WATER RESOURCES RESEARCH, VOL. 49, 3229–3242, doi:10.1002/wrcr.20146, 2013

Monthly river flow simulation with a joint conditional density estimation network

Chao Li,[1] Vijay P. Singh,[1,2] and Ashok K. Mishra[3]

Received 20 April 2012; revised 9 February 2013; accepted 12 February 2013; published 6 June 2013.

[1] River flow synthesizing and downscaling are required for the analysis of risks associated with water resources management plans and for regional impact studies of climate change. This paper presents a probabilistic model that synthesizes and downscales monthly river flow by estimating the joint distribution of flows of two adjacent months conditional on covariates. The covariates may consist of lagged and aggregated flow variables (synthesizing), exogenous climatic variables (downscaling), or combinations of these two types. The joint distribution is constructed by connecting two marginal distributions in terms of copulas. The relationship between covariates and distribution parameters is approximated by an artificial neural network, which is calibrated using the principle of maximum likelihood. Outputs of the neural network yield parameters of the joint distribution. From the estimated joint distribution, a conditional distribution of river flow of current month given the estimation of the previous month can be derived. Depending on the different types of covariate information, this conditional distribution may serve as the "engine" for synthesizing or downscaling river flow sequences. The idea of the proposed model is illustrated using three case studies. The first case deals with synthetic data and shows that the model is capable of fitting a nonstationary joint distribution. Second, the model is utilized to synthesize monthly river flow at four sample stations on the main stream of the Colorado River. Results reveal that the model reproduces essential evaluation statistics fairly well. Third, a simple illustrative example for river flow downscaling is presented. Analysis indicates that the model can be a viable option to downscale monthly river flow as well.

Citation: Li, C., V. P. Singh, and A. K. Mishra (2013), Monthly river flow simulation with a joint conditional density estimation network, *Water Resour. Res.*, *49*, 3229–3242, doi:10.1002/wrcr.20146.

1. Introduction

[2] Generating synthetic river flow sequences is required for the estimation of risks associated with water resources management plans [*Sharma and O'Neill*, 2002]. Downscaling general circulation model outputs to flows at site or river basin scale is gaining importance in assessing regional impacts of climate change [*Tisseuil et al*., 2010]. From the viewpoint of stochastic hydrology, river flow synthesizing and downscaling can be generalized as an exercise in the conditional distribution of $\mathbf{Y}|\mathbf{X}$, where $\mathbf{Y}$ is a random vector representing river flows (e.g., flows of two adjacent months), and $\mathbf{X}$ is a random vector representing covariates. The covariate vector $\mathbf{X}$ may be lagged and aggregated flow variables for river flow synthesizing or may be exogenous climatic variables for downscaling. Hereafter, upper case bold letters (e.g., $\mathbf{Y}$) denote random vectors and the corresponding lower case letters (e.g., $\mathbf{y}$) their values.

[3] River flow synthesizing is to generate flow sequences that are statistically consistent with historical records. To that end, several approaches have been developed in a parametric or nonparametric framework. A commonly used parametric method is the autoregressive moving average (ARMA) model, wherein river flow is assumed to be normally distributed [*Lettenmaier and Burges*, 1977; *Salas and Delleur*, 1980]. The normality assumption, nevertheless, is problematic since flow data are right-skewed and very often even heavy-tailed [*Bernadara et al*., 2008; *Carreau et al*., 2009]. To render the data to be normal, transformation techniques are required, which consequently give rise to other drawbacks [*Hao and Singh*, 2011]. To sidestep data transformation, the ARMA model with gamma distribution was proposed by *Fernandez and Salas* [1990]. Besides parametric models, nonparametric alternatives, which are often based on bootstrap and/or kernel density estimation, have addressed parametric limitations [*Lall and Sharma*, 1996; *Sharma et al*., 1997; *Sharma and O'Neill*, 2002; *Salas and Lee*, 2010]. Nonparametric models allow the data to speak for themselves [*Wand and Jones*, 1995;

[1]Department of Biological & Agricultural Engineering, Texas A&M University, College Station, Texas, USA.
[2]Department of Civil & Environmental Engineering, Texas A&M University, College Station, Texas, USA.
[3]Pacific Northwest National Laboratory, Richland, Washington, USA.

Corresponding author: C. Li, Department of Biological & Agricultural Engineering, Texas A&M University, College Station, TX 77843-2117, USA. (lichsunny@gmail.com)

0043-1397/13/10.1002/wrcr.20146

Sharma and O'Neill, 2002]. Therefore, the conditional distribution of $\mathbf{Y}|\mathbf{X}$ can be derived without either the assumption of normality or resorting to normal transformation. Admiring successes of nonparametric approaches, their application requires large sample size [*Evin et al.*, 2011].

[4] River flow downscaling aims to bridge the gap between large-scale climatic variables ($>10^4$ km^2) and flows at site or river basin scale. River flow downscaling is typically accomplished in terms of statistical methods, which have the advantage of being flexible in modeling and tractable in computation. Broadly speaking, statistical downscaling models can be classified into three categories: (1) regression functions, (2) weather generators, and (3) weather typing schemes. Weather typing usually serves as a preprocessor before building regression functions or weather generators [*Huth et al.*, 2008; *Vrac et al.*, 2007; *Carreau and Vrac*, 2011]. Regression functions seek to estimate the conditional mean $E[\mathbf{Y}|\mathbf{X}=\mathbf{x}]$ of small-scale variables (e.g., river flows) given large-scale climatic covariates $\mathbf{X}=\mathbf{x}$, like the work of *Schoof and Pryor* [2001], *Cannon and Whitfield* [2002], *Ghosh and Mujumdar* [2008], and *Tisseuil et al.* [2010] among many others. In weather generators, small-scale values are not estimated by $E[\mathbf{Y}|\mathbf{X}=\mathbf{x}]$ but rather generated from a conditional distribution of $\mathbf{Y}|\mathbf{X}$ structured in a parametric [*Wilks*, 1999], nonparametric [*Rajagopalan and Lall*, 1999; *Sharma*, 2000; *Mehrotra and Sharma*, 2007a, 2007b], or semiparametric way [*Cannon*, 2008; *Carreau and Vrac*, 2011]. It is acknowledged that downscaled results are subject to uncertainties [*Mujumdar and Ghosh*, 2008]. A reasonable estimate of uncertainty in hydrologic prediction is valuable in water resources and other relevant decision-making processes [*Liu and Gupta*, 2007]. To evaluate uncertainty, one is less interested in a crisp projected value $E[\mathbf{Y}|\mathbf{X}=\mathbf{x}]$ than in a conditional distribution of $\mathbf{Y}|\mathbf{X}$ [*Hsieh*, 2009]. For this reason, weather generator-based downscaling methods are preferred. As the name suggests, weather generator-based models are indeed adapted weather (river flow) generators. In this sense, synthesizing and downscaling flow sequences are virtually the same. The only difference resides in that the covariate vector $\mathbf{X}$ for river flow synthesizing comprises variables created from the historical observations, whereas for downscaling, exogenous large-scale climatic variables are included.

[5] Redefining $\mathbf{Y}$ as a random vector of flows of two adjacent months, i.e., $\mathbf{Y}=[Y_{t-1}, Y_t]$ (in the boundary season, Y_{t-1} denotes the flow in December of the previous year, and Y_t represents that in January of the current year), this paper primarily attempts to model the joint density $\phi(y_{t-1}, y_t|\mathbf{x})$ conditioned on covariate information $\mathbf{x}$. Right-skewed distributions with positive support, such as lognormal and gamma distributions, are deemed as plausible choices for representing monthly river flow [*Sangal and Biswas*, 1970; *Fernandez and Salas*, 1990; *Nadarajah*, 2007]. The dependence structure of adjacent flows is captured using copulas [*Joe*, 1997; *Nelsen*, 2006; *Lee and Salas*, 2011]. Parameters of the joint density are approximated as functions of covariates via a neural network, which is calibrated by the principle of maximum likelihood (ML) [*Bishop*, 1995; *Cawley et al.*, 2007; *Cannon*, 2008; *Carreau et al.*, 2009; *Carreau and Vrac*, 2011]. The joint density $\phi(y_{t-1}, y_t|\mathbf{x})$ is allowed to evolve as the covariate information changes. We will hereinafter refer to this framework as the joint conditional density network (JCDN). From $\phi(y_{t-1}, y_t|\mathbf{x})$, a new conditional distribution of Y_t given Y_{t-1} can be derived. Random numbers are then sequentially simulated from this distribution as synthesized and downscaled river flows.

[6] A similar modeling framework has been employed for precipitation downscaling [*Cawley et al.*, 2007; *Cannon*, 2008; *Carreau and Vrac*, 2011] and rainfall-runoff simulation [*Carreau et al.*, 2009]. The major difference resides in that only precipitation or runoff at current time step is modeled instead of that at adjacent time lags. Yet these univariate models tend to result in inadequate serial correlated sequences, as pointed out by *Cannon* [2008] and *Carreau and Vrac* [2011]. To address this issue, *Cannon* [2008] recommended guiding the simulation by a latent Gaussian process, which has to be determined by a trial and error procedure. JCDN explicitly takes the first-order Markovian dependence into account. As a consequence, the lag-1 autocorrelation of river flows is preserved.

[7] This paper proceeds as follows. Definition of the proposed JCDN generator is described in section 2. Section 3 discusses how to identify a proper JCDN generator. Section 4 discusses the major numerical algorithms involved in the implementation of JCDN, followed by three applications in section 5: one is for demonstrating the ideas of JCDN on synthetic data, and the others are for monthly river flow synthesizing and downscaling, respectively. Conclusions are generalized in section 6.

