CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

USI Compression Algorithm

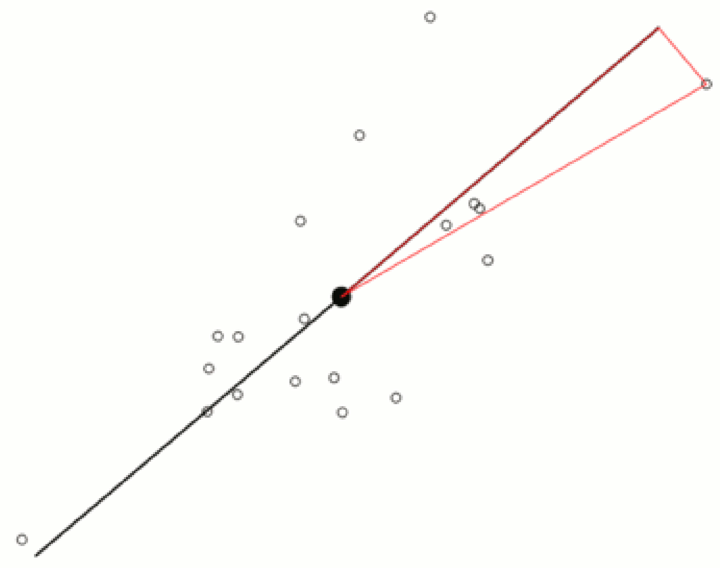
Manuel Schmid, manuel.schmid@epfl.ch

Internship, Summer 2014

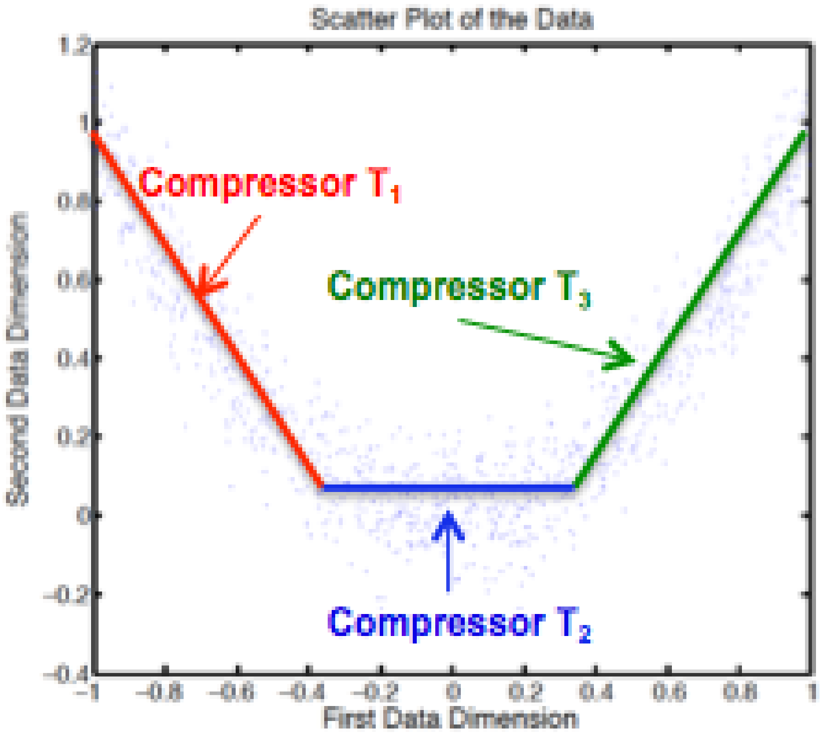
I will explain the **algorithm**, present the **implementation** and my work on it, and show some **results**.

The Algorithm

The algorithm is an **expanded form of Principal Component Analysis** (PCA).



Instead of doing a single PCA for the whole data set, we **group vectors into clusters**.

