

# Real Information Status

# Recap of analysis in DRBSD-22 Paper

Total amount of real information M and corresponding compression parameters necessary to save 99%

Variable	M	K for BG_K	P for ZFP_P
FLNS	0.87	2	8
ICEFRAC	6.06	N/A	N/A
LHFLX	3.09	2	8
PRECT	0.61	2	N/A
PSL	2.91	4	12
TAUX	1.33	2	8
TS	3.81	3	12
Q200	2.62	2	8
WSPDSRFAV	1.58	2	8
Z500	3.00	3	12

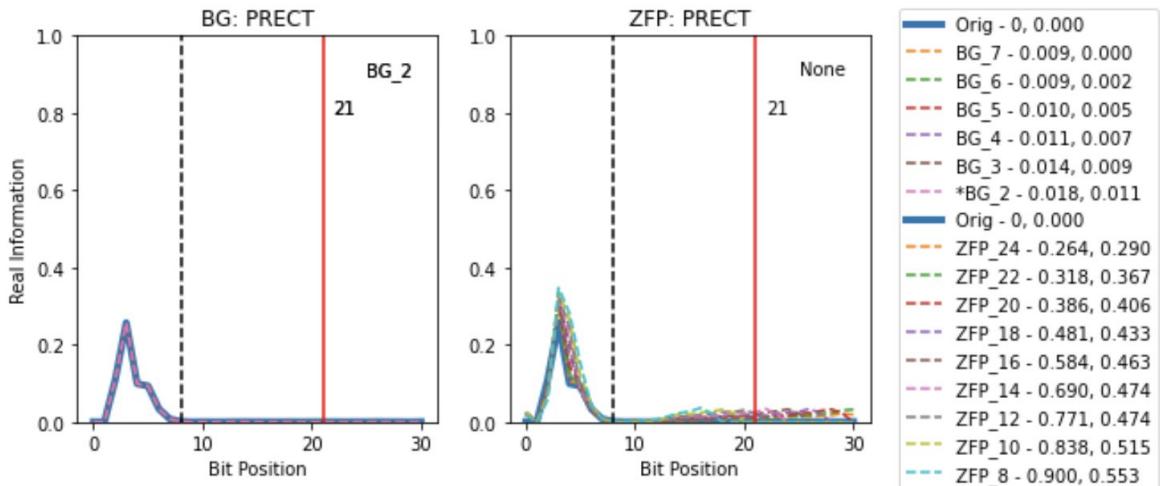
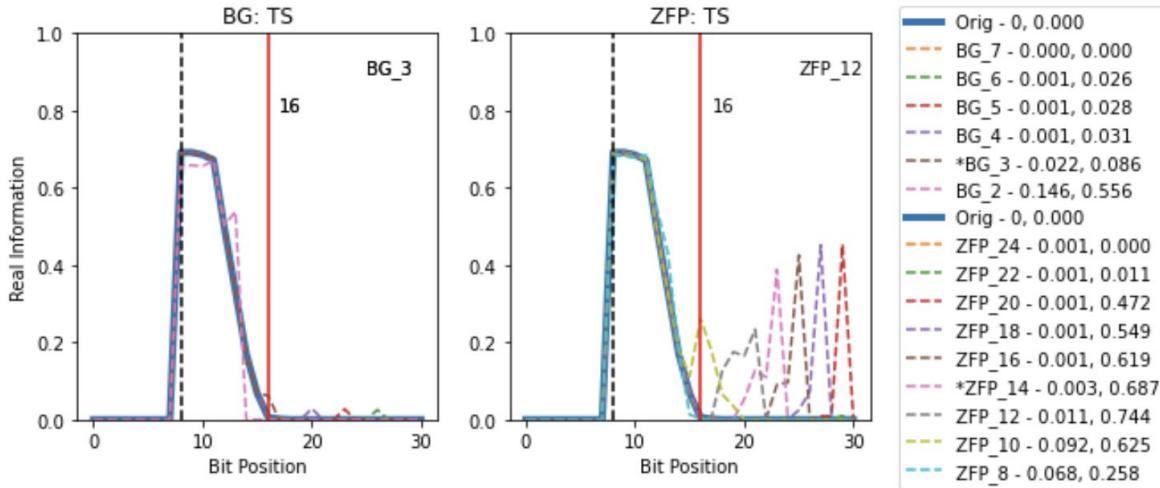
# Recap of analysis in DRBSD-22 Paper

