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 xtract arguments are used for limiting exploration to specified regions of the XML (-group, -block, and -subset), filtering by data content (-match and -avoid), and customizing the report presentation. Individual fields can be grouped with commas and given specific prefix, suffix, and separator characters.

Visual inspection of the XML contents helps determine the appropriate arguments to employ. 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 | 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
ADB43127.1  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 "(-)"
```

The normal column-oriented output can be extensively modified to produce a custom report. A query on a gene that undergoes messenger RNA splicing:

```
  efetch -db nuccore -id "GQ370762.1" -format gbc -mode xml | \
  xtract -pattern INSDSeq -pfx ">Feature " -sfx "\n" -tab "" -first INSDSeqid \
    -group INSDFeature -avoid "INSDFeature_key>source<" -FKEY INSDFeature_key \
      -block INSDInterval -element INSDInterval_from INSDInterval_to \
        INSDInterval_point INSDInterval_point "&FKEY" -FKEY "" -tab "" -lbl "\n" \
      -block INSDQualifier -avoid ">translation<" -and ">peptide<" -pfx "\t\t\t" \
        -sfx "\n" -tab "" -element INSDQualifier_name,INSDQualifier_value
```

generates the 5-column feature table format used for GenBank submissions:

```
  >Feature gb|GQ370762.1|
  51      1474    gene
                              gene            HBB
                              allele          GZ
  51      142     mRNA
  273     495
  1346    1474
                              gene            HBB
                              allele          GZ
                              product         beta-globin
  51      142     CDS
  273     495
  1346    1474
                              gene            HBB
                              allele          GZ
                              codon_start     1
                              transl_table    1
                              product         beta-globin
                              protein_id      ACU56984.1
                              db_xref         GI:256028940
  310     310     variation
                              gene            HBB
                              note            K43E
                              replace         g
```

(A location interval will have either an INSDInterval_from and INSDInterval_to pair or a single INSDInterval_point. Additional work would be required to support the 5' and 3' partial flags for features with incomplete locations.)