
Subject: Re: Finding Common Elements in Two Arrays

Posted by [stl](#) on Mon, 06 Jun 1994 08:09:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

Good morning,

My program, where_array() that I posted on friday has a few bugs. I will modify it with some of the ideas from below.

NOTE: I beleive the below posting has an error in it:

In article <1994Jun3.131050.3790@mksol.dseg.ti.com> landers@tsunami.dseg.ti.com writes:

> It's me again...

>

> Now I noticed you want the indices rather than the values.... So

>

>

> In article <1994Jun3.130243.3453@mksol.dseg.ti.com>, landers@tsunami.dseg.ti.com (David Landers) writes:

>

> |>

> |> Na = N_ELEMENTS(a)

> |> Nb = N_ELEMENTS(b)

> |> L = LINDGEN(Na,Nb)

> |> AA = A(L MOD Na)

> |> BB = B(L / Na)

>

> Then...

>

> I = AA EQ BB

> Ia = UNIQUE(I MOD Na)
 ^^^

> Ib = UNIQUE(I / Na)
 ^^^

>

> Now Ia has the indices of A that are in B, and Ib has the indices of B that

> are in A. Leave off the UNIQUE if you care that an element matches multiple

> times.

I believe the above two lines with the unique(i ...) should be as follows:

Ia = unique(where(i) mod na)

Ib = unique(where(i) / na)

then clearly the common elements are just a(ia) and b(ib)

I will post a more stable routine with these new ideas. This is indeed

better because it should handle string arrays, i too believe it to be faster.

-stephen Strebel

--

Stephen C Strebel / SKI TO DIE
stl@maz.sma.ch / and
Swiss Meteorological Institute, Zuerich / LIVE TO TELL ABOUT IT
01 256 93 85 / (and pray for snow)