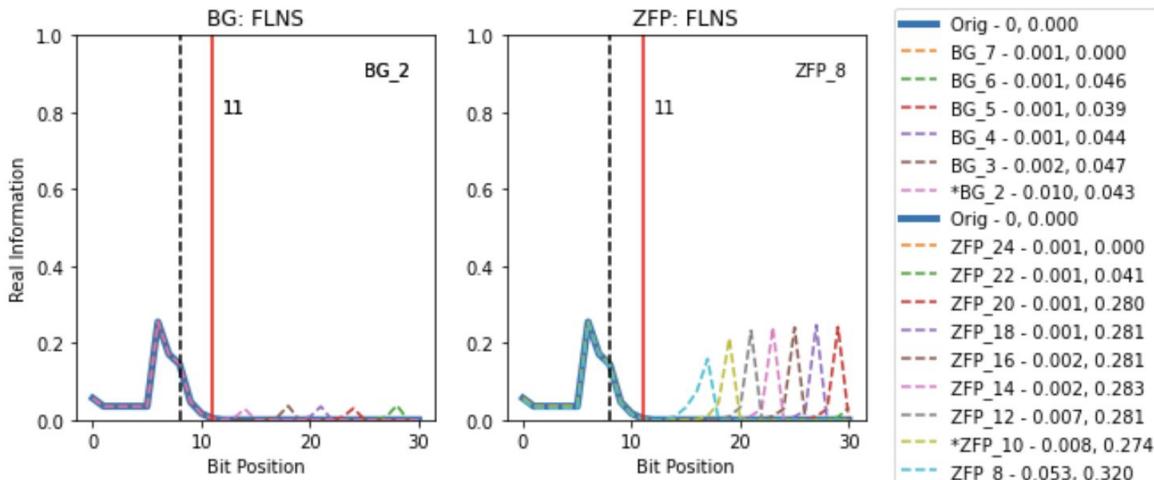
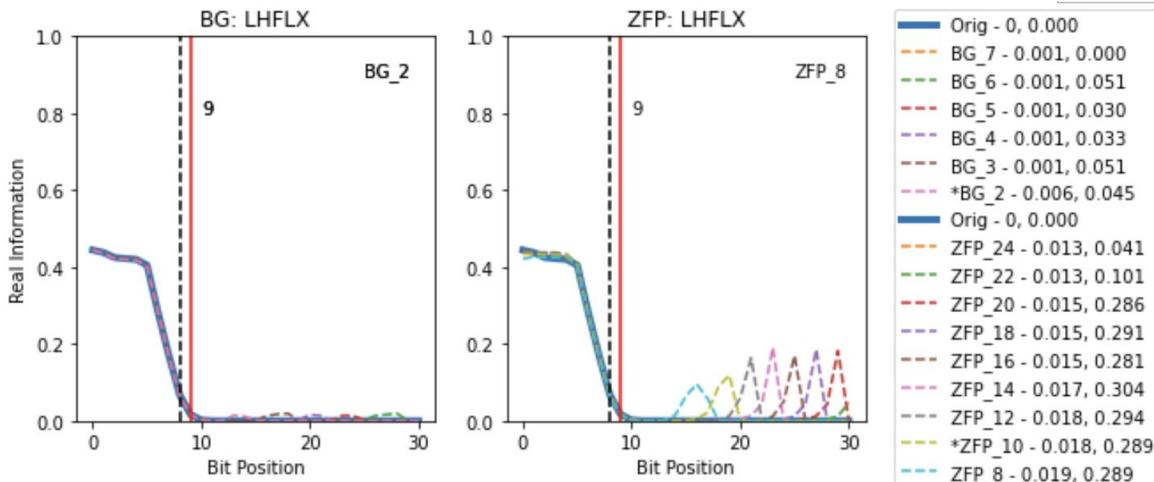
Comparison between real information and DSSIM cutoff parameters to maintain 99% of real information

