
Subject: Re: Inexpensive / free-ware similar to IDL?
Posted by [George White](#) on Sat, 02 Oct 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 21 Sep 1999, K. S. Balasubramaniam wrote:

- > I am looking for inexpensive solutions for data analysis, other than
- > IDL, but similar in capability.
- >
- > IDL has been a wonderful tool and has enormous strengths, but
- > individuals cannot afford it. I have used it for over a decade while
- > I worked at govt. sponsored/educational labs. Its flexibility is
- > addictive. Having moved over to the private sector, and switching
- > fields, in a different area of data analysis things are different!
- > Many folks have not heard of IDL, and when informed of its use are
- > able to nod in appreciation but quickly loose interest because it is
- > way too pricey! and very compareable to the price of some
- > workstations.
- >
- > I am not sure how many folks out there use ANA <http://ana.lmsal.com/>
- > It has the potential for a robust freeware and needs lots of work.
- > Are there similar freeware packages? I am also aware of xmgrace,
- > xmgr, etc for less intensive applications. Suggestions appreciated

There are lots of open source tools whose functionality overlaps IDL. PerlIDL and IBM's DataExplorer are two recent additions to the list. Omegahat is a new and promising project with roots in the statistical community (<http://www.omegahat.org>). Many open source tools are designed to perform a limited task well and to intact with other "small" tools. To get the capabilities of IDL you would several such tools and a system (e.g. unix) that supports this approach over the single large app model.

As hardware gets cheaper, software costs should increase because the software can be more capable. These days, it is reasonable for software on a typical workstation to cost more than the hardware. Open source software may not have direct costs, but there is still a very real cost associated with each package you install on your system. For most of us, monetary costs are secondary to the losses associated with time spend learning a package that turned out to be ill-suited to the problem or had a show-stopping bug.

In my experience, there is a certain level of reluctance to invest in new software unless the user has strong indications that it will prove useful. Many vendors offer demo versions of their software, but open source software has the advantage that a user can work with it on their own schedule, get updates to fix bugs and incompatibilities, etc. It may impossible to make time to really explore the capabilities of a new package

during a two-week trial period, but not difficult to spend a few minutes a day over a period of several months getting acquainted with a new package.

Matrix languages overlapping in functionality with IDL are proliferating, and it is not always easy to match project requirements with a suitable language. My current project, for example, involves 2.5 million pixel images with many channels of varying data types (byte, 16-bit ints, and floats). Many channels have missing value codes (some defined in a way that depends on a particular floating point implementation!).

Some of the programs used in this project were designed to use ASCII CSV files. IDL (on SGI irix 6.5 R4000) bogs down with the CSV files and requires ad-hoc programming to deal with missing value codes, but perhaps the biggest problem is that IDL allocates most of the free memory and appears to do some sort of garbage collection within that arena. Some of the processing takes overnight, but too often I discover next morning that some other process wasn't able to allocate memory and was killed. For my project, handling missing values, support for 16-bit images, and robust memory management are key considerations.

Other large projects will have different requirements.

There are a number of options open to me:

1. seek alternatives to IDL that do "better" in some respects
2. rewrite and restructure parts of the system to better match the capabilities profile of IDL
3. implement critical sections in a compiled language to get better control over time/space tradeoffs and memory usage.

One of the motivations for developing open source software is to get around a limitation of an existing tool. The R stats package (modelled after S-plus) has a very different approach to memory management than S-plus. Octave (modelled after matlab) supports basic unix IPC (popen, waitpid, etc.) that is lacking in matlab.

--

George White <gnw3@acm.org> tel: 902.426.8509
Bedford Inst. of Oceanography, Nova Scotia, Canada.