2. Model Definition

2.1. Joint Conditional Density Network

[8] Inspired by the work of *Cawley et al.* [2007], *Cannon* [2008], *Carreau et al.* [2009], and *Carreau and Vrac* [2011], the idea of JCDN is to let an artificial neural network estimate parameters of the joint probability density of river flows of two adjacent months conditioned upon a set of covariates. To that end, an appropriate joint distribution family should be specified first. The copula theory is used to build the joint distribution by linking together its marginal distributions. The major benefit of using copulas to construct the joint distribution is reflected by its flexibility in describing diverse dependence structures and relaxing marginal restrictions.

[9] In order to properly capture the behavior of adjacent monthly flows, several univariate distributions and copula families are selected as candidates. Lognormal and gamma families are selected as alternatives for representing the probability distribution of monthly river flow. Both of them have been frequently used for river flow simulation [*Sangal and Biswas*, 1970; *Fernandez and Salas*, 1990; *Nadarajah*, 2007]. Three copula families, i.e., Clayton, survival Clayton, and Gaussian, are chosen as options to capture the serial dependence.

[10] Following Sklar's theorem [*Joe*, 1997; *Nelsen*, 2006], the joint conditional density $\phi(y_{t-1}, y_t|\mathbf{x})$ of river flows of two adjacent months can be expressed as

$$\phi(y_{t-1}, y_t|\mathbf{x}) = f(y_{t-1}|\mathbf{x})g(y_t|\mathbf{x})c(F(y_{t-1}|\mathbf{x}), G(y_t|\mathbf{x})|\mathbf{x}), \quad (1)$$

in which $f(\cdot|\mathbf{x})$ and $g(\cdot|\mathbf{x})$ are the marginal probability density functions (PDFs) of flows of months $t-1$ and t

conditioned upon covariates $\mathbf{x}$, respectively; $F(\cdot|\mathbf{x})$ and $G(\cdot|\mathbf{x})$ are the corresponding cumulative distribution functions (CDFs); and $c(\cdot|\mathbf{x})$ is the mixed partial derivative of the copula $C(\cdot|\mathbf{x})$. The joint distribution is thought of as conditional in the sense that its parameters depend on the covariate vector $\mathbf{X}=\mathbf{x}$. It may be noted that the covariate vector may consist of any variable as long as it is a driving factor of the monthly river flow process. Examples may include lagged flow variables, large-scale climatic variables, monthly mean precipitation, temperature, and date variables encoded seasonal cycles, to mention a few. In practice, one can specify the variables that should be included in $\mathbf{X}$ according to their purposes, as will be shown later.

[11] A standard one-hidden-layer perceptron neural network was used to approximate the covariate-parameter relationship. Outputs of the neural network were interpreted as parameters of the joint conditional distribution. Let n be the number of hidden neurons. For each hidden neuron, its output a_h^i is computed as

$$a_h^i=\tanh\left(\sum_{d=1}^{D}x_d^i w_{dh}+b_h\right),\quad h=1,2,...,n \tag{2}$$

where x_d^i represents values of the dth covariate; w_{dh} and b_h are the input-hidden-layer weights and offset parameters, respectively. Suppose there are K parameters in the specified joint distribution, then the kth parameter is estimated as

$$\theta_k^i=\hbar_k\left(\sum_{h=1}^{n}a_h^i\tilde{w}_{hk}+\tilde{b}_k\right),\quad k=1,2,...,K \tag{3}$$

where $\tilde{w}_{hk}$ and $\tilde{b}_k$ correspondingly denote the hidden-output-layer weights and offset parameters; $\hbar_k(\cdot)$ is a transfer function to project the linear combination term in equation (3) onto the parameter space of the kth parameter. Functions $\hbar_k(\cdot)$ ($k=1, 2, \ldots, K$) are determined by the specified distribution. For example, for a gamma distribution with shape parameter α and scale parameter β, since both of them should be positive, $\hbar_k(\cdot)$ ($k=1, 2$) will be exponential functions.

[12] The advantages of using a neural network to approximate the covariate-parameter relationship include the following: (1) it is flexible for both linear and nonlinear modeling; (2) it is capable of accounting for complex interactions among covariates [*Cannon*, 2008; *Carreau et al.*, 2009; *Hsieh*, 2009; *Carreau and Vrac*, 2011]; (3) it is apt at incorporating different covariates as many as necessary; and (4) it allows the joint PDF to change over time.

2.2. JCDN-Based River Flow Generator

[13] From the joint distribution of river flows of each paired adjacent months (e.g., January–February and February–March), a conditional CDF of Y_t given $Y_{t-1}=y_{t-1}$ can be expressed as

$$\Phi_{Y_t|Y_{t-1}}(y_t|y_{t-1},\mathbf{x})=c_1(F(y_{t-1}|\mathbf{x}),G(y_t|\mathbf{x})|\mathbf{x}), \tag{4}$$

where $c_1(\cdot)$ is the partial derivative of copula $C(\cdot)$ with respect to its first argument [*Zhang and Singh*, 2007]. This conditional CDF is the "engine" used to generate river flow at month t given the value of its previous month by entering a uniform random number p into the quantile function

$$y_t(p|y_{t-1},\mathbf{x})=G^{-1}(c_{1_2}^{-1}(F(y_{t-1}|\mathbf{x}),p|\mathbf{x})|\mathbf{x}), \tag{5}$$

where $G^{-1}(\cdot)$ is the inverse of the CDF $G(\cdot)$, and $c_{1_2}^{-1}(\cdot)$ is the inverse of $c_1(\cdot)$ with respect to its second argument.

[14] The step-by-step chain-dependent generation procedure may (1) start at any given month, for instance, the first month of historical observations, which is denoted as t, by selecting any random number (or the mean, median) from the marginal distribution of river flow of month t; (2) then proceed to the next month, denoted as $t+1$, by evaluating the quantile function (equation (5)) at the generated flow of month t; and (3) then continue by repeating step 2 until flow for a given month, say, the last month of the historical period, is generated. To diversify scenarios of synthetic data, multiple sequences are usually required. In this situation, the generation procedure might be repeated as many times as necessary.

[15] The major advantage of JCDN-based generator over its univariate analogues is that JCDN explicitly takes the first-order Markovian dependence into account. It follows that the short-term persistence of river flow is preserved in the generated sequences.

3. Model Identification

3.1. JCDN Parameter Identification

[16] Weights and offset parameters are adjustable terms in JCDN and are calibrated following the principle of ML, which is by far the most popular technique for deriving distribution parameter estimates [*Casella and Berger*, 2001]. Often, it is more convenient to work with the log-likelihood function instead. Given covariates $\mathbf{x}=[x_1,x_2,...,x_D]$ ($N\times D_\mathrm{N}$ matrix) and flow observations of two adjacent months $\mathbf{y}=[y_{t-1},y_t]$ ($N\times 2$ matrix), the log-likelihood function based on the specified distribution is expressed as

$$\mathrm{LL}=\mathrm{LL}_f+\mathrm{LL}_g+\mathrm{LL}_c \tag{6}$$

$$\mathrm{LL}_f=\sum_{i=1}^{N}\log\left(f\left(y_{t-1}^i|\mathbf{\Theta}_f\left(\mathbf{x}^i\right)\right)\right) \tag{7a}$$

$$\mathrm{LL}_g=\sum_{i=1}^{N}\log\left(g\left(y_t^i|\mathbf{\Theta}_g\left(\mathbf{x}^i\right)\right)\right) \tag{7b}$$

$$\mathrm{LL}_c=\sum_{i=1}^{N}\log\left(c\left(F\left(y_{t-1}^i|\mathbf{\Theta}_f\left(\mathbf{x}^i\right)\right),G\left(y_t^i|\mathbf{\Theta}_g\left(\mathbf{x}^i\right)\right)|\theta\left(\mathbf{x}^i\right)\right)\right), \tag{7c}$$

in which N is the number of observations; $\mathbf{\Theta}_f$ and $\mathbf{\Theta}_g$ are the parameter vectors of marginal distributions of river flows (e.g., suppose $f(\cdot)$ is a gamma density with shape parameter α and scale parameter β, $\mathbf{\Theta}_f$ will be $[\alpha,\beta]$); and θ is the copula parameter. Of particular note, $\mathbf{\Theta}_f(\cdot)$, $\mathbf{\Theta}_g(\cdot)$, and $\theta(\cdot)$ are used to emphasize that these parameters are functions of covariates. Our simulation experiments (results not shown) illustrate that the ML method performs reasonably well as long as (1) the data are from the specified parametric family, or at least approximately so, and (2) the sample is sufficiently large.

[17] Equation (6) implies that parameters of the marginal distributions and the copula function can be estimated separately by three different neural networks. In this case, the objective function (log likelihood) of each neural network becomes simple, which will in turn make the calibration easy to converge. It should be noted that this benefit is at the expense of an increased overall model complexity. For the sake of a parsimonious model, here we estimate the joint distribution using one neural network. This treatment can also provide adequate goodness of fit with the aid of a novel optimization algorithm, as will be seen in section 5.

3.2. JCDN Hyperparameter Identification

[18] Families of the marginal distributions and the copula function are two of hyperparameters of the JCDN-based generator. In addition, the number of hidden neurons, which controls the generalization ability of the neural network to the underlying covariate-parameter relationship, is another hyperparameter. Hyperparameters of JCDN may be determined via cross-validation, which can efficiently avoid overfitting and underfitting, and hence identify a parsimonious model with decent generalization ability.

