Supplement of Hydrol. Earth Syst. Sci., 20, 3183–3191, 2016
http://www.hydrol-earth-syst-sci.net/20/3183/2016/
doi:10.5194/hess-20-3183-2016-supplement

Hydrology and Earth System Sciences
Open Access

Supplement of

Technical note: Multiple wavelet coherence for untangling scale-specific and localized multivariate relationships in geosciences

Wei Hu and Bing Cheng Si

Correspondence to: Wei Hu (wei.hu@plantandfood.co.nz) and Bing Cheng Si (bing.si@usask.ca)

S1 Calculation of smoothed auto- and cross-wavelet power spectra

In this section, we will only introduce the basics related to the calculation of smoothed auto- and cross-wavelet power spectra. Detailed information on the calculations of wavelet coefficients, cross-wavelet power spectra, and bivariate wavelet coherence can be found elsewhere (Kumar and Foufoula-Georgiou, 1997; Torrence and Compo, 1998; Torrence and Webster, 1999; Grinsted et al., 2004; Das and Mohanty, 2008; Si, 2008). The smoothed auto- and cross-wavelet power spectra require the calculation of wavelet coefficients, at different scales and spatial (or temporal) locations, for the response variable and all predictor variables. For convenience, only spatial variables will be referred to, as temporal variables can be similarly analyzed.

The continuous wavelet transform (CWT) of a spatial variable X_1 of length N (X_{1h}, h=1, 2, …, N) with equal incremental distance δx, can be calculated as the convolution of X_{1h} with the scaled and normalized wavelet (Torrence and Compo, 1998)

$$W^{X_1}(s,\tau) = \sqrt{\frac{\delta x}{s}} \sum_{\tau=1}^{N} X_{1h} \psi\left[(h-\tau)\frac{\delta x}{s}\right], \qquad (1)$$

where $W^{X_1}(s,\tau)$ is the wavelet coefficient of spatial variable X_1 at scale s and location τ, and $\psi[\ ]$ is the mother wavelet function. The Morlet wavelet is used in the CWT because it allows for the identification of both location-specific amplitude and phase information at different scales in a spatial series (Torrence and Compo, 1998). The Morlet wavelet can be expressed as (Grinsted et al., 2004)

$$\psi(\eta) = \pi^{-1/4} e^{i\omega\eta - 0.5\eta^2}, \qquad (2)$$

where ω and η are the dimensionless frequency and space ($\eta = s/x$), respectively.

The auto-wavelet power spectrum of spatial variable X_1 can be expressed as

$$W^{X_1,X_1}(s,\tau)=W^{X_1}(s,\tau)\overline{W^{X_1}(s,\tau)}, \tag{3}$$

where $\overline{W^{X_1}(s,\tau)}$ is a complex conjugate of $W^{X_1}(s,\tau)$. Therefore, Eq. (3) can also be expressed as the squared amplitude of $W^{X_1}(s,\tau)$, which is

$$W^{X_1,X_1}(s,\tau)=\left|W^{X_1}(s,\tau)\right|^2. \tag{4}$$

The cross-wavelet spectrum between spatial variables of Y and X_1 can be defined as

$$W^{Y,X_1}(s,\tau)=W^{Y}(s,\tau)\overline{W^{X_1}(s,\tau)}, \tag{5}$$

where $W^{Y}(s,\tau)$ is the wavelet coefficient of spatial variable Y.

Both the auto- and cross-wavelet spectra can be smoothed using the method suggested by Torrence and Compo (1998),

$$\overleftrightarrow{W}(s,\tau)=\mathrm{SM}_{scale}\left[\mathrm{SM}_{space}\left(W(s,\tau)\right)\right], \tag{6}$$

where $\overleftrightarrow{(\cdot)}$ is a smoothing operator. SM_{scale} and SM_{space} indicate the smoothing along the wavelet scale axis and spatial distance, respectively (Si, 2008). The $\overleftrightarrow{W}$ is the normalized real Morlet wavelet and has a similar footprint as the Morlet wavelet

$$\frac{1}{s\sqrt{2\pi}}e^{\left(-\tau^2/\left(2s^2\right)\right)}. \tag{7}$$

Therefore, the smoothing along spatial distance can be calculated as

$$\mathrm{SM}_{scale}\left(W(s,\tau)\right)=\sum_{k=1}^{N}\left(W(s,\tau)\frac{1}{s\sqrt{2\pi}}e^{\left(-(\tau-x_k)^2/\left(2s^2\right)\right)}\right)\Big|_s, \tag{8}$$

where $\big|_s$ represents a fixed s value. The Fourier transform of Eq. (7) is $e^{\left(-2s^2\omega^2\right)}$. Therefore, Eq. (8) can be implemented using Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) based on the convolution theorem, and is written as

$$\mathrm{SM}_{scale}\left(W\left(s,x\right)\right)=\mathrm{IFFT}\left(\mathrm{FFT}\left(W\left(s,x\right)\right)\left(e^{\left(-2s^2\omega^2\right)}\right)\right). \quad (9)$$

The smoothing along scales is then written as [*Torrence and Compo*, 1998]

$$\mathrm{SM}_{scale}\left(W\left(s_k,x\right)\right)=\frac{1}{2m+1}\sum_{l=k-m}^{k+m}\left(\mathrm{SM}_{space}\left(W\left(s_l,x\right)\right)\Pi\left(0.6s_l\right)\right)\Big|_x\ , \quad (10)$$

where Π is the rectangle function, $\big|_x$ indicates a fixed x value, and l is the index for the scales. The coefficient of 0.6 is the empirically determined scale decorrelation length for the Morlet wavelet (Torrence and Compo, 1998).

