Subject: Re: efficient comparing 1D and 3D arrays
Posted by Jelle on Wed, 11 Jun 2008 17:29:23 GMT
View Forum Message <> Reply to Message

On Jun 11, 5:37 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
> Hi,
> Yes you can...do something like:
> minVals = [15,30,12] ;Min val in each band
> maxVals = [75,80,60] ;Max val in each band
>
> allMin = rebin(minVals, nb,nl,ns) ;repeat the min band value for every
> pixels in each band
> allMax = rebin(maxVals, nb,nl,ns)
>
> goodPixels = where(imgData gt allMin and imgData lt allMax)
>
> ==> returns the indexes of imgData that satisfy the condition in EVERY band.
>
> Jean

:(unfortunately it does not return the indices I was hoping for. It
returns an array with between 0 and 75 elements, for each individual
element. So it tells me for which pixel which band matches. This I
could of course use to calculate the array_indices, subset by only
row/colum indices and reverse back into 1D positional elements, make
uniq() and then create the layer, but I am hoping there is a fell
swoop with which I can do this in one step; a way to create a [ns, nl]
bytearr with 0/1 to match the condition that each band for that bixel
falls within the desired range...

J