[19] The L-fold cross-validation is implemented as follows. First, data including flow observations and covariates are split into L subsets, i.e., $S_1, S_2, \ldots, S_L$. Second, holding the subset S_1 out, JCDN is calibrated on the other subsets, parameter estimations are made on the held-out set S_1, and then the log-likelihood is evaluated on S_1. This procedure is repeated L times, rotating each time the held-out set. The best hyperparameters are chosen as those which maximize the sum of log-likelihood of each held-out set. Once the optimal hyperparameters are determined, JCDN is calibrated anew with all the data. The final model is then used for river flow synthesizing and downscaling.

[20] Note that it is reasonable to evaluate competing models based on the log likelihood evaluated on the held-out set in view of the following two facts. First, each time data in the held-out set is unseen in the calibration set [*Carreau and Vrac*, 2011]. Second, given a model fitted from the calibration set, the log likelihood evaluated on validation set can be interpreted as the likelihood that the sample is from the specified distribution. The larger the log likelihood, the more likely the sample is from the distribution.

4. Model Implementation

[21] For the implementation of the JCDN-based generator, a suite of MATLAB scripts and functions are written, which are rooted in the R package for the univariate conditional density network developed by *Cannon* [2012]. Several critical points need more description, for instance, the numerical algorithm used for maximizing the log likelihood since to some extent, alternative numerical techniques might affect the results reported in this paper.

[22] The log-likelihood function might be maximized following the procedure in *Cannon* [2012]. First, the Nelder-Mead simplex (NMS) algorithm is conducted. After a number of iterations with NMS, the obtained weights and offset parameters are sent to initialize the Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization algorithm. One inherent difficulty associated with the problem of finding the maximum of a function is that of converging to a local maximum. Avoiding local maxima might be achieved in one of the two ways, in terms of a global optimization algorithm like particle swarm optimization (PSO), or by randomly initializing the optimization a number of times and selecting the best model. We follow the first way for this study. To accomplish the PSO optimization, an extra MATLAB package is required and is accessible from http://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox. Besides PSO, another global optimization algorithm, shuffled complex evolution method (SCE-UA), developed by *Duan et al.* [1992] is a viable option too. Compared with the hybrid NMS-BFGS optimization strategy, the PSO algorithm offers more accurate parameter estimates but with increased computational burden. One point worth noting is that in order to obtain a stable JCDN, the PSO optimization is repeated 10 times, and the resulting distribution parameters are averaged, which are used as the ultimate parameter estimates of the joint distribution.

[23] One other minor but sometimes fatal pitfall, which is very often encountered by practitioners, is also worthy of more explanation. Theoretically, there are two different manners to calculate the log-likelihood function. Taking the gamma distribution, for instance, given an observation y, the log likelihood can be calculated as

$$\mathrm{LL}\,(\alpha,\beta|y)=\log\left((\Gamma(\alpha)\beta)^{-1}\left(\frac{y}{\beta}\right)^{\alpha-1}\exp\left(\frac{-y}{\beta}\right)\right), \qquad (8)$$

or as

$$\mathrm{LL}\,(\alpha,\beta|y)=(\alpha-1)\log\left(\frac{y}{\beta}\right)-\frac{y}{\beta}-\log(\Gamma(\alpha))-\log(\beta). \qquad (9)$$

[24] In recent days, many statistical software packages are available. Most, if not all of these packages, provide procedures for the computation of PDFs of basic distributions. As such, it would be natural to choose equation (8) to compute the log-likelihood, i.e., first compute the PDF, and then take its logarithm, because not only this way is straightforward but also it can take advantage of existing software resources. Yet, due to the roundoff error, the PDF of data from extreme right tail (e.g., $y = 1200$ for $\alpha = 1$ and $\beta = 1.5$) will be 0. Note that it is not uncommon to see some extremely large values appearing far away from the "bulk" of monthly river flow data. After taking the logarithm of 0, the log-likelihood will be $-\infty$, which in turn will mislead the optimization search. To circumvent this risk, a somewhat more reliable and robust approach is the second one, i.e., through equation (9).

[25] The flowchart in Figure 1 generalizes the basic steps of JCDN for monthly river flow synthesizing and downscaling.

5. Application

[26] In this section, the idea of JCDN is tested first on synthetic data and is then applied to synthesizing and

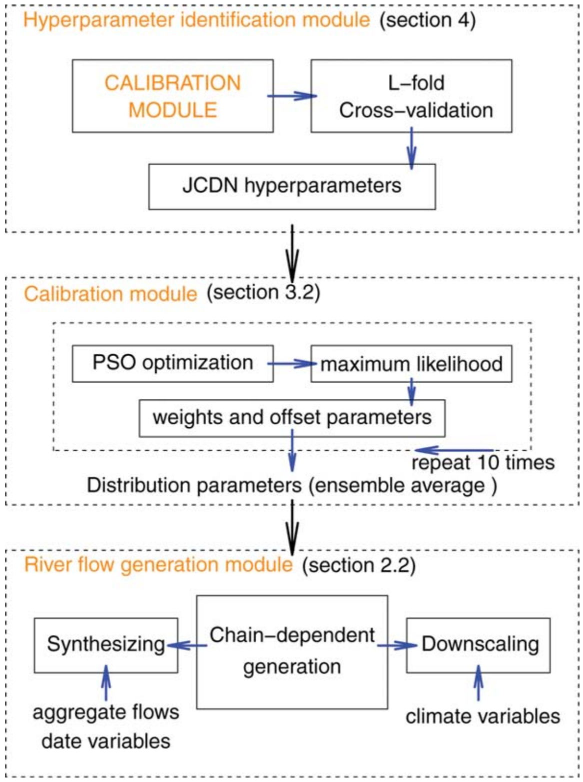

Figure 1. Basic flowchart of JCDN for monthly river flow synthesizing and downscaling.

downscaling monthly river flows at sample stations over the main stream of the Colorado River.

5.1. Case 1: Testing With Synthetic Data

[27] Case 1 is to illustrate the idea of JCDN on a synthetic data set simulated from a known bivariate distribution, which was designed such that it can mimic typical behavior of seasonal variability of monthly river flow. The Clayton copula was used to construct the parent distribution by combining two identical gamma margins. Parameters of the distribution are functions of covariate $X \in [1, 12]$, which were set up as follows. First, we obtained 51 years of monthly streamflow records from 1956 to 2006 of the Lees Ferry stream gauge in the Colorado River basin. Information about this gauge can be found elsewhere in this paper. Then, we fitted a gamma distribution for each calendar month as well as a Clayton copula for each paired adjacent months. The corresponding parameter estimates are shown by black circles in Figures 2a–2c. In the next step, these discrete points were smoothed by sine and cosine functions and their combinations, as shown by the imposed solid lines over the circles. These smoothed lines were used to represent the underlying covariate-parameter relationships. Following *Nelsen* [2006, p. 40], a sample consisting of 2500 data pairs was generated from the joint distribution with x drawn uniformly from [1, 12]. Figures 2d and 2e illustrate the distributional patterns of the sample data. One can observe a noticeable trend of the random numbers as the covariate X varies. The objective herein is to identify the underlying covariate-parameter relationship that is most likely to have generated the random sample. For that purpose, we used $\sin(X)$ and $\cos(X)$ in addition to X as covariate variables. For simplicity, in the following we assumed that the hidden neuron number was the only unknown hyperparameter to be determined.

[28] In order to identify the most parsimonious model with nice generalization ability, a simple calibration-validation experiment was carried out. First, the whole sample set was split into two parts in terms of random sampling without replacement. About two third of the data (1800) was used for model calibration and the other part (700) for validation. JCDNs with increasing number of hidden neurons were tuned on the calibration set, and each time the resulting model was evaluated on the validation set. The number of hidden neurons was increased from two to eight with a step size of one. Finally, JCDN with five hidden neurons was identified as optimal for the parent distribution. As a

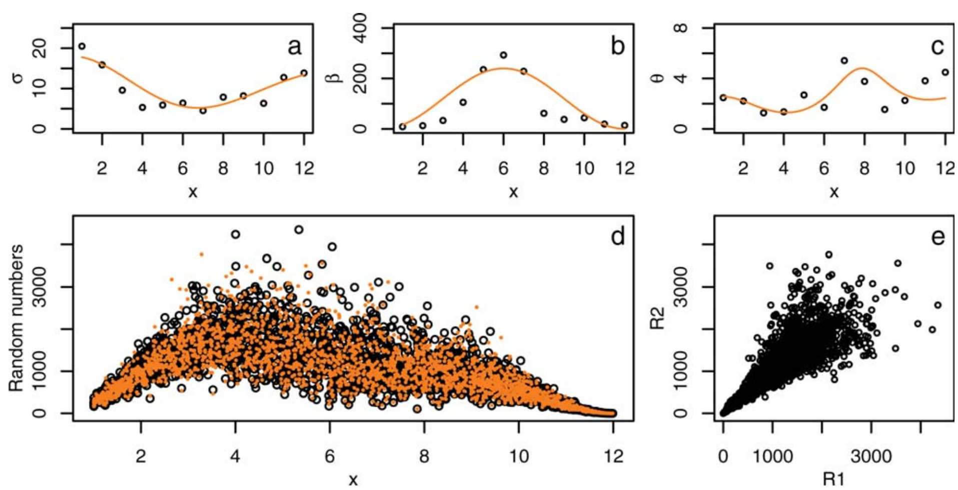

Figure 2. Parameters and representative random vectors generated from the parent distribution. (a and b) Shape and scale parameters of the gamma marginal distribution, (c) Clayton copula parameter, (d) the generated random vectors plotted as a function of covariate x, and (e) scatterplot of random sample R1 against R2.

