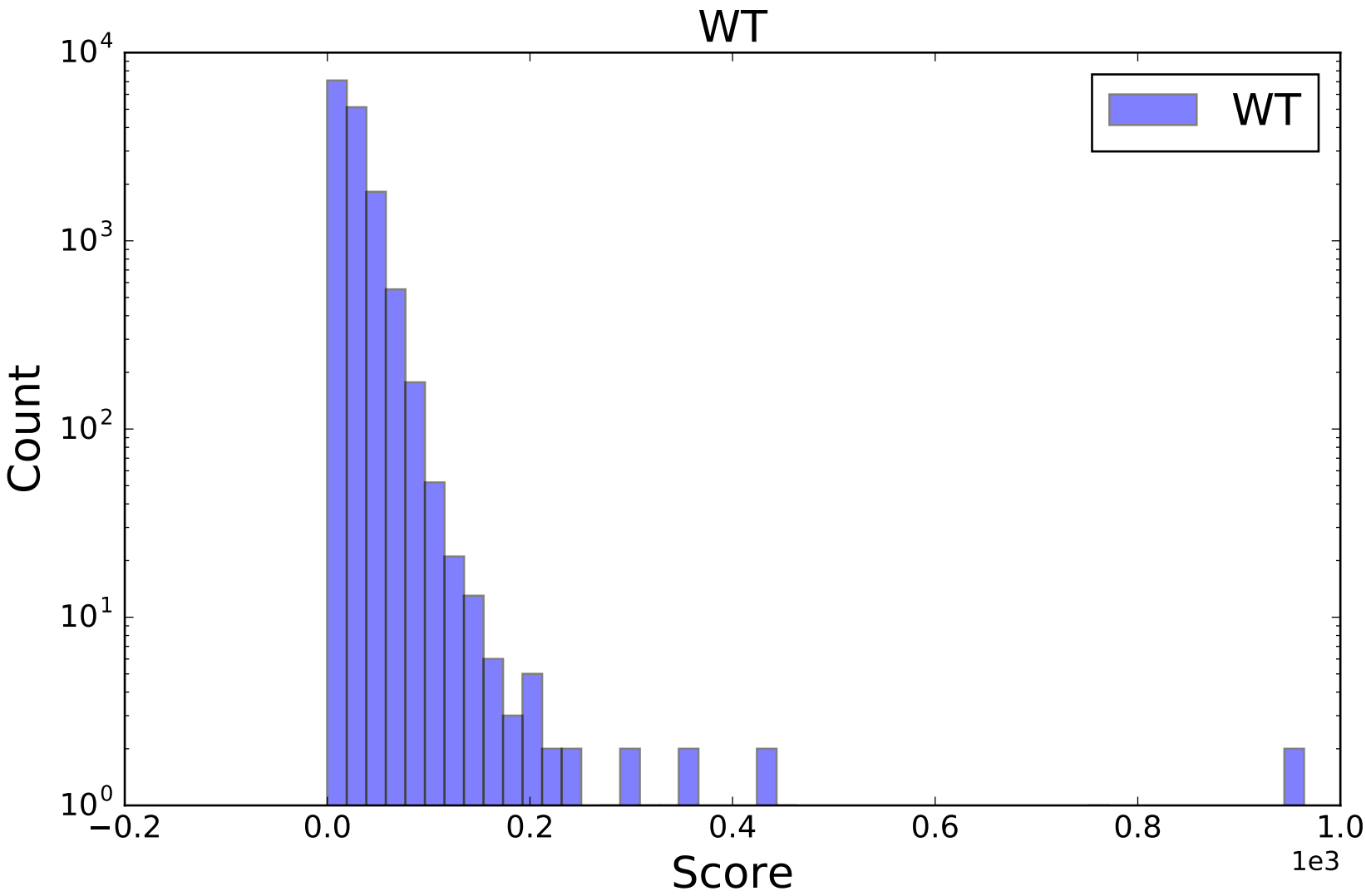
Where:

