

# Hardware and operational factors affecting the data pre-selection in FITACF

**Emma Bland**<sup>1</sup>, Pasha Ponomarenko<sup>2</sup>, Kathryn McWilliams<sup>2</sup> & Nozomu Nishitani<sup>3</sup>

<sup>1</sup>University Centre in Svalbard, Norway

<sup>2</sup>University of Saskatchewan, Canada

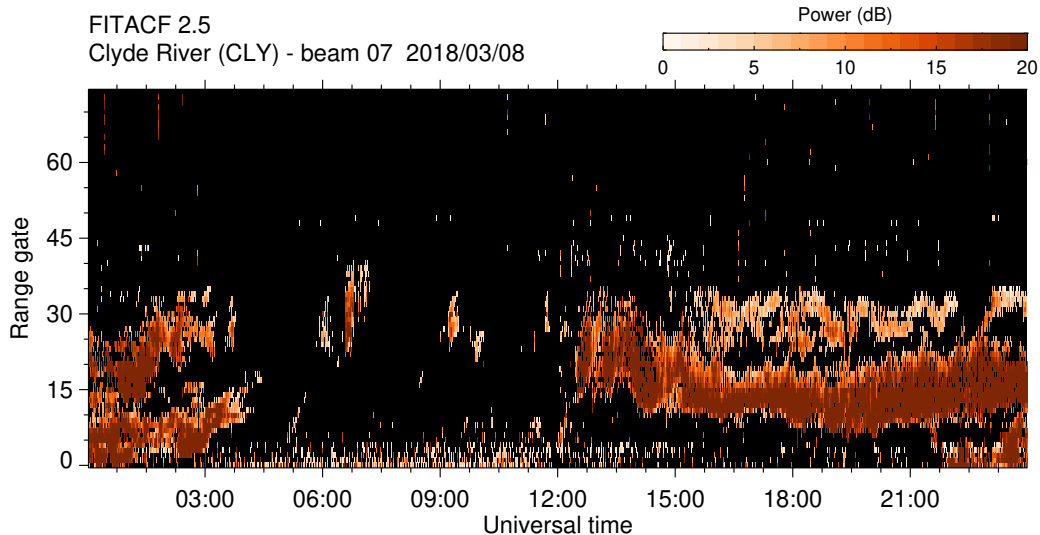
<sup>3</sup>Nagoya University, Japan

Compared to FITACF 2.5, data processed with FITACF 3.0 generally includes. . .

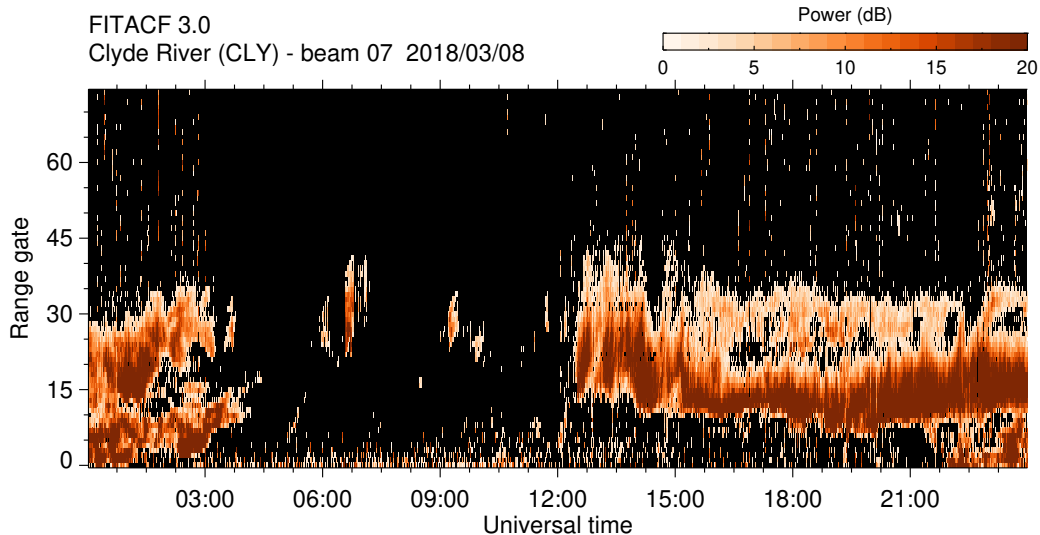
- ▶ More good-quality backscatter from coherent scattering targets
- ▶ Occasional high-power vertical 'streaks' in range-time plots
- ▶ Additional low-power 'noise' echoes

**However, these differences vary significantly from radar to radar**

# Example comparison between FITACF 2.5 and FITACF 3.0



# Example comparison between FITACF 2.5 and FITACF 3.0





These differences relate to the **data pre-selection**, not the actual fitted parameters

How does FITACF define a 'good' ACF?

- ▶ Signal-to-noise ratio threshold:  $\text{SNR} > 1$
- ▶ FITACF 2.5 does some additional filtering based on ACF shape

# Data pre-selection

These differences relate to the **data pre-selection**, not the actual fitted parameters

## How does FITACF define a 'good' ACF?

- ▶ Signal-to-noise ratio threshold:  $\text{SNR} > 1$
- ▶ FITACF 2.5 does some additional filtering based on ACF shape

## How is the noise level calculated?

```
/* look for the lowest 10 values of lag0 power and average to get the noise level. Ignore values  
that are exactly 0. If you can't find 10 useable values within the first 1/3 of the sorted power  
list, then just use whatever you got in that first 1/3. If you didn't get any useable values, then  
use the NOISE parameter */
```

See companion talk: *On atmospheric noise estimate by SuperDARN software* (P. Ponomarenko)

# Outline of this talk

## Key points

- ▶ These data quality issues are not directly related to the fitting algorithm
- ▶ ACF shape rejection criterion in FITACF 2.5 removes most of the contamination. . .  
...but it also rejects a lot of good quality data

# Outline of this talk

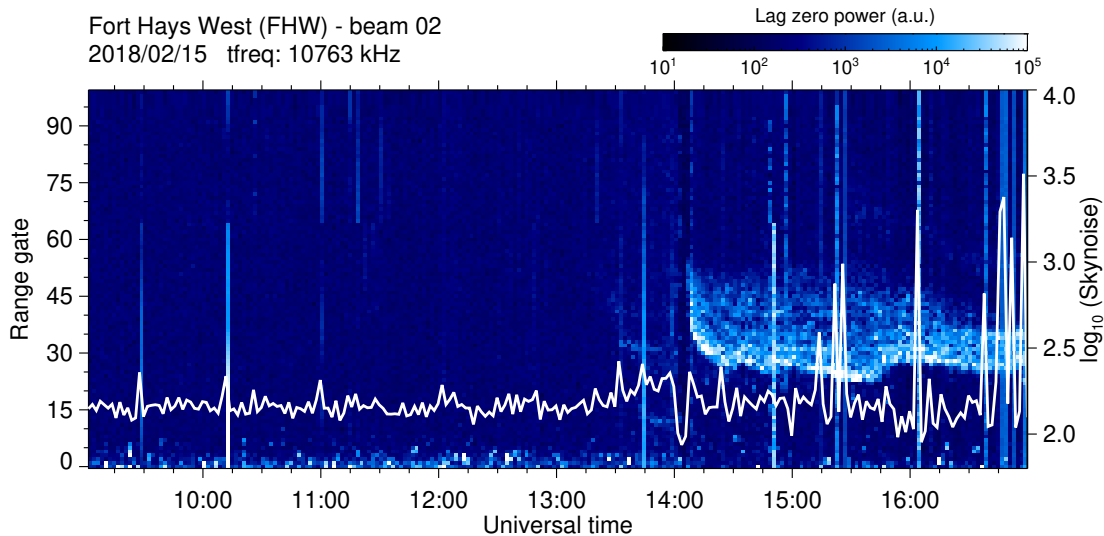
## Key points

- ▶ These data quality issues are not directly related to the fitting algorithm
- ▶ ACF shape rejection criterion in FITACF 2.5 removes most of the contamination. . .  
...but it also rejects a lot of good quality data

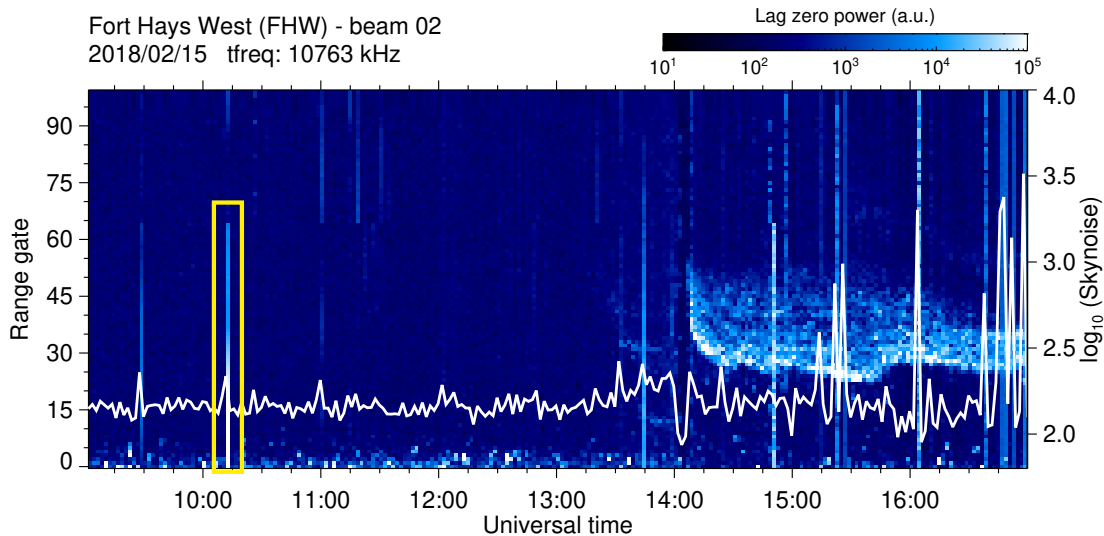
## We need *solutions*, not more problems. . .