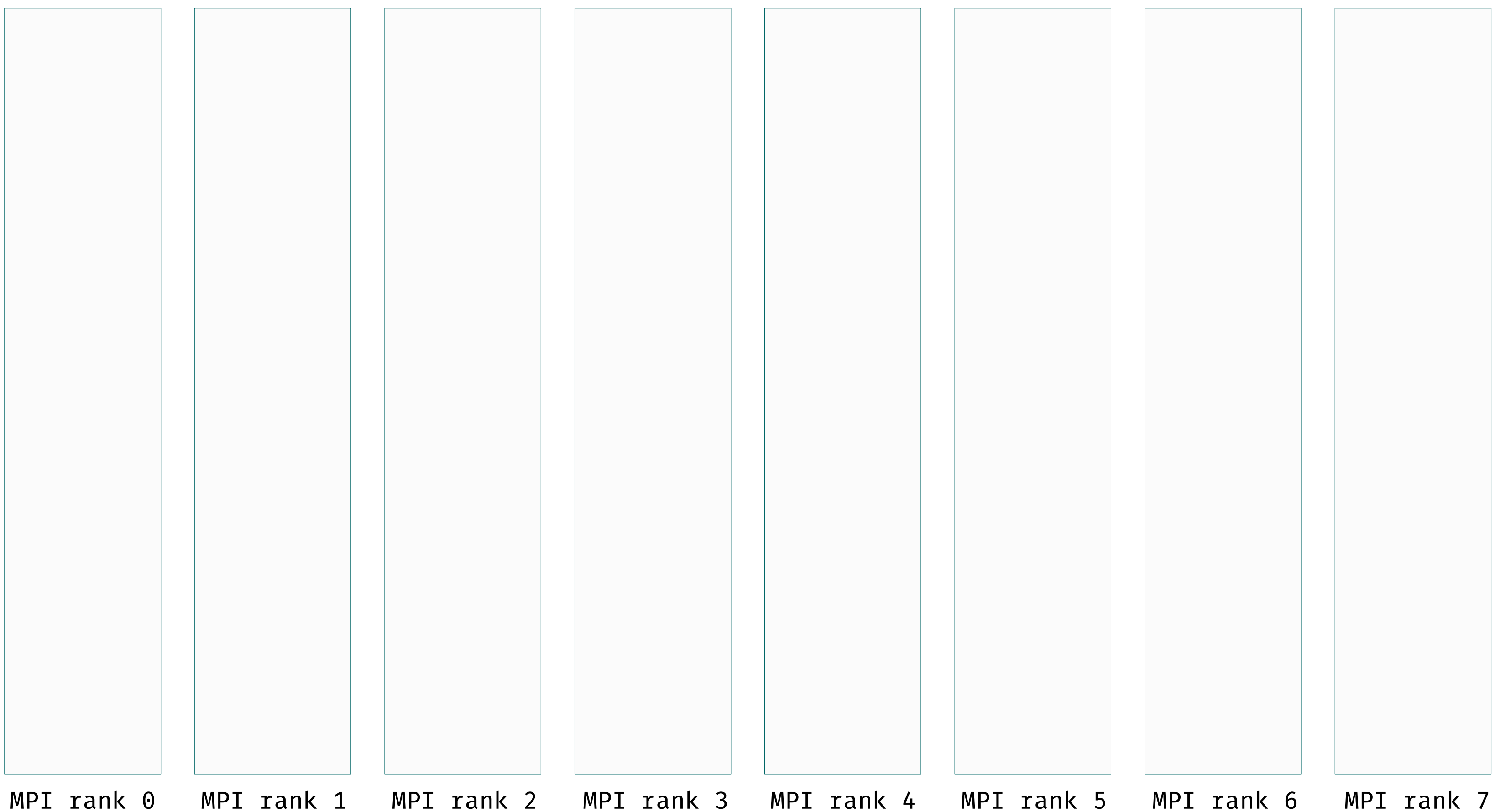


The **clustering** is done **iteratively** by doing PCA for each cluster and reassigning vectors to the best cluster.

```
→ initial clustering
→ PCA for each cluster (1 vector)
→ reassign vectors to clusters
→ PCA for each cluster (1 vector)
→ reassign vectors to clusters
  ...
→ final PCA (M vectors)
```

The Implementation

Our implementation of the USI algorithm is **distributed** and uses a **Lanczos solver** to find eigenvectors.

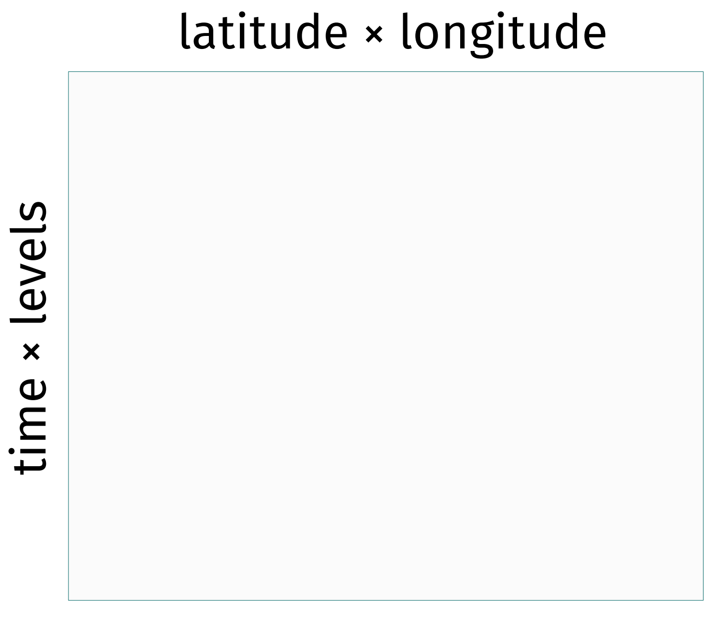


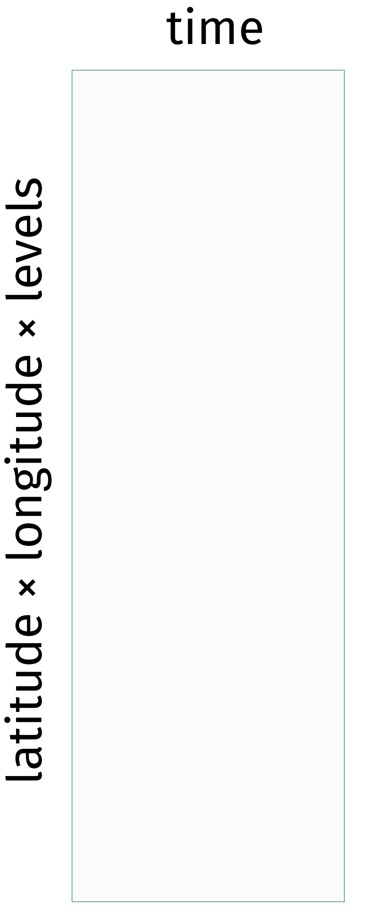
It **reads data** from a NetCDF file, **compresses** and **decompresses** the data, and **writes it back** to NetCDF.

```
// ... (parse command line arguments, set up MPI)

NetCDFInterface<Scalar> netcdf_interface(filename, variables,
        compressed_dims, indexed_dims, stacking);
DeviceMatrix<Scalar> X = netcdf_interface.construct_matrix()

std::vector<int> col_ids = netcdf_interface.get_column_ids();
CompressedMatrix<Scalar> X_compressed(X, K_size, M_size, col_ids);

DeviceMatrix<Scalar> X_reconstructed = X_compressed.reconstruct();

netcdf_interface.write_matrix(X_reconstructed, filename_out, append);

// ... (clean up)
```