- $\hat{\mathbf{s}}$ is the output image (the "cleaned" image) $\in \mathbb{R}^d$
- $\mathbf{s}^*$ is the reference image (the "clean" image) $\in \mathbb{R}^d$
- i is the index of a PMT (i.e. of a pixel) within an image

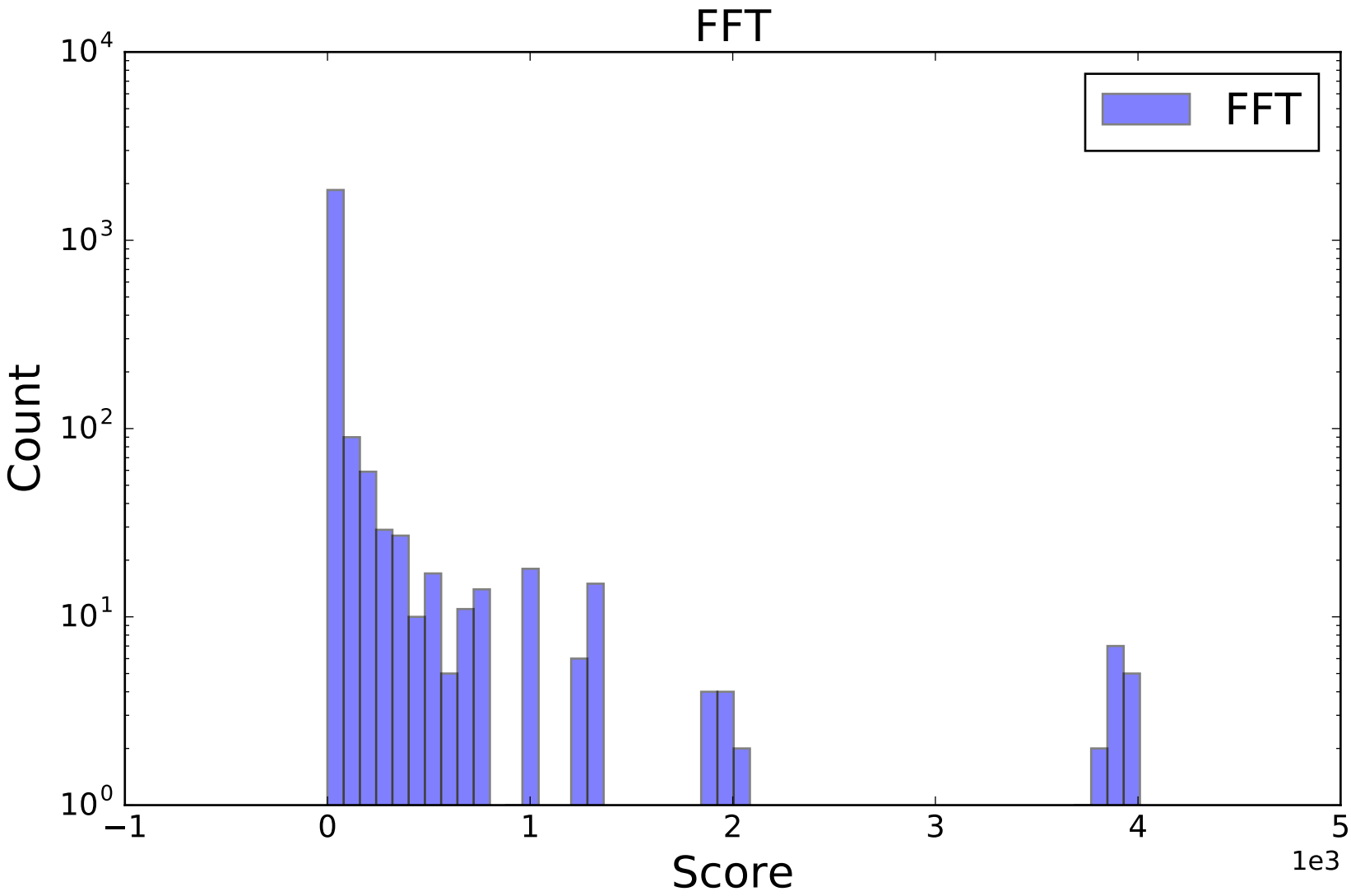
Gamma

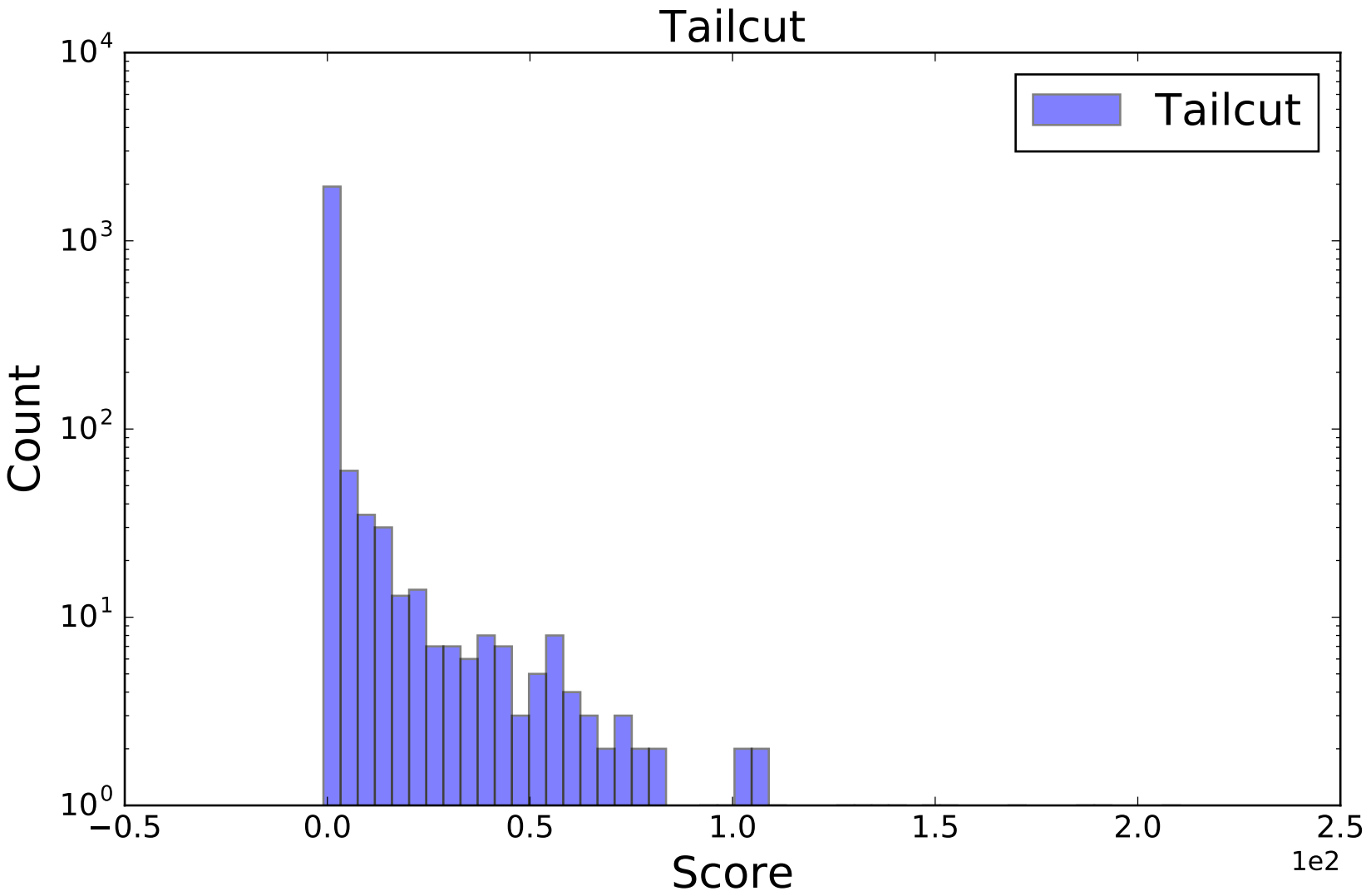

FFT
FFT
Count
Score
1e3
10^0
10^1
10^2
10^3
10^4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