- ▶ Recommendations for control program design  
e.g. number of averages, group range span
- ▶ Possibility to correct some issues related to radar OS/hardware
- ▶ New filtering routine in RST  
To mitigate the impacts of noise contamination when present

# High-power streaks — type I



# High-power streaks — type I



# High-power streaks — type I

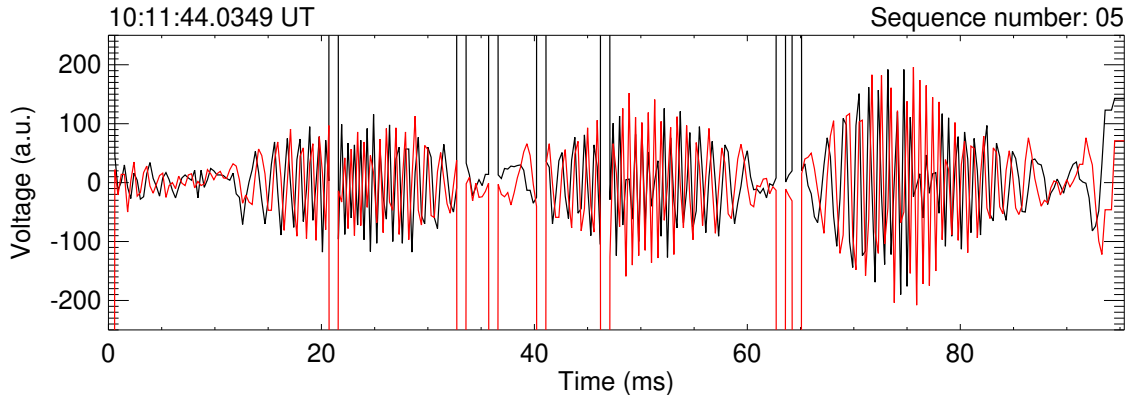
**Usually just 1–2 pulse sequences are affected**

See also:

*Rankin Inlet Interference Summary (May 1st – August 31st 2020)*, report by Cooper Robertson, University of Saskatchewan, 2020

# High-power streaks — type I

Usually just 1–2 pulse sequences are affected



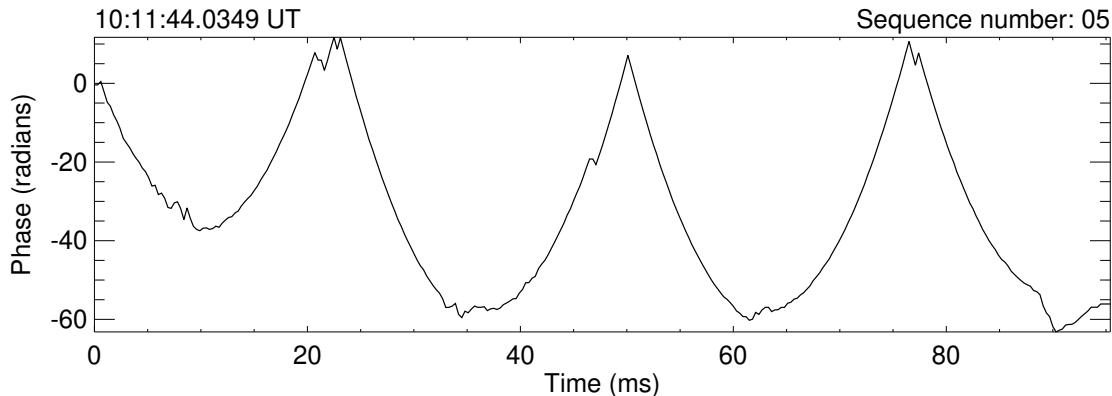
See also:

*Rankin Inlet Interference Summary (May 1st – August 31st 2020)*, report by Cooper Robertson, University of Saskatchewan, 2020



# High-power streaks — type I

Usually just 1–2 pulse sequences are affected

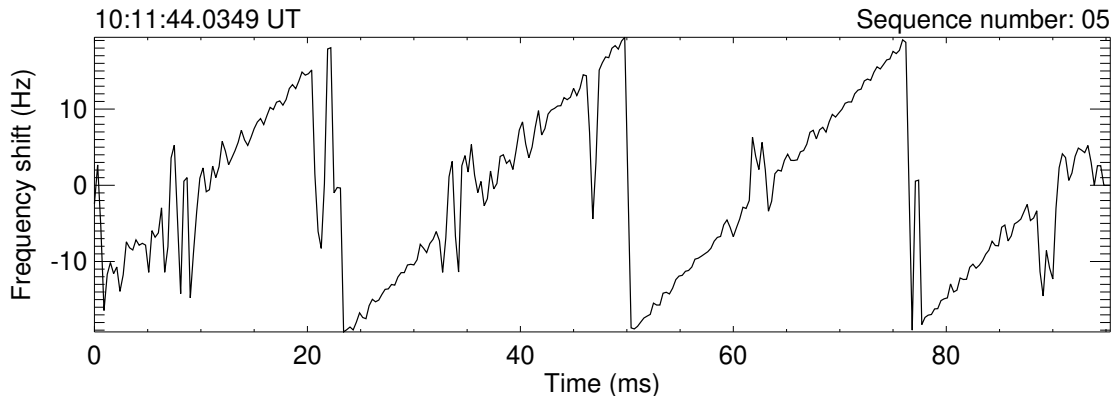


See also:

*Rankin Inlet Interference Summary (May 1st – August 31st 2020)*, report by Cooper Robertson, University of Saskatchewan, 2020

# High-power streaks — type I

Usually just 1–2 pulse sequences are affected



See also:

*Rankin Inlet Interference Summary (May 1st – August 31st 2020)*, report by Cooper Robertson, University of Saskatchewan, 2020

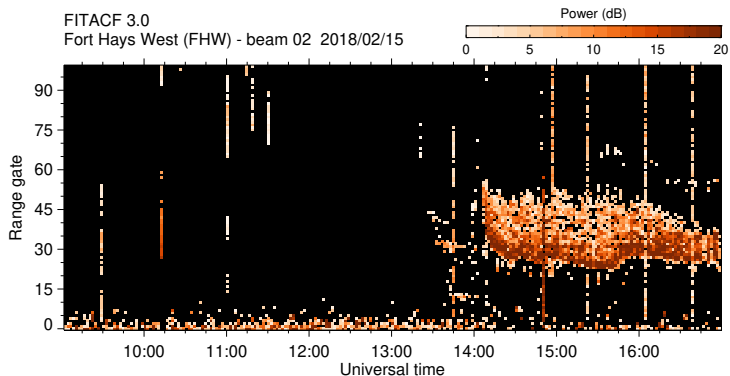
## High-power streaks — type II

**Suspected pulses from other SuperDARN radars**

# High-power streaks

## Can we solve this problem?

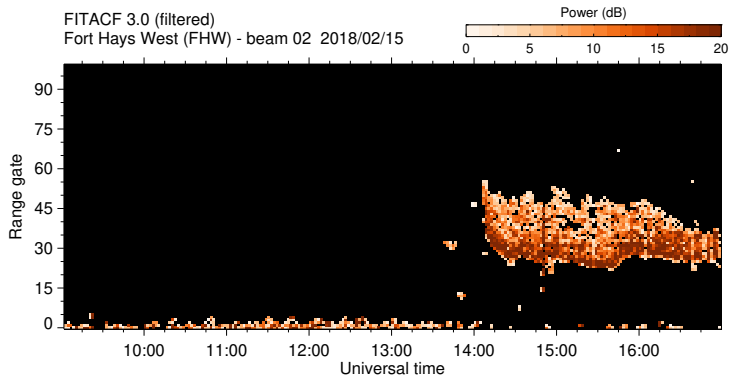
- ▶ Clear frequency search? *No, the interference has short duration*
- ▶ Filter bad sequences before running `make_raw`? *Cannot monitor what is being removed*
- ▶ Filter after running `make_fit`? *Yes...*



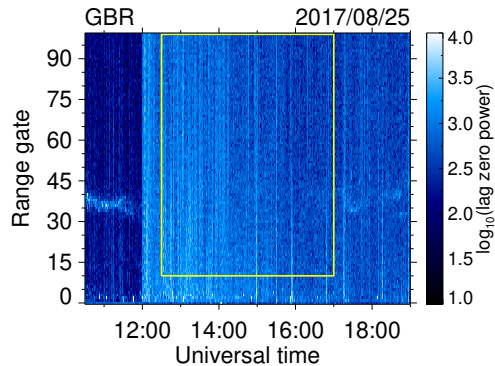
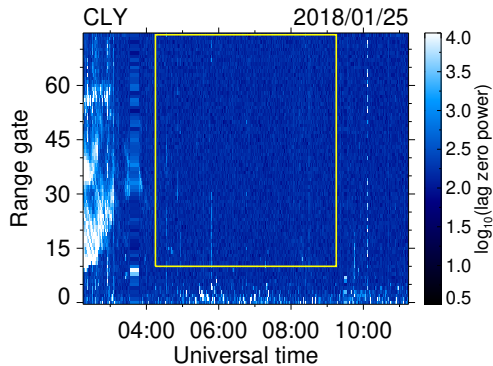
# High-power streaks

## Can we solve this problem?

- ▶ Clear frequency search? *No, the interference has short duration*
- ▶ Filter bad sequences before running `make_raw`? *Cannot monitor what is being removed*
- ▶ Filter after running `make_fit`? *Yes...*



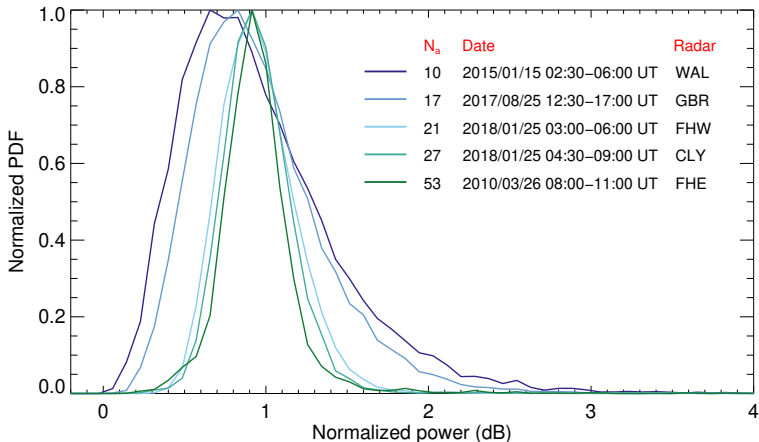
# Shape of the noise distribution



See also the companion talk: *On atmospheric noise estimate by SuperDARN software* (P. Ponomarenko)

