Hydrol. Earth Syst. Sci., 17, 3171–3187, 2013
www.hydrol-earth-syst-sci.net/17/3171/2013/
doi:10.5194/hess-17-3171-2013

Hydrology and Earth System Sciences
Open Access

Data compression to define information content of hydrological time series

S. V. Weijs[1], **N. van de Giesen**[2], **and M. B. Parlange**[1]

[1]School of Architecture, Civil and Environmental Engineering, Ecole Polytechnique Fédérale de Lausanne, Station 2, 1015 Lausanne, Switzerland
[2]Water resources management, Delft University of Technology, Stevinweg 1, P.O. Box 5048, 2600 GA Delft, The Netherlands

Correspondence to: S. V. Weijs (steven.weijs@epfl.ch)

Received: 31 January 2013 – Published in Hydrol. Earth Syst. Sci. Discuss.: 14 February 2013
Revised: 21 June 2013 – Accepted: 24 June 2013 – Published: 6 August 2013

Abstract. When inferring models from hydrological data or calibrating hydrological models, we are interested in the information content of those data to quantify how much can potentially be learned from them. In this work we take a perspective from (algorithmic) information theory, (A)IT, to discuss some underlying issues regarding this question. In the information-theoretical framework, there is a strong link between information content and data compression. We exploit this by using data compression performance as a time series analysis tool and highlight the analogy to information content, prediction and learning (understanding is compression). The analysis is performed on time series of a set of catchments.

We discuss both the deeper foundation from algorithmic information theory, some practical results and the inherent difficulties in answering the following question: "How much information is contained in this data set?".

The conclusion is that the answer to this question can only be given once the following counter-questions have been answered: (1) information about which unknown quantities? and (2) what is your current state of knowledge/beliefs about those quantities?

Quantifying information content of hydrological data is closely linked to the question of separating aleatoric and epistemic uncertainty and quantifying maximum possible model performance, as addressed in the current hydrological literature. The AIT perspective teaches us that it is impossible to answer this question objectively without specifying prior beliefs.

1 Introduction

How much information is contained in hydrological time series? This question is not often explicitly asked, but is actually underlying many challenges in hydrological modeling and monitoring. The information content of hydrological time series is, for example, relevant for decisions regarding what to measure and where in order to achieve optimal monitoring network designs (Alfonso et al., 2010a,b; Mishra and Coulibaly, 2010; Li et al., 2012). Also, in hydrological model inference and calibration, the above question can be asked in order to decide how much model complexity is warranted by the data (Jakeman and Hornberger, 1993; Vrugt et al., 2002; Schoups et al., 2008; Laio et al., 2010; Beven et al., 2011).

There are, however, some issues in quantifying information content of data. Although the question seems straightforward, the answer is not. This is partly due to the fact that the question is not completely specified. The answers found in data are relative to the question that one asks of the data. Moreover, the information content of those answers depends on how much was already known before the answer was received. An objective assessment of information content is therefore only possible when prior knowledge is explicitly specified.

In this paper, we take a perspective from (algorithmic) information theory, (A)IT, on quantifying information content in hydrological data. This puts information content in the context of data compression. The framework naturally shows how specification of the question and prior knowledge enter the problem, and to what degree an objective assessment is

Published by Copernicus Publications on behalf of the European Geosciences Union.

possible using tools from information theory. The illustrative link between information content and data compression is elaborated in practical explorations of compressibility, using common compression algorithms.

This paper must be seen as a first exploration of the compression framework to define information content in hydrological time series, with the objective of introducing the analogies and showing how they work in practice. Section 2 first gives the detailed background of information theory, algorithmic information theory and the connections between probability, information and description length. Section 3 describes the compression experiment to determine information content of hydrological time series. The results will also serve as a benchmark in a follow-up study by Weijs et al. (2013b), where use of a newly developed hydrology-specific compression algorithm leads to improved compression that is interpreted as a reduction in information content of data due to prior knowledge. The inherent subjectivity of information content is the focus of the discussion in Sect. 5. The paper is concluded by a summary of the findings and an outlook to future experiments.

2 Information content, patterns and compression of data

From the framework of information theory (IT), originating from Shannon (1948), we know that information content of a message, data point, event or observation can be equated to surprisal, defined as $-\log p_i$, where p_i is the probability assigned to event i before observing it. Subsequently, Shannon (1948) defined a measure for uncertainty named "entropy", which is the expectation of the surprisal of observed outcomes of a discrete random variable X, with known probability mass function (PMF) $\boldsymbol{p}(X)$:

$$H(X) := H(\boldsymbol{p}(X)) := \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}, \tag{1}$$

where $H(X)$ is the entropy of random variable X, or more precisely of its probability distribution $\boldsymbol{p}(X)$, measured in bits; and p_i is the probability of observing the i-th out of n possible values X can take. Uncertainty or Shannon entropy of a distribution can be interpreted as the missing information to obtain certainty, or equivalently as the average information content of the observation of one outcome x of random variable X if $\boldsymbol{p}(X)$ was known before.

The base of the logarithm determines the unit in which uncertainty or missing information is measured. In this paper, we use logarithms to base 2, yielding information measured in bits. This facilitates the connection to file sizes, which are measured in the same unit. One bit can also be interpreted as the information content of the answer to one optimal yes/no (polar) question. An optimal yes/no question ensures both answers are a priori equally likely in order to maximize the uncertainty resolved by the answer. If, for example, a streamflow value must be guessed, the most efficient way is to ask a series of questions of the form "Is Q higher than x?", where x is the median of the distribution reflecting the current knowledge of Q, given all previous answers received. Another unit for information that is commonly used the in hydrological literature is the "nat" (1 nat $\approx$ 1.44 bits), resulting from use of the natural logarithm. We do not use it here due to the lack of a clear interpretation in the data compression context.

We refer the reader to Shannon (1948) and Cover and Thomas (2006) for more background on information theory. See also Weijs et al. (2010a,b) for introduction and interpretations of information measures in the context of hydrological prediction and model calibration. We also refer the reader to Singh and Rajagopal (1987), Singh (1997) and Ruddell et al. (2013) for more references on applications of information theory in the geosciences. In the following, the interpretation of information content as description length is elaborated.

2.1 Information theory: entropy and description length

For the data compression perspective, data can be regarded as a file stored on a computer, i.e., as a sequence of symbols, e.g., numbers, that represent events or values that correspond to quantities in a real or modeled world. Data compression seeks more efficient descriptions for data stored in a specific format so they can be stored or transmitted more efficiently, which can save resources. Of greater interest to hydrology is the fact that the size of a description can be interpreted as the information content of data.

In this paper, we focus on lossless compression as opposed to lossy compression. This means that we look exclusively at descriptions from which it is possible to reproduce the original data exactly. Lossy compression achieves further compression by approximate instead of exact descriptions of the data set. Lossy compression is mainly used for various media formats (pictures, video, audio), where these errors are often beyond our perceptive capabilities. This is analogous to a description of the observed values to within measurement precision, which could be a way to account for uncertainties in observation (Beven and Westerberg, 2011; Westerberg et al., 2011; Weijs and Van de Giesen, 2011; Weijs et al., 2013a). In this paper, we use lossless compression of time series after first coarse-graining them to deal with limited observation precision and time series length (Paluš, 1996); see Sect. 3.1.

Generally speaking, lossless compression is achieved by exploiting patterns in a data set. One of those patterns is the fact that not all symbols or events are equally likely to occur in the data set. Data compression seeks to represent the most likely events (e.g., the most frequent characters in a text file or the most frequent daily rainfall amount in a time series) with the shortest descriptions (sequences of symbols), yielding the shortest total description length. On the most basic level of a binary computer, a data point is described by a

	occurrence frequencies			codes		expected code lengths per value					
event	I	II	III	A	B	A_I	B_I	A_II	B_II	A_III	B_III
CC	0.25	0.5	0.4	00	0	0.5	0.25	1	0.5	0.8	0.4
OO	0.25	0.25	0.05	01	10	0.5	0.5	0.5	0.5	0.1	0.1
GG	0.25	0.125	0.35	10	110	0.5	0.75	0.25	0.375	0.7	1.05
RR	0.25	0.125	0.2	11	111	0.5	0.75	0.25	0.375	0.4	0.6
total	H=2	H=1.75	H=1.74			2	2.25	2	1.75	2	2.15

```
OOCCCCRRCCGGOOCC

0100001000110100   CODE A: 16 bits, 2/color

10001110110100     CODE B: 14 bits, 1.75/color

               <<  COMPRESSION
```

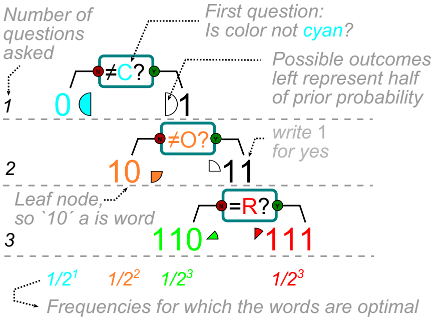
Number of questions asked
First question: Is color not cyan?
Possible outcomes left represent half of prior probability
≠C?
1
0
1
≠O?
write 1 for yes
2
10
11
Leaf node, so `10´ a is word
=R?
3
110
111
1/2^1
1/2^2
1/2^3
1/2^3
Frequencies for which the words are optimal

Fig. 1. Assigning code lengths proportional to minus the log of their probabilities leads to optimal compression. Code B is optimal for distribution II, but not for the other distributions. The figure on the right shows that in an optimal dictionary, every bit in a word answers one yes/no question with maximum a priori uncertainty (50 / 50 %), hence giving 1 bit of information. See the text for more explanation. Distribution III has no optimal code that achieves the entropy bound, because the probabilities are not negative integer powers of 2.

binary code or "word" (a sequence of zeros and ones), and the word lengths can be measured in bits.

