ENTREZ DIRECT - XML DATA EXTRACTION

The Entrez Direct (EDirect) xtract function uses command-line arguments to direct the selective conversion of XML data into a tab-delimited table. The -pattern argument divides the results into rows, while placement of data into columns is controlled by -element.

Additional arguments limit exploration to specified XML regions (-group, -block, and -subset), filter by data content (-match and -avoid) or elements (-present and -absent), and customize the report presentation. Individual fields can be grouped with commas and given specific prefix, suffix, and separator characters.

Tabular data can be piped to standard UNIX utilities or custom scripts for further processing. A query by specific PubMed unique identifiers:

```
  efetch -db pubmed -id 781293,2678811,6301692,8332518 -format xml | \
  xtract -pattern PubmedArticle -element "MedlineCitation/PMID" \
    -block AuthorList -sep "|" -element LastName "#Author" \
    -block PubDate -sep " " -element Year,Month MedlineDate \
    -block DateCreated -sep "-" -element Year,Month,Day | \
  sort -t $'\t' -k 3,3n | column -s $'\t' -t
```

produces a table that allows easy parsing of author names, counts the number of authors present, and prints the date each record was published and the date it was entered into PubMed, sorting the results by the computed author count:

```
  781293   Casadaban                         1  1976 Jul      1976-10-02
  6301692  Krasnow|Cozzarelli                2  1983 Apr      1983-06-17
  8332518  Benson|Lipman|Ostell              3  1993 Jul      1993-08-17
  2678811  Mortimer|Schild|Contopoulou|Kans  4  1989 Sep-Oct  1989-11-22
```

(The PubDate object can exist either in structured or string form, but would not contain a mixture of both types, so "-element Year,Month MedlineDate" will only contribute a single column to the output.)

Sequence records can be retrieved in an XML version of the GenBank or GenPept flatfile. Feature and qualifier names are data values, not XML tags, and require -match to select the desired object. A query for snail venom mature peptides:

```
  esearch -db protein -query "conotoxin AND mat_peptide [FKEY]" | \
  efetch -format gpc -mode xml | \
  xtract -pattern INSDSeq -ACCN INSDSeq_accession-version \
    -group INSDFeature -match ">mat_peptide<" \
        -avoid "<INSDFeature_partial" -pfx "\n" -element "&ACCN" \
      -block INSDQualifier -match ">peptide<" -element "%INSDQualifier_value" \
      -block INSDQualifier -match ">product<" -element INSDQualifier_value \
      -block INSDQualifier -match ">peptide<" -element INSDQualifier_value | \
  grep conotoxin | sort -t $'\t' -u -k 3,4 | \
  sort -t $'\t' -k 2,2n -k 3,3f | column -s $'\t' -t
```

calculates the length of each mature peptide, and prints the product name and peptide sequence, removing redundant entries and sorting by peptide length:

```
  ADB43130.1  15  conotoxin Cal 1a       KCCKRHHGCHPCGRK
  ADB43131.1  15  conotoxin Cal 1b       LCCKRHHGCHPCGRT
  AAO33169.1  16  alpha-conotoxin GIC    GCCSHPACAGNNQHIC
  ADB43128.1  16  conotoxin Cal 5.1      DPAPCCQHPIETCCRR
  AAD31913.1  18  alpha A conotoxin Tx2  PECCSHPACNVDHPEICR
  ...
```

(The sequence accession is captured in a variable for use with each mat_peptide in the record. Prefix substitution ensures that every peptide is shown on a separate line. This also works for multi-product precursor proteins such as proinsulin.)

(Incorporating fragments of the surrounding XML tags in -match arguments prevents coincidental matches to free text in note qualifiers.)

Multiple -avoid or -match conditions are specified with -and and -or commands:

```
  -group INSDFeature -avoid ">proprotein<" -and ">sig_peptide<" \
    -block INSDQualifier -match ">calculated_mol_wt<" -or ">peptide<" \
```

Setting a variable default value, by placing text in parentheses, can be used to prevent columns from shifting when data fields are missing:

```
  -MLWT "(-)"
```

Visual inspection of the XML structure with the -outline argument helps determine the appropriate xtract arguments to employ. A pubmed summary query:

```
  esummary -db pubmed -id 16589597
```

retrieves an XML document summary:

```
  <eSummaryResult>
    <DocumentSummarySet status="OK">
      <DocumentSummary uid="16589597">
        <Id>16589597</Id>
        <PubDate>1954 Dec</PubDate>
        <EPubDate></EPubDate>
        <Source>Proc Natl Acad Sci U S A</Source>
        <Authors>
          <Author>
            <Name>Garber ED</Name>
            ...
```

that can be piped through:

```
  xtract -pattern DocumentSummary -outline
```

to return an outline that shows XML structure and tag names, but without closing tags, attributes, or content values:

```
  DocumentSummary
    Id
    PubDate
    EPubDate
    Source
    Authors
      Author
        Name
        AuthType
        ClusterID
    LastAuthor
    Title
    ...
```

The outline view presents a clear, uncluttered picture of the XML hierarchy. Copy and paste from the -outline output to xtract arguments (-element, -group, -match, etc.) can help avoid typographical errors.