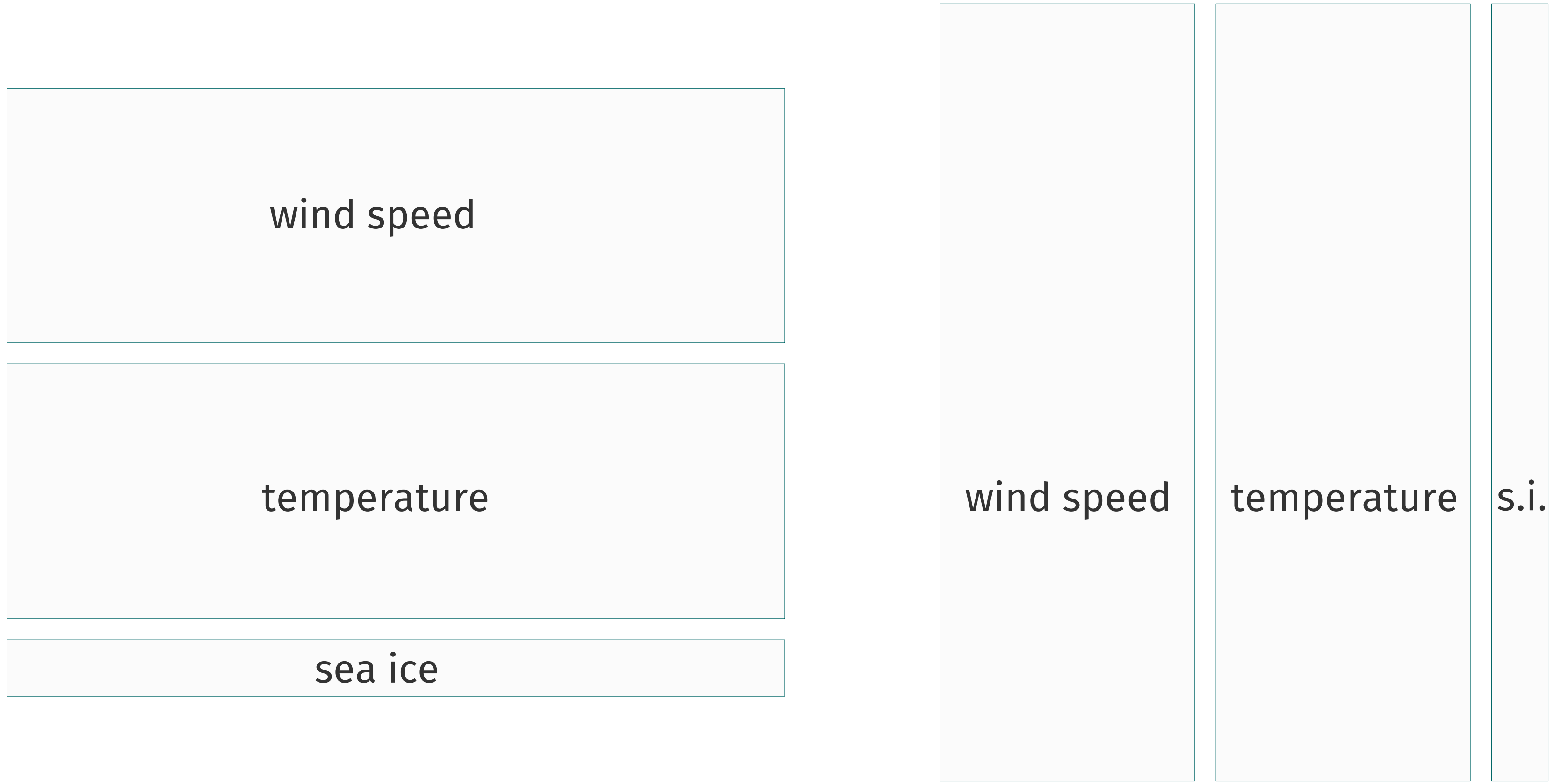
There are **three different versions** of the program:
eigen, minlin_host, and minlin_device

We can specify **arbitrary dimensions along rows/columns**.





We can read and combine **multiple variables**.



The program can print out **statistics** comparing the original and reconstructed data.

```
--------------------------------------------------------------------------------
Statistics
--------------------------------------------------------------------------------

Maximum time for input: 2.75187
Maximum time for solve: 962.085

Compression ratio: 0.0999069

Variable CCN3:

          min           max           mean           std
     3.66e-05       1251.36        26.0719       54.3484   (original data)
     -1.72935       1237.73        26.0719       54.3374   (reconstructed data)

  maximum error:  0.0372324 (normalized with range)
  RMS error:      0.000872772 (normalized with range)
  correlation:    0.999798
  SRR:            5.63699
  PrecisionBits:  3.7473

--------------------------------------------------------------------------------
```

The Results

The results are based on **yearly average** data from the **Community Earth System Model** (CESM).

1 horizontal dimension (ncol) with 48602 entries

1 vertical dimension (lev or ilev) with 30 or 31 entries

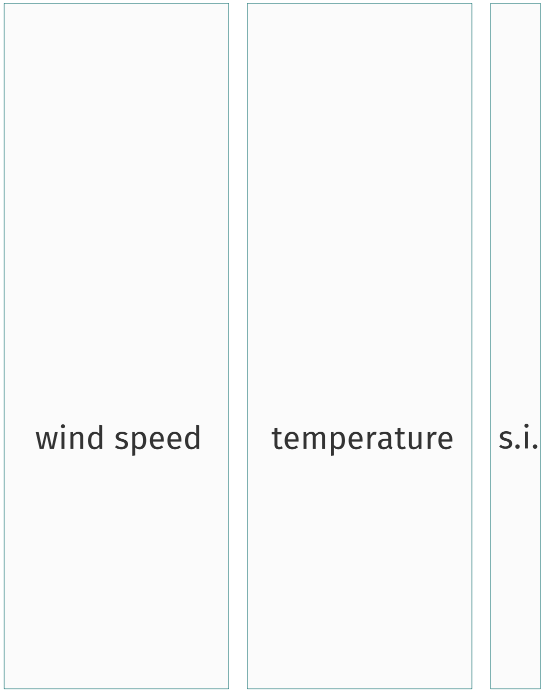
88x 3D variable with 30 levels

09x 3D variable with 31 levels

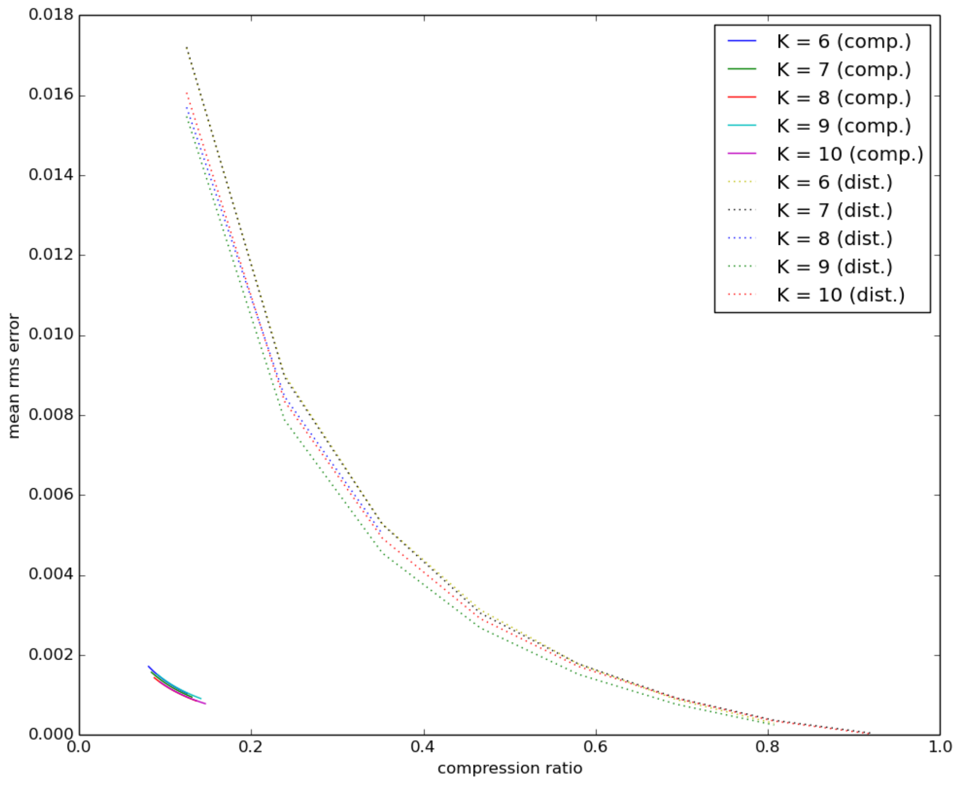
101x 2D variable (89 used)