In an efficient description, there is a close connection between word lengths and the probabilities of the events the words represent. A compression problem can be viewed as a prediction problem. As is the case with dividing high predictive probabilities, also short words are a limited resource that has to be allocated as efficiently as possible: short words come at the cost of longer words elsewhere. This follows from the fact that to be uniquely decodable from a sequence, such words must be prefix-free; that is, no word can be the first part (prefix) of another one.

The binary tree in Fig. 1 illustrates the connection between word lengths and probabilities for prefix-free words. When the variable length binary words for data points are concatenated in one file without spaces, they can only be unambiguously deciphered when no word in the dictionary forms the beginning of another word of the dictionary. In the binary tree, the prefix-free words must be at the leaves, since any word defined by an intermediate node is the prefix of all words on the downstream nodes. The depth of each branch represents the length of the corresponding word. The corresponding optimal probabilities of the events the words of length l_i encode are 2^{-l_i}. A way to interpret these optimal probabilities is the idea that every branching represents one yes/no question whose answer is encoded in one bit of the word. These questions are optimal if they represent 50 / 50 % uncertainty, leading to the optimal probabilities given in the figure. If the questions on each branch have less than 1-bit uncertainty (entropy), the answers give less than 1 bit of information per bit of word length and hence the efficiency of the coding is reduced. The scarcity of short words is formalized by the following theorem of McMillan (1956), who generalized the inequality (Eq. 2) of Kraft (1949) to all uniquely decodable codes (including those that do use, e.g., spaces).

$$\sum_i A^{-l_i} \leq 1, \tag{2}$$

in which A is the alphabet size (2 in the binary case, where the alphabet contains only the symbols 0 and 1) and l_i is the length of the word assigned to event i. In Fig. 1, the four 2-bit binary numbers of dictionary "A" are prefix-free, and can be used to uniquely describe the sequence of colors in Fig. 1. For dictionary "B", the use of a word of length 1 invalidates two words of 2-bit length and makes it necessary to use two words of length 3. We can verify that the word lengths of B sharply satisfy Eq. (2): using $A = 2$, we find $1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 2 \cdot 2^{-3} = 1 \leq 1$.

In Fig. 1, it is illustrated how the total description length of the color sequence can be reduced using dictionary B, which assigns words of varying length depending on occurrence frequency. As shown by Shannon (1948), if every value could be represented with one word, allowing for non-integer word lengths, the optimal word length for an event i is $l_i = \log(1/p_i)$. The minimum average word length is the expectation of this word length over all events, H bits per symbol (bps), where H can be recognized as the entropy of the distribution (Shannon, 1948; Cover and Thomas, 2006), which is a lower bound for the average description length per data point.

$$H(\boldsymbol{p}) = E_p\{l\} = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} \tag{3}$$

In contrast to probabilities p_i, which can be chosen freely, the word lengths l_i are limited to an integer number of bits. This results in some extra description length (overhead). The rounded coding would be optimal for a probability distribution of events,

$$q_i = \frac{1}{2^{l_i}} \forall i, \tag{4}$$

such as frequency II in Fig. 1. In Eq. (4), q_i is the i-th element of the PMF $\boldsymbol{q}$ for which the dictionary would be optimal, and l_i is the word length assigned to event i. The overhead in the case where $\boldsymbol{p} \neq \boldsymbol{q}$ is given by the relative entropy or

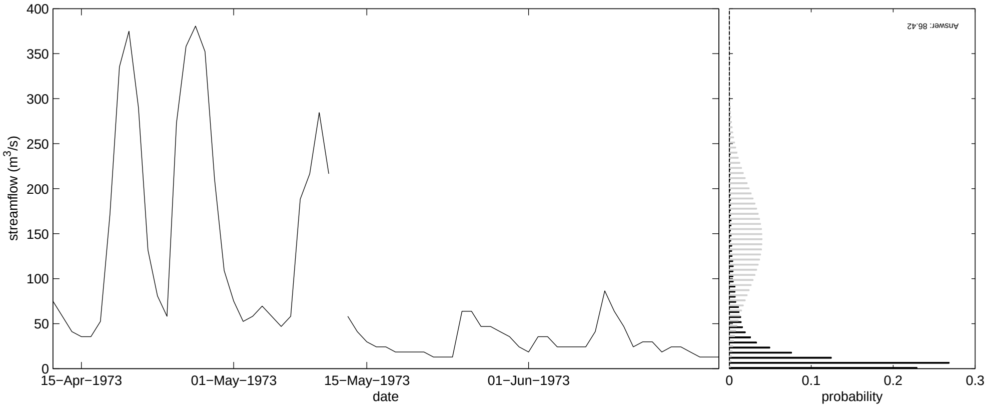

Fig. 2. The missing value in the flow time series can be guessed from the surrounding values (a guess would for example be the gray histogram). This will usually lead to a better guess than one purely based on the occurrence frequencies over the whole 40 yr data set (dark histogram) alone. The missing value therefore contains less information than when assumed independent.

Kullback–Leibler divergence, D_{KL}, (Kullback and Leibler, 1951) from $\boldsymbol{p}$ to $\boldsymbol{q}$,

$$D_{\mathrm{KL}}(\boldsymbol{p}||\boldsymbol{q}) = \sum_{i=1}^{n} p_i \log_2 \frac{p_i}{q_i}. \tag{5}$$

This divergence measure between two probability distributions measures the extra uncertainty introduced by approximating $\boldsymbol{p}$ with $\boldsymbol{q}$, or the extra description length per symbol when describing a sequence with symbol frequencies $\boldsymbol{p}$ when words are optimal for $\boldsymbol{q}$. This yields a total average word length of

$$H(\boldsymbol{p}) + D_{\mathrm{KL}}(\boldsymbol{p}||\boldsymbol{q}) \quad \text{bps}. \tag{6}$$

This extra description length is analogous to the reliability term in the decomposition of an information-theoretical score for forecast quality presented in Weijs et al. (2010b), where the extra overhead to store the dictionary can be interpreted as a complexity penalization; see Appendix A for an elaboration of this connection.

For probability distributions that do not coincide with integer ideal word lengths, the algorithm known as Huffman coding (Huffman, 1952) was proven to be optimal for value by value (one word per observation) compression. It constructs dictionaries with an expected word length closest to the entropy bound and is applied in popular compressed picture and music formats like JPEG, TIFF, MP3 and WMA. For a good explanation of the workings of this algorithm, the reader is referred to Cover and Thomas (2006). In Fig. 1, dictionary A is optimal for probability distribution I, and dictionary B is optimal for distribution II; see tree diagram. Both these dictionaries achieve the entropy bound. Dictionary B is also an optimal Huffman code for distribution III (last column in Fig. 1). Although the expected word length is now more than the entropy, it is impossible to find a shorter code. The overhead is equal to the Kullback–Leibler divergence from the true distribution (III) to the distribution for which the code would be optimal.

$$D_{\mathrm{KL}}(\mathrm{III}||\mathrm{II}) = 0.4106 \tag{7}$$

If the requirement that the codes are value by value is relaxed, blocks of values can be grouped together to approach an ideal probability distribution. When the series are long enough, entropy coding methods such as Shannon and Huffman coding using blocks can get arbitrarily close to the entropy bound (Cover and Thomas, 2006). This bound is also closely approached by arithmetic coding (Rissanen and Langdon, 1979), where the entire time series is coded as one single number. Range coding (Martin, 1979) is mathematically equivalent to arithmetic coding. Both have less overhead than Huffman coding.

To conclude, all the compression methods discussed so far make use of the marginal PMF of the variable, without taking temporal patterns into account. They are called entropy coding methods because they approach the entropy of the PMF, which is the lower bound for the average description length per data point in this case.

2.2 Dependency

If the values in a time series are not independent, however, the dependencies can be used to achieve even better compression. This high compression results from the fact that, for dependent values, the joint entropy is lower than the sum of entropies of indi dual values. In other words, average uncertainty per value decreases when all the other values in the series are known because we can recognize patterns in the series that therefore contain information about themselves. Hydrological time series often show strong internal depen encies, leading to bett r compression and better prediction. Consider, for example, the case where one is asked to assign probabilities (or code lengths) to possible streamflow values on 12 May 1973. In one case, the information offered is the dark-colored climatological histogram (Fig. 2 on the right), and in the second case, the time series is available (the left of the same figure). Obviously, the e pected compression and expected return for the bets are better in the second case, which shows the value of exploiting dependencies in the data set. The surprise ($-\log P_{\text{true value}}$) upon hearing the true value is 3.72 bits in the case where the guessed distribution was assumed, and 4.96 bits when using the climate as prior. These surprises are equivalent to the divergence scores proposed in Weijs et al. (2010b).

Another example is the omitted characters that the careful reader may (not) have found in the previous paragraph. There are 49 different characters used, but the entropy of the text is 4.3 bits, far less than $\log(49) = 5.6$, because of, for example, the relatively high frequencies of the space (16 %) and the letter "e" (13 %). Although the entropy is more than 4 bits, the actual uncertainty about the missing letters is far less for most readers because the structure in the text is similar to the English language, and that structure can be used to predict the missing characters. On the one hand, this means that the English language is compressible and therefore fairly inefficient. On the other hand, this redundancy leads to more robustness in the communication because even with many typographical errors, the meaning is still clear. If English were 100 % efficient, any error would obfuscate the meaning.

In general, better prediction, i.e., less surprise, gives better results in compression. In water resources management and hydrology we are generally concerned with predicting one series of values from other series of values, such as predicting streamflow (Q) from precipitation (P) and evaporation (E). In terms of data compression, knowledge of P and E would help compressing Q, but would also be needed for decompression. When P, E and Q would be compressed together in one file, the gain compared to compressing the files individually is related to what a hydrological model learns from the relation between these variables (Cilibrasi, 2007). Similarly, we can try to compress hydrological time series individually to investigate how much information those compressible series really contain for hydrological modeling.