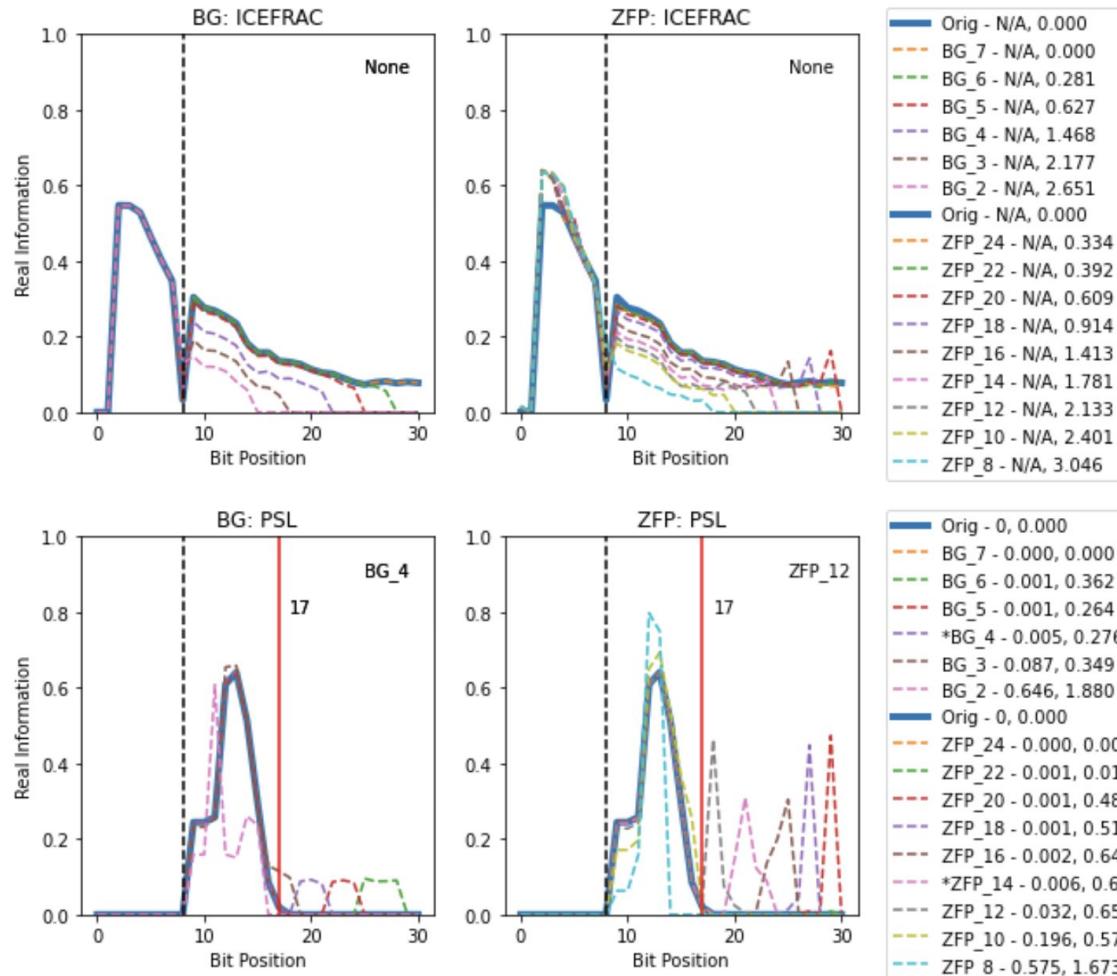
Variable Name	Real Information Content	(.95)	DSSIM (.995)	D (.9995)
FLNS	3	2	3	3
ICEFRAC	N/A	2	4	6
LHFLX	3	2	3	5
PRECT	2	2	3	5
PSL	5	5	6	7
TAUX	2	2	3	4
TS	5	4	5	7
Q200	3	2	4	5
WSPDSRFAV	3	2	3	4
Z500	5	4	5	7