S2 Matlab code for MWC (mwc.m)

```
% This is a Matlab code (mwc.m) for calculating multiple wavelet coherence.
% Please copy the following content into a txt file and rename it to “mwc.m” prior to running.

function varargout=mwc(X,varargin)
%  Multiple Wavelet coherence
% Creates a figure of multiple wavelet coherence
% USAGE: [Rsq,period,scale,coi,sig95]=mwc(X,[,settings])
%
% Input: X: a matrix of multiple variables equally distributed in space
%            or time. The first column corresponds to the dependent variable,
%        and the second and consequent columns are independent variables.
%
% Settings: Pad: pad the time series with zeros?
% .        Dj: Octaves per scale (default: '1/12')
% .        S0: Minimum scale
% .        J1: Total number of scales
% .        Mother: Mother wavelet (default 'morlet')
% .        MaxScale: An easier way of specifying J1
% .        MakeFigure: Make a figure or simply return the output.
% .        BlackandWhite: Create black and white figures
% .        AR1: the ar1 coefficients of the series
% .            (default='auto' using a naive ar1 estimator. See ar1nv.m)
% .        MonteCarloCount: Number of surrogate data sets in the significance calculation. (default=1000)

% Settings can also be specified using abbreviations. e.g. ms=MaxScale.
% For detailed help on some parameters type help wavelet.
% Example:
%    t=[1:200]';
%    mwc([sin(t),sin(t.*cos(t*.01)),cos(t.*sin(t*.01))])

% Please acknowledge the use of this software package in any publications,
% by including text such as:

%   "The software for the multiple wavelet coherence was  provided by W. Hu
%   and B. Si, and is available in the Supplement of Hu and Si (2016)."
%   and cite the paper:
% "Hu, W., and B. Si (2016), Technical Note: Multiple wavelet coherence for untangling scale-specific and localized
%  multivariate relationships in geosciences, Hydrol. Earth Syst. Sci., volume and page numbers to be allocated."
```

```
%  (C) W. Hu and B. Si 2016
%
% ------------------------------------------------------------
%
%   Copyright (C) 2016, W. Hu and B. Si 2016
%   This software may be used, copied, or redistributed as long as it is not
%   sold and this copyright notice is reproduced on each copy made.  This
%   routine is provided as is without any express or implied warranties
%   whatsoever.
%
%    Wavelet software was provided by C. Torrence and G. Compo,
%       and is available at URL: http://paos.colorado.edu/research/wavelets/.
%
%    Crosswavelet and wavelet coherence software were provided by
%      A. Grinsted and is available at URL:
%    http://www.glaciology.net/wavelet-coherence
%
%   We acknowledge Aslak Grinsted for his wavelet coherency code (wtc.m) on
% which this code builds.
%
%----------------parse function arguments----------------------------------------------------

[row,col]=size(X);
[y,dt]=formatts(X(:,1));
mm=y(1,1);
nn=y(end,1);

for i=2:col;
[x,dtx]=formatts(X(:,i));

if (dt~=dtx)
    error('timestep must be equal between time series');
end

mm1=x(1,1);
nn1=x(end,1);

if mm1>mm
mm=mm1;
end

if nn1<nn
```

```
nn=nn1;
end

x1(:,(i-1))=x(:,1);
x2(:,(i-1))=x(:,2);

end

t=(mm:dt:nn)';



%common time period
if length(t)<4
   error('The three time series must overlap.');
end

n=length(t);

%----------default arguments for the wavelet transform-----------
Args=struct('Pad',1,...      % pad the time series with zeroes (recommended)
            'Dj',1/12, ...    % this will do 12 sub-octaves per octave
            'S0',2*dt,...    % this says start at a scale of 2 years
            'J1',[],...
            'Mother','Morlet', ...
            'MaxScale',[],...   %a more simple way to specify J1
            'MakeFigure',(nargout==0),...
            'MonteCarloCount',1000,...
            'BlackandWhite',0,...
            'AR1','auto',...
            'ArrowDensity',[30 30],...
            'ArrowSize',1,...
            'ArrowHeadSize',1);

Args=parseArgs(varargin,Args,{'BlackandWhite'});

if isempty(Args.J1)
   if isempty(Args.MaxScale)
      Args.MaxScale=(n*.17)*2*dt; %auto maxscale;
   end
   Args.J1=round(log2(Args.MaxScale/Args.S0)/Args.Dj);
end