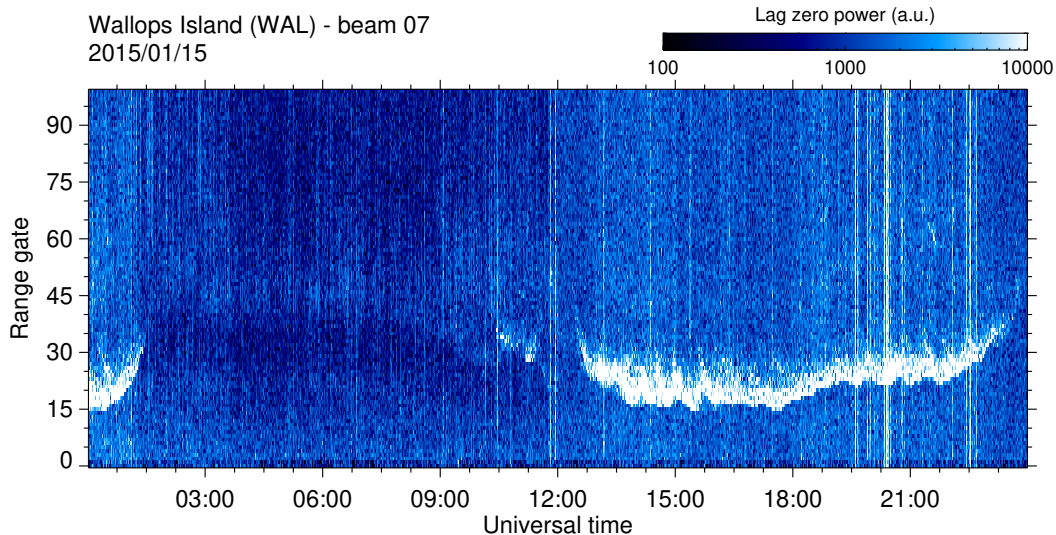
# Shape of the noise distribution

Increased low-power contamination when the noise distribution has a high-power tail



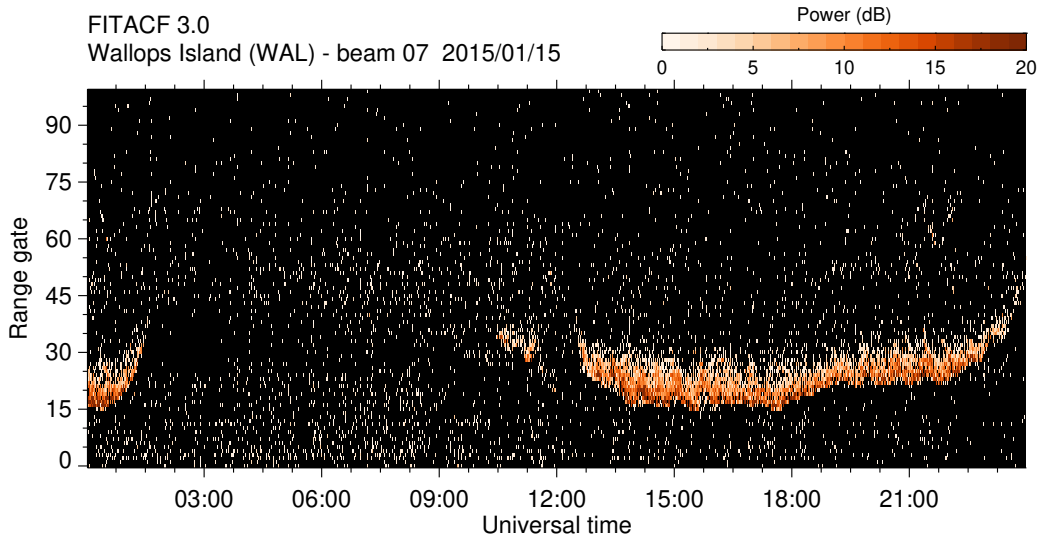
See also the companion talk: *On atmospheric noise estimate by SuperDARN software* (P. Ponomarenko)