2.3 Algorithmic information theory

Algorithmic information theory (AIT) was founded as a field by the appearance of three independent publications (Solomonoff, 1964; Chaitin, 1966; Kolmogorov, 1968). The theory looks at data through the lens of algorithms that can produce those data. The basic idea is that information content of an object, like a data set, is related to the shortest way to describe it. The use of algorithms instead of code words for the description could be compared to switching from a language with only efficiently assigned nouns to one with grammar. Hence, AIT is a framework to give alternative estimations of information content, taking more complex dependencies into account. Although description length generally depends on the language used, AIT uses the construct of a universal computer introduced by Turing (1937), the universal Turing machine (UTM), to show that this dependence takes the form of an additive constant, which becomes relatively less important when more data are available. Chaitin (1975) offered some refinements in the definitions of programs and showed a very complete analogy with Shannon's information theory, including, e.g., the relations between conditional entropy and conditional program lengths.

Using the thesis that any computable sequence can be computed by a UTM and that program lengths are universal up to an additive constant (the length of the program that tells one UTM how to simulate another), Kolmogorov (1968) gave very intuitive definitions of complexity and randomness; see also Li and Vitanyi (2008) for more background. Kolmogorov defined the complexity of a certain string (i.e., data set, series of numbers) as the length of the minimum computer program that can produce that output on a UTM and then halt. Complexity or information content of a data set is thus related to how complicated it is to describe. If there are clear patterns in the data set, then it can be described by a program that is shorter than the data set itself; otherwise, they are defined as random. This is analogous to the fact that a "law" of nature cannot really be called a law if its statement is more elaborate than the phenomenon that it explains; cf. Feynman (1967, p. 171): "When you get it right. it is obvious that it is right – at least if you have any experience – because usually what happens is that more comes out than goes in.". A problem with Kolmogorov complexity is that it is incomputable, but can only be approached from above. This is related to the unsolvability of the halting problem (Turing, 1937): it is always possible that there exists a shorter program that is still running (possibly in an infinite loop) that might eventually produce the output and then halt. A paradox that would arise if Kolmogorov complexity were computable is the following definition known as the Berry paradox: "the smallest positive integer not definable in under eleven words".

AIT can be seen as a theory underlying inference problems. Data mining techniques can be viewed as practical techniques that approximate idealized AIT methods such as Solomonoff's formal theory for inductive inference (Solomonoff, 1964), which can be seen as a golden but incomputable standard for prediction from data. AIT gives the bounds on what is possible and impossible and could give insights into assumptions underlying commonly used techniques. Any practical technique for inference of laws from data must make such assumptions to be computable, and AIT could serve to make explicit what these assumptions are.

2.4 Compression as practical approach to AIT

A shortcut approximation to measuring information content and complexity is to use a language that is sufficiently flexible to describe any sequence, while still exploiting most of commonly found patterns. While this approach cannot discover all patterns, like a Turing complete description language can, it will offer an upper-bound estimation, without having the problems of incomputability. Compressed files are such a language that use a decompression algorithm to recreate the object in its original, less efficient language. The compressed files can also be seen a programs for a computer, which is simulated by the decompression algorithm on another computer. Since the language is not Turing complete (e.g., no recursion is allowed), it is less powerful than the original computer. The constant additional description length for some recursive patterns is replaced by one that grows indefinitely with growing numbers of data. As an example, one can think of using a common compressed image format to store an ultra-high-resolution image of a fractal generated by a simple program. Although the algorithmic complexity with respect to the Turing complete executable fractal program language is limited by the size of the fractal program executable and its settings, the losslessly compressed output image will continue to grow with increasing resolution.

Notwithstanding these limitations, the compression framework can serve to give upper-bound estimates for information content of hydrological time series, given the specification of the context. We now present a practical experiment employing this method, and will subsequently use the results to discuss some important issues surrounding the concept of information content.

3 Compression experiment setup

In this experiment, a number of compression algorithms are applied to different data sets to obtain an indication of the amount of information they contain. Most compression algorithms use entropy-based coding methods such as introduced in the previous section, often enhanced by methods that try to discover dependencies and patterns in data sets, such as autocorrelation and periodicity.

The data compression perspective indicates that formulating a rainfall–runoff model has an analogy with compressing rainfall–runoff data. A short description of the data set will contain a good model about it, whose predictive power outperforms the description length of the model. However, not all patterns found in the data set should be attributed to the rainfall–runoff process. For example, a series of rainfall values is highly compressible due to the many zeros (a far from uniform distribution), the autocorrelation and the seasonality. These dependencies are in the rainfall alone and can tell us nothing about the relation between rainfall and runoff. The amount of information that the rainfall contains for the hydrological model is thus less than the number of data points multiplied by the number of bits to store rainfall at the desired precision. This amount is important because it determines the model complexity that is warranted by the data (Schoups et al., 2008). In fact, we are interested in the Kolmogorov complexity of the data, but this is incomputable. A crude practical approximation of the complexity is the file size after compression by some commonly available compression algorithms. This provides an upper bound for the information in the data.

Actually, also the code length of the decompression algorithm should be counted towards this file size (cf. a self-extracting archive). In the present exploratory example the inclusion of the algorithmic complexity of the decompression algorithm is not so relevant since the algorithm is general purpose and not biased towards hydrological data. This means that any specific pattern still needs to be stored in the compressed file. The compression algorithms will be mainly used to explore the relative differences in information content between different signals, since absolute determination of information content remains elusive.

3.1 Quantization

Due to the limited amount of data, quantization is necessary to make meaningful estimates of the representative frequency distributions, which are needed to calculate the amount of information and compression (Paluš, 1996). This is analogous to the maximum number of bins permitted to draw a representative histogram. As will be argued in the discussion, different quantizations imply different questions for which the information content of the answers is analyzed. All series were first quantized to 8-bit precision. Eight-bit size was chosen because the commonly available compression algorithms used in this study operate at the byte (8 bits) level: they assume that each byte represents one value and would not be able to detect dependencies and nonuniform distributions if a different number of bits per value is used. Furthermore, the lengths of the time series are sufficient to make a 256-bin histogram roughly representative. The quantization used a simple linear scheme (Eq. 8). Using this scheme, the series were split into $2^8 = 256$ equal intervals and converted into a series of 8-bit unsigned integers, $\boldsymbol{x}_{\text{integer}}$ (integers ranging from 0 to 255 that can be stored in 8 binary digits).

$$x_{\text{integer}} = \rfloor 0.5 + 255 \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}, \tag{8}$$

where $\min(\boldsymbol{x})$ and $\max(\boldsymbol{x})$ are the minimum and maximum values occurring in time series $\boldsymbol{x}$. These can be converted back to real numbers using

$$\boldsymbol{x}_{\text{quantized}} = \left(\frac{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}{255} \right) \boldsymbol{x}_{\text{integer}} + \min(\boldsymbol{x}). \tag{9}$$

Because of the limited precision achievable with 8 bits, $\boldsymbol{x}_{\text{quantized}} \neq \boldsymbol{x}$. This leads to rounding errors, which can be quantified as a signal-to-noise ratio (SNR). The SNR is the ratio of the variance of the original signal to the variance of the rounding errors.

$$\text{SNR} = \frac{\frac{1}{n} \sum_{t=1}^{n} (x_t - \bar{x})^2}{\frac{1}{n} \sum_{t=1}^{n} \left(x_t - x_{t,\text{quantized}} \right)^2} \tag{10}$$

Because the SNR can have a large range, it is usually measured in the form of a logarithm, which is expressed in the unit decibel: $\mathrm{SNR_{dB}} = 10\log_{10}(\mathrm{SNR})$.

To investigate the influence of quantization on the results, we also performed a quantization with 6-bit precision for one experiment. The 0–255 range integers were mapped to 0–63 range integers and stored as 8-bit integers with the first 2 bits always set to zero to allow for algorithms working at the byte level.

3.2 Compression algorithms

The algorithms that were used are a selection of commonly available compression programs and formats. Below are very short descriptions of the main principles and main features of each of the algorithms used and some references for more detailed descriptions. The descriptions are sufficient to understand the most significant pattern in the results. It is beyond the scope of this paper to describe the algorithms in detail. Links to executables and source codes of the used algorithms can be found in the Supplement.

- ARJ: Uses LZ77 (see LZMA) with sliding window and Huffman coding.

- WAVPACK: Is a lossless compression algorithm for audio files.

- JPG: The Joint Photography Experts Group created the JPEG standard, which includes a range of lossless and lossy compression techniques. Here the lossless coding is used, which uses a Fourier-like type of transform (discrete cosine transform) followed by Huffman coding of the errors.

- HDF_RLE: HDF (hierarchical data format) is a data format for scientific data of any form, including pictures, time series and metadata. It can use several compression algorithms, including run-length encoding (RLE). RLE replaces sequences of reoccurring data with the value and the number of repetitions. It would therefore be useful to compress pictures with large uniform surfaces and rainfall series with long dry periods.

- PPMD: A variant of prediction by partial matching, implemented in the 7Zip program. It uses a statistical model for predicting each value from the preceding values using a variable sliding window. Subsequently the errors are coded using Huffman coding.

- LZMA: The Lempel–Ziv–Markov-chain algorithm combines the Lempel–Ziv algorithm, LZ77 (Ziv and Lempel, 1977), with a Markov-chain model. LZ77 uses a sliding window to look for reoccurring sequences, which are coded with references to the previous location where the sequence occurred. The method is followed by range coding.

- BZIP2: Uses the Burrows and Wheeler (1994) block-sorting algorithm in combination with Huffman coding.

- PNG: Portable Network Graphics (PNG) uses a filter based on prediction of one pixel from the preceding pixels. Afterward, the prediction errors are compressed by the algorithm "deflate" that uses dictionary coding (matching repeating sequences) followed by Huffman coding.