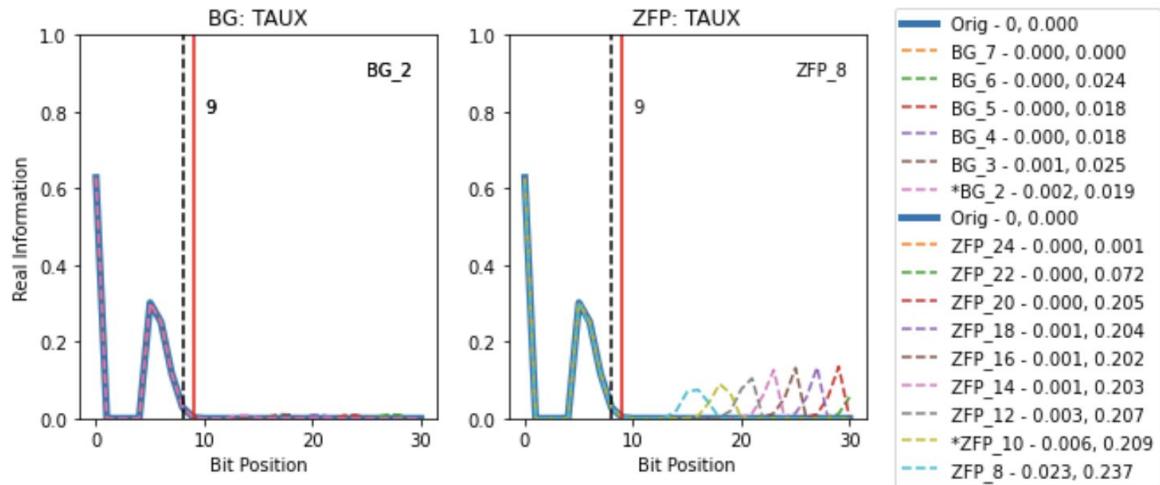
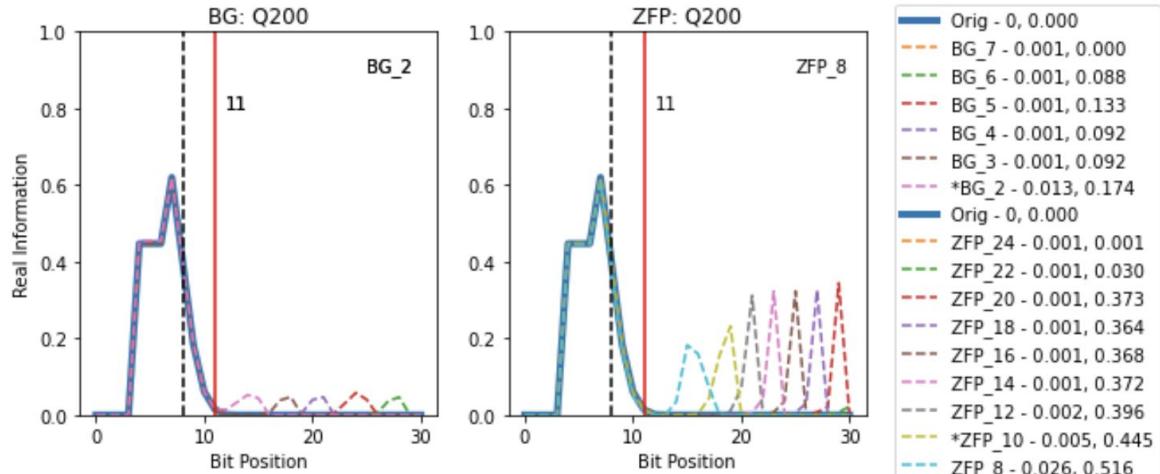
# Recap of analysis in DRBSD-22 Paper