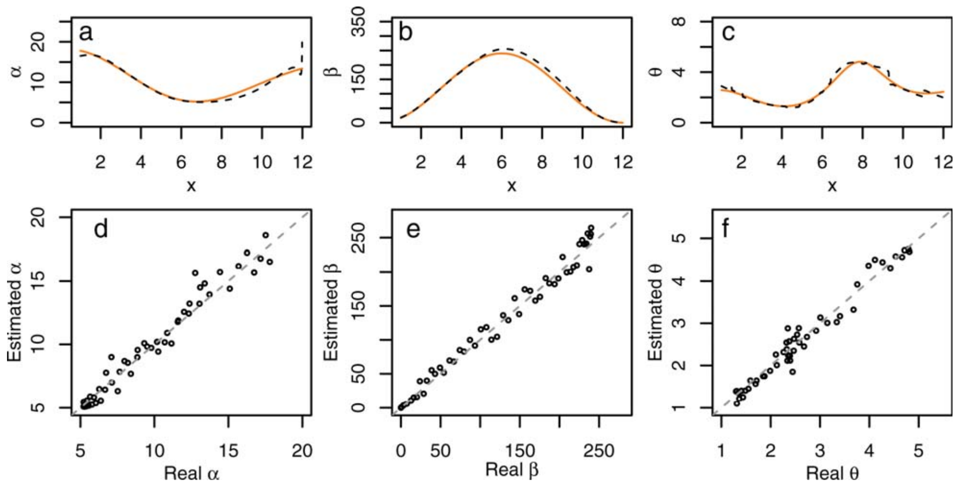

Figure 3. Calibration results: (a–c) the true (solid lines) and approximated (dashed lines) covariate-parameter relationship for each parameter of the parent distribution, and validation results: (d–f) scatterplots of the true versus estimated parameters of the parent distribution.

note, the L-fold cross-validation was not carried out mainly for the consideration of saving implementation effort. Nevertheless, later in this section we would make a double check about the reasonability of the identified hidden neuron numbers.

[29] To verify how well the JCDN with identified hidden neuron number generalizes the underlying covariate-parameter relationship, another random sample also with a size of 2500 was generated from the joint distribution. JCDN was calibrated on the sample with the identified hidden neuron number, and prediction was made on the same sample using the calibrated JCDN. The obtained covariate-parameter relationships were plotted by dashed lines in Figures 3a–3c. As can be inferred from Figures 3a–3c, in general, the modeled covariate-parameter relationships were consistent with or almost the same as the actual ones. It is noted that as was mentioned in section 4, each covariate-parameter relationship represents the ensemble average of 10 members obtained by repeatedly performing the PSO optimization 10 times.

[30] Additionally, it is needed to check if under- or overfitting occurred. First, the interval [1, 12] was partitioned into 50 equal-sized subintervals. Endpoints of these subintervals were held out for validation. With X drawn uniformly from each subinterval, 50 random vectors were sampled from the joint distribution. These random vectors were then concatenated to form a calibration set. This sampling design ensured that the validation data were unseen in the calibration set. JCDN with identified number of hidden neurons was trained on the calibration set. Parameter estimation was made on the held-out interval endpoints, as presented in Figures 3d–3f. Obviously, the generalization ability of the trained JCDN was consistent with the results in Figures 3a–3c. No significant evidence of under- or overfitting was found, which also provided an empirical justification for the rationale of the identified JCDN hyperparameters. After all, this preliminary study showed that JCDN was capable of capturing the underlying covariate-parameter relationship and can be used for fitting a nonstationary bivariate distribution.

5.2. Case 2: River Flow Synthesizing

[31] Considering the fact that to date to our knowledge, the idea of conditional density network has not yet been used for river flow synthesizing; case 2 presents a comprehensive treatment of the utility of JCDN for generating synthetic river flow sequences. Fifty-one years of monthly river flow records spanning over a period of 1956–2006 at four sites located on the main stream of the Colorado River were used to test the model performance. Among the four sites, one is station 09380000, Lees Ferry, Arizona, which has been utilized several times in related studies, like *Prairie et al*. [2006], *Salas and Lee* [2010], and *Hao and Singh* [2011]. The other three stations sited to the downstream of station 09380000 are station 09402500 near Grand Canyon, station 09427520 below Parker Dam, and station 09429490 above Imperial Dam, respectively.

[32] Before proceeding to hyperparameter identification, it was required to select appropriate covariates. Note that for river flow synthesizing the covariates are typically flow variables created from historical observations, for instance, lagged flows and aggregated flows over several past months. Herein there were in total four covariates involved in estimating the joint distribution of river flows of months $t-1$ and t: (1) the summation of river flows of months $t-1$ and t; (2) the summation of river flows of months $t-2$, $t-3$, $t-4$, and $t-5$; (3) the summation of sine values of $t-1$ and t; and (4) the summation of cosine values of $t-1$ and t. The second covariate bears similarity to the one used in the nonparametric model for river flow preserving long-term variability (NPL) developed by *Sharma and O'Neill* [2002]. Therein the trivariate distribution of adjacent monthly and annual aggregated flows is approximated by kernel density estimation and then is used for river flow simulation. The last two covariates aim to capture seasonal cycles inherent in the data [*Cannon*, 2008; *Carreau et al*., 2009; *Carreau and Vrac*, 2011]. These covariates were first transferred to be normally distributed.

The quantile matching method was used for data transformation, which is mathematically generalized as

$$x_{i:N} \rightarrow z_{i:N} \quad := \quad z_{i:N} = \Phi^{-1}\left(\frac{i-0.5}{N}\right), \tag{10}$$

where $x_{i:N}$ and $z_{i:N}$ are the ith order statistics of data in the original scale and the corresponding normal transformed value, respectively; and $\Phi^{-1}(\cdot)$ is the quantile function of the standard Gaussian distribution.

[33] Hyperparameters of JCDN, i.e., the copula and the marginal distribution families and the number of hidden neurons, were identified by fivefold cross-validation. To that end, data were split into five decade-long segments, i.e., 1957–1966, 1967–1976, 1977–1986, 1987–1996, and 1997–2006. It is noted that flows for the year of 1956 were excluded from cross-validation, since data of this year were used for preparing covariates. The fivefold cross-validation was implemented following the procedure described in section 3.2. For simplicity but without losing rationale, marginal distributions of river flows of two adjacent months were assumed to belong to the same family. There were two admissible alternatives for the marginal distribution and three for the copula function. The number of hidden

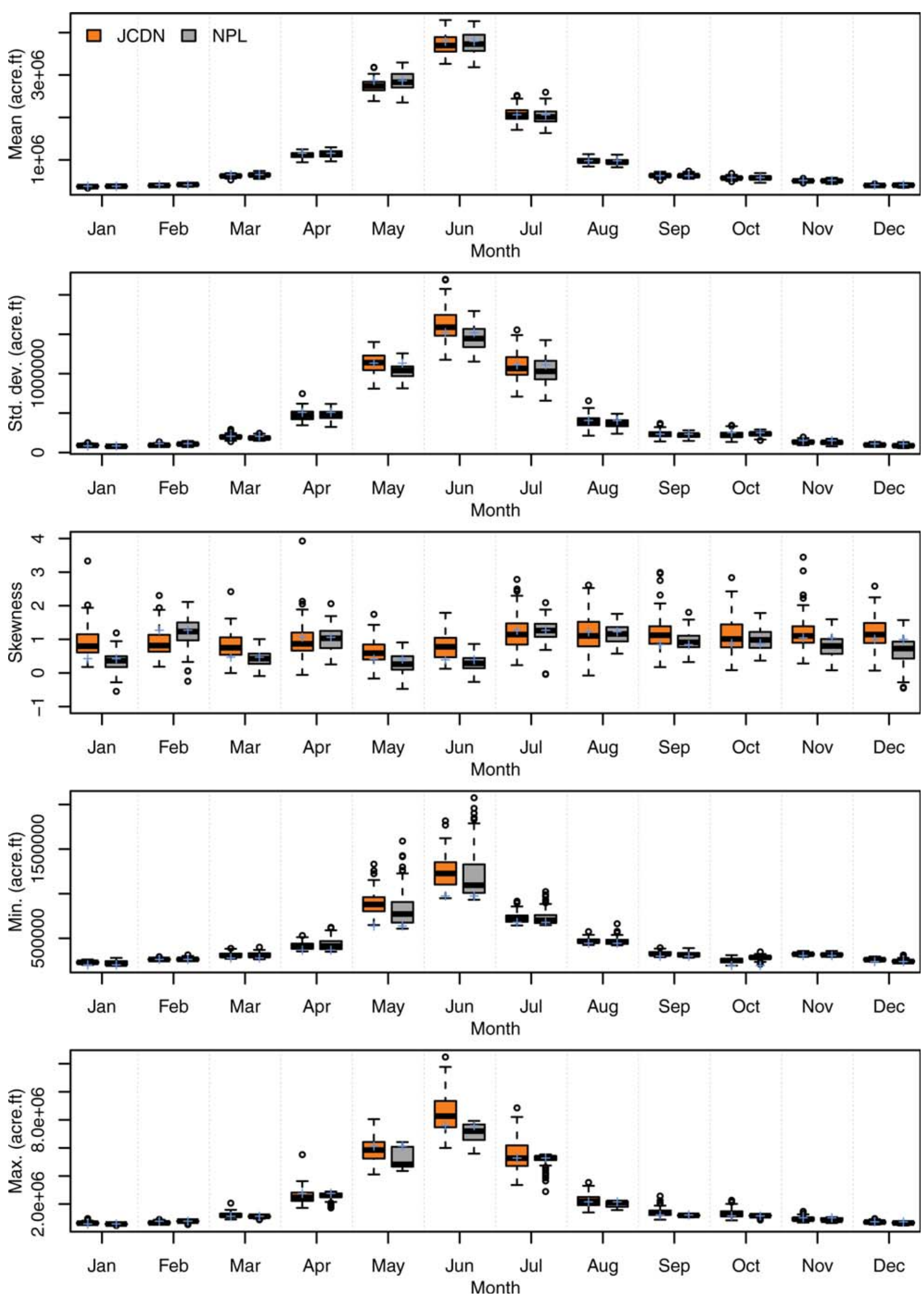

Figure 4. Box plots of basic distributional statistics of observed and synthesized sequences by JCDN and NPL for station 09380000.

neurons was chosen from [2, 3, 4, 5, 6, 7, 8]. Therefore, there were in total 42 possible combinations involved in the cross-validation exercise. For each station, the combination resulting in the maximum log likelihood was finally used to set up JCDN.