- TIFF: A container image format that can use several compression algorithms. In this case PackBits compression was used, which is a form of run-length encoding.

3.3 Experiment A: comparison of generated and hydrological time series

In the first experiment, the algorithms are tested on a hydrological data set from Leaf River (MS, USA) near Collins, MS, at an elevation of 60 m above sea level. The upstream basin has an area of 1924×10^6 m^2 and is located in a humid subtropical climate (Köppen climate class Cfa). The annual runoff ratio is 0.42. The data consist of time series for rainfall "potential evapotranspiration", which is better described as apparent potential evaporation (Brutsaert, 2005), and streamflow from October 1948 to October 1988. The maximum recorded daily rainfall in this period was 222 mm. The discharge ranged from 0.044 m^3 s^{-1} to a peak of 1444 m^3 s^{-1}; see Fig. 2 for an example 2-month period and histogram. See, e.g., Vrugt et al. (2003) for a description and more references for this data set. As a reference, various artificially generated series were used; see Table 1. The generated series consist of 50 000 values, while the time series of the Leaf River data set contains 14 610 values (40 yr of daily values). All are quantized directly with the linear scheme using Eq. (8).

3.4 Experiment B: compression with a hydrological model

The second experiment is a first exploration of jointly compressing time series. In the previous experiment single time series were compressed to obtain an indication of their information content. Given the connection between modeling and data compression, a hydrological model should in principle be able to compress hydrological data. This can be useful to identify good models in information-theoretical terms, but can also be useful for actual compression of hydrological data. Although a more comprehensive experiment is left for future work, we perform a first test of estimating the performance of hydrological models using data compression tools.

The hydrological model HYMOD was used to predict discharge from rainfall for the Leaf River data set; see, e.g., Vrugt et al. (2009) for a description of model and data. Subsequently, the modeled discharges were quantized using the same scheme as the observed discharges in order to make resulting information measures comparable, i.e., assessed

Table 1. Signals used in experiment A. For implementation details, see the Supplement.

Signal	Description
constant	contains only one value repeatedly
linear	contains a slowly linearly increasing trend
uniform white	the output from the Matlab® function "rand", uniform white noise
Gaussian white	the output from the Matlab® function "randn", normally distributed white noise
sin 1	single sinusoidal wave (with a wavelength spanning all 50 000 values)
sin 100	repetition of 100 sinusoidal waves (with a wavelength spanning 1/100 of 50 000 values)
Leaf P	daily area-averaged rainfall series from the catchment of Leaf River (1948–1988)
Leaf Q	corresponding daily series of observed streamflow in Leaf River

relative to the same question. An error signal was defined by subtracting the modeled (Q_{mod}) from the observed (Q) quantized discharge. This gives a signal that can range from −255 to +255, but because the errors are sufficiently small, ranging from −55 to +128, this allows for 8-bit coding. In order to losslessly reproduce Q, we could store P, a rainfall–runoff model and the error time series needed to correct the modeled Q to the original measured time series. This way of storing Q and P leads to compression if the model is sufficiently parsimonious and the errors have a small range and spread (entropy), enabling compact storage. Since the model also takes some description space, it is a requirement for compression that the error series can be more compactly described than the original time series of Q. In this experiment we test whether that is the case for the HYMOD model applied to Leaf River. The reduction in file size could then be interpreted as the portion of the uncertainty in Q explained by P.

3.5 Experiment C: compression of hydrological time series from the MOPEX data set

In a third experiment, we looked at the spatial distribution of compressibility for daily streamflow and area-averaged precipitation data in the 431 river basins across the continental USA, as contained in the MOPEX data set, available at http://www.nws.noaa.gov/oh/mopex/mo_datasets.htm. The basins span a wide range of climates and characteristics, as can be seen from the summary of Table 2. For these experiments, the streamflow values are log-transformed before quantization to reflect the heteroscedastic uncertainty in the measurements. This results in a quantized signal with a moderate information loss, which retains a relatively high information content, reflected in an entropy close to the maximum possible 8 bits. Quantifying the information loss compared to the original signal remains elusive, since these signals are typically quantized and stored at much higher precision than the measurement precision warrants. Moreover, the series are too short to have representative histograms at the original precision. Missing values, which were infrequent, were removed from the series. Although this can have some impact on the ability to exploit autocorrelation and periodicity, the effect is deemed to be small and has a smaller influence than other strategies such as replacing the missing values by zero or a specific marker. We repeated the experiments for the time series quantized at 6-bit precision. Results of this compression experiment are presented in Sect. 4.3.

4 Results of the compression experiments

This section shows results from the compression analysis for single time series. Also, an example of compression of discharge, using a hydrological model in combination with knowledge of rainfall, is shown.

4.1 Results A: generated data

As expected, the file sizes after quantization are exactly equal to the number of values in the series, as each value is encoded by 1 byte (8 bits), allowing for $2^8 = 256$ different values, and stored in binary raw format. From the occurrence frequencies of the values, the entropy of their distribution was calculated. Normalized with the maximum entropy of 8 bits, the fractions in row 3 of Table 3 give an indication of the entropy bound for the ratio of compression achievable by entropy encoding schemes such as Huffman coding, which do not use temporal dependencies.

The signal-to-noise ratios in row 4 give an indication of the amount of data corruption that is caused by the quantization. As a reference, the uncompressed formats BMP (bitmap), WAV (waveform audio file format), and uncompressed HDF are included, indicating that the file size of those formats, relative to the raw data, does not depend on file contents, but only on the series length, because they have a fixed overhead that is relatively smaller for larger files.

The results for the various lossless compression algorithms are shown in rows 7–17. The numbers are the percentage of the file size after compression, relative to the original file size (a lower percentage indicates better compression). The best compression ratios per time series are highlighted. From the result it becomes clear that the constant, linear and

Table 2. Some statistics of the annual MOPEX basin characteristics show a wide range of behavior.

Statistic	Min.	Max.	Median	Mean	Std.
area (10^9 m^2)	.067	10.3	2.19	3.02	2.52
aridity index (P/PET)	0.22	4.32	1.17	1.21	0.55
runoff ratio	0.02	0.76	0.35	0.34	0.14
ET/PET	0.21	1.06	0.76	0.72	0.16
Climates (Köppen)	BSk, BWk, Cfa/b, Csa/b, Dfa/b/c, Dsb, Dwa				

Table 3. The performance, as percentage of the original file size, of well-known compression algorithms on various time series (see Table 1). The best results per signal are highlighted in italic.

Data set	Constant	Linear	Uniform white	Gaussian white	Sin 1	Sin 100	Leaf Q	Leaf P
file size	50 000	50 000	50 000	50 000	50 000	50 000	14 610	14 610
$\frac{H}{\log N}$	0.0	99.9	99.9	86.3	96.0	92.7	42.1	31.0
SNR	NaN	255.0	255.6	108.0	307.4	317.8	42.6	39.9
Uncompressed formats								
BMP	102.2	102.2	102.2	102.2	102.2	102.2	407.4	407.4
WAV	100.1	100.1	*100.1*	100.1	100.1	100.1	100.3	100.3
HDF_NONE	100.7	100.7	100.7	100.7	100.7	100.7	102.3	102.3
Lossless compression algorithms								
JPG_LS	12.6	12.8	110.6	94.7	12.9	33.3	33.7	49.9
HDF_RLE	2.3	2.7	101.5	101.5	3.2	92.3	202.3	202.3
WAVPACK	*0.2*	1.9	103.0	*87.5*	2.9	25.6	38.0	66.2
ARJ	0.3	1.0	100.3	88.0	3.1	1.9	33.7	40.0
PPMD	0.3	2.1	102.4	89.7	3.6	1.4	*27.7*	*36.4*
LZMA	0.4	0.9	101.6	88.1	1.9	1.2	31.0	37.8
BZIP2	0.3	1.8	100.7	90.7	3.0	2.3	29.8	40.5
PNG	0.3	*0.8*	100.4	93.5	*1.5*	*0.8*	40.2	50.0
GIF	2.3	15.7	138.9	124.5	17.3	32.0	38.8	45.9
TIFF	2.0	2.4	101.2	101.2	2.9	91.2	201.5	201.5

periodic signals can be compressed to a large extent. Most algorithms achieve this high compression, although some have more overhead than others. The uniform white noise is theoretically incompressible, and indeed none of the algorithms appears to know a clever way around this. The Gaussian white noise is also completely random in time, but does not have a uniform distribution. Therefore the theoretical limit for compression is the entropy bound of 86.3 %. The WAVPACK algorithm gets closest to the theoretical limit, but also several file archiving algorithms (ARJ, PPMD, LZMA BZIP2) approach that limit very closely. This is because they all use a form of entropy coding as a back end (Huffman and range coding). Note that the compression of this nonuniform white noise signal is equivalent to the difference in uncertainty or information gain due to knowledge of the occurrence frequencies of all values (the climate), compared to a naive uniform probability estimate; cf. the first two bars in Fig. 1 of Weijs et al. (2010a).

The results for the hydrological series firstly show that the streamflow series is better compressible than the precipitation series. This is interesting because the rainfall series has the lower entropy. The higher predictability and compressibility of the linearly quantized streamflow results from its strong autocorrelation. Furthermore, it can be seen that, for the rainfall series, the entropy bound is not achieved by any of the algorithms, presumably because of the overhead caused by the occurrence of 0 rainfall more than 50 percent of the time, while the code words cannot be shorter than 1 bit; see Eqs. (4) and (6). Further structure-like autocorrelation and seasonality cannot be used sufficiently to compensate for this overhead. In contrast to this, the streamflow series can be compressed to well below the entropy bound (27.7 % vs. 42.1 %) because of the strong autocorrelation between the data. These dependencies are best exploited by the PPMD algorithm, which uses a local prediction model that apparently can predict the correlated values quite accurately. Many of the algorithms cross the entropy bound, indicating that they use at least part of the temporal dependencies in the data set.