# Shape of the noise distribution

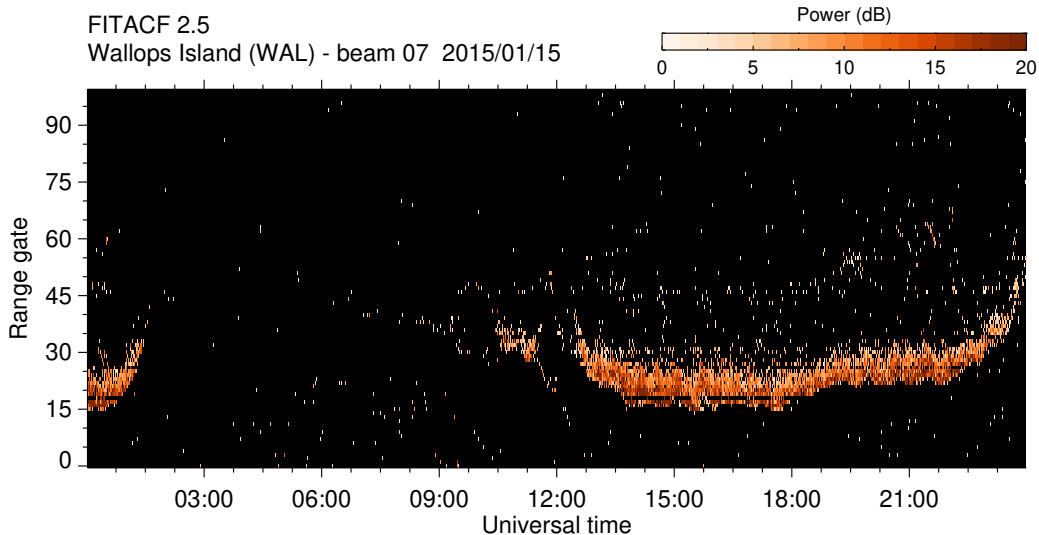




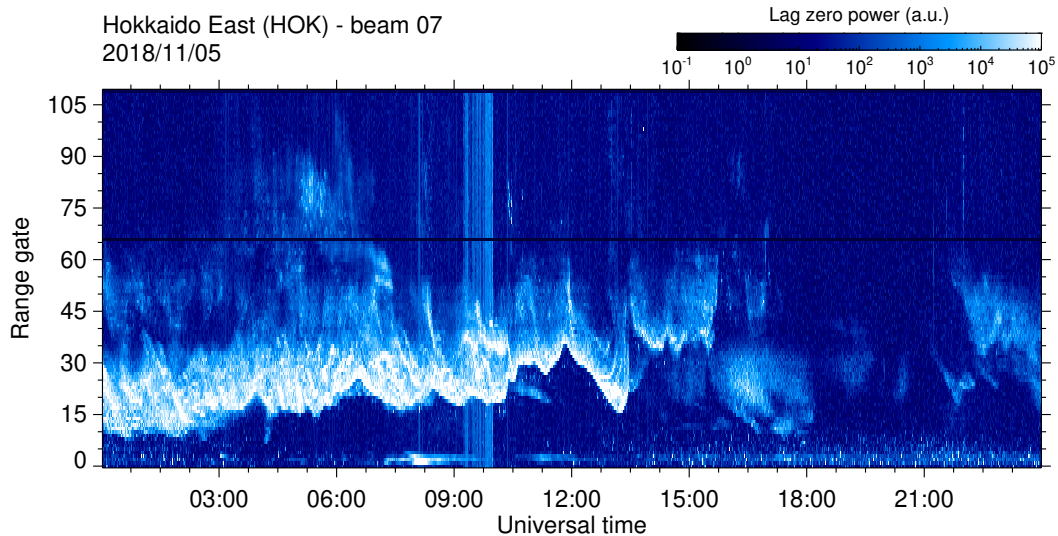
# Shape of the noise distribution



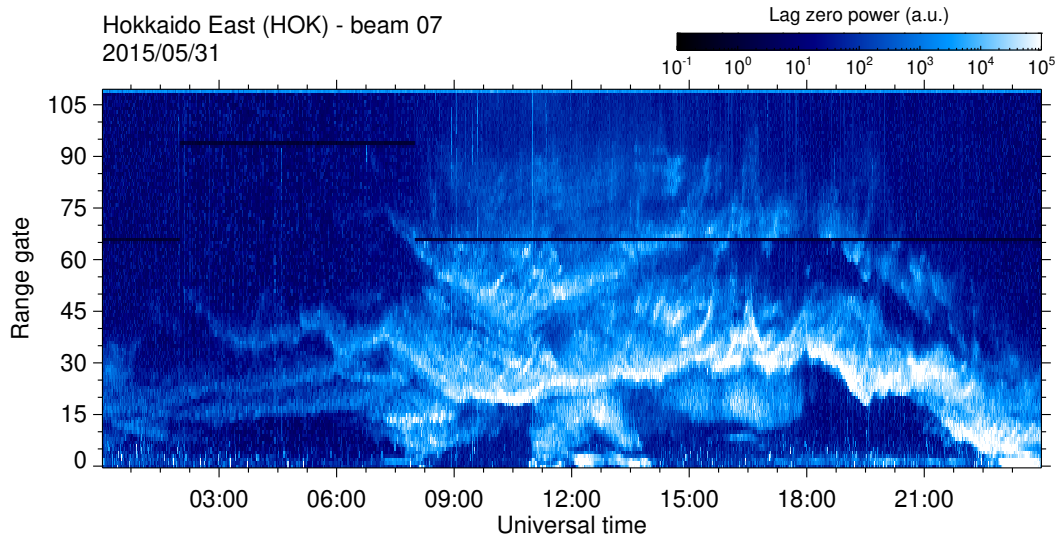
# Shape of the noise distribution



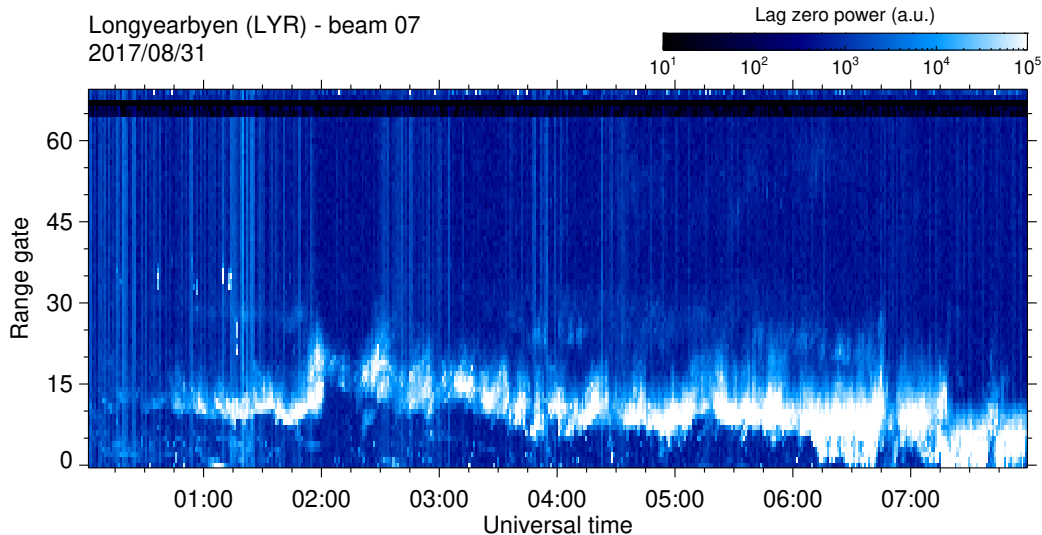
# Lag zero blanking



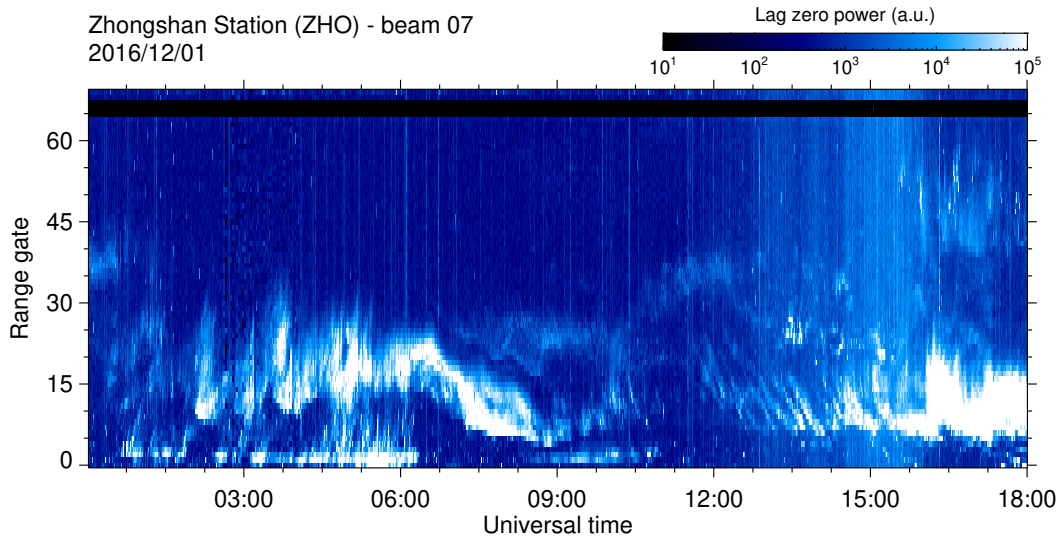
# Lag zero blanking



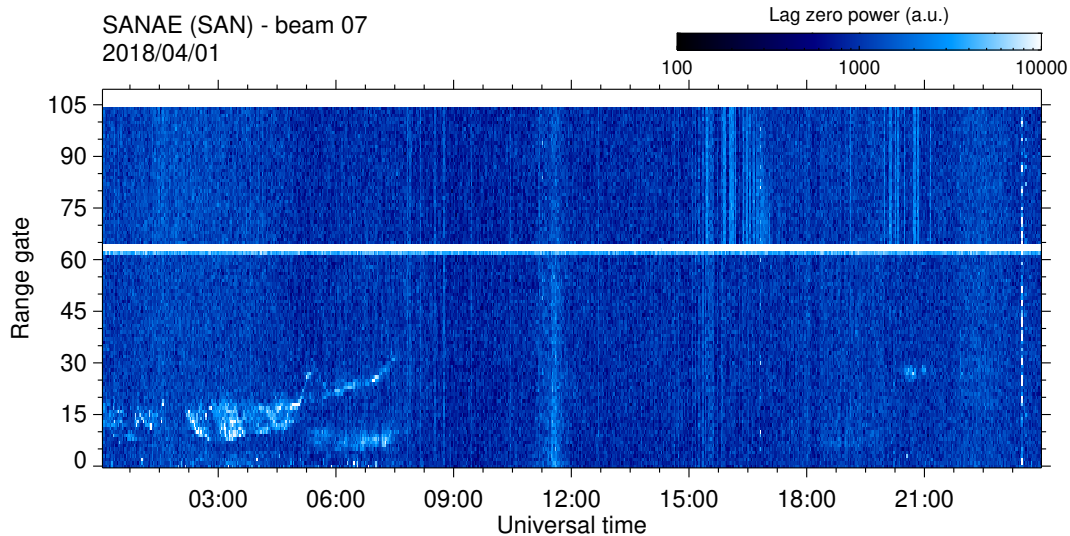
# Lag zero blanking



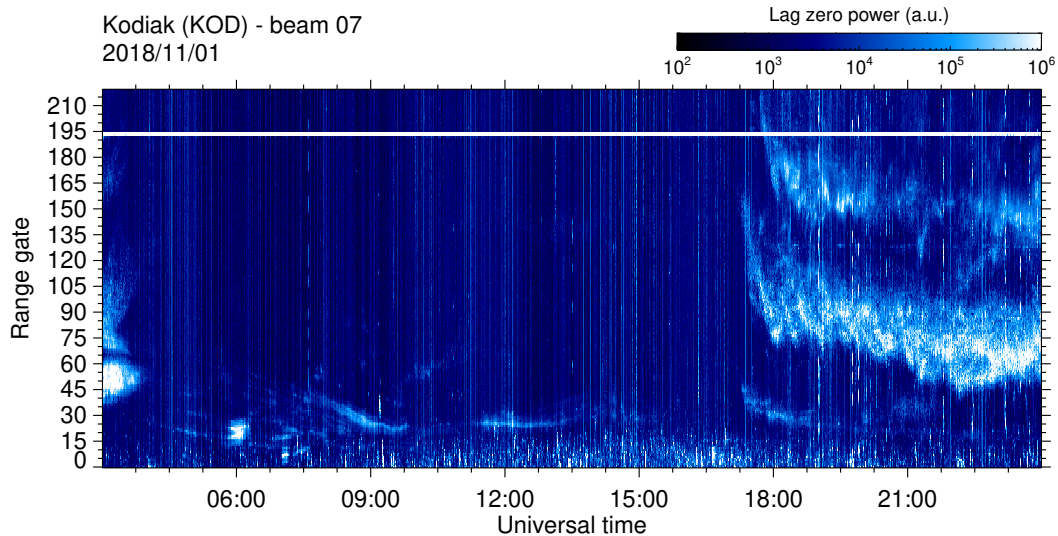
# Lag zero blanking



# Lag zero blanking



# Lag zero blanking





# Data types error in FITACF 2.5

## FITACF 2.5 reads the lag zero power as an integer

Problematic when  $pwr0 \sim 1$

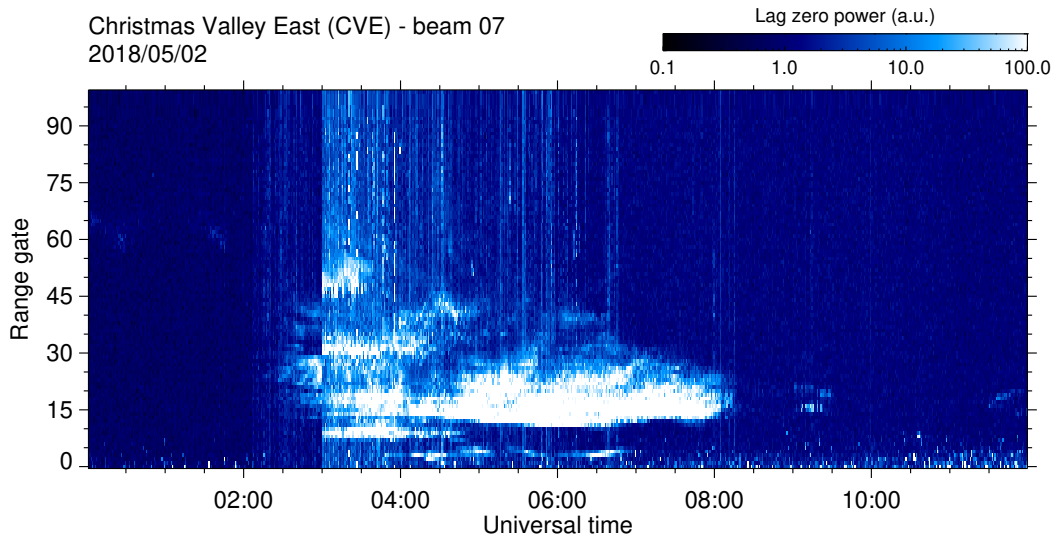
### Recall noise estimation:

- ▶ look for the lowest 10 values of lag0 power and average to get the noise level.
- ▶ Ignore values that are exactly zero
- ▶ If you can't find 10 useable values within the first 1/3 of the sorted power list, then just use whatever you got in that first 1/3.
- ▶ If you didn't get any useable values, then use the NOISE parameter