[34] To evaluate how good the performance of JCDN in river flow synthesizing is, 1000 sequences, each with the same length as the effective historical records (50 years), were generated for each station. In view of the possibility of converging to local minima and in order to achieve a relative objective evaluation, the 1000 sequences were generated as follows. First, JCDN with identified hyperparameters was trained on the whole data set to estimate the nonstationary joint distribution of monthly river flow. Then, 100 sequences were generated from the estimated distribution. The above two steps were repeated 10 times resulting in 1000 sequences.

[35] A set of distributional, autocorrelation, surplus, and deficit statistics were used to evaluate the performance of JCDN-based generator in river flow synthesizing. The basic distributional statistics included monthly (1) mean, (2) standard deviation, (3) skewness, (4) minimum, and (5) maximum. The serial autocorrelation statistics were (6) lag-1 autocorrelation and (7) lag-2 autocorrelation. Surplus statistics included (8) maximum surplus length and (9) maximum surplus amount for different static water demand levels. Deficit statistics were simply the counterpart of surplus statistics. They were (10) maximum deficit length and (11) maximum deficit amount. These statistics have been extensively used for assessing products of river flow synthesizing [*Sharma and O'Neill*, 2002; *Prairie et al.*, 2008; *Salas and Lee*, 2010; *Hao and Singh*, 2011].

[36] The monthly mean, standard deviation, skewness, minimum, and maximum values of synthetic and historical sequences for each month are summarized by box plot in Figure 4 for the representative station 09380000, along with the corresponding results obtained from the NPL model for comparison. Figure 4 reveals one major point that JCDN exhibited fairly good performance in reproducing these statistics. One additional interesting point is related to the minimum and maximum statistics, both of which have been recognized as relatively hard to simulate

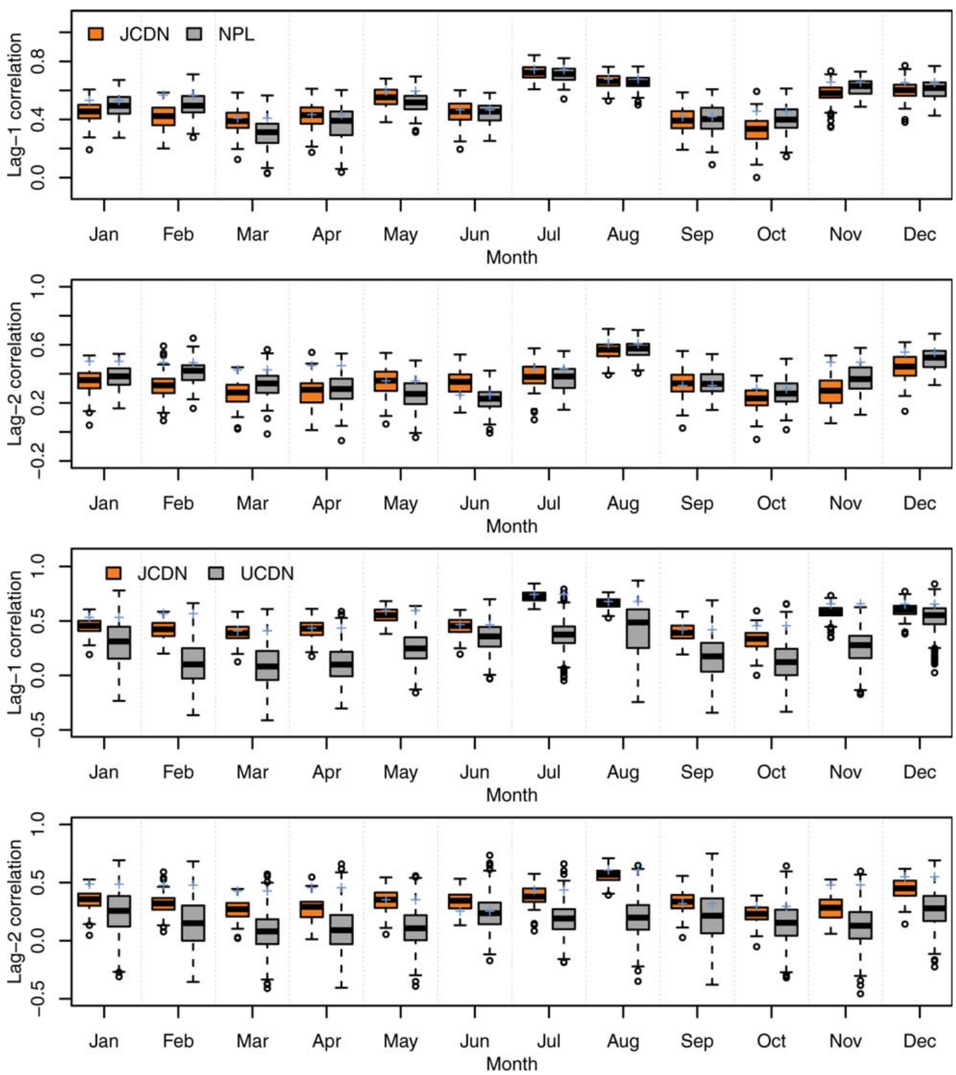

Figure 5. Box plots of autocorrelation statistics of observed and synthesized sequences by JCDN, NPL, and UCDN for station 09380000.

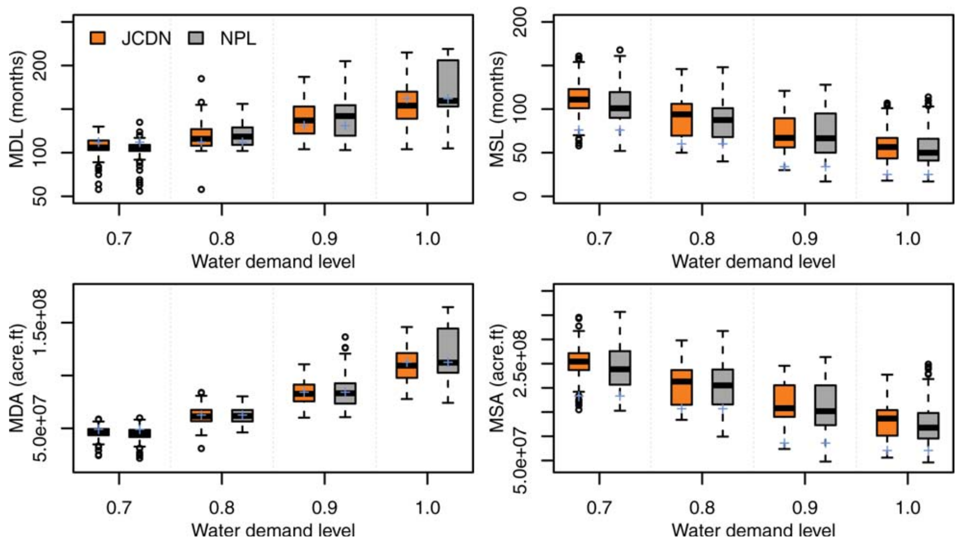

Figure 6. Box plots of deficit and surplus statistics (MDL: maximum deficit length; MDA: maximum deficit amount; MSL: maximum surplus length; MSA: maximum surplus amount) of observed and synthesized sequences by JCDN and NPL for station 09380000.

[*Salas and Lee*, 2010; *Hao and Singh*, 2011]. Apparently, these two statistics were reasonably captured by JCDN.

[37] Figure 5 illustrates lag-1 and lag-2 autocorrelations of the synthetic and observed sequences. As expected, the JCDN-based generator presents decent performance in preserving the short-term persistence of available observed flow data. Particularly, lag-1 autocorrelation was better reproduced than that of lag-2, which was underestimated through the year. There was no significant difference between JCDN and NPL in reproducing short-term autocorrelation statistics.

[38] In addition, to understand better how good the JCDN-based generator is in preserving autocorrelation of observed river flow comparing with its univariate analog (UCDN) [*Cannon*, 2008; *Carreau et al*., 2009; *Carreau and Vrac*, 2011], another 1000 sequences were generated by UCDN for this representative station. Lag-1 and Lag-2 autocorrelations of UCDN-synthesized and observed sequences are plotted in Figure 5 as well. Simulation experiment showed that no matter how we changed the hyperparameters we could not obtain a model with satisfactory results if employing the same covariates as in JCDN. We therefore tested different covariate combinations and finally selected the one (river flow of previous month and sine and cosine values of the current month) which could match the basic distributional statistics simulated by JCDN as well as possible. Results revealed that as expected UCDN underestimated the observed serial correlation. Sometimes, it might even generate sequences with unrealistic zero or negative autocorrelations. In the context of crisp prediction ("crisp" was used to distinguish with "probabilistic"), expectation of the predictive distribution or sample mean of multiple realizations from the predictive distribution is usually used as the predicted value. In this situation, the autocorrelation might be adequately preserved as long as lag covariates were included, as discussed by *Cannon* [2012]. One should be careful not to confuse that the reproduction of autocorrelation in the ensemble average of UCDN simulations does not mean the reproduction of autocorrelation in each ensemble member. The JCDN-based generator ensures that observed autocorrelation can be statistically preserved in each member.

