
Subject: Re: How to get matching elements of array efficiently
Posted by [Jeremy Bailin](#) on Wed, 24 Feb 2010 14:46:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 23, 3:36 pm, pp <pp.pente...@gmail.com> wrote:

> On Feb 23, 5:11 pm, JJ <j...@cornell.edu> wrote:

>

>> I'm looking for an efficient (non-loop) solution to this problem. I

>> have two arrays A, B. The elements of A are not necessarily unique,

>> nor is A necessarily sorted. I'd like to find all the indices in A

>> that match values that occur in B.

>

>> For example, if A = [7,1,8,7,8], B = [7,8], the result should be

>> [0,2,3,4].

>

>> A and B can have many (order millions) of unique values, so I'd rather

>> avoid a loop if I can.

>

>> Is there any way to do this efficiently?

>

> Are your values integers as in the example? If they are, and if there

> are no big gaps in B (the number of elements of B is not much smaller

> than max(B)-min(B)), histogram comes to mind:

>

> h=histogram(A,binsize=1,min=min(b),max=max(b),reverse_indices=ri)

> if (max(h) gt 0L) then res=ri[n_elements(h)+1:.*] else (deal with the

> case of none found in b)

>

> Which would give the result you want in res, but ordered by bin (their

> order in B). In this example, res would be [0,3,2,4].

Of course, I'll have to pipe in here and say that if your values aren't integers, or if there are large gaps in them, you can use `UNIQ` and `VALUE Locate` to map them into a set of consecutive integers, and then use `Histogram`. :-)=

-Jeremy.