4.2 Results B: compression with a hydrological model

As a first attempt to estimate the information that P delivers about Q through a hydrological model, we analyzed the time series of Q and P for leaf river, along with the modeled Q (Q_{mod}) and its errors (Q_{err}). In Table 4, the entropies of the signals are shown. The second row shows the resulting file size as percentage of the original file size for the best compression algorithm for each series (PPMD or LZMA). The compressed size of the errors is an indication of the information content of the errors, i.e., the missing information about Q when Q_{mod} is available (since $Q = Q_{\text{mod}} + Q_{\text{err}}$). The compressed error size can be interpreted as the predictive uncertainty, assuming an additive error model, no biases and an error model using temporal dependencies to reduce predictive uncertainty. In that sense, it is more an indication of potential model performance, which can only be reached when using the model in combination with these error-correcting schemes.

The table also shows the statistics for the series where the order of the values was randomly permuted (Q^{perm} and $Q_{\text{err}}^{\text{perm}}$). As expected, this does not change the entropy, because that depends only on the histograms of the series. In contrast, the compressibility of the signals is significantly affected, indicating that the compression algorithms made use of the temporal dependence for the non-permuted signals. The joint distribution of the modeled and observed discharges was also used to calculate the conditional entropy $H(Q|Q_{\text{mod}})$, which should give an indication of potential minimum achievable predictive uncertainty, given the model and a not-necessarily-additive error model. It must be noted, however, that this conditional entropy is probably underestimated compared to what is representative for a longer time series, as it is based on a joint distribution with 255^2 probabilities estimated from 14 610 value pairs. This is the cost of estimating dependency without limiting it to a specific functional form. The estimation of mutual information needs more data than Pearson correlation because the latter is limited to a linear setting and looks at variance rather than uncertainty. In the description length, the underestimation of $H(Q|Q_{\text{mod}})$ is compensated by the fact that the dependency must be stored by the entire joint distribution. If joint distribution of Q and Q_{mod} is known a priori or enough data are available to make its description length negligible, $H(Q|Q_{\text{mod}})$ gives a theoretical limit of compressing Q with knowledge of P and the model, while not making use of temporal dependencies in Q unexplained by the model.

A somewhat unexpected result is that the errors seem more difficult to compress (31.5 %) than the observed discharge itself (27.7 %) even though the entropy is lower. Apparently the reduced temporal dependence in the errors (lag-1 autocorrelation coefficient $\rho = 0.60$), compared to that of the discharge ($\rho = 0.89$), offsets the gain in compression due to the lower entropy of the errors. Possibly, the temporal dependence in the errors becomes too complex to be detected by the compression algorithms. Further research is needed to determine the exact cause of this result, which should be consistent with the theoretical idea that the information in P should reduce uncertainty in Q. The Nash–Sutcliffe efficiency (NSE) of the model over the mean is 0.82, while the NSE over the persistence forecast ($Q_{\text{mod}}(t) = Q_{t-1}$) is 0.18 (see Schaefli and Gupta, 2007), indicating a reasonable model performance. Furthermore, the difference between the conditional entropy and the entropy of the errors could indicate that an additive error model is not the most efficient way of coding and consequently not the most efficient tool for probabilistic prediction. The use of, for example, heteroscedastic probabilistic forecasting models (e.g., Pianosi and Soncini-Sessa, 2009) for compression is left for future work.

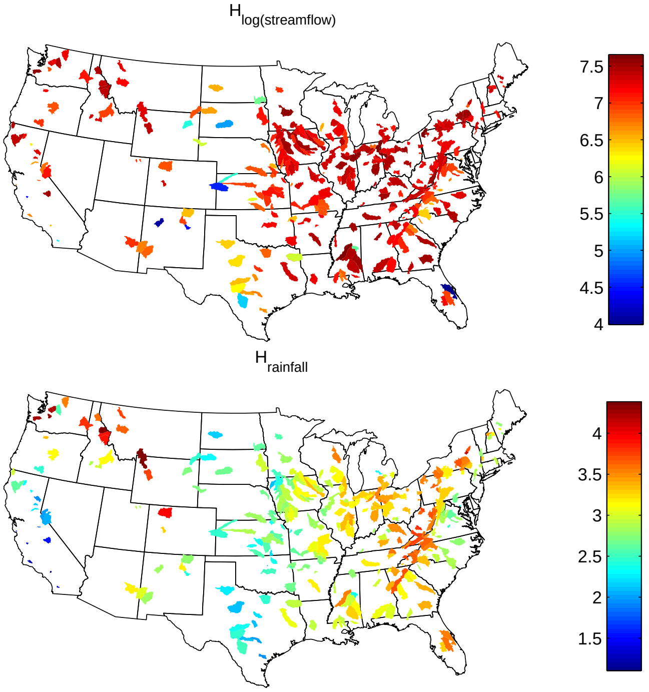

Fig. 3. Spatial distribution of entropy for quantized streamflow and rainfall shows the drier climate in the central part of the USA.

4.3 Results C: MOPEX data set

For the time series of the quantized scaled log streamflow and scaled quantized rainfall of the MOPEX basins, from now on simply referred to as streamflow (Q) and rainfall (P) for brevity, the compressibility and entropy show clear spatial patterns. For most of the streamflow time series, the entropy

Table 4. Information-theoretical and variance statistics and compression results (remaining file size %) for rainfall–runoff modeling.

Statistic	P	Q	Q_{mod}	Q_{err}	$Q \mid Q_{\text{mod}}$	Q^{perm}	$Q^{\text{perm}}_{\text{err}}$
entropy (% of 8 bits)	31.0	42.1	44.9	38.9	26.4	42.1	38.9
best compression (%)	36.4	27.7	25.8	31.5	N.A.	45.4	44.1
std. dev. (range = 256)	11.7	11.6	10.4	4.95	N.A.	11.6	4.95
autocorrelation ρ	0.15	0.89	0.95	0.60	N.A.	< 0.01	< 0.01

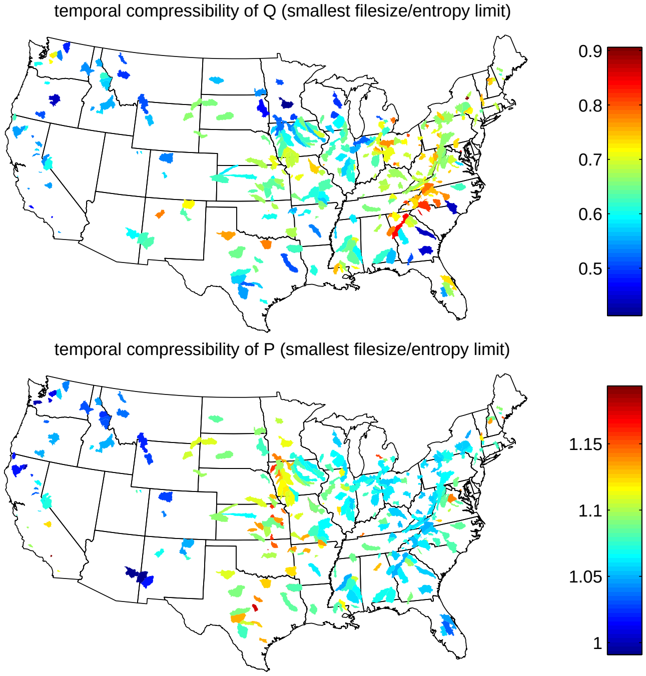
temporal compressibility of Q (smallest filesize/entropy limit)
0.9
0.8
0.7
0.6
0.5
temporal compressibility of P (smallest filesize/entropy limit)
1.15
1.1
1.05
1

Fig. 4. Spatial distribution the compression size normalized by entropy for streamflow and rainfall; this gives an indication of the amount of temporal structure found in the different basins. The streamflow is better temporally compressible due to the strong autocorrelation structure, but not enough to compensate for the higher entropy.

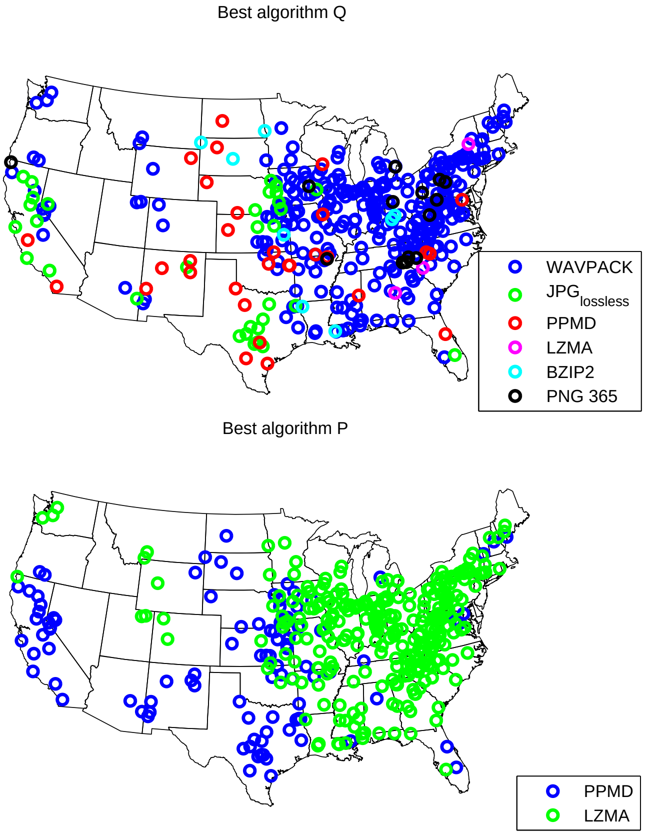

Fig. 5. Spatial distribution of the best-performing algorithms for streamflow and rainfall. This can give an indication as to what type of structure is found in the data. Especially for rainfall, the best-performing algorithm is linked to the number of dry days per year. See also Fig. 6.

is close to 8 bits, indicating that the frequency distribution of the preprocessed streamflow does not diverge much from a uniform distribution. An exception is the basins in the central part of the USA, which show lower entropy time series due to high peaks and relatively long, low base flow periods. Also for the rainfall, entropy values are lower in this region due to longer dry spells; see Fig. 3.

