Subject: finding exclusive elements between two not-quite identical arrays
Posted by havok2063 on Wed, 26 Nov 2014 06:51:15 GMT
View Forum Message <> Reply to Message

So I know how to find elements in one array that are not in a second, when both arrays have identical elements.

```
A = [0,2,3,4]
B = [0,1,2,3,4]

print, where(histogram(A, omin=om) eq 0 and histogram(B,min=om) ne 0)+om
          1
```

Now I want to do the same thing, but with two arrays containing integers that aren't quite identical in each one. Some of the elements can be off by +- 1. So

```
A = [11, 19, 40]
B = [10, 20, 30, 40]
```

Doing the above should return element index 2 (30) in B that is not in A, but I don't know how to do this. Any ideas?

Thanks, Brian

Subject: Re: finding exclusive elements between two not-quite identical arrays
Posted by Nikola on Wed, 26 Nov 2014 11:05:50 GMT
View Forum Message <> Reply to Message

If the arrays are small and there is no other restriction on using loops, you can do it like this:

```
a = [11, 19, 40]
b = [10, 20, 30, 40]
c = b
margin = 5
FOR i = 0, N_ELEMENTS(b)-1 do c[i] = TOTAL(ABS((b[i]-a)) LT margin) EQ 0
PRINT, WHERE(c NE 0)
     2

a = [11, 23, 40]
b = [10, 20, 30, 40]
c = b
margin = 5
FOR i = 0, N_ELEMENTS(b)-1 do c[i] = TOTAL(ABS((b[i]-a)) LT margin) EQ 0
PRINT, WHERE(c NE 0)
     2

a = [11, 23, 40]
```

```
b = [10, 20, 30, 40]
c = b
margin = 2
FOR i = 0, N_ELEMENTS(b)-1 do c[i] = TOTAL(ABS((b[i]-a)) LT margin) EQ 0
PRINT, WHERE(c NE 0)
    1     2
```

Where margin obviously specify how close elements of A and B should be to consider them matching. I don't see a quick solution to do this with histograms especially if b does not have to be equispaced. The method above work well in that case as well:

```
a = [11, 35, 40]
b = [10, 17, 33, 40]
c = b
margin = 5
FOR i = 0, N_ELEMENTS(b)-1 do c[i] = TOTAL(ABS((b[i]-a)) LT margin) EQ 0
PRINT, WHERE(c NE 0)
    1
```

On Wednesday, November 26, 2014 6:51:19 AM UTC, Brian Cherinka wrote:
> So I know how to find elements in one array that are not in a second, when both arrays have identical elements.
>
> A = [0,2,3,4]
> B = [0,1,2,3,4]
>
> print, where(histogram(A, omin=om) eq 0 and histogram(B,min=om) ne 0)+om
>   1
>
> Now I want to do the same thing, but with two arrays containing integers that aren't quite identical in each one. Some of the elements can be off by +- 1. So
>
> A = [11, 19, 40]
> B = [10, 20, 30, 40]
>
> Doing the above should return element index 2 (30) in B that is not in A, but I don't know how to do this. Any ideas?
>
> Thanks, Brian

Subject: Re: finding exclusive elements between two not-quite identical arrays
Posted by Heinz Stege on Wed, 26 Nov 2014 11:37:48 GMT

View Forum Message <> Reply to Message

Hi Brian,

here is a quick draft:

```
A = [11, 19, 40]
B = [10, 20, 30, 40]
ii=value_locate(a,b)   ; needs a to be sorted
delta=1
jj=where(b-a[ii] gt delta and a[ii+1]-b gt delta,count)   ; needs the
                    ; compile option strictarrsubs NOT to be set
if count ge 1 then print,'Part II: ',b[jj]
kk=where(a[0]-b gt delta,count)   ; use min(a) if a is not sorted
if count ge 1 then print,'Part I: ',b[kk]
```

Good luck, Heinz

Subject: Re: finding exclusive elements between two not-quite identical arrays
Posted by Phillip Bitzer on Wed, 26 Nov 2014 14:15:40 GMT
View Forum Message <> Reply to Message

On Wednesday, November 26, 2014 12:51:19 AM UTC-6, Brian Cherinka wrote:
> So I know how to find elements in one array that are not in a second, when both arrays have identical elements.
>
> A = [0,2,3,4]
> B = [0,1,2,3,4]
>

You can also try Craig's CMSET_OP:
http://www.physics.wisc.edu/~craigm/idl/arrays.html

Subject: Re: finding exclusive elements between two not-quite identical arrays
Posted by havok2063 on Wed, 26 Nov 2014 19:59:20 GMT
View Forum Message <> Reply to Message

On Wednesday, November 26, 2014 6:37:34 AM UTC-5, Heinz Stege wrote:
> Hi Brian,
>
> here is a quick draft:
>
> A = [11, 19, 40]
> B = [10, 20, 30, 40]
> ii=value_locate(a,b)   ; needs a to be sorted
> delta=1
> jj=where(b-a[ii] gt delta and a[ii+1]-b gt delta,count)   ; needs the
>   ; compile option strictarrsubs NOT to be set
> if count ge 1 then print,'Part II: ',b[jj]

> kk=where(a[0]-b gt delta,count) ; use min(a) if a is not sorted
> if count ge 1 then print,'Part I: ',b[kk]
>
> Good luck, Heinz

Thanks guys. The solution offered by Heinz works best for me. I was also looking for a simple non-loop solution. I like Craig's functions, but I think CMSET_OP only works on identical elements. Thanks again everyone.