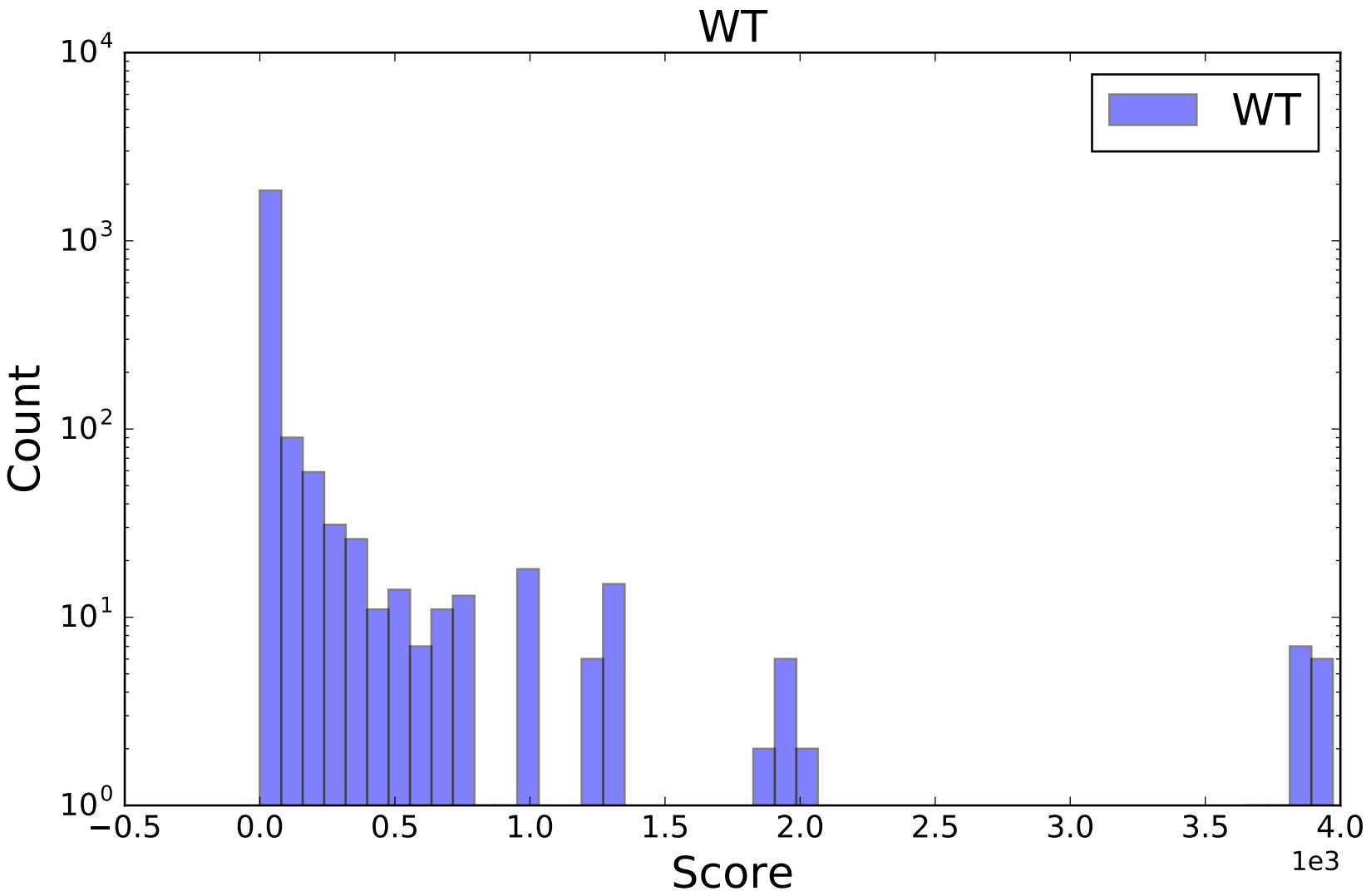

Tailcut
Tailcut
Count
Score
1e1
10^0
10^1
10^2
10^3
10^4
10^5
−1
0
1
2
3
4
5
6


WT
WT
Count
10^4
10^3
10^2
10^1
10^0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1e3
Score