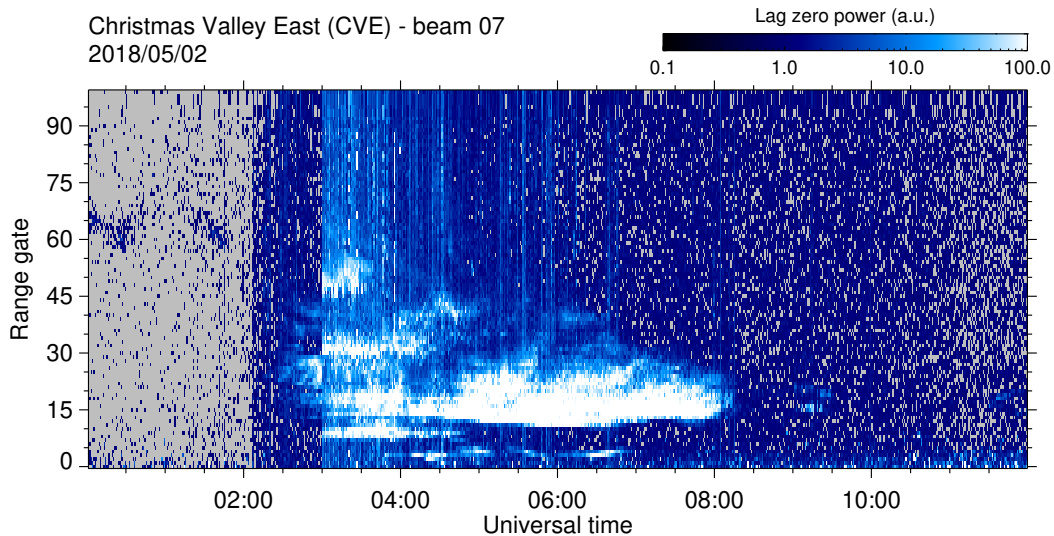
FITACF 2.5 parameter block:

```
struct FitPrm {  
    int channel; /* zero=mono 1 or 2 is stereo */  
    int offset; /* used for stereo badlags */  
    int cp;  
    int xcf;  
    int tfreq;  
    .  
    .  
    .  
    int *lag[2];  
    int *pulse;  
    int *pwr0;  
    int maxbeam;  
    double interfer[3];  
    double bmsep;  
    double phidiff;  
    double tdiff;  
    double vdir;  
};
```

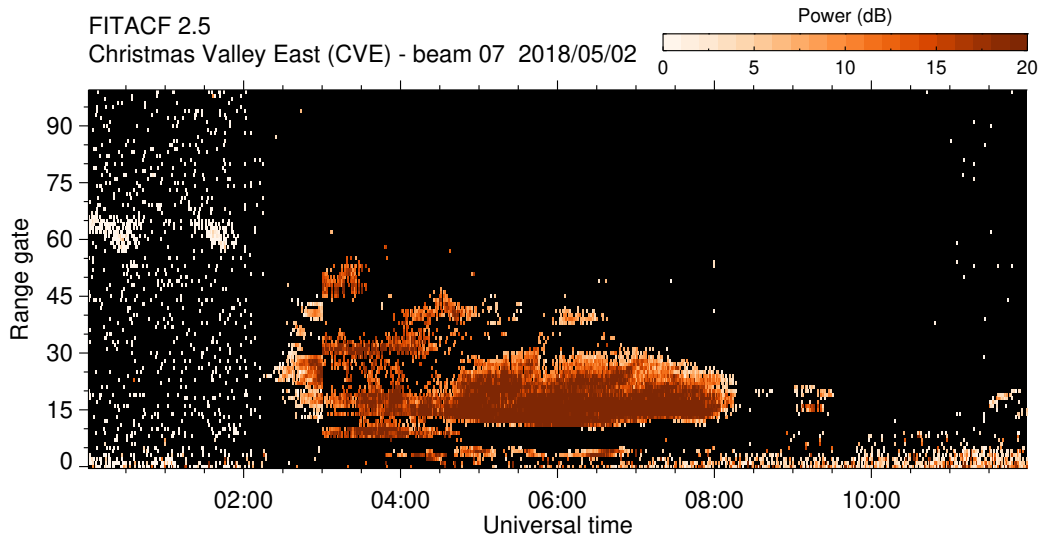
# Data types error in FITACF 2.5



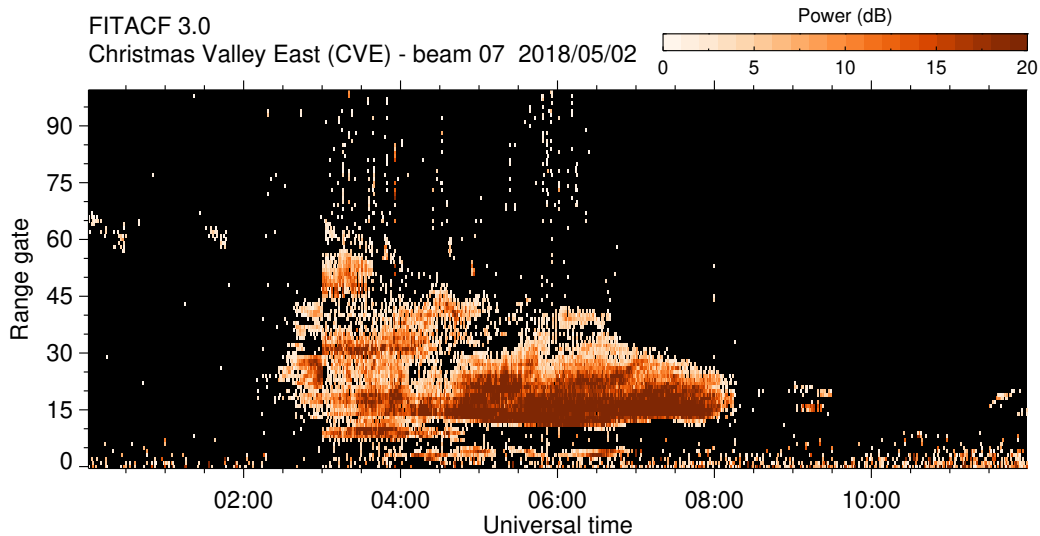
# Data types error in FITACF 2.5



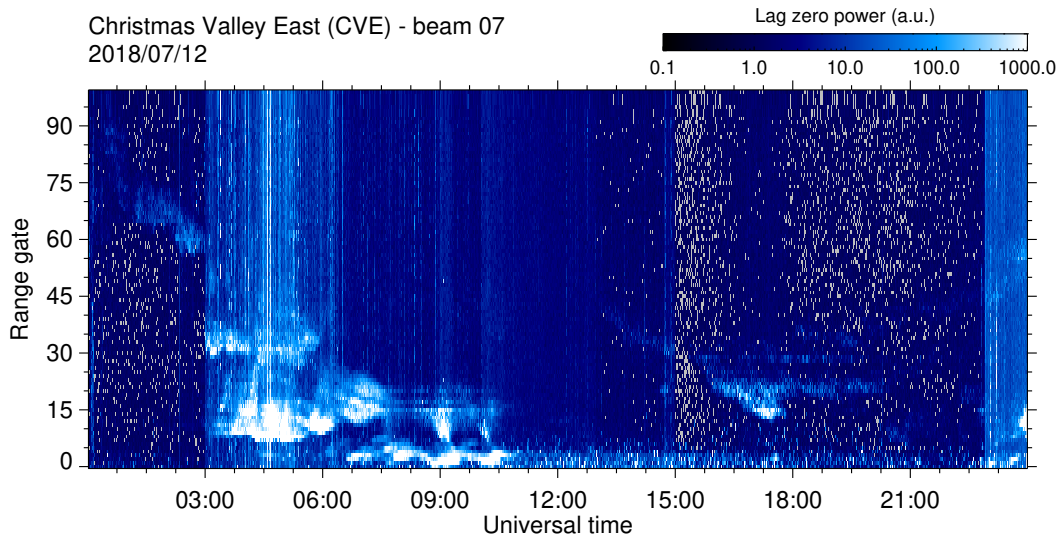
# Data types error in FITACF 2.5



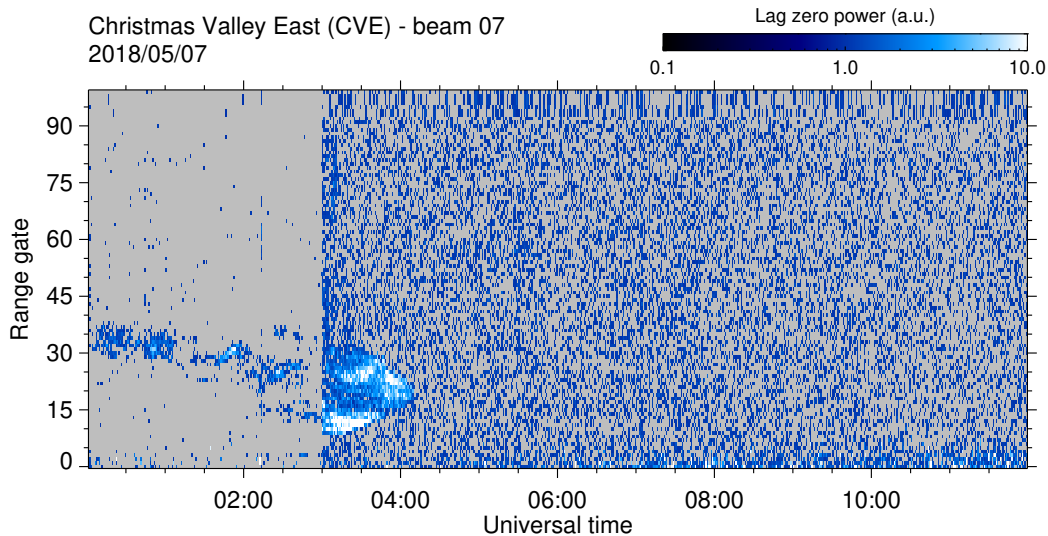
# Data types error in FITACF 2.5



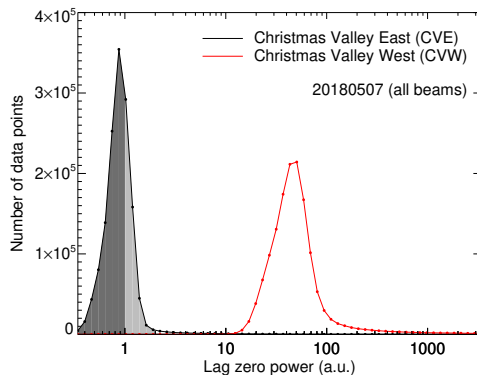
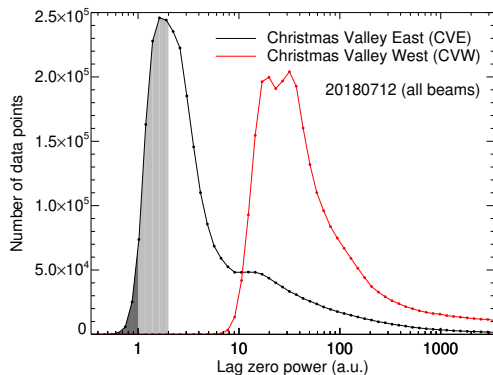
# Data types error in FITACF 2.5