```

```
ad=mean(Args.ArrowDensity);
Args.ArrowSize=Args.ArrowSize*30*.03/ad;
%Args.ArrowHeadSize=Args.ArrowHeadSize*Args.ArrowSize*220;
Args.ArrowHeadSize=Args.ArrowHeadSize*120/ad;

if ~strcmpi(Args.Mother,'morlet')
    warning('MWC:InappropriateSmoothingOperator','Smoothing operator is designed for morlet wavelet.');
end

if strcmpi(Args.AR1,'auto')
      for i=1:col
arc(i)= ar1nv(X(:,i));
       end
Args.AR1=arc
    if any(isnan(Args.AR1))
        error('Automatic AR1 estimation failed. Specify it manually (use arcov or arburg).');
    end
end

%-----------:::::::::::::--------- ANALYZE ----------::::::::::::------------

%Calculate and smooth wavelet spectrum Y and X


[Y,period,scale,coiy] = wavelet(y(:,2),dt,Args.Pad,Args.Dj,Args.S0,Args.J1,Args.Mother);
sinv=1./(scale');
smY=smoothwavelet(sinv(:,ones(1,n)).*(abs(Y).^2),dt,period,Args.Dj,scale);


dte=dt*.01;
idx=find((y(:,1)>=(t(1)-dte))&(y(:,1)<=(t(end)+dte)));
Y=Y(:,idx);
smY=smY(:,idx);
coiy=coiy(idx);

coi=coiy;

for  i=2:col
 [XS,period,scale,coix] = wavelet(x2(:,(i-1)),dt,Args.Pad,Args.Dj,Args.S0,Args.J1,Args.Mother);

idx=find((x1(:,(i-1))>=(t(1))-dte)&(x1(:,(i-1))<=(t(end)+dte)));
XS=XS(:,idx);
```

```
coix=coix(idx);

XS1(:,:,(i-1))=XS;
coi=min(coi,coix);

end

% -------- Calculate Cross Wavelet Spectra----------------------------

% ---- between dependent variable and independent variables------

for i=1:(col-1)
Wyx=Y.*conj(XS1(:,:,i));
sWyx=smoothwavelet(sinv(:,ones(1,n)).*Wyx,dt,period,Args.Dj,scale);
sWyx1(:,:,i)=sWyx;
end

% ----between independent variables and independent variables------
for i=1:(col-1);
for j=1:(col-1);
Wxx=XS1(:,:,i).*conj(XS1(:,:,j));
sWxx=smoothwavelet(sinv(:,ones(1,n)).*Wxx,dt,period,Args.Dj,scale);
sWxx1(:,:,i,j)=sWxx;
end
end

% --------------- Mutiple wavelet coherence ---------------------------------
% calculate the multiple wavelet coherence
for i=1:length(scale)
  parfor j=1:n
a=transpose(squeeze(sWyx1(i,j,:)));
b=inv(squeeze(sWxx1(i,j,:,:)));
c=conj(squeeze(sWyx1(i,j,:)));
d=smY(i,j);
Rsq(i,j)=real(a*b*c/d);
  end
end

% --------------- make figure-------------------------------------------
if (nargout>0)||(Args.MakeFigure)
```

```
mwcsig=mwcsignif(Args.MonteCarloCount,Args.AR1,dt,length(t)*2,Args.Pad,Args.Dj,Args.S0,Args.J1,Args.Mother,.
6);
    mwcsig=(mwcsig(:,2))*(ones(1,n));
    mwcsig=Rsq./mwcsig;
end

if Args.MakeFigure

    Yticks = 2.^(fix(log2(min(period))):fix(log2(max(period))));

    if Args.BlackandWhite
        levels = [0 0.5 0.7 0.8 0.9 1];
        [cout,H]=safecontourf(t,log2(period),Rsq,levels);

        colorbarf(cout,H)
        cmap=[0 1;.5 .9;.8 .8;.9 .6;1 .5];
        cmap=interp1(cmap(:,1),cmap(:,2),(0:.1:1)');
        cmap=cmap(:,[1 1 1]);
        colormap(cmap)
        set(gca,'YLim',log2([min(period),max(period)]), ...
            'YDir','reverse', 'layer','top', ...
            'YTick',log2(Yticks(:)), ...
            'YTickLabel',num2str(Yticks'), ...
            'layer','top');
        ylabel('Period');
        hold on

        if ~all(isnan(mwcsig))
            [c,h] = contour(t,log2(period),mwcsig,[1 1],'k');%#ok
            set(h,'linewidth',2);
        end
        %suptitle([sTitle ' coherence']);
        %plot(t,log2(coi),'k','linewidth',2)
                tt=[t([1 1])-dt*.5;t;t([end end])+dt*.5];
        %hcoi=fill(tt,log2([period([end 1]) coi period([1 end])]));
        %hatching- modified by Ng and Kwok
        hcoi=fill(tt,log2([period([end 1]) coi period([1 end])]),'w');

        hatch(hcoi,45,[0 0 0]);
        hatch(hcoi,135,[0 0 0]);
        set(hcoi,'alphadatamapping','direct','facealpha',.5);
```

```
        plot(t,log2(coi),'color','black','linewidth',1.5);
        hold off
    else
        H=imagesc(t,log2(period),Rsq);%#ok
        %[c,H]=safecontourf(t,log2(period),Rsq,[0:.05:1]);
        %set(H,'linestyle','none')

        set(gca,'clim',[0 1]);

        HCB=safecolorbar;%#ok

        set(gca,'YLim',log2([min(period),max(period)]), ...
            'YDir','reverse', 'layer','top', ...
            'YTick',log2(Yticks(:)), ...
            'YTickLabel',num2str(Yticks'), ...
            'layer','top');
        ylabel('Period');
        hold on

        if ~all(isnan(mwcsig))
            [c,h] = contour(t,log2(period),mwcsig,[1 1],'k');%#ok
            set(h,'linewidth',2);
        end
        %suptitle([sTitle ' coherence']);
        tt=[t([1 1])-dt*.5;t;t([end end])+dt*.5];
        hcoi=fill(tt,log2([period([end 1]) coi period([1 end])]),'w');
        set(hcoi,'alphadatamapping','direct','facealpha',.5);
        hold off
    end
end
%--------------------------------------------------------------------------%

varargout={Rsq,period,scale,coi,mwcsig};
varargout=varargout(1:nargout);

function [cout,H]=safecontourf(varargin)
vv=sscanf(version,'%i.');
if (version('-release')<14)|(vv(1)<7)
    [cout,H]=contourf(varargin{:});
else
    [cout,H]=contourf('v6',varargin{:});
end
```

```
function hcb=safecolorbar(varargin)
vv=sscanf(version,'%i.');

if (version('-release')<14)|(vv(1)<7)
    hcb=colorbar(varargin{:});
else
    hcb=colorbar('v6',varargin{:});
end
```