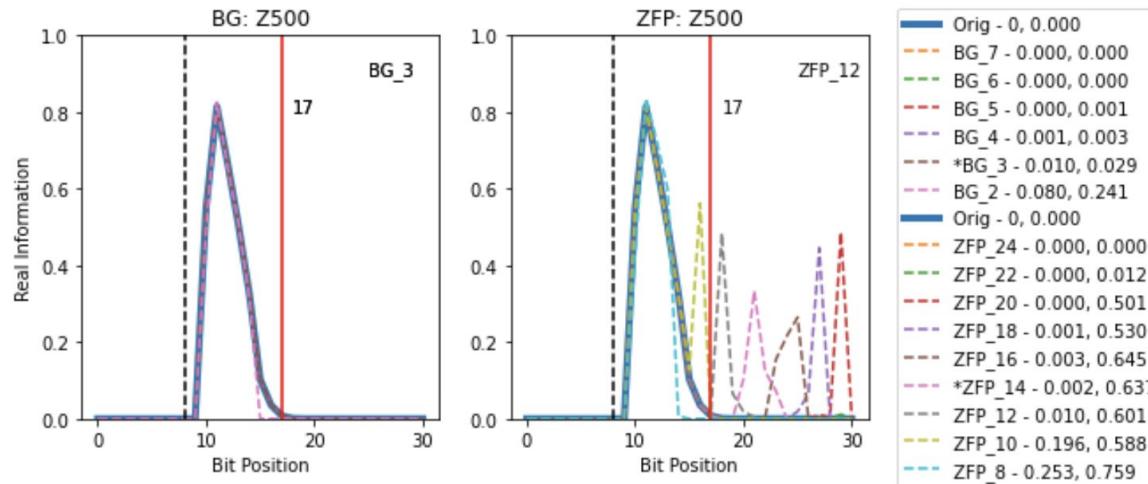
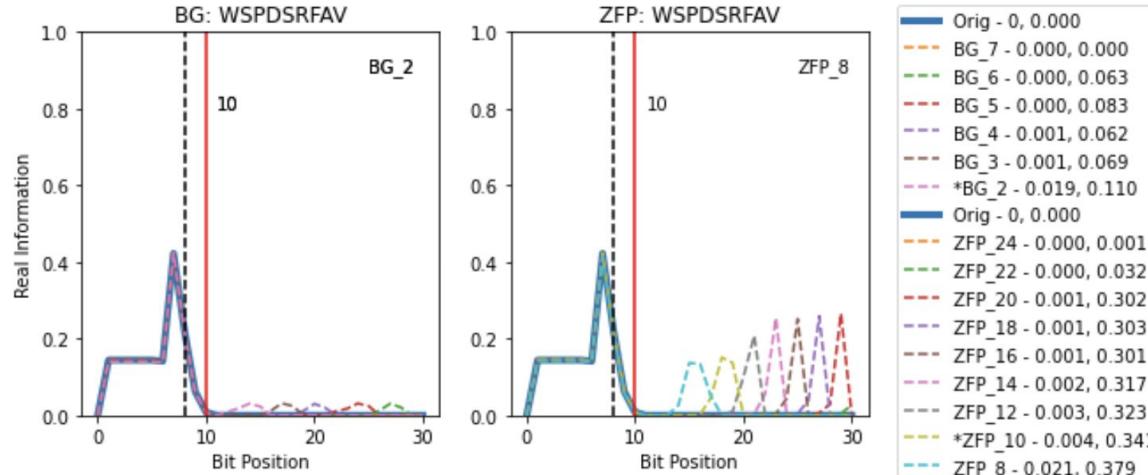
Plots of real information content for variables in original and selected compressed datasets indicating compression artifacts (spikes) and 99% information cutoff bit (vertical red line). Show over next few slides.











# Repo created for real information research files

Jupyter:

MutualInformation.ipynb

Lit:

DRBSD-22 Paper

Klower Paper

Presentations

Peter Correspondence

 pinarda	update	e16840f · 16 minutes ago	 4 Commits
 jupyter	update	16 minutes ago	
 lit	update	16 minutes ago	
 real_info_py	init	2 months ago	
 .gitignore	Initial commit	2 months ago	
 LICENSE	Initial commit	2 months ago	
 README.md	Initial commit	2 months ago	
 environment.yml	update	16 minutes ago	

 README	 GPL-3.0 license	
<hr/> <b>real_info_py</b> <hr/>		

# Existing Code Functions in Idcpy and jupyter

get\_dict\_list: Compute current and adjacent bit probabilities

get\_mutual\_info: Compute mutual info for a set of probabilities

get\_real\_info: Uses above functions to get real info array

get\_bit\_cutoff: Get cutoff to maintain given % of real info

get\_compression\_level: Find level so that, up to cutoff bit, diff between orig and compressed data real info is small

get\_real\_info\_all: Loop over get\_real\_info for multiple arrays.