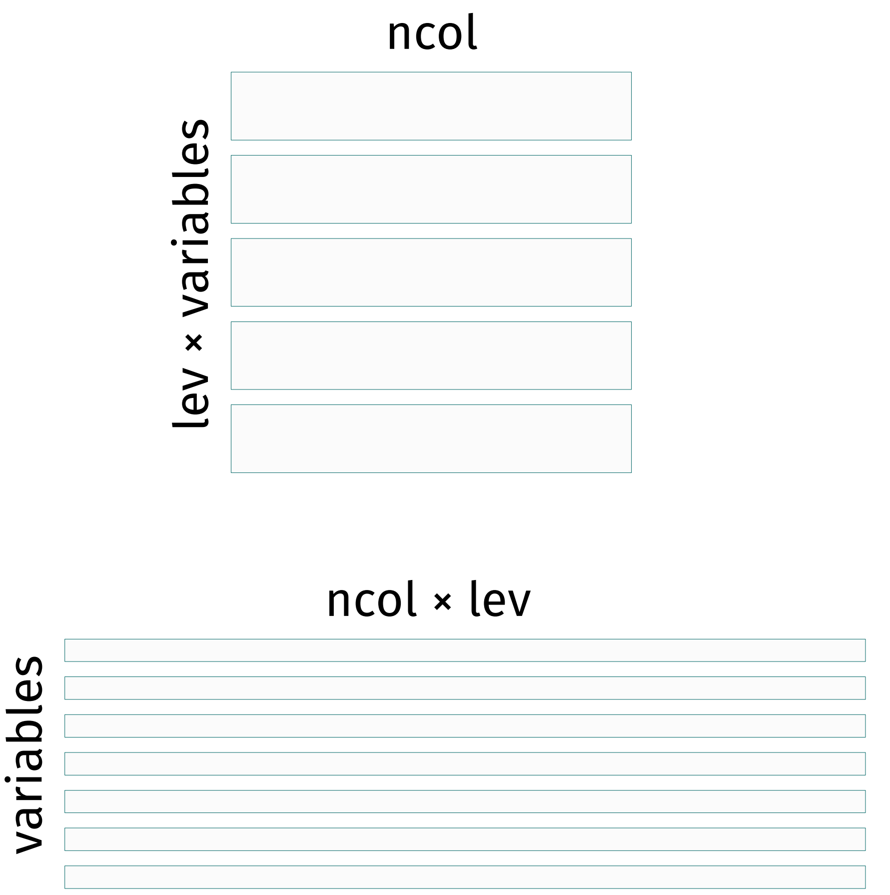
Horizontal stacking didn’t work very well.

rank	cl. 1	cl. 2	cl. 3	cl. 4	cl. 5	cl. 6	cl. 7	cl. 8	cl. 9	cl. 10
0	**259**	**11**	**11**	**1**	**6**	**0**	**0**	**0**	**1**	**14**
1	**271**	**9**	**5**	**0**	**1**	**1**	**1**	**2**	**0**	**12**
2	**267**	**4**	**0**	**9**	**0**	**1**	**4**	**7**	**2**	**8**
3	**255**	**5**	**0**	**11**	**6**	**0**	**3**	**9**	**7**	**6**
4	**235**	**8**	**2**	**6**	**14**	**1**	**3**	**9**	**17**	**7**
5	**220**	**12**	**12**	**5**	**21**	**1**	**7**	**8**	**10**	**6**
6	**208**	**15**	**9**	**1**	**26**	**0**	**6**	**6**	**15**	**16**
7	**412**	**71**	**23**	**0**	**111**	**56**	**99**	**27**	**28**	**66**

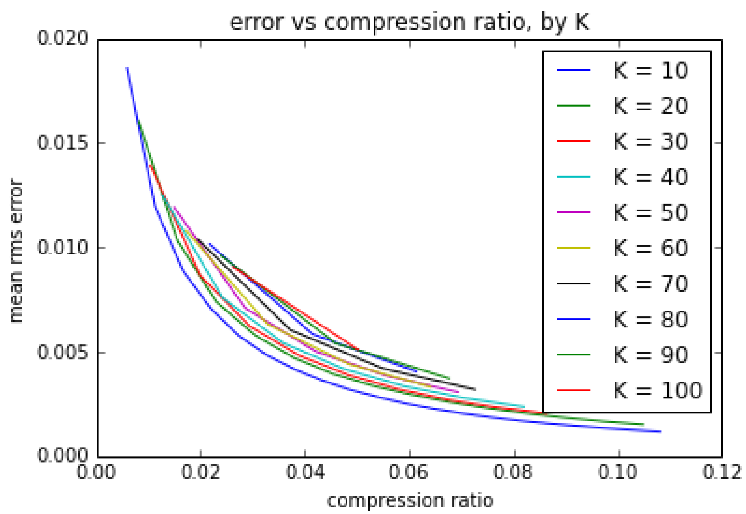


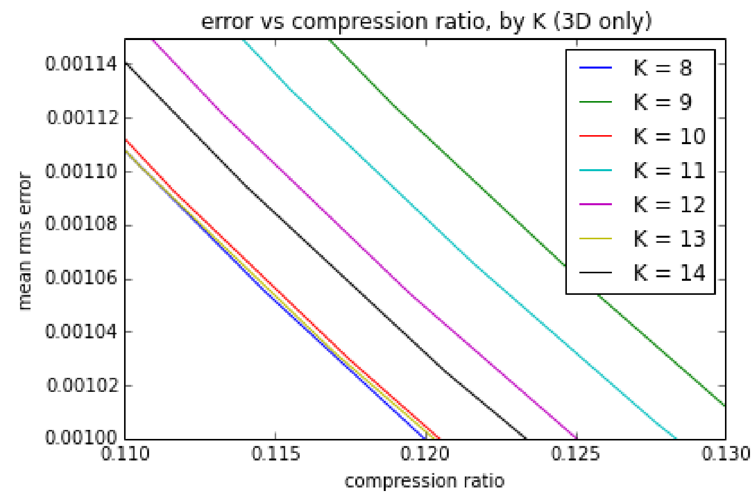
Compression is better if we only place ncol along rows rather than ncol × lev.