S3 Matlab code for significance test on multiple wavelet coherence

```
% This is a Matlab file (mwcsignif.m) for calculating significance tests on multiple wavelet coherence.
%Please copy the following content into a txt file and rename this file to “mwcsignif.m” prior to running.

function mwcsig=mwcsignif(mccount,ar1,dt,n,pad,dj,s0,j1,mother,cutoff)
% Multiple Wavelet Coherence Significance Calculation (Monte Carlo)
%
% mwcsig=mwcsignif(mccount,ar1,dt,n,pad,dj,s0,j1,mother,cutoff)
%
% mccount: number of time series generations in the monte carlo run
%(the greater the better)
% ar1: a vector containing two (in case of calculating wavelet
% coherence between two variables) or
% multiple (≥3) (in case of calculating multiple wavelet coherence
% with three or more variables)
% AR1 coefficients.
% dt,pad,dj,s0,j1,mother: see wavelet help...
% n: length of each generated timeseries. (obsolete)
%
% cutoff: (obsolete)
%
% RETURNED
% mwcsig: the 95% significance level as a function of scale... (scale,sig95level)
% -----------------------------------------------------------
% Please acknowledge the use of this software package in any publications,
% by including text such as:
%
%   "The software for the multiple wavelet coherence was  provided by W. Hu
%   and B. Si, and is available in the supplement of Hu and Si (2016)."
%   and cite the paper:
% "Hu, W., and B. Si (2016), Technical Note: Multiple wavelet coherence for untangling scale-specific and localized
%  multivariate relationships in geosciences, Hydrol. Earth Syst. Sci., volume and page numbers to be allocated ."
%
%  (C) W. Hu and B. C. Si 2016
%
% -----------------------------------------------------------
%
%  Copyright (C) 2016, W. Hu and B. C. Si 2016
%   This software may be used, copied, or redistributed as long as it is not
%   sold and this copyright notice is reproduced on each copy made.  This
%   routine is provided as is without any express or implied warranties
```

```
%  whatsoever.
%
%  Wavelet software was provided by C. Torrence and G. Compo,
%   and is available at URL: http://paos.colorado.edu/research/wavelets/.
%
%  Crosswavelet and wavelet coherence software were provided by
%    A. Grinsted and is available at URL:
%    http://www.glaciology.net/wavelet-coherence
%
%
% We acknowledge Aslak Grinsted for his code (wtcsignif.m) on
% which this code builds.
%
%-------------------------------------------------------------------
cachedir=fileparts(mfilename('fullpath'));
cachedir=fullfile(cachedir,'.cache');

%we don't need to do the monte carlo if we have a cached
%siglevel for ar1s that are almost the same. (see fig4 in Grinsted et al., 2004)
aa=round(atanh(ar1(:)')*4); %this function increases the sensitivity near 1 & -1
aa=abs(aa)+.5*(aa<0); %only positive numbers are allowed in the checkvalues (because of log)

% do a check that it is not the same as last time... (for optimization purposes)
checkvalues=single([aa dj s0/dt j1 double(mother)]); %n & pad are not important.
%also the resolution is not important.

checkhash=['' mod(sum(log(checkvalues+1)*127),25)+'a' mod(sum(log(checkvalues+1)*54321),25)+'a'];

cachefilename=fullfile(cachedir,['mwcsignif-cached-' checkhash '.mat']);

%the hash is used to distinguish cache files.