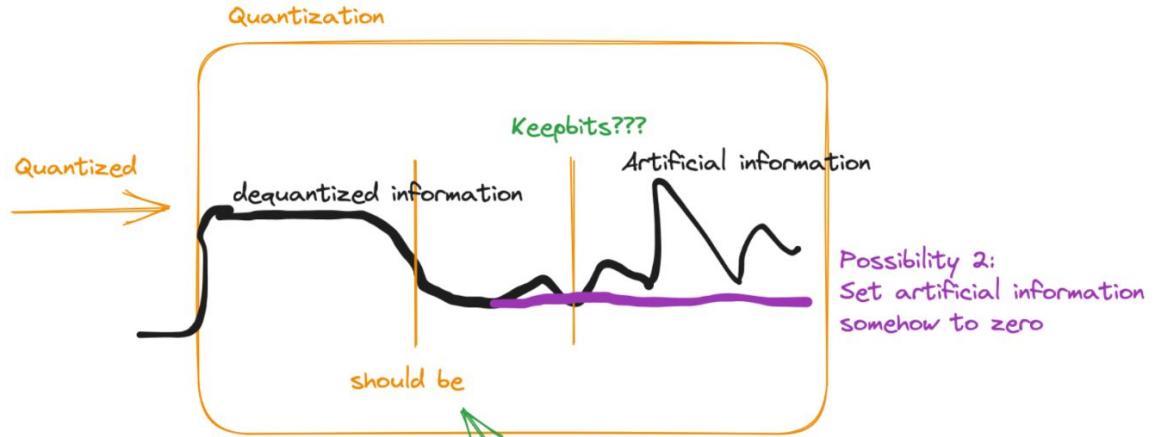
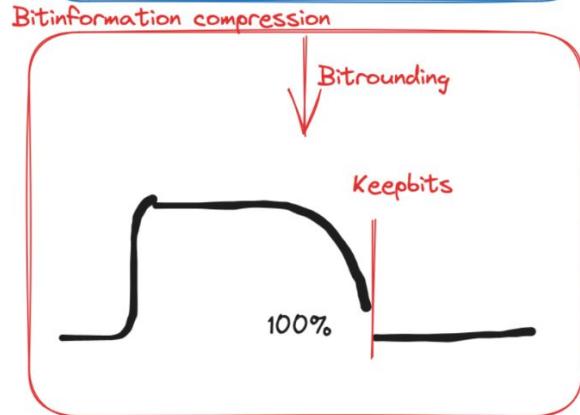
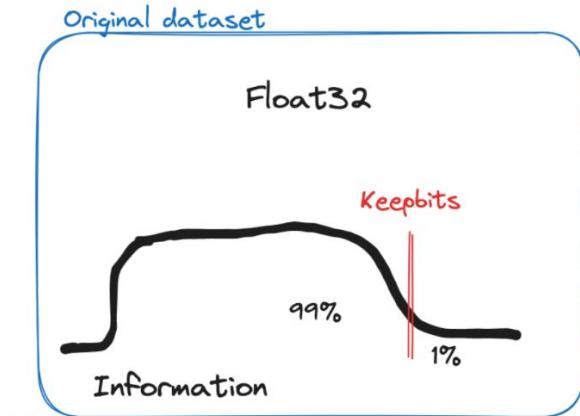
plot\_real\_info: Create single plot for real info over range of bits.

# Future directions

## What kinds of artifacts are produced by different compressors?

We have looked at the artifacts introduced by ZFP and BitGrooming, but we can also compress data using SPERR and SZ. Research what is causing these artifacts, if they behave in similar ways across compressors and error modes including the precision and absolute error mode in ZFP (of which we only example the precision mode so far), and can adjustments to the compressors be made to remove these spatial artifacts?

