Concept and Analysis of Information Spaces to improve Prediction-Based Compression

Ugur Cayoglu[†*], Frank Tristram[*], Jörg Meyer[*], Tobias Kerzenmacher[†], Peter Braesicke[†] and Achim Streit[*]
[*]Steinbuch Centre for Computing
Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
Email: {Ugur.Cayoglu, Frank.Tristram, Joerg.Meyer2, Achim.Streit}@kit.edu
[†]Institute of Meteorology and Climate Research - Atmospheric Trace Gases and Remote Sensing
Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
Email: {Peter.Braesicke, Tobias.Kerzenmacher}@kit.edu

***Abstract*—One of the scientific communities that generate the largest amounts of data today are the climate sciences. New climate models enable model integration at unprecedented resolution, simulating decades and centuries of climate change, including many complex interactions in the Earth system, under different scenarios. Previously, the CPU intensive numerical integration's used to be the bottleneck. Nowadays, limited storage space and ever increasing model output is the bigger challenge. The number of variables stored for post-processing analysis has to be limited to keep the data amounts small. For this reason, we look at lossless compression of climate data to make better use of available storage space. More specifically, we investigate prediction-based data compression. In prediction-based compression, data is processed in a predefined sequence. A prediction is provided for each data point based on prior data in the sequence. We show that there is a significant dependence of the compression ratio on the chosen traversal method and the underlying spatio-temporal data model. We examine the influence of this structural dependency on compression algorithms and explore possibilities to retrieve this information to improve compression ratios. To do this, we introduce the concept of Information Spaces (IS), which helps improve the predictions made by individual predictors by nearly 10% on average. More importantly, the standard deviation of the compression results is decreased by over 20% on average. The use of IS provides better predictions and more consistent compression ratios. Furthermore, it allows options for consolidation and fine-granular tuning of predictions, which are not possible with many common approaches used today.**

***Index Terms*—compression, information retrieval, information spaces, climate, data model, prediction-based compression**

I. Introduction

Climate sciences are in a state of upheaval. New climate models such as ICON-ART [1] make it possible to run high-resolution simulations of the atmosphere and its composition at an unprecedented scale while making full use of the available capacity of high-performance computers. But with these improvements, the storage space required to save the output of the simulations also increases. The current **E**uropean **ReA**nalysis (ERA5) dataset, which is being used for evaluation and initialisation of simulations, is comprised of hourly data starting from 1979 to the present on a 1440×721 (about 31 km) horizontal and 137 level vertical (up to 0.01 hPa = 80 km) grid[1]. If we assume 16 bit integer values for each variable this amounts to 2.26 TiB p.a. per variable with support for 120 variables[2].

In such situations an efficient compression method can help reduce the required storage space. Although studies suggest that scientific results obtained from lossily compressed data are not distinguishable from lossless compressed ones [2], for most scientists only lossless compression is an option. For this reason we focus our work on lossless compression.

Lossless compression works in two steps: decorrelation and encoding. The decorrelation step transforms the data by removing correlation with itself (auto-correlation) or with others (cross-correlation). Since decorrelated data is easier to compress the result overall will be a better compressed file. The encoding step writes the data as a compressed stream on disk. Prediction-based compression for climate data reduces autocorrelation by using data points along the temporal dimension and cross-correlation by using data points on the spatial dimensions. This work contributes to the decision making which neighbouring data points should be used for the decorrelation of the dataset.

While there are several factors important for evaluating a good compression algorithm like compression ratio and throughput, our method concentrates on compression ratio. The main reason for this is the increase in storage requirements in climate research mentioned above.

There are four main contributions of this paper: (1) we analyse how the choice of a starting point and traversal sequence can influence the compression ratio, (2) we introduce new traversal sequences which help stabilise the compression ratio, (3) we define the notion of Information Spaces (IS) and show how they can help gain robust compression ratios, and (4) we introduce consolidation techniques to further improve compression ratios.

In the next section we describe related work in prediction-based compression and make clear how our work differs. Section III provides preliminaries to prediction-based compression

[1] European Centre for Medium-Range Weather Forecasts (ECMWF) Newsletter No. 147 – Spring 2016 (p.7)

[2] While some of these variables are simulated, others can be deduced from simulated variables. For reference http://apps.ecmwf.int/codes/grib/param-db

and defines necessary notations used in this paper. In Section IV we introduce IS and describe our proposed method. We further describe new traversal sequences complementing IS and consolidation methods used to improve the predictions. Section V then moves on to describe our test dataset, metrics used for evaluation and explain the conducted experiments. In Section VI we evaluate our results. Finally we conclude with a summary and outlook.

II. Related Work

Prediction-based compression algorithms have long been used in image [3], [4] , audio [5], [6] and floating-point data [7]–[10] compression. Recently, these methods have been used more frequently for the compression of structured climate data [11]–[13].

The work so far has concentrated on the predictor component of the prediction-based compression algorithm [14], [15]. It is irrevocable that the predictor plays a major role in the success or failure of the process, but other important questions such as the available traversal options and its relationship to the structure of the data has either been only scratched on the surface [12] or completely neglected.

Many algorithms rely on linear traversal of the data [7], [16]. Other traversal sequences so far have only been considered in image compression, but not in relationship with climate data. Huang et. al [12] look at the first couple of data points along each dimension to decide how to traverse the data, but do not adjust later in the processing chain.

Furthermore, investigating the effect of the selection of a starting point on the compression ratio is missing in previous work. The quality of the prediction might be affected by the chosen point and its surrounding data points and changing it could improve the overall compression ratio.

Our work is a key step to solve these problems. Different starting points are tested and their effects on the compression ratio are evaluated. The internal structure of the data and its influence on the compression are quantified.

III. Preliminaries

In this section we will give a brief introduction to prediction-based compression and explain the terms and notation used throughout the paper.

A. Prediction-based compression

A prediction-based compression algorithm involves following steps:

1) Reading in the floating-point values
2) Mapping the data to integer values
3) Defining a traversal sequence
4) Giving a prediction for each value
5) Calculating the difference between prediction and true value
6) Encoding these residuals and saving on disk

The output of each step is depicted in Figure 1. We improve steps three and four of this process. Section IV-A introduces the proposed method, which will help improve the predictions in step four. Complementary traversal sequences are presented in Section IV-C, which can be used by arbitrary predictors.