try
    last=load(cachefilename);
    if (last.mccount>=mccount) && (isequal(checkvalues,last.checkvalues))
        mwcsig=last.mwcsig;
        return
    end
catch
end

%choose a n so that largest scale have atleast some part outside the coi
ms=s0*(2^(j1*dj))/dt; %maxscale in units of samples
```

```
n=ceil(ms*6);

warned=0;
%precalculate stuff that's constant outside the loop
%d1=ar1noise(n,1,ar1(1),1);
d1=rednoise(n,ar1(1),1);
[W1,period,scale,coi] = wavelet(d1,dt,pad,dj,s0,j1,mother);
outsidecoi=zeros(size(W1));
for s=1:length(scale)
    outsidecoi(s,:)=(period(s)<=coi);
end
sinv=1./(scale');
sinv=sinv(:,ones(1,size(W1,2)));

if mccount<1
    mwcsig=scale';
    mwcsig(:,2)=.71; %pretty good
    return
end

sig95=zeros(size(scale));

maxscale=1;
for s=1:length(scale)
    if any(outsidecoi(s,:)>0)
        maxscale=s;
    else
        sig95(s)=NaN;
        if ~warned
warning('Long wavelengths completely influenced by COI. (suggestion: set n higher, or j1 lower)');
         warned=1;
        end
    end
end

%PAR1=1./ar1spectrum(ar1(1),period');
%PAR1=PAR1(:,ones(1,size(W1,2)));
%PAR2=1./ar1spectrum(ar1(2),period');
%PAR2=PAR2(:,ones(1,size(W1,2)));

nbins=1000;
wlc=zeros(length(scale),nbins);
```

```
wbh = waitbar(0,['Running Monte Carlo (significance)... (H=' checkhash ')'],'Name','Monte Carlo (MWC)');

for ii=1:mccount
    waitbar(ii/mccount,wbh);

dy=rednoise(n,ar1(1),1);
[Wdy,period,scale,coiy] = wavelet(dy,dt,pad,dj,s0,j1,mother);
sinv=1./(scale');
smdY=smoothwavelet(sinv(:,ones(1,n)).*(abs(Wdy).^2),dt,period,dj,scale);

col=size(ar1,2);

for  i=2:col
dx=rednoise(n,ar1(i),1);
 [Wdx,period,scale,coix] = wavelet(dx,dt,pad,dj,s0,j1,mother);
Wdx1(:,:,(i-1))=Wdx;
end

% -------- Calculate Cross Wavelet Spectra----------------------------

% ----between dependent variable and independent variables------

parfor i=1:(col-1)
Wdyx=Wdy.*conj(Wdx1(:,:,i));
sWdyx=smoothwavelet(sinv(:,ones(1,n)).*Wdyx,dt,period, dj,scale);
sWdyx1(:,:,i)=sWdyx;
end

% ----between independent variables and independent variables------
for i=1:(col-1);
parfor j=1:(col-1);
Wdxx=Wdx1(:,:,i).*conj(Wdx1(:,:,j));
sWdxx=smoothwavelet(sinv(:,ones(1,n)).*Wdxx,dt,period,dj,scale);
sWdxx1(:,:,i,j)=sWdxx;
end
end

% calculate the multiple wavelet coherence
for i=1:length(scale)
  parfor j=1:n
a=transpose(squeeze(sWdyx1(i,j,:)));
```

```
b=inv(squeeze(sWdxx1(i,j,:,:)));
c=conj(squeeze(sWdyx1(i,j,:)));
d=smdY(i,j);
Rsq(i,j)=real(a*b*c/d);
  end
end

 for s=1:maxscale
        cd=Rsq(s,find(outsidecoi(s,:)));
        cd=max(min(cd,1),0);
        cd=floor(cd*(nbins-1))+1;
        for jj=1:length(cd)
            wlc(s,cd(jj))=wlc(s,cd(jj))+1;
        end
    end
end
close(wbh);

for s=1:maxscale
    rsqy=((1:nbins)-.5)/nbins;
    ptile=wlc(s,:);
    idx=find(ptile~=0);
    ptile=ptile(idx);
    rsqy=rsqy(idx);
    ptile=cumsum(ptile);
    ptile=(ptile-.5)/ptile(end);
    sig95(s)=interp1(ptile,rsqy,.95);
end
mwcsig=[scale' sig95'];

if any(isnan(sig95))&(~warned)
    warning('Sig95 calculation failed. (Some NaNs)');
else
    try
        save(cachefilename,'mccount','checkvalues','mwcsig'); %save to a cache....
    catch
        warning(['Unable to write to cache file: ' cachefilename]);
    end
end
```