[39] Comparison of surplus and deficit statistics between synthetic and observed sequences is presented in Figure 6. These statistics were computed using static water demand levels. Following *Hao and Singh* [2011], the water demand level was selected as a fraction (0.7, 0.8, 0.9, and 1.0) of the historical mean. As can be inferred from Figure 6, the deficit statistics were preserved quite well, whereas the surplus statistics were somewhat underrepresented.

[40] For economy of space, results for the other stations were pooled together and are depicted in Figures 7–9. Note that the simulations were carried out separately at each station, with all stations assumed to be spatially independent. Figure 7 is for basic distributional statistics, i.e., mean, standard deviation, skewness, maximum, and minimum. Figure 8 presents the serial autocorrelation statistics, and Figure 9 displays the surplus and deficit statistics. As seen from Figures 7–9, similar observations as those at station 09380000 can be found at the other stations. Therefore, we can remark that generally the JCDN-based generator can be a decent approach for river flow synthesizing.

5.3. Case 3: River Flow Downscaling

[41] The advantage of JCDN is that it is flexible at incorporating covariates. Changing different covariates makes JCDN applicable for different applications. For instance, in case 2 we have used covariates that are created from historical observations for river flow synthesizing. If we change the covariates to exogenous large-scale climate variables, then JCDN can be applied for river flow downscaling. Case 3 is to illustrate how JCDN can be used for this purpose. Note that our intention here is to give an illustrative example to demonstrate the utility of JCDN for river flow downscaling rather than to conduct a comprehensive downscaling exercise.

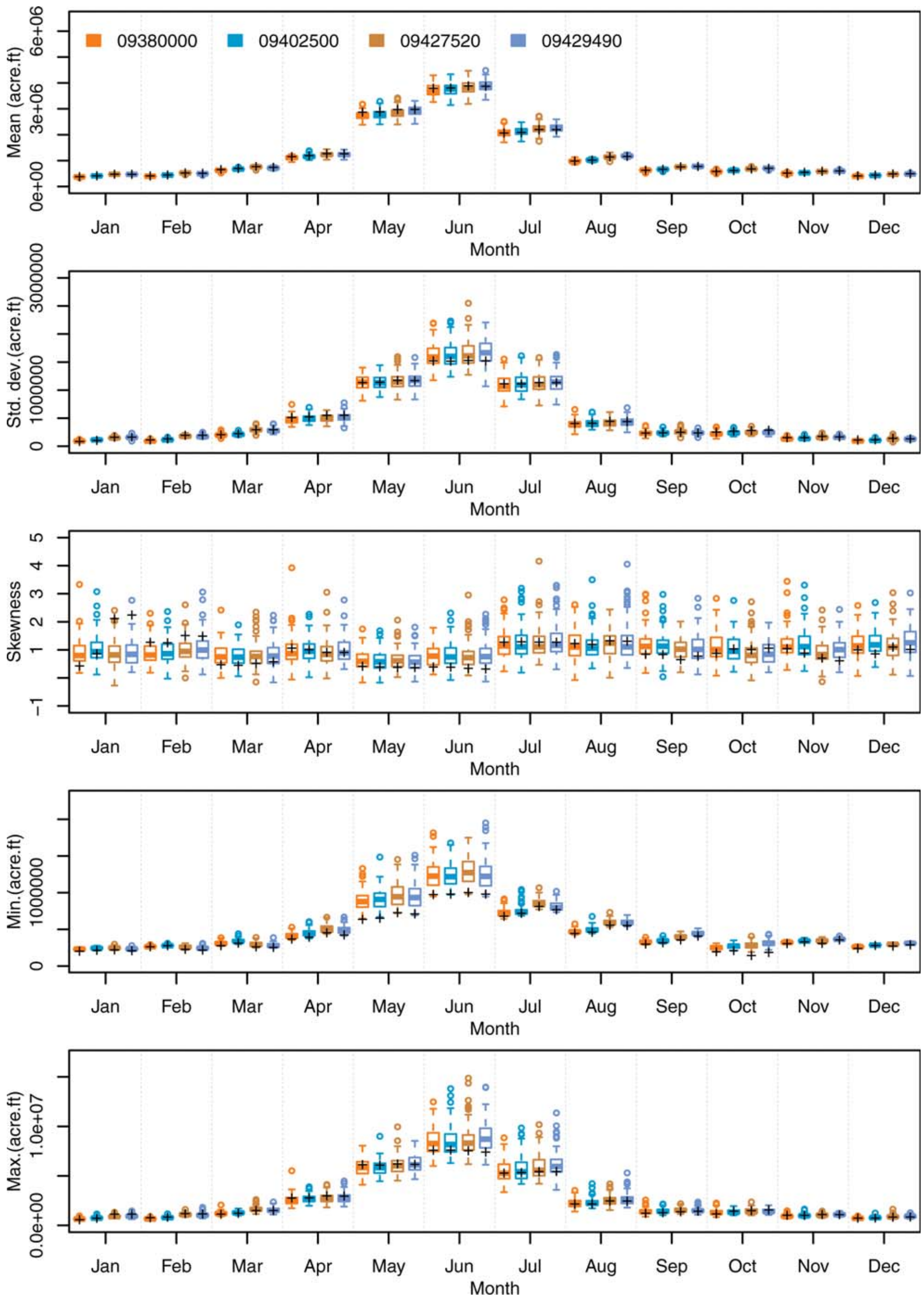

Figure 7. Box plots of basic distributional statistics of observed and synthesized sequences by JCDN for all stations selected in this study.

[42] We simply chose station 09380000 as an example. Potential covariates were drawn from the literature review of *Cannon and Whitfield* [2002] and *Ghosh and Mujumdar* [2008]. They were mean sea level pressure, geopotential height at 500 hpa, and specific humidity at surface and 850 hpa. Gridded climate data at monthly scale of a period from 1960 to 1999 were obtained for an area spanning over 30°–41°N in latitude and 105°–117°E in longitude. These data can be found at http://www.esrl.noaa.gov/psd/data/reanalysis/reanalysis.shtml. The period from 1960 to 1999 was used mainly for the consideration that it is neither too long nor too contemporary to include strong global change signals [*Ghosh and Mujumdar*, 2008].

[43] There were in total 16 grids involved. Climate data of months $t-1$ and t were averaged for each variable at each gird, resulting in 80 covariates. After normalization with the quantile matching method, principal component analysis was applied to transform the normalized covariate set into another set of perpendicular vectors. It was observed that the first three principal components explained approximately 98% of the information content retained by all the 80 covariates, and therefore, they were finally used

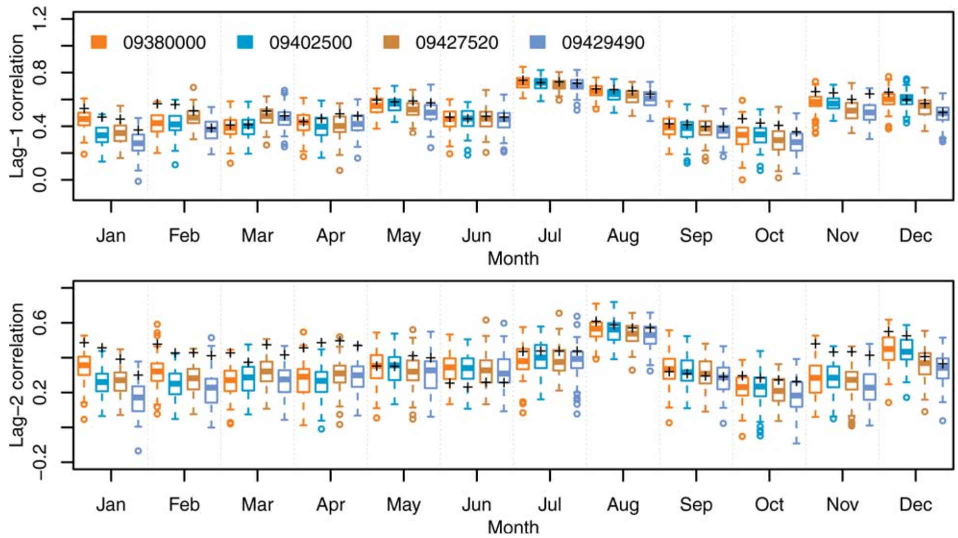

Figure 8. Box plots of autocorrelation statistics of observed and synthesized sequences by JCDN for all stations selected in this study.

for downscaling. In addition, the summations of sine values of months $t-1$ and t and of cosine values of months $t-1$ and t were also included as covariates to encode seasonal cycles. It is worth noting that the quantile matching method may not be suitable for an "out-of-sample" context, for example, projecting future climate scenarios to river flow. In this situation, we promote the data preprocessing approaches discussed by *Ghosh and Mujumdar* [2008] and *Cannon* [2012]. The whole data from 1960 to 1999 were then split into two parts. Date for the first 35 years was used for JCDN calibration and the remainder of the data for validation (or evaluation). Fivefold cross-validation was performed on the calibration set to select the best hyperparameter combination. The identified JCDN had a structure of being a Gaussian copula linking two gamma marginal distributions with five hidden neurons. Then JCDN was trained anew on the whole calibration set. Prediction was made on the validation set. One thousand sequences each with the same length as the validation period were generated following the procedure as described in section 5.2.