Compression beyond the entropy bound can be achieved by using temporal patterns. This is visible in Fig. 4, where the compression ratio of the best-performing algorithm is visualized relative to the entropy of the signals. The temporal compressibility is much better for streamflow. This is likely the result of autocorrelation due to low-pass-filter behavior of catchments, which are sufficiently large to dampen daily fluctuations in rainfall. Different algorithms are specialized in describing different kinds of patterns, so the map of best-performing algorithms (Fig. 5) can be used as an indication for which types of patterns are found in a data set. In this paper, we refrain from more elaborate interpretations of the results in connection with hydrological characteristics, and focus instead of properly understanding the technicalities and inherent difficulties in estimation of information content. The full results of the compression experiment are available in the Supplement to allow for further research into hydrological interpretations. In Fig. 6, two influences on compression rate are shown. Firstly, due to temporal dependencies in the streamflow, the conditional entropy given the previous value $H(Q_t|Q_{t-1})$, known as the entropy rate $H'(Q)$, is much lower than the entropy itself. This could theoretically lead to a compression describing the signal with $H'(Q)$ bits per time step. However, because of the relatively short length of the time series compared to the complexity of the model that describes it (a two-dimensional 256-bin histogram), this compression is not reached in practice, because the model needs to be stored too. This is a natural way of accounting for model complexity in the context of estimating information content of data.

The compression performance was also tested for time series at a coarser, 6-bit quantization. The compression percentages were then defined as the compressed file size divided by the original file size with 6 bits per sample ($= 0.75N$ bytes, where N is the length of the time series). Like for the 8-bit experiment, compression performance was compared to entropy. Figure 7 shows how the results for 6 and 8 bits compare when appropriately normalized. For P, a coarser quantization leads to higher compressibility due to a lower normalized entropy (non-uniformness of the distribution), which is only partly offset by the increased overhead caused by an higher percentage of days considered as dry (the threshold rainfall for a wet day goes up). For Q, the normalized entropy also decreases and the resulting improvement in compression rate is enhanced by the stronger temporal dependence, which is now less disturbed by small fluctuations.

In Table 5, correlations were calculated between the 6- and 8-bit quantization-based results. The change in results indicates that information content is to some degree subjective, but the relatively high correlations show that the tendencies in results are only moderately affected in this case. We will discuss the reasons behind subjectivity in more detail in the next section.

Table 5. Correlation coefficients (ρ) and Spearman rank correlations (r_S) between various results for 6- and 8-bit quantization for precipitation (P) and streamflow (Q). Correlations are shown for entropy (H), best compression rate (C), temporal structure (C/H) and the compression rates for the individual algorithms. Correlations are generally high, showing that tendencies in information content and temporal structure are similar for the different questions the 6- and 8-bit quantizations represent.

Statistic	ρ, P	r_S, P	ρ, Q	r_S, Q
H	0.973	0.971	0.935	0.952
C	0.973	0.966	0.956	0.978
C/H	0.894	0.814	0.971	0.971
WAVPACK	0.995	0.995	0.992	0.996
ARJ	0.975	0.976	0.983	0.993
PPMD	0.958	0.959	0.917	0.949
LZMA	0.976	0.975	0.924	0.943
BZIP2	0.966	0.959	0.930	0.972
PNG	0.978	0.977	0.967	0.983
GIF	0.977	0.974	0.983	0.993
TIFF	0.974	0.970	0.988	0.989

5 Discussion

The data compression results give an indication of the information content or complexity of the data sets. Eventually these may be linked to climate and basin characteristics and become a tool for hydrological time series analysis and inference. One possibility is the use of the temporal dependency indicators plotted in Fig. 4 as a correction factor for the number of data points, for use in significance tests or model complexity control measures such as the Akaike information criterion (Akaike, 1974), when dependence is not accounted for in the likelihood by using, e.g., a Markov-chain model (Katz, 1981; Cahill, 2003). This would facilitate choosing a model with appropriate complexity for the data. Additionally, estimates of the information content in measured data sets may be used to optimize the collection of information in, e.g., sensor networks. Although information theory may eventually provide a solid foundation for hydrological modeling, it is also important to first consider the limitations of such approaches. The patterns visible in the compression results can probably be given further hydrological interpretations, but there are several subtleties that should be considered before doing so. We therefore focus the discussion on some inherent issues in quantifying the information content, which make the results subjective and not straightforward to analyze.

5.1 How much information is contained in this data set?

From the presented theoretical background, results and analysis it can be concluded that although information theory can quantify information content, the outcome depends on a number of subjective choices. These subjective choices include the quantization, auxiliary data and prior knowledge used.

The quantization can be linked to what question the requested information answers. When quantizing streamflow into 256 equally sized classes, the question that is implicitly posed is “In which of these equally spaced intervals does the streamflow fall?”. When only 64 classes are used, less information is requested from the data, and hence less information will be contained in the answer. Also, the log-transform of Q changes the intervals, and therefore also the questions change. They request more absolute precision on the lower flows than on the higher flows. The information contained in the answers given by the data, i.e., the information content of the time series, depends on the question that is asked.

The information content of time series depends also on what prior knowledge one has about the answers to the question asked. If one knows the frequency distribution but has

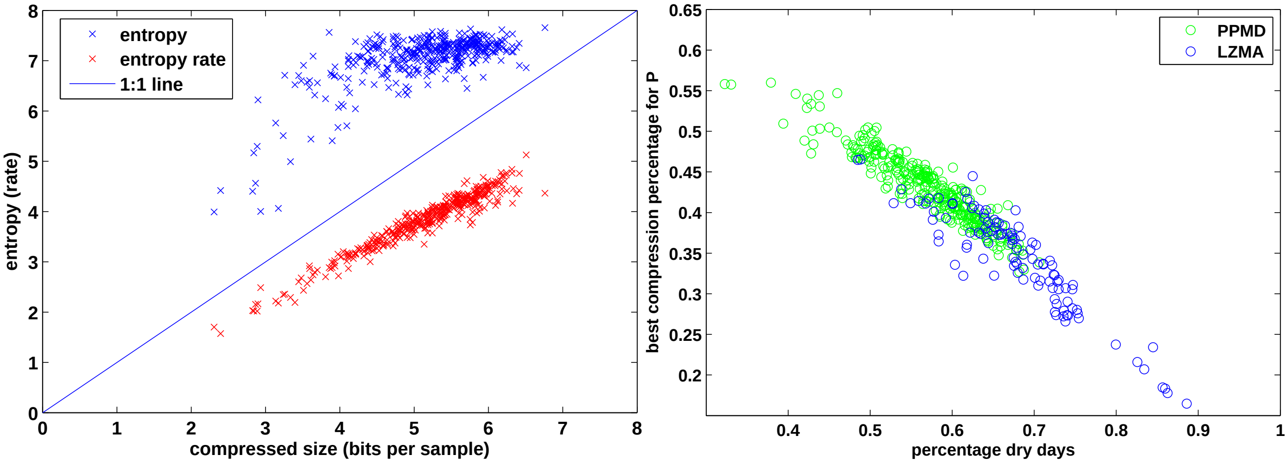

Fig. 6. Left: best compression of Q against entropy and against entropy rate. Temporal dependencies cause better compression than the entropy, but model complexity prevents achievement of the entropy rate. Right: the best achieved compression of P depends strongly on the percentage of dry days, mostly through the entropy. Also the best-performing algorithm changes with the climate.

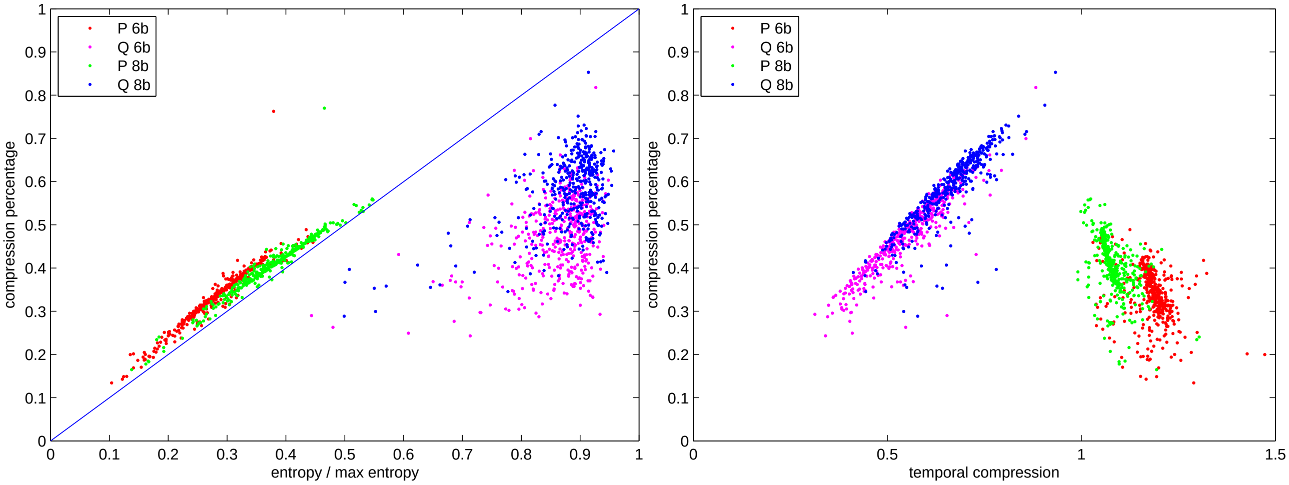

Fig. 7. Changes in results due to coarsening the quantization from 8 to 6 bits (8b and 6b) for precipitation (P) and streamflow (Q). For interpretation, we refer to the text.

no knowledge of surrounding values, the prior knowledge takes the form of a probability distribution that matches the observed frequencies. In that case, the expected information content of each observation is given by the entropy of the frequency distribution. The entropy in bits gives the limit of the minimum average space per observation needed to store a long i.i.d. time series of that distribution.