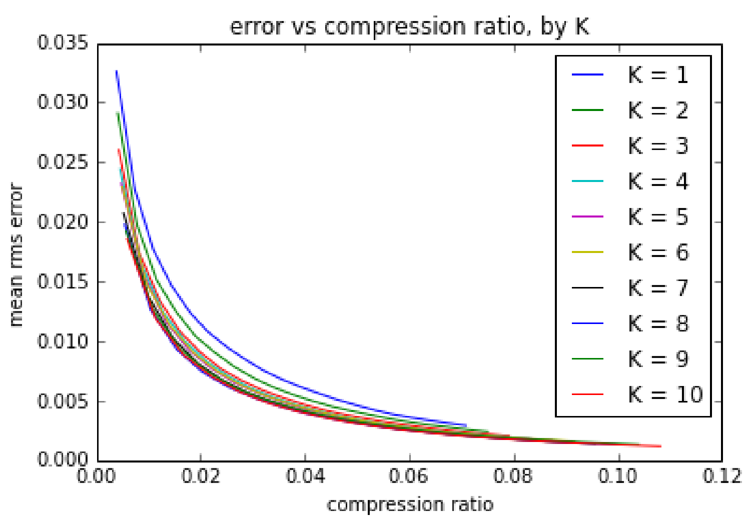


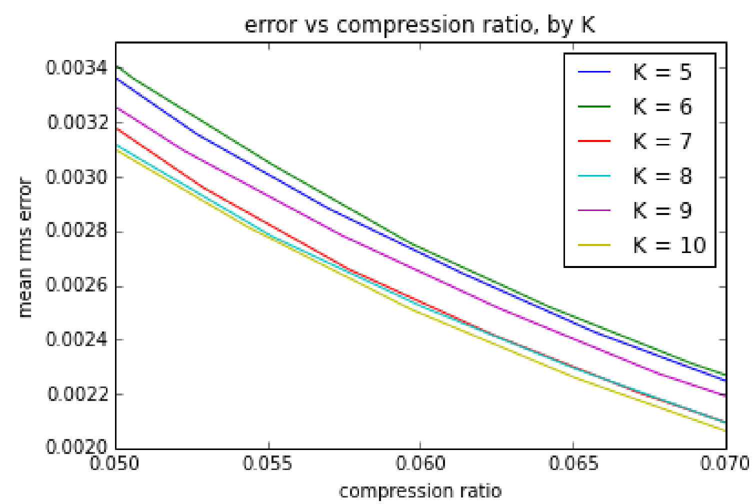


There is an **optimal number of clusters** at around K=10.