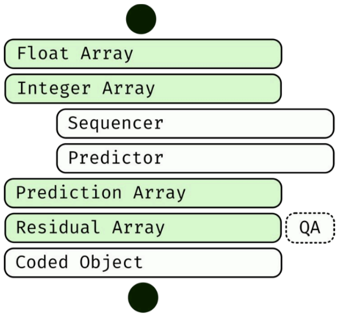


Fig. 1: Output of each step of a prediction-based compression algorithm. Our work improves the Sequencer and Predictor involved (white and right aligned). The dotted QA depicts the step in which the quality of the predictors will be assessed. The last encoding step is not analysed further.

Before we examine the proposed method in more detail, we define the basic concepts and introduce our notation.

B. Definitions and Notations

The most common data used in environmental sciences are structured data cubes with four dimensions representing longitude, latitude, altitude and time. Each data point can can be identified by its coordinate c.

a) Coordinate: A coordinate c is a d-tuple which defines the position of a data point.

$$c = (a_1, \ldots, a_{d-1}, a_d) \tag{1}$$

with $a_j \in \mathbb{N}_0$ and $\forall j \in \{1, 2, \ldots, \mathrm{d}-1, \mathrm{d}\}$.

For the application of prediction-based compression the following components of the algorithm must be determined in advance: Mapping function, starting point, traversal and prediction method, residual calculation method and encoding method.

b) Mapping function: The mapping function m defines a method for mapping floating-point values to unsigned integers. This step is necessary for reproducible calculations across CPU architectures due to possible errors in numerical precision if calculations were to be done via floating-point operations.

$$m : \mathbb{R} \mapsto \mathbb{N}_0 \tag{2}$$

c) Starting point: The starting point s_0 is the coordinate which defines the position from which the traversal will start.

$$s_0 = (a_1^0, \ldots, a_{d-1}^0, a_d^0) \tag{3}$$

d) Traversal method: Given a starting point s_0 the traversal method defines the order of data points to be predicted. The result is a sequence S of coordinate positions to be traversed.

$$S = \{s_i | \forall i \in \mathbb{N}_0 \text{ and } 0 \leq i < \prod_{j=1}^{d} D_j\} \tag{4}$$

with D_j representing the size of dimension j.

e) Prediction method: The prediction method p gives a prediction $\hat{v}_i$ for value v_i at coordinate position s_i of sequence S using the set S_i of known elements.

$$\begin{aligned} p(S_i) &= \hat{v}_i \text{ with} \\ S_i &= \{s_j | s_j \in S \text{ and } \forall j < i\} \end{aligned} \tag{5}$$

The predictor p can either use all values v_i at position s_i with $s_i \in S_i$ to calculate $\hat{v}_i$ or only a subset of values.

f) Residual calculation: The residual calculation method defines the method to be used for calculating the difference between the prediction and true value. This difference is called the residual. In most cases (e.g. [12], [14], [17]) this will be the XOR on bit level between the prediction $\hat{v}_i$ and true value v_i.

$$r(\hat{v}_i, v_i) = \hat{v}_i \text{ XOR } v_i \tag{6}$$

The better the prediction, the more zeros are at the beginning of the residual. With a lossless compression method, the number of zeros must be encoded and the remaining binary values must be stored verbatim. The original binary value can then be reconstructed using the prediction method and residual.

g) Encoding method: The encoding method will define the coding of the residuals on disk. In prediction-based compression there is a clear distinction between the prediction and encoding phase. The goal of the prediction phase is to use the available information in the best possible way so that the residual is close to 0. The task of the encoding method is to write these residuals on disk in a space-efficient way. There are several options to choose from: Huffman Coding [18], Arithmetic Coding [19], Range Coding [20], Golomb Coding [21], Asymmetric Numeral System Coding [22] or any combination of them. The effects of the encoding method on the compression ratio are outside the scope of this article.

After introducing the necessary terms and notations, we will now present our proposed method.

IV. Proposed Method

In this section we will define our concept of IS and their components the Information Contexts (IC). Afterwards we will present methods for merging predictions from different Information Contexts and consolidate the final prediction. Finally, we introduce three new traversal methods before ending the section with a description of the individual predictors.

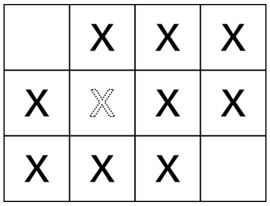
A. Information Spaces and Contexts

Our proposed method calculates position and neighbourhood information of each point s_i during the traversal to improve prediction results. While the term Information Space is mainly associated with Max Boisot [23], we define the term Information Space in the rest of this paper as follows:

Definition IV-A.1. *The Information Space of a data point s_i is the set of data points $s_k \in S_i$ with $k < i$ and a Chebyshev distance [24] of r with s_i.*

$$IS(s_i) = \{s_k | \forall s_k \in S_i : a^i_j - r \leq a^k_j \leq a^i_j + r\} \tag{7}$$

with a^n_m defining the coordinate position at dimension m of element n of sequence S.



	X	X	X
X	X	X	X
X	X	X	

Fig. 2: Information Space for example given in IV-A0a. The value to be predicted is depicted as a dotted X and values known at the time of prediction are marked with a filled X.

The restriction r is necessary to constrain locally close information for the prediction of s_i. This Information Space is a first selection of d-dimensional data that can be used to predict s_i. The Information Space is now divided into its components to isolate the information contained in the various dimensions. These components are called Information Context which divide the existing information into d levels (one per dimension).

Definition IV-A.2. *The Information Context splits the Information Space into different subsets based on their information level for each dimension and if applicable to each combination of dimensions.*

$$\begin{aligned} IC^p_l(s_i) &= \{s_k : \sum_{j=0}^{d} z_j(s_i, s_k) = l\} \\ z_j(s_i, s_k) &= \begin{cases} 1 & \textit{if } a^i_j = a^k_j \\ 0 & \textit{else} \end{cases} \end{aligned} \tag{8}$$

with $0 \leq p \leq \binom{d}{l}$ being the index position of IC_l at level l.

Each Information Context contains information along one or more dimensions. All Information Contexts on one level can contain overlapping data points, but none is a subset of the others. This distribution of data allows predictions to be made on the basis of information from different dimensions and later merge them into a consolidated prediction.

a) Example: Given a grid of size 3×4 and following sequence: $S = \{(0,1), (0,2), (0,3), (1,0), (1,2), (1,3), (2,0), (2,1), (2,2), (1,1), (0,0), (2,3)\}$. For the prediction of $s_{10} = (1,1)$ the resulting Information Space is depicted in Figure 2. This Information Space consists of five different Information Contexts depicted in Figure 3. These Information Contexts can then be used to improve the prediction of the value at s_{10} by using different methods described in the next section.