S4 User manual for S2 (mwc.m) and S3 (mwcsignif.m)

Multiple wavelet coherence package

by Wei Hu and Bingcheng Si

Release date: 27 April 2016

This software package is written for performing multiple wavelet coherence.

This software package includes mwc.m and mwcsignif.m, which

are written in the Matlab program based on wtc.m and wtcsignif.m provided by A.

Grinsted

(http://www.glaciology.net/wavelet-coherence).

Users are, therefore, required to download his software package and

combine these two packages into one to run the multiple wavelet coherence analysis. In the package provided by A. Grinsted, Matlab codes included are *anglemean.m*, *ar1.m*, *ar1nv.m*, *boxpdf.m*, *formatts.m*, *normalizepdf.m*, *phaseplot.m*, *smoothwavelet.m*, *wt.m*, *wtc.m*, *wtcdemo.m*, *wtcsignif.m*, *xwt.m*.

Please acknowledge the use of this software package in any publications by including

text such as:

****
The software for the multiple wavelet coherence was provided by W. Hu and B. C. Si, and is available in the supplement of Hu and Si (2016).
****
and cite the paper:

%%%

Hu, W., and B.C. Si (2016), Technical Note: Multiple wavelet coherence for untangling scale-specific and localized multivariate relationships in geosciences, Hydrol. Earth Syst. Sci., *volume and page numbers to be allocated.*

%%%

Acknowledgements:

Wavelet software was provided by C. Torrence and G. Compo, and is available at URL: http://paos.colorado.edu/research/wavelets/.

Crosswavelet and wavelet coherence software were provided by A. Grinsted and is available at URL: http://www.glaciology.net/wavelet-coherence

Should there be any enquiries, please feel free to contact:

Wei Hu
Email: wei.hu@plantandfood.co.nz

Bing Si
Email: bing.si@usask.ca

S5 Results of MEMD

Six or seven intrinsic mode functions (IMFs) corresponding to different scales are obtained for multivariate data series (i.e., a combination of the response variable with two (y_2 and y_4, or z_2 and z_4) or three (y_2, y_3, and y_4, or z_2, z_3, and z_4) predictor variables) by MEMD. Due to IMFs, with a number of 6 or greater, contributing negligible variance to the total, only the first five IMFs are presented (Fig. S1). For each IMF, the scale is calculated as the total number of points (i.e., 256) divided by the number of cycles for each IMF. The obtained scales and percentage (%) of variance explained by each IMF are shown in Table S1. While the obtained scales for the response variable *y* are in agreement with the true scales for the stationary case, the obtained scales (i.e., 3, 6, 11, 21, and 43) for the response variable *z* deviate slightly from the average scales for the non-stationary case. For the response variable, the contribution of IMFs to the total variance generally decreases (20% to 13% for stationary and 27% to 11% for non-stationary) from IMF1 to IMF5, which disagrees with the fact that each scale contributes equally (i.e., 20%) to the total variance. The scale of the dominant variance from each predictor variable can be obtained (Table S1). However, the sum of variances over all IMFs for each variable is less than 100% (ranging from 84% to 93%), indicating that MEMD cannot capture all the variances, as was also previously observed (Hu et al., 2013; She et al., 2014).