# Data types error in FITACF 2.5



# Data types error in FITACF 2.5



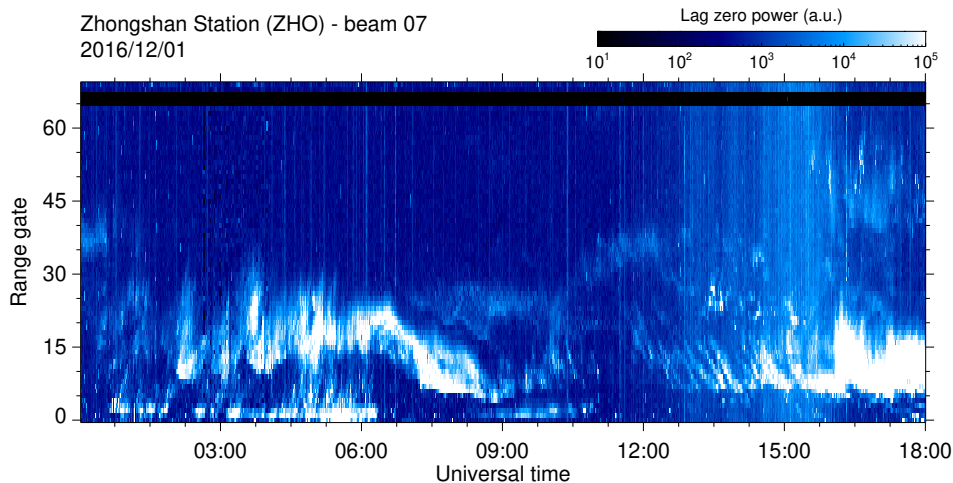
## Two competing effects

- ▶  $\text{pwr0} < 1$ : ignored  $\rightarrow$  noise level overestimated
- ▶  $\text{pwr0} \geq 1$ : rounded down to nearest integer  $\rightarrow$  noise level underestimated



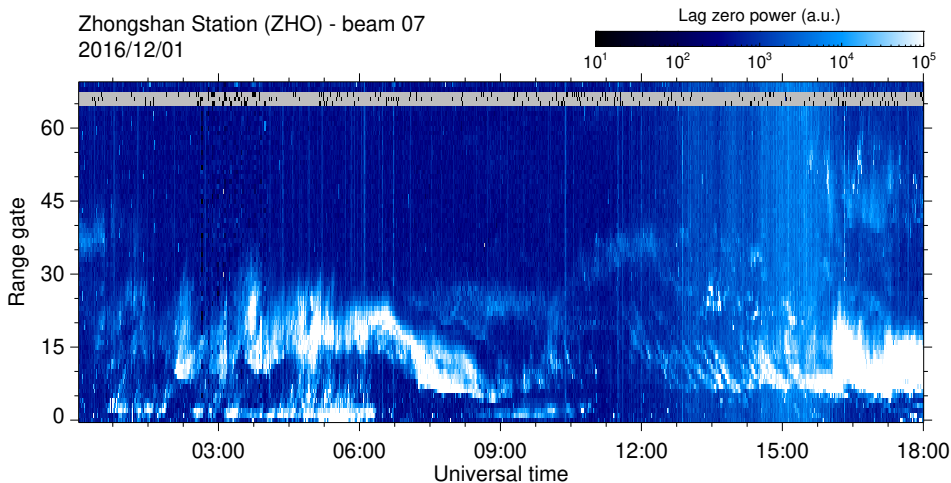
# Data types error in FITACF 2.5

Sometimes the data types error corrects the blanking problem...

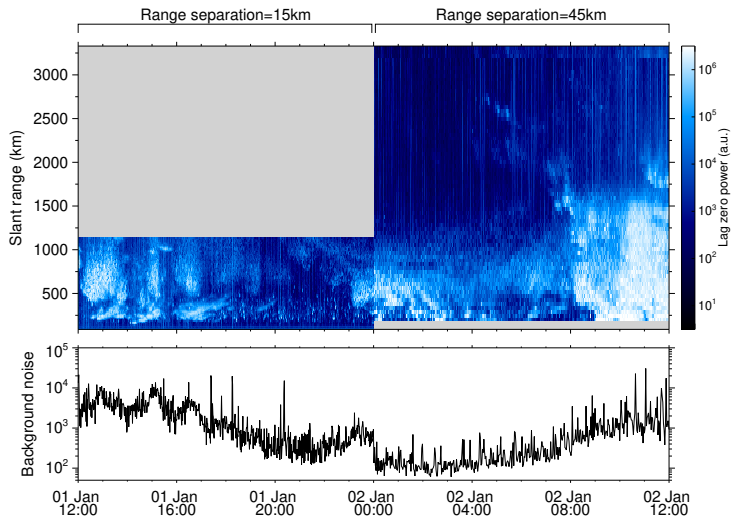


# Data types error in FITACF 2.5

Sometimes the data types error corrects the blanking problem...



# Group range span



# New filtering routine in RST: `fit_speck_removal`

**Purpose:** filter non-Gaussian noise and interference

## Procedure

- ▶ Calculate median of the quality flags in 3x3 range-time grid
- ▶ If matrix median is zero, the quality flag for the central data point (red) is set to 0

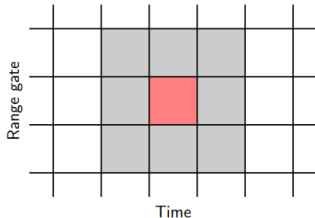
## Notes

- ▶ Filtering is performed separately for each beam and channel
- ▶ No smoothing (unlike conventional median filtering)
- ▶ Side effect: meteor scatter may be removed

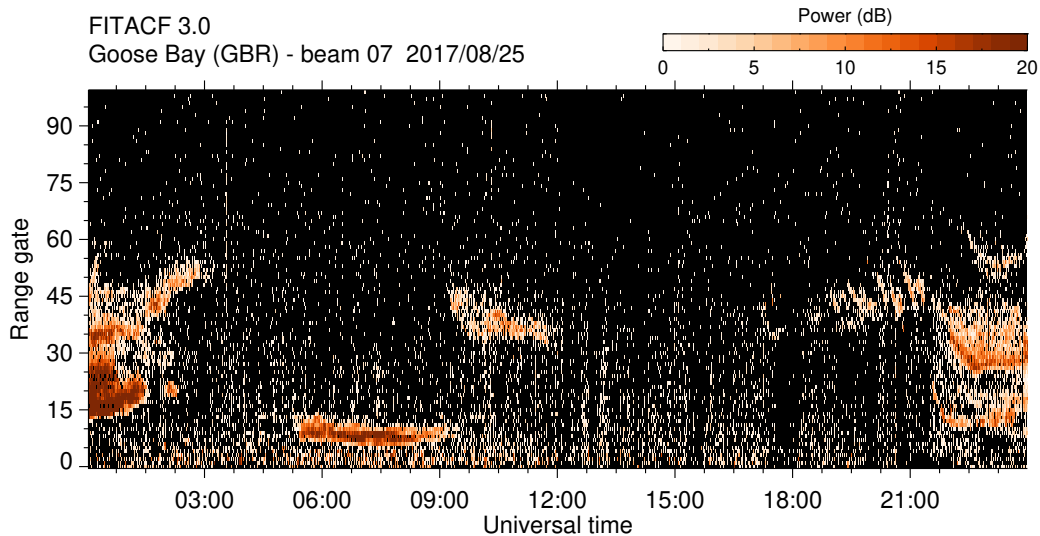
## How to use:

```
fit_speck_removal [filename].fitacf > [filename].filter.fitacf
```

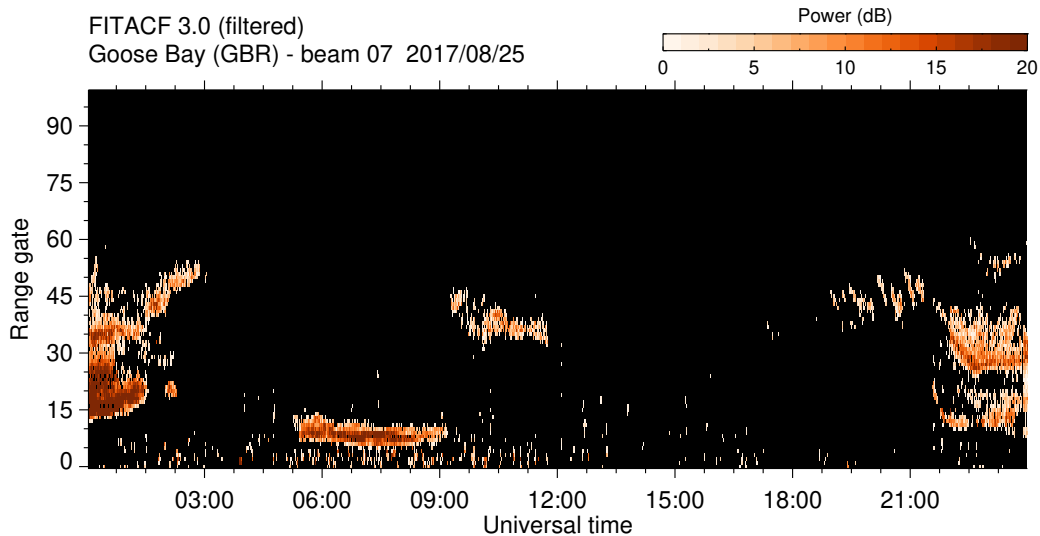
*This new routine will be included in the next minor release of RST*