# Xbitinfo



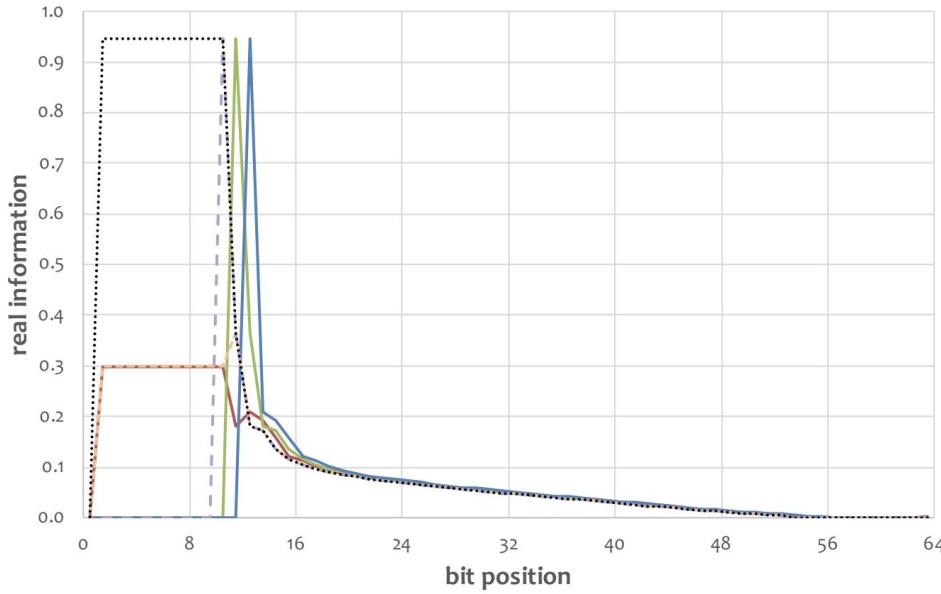
# Future directions

What are the shortcomings of the RIC, when do they apply, and how can the RIC be improved?

The same numerical values stored in different ways, such as with exponent values that differ from neighboring points, causes alterations in the computed RIC. This will affect data that spans a large range of exponent values in a small spatial area, such as PRECT (precipitation rate) data. Adjustments to the process of computing RIC can be made to account for these shortcomings. Additionally, the RIC also operates 1-dimensionally, where the dataset is flattened and neighboring bits are considered to be only the bits adjacent in the resulting bit stream. The measure could be extended into two dimensions to more accurately capture the notion of spatial structure.

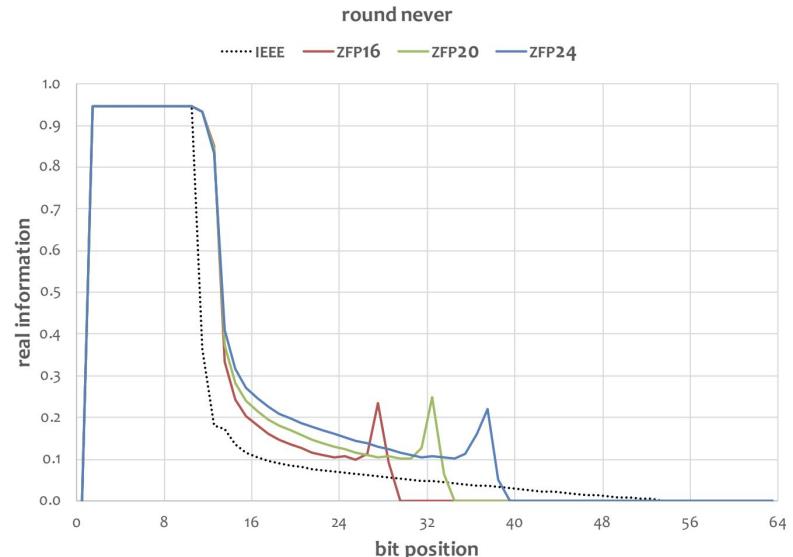
### source data transformations

— RHO + 1  
 — RHO + 2  
 — RHO + 4  
 --- RHO \* 2  
 --- RHO \* 4  
 ..... RHO



$-1 \Rightarrow 0111111110$  (23.3%)  
 $0 \Rightarrow 0111111111$  (30.1%)  
 $1 \Rightarrow 1000000000$  (46.6%)

$0 \Rightarrow 0111111111$  (23.3%)  
 $1 \Rightarrow 1000000000$  (30.1%)  
 $2 \Rightarrow 1000000001$  (46.6%)