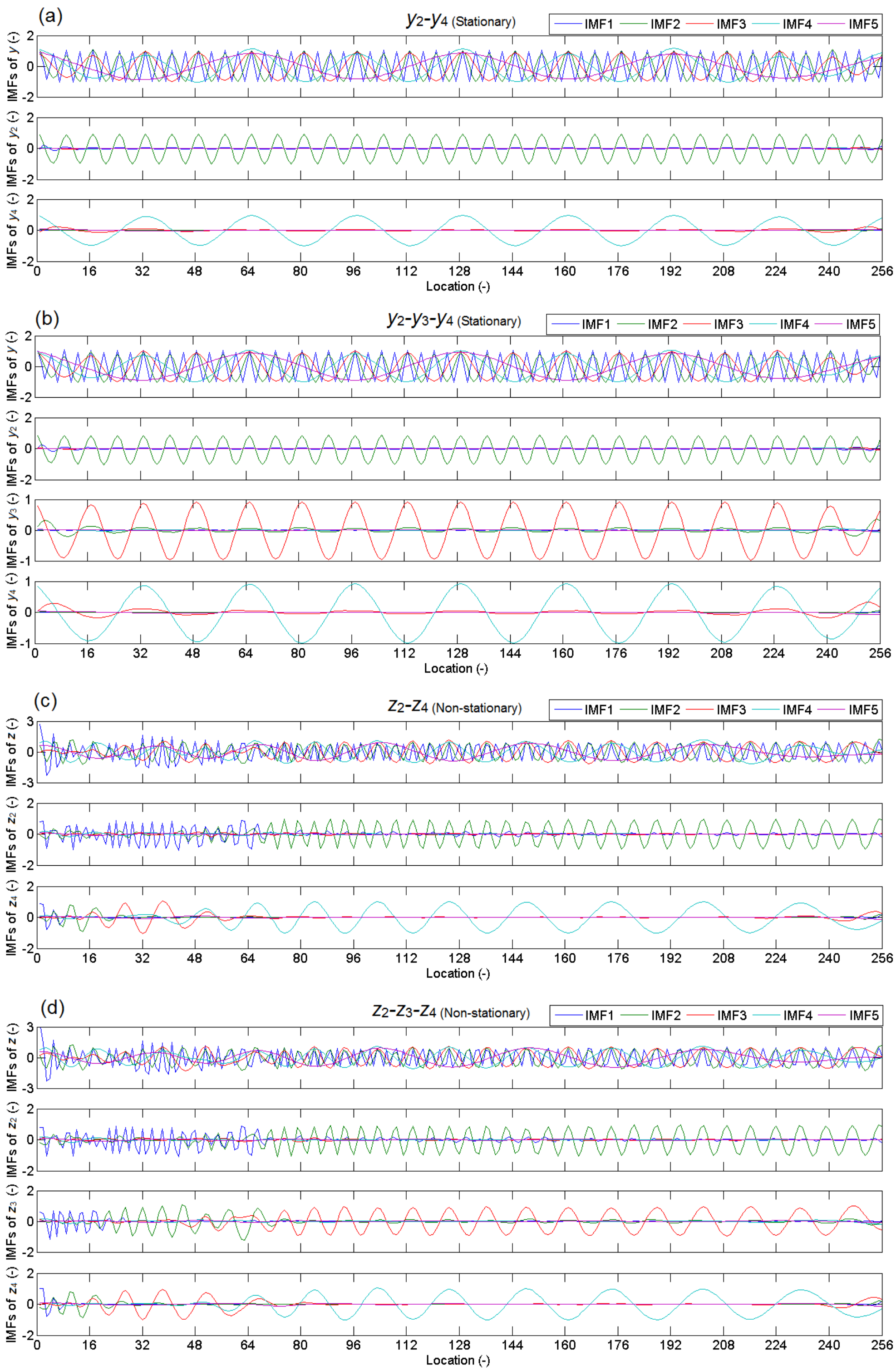

Figure S1. The first five intrinsic mode functions (IMFs) of response variable y (or z) and predictor variables (y_2 and y_4; y_2, y_3, and y_4; z_2 and z_4; or z_2, z_3, and z_4) obtained by multivariate empirical mode decomposition.

Table S1. Scales and percentage (%) of variance explained by each intrinsic mode function (IMF) of response variable *y* (or *z*) and predictor variables (y_2 and y_4; y_2, y_3 and y_4; z_2 and z_4; or z_2, z_3, and z_4) using the multivariate empirical mode decomposition method.

		Scale (-)	*y* (%)	y_2 (%)	y_3 (%)	y_4 (%)
y_2-y_4 (Stationary)	IMF1	4	20	0		0
	IMF2	8	18	**90**		0
	IMF3	16	15	0		1
	IMF4	32	18	0		**88**
	IMF5	64	13	0		0
y_2-y_3-y_4 (Stationary)	IMF1	4	20	1	0	0
	IMF2	8	17	**85**	1	0
	IMF3	16	16	0	**82**	2
	IMF4	32	16	0	0	**82**
	IMF5	64	15	0	0	0
z_2-z_4 (Non-stationary)	IMF1	3	27	22		2
	IMF2	6	17	**68**		4
	IMF3	11	17	0		11
	IMF4	21	17	0		**75**
	IMF5	43	11	0		0
z_2-z_3-z_4 (Non-stationary)	IMF1	3	27	22	7	3
	IMF2	6	18	**69**	17	4
	IMF3	11	17	0	**61**	14
	IMF4	21	16	0	1	**68**
	IMF5	43	11	0	0	0

S6 Results of bivariate wavelet coherency for *E*

The evaporation from free water surface was significantly correlated to each meteorological factor at scales of around 1 year, at all times, with exception to a certain period for relative humidity and sun hours (Fig. S2). Each of mean temperature, sun hours, and wind speed was positively correlated to *E* at different scales. For relative humidity however, its influences on *E* changed with scale. For example, at scales of around 1 year, relative humidity was positively correlated to *E* during the period of 1979 to 1997. This is due to high relative humidity usually being associated with high summer temperatures, when high evaporation occurs. At other scales (e.g., 2–6 months or 5–10 years), the relative humidity was negatively correlated to the *E*, which was expected. The dominant factors explaining variation in *E* differed with scale. For example, the relative humidity was the dominating factor at small (2–8 months) and large (>32 months) scales, while temperature was the dominating factor at the medium (8–32 months) scales (Fig. S2). The relative humidity corresponded to the greatest mean MWC (0.62) and PASC value (40%) at multiple scale-location domains.