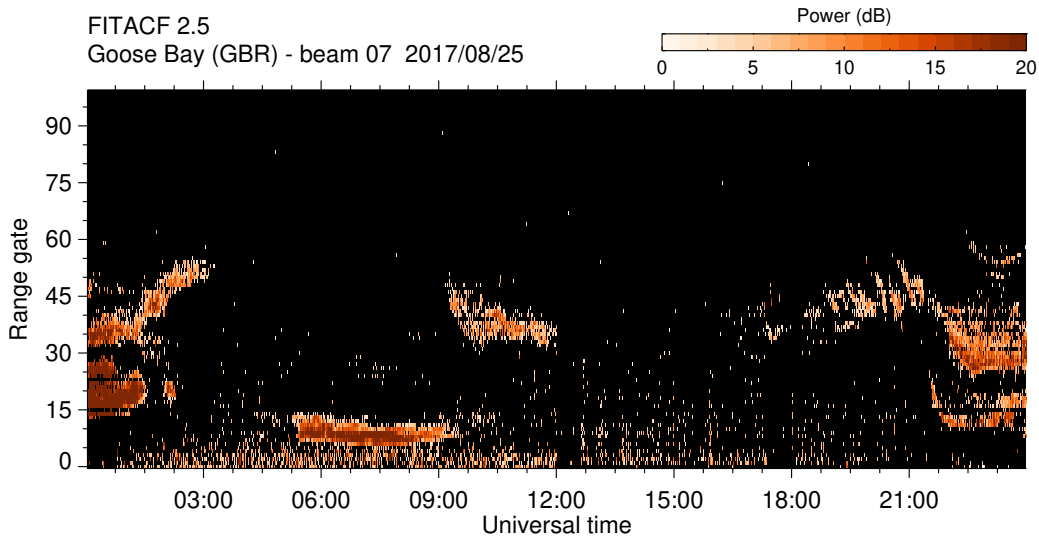
# New filtering routine in RST: `fit_speck_removal`



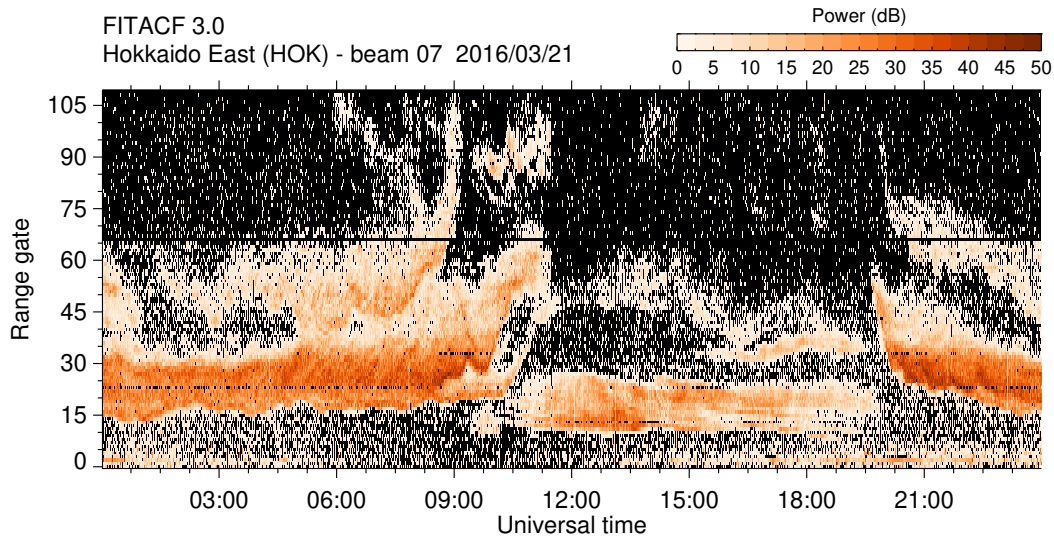
# New filtering routine in RST: `fit_speck_removal`



# New filtering routine in RST: `fit_speck_removal`

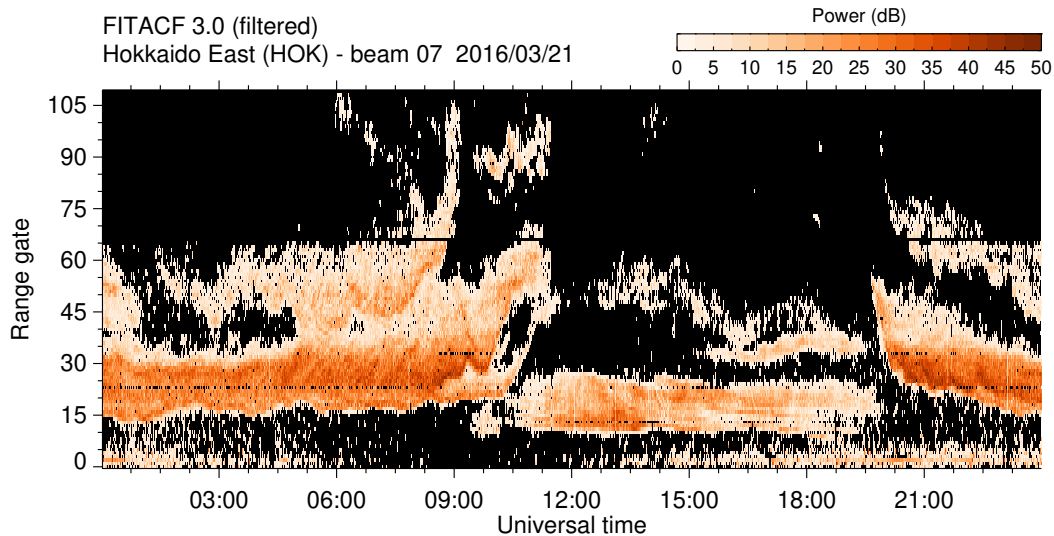


# New filtering routine in RST: `fit_speck_removal`

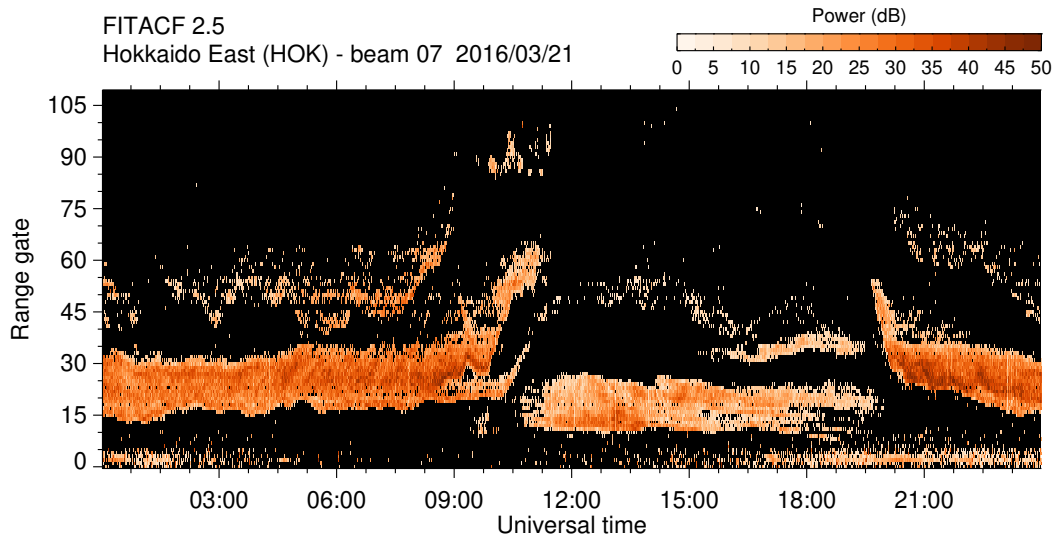




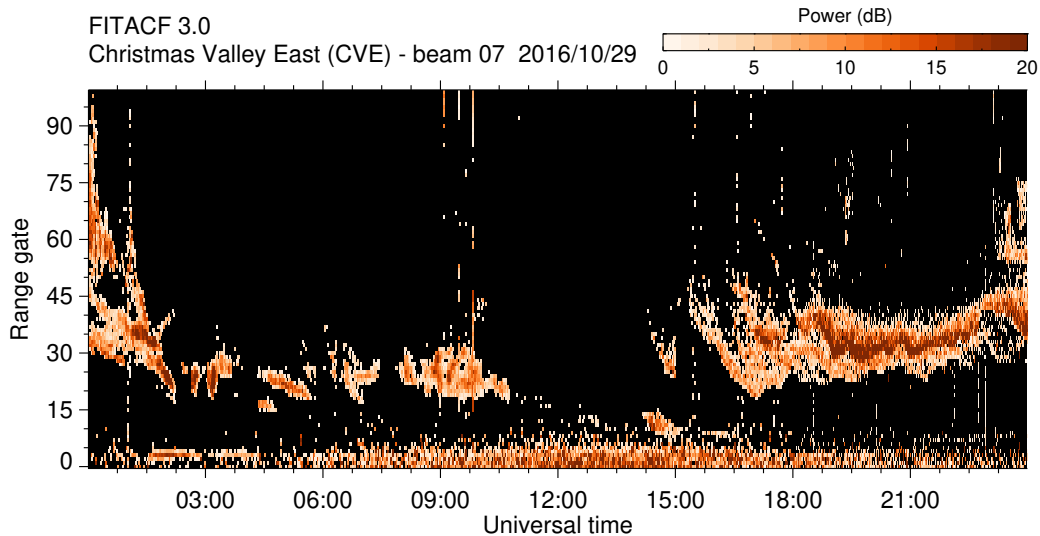
# New filtering routine in RST: `fit_speck_removal`