# Future directions

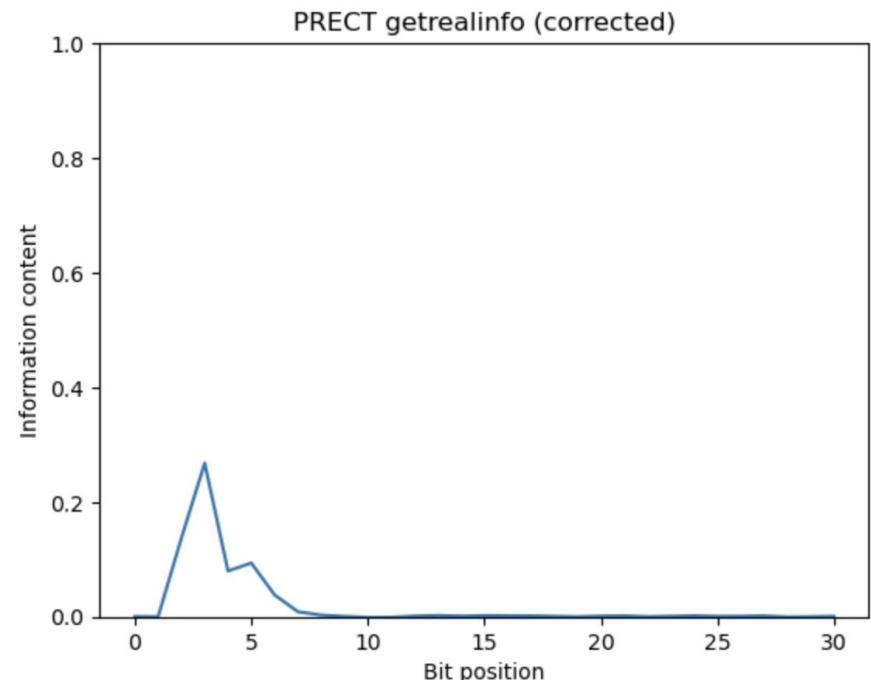
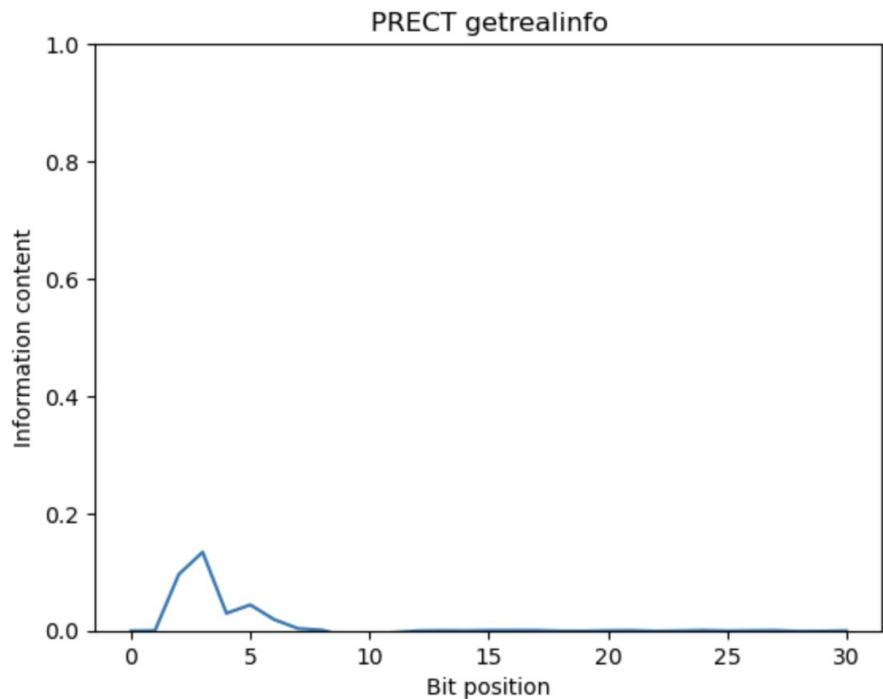
How can the cutoff bit recommended by the RIC be translated into a parameter setting such as relative error for a given compression algorithm?

We will explore the parameter settings that align with the RIC cutoff bit and the corresponding levels of compression that can be obtained. This can be done via brute-force, where multiple compression algorithms and settings are applied, and the parameters that lead to median errors near the location of the bit cutoff line are a ``match'' for that RIC cutoff.

Additionally, explore the connection between the RIC and other quantities of interest such as the DSSIM.

**END**

# Corrected real information jupyter notebook



# Corrected real information jupyter notebook

