SRA nucleotide search expressions

Summary

Our parser accepts ASCII expressions as queries, and packed NCBI2na strings as database input.

An expression grammar summary follows:

```
expr          : unary_expr
              | unary_expr boolean_op expr

unary_expr    : primary_expr
              | '!' unary_expr

primary_expr  : fasta_expr
              | '^' fasta_expr
              | fasta_expr '$'
              | '(' expr ')'

fasta_expr    : FASTA
              | "'" FASTA "'"
              | '"' FASTA '"'

boolean_op    : '&', '|', '&& ', '||'

FASTA         : [ACGT ]<1,61>
              | [-ACMGRSVTWYHKDBN ]<1,29>
```

The grammar supports expressions involving one or more FASTA strings with optional operators and sub-expressions. The strings are case insensitive, and belong to either 2NA or 4NA alphabets. The 2NA alphabet has a limit of 61 characters per string, while the 4NA alphabet has a limit of 29. These limits are due to internal implementation details.

Expressions are evaluated left to right, where 'AND' and 'OR' operations have identical precedence. White space is ignored.

All expression punctuation is subject to revision and/or requests for change/enhancement. In particular, where the existing punctuation causes confusion or difficulties for the end user or web scripting, there may be changes.

Details

FASTA

FASTA expressions are case insensitive character strings from either the 2na or 4na character sets. Their evaluation is left to right within a sequence until a match occurs or the right edge of the sequence has been met. 2na queries apply an exact match algorithm, while 4na queries are more flexible. These expressions have an implementation-imposed upper limit on length, which for Intel 64 bit architectures is currently 61 characters for 2na and 29 for 4na.

fasta_expr

FASTA strings are normally not quoted. However, this production allows for gratuitous single or double quoting. Correctness is enforced, but the quotes are otherwise ignored.

primary_expr

This production allows for a simple fasta_expr, an anchored fasta_expr, or a complete sub-expression. An anchored fasta_expr restricts matching to either the beginning ('^') or the end ('$') of a sequence; internal only matches always return false.

unary_expr

This production allows for a primary_expr or its Boolean negation.

expr with Boolean operators

There are only two Boolean operators supported (although others are easily added): 'AND' and 'OR'. They are symbolically represented with either single or double versions of their corresponding symbols; there is no difference made between single or double. All operators have identical precedence, and are evaluated left to right.

An 'AND' operation ('&' or '&&') will first evaluate the left operand. Iff it returns true, the right operand will be evaluated.

Likewise, the 'OR' operation ('|' or '||') will evaluate first the left operand, and only if it returns false, then evaluate the right operand.