[44] Since the validation period (5 years) is too short to calculate the evaluation statistics in section 5.2 with reliable accuracy, the model evaluation strategy used there may not be feasible here. We therefore averaged 1000 sequences and compared them against observations, as presented in Figure 10. In order to properly appreciate the skill of JCDN, we compared it with another reference model: relevance vector machine (RVM), which has been successfully used for downscaling monthly flow at river basin

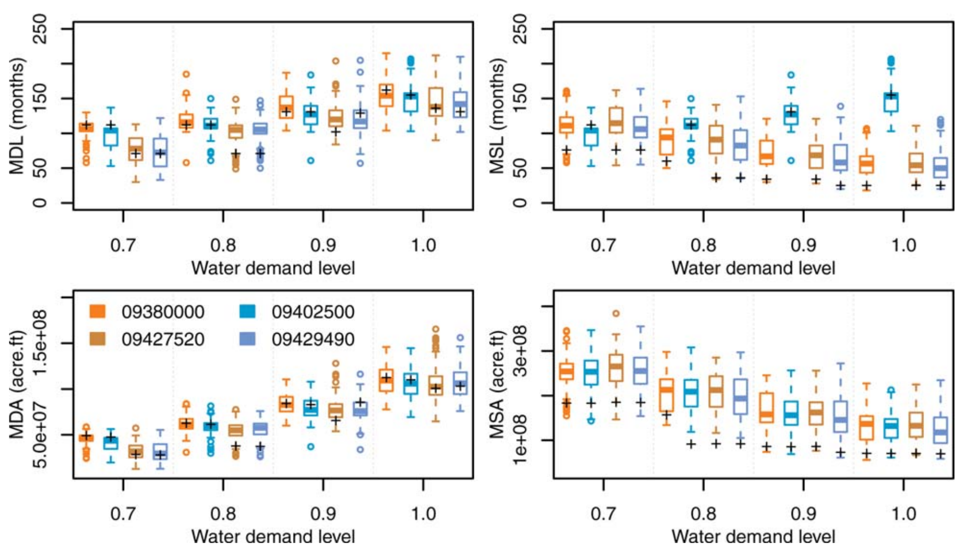

Figure 9. Box plots of deficit and surplus statistics (MDL, maximum deficit length; MDA: maximum deficit amount; MSL, maximum surplus length; MSA, maximum surplus amount) of observed and synthesized sequences by JCDN for all stations selected in this study.

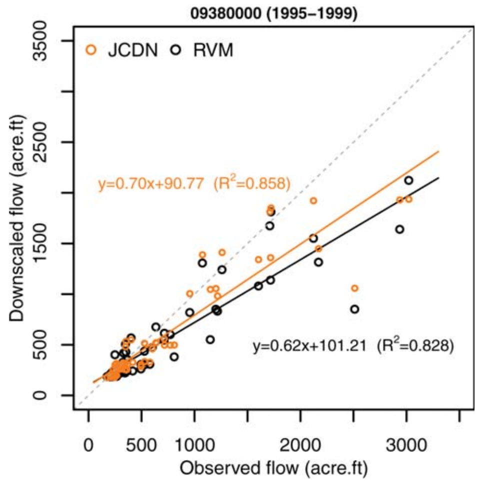

Figure 10. Observed versus JCDN and RVM downscaled monthly river flow of the validation period.

scale [*Ghosh and Mujumdar*, 2008]. As a note, the Gaussian kernel with a bandwidth of 3.5 was used to set up RVM. The bandwidth was simply determined through five-fold cross-validation by trial and error of values from 1 to 10 with a step size of 0.1. Implementations of RVM were accomplished with the MATLAB package developed by *Tipping* [2001]. The results were included in Figure 10 as well. One can observe that JCDN presented decent performance with a determination coefficient R^2 of 0.858 which is approximately the same as that of RVM. One should realize that the selection of large-scale climate covariates is critical for a downscaling model. The work of *Sharma* [2000a, 2000b] discussed a novel approach based on nonparametric partial mutual information for predictor (or covariate) selection. It is likely that the results of JCDN would improve if the covariates were more carefully selected.

[45] Figure 11a presents the observed and projected river flow time series by JCDN and RVM of the validation period. It shows a quite good consistency between observed and projected series by both JCDN and RVM. JCDN provides a full predictive distribution for river flow of each month, from which a predictive confidence interval can be obtained, as shown by the gray-shaded envelope. One may have also observed a right-skewed confidence interval for each month of each year, which may be easier seen from the superimposed box plots of JCDN realizations of several representative months. This is expected since JCDN assumes gamma or lognormal distribution for monthly river flow. This is an interesting advantage of JCDN over most multivariate regression techniques, in which river flow is assumed to be Gaussian distributed. From Figure 11a, one may also notice that not only the ensemble average but also each ensemble member (dashed line) preserved the seasonal variability, which may not necessarily hold for the UCDN model.

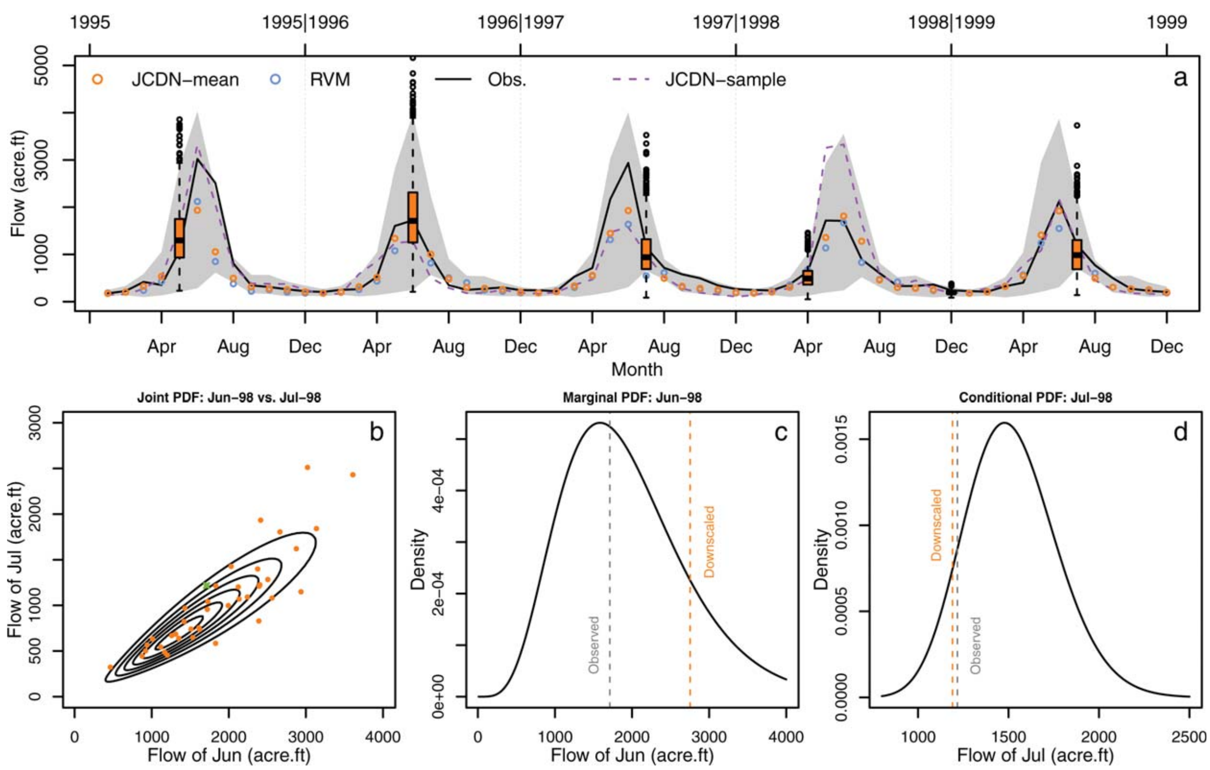

Figure 11. (a) Observed sequence (solid line), ensemble mean of JCDN downscaled sequences (orange circles), and the 95% confidence interval (gray-shaded envelop), RVM downscaled sequence (light blue circles), and a randomly selected ensemble member of JCDN downscaled sequences (dashed line) of the validation period; (b) the predictive joint distribution of flows of June and July 1998; (c) the marginal distribution of June 1998; and (d) the conditional predictive distribution of July 1998 derived from the joint predictive distribution given a random realization of a downscaled value of June.

[46] At last, it is worthwhile to have some intuition about the working machinery of the JCDN-based generator. The contour plot in Figure 11b represents the JCDN-derived predictive PDF of flows of June and July of the year 1998. The scatter points are the corresponding observed data pairs from 1960 to 1999. It is observed that the scatter points were not captured well by the contour plot. This is not unreasonable since the predictive distribution was for flows of June and July of 1998 rather than for the whole observed data pairs. Figure 11c shows the marginal distribution of June of 1998. The observed flow of this month is marked by a gray-dashed line, which is roughly the same as the mode of the distribution. The downscaled flow, which is a random realization rather than the expectation of the conditional distribution of June given the downscaled value of May, is marked by an orange-dashed line. Given this value and from the joint predictive distribution, a conditional predictive distribution of July was derived, as presented in Figure 11d. Similarly, a random realization of this conditional distribution and the corresponding observed flow are marked by the orange- and gray-dashed lines, respectively.

6. Conclusions

[47] We have presented a probabilistic model, termed JCDN, for fitting a joint distribution whose parameters are not constant but vary with covariates. The fitted distribution can be used for generating synthetic river flow sequences in a chain-dependent way. Since JCDN is flexible at incorporating covariates, by altering different covariates it may have different potential applications, like river flow synthesizing, downscaling, rainfall-runoff simulation, and for any purpose of probabilistic prediction. With three case studies, we tested the idea of JCDN on synthetic data first and then applied it for river flow synthesizing and downscaling. Particularly, a detailed treatment was given for river flow synthesizing in view of the fact that similar modeling framework has not been used for generating artificial hydrologic sequences. Based on the analysis, three major conclusions can be generalized:

[48] (1) The synthetic case study shows that the presented JCDN can be used for fitting a nonstationary joint distribution with decent performance.