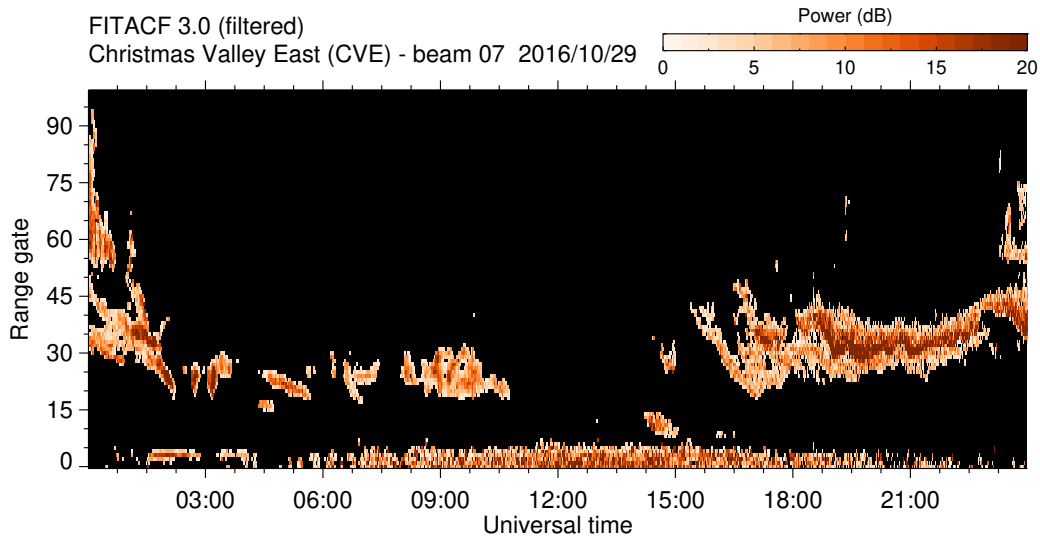
# New filtering routine in RST: `fit_speck_removal`



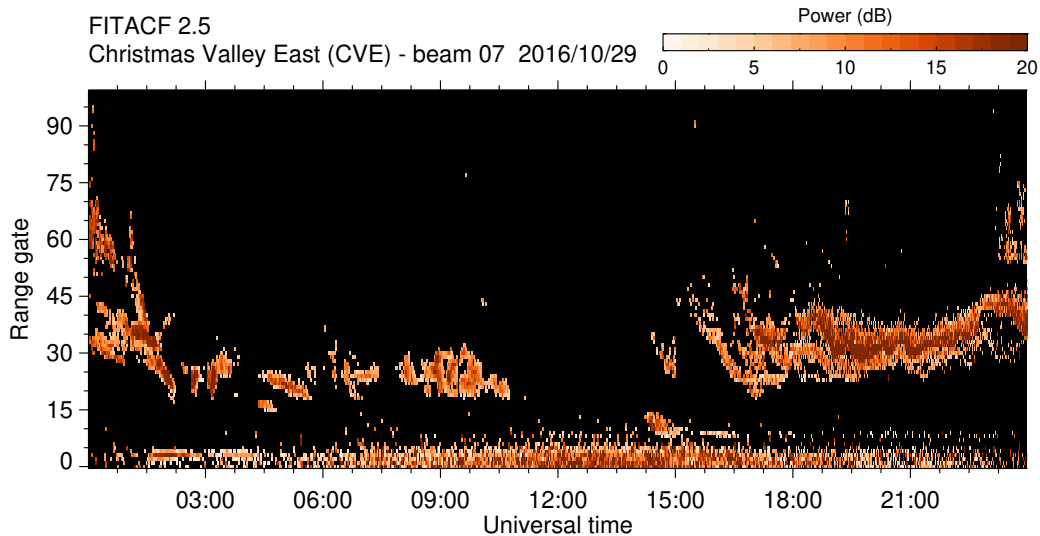
# New filtering routine in RST: `fit_speck_removal`



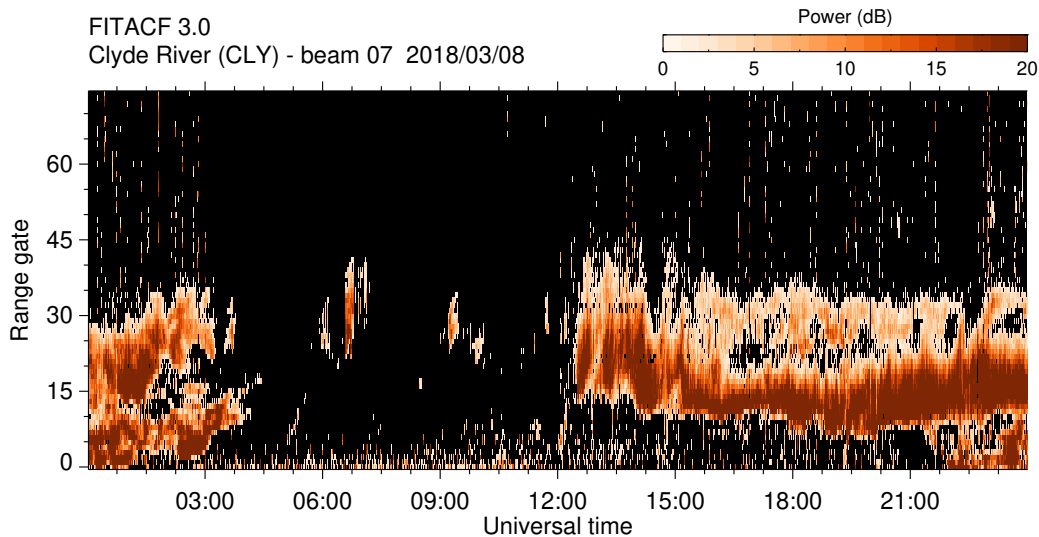
# New filtering routine in RST: `fit_speck_removal`



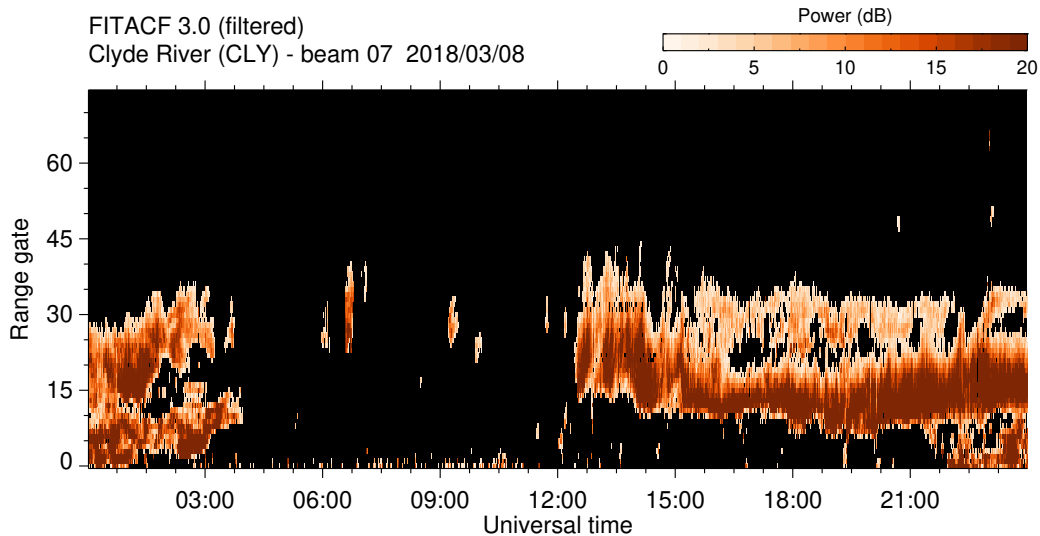
# New filtering routine in RST: `fit_speck_removal`



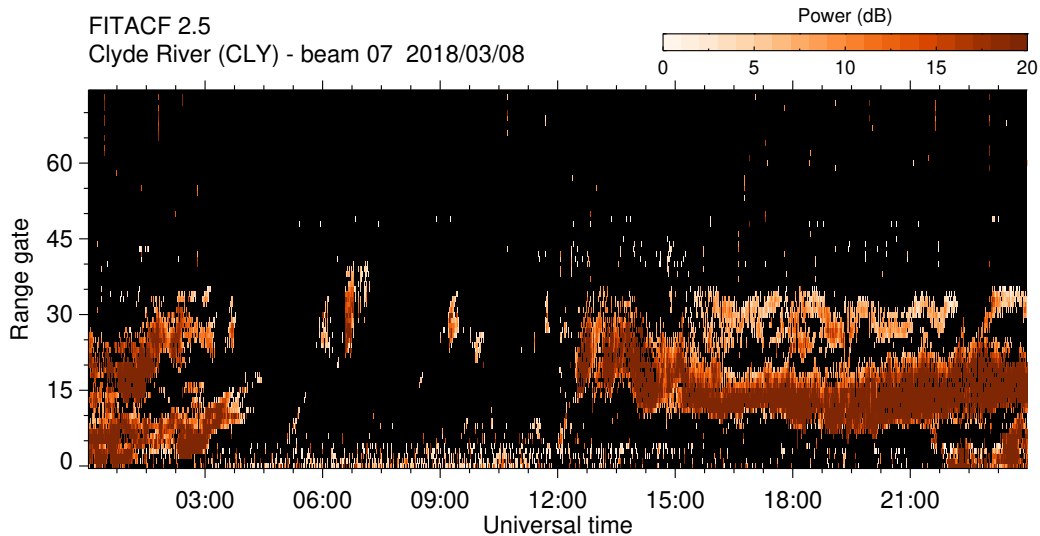
# New filtering routine in RST: `fit_speck_removal`



# New filtering routine in RST: `fit_speck_removal`



# New filtering routine in RST: `fit_speck_removal`





Data pre-selection results for FITACF 2.5 and FITACF 3.0 are strongly radar-dependent

This points to radar-related problems, rather than the fitting algorithm

- ▶ **Control program design:** number of averages (nave) & group range span (rsep\*nrang)
- ▶ **Unphysical lag zero power values** → underestimation of noise level
- ▶ **Data types problem in FITACF 2.5** → affects noise level accuracy

Proposed solution: new filtering routine in RST

- ▶ Removes echoes that are isolated in range and time
- ▶ Applying the filtering is optional

*And the user can easily determine what has been filtered*

**Questions?** Please contact Emma: [emmab@unis.no](mailto:emmab@unis.no)