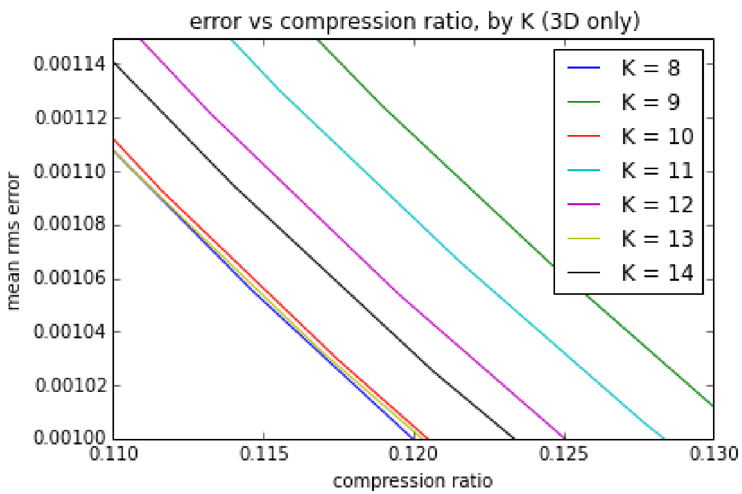


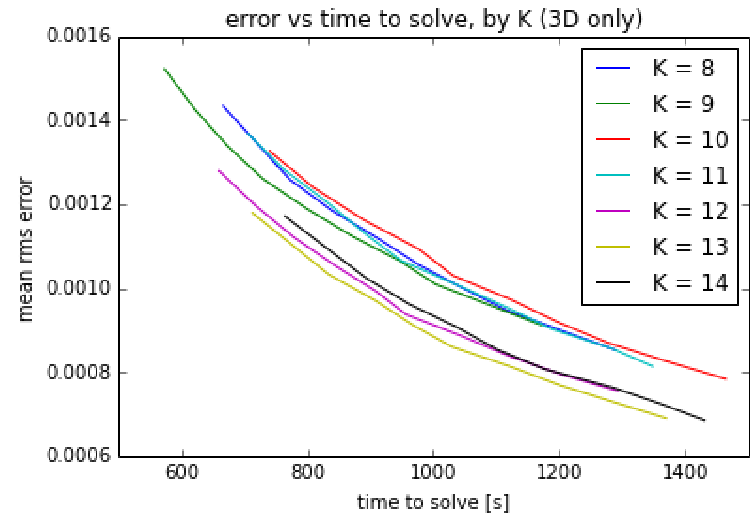


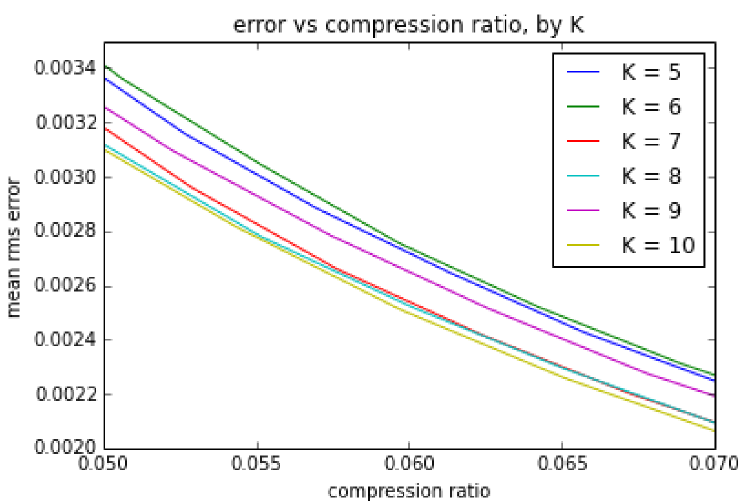


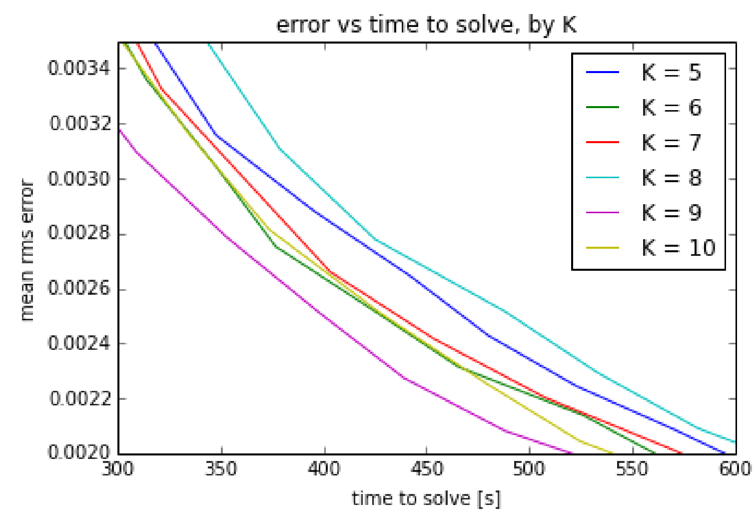


There is an **optimal number of clusters** at around K=10.

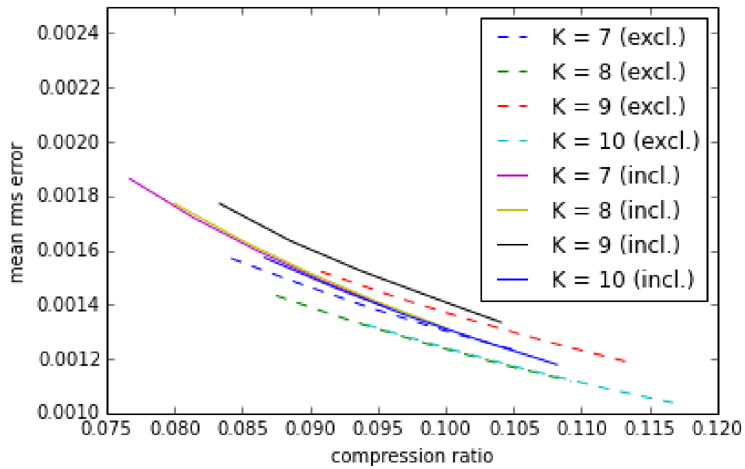


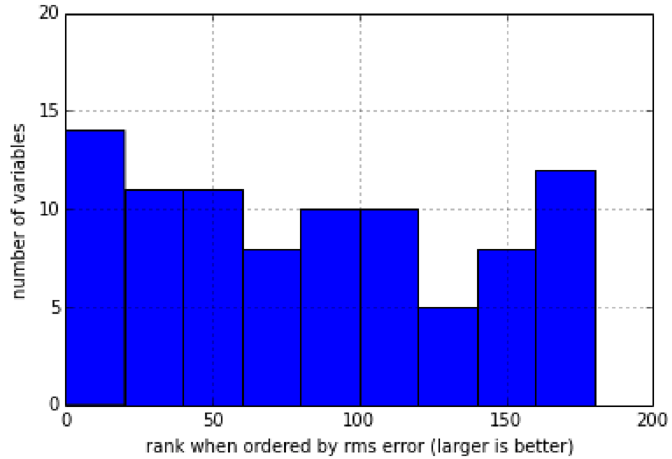


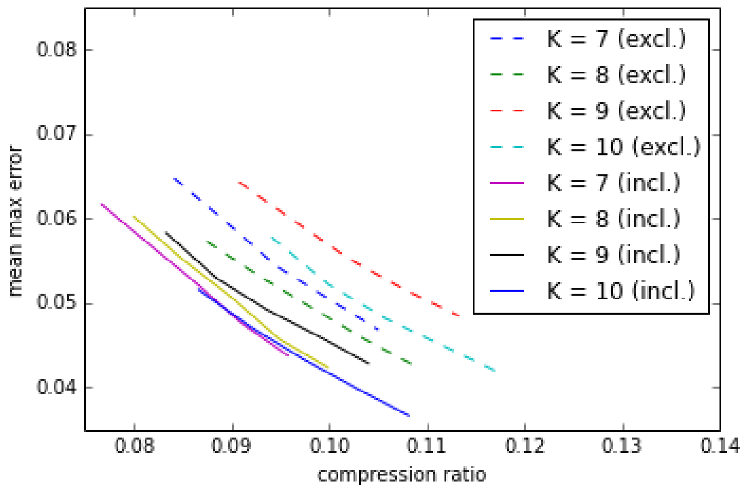


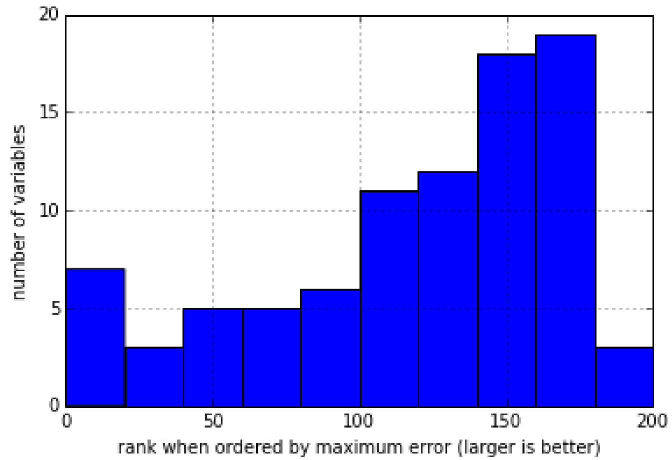


Excluding 2D variables doesn't improve compression considerably.