B. Consolidation of predictions

Each IC generates a prediction $\hat{v}_i$ for the value v_i. These predictions then need to be consolidated to achieve the IS prediction. Therefore appropriate consolidation methods are necessary. We have implemented and tested five different techniques:

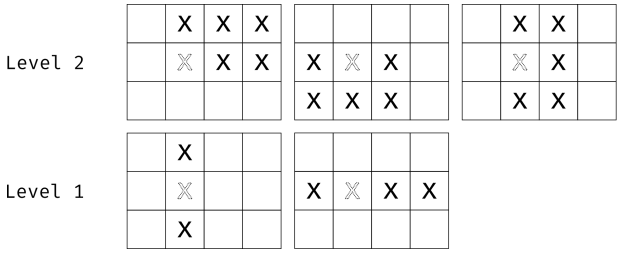


Fig. 3: The Information Space in Figure 2 can be split into five Information Contexts on two levels. These IC can then be used to predict the value depicted as a dotted X.

- Average (AV)
  Takes the average of the IC predictions.
- Minimum (Min)
  Takes the minimum of the IC predictions.
- Maximum (Max)
  Takes the maximum of the IC predictions.
- LastBest (LB)
  Tracks which IC was best for s_{i-1} and uses its prediction.
- Reforced (R)
  Given an order of dimensions, the ICs are sorted according to the number of data points used from each dimension and the prediction from the IC with the most data points from the preferred dimension is used.

The motivation behind using Minimum and Maximum for the consolidation process is to find out if the predictor has a bias towards one or the other.

With the introduction of Information Spaces and Contexts, as well as methods for consolidation of the prediction, we will now go into more detail of the traversal step building the sequence S in Eq. 4.

C. *Traversal methods*

In order to ensure an ideal use of the Information Spaces we have to look at the traversal method again. Complementary traversal methods to the use of Information Spaces could improve the predictions. For this purpose we propose next to the common linear traversal three new ones:

- Linear Traversal
- Chequerboard Traversal
- Blossom Traversal
- Block Traversal

An example for each traversal method is given in Figure 4.

a) Linear Traversal: The linear traversal determines an order for the dimensions and processes the data in this order. For the example given in Figure 4 the order is to first traverse the x-axis and then the y-axis.

b) Chequerboard Traversal: The sequence based on the chequerboard traversal is structured like a chessboard. As in the linear traversal an order must first be determined for the dimensions. For Figure 4 it is first x-axis and then y-axis.

c) Blossom Traversal: This traversal is structured like a blossoming rose which spreads around the starting point. Here, too, an order must be determined in which the dimensions will be processed. In Figure 4 the traversal runs clockwise starting at 12 o'clock.

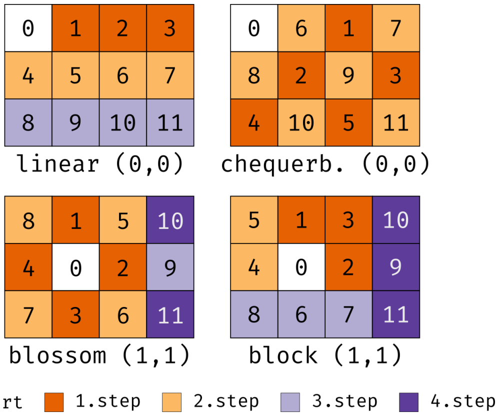


Fig. 4: Each traversal method creates a different Information Context to be used for the prediction of a point in the grid of size 3 × 4. Since building the sequence is an iterative process the colours depict each step. All traversals use an ordering for the dimensions to create reproducible results.

d) Block Traversal: The block traversal follows a sequence around the starting point with the aim of building fully connected blocks. In the two dimensional case this may look like a spiral around the starting point (see Fig. 4). Again, an order for the dimension needs to be considered.

Following our description of the common prediction-based compression algorithm at the beginning of this section we have defined the concept of Information Spaces, described new traversal methods for generating these Spaces and introduced techniques for merging the predictions of each Information Context defining the Information Space. Now we will describe the predictors used in the experiments followed by our applied metrics.

D. *Predictors*

Following predictors have been implemented and tested during our experiments:

- Akumuli [16]
- LastValue [15]
- Stride, TwoStride, Stride Confidence [15]
- Ratana 3, Ratana 5 [14]
- Pascal 1, Pascal 2, ... , Pascal 5 (based on [25])

The details for the individual predictors can be found in the respective articles in which they were introduced. The different variations of the predictors Ratana x and Pascal x define the number of elements used for prediction. In case of Ratana 3 this would be: $S_i = \{s_{i-3}, s_{i-2}, s_{i-1}\}$.

a) Pascal predictor: The Pascal predictor makes a prediction for an element s_i using the last k elements assuming that no noise is in the dataset. The predictor used here is based on an prediction method used in audio compression [25] and polynomial interpolation. The Pascal k is the optimal predictor for data without white noise and on a uniform grid which can be described by a polynomial function f of degree $i-1$ (Eq. 9). The coefficients of Pascal 1-5 are shown in Table I.

$$f(x) = \sum_{j=0}^{i-1} a_j \ x^j \tag{9}$$

TABLE I: Coefficients for Pascal k predictor using the last k values for prediction of s_i

Predictor	Formula
Pascal 1	$s_i = s_{i-1}$
Pascal 2	$s_i = 2 \ s_{i-1} - s_{i-2}$
Pascal 3	$s_i = 3 \ s_{i-1} - 3 \ s_{i-2} + s_{i-3}$
Pascal 4	$s_i = 4 \ s_{i-1} - 6 \ s_{i-2} + 4 \ s_{i-3} - s_{i-4}$
Pascal 5	$s_i = 5 \ s_{i-1} - 10 \ s_{i-2} + 10 \ s_{i-3} - 5 \ s_{i-4} + s_{i-5}$

The coefficients of Pascal k predictor can predict a polynomial function of order $k-1$ exactly.

Lemma IV-D.1. *Given the n-th order backwards difference* $\nabla_h^n[p](x)$ *the optimal coefficients are* $p(x) = \sum_{i=1}^{n}(-1)^{i+1}\binom{n}{i}p(x-i)$ *for uniform spacing* $h = 1$.

Proof. This can be shown using finite differences (which are zero in orders higher than those of the polynomial function):

$$\begin{aligned} \nabla_h^n[p](x) &= \sum_{i=0}^{n}(-1)^i\binom{n}{i}p(x-ih) \\ &\text{with } h = 1 \text{ and } \nabla_h^n[p](x) := 0 \\ 0 &= \sum_{i=0}^{n}(-1)^i\binom{n}{i}p(x-i) \\ 0 &= -1^0\binom{n}{0}p(x) + \sum_{i=1}^{n}(-1)^i\binom{n}{i}p(x-i) \\ p(x) &= \sum_{i=1}^{n}(-1)^{i+1}\binom{n}{i}p(x-i) \end{aligned}$$

□

The name Pascal has been chosen, because the coefficients can also be derived from Pascal's triangle. Now we will describe the experiments carried out and the data and metrics used in evaluation.

V. Experimental setup

This section will start with a description of the data used in the experiments and move on to the metrics for evaluation. Finally we will describe the experiments carried out.

A. Data

The data used in this paper was obtained from a composition simulation created by the ECHAM/MESSy[3] Atmospheric Chemistry (EMAC) model [26]. It consisted of a 128x64 (longitude, latitude) grid with 47 vertical levels. Three different time scales were used:

- January, 2013 with 74 time steps (every 10 hours)
- The year of 2013 with 365 time steps (every 24 hours)
- The years 2000-2013 with 168 time steps (every month)

The variables given in Table II were available as single-precision 32 bit floating-point values. For more representative results the experiments were contucted on randomly selected subsamples of the datasets described above (more details in Section V-C). The computations were done on an Intel Xeon E5-2640 v2 with 16 cores and 128 GiB memory.

B. Metrics

The main metric we will analyse is the leading zero count (LZC) of each residual defined in Equation 6. The LZC represents the amount of significant zeros of a number in binary representation.

The LZC is a measure for the quality of the prediction. It represents how many bits we do not need to safe on disk. Further on we can save an additional bit for each value since we know that the first bit of the residual must be one. Therefore we will use the following definition for LZC:

$$\text{LZC}(r) = \text{\#Significant zeros of } r + 1 \tag{10}$$

Another metric we will use is the compression ratio (CR) of the files being compressed:

$$\text{CR} = \frac{\text{Number of bits of compressed file}}{\text{Number of bits of original file}} \tag{11}$$

A ratio closer to zero suggests ideal compression and closer to one a bad compression[4].

C. Experiments

Several experiments have been conducted to investigate each step of the compression algorithm.

1) Expt 1: Influence of starting point: First we focus on the influence of different starting points on the compression ratio. For this purpose, we first choose random blocks[5] with 1024 data points from each data set and for each variable. Then we randomly select ten starting points per block and compress the data. This gives us unbiased information on whether and how susceptible the predictors (and therefore the compression ratio) are to different starting points.

2) Expt 2: Traversal order of dimensions: Since most prediction methods use linear traversal as described in Section II, in our second experiment, we analyse how the order of dimensions influences the CR of the files. We choose random blocks of 1024 data points. After this step we traverse along every possible ordering of dimensions using linear traversal. This experiment provides us information if the predictors need to be adjusted to the data structure and order of dimensions.

[3] ECMWF Hamburg (ECHAM)/Modular Earth Submodel System (MESSy)

[4] Please note that this is not the final compression ratio, since the encoding process (described in Section III-B0g) creates additional overhead.

[5] Most compression algorithm split the data into several blocks and compress each separately to save time during decompression by decompressing only the requested blocks of data.

TABLE II: Variables available in each dataset being used in the experiments.

Variable	Abbreviation
Specific humidity	Spec.Hum.
Relative humidity	Rel.Hum.
Pressure	Press.
Dry air temperature	Temp.
Zonal wind (S-N)	Wind (S-N)
Meridional wind (W-E)	Wind (E-W)

3) Expt 3: New traversal without the use of Information Spaces: In the third experiment we apply the newly proposed traversal methods but do not change the predictors. Since most of the predictors consider the traversal sequence as a data stream we are expecting changes in the compression ratio using the new traversal methods.

4) Expt 4: New traversal with the use of Information Spaces: Finally we conduct experiments with the fully adjusted predictors to the Information Space and the different consolidation methods suggested in Section IV-B. In case that no IS could be constructed (at start or in case of an empty IC) a Stride predictor was used as a fall-back predictor. For the neighbourhood constraints of IS (see Eq. 7) a value of r = 5 was chosen, since this is the highest amount of neighbours considered by the predictors (see Section IV-D).

This concludes our description of the experiments and metrics used. In the next section we will discuss and evaluate the results.

VI. Evaluation

In this section we will present and evaluate the results of the experiments described in the previous section.

A. Expt 1: Influence of starting point

The influence of different starting points are depicted in Table IIIa. The achieved LZC seems to be independent of the initial value for most predictors. The Stride Conf (SC) predictor had the highest relative standard deviation (SD) in the monthly data record. Here the LZC was 11.192 with SD being 0.124 which is about 1.1%. While overall the SD seems very low for starting point changes for any of the predictors, the SC and Ratana x predictor seem to be the most prone for changes of s_0. The SD of the remaining predictor were usually around 3%■. This is a magnitude lower than SC and Ratana.

This sensitivity can also be observed by looking at the difference plots in Figure 5. With Akumuli and Pascal one can see the two starting points of the different sequences. After a short time, the predictors give the same predictions as if the starting point had not changed. This is not the case with Stride Conf and Ratana. The differently predicted values are much more scattered or do not show a uniform pattern.

Another observation is the steady increase of SD of Pascal x. The more values are used for the prediction, the higher the fluctuation. This is valid across all datasets. The reason for this is that high order polynomials such as Pascal 4 and Pascal 5 lead to large local fluctuations and therefore worse extrapolation.

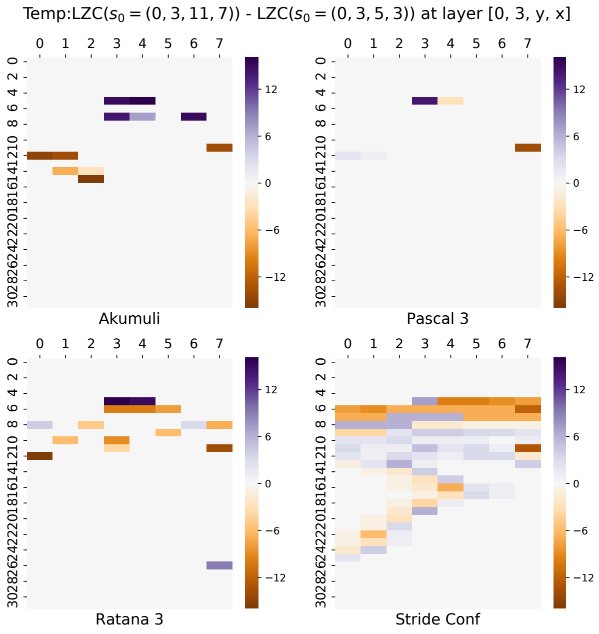


Fig. 5: [Expt 1] Difference plot of LZC for two different starting points s_0. The starting points were at (11,7) and (5,3) with (y, x). Akumuli and Pascal have only differences around the starting points, while the differences from Ratana and Stride Conf are distributed and more frequent (hence the higher SD).

B. Expt 2: Traversal order of dimensions

In our second experiment we analysed the effect of traversal order (see Table IIIb). The standard deviation has worsened several magnitudes, which confirms that the simple walk through the data in an arbitrary order does not lead to success.

In comparison to Expt 1 the LZC decreased and SD increased dramatically. The SD reaches rates higher than 21% for Pascal 1 and is significant. These results are not dependent on the variable being compressed.

This fluctuation is also reflected in Figure 6. The quality of the predictions are wildly disrupted for Pascal 3 and Ratana 3. The traversal order (0, 1, 2) seems to have been almost consistently better for Akumuli than the order of (1,2,0).

Also the observation done in the first experiment regarding the SD of the different Pascal predictors is not valid any more and the SD does not steadily increase.

C. Expt 3: New traversal without the use of Information Spaces

In most cases the predictors are getting the best results using linear traversal. Only twice did one of the new traversal methods perform slightly better: The Akumuli predictor reached 5.360 (before: 5.213) bits using block instead of linear traversal and Stride Conf reached 11.99 (before: 11.843) bits

TABLE III: [Expt 1 & 2] Leading Zero Count (LZC) and standard deviation (SD) across predictors for the daily, monthly and 10h dataset. On the left for varying starting points and on the right for all possible dimension orders using linear traversal.

(a) Expt 1: Starting points

	Daily		Monthly		10h	
	LZC	SD	LZC	SD	LZC	SD
Akumuli	11.55	0.041	12.89	0.036	12.73	0.029
Last Value	13.04	0.004	14.20	0.003	14.31	0.003
Stride	13.30	0.007	14.95	0.006	14.24	0.005
Stride Conf	10.77	0.056	11.19	0.124	13.64	0.053
Stride 2	12.36	0.011	13.59	0.010	13.80	0.006
Ratana 3	12.76	0.048	14.24	0.038	13.84	0.024
Ratana 5	12.76	0.048	14.24	0.039	13.84	0.023
Pascal 1	13.04	0.004	14.19	0.003	14.31	0.003
Pascal 2	11.42	0.005	12.57	0.005	12.97	0.004
Pascal 3	13.15	0.009	14.78	0.008	13.88	0.006
Pascal 4	12.51	0.011	14.20	0.010	13.07	0.008
Pascal 5	12.05	0.014	13.45	0.014	12.38	0.010

(b) Expt 2: Traversal order

	Daily		Monthly		10h	
	LZC	SD	LZC	SD	LZC	SD
Akumuli	9.90	0.98	9.81	1.09	10.70	0.57
Last Value	10.82	1.59	10.82	1.55	10.87	1.90
Stride	10.95	1.43	11.26	1.50	10.71	1.82
Stride Conf	9.53	0.68	9.50	0.66	9.88	1.00
Stride 2	10.27	1.48	10.13	1.56	10.24	1.88
Ratana 3	10.77	1.19	10.79	1.38	10.85	1.60
Ratana 5	10.77	1.19	10.79	1.38	10.84	1.60
Pascal 1	10.82	1.59	10.82	1.55	10.87	1.90
Pascal 2	9.30	1.47	9.17	1.58	9.37	1.83
Pascal 3	10.56	1.22	10.86	1.40	10.07	2.03
Pascal 4	9.69	1.27	9.87	1.45	9.25	2.00
Pascal 5	8.85	1.27	9.13	1.40	8.34	2.17

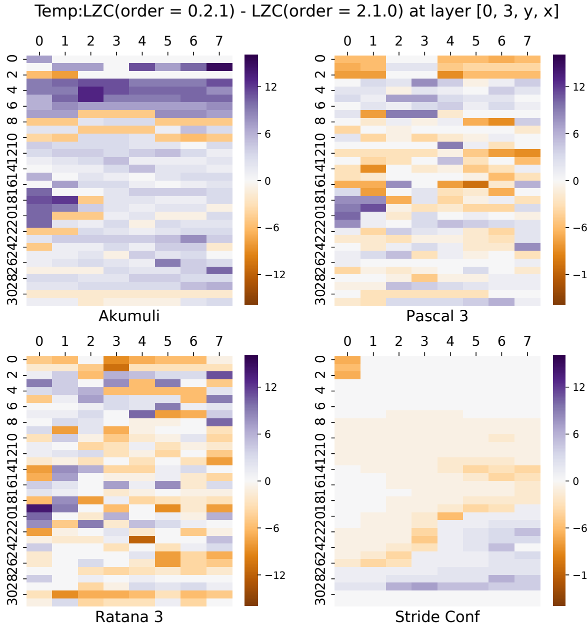


Fig. 6: [Expt 2] Difference plot of LZC for traversal orders (0, 1, 2) and (1,2,0). The figure suggests that the traversal order of the dimensions greatly influences the compression ratio. This explains the high variance in the LZC depicted in Table IIIb.

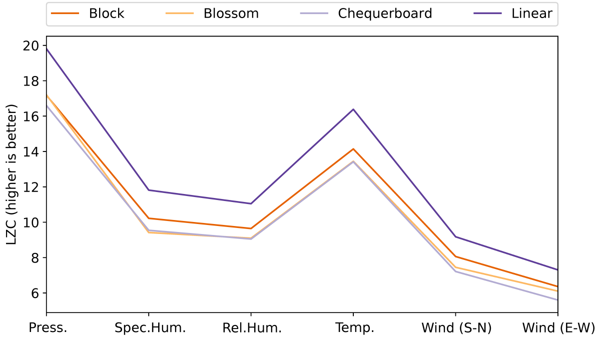


Fig. 7: [Expt 3] Depicted is the maximum reached LZC for each variable in the daily dataset without the use of IS.

compared to the linear traversal. This suggests that the linear traversal (given the correct ordering) is a safe choice.

Figure 7 depicts the maximum reached LZC for each variable across predictors for the monthly dataset. While the linear traversal is several bits better than the other traversal methods, there is close order for the other traversal methods: Block > Blossom > Chequerboard. There might be several reasons for this.

a) Chequerboard: Due to the usage of every other data point in the first half of the algorithm (described in IV-C) the data locality of the points in the sequence is scattered. This has more significance at the borders of the data cube since a jump might occur very often depending on the size of the cube. The value differences of jumps along a big dimension could be larger than those along a small dimension, because the distance covered is a larger one.

b) Blossom v Blocks: While the difference between both traversal algorithms is small, the data suggests that the Block algorithm is on average better for the tested predictors. The reason for this might be the building and prediction structure of Blocks. The number of interpolations vs. extrapolation is higher in the Block traversal than Blossom.

Now that we have seen the results without the use of IS in the next section we describe the results with the use of IS.

D. Expt 4: New traversal with the use of Information Spaces

Before we go into the details lets first give an overview of the overall results (Table IV). The LZC was increased by $9.622\% \pm .364\%$ and SD decreased by $23.547 \pm .932$ % on average. For Pascal 5 the LZC climbed from 9.448 to 12.416 bits (+31.411%, 10h dataset, LB consolidation) while the effective SD declined by 5.077%.

a) Performance of individual predictors: Next, we will rank all predictors by their LZC performance on each dataset. Each predictor was used with and without IS using each

TABLE IV: [Expt 4] Changes induced to LZC and SD by using IS for each dataset with linear traversal.

	10h	daily	monthly
ΔLZC	12.26%	9.03%	7.60%
ΔSD	-12.91%	-29.17%	-34.72%

consolidation method described in Section IV-B. The results are depicted in Table V.

TABLE V: [Expt 4] Ranking of individual predictors for each dataset. Due to reasons of space the Pascal *x* predictor is abbreviated with P*x*. The consolidation method is given in round brackets, if the predictor used our proposed method.

10h		daily		monthly	
Predictor	LZC	Predictor	LZC	Predictor	LZC
1. P2 (LB)	13.29	1. P3 (LB)	13.33	1. P3 (LB)	15.88
2. P3 (LB)	13.13	2. P2 (LB)	13.26	2. P3 (R)	15.76
3. P1 (LB)	13.00	3. P3 (R)	13.11	3. P4 (LB)	15.75
...	...	...	...	...	...
17. P1	12.30	16. P2	12.48	12. P2	14.86
17. Last Value	12.30	18. Stride	12.47	13. Stride	14.84
23. P2	12.11	19. P1	12.40	14. P3	14.62

The best results were achieved by predictors using IS with 15.88 (monthly dataset), 13.33 (daily dataset) and 13.29 LZC for the 10h dataset. These results were achieved by Pascal 3 (for the monthly and daily dataset) and Pascal 2 (10h dataset) in combination with the Last Best (LB) consolidation method.

The best predictors without the use of IS were ranked 12th (14.86 LZC, monthly), 16th (12.48, daily) and 17th (12.30, 10h) overall. While the best predictor for the monthly and daily dataset was Pascal 2, the best performance for the 10h dataset delivered Pascal 1 and Last Value.

Using IS helped Pascal 3 improve the LZC from 11.40 to 13.13 LZC for the 10h dataset. It jumped from 50th to the 2nd place in the ranking. This is a huge gain considering that we are dealing with 32 bit integers and the goal is lossless compression.

The results for each consolidation and traversal method for the daily dataset is represented in Figure 8. The relative differences to the common method are given in Table VI and VII, while Table VIII shows the compression ratios for each variable. For brevity we only depict the results for the daily dataset since the results for the monthly and 10h dataset are similar.

b) Comparing of traversal methods: In almost all cases the linear traversal method delivered better LZC results on average. The only exception was the Minimum consolidation method in combination with Blossom traversal. The LB method also had less fluctuation in its results than most other traversal method being in worst case the runner-up.

The results are interesting since the in Section VI-C mentioned possible order of traversal: Block > Blossom > Chequerboard does not seem to be valid any more. While the Chequerboard still performs worst, the Blossom method outperforms Block in every case regarding LZC.

TABLE VI: [Expt 4] LZC comparison (higher is better) of consolidation methods with the common method (SQ) using linear traversal. These results were obtained using the daily dataset.

	Block	Blossom	Cheq.	Linear
AV	70.69%	76.63%	67.04%	77.26%
LB	90.85%	105.82%	80.48%	107.55%
Max	73.08%	83.40%	65.86%	83.82%
Min	83.30%	89.74%	66.41%	90.16%
R	88.60%	100.77%	69.73%	105.07%
SQ	64.83%	69.70%	66.32%	100.00%

TABLE VII: [Expt 4] SD comparison (lower is better) of consolidation methods with the common method (SQ) using linear traversal. These results were obtained using the daily dataset.

	Block	Blossom	Cheq.	Linear
AV	214.50%	167.36%	195.66%	178.82%
LB	50.47%	63.67%	223.30%	50.66%
Max	253.35%	212.16%	537.39%	215.39%
Min	80.04%	87.71%	316.41%	95.48%
R	112.90%	122.96%	418.52%	60.37%
SQ	168.09%	194.27%	201.06%	100.00%

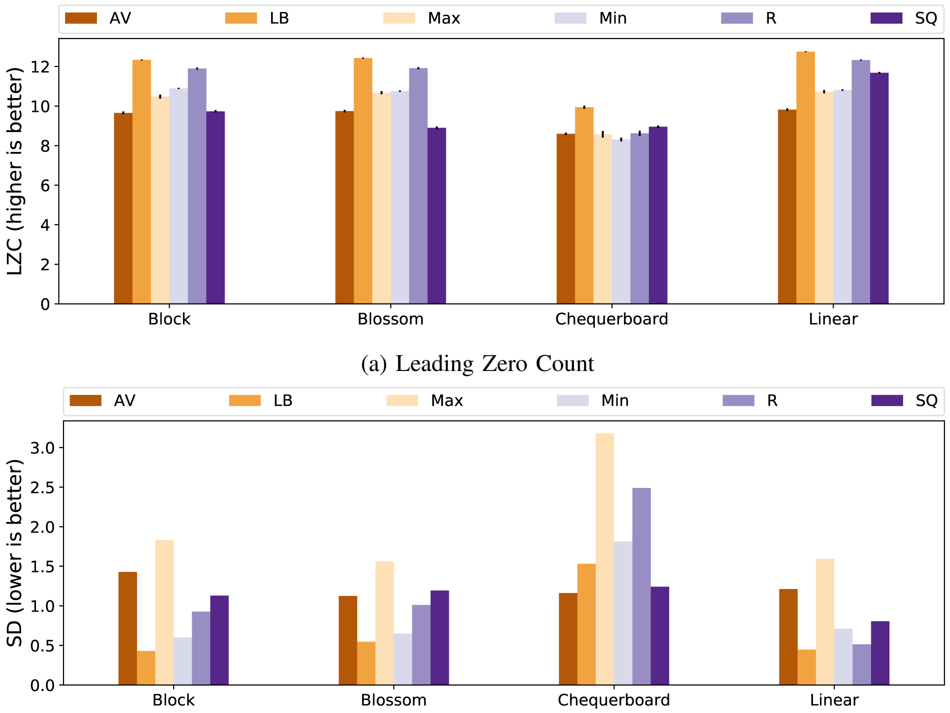
c) Maximum and Minimum consolidation: At the beginning of the experiments the Maximum and Minimum consolidation methods were used to gain information about possible biases of the predictions, interesting results came to be. While the LZC of both methods are similar the SD of Maximum is several times worse than the Minimum method. Using Block traversal the SD of the Maximum method increases by a factor of three compared to Minimum. This could suggest that the predictors are somewhat biased against the minima.

d) Performance per variable: In this section we will discuss the performance of IS with respect to the individual variables to find out whether the performance we have seen so far depended on the variable under consideration or not. The results are represented in Table VIII. A visual comparison of the best performing IS and the common sequential method is shown in Figure 9.

As in the previous results the LB consolidation method dominates in comparison to all other methods. With the exception for pressure and temperature it always emerges as the best method. In these two cases the method is only second best. The common method (SQ in Table VIII) is for every variable worse than LB and R consolidation.

It seems that the results are independent of the variables and it is recommended to use IC across the board.

e) Poor performance of AV: The results in Table VIII and VI suggest a poor performance of the Average method for consolidation. Each IC considers different neighbour points for the prediction of a single data point, but all of them are calculated using the same predictor. A bias - which the results from the Maximum/Minimum consolidation hints to -



(a) Leading Zero Count

(b) Standard deviation

Fig. 8: [Expt 4] Leading Zero Count and standard deviation of predictors using IS and not IS. Depicted are the mean leading zero counts/standard deviations across all variables for the daily dataset. Each bar represents a consolidation method described in Section IV-B with *SQ* being the classic approach without the use of Information Spaces.

might be affirmed by averaging and would explain the poor performance.

TABLE VIII: [Expt 4] Highest achieved compression ratio using the best traversal and prediction method per variable for the daily dataset.

bpf	Press.	Spec. Hum.	Rel. Hum.	Temp.	Wind (N-S)	Wind (E-W)
AV	0.376	0.659	0.673	0.520	0.736	0.795
LB	0.337	0.623	0.644	0.464	0.681	0.740
Max	0.350	0.657	0.671	0.512	0.729	0.775
Min	0.375	0.654	0.673	0.508	0.725	0.786
R	0.360	0.619	0.641	0.459	0.692	0.750
SQ	0.381	0.631	0.655	0.488	0.713	0.772

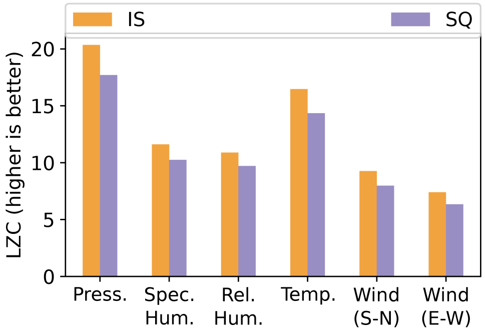


Fig. 9: [Expt 4] LZC per variable using the best traversal and prediction method per variable for the daily dataset.

VII. Summary and Outlook

We analysed the performance of different prediction-based compression algorithms on climate data. The results of our experiments showed that changing the starting point of the compression algorithm has only negligible effects on the compression ratio, while changing the traversal direction can influence the compression ratio significantly.

We further introduced the concept of Information Spaces (IS) and showed that with the help of IS it is possible to improve the predictions of each predictor. More importantly, the stability of the predictions was increased. The Information Contexts which define the Information Space helped consolidate information from several dimensions. This resulted in higher quality forecasts with less fluctuation than with the usual method.

The advantages of our method are higher stability and better compression ratios. Of course, the use of Information Spaces increased the complexity of the process. The calculation of the IS on each step was memory intensive and created an overhead. However, the potential advantages of this new model have not yet been exhausted.

There are still different optimisation possibilities. For example, possible weights can be considered which can be used within the Information Contexts for the prediction. The individual Information Contexts can be evaluated by calculating grading factors (such as information density), which might allow to decide which Information Contexts to use or to avoid using for the prediction. The different subgrids of the Information Contexts could also be considered separately in this grading process. However, our current configuration achieved already a 10% improvement on LZC and decreased the standard deviation of the compression results by over 20% on average.

While 10% for LZC and 20% may not be high for small scale datasets (<10 GiB) but for climate research, which is dealing with high volume of data (>300 TiB) a lossless data reduction of 10% is rather significant. This gain in storage space could reduce acquisition costs for new HPC systems and help improve more efficient usage of available storage space.

The use of Information Spaces offers new possibilities and levers to further increase compression ratios and gain independence from the internal structure of the data. While our focus is on high volume spatio-temporal climate data, the proposed method can also be used for any kind of gridded or meshed data.

Code and Data availability

The data and an implementation of the concepts described in this work will be made available under GNU GPLv3 license at [27].

References

[1] J. Schröter, D. Rieger, C. Stassen, H. Vogel, M. Weimer, S. Werchner, J. Förstner, F. Prill, D. Reinert, G. Zängl, M. Giorgetta, R. Ruhnke, B. Vogel, and P. Braesicke, "ICON-ART 2.1: a flexible tracer framework and its application for composition studies in numerical weather forecasting and climate simulations," *Geoscientific Model Development*, vol. 11, no. 10, pp. 4043–4068, 2018. [Online]. Available: https://www.geosci-model-dev.net/11/4043/2018/

[2] A. H. Baker, H. Xu, J. M. Dennis, M. N. Levy, D. Nychka, S. A. Mickelson, J. Edwards, M. Vertenstein, and A. Wegener, "A Methodology for Evaluating the Impact of Data Compression on Climate Simulation Data," in *Proc. 23rd Int. Symp. High-performance Parallel Distrib. Comput.* New York, New York, USA: ACM Press, 2014, pp. 203–214. [Online]. Available: http://doi.acm.org.ezproxy.lib.utexas.edu/10.1145/2600212.2600217

[3] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, 1997.

[4] X. Li and M. T. Orchard, "Edge-directed prediction for lossless compression of natural images," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 813–817, 2001.

[5] G. Ulacha and R. Stasinski, "Entropy Coder for Audio Signals," *Int. J. Electron. Telecommun.*, vol. 61, no. 2, pp. 219–224, Jan 2015.

[6] C. D. Giurcaneanu, I. Tabuş, and J. Astola, "Adaptive context-based sequential prediction for lossless audio compression," *Signal Processing*, vol. 80, no. 11, pp. 2283–2294, 2000.

[7] P. Lindstrom and M. Isenburg, "Fast and Efficient Compression of Floating-Point Data," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 1245–1250, Sep 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4015488

[8] N. Fout and K.-L. Ma, "An Adaptive Prediction-Based Approach to Lossless Compression of Floating-Point Volume Data," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2295–2304, Dec 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6327234

[9] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization," in *2017 IEEE Int. Parallel Distrib. Process. Symp.* IEEE, May 2017, pp. 1129–1139. [Online]. Available: http://ieeexplore.ieee.org/document/7967203/

[10] U. Cayoglu, J. Schröter, J. Meyer, A. Streit, and P. Braesicke, "A Modular Software Framework for Compression of Structured Climate Data," in *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '18)*, 2018. [Online]. Available: https://doi.org/10.1145/3274895.3274897

[11] S. Liu, X. Huang, Y. Ni, H. Fu, and G. Yang, "A High Performance Compression Method for Climate Data," in *2014 IEEE Int. Symp. Parallel Distrib. Process. with Appl.* IEEE, Aug 2014, pp. 68–77. [Online]. Available: http://ieeexplore.ieee.org/document/6924431/

[12] X. Huang, Y. Ni, D. Chen, S. Liu, H. Fu, and G. Yang, "Czip: A Fast Lossless Compression Algorithm for Climate Data," *Int. J. Parallel Program.*, vol. 44, no. 6, pp. 1248–1267, Dec 2016. [Online]. Available: http://link.springer.com/10.1007/s10766-016-0403-z

[13] U. Cayoglu, P. Braesicke, T. Kerzenmacher, J. Meyer, and A. Streit, "Adaptive lossy compression of complex environmental indices using seasonal auto-regressive integrated moving average models," in *2017 IEEE 13th International Conference on e-Science (e-Science)*, Oct 2017, pp. 315–324.

[14] P. Ratanaworabhan, J. Ke, and M. Burtscher, "Fast Lossless Compression of Scientific Floating-Point Data," in *Data Compression Conf.*, no. August. IEEE, 2006, pp. 133–142. [Online]. Available: http://ieeexplore.ieee.org/document/1607248/

[15] B. Goeman, H. Vandierendonck, and K. de Bosschere, "Differential FCM: increasing value prediction accuracy by improving table usage efficiency," in *Proc. HPCA Seventh Int. Symp. High-Performance Comput. Archit.* IEEE Comput. Soc, 2001, pp. 207–216.

[16] L. Eugene, "Akumuli Time-series Database," 2017. [Online]. Available: http://www.akumuli.org

[17] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for Similarity Search: A Survey," pp. 1–29, Aug 2014. [Online]. Available: http://arxiv.org/abs/1408.2927

[18] D. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep 1952. [Online]. Available: http://ieeexplore.ieee.org/document/4051119/

[19] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun 1987. [Online]. Available: http://portal.acm.org/citation.cfm?doid=214762.214771

[20] G. N. N. Martin, "Range encoding: an algorithm for removing redundancy from a digitised message," in *Proc. IERE Video Data Rec. Conf*, 1979.

[21] S. Golomb, "Run-length encodings (Corresp.)," *IEEE Trans. Inf. theory*, vol. 12, no. 3, pp. 399–401, 1966.

[22] J. Duda, "Asymmetric numeral systems: entropy coding combining speed of Huffman coding with compression rate of arithmetic coding," pp. 1–24, Nov 2013. [Online]. Available: http://arxiv.org/abs/1311.2540

[23] M. Boisot, *Information Space (RLE: Organizations).* Routledge, 2013, vol. 2.

[24] C. D. Cantrell, *Modern mathematical methods for physicists and engineers.* Cambridge University Press, 2000.

[25] T. Robinson, "Shorten: Simple lossless and near-lossless waveform compression," 1994.

[26] P. Jöckel, H. Tost, A. Pozzer, C. Brühl, J. Buchholz, L. Ganzeveld, P. Hoor, A. Kerkweg, M. G. Lawrence, R. Sander, B. Steil, G. Stiller, M. Tanarhte, D. Taraborrelli, J. van Aardenne, and J. Lelieveld, "The atmospheric chemistry general circulation model ECHAM5/MESSy1: consistent simulation of ozone from the surface to the mesosphere," *Atmospheric Chemistry and Physics*, vol. 6, no. 12, pp. 5067–5104, 2006. [Online]. Available: http://www.atmos-chem-phys.net/6/5067/2006/

[27] U. Cayoglu, "Repository for "Concept and Analysis of Information Spaces to Improve Prediction-Based Compression"," https://github.com/ucyo/informationspaces, 2018, [Online; accessed 22-August-2018].