Proton


FFT
FFT
Count
10^4
10^3
10^2
10^1
10^0
−1
0
1
2
3
4
5
1e3
Score

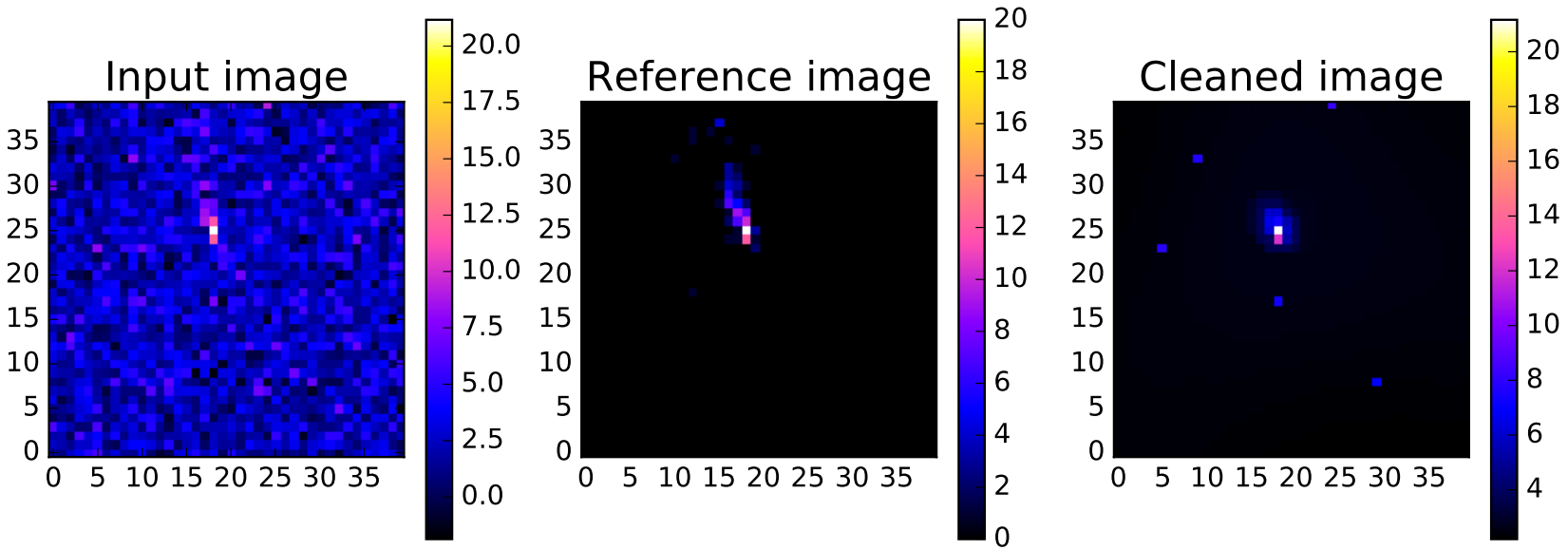

Tailcut
Tailcut
Count
10^4
10^3
10^2
10^1
10^0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Score
1e2


