Interactive comment on "A methodology to estimate flow duration curves at partially ungauged basins" *by* Elena Ridolfi et al.

F. Serinaldi

francesco.serinaldi@ncl.ac.uk

Received and published: 20 July 2018

Dear Elena,

This is a very quick comment about your algorithm. Please note that, in principle, you do not need to introduce any probability P_i and P_j (leaving aside that the standardized ranks of nonhomogeneous seasonal flow are not estimators of probabilities). In fact, your sorted API_A_r and q_B_r define a monotonic function $f : API\_A_r \rightarrow q\_B_r$ such that new/unknown values q_B_t can be found by simple interpolation. Their standardized ranks are not needed for interpolation, and those of the sorted q_B_t are automatically defined by the position of each value in the sorted target sequence.

What you do can easily be performed without nuisance (I would say, superfluous) prob-

abilites by the R function 'approx' (see codes below) or whatever similar interpolation function in other software. By the way, instead of using API values corresponding to 'probability' values Pj and Pj+1 when the target value is not available in the reference set, a better approach is the linear interpolation (which is what 'approx' and whatever standard interpolator does). This is standard when dealing fractional order statistics, which are actually the objects you are handling in this work.

Moreover, interpolation works if the range API_Ar includes all the values of API_At. Otherwise, extrapolation is needed, and it can be problematic if we rely on empirical survival functions. Another minor comment: you state 'Specifically, the procedure involves daily Antecedent Precipitation Index (API) and daily streamflow values of both A and B.' I may have missed something, but from your example, it seems that you need only API in A for the reference and target period, and flow q in B for the reference period.

However, my main concern is about the justification of the procedure, i.e. the link between weather variables (e.g. API) and flow. Since you work with sorted sequences, when you arrange API and q in ascending order, you lose the correlation between simultaneous pairs (API, q). This means that the physical link between API and q (...showing the correlation/scatter plots of such variables could be useful) does not play any role in your procedure. The simple R code below show how we can closely reproduce your results dealing with perfectly independent or dependent random variables X and Y. I used three setups: (1) a trivial iid, (2) a scenario where I introduced level shifts for groups of 365 realizations, mimicking the possible (and generally evident) fluctuations of the annual average flow (in the spirit of the index-flow flow duration curves of Castellarin et al.), and (3) a third case with shifts and some correlation. Running the code a few times, in all cases, I obtained results close to yours, even though these realizations are purely fictional. Of course, you can play with this code changing parameters, sample size, distributions, etc.

However, my message is to explore a bit more what you are doing and why you obtain

those results. To be honest, I think that whatever variable recorded in the reference and target period can be used as a support for interpolation, if its realizations in the reference period explore the sample space in the same way as the realizations of the target variable do. What we see in your diagrams concerns the properties of order statistics and sampling rather than physics. In any case, you should perform extensive simulations under controlled conditions to support your assumptions, in order to understand which parameter/aspect matters. Please, remember that our algorithms should be falsified rather than verified. Another way to check your results is to use other random variables as a reference, e.g. pure fictional variables, weather/climate indices showing weak or null correlation with flow, etc.

Minor remark: you use 'target site', 'reference site', 'target period', and 'reference period'. This can be confusing. I can suggest 'donor site' and 'ungauged site'. Finally, 'simulated' values in the target period can be denoted more properly as 'interpolated' values I think, because 'simulated' usually refers to some pseudo-random generation or stuff like that, which is not what is done here.

As ever

F

PS: if you need further information, discussion, etc., feel free to mail me.

```
################################################
set.seed(666)
xa.r = rnorm(365*20)
yb.r = rlnorm(365*20)
xa.t = rnorm(365)
yb.t.obs = rlnorm(365)
```

```
yb.t.sim = approx(x = sort(xa.r), y = sort(yb.r), xout = sort(xa.t))$y

plot(1-ppoints(365,0),sort(yb.t.obs) , type="l", log="y") lines(1-ppoints(365,0), yb.t.sim
,col=2)

# with perturbation mimicking fluctuations of annual mean levels

set.seed(6666)

xa.r = rnorm(365*20)+rep(rnorm(20,sd=0.2),each=365)

yb.r = rlnorm(365*20)*rep(rlnorm(20,sd=0.6),each=365)

xa.t = rnorm(365)

yb.t.obs = rlnorm(365)

yb.t.sim = approx(x = sort(xa.r), y = sort(yb.r), xout = sort(xa.t))$y

plot(1-ppoints(365,0),sort(yb.t.obs) , type="l", log="y")

lines(1-ppoints(365,0), yb.t.sim ,col=2)

# with perturbation and some cross-correlation

xa.r = rnorm(365*20)+rep(rnorm(20,sd=0.4),each=365)

yb.r = rlnorm(365*20)*rep(rlnorm(20,sd=0.6),each=365)

rho = 0.8

corM <- rbind(c(1.0, rho), c(rho, 1.0))

SigmaEV <- eigen(corM)

Meps <- rbind(xa.r,log(yb.r))

Meps <- SigmaEV$vectors %*% diag(sqrt(SigmaEV$values)) %*% Meps

Meps <- t(Meps)
```

```
#plot(Meps)

xa.r = Meps[,1]

yb.r = exp(Meps[,2])

plot(xa.r,yb.r)

xa.t = rnorm(365)

yb.t.obs = rlnorm(365)

yb.t.sim = approx(x = sort(xa.r), y = sort(yb.r), xout = sort(xa.t))$y

plot(1-ppoints(365,0),sort(yb.t.obs) , type="l", log="y")

lines(1-ppoints(365,0), yb.t.sim ,col=2)

#############################################
```