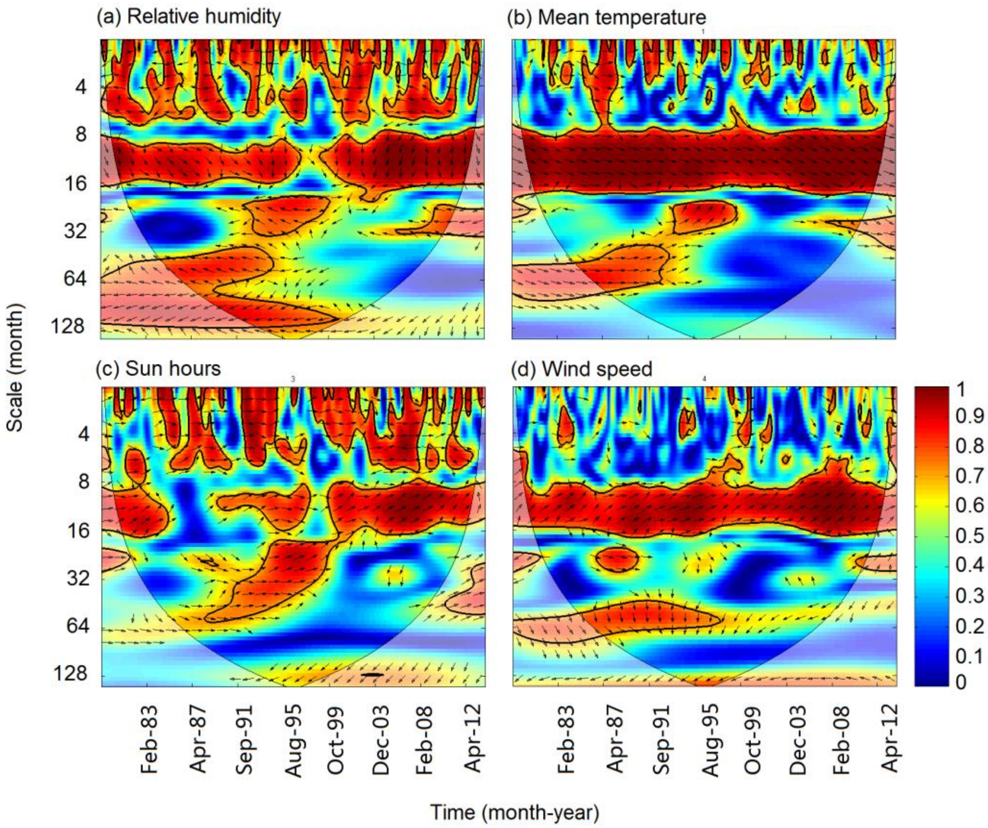

Figure S2. Bivariate wavelet coherency between evaporation (*E*) from water surfaces and each of the meteorological factors (relative humidity, mean temperature, sun hours, and wind speed) at Changwu site in Shaanxi, China. Arrows show the correlation type with the right pointing arrows being positive and left pointing arrows being negative. Thin solid lines demarcate the cones of influence and thick solid lines show the 95% confidence levels.

References

Das, N.N. and Mohanty, B. P.: Temporal dynamics of PSR-based soil moisture across spatial scales in an agricultural landscape during SMEX02: A wavelet approach, Remote Sens. Environ., 112, 522–534, doi:10.1016/j.rse.2007.05.007, 2008.

Grinsted, A., Moore, J. C., and Jevrejeva, S.: Application of the cross wavelet transform and wavelet coherence to geophysical time series, Nonlinear Proc. Geoph., 11, 561–566, 2004.

Hu, W. and Si, B. C.: Soil water prediction based on its scale-specific control using multivariate empirical mode decomposition, Geoderma, 193–194,180–188, doi: 10.1016/j.geoderma.2012.10.021, 2013.

Kumar, P. and Foufoula-Georgiou, E.: Wavelet analysis for geophysical applications, *Rev. Geophys*., 35, 385–412, doi: 10.1029/97RG00427, 1997.

She, D. L., Tang, S. Q., Shao, M. A., Yu, S. E., and Xia, Y. Q.: Characterizing scale specific depth persistence of soil water content along two landscape transects, *J. Hydrol*., *519*, 1149–1161, doi:10.1016/j.jhydrol.2014.08.034, 2014.

Si, B. C.: Spatial scaling analyses of soil physical properties: A review of spectral and wavelet methods, Vadose Zone J., 7, 547–562, doi: 10.2136/vzj2007.0040, 2008.

Torrence, C. and Compo, G. P.: A practical guide to wavelet analysis, Bull. Am. Meteorol. Soc., 79, 61–78, doi: 10.1175/1520-0477(1998)079<0061:apgtwa>2.0.co;2, 1998.

Torrence, C. and Webster, P. J.: Interdecadal changes in the ENSO-monsoon system, J. Clim., 12, 2679–2690, doi: 10.1175/1520-0442(1999)012<2679:ICITEM>2.0.CO;2, 1999.