[49] (2) JCDN can be used for monthly river flow synthesizing with the basic distributional statistics being adequately reproduced, as well as lag-1 autocorrelation and long-term deficit statistics.

[50] (3) JCDN is flexible at incorporating covariates. Therefore, it might be adapted for monthly flow downscaling. A simple intuitive downscaling example signifies the decent performance of JCDN.

[51] It should be noted that still there are some problems inherent in JCDN that require further study in the future. First, the presented JCDN does not account for modeling zero-inflated data, which are commonly seen in arid areas. Nevertheless, theoretically, there is nothing to restrict the JCDN modeling framework to fit a bivariate discrete-continuous mixture distribution, like the work of *Cannon* [2008, 2012] and *Carreau and Vrac* [2011], therein a number of zero-inflated univariate distributions were estimated by neural networks. Second, the proposed model cannot adequately reproduce the higher-order autocorrelation property. Multivariate copula (higher than 2) might be a good solution for this problem by constructing a joint distribution of adjacent monthly river flows of longer time lags.

[52] **Acknowledgments.** This work was financially supported in part by the U.S. Geological Survey (USGS, project 2009TX334G) and TWRI through the project "Hydrological Drought Characterization for Texas Under Climate Change, With Implications for Water Resources Planning and Management."

References

Bernadara, P., D. Schertzer, E. Sauquet, I. Tchiguirinskaia, and M. Lang (2008), The flood probability distribution tail: How heavy is it? *Stochastic Environ. Res. Risk Assess.*, *22*, 107–122.

Bishop, C. M. (1995), Neural Networks for Pattern Recognition, Clarendon, Oxford, U.K.

Cannon, A. J. (2008), Probabilistic multisite precipitation downscaling by an expanded Bernouilli-Gamma density network, *J. Hydrometeorol.*, *9*, 1284–1300.

Cannon, A. J. (2012), Neural network for probabilistic environmental prediction: Conditional Density Estimation Network Creation & Evaluation (CaDENCE) in R, *Comput. Geosci.*, *41*, 126–135, doi:10.1016/j.cageo.2001.08.23.

Cannon, A. J., and P. H. Whitfield (2002), Downscaling recent streamflow conditions in British Columbia, Canada using ensemble neural network models, *J. Hydrol., 259*, 136–151.

Carreau, J., and M. Vrac (2011), Stochastic downscaling of precipitation with neural network conditional mixture models, *Water Resour. Res.*, *47*, W10502, doi:10.1029/2010WR010128.

Carreau, J., P. Naveau, and E. Sauquet (2009), A statistical rainfall-runoff mixture model with heavy-tailed components, *Water Resour. Res.*, *45*, W10437, doi:10.1029/2009WR007880.

Casella, G., and R. L. Berger (2001), *Statistical Inference*, 2nd ed., Thomson Learning, Pacific Grove, Calif.

Cawley, G. C., G. J. Janacek, M. R. Haylock, and S. R. Dorling (2007), Predictive uncertainty in environmental modeling, *Neural Netw.*, *20*, 537–549.

Duan, Q., S. Sorooshian, and V. Gupta (1992), Effective and efficient global optimization for conceptual rainfall-runoff models, *Water Resour. Res.*, *28*(4), 1015–1031.

Evin, G., J. Merleau, and L. Perreault (2011), Two-component mixtures of normal, gamma, and Gumbel distributions for hydrological applications, *Water Resour. Res.*, *47*, W08525, doi:10.1029/2010WR010266.

Fernandez, B., and J. D. Salas (1990), Gamma autoregressive models for stream flow simulation, *J. Hydraul. Eng.*, *116*(11), 1403–1414.

Ghosh, S., and P. P. Mujumdar (2008), Statistical downscaling of GCM simulations to streamflow using relevance vector machine, *Adv. Water Resour.*, *31*(1), 132–146.

Hao, Z., and V. P. Singh (2011), Single-site monthly streamflow simulation using entropy theory, *Water Resour. Res.*, *47*, W09528, doi:10.1029/2010WR010208.

Hsieh, W. W. (2009), *Machine Learning Methods in the Environmental Science: Neural Networks and Kernels*, Cambridge Univ. Press, Cambridge, U.K.

Huth, R., S. Kliegrová, and L. Metelka (2008), Non-linearity in statistical downscaling: Does it bring an improvement for daily temperature in Europe?, *J. Climatol.*, *28*, 465–477.

Joe, H. (1997), *Multivariate Models and Dependence Concepts*, Chapman & Hall, New York.

Lall, U., and A. Sharma (1996), A nearest neighbor bootstrap for resampling hydrologic time series, *Water Resour. Res.*, *32*(3), 679–693, doi:10.1029/95WR02966.

Lee, T., and J. D. Salas (2011), Copula-based stochastic simulation of hydrological data applied to Nile River flows, *Hydrol. Res.*, *42*(4), 318–330.

Lettenmaier, D., and S. Burges (1977), An operational approach to preserving skew in hydrologic models of long-term persistence, *Water Resour. Res.*, *13*(2), 281–290, doi:10.1029/WR1013i1002p00281.

Liu, Y., and H. V. Gupta (2007), Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework, *Water Resour. Res.*, *43*, W07401, doi:10.1029/2006WR005756.

Mehrotra, R., and A. Sharma (2007a), A semi-parametric model for stochastic generation of multi-site daily rainfall exhibiting low-frequency variability, *J. Hydrol.*, *335*, 180–193.

Mehrotra, R., and A. Sharma (2007b), Preserving low-frequency variability in generated daily rainfall sequences, *J. Hydrol.*, *345*, 102–120.

Mujumdar, P. P., and S. Ghosh (2008), Modeling GCM and scenario uncertainty using a possibilistic approach: Application to the Mahanadi River, India, *Water Resour. Res.*, *44*, W06407, doi:10.1029/2007WR006137.

Nadarajah, S. (2007), Exact distribution of the peak streamflow, *Water Resour. Res.*, *43*, W02501, doi:10.1029/2006WR005300.

Nelsen, R. B. (2006), *An Introduction to Copulas*, 2nd ed., Springer, New York.

Prairie, J., B. Rajagopalan, T. Fulp, and E. Zagona (2006), Modified K-NN model for stochastic streamflow simulation, *J. Hydrol. Eng.*, *11*(4), 371–378.

Prairie, J., K. Nowak, B. Rajagopalan, U. Lall, and T. Fulp (2008), A stochastic nonparametric approach for streamflow generation combining observational and paleoreconstructed data, *Water Resour. Res.*, *44*, W06423, doi:10.1029/2007WR006684.

Rajagopalan, B., and U. Lall (1999), A k-nearest-neighbor simulator for daily precipitation and other variables, *Water Resour. Res.*, *35*(10), 3089–3101.

Salas, J., and J. Delleur (1980), *Applied Modeling of Hydrologic Time Series*, Water Resour. Publ., Littleton, Colo.

Salas, J., and T. Lee (2010), Nonparametric simulation of single-site seasonal streamflows, *J. Hydrol. Eng.*, *15*(4), 284–296.

Sangal, B. P., and A. K. Biswas (1970), The 3-parameter lognormal distribution and its application in hydrology, *Water Resour. Res.*, *62*(2), 505–515, doi:10.1029/WR006i002p00505.

Schoof, J. T., and S. C. Pryor (2001), Downscaling temperature and precipitation: A comparison of regression-based methods and artificial neural networks, 773–790.

Sharma, A. (2000), Seasonal to interannual rainfall probabilistic forecasts for improved water supply management. Part 1: A strategy for system prediction identification, *J. Hydrol.*, *239*, 232–239.

Sharma, A. (2010), Seasonal to interannual rainfall probabilistic forecasts for improved water supply management. Part 3: A nonparametric probabilistic forecast model, *J. Hydrol.*, *239*, 249–258.

Sharma, A., and R. O'Neill (2002), A nonparametric approach for representing interannual dependence in monthly streamflow sequences, *Water Resour. Res.*, *38*(7), 1100, doi:10.1029/2011WR000953.

Sharma, A., D. Tarboton, and U. Lall (1997), Streamflow simulation: A nonparametric approach, *Water Resour. Res.*, *33*(2), 291–308, doi:10.1029/96WR02839.

Sharma, A., K. C. Lik, I. Corderly, U. Lall (2000), Seasonal to interannual rainfall probabilistic forecasts for improved water supply management. Part 2: Predictor identification of quarterly rainfall using ocean-atmosphere information, *J. Hydrol.*, *239*, 240–248.

Tipping, M. E. (2001), Sparse Bayesian learning and the relevance vector machine, *J. Mach. Learn. Res.*, *1*, 211–244.

Tisseuil, C., M. Vrac, S. Lek, and A. J. Wade (2010), Statistical downscaling of river flows, *J. Hydrol.*, *385*, 279–291.

Vrac, M., M. Stein, and K. Hayhoe (2007), Statistical downscaling of precipitation through nonhomegeneous stochastic weather typing, *Clim. Res.*, *34*, 169–184.

Wand, M. P., and M. C. Jones (1995), *Kernel Smoothing*, Chapman and Hall, New York.

Wilks, D. S. (1999), Multisite downscaling of daily precipitation with a stochastic weather generator, *Clim. Res.*, *11*, 125–136.

Zhang, L., and V. P. Singh (2007), Bivariate rainfall frequency distributions using Archimedean copulas, *J. Hydrol.*, *332*, 93–109.