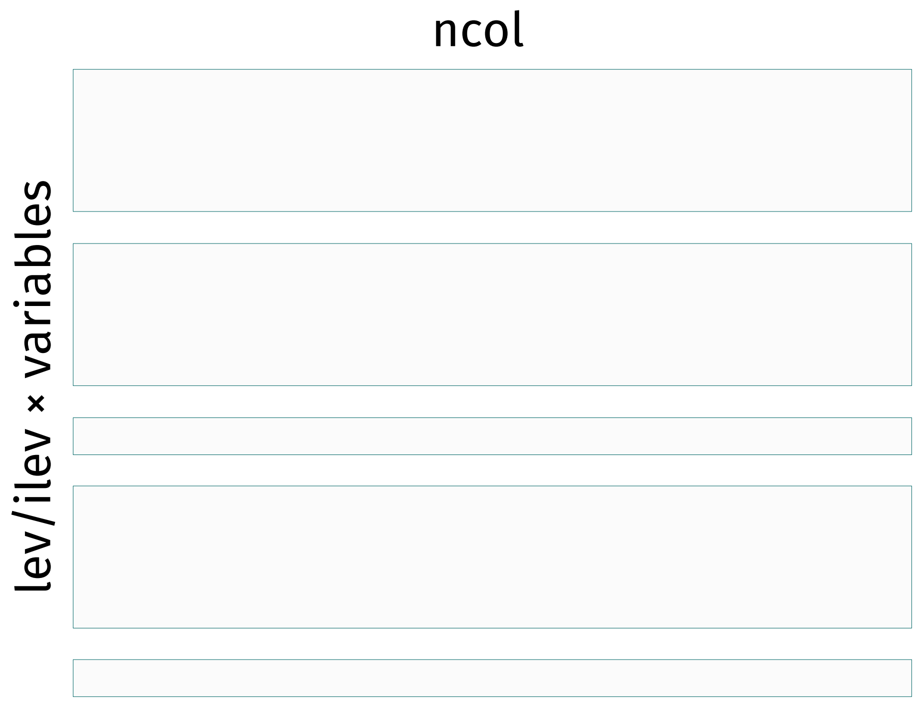








We therefore compress all variables together, using 10 clusters, placing only ‘ncol’ along rows.



We can make **comparisons between variables.**

for K=10, M=200

RMS error	max. error	prec. bits	SSR	variable name
0.000093	0.00183	8.10	11.5	Geopotential Height (above sea level)
0.000187	0.000419	6.90	10.2	Liquid water static energy
0.000187	0.00415	6.91	10.2	Liq wat virtual static energy
0.000347	0.0518	3.27	5.00	Aerosol absorption
0.000376	0.0457	3.45	3.62	Zonal momentum flux
...	...	...	...	...
0.00217	0.0514	3.28	6.26	Fractional occurance of snow
0.00226	0.0227	4.46	6.60	Fractional occurance of ZM convection
0.00230	0.0243	4.36	6.30	Sea level pressure
0.00249	0.0304	4.04	6.08	Vertically-integrated mid-level cloud
0.00260	0.0627	3.00	6.29	Mobilization flux at surface

We can also try to compare the results to **other publications**, but currently this is **difficult**.

Evaluating Lossy Compression on Climate Data

Nathanael Hübbe[1], Al Wegener[2], Julian Kunkel[1], Yi Ling[2], and
Thomas Ludwig[3]

A Methodology for Evaluating the Impact of Data Compression on Climate Simulation Data *

Allison H. Baker
National Center for Atmospheric Research
1850 Table Mesa Drive
Boulder, CO 80305
abaker@ucar.edu

Haiying Xu
National Center for Atmospheric Research
1850 Table Mesa Drive
Boulder, CO 80305
haiyingx@ucar.edu

John M. Dennis
National Center for Atmospheric Research
1850 Table Mesa Drive
Boulder, CO 80305
dennis@ucar.edu

Michael N. Levy
National Center for Atmospheric Research
1850 Table Mesa Drive
Boulder, CO 80305
mlevy@ucar.edu

Doug Nychka
National Center for Atmospheric Research
1850 Table Mesa Drive
Boulder, CO 80305
nychka@ucar.edu

Sheri A. Mickelson
National Center for Atmospheric Research
1850 Table Mesa Drive
Boulder, CO 80305
mickelso@ucar.edu

Compared to Hübbe et al., our **errors are much larger** but our **compression is also much stronger**.

file	rel. size	SRR		PrecisionBits	
		APAX	GRIB2/JPEG 2000	APAX	GRIB2/JPEG 2000
trads	65.4%	20.7	20.8	20.3	21.1
aclcov	64%	23.2	22.1	21.0	21.0
trafl	56.5%	20.8	21.4	20.4	22.0
trflwac	53.2%	21.5	20.9	20.2	21.8
soflwac	35.8%	21.0	22.0	18.7	22.0
wsmx	29.3%	23.7	21.6	21.8	21.8
ahfllac	28%	21.7	19.0	21.1	21.7
vdisgw	22.9%	24.6	19.6	22.9	21.9
srad0d	22.6%	13.5	21.6	12.2	21.3
alsom	2.9%/1.9%	lossless	22.8	lossless	21.5

Table 1. GRIB2/JPEG2000 and APAX Signal Quality Metrics at the same compression ratios. N=22 for GRIB2/JPEG2000 files.

Compared to Baker et al., we only have a **few values** that have similar compression ratios, but those have **similar errors**.

Table 3: NRMS errors (and compression ratio CR) between the original and reconstructed datasets for variables U, FSDSC, Z3, and CCN3.

Comp.	U	FSDSC	Z3	CCN3
Method		NRMSE (CR)		
GRIB2	3.6e-4 (.10)	1.4e-4 (.22)	7.8e-8 (.32)	2.3e-8 (.37)
APAX-2	5.8e-7 (.50)	8.3e-7 (.50)	7.0e-8 (.50)	1.6e-7 (.50)
APAX-4	1.4e-4 (.25)	2.1e-4 (.26)	2.0e-5 (.25)	4.1e-5 (.25)
APAX-5	4.3e-4 (.20)	5.4e-4 (.21)	5.1e-5 (.19)	9.9e-5 (.20)
fpzip-24	2.2e-6 (.39)	1.8e-5 (.34)	5.1e-6 (.19)	6.5e-7 (.36)
fpzip-16	5.7e-4 (.15)	4.6e-3 (.10)	1.2e-3 (.04)	1.7e-4 (.12)
ISA-0.1	8.7e-5 (.57)	4.1e-4 (.37)	3.8e-5 (.39)	2.8e-5 (.37)
ISA-0.5	2.7e-4 (.44)	9.1e-4 (.36)	9.8e-5 (.37)	1.2e-4 (.38)
ISA-1.0	3.7e-4 (.41)	1.1e-3 (.36)	1.5e-4 (.36)	2.0e-4 (.37)

ours (0.10) 1.2e−3 1.0e−3 9.3e−4 8.1e−4

Table 4: Maximum relative pointwise errors (e_{nmax}) (and compression ratio) between the original and reconstructed datasets for variables U, FSDSC, Z3, and CCN3.