WT
WT
Count
10^4
10^3
10^2
10^1
10^0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e3
Score

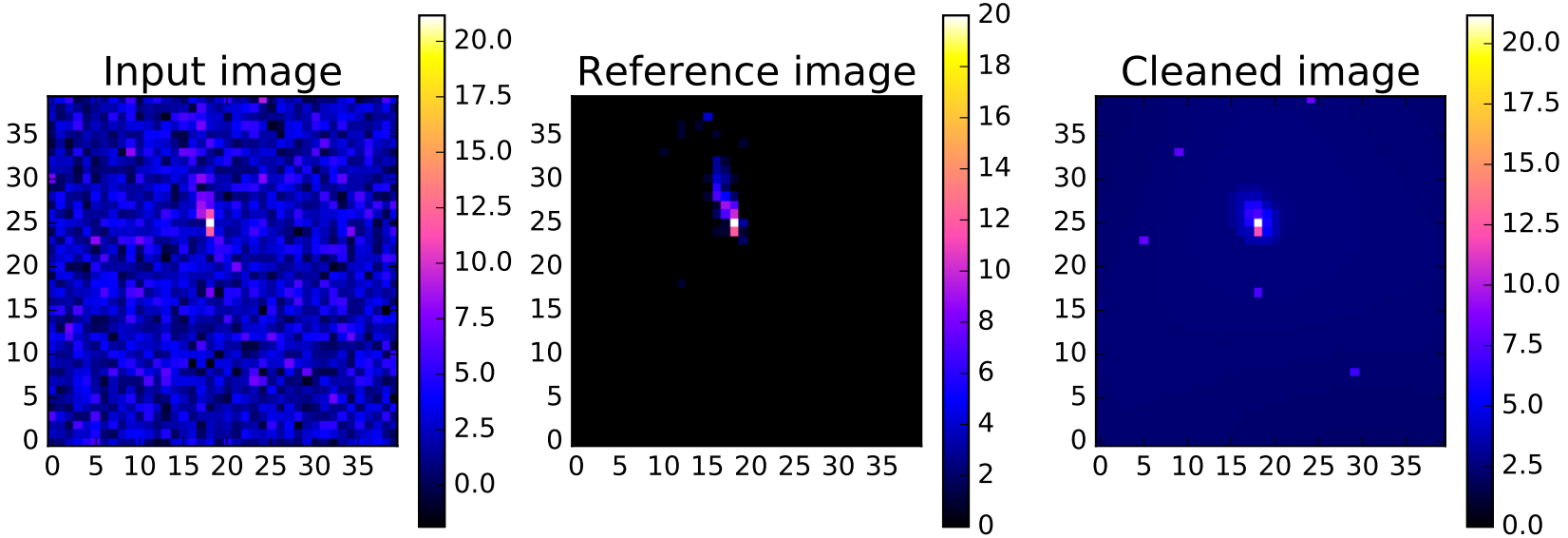
Analysis

Background > 0 with FFT and WT

Look at the scale of the cleaned image...



It's more obvious with 'vmin=0' in matplotlib...



- It explains the shift on all plots for FFT and WT (in addition to the wrong threshold ?)
- It should explain why the classification works despite that mistake (the background is ignored by the classifier)

Conclusion

TODO

TODO

- fix the issue with the non null background with FFT and WT by using the right option in mrtransform / mrfilter (also remember that some pixels in the input image have negative value...)
- use vmin=0 in matplotlib to make this mistake more visible
- plot score vs N_{pe}, ...
- add metadata into input FITS files (E_{MC}, ...)
- use these metadata to plot $\log(E_{MC})$ vs N_{pe}, score vs $\log(E_{MC})$, ...
- plot the Chi^2
- tag images where the signal is on the border
- optimize meta parameters (multicriteria optimization)

References I