

## split-tags

```

#
# state machine loop to read fasta format sequences, make them into
# perl character strings, and then process them
#
while ( <> )                                # keep reading lines
{
    if ( />(.*)/ )                           # is it a FASTA >header?
    {
        $new_header = $1;                      # handle current sequence
        process( $seq, $header ) if ( defined( $header ) );
        $header = $new_header;                  # reset sequence header
        $seq = "";
    }
    else
    {
        chomp;
        s/\s/g;
        $seq .= uc( $_ );
    }
}
process( $seq, $header ) if ( defined( $header ) ); #handle last sequence

sub process
{
    my ( $seq, $header ) = @_;
    my $pos = 0;
    if ( $seq =~ s/^$anch// )                 # trim any beginning anchor seq
    {
        $pos += $anch_len; }                   # (affects position counter)
    $seq =~ s/$anch//;                      # trim any trailing anchor sequences
    @raw_tags = split( /$anch/, $seq ); # capture sequences between anch