Comp.	U	FSDSC	Z3	CCN3
Method		e_{nmax} (CR)		
GRIB2	6.2e-4 (.10)	2.5e-4 (.22)	1.6e-7 (.32)	4.9e-8 (.37)
APAX-2	3.3e-6 (.50)	4.7e-6 (.50)	3.3e-6 (.50)	2.9e-6 (.50)
APAX-4	9.0e-4 (.25)	1.1e-3 (.26)	8.3e-4 (.25)	7.5e-4 (.25)
APAX-5	2.7e-3 (.20)	2.7e-3 (.21)	3.1e-3 (.19)	1.9e-3 (.20)
fpzip-24	1.2e-5 (.39)	3.9e-5 (.34)	3.3e-6 (.19)	2.4e-5 (.36)
fpzip-16	3.1e-3 (.15)	9.9e-3 (.10)	6.8e-3 (.04)	5.3e-3 (.12)
ISA-0.1	6.4e-4 (.57)	1.6e-3 (.37)	9.8e-4 (.39)	8.7e-4 (.37)
ISA-0.5	2.9e-3 (.44)	7.6e-3 (.36)	4.9e-3 (.37)	3.9e-3 (.38)
ISA-1.0	4.9e-3 (.41)	1.5e-2 (.36)	9.9e-3 (.36)	7.9e-3 (.37)

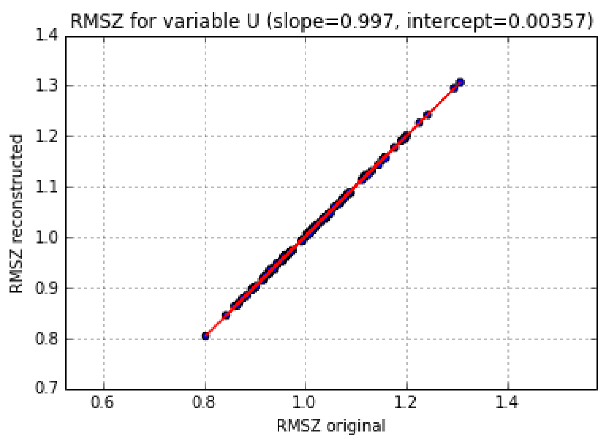
1.8e−2 1.2e−2 1.8e−3 4.3e−3

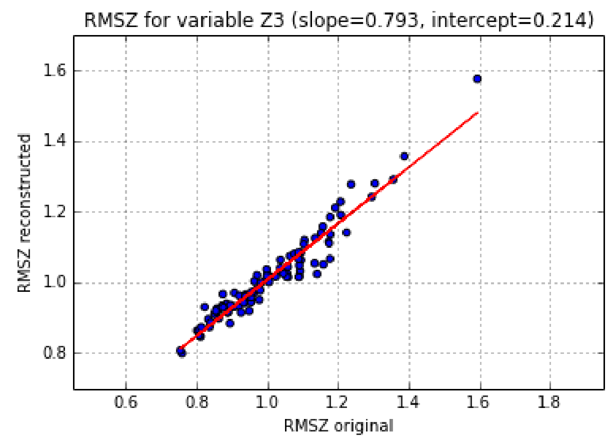
Baker et al. propose a **method to evaluate lossy compression** based on an **ensemble** of 101 files

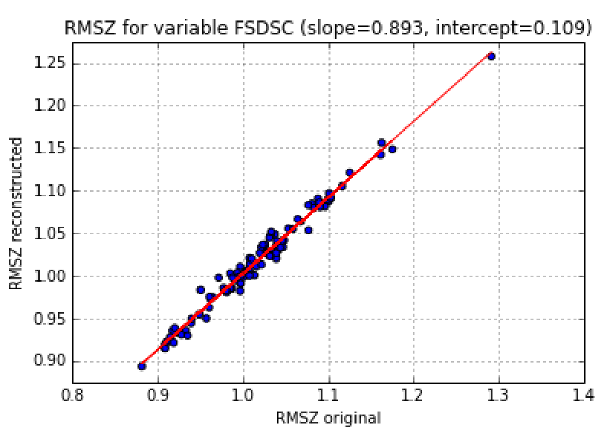
$$Z_{x_i}^m = \frac{x_i^m - \bar{x}_i^{E\setminus m}}{\sigma_{x_i}^{E\setminus m}}$$

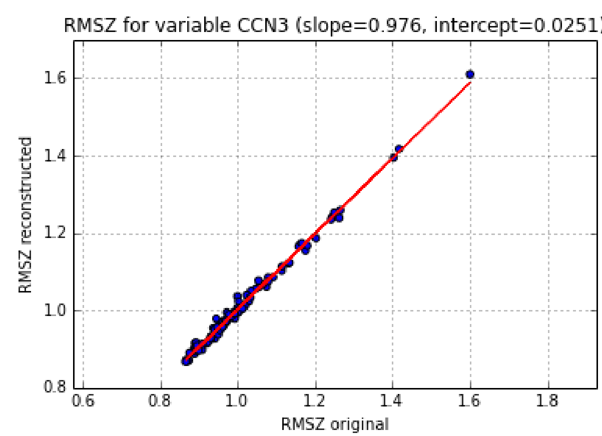
$$RMSZ_X^m = \sqrt{\frac{1}{N_X}\sum_i (Z_{x_i}^m)^2}$$

We can use this to test for a **bias** of the compression method.










```
slope
U      0.99 – 1.0001
Z3     0.75 – 0.84
FSDSC  0.87 – 0.92
CCN3   0.96 – 0.99

intercept
U      0.00035 – 0.0067
Z3     0.17    – 0.26
FSDSC  0.083   – 0.14
CCN3   0.0081  – 0.042

max. Diff.
U        0.006
Z3       0.118
FSDSC    0.034
CCN3     0.035
```

Next, we should **implement lossless compression**, **fix the GPU version** and do **runs with weaker compression**.

Thank you.