In many situations in practice, however, prior knowledge does not include knowledge of the occurrence frequencies, or does include more knowledge than frequencies alone, e.g., temporal dependencies. In the first case the information content of the data set should also include the knowledge gained from observing the frequencies. Also in compression, optimal coding table, which depends on the frequencies, should be stored, and adds to the file size. One could see the histogram as a simple form of a model that is inferred from the data. The model generally forms part of the information content.

In the second case, temporal dependencies reduce the average information content per observation. Also, when the forms of the temporal dependencies are not known a priori, but are inferred from the data, they can decrease the information content if the gain in compression offsets the space needed to store the model describing the dependencies. In the theoretical framework of algorithmic information theory, model and data are unified in one algorithm (one could see it as a self-extracting archive) and the length of the shortest algorithm that reproduces the data is the information content, or Kolmogorov complexity (Kolmogorov, 1968).

Flexible data compression algorithms, such as those used in this paper, are able to give an upper bound for the information content of hydrological data because they are not specifically tuned towards hydrological data. All patterns inferred from the data set are stored in the compressed file, and very little is considered as prior information. Theoretically, prior information can be explicitly fed to new compression algorithms in the form of auxiliary data files (e.g., rainfall to compress runoff) or function libraries (e.g., hydrological models), which should reduce information content of the data set due to the increase in prior knowledge (Weijs et al., 2013b).

To summarize, we can state that information content of a data set depends on (1) what question we ask of the data, and (2) how much is already known about the answer before seeing the data.

5.2 Aleatoric and epistemic uncertainty

In the current hydrological literature, attempts are sometimes made to separate epistemic (due to incomplete knowledge of the process) from aleatoric (the "inherent" randomness in the system) uncertainty (Montanari et al., 2009; Gong et al., 2013). The approach to answering this question is equivalent to trying to separate pattern from scatter (signal from noise) in high-dimensional data spaces to see how much of the variability can potentially be explained by any model.

However, the inherent problem in answering this question is the subjectivity of what we call pattern and what we call scatter. The remaining uncertainty in discharge given rainfall, $H(Q|P)$ (the aleatoric uncertainty), can be made arbitrarily small by choosing an extremely fine quantization and calculating H based on a corresponding joint histogram. It is important to realize that such a histogram is a model, and any smoothing or dimensionality reduction method used is also a model, so in principle no assessment of mutual information is model-free. Although model complexity control methods can give guidelines on how much pattern can be reasonably inferred from a data set, they usually do not account for prior knowledge. This prior knowledge may affect to a large degree what is considered a pattern – for example, by constraining the model class that is used to search for patterns or by introducing knowledge of underlying physics. In the algorithmic information theory sense, the prior knowledge can be expressed in the use of a specific computer language that offers a shorter program description for that specific pattern. Prior knowledge is then contained in the code, data and libraries available to the compiler for the language. An analogy in hydrology would be to have, e.g., a digital elevation model available, or the principle of mass balance, which a hydrological model can use but is considered as a truth not inferred from the current data set, and hence should not be considered part of the complexity of the explanation of those data.

As a somewhat extreme, unlikely but illustrative example of the subjectivity of randomness, consider that we encounter 100 consecutive digits of π as a streamflow time series. Our prior hydrological knowledge would indicate those values as random, and containing a large amount of information (no internal dependence or predictability). With different prior knowledge, however, for example that the data set is the output of a computer program authored by a student, we would consider the data set as having a pattern, and could use this to make predictions or compress the data set (by inferring one of the possible programs that enumerate digits of π as a probable source of it). There would be little surprise in the second half of the data set, given the first, and information content is drastically reduced.

6 Conclusions

Determining information content of a data set is a similar process to building a model of the data or compressing the data. These processes are subject to prior knowledge, and therefore this knowledge should be explicitly considered in determining information content. Quantization of the data can be seen as a formulation of the question the data set is asked to give information about. Upper bounds for information content for that question can then be found using compression algorithms on the quantized data.

A hydrological model actually is such a compression tool for a hydrological data set. It makes use of the dependencies between, for example, rainfall and streamflow. The patterns that are already present in the rainfall and runoff individually reduce the information that the hydrological model can learn from: a long dry period could, for example, be summarized by encoding the dry spell length, or one parameter for an exponential recession curve in the streamflow. The information content of individual series are estimated in this paper by compression algorithms, and compression of model errors was tried as a first approach to estimate joint information content.

6.1 Future work

A more comprehensive framework for joint compression of hydrological input and output data will be addressed in future research. The information available for a rainfall–runoff model to explain could theoretically be estimated by comparing the file size of compressed rainfall plus the file size of compressed streamflow with the size of a file where rainfall and streamflow are compressed together, exploiting their mutual dependencies. We could denote this as

$$\text{learnable info} = |\text{ZIP}(P)| + |\text{ZIP}(Q)| - |\text{ZIP}(P,Q)|, \quad (11)$$

where $|\text{ZIP}(X)|$ stands for the file size of a theoretically optimal compression of data set X, which includes the size of the decompression algorithm. This brings us back to the ideas of algorithmic information theory, which uses lengths of programs that reproduce data sets on computers (Turing machines). The shortening in description length when merging input and output data, i.e., the compression progress, could be seen as the amount of information learned by modeling, or the number of observations replaced by a "law". One could figuratively say "a law is worth a thousand data points". The hydrological model that is part of the decompression algorithm embodies the knowledge gained from the data. The expression $|\mathrm{ZIP}(P, Q)|$ can be seen as the AIT formulation of what Gong et al. (2013) call the aleatoric (irreducible) uncertainty, which can now also be interpreted as the incompressible minimum representation of that P, Q data set. When prior knowledge is available, this may be employed to decompress without counting the the representation size, indicating that the line between epistemic and aleatoric uncertainty may be drawn differently in the presence of prior knowledge.

Further explorations of these ideas from algorithmic information theory are expected to put often-discussed issues in hydrological model inference in a wider perspective with more general and robust foundations. This is important since every choice in model building, data analysis, data collection, model calibration, data assimilation and prediction involves implicit assumptions on prior knowledge and the information content of the data set. The information-theoretical framework can serve to make these choices more explicit.

Appendix A

Correspondence of resolution–reliability–uncertainty decomposition to compression and structure

In this appendix, we give a data-compression interpretation of Kullback–Leibler divergence as a forecast skill score and its decomposition into uncertainty, reliability and resolution, as proposed in Weijs et al. (2010b,a). For definitions of the terminology used in this appendix, the reader is referred to those papers, and scripts available at divergence.wrm.tudelft.nl. As noted in Sect. 2.1, when observations have distribution $\boldsymbol{p}$, but an optimal fixed dictionary is chosen assuming the distribution is $\boldsymbol{q}$, the expected average word length per observation, $\bar{L}$, is given by

$$\bar{L} = H(\boldsymbol{p}) + D_{\mathrm{KL}}(\boldsymbol{p}||\boldsymbol{q}). \tag{A1}$$

The code length is related to the remaining uncertainty, i.e., the missing information – the amount of information that remains to be specified to reproduce the data. In terms of forecast evaluation and the decomposition presented in Weijs et al. (2010b), using the same notation, this remaining uncertainty is the divergence score associated with a forecast with zero resolution (forecasts do not change), and non-zero reliability (forecast distribution $\boldsymbol{f}$ is not equal to climatological distribution $\bar{\boldsymbol{o}}$):

$$\bar{L} = \mathrm{DS} = H(\bar{\boldsymbol{o}}) + D_{\mathrm{KL}}(\bar{\boldsymbol{o}}||\boldsymbol{f}) = \mathrm{UNC} + \mathrm{REL}. \tag{A2}$$

The resolution term, given by the Kullback–Leibler divergence from the marginal distribution $\bar{\boldsymbol{o}}$ to the conditional distributions of observations $\bar{\boldsymbol{o}}_k$, given forecast $\boldsymbol{f}_k$,

$$\mathrm{RES} = D_{\mathrm{KL}}(\bar{\boldsymbol{o}}_k||\bar{\boldsymbol{o}}), \tag{A3}$$

is zero since $\bar{\boldsymbol{o}}_k = \bar{\boldsymbol{o}}$ for an unconditioned, constant forecast (code for compression).

When a data set with temporal dependencies is compressed, a lower average code length per observation can be achieved since we can use a dynamically changing coding for next observations, depending on the previous ones. In terms of forecast quality, this means that the individual probability estimates now have non-zero resolution. This resolution, which is equivalent to the mutual information between the forecast based on the past time series and the value to code, will reduce the average code length per observation. Since also the individual forecasts will not be completely reliable, the average code length per observation will now have a contribution from each term in the decomposition of the divergence score

$$\begin{aligned}\bar{L} &= H(\bar{\boldsymbol{o}}) + \sum_{k=1}^{K} \frac{n_k}{N}\left[D_{\mathrm{KL}}(\bar{\boldsymbol{o}}||\boldsymbol{f}_k) - D_{\mathrm{KL}}(\bar{\boldsymbol{o}}_k||\bar{\boldsymbol{o}})\right] \\ &= \mathrm{UNC} + \mathrm{REL} - \mathrm{RES},\end{aligned} \tag{A4}$$

where n_k is the number of observations for which unique forecast number k is given and N is the total number of observations. When compressing data, however, the prediction model that describes the temporal dependence needs to be stored as well. Therefore, the average total code length per data point will become

$$\bar{L} = \mathrm{UNC} + \mathrm{REL} - \mathrm{RES} + |\mathrm{model}|/N, \tag{A5}$$

where |model| is the description length of the model algorithm, i.e., model complexity. Although this model length is language dependent, it is known from AIT that this dependence is just an additive constant, and can be interpreted as the prior knowledge encoded in the language. If the language is not specifically geared towards a certain type of data, the total code length will give a fairly objective estimate of the amount of new information in the data set, which cannot be explained from the data set itself. The number of bits per symbol needed to store data set can therefore be interpreted as a complexity-penalized version of the divergence score presented in Weijs et al. (2010a,b), applied to a "self-prediction" of the data based on previous time steps. We can make the following observations. Firstly, a data set can only be compressed if there is a pattern – i.e., something that can be described be an algorithm where the resolution or gain in description efficiency or predictive power outweighs the loss due to complexity. Secondly, the data compression view naturally leads to the notion that we have to penalize model complexity when evaluating the predictive performance of models.

Supplementary material related to this article is available online at: http://www.hydrol-earth-syst-sci.net/17/3171/2013/hess-17-3171-2013-supplement.zip.

Acknowledgements. Steven Weijs is a beneficiary of a postdoctoral fellowship from the AXA research fund, which is gratefully acknowledged. Funding from the Swiss National Science Foundation, the NCCR-MICS and CCES programs are also gratefully acknowledged.

Edited by: E. Gargouri-Ellouze

References

Akaike, H.: A new look at the statistical model identification, IEEE Trans. Automatic Control, 19, 716–723, 1974.

Alfonso, L., Lobbrecht, A., and Price, R.: Information theory–based approach for location of monitoring water level gauges in polders, Water Resour. Res., 46, W03528, doi:10.1029/2009WR008101, 2010a.

Alfonso, L., Lobbrecht, A., and Price, R.: Optimization of water level monitoring network in polder systems using information theory, Water Resour. Res., 46, W12553, doi:10.1029/2009WR008953, 2010b.

Beven, K. and Westerberg, I.: On red herrings and real herrings: disinformation and information in hydrological inference, Hydrol. Process., 25, 1676–1680, doi:10.1002/hyp.7963, 2011.

Beven, K., Smith, P. J., and Wood, A.: On the colour and spin of epistemic error (and what we might do about it), Hydrol. Earth Syst. Sci., 15, 3123–3133, doi:10.5194/hess-15-3123-2011, 2011.

Brutsaert, W.: Hydrology: an introduction, Cambridge University Press, New York, 2005.

Burrows, M. and Wheeler, D. J.: A block-sorting lossless data compression algorithm, Tech. rep., Systems Research Center, Palo Alto, CA, 1994.

Cahill, A. T.: Significance of {AIC} differences for precipitation intensity distributions, Adv. Water Resour., 26, 457–464, doi:10.1016/S0309-1708(02)00167-7, 2003.

Chaitin, G. J.: On the length of programs for computing finite binary sequences, J. ACM, 13, 547–569, 1966.

Chaitin, G. J.: A theory of program size formally identical to information theory, J. ACM, 22, 329–340, 1975.

Cilibrasi, R.: Statistical inference through data compression, Ph.D. thesis, UvA, Amsterdam, 2007.

Cover, T. M. and Thomas, J. A.: Elements of information theory, Wiley-Interscience, New York, 2006.

Feynman, R.: The character of physical law, MIT Press, 1967.

Gong, W., H. V. Gupta, D. Yang, K. Sricharan, and A. O. Hero III , Estimating epistemic and aleatory uncertainties during hydrologic modeling: An information theoretic approach, Water Resour. Res., 49, 2253–2273, doi:10.1002/wrcr.20161, 2013.

Huffman, D. A.: A Method for the Construction of Minimum-Redundancy Codes, Proceedings of the IRE, 40, 1098–1101, 1952.

Jakeman, A. J. and Hornberger, G. M.: How much complexity is warranted in a rainfall-runoff model?, Water Resour. Res., 29, 2637–2649, 1993.

Katz, R.: On some criteria for estimating the order of a Markov-chain, Technometrics, 23, 243–249, doi:10.2307/1267787, 1981.

Kolmogorov, A. N.: Three approaches to the quantitative definition of information, Int. J. Comput. Math., 2, 157–168, 1968.

Kraft, L. G.: A device for quantizing, grouping, and coding amplitude-modulated pulses, Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1949.

Kullback, S. and Leibler, R. A.: On Information and Sufficiency, Ann. Math. Stat., 22, 79–86, 1951.

Laio, F., Allamano, P., and Claps, P.: Exploiting the information content of hydrological "outliers" for goodness-of-fit testing, Hydrol. Earth Syst. Sci., 14, 1909–1917, doi:10.5194/hess-14-1909-2010, 2010.

Li, C., Singh, V., and Mishra, A.: Entropy theory-based criterion for hydrometric network evaluation and design: Maximum information minimum redundancy, Water Resour. Res., 48, W05521, doi:10.1029/2011WR011251, 2012.

Li, M. and Vitanyi, P. M. B.: An introduction to Kolmogorov complexity and its applications, Springer-Verlag New York Inc, 2008.

Martin, G. N. N.: Range encoding: an algorithm for removing redundancy from a digitised message, in: Video & Data Recording conference, Southampton, UK, 1979.

McMillan, B.: Two inequalities implied by unique decipherability, IEEE Trans. Inf. Theory, 2, 115–116, 1956.

Mishra, A. and Coulibaly, P.: Hydrometric network evaluation for Canadian watersheds, J. Hydrol., 380, 420–437, 2010.

Montanari, A., Shoemaker, C. A., and van de Giesen, N.: Introduction to special section on Uncertainty Assessment in Surface and Subsurface Hydrology: An overview of issues and challenges, Water Resour. Res., 45, W00B00, doi:10.1029/2009WR008471, 2009.

Paluš, M.: Coarse-grained entropy rates for characterization of complex time series, Physica D, Nonlinear Phenomena, 93, 64–77, 1996.

Pianosi, F. and Soncini-Sessa, R.: Real-time management of a multipurpose water reservoir with a heteroscedastic inflow model, Water Resour. Res., 45, W10430, doi:10.1029/2008WR007335, 2009.

Rissanen, J. and Langdon, G. G.: Arithmetic coding, IBM J. Res. Develop., 23, 149–162, 1979.

Ruddell, B. L., Brunsell, N. A., and Stoy, P.: Applying Information Theory in the Geosciences to Quantify Process Uncertainty, Feedback, Scale, Eos, Transactions American Geophysical Union, 94, 56–56, doi:10.1002/2013EO050007, 2013.

Schaefli, B. and Gupta, H. V.: Do Nash values have value?, Hydrol. Process., 21, 2075–2080, 2007.

Schoups, G., van de Giesen, N. C., and Savenije, H. H. G.: Model complexity control for hydrologic prediction, Water Resour. Res., 44, W00B03, doi:10.1029/2008WR006836, 2008.

Shannon, C. E.: A mathematical theory of communication, Bell System Technical J., 27, 379–423, 1948.

Singh, V. P.: The use of entropy in hydrology and water resources, Hydrol. Process., 11, 587–626, 1997.

Singh, V. P. and Rajagopal, A. K.: Some recent advances in application of the principle of maximum entropy (POME) in hydrology, IAHS, 194, 353–364, 1987.

Solomonoff, R. J.: A formal theory of inductive inference, Part I, Information Control, 7, 1–22, 1964.

Turing, A. M.: On computable numbers, with an application to the Entscheidungsproblem, Proceedings of the London Mathematical Society, 2, 230–265, 1937.

Vrugt, J. A., Bouten, W., Gupta, H. V., and Sorooshian, S.: Toward improved identifiability of hydrologic model parameters: The information content of experimental data, Water Resour. Res, 38, 1312, doi:10.1029/2001WR001118, 2002.

Vrugt, J. A., Gupta, H. V., Bouten, W., and Sorooshian, S.: A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, Water Resour. Res., 39, 1201, doi:10.1029/2002WR001642, 2003.

Vrugt, J. A., Ter Braak, C. J. F., Gupta, H. V., and Robinson, B. A.: Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling?, Stochastic Environ. Res. Risk Assess., 23, 1011–1026, 2009.

Weijs, S. V. and Van de Giesen, N.: Accounting for observational uncertainty in forecast verification: an information–theoretical view on forecasts, observations and truth, Mon. Weather Rev., 139, 2156–2162, doi:10.1175/2011MWR3573.1, 2011.

Weijs, S. V., Schoups, G., and van de Giesen, N.: Why hydrological predictions should be evaluated using information theory, Hydrol. Earth Syst. Sci., 14, 2545–2558, doi:10.5194/hess-14-2545-2010, 2010a.

Weijs, S. V., Van Nooijen, R., and Van de Giesen, N.: Kullback–Leibler divergence as a forecast skill score with classic reliability–resolution–uncertainty decomposition, Mon. Weather Rev., 138, 3387–3399, 2010b.

Weijs, S. V., Mutzner, R., and Parlange, M. B.: Could electrical conductivity replace water level in rating curves for alpine streams?, Water Resour. Res., 49, WR012181, doi:10.1029/2012WR012181, 2013a.

Weijs, S. V., van de Giesen, N., and Parlange, M. B.: HydroZIP: How Hydrological Knowledge can Be Used to Improve Compression of Hydrological Data, Entropy, 15, 1289–1310, doi:10.3390/e15041289, 2013b.

Westerberg, I., Guerrero, J., Seibert, J., Beven, K., and Halldin, S.: Stage-discharge uncertainty derived with a non-stationary rating curve in the Choluteca River, Honduras, Hydrol. Process., 25, 603–613, doi:10.1002/hyp.7848, 2011.

Ziv, J. and Lempel, A.: A universal algorithm for sequential data compression, IEEE Trans. Information Theory, 23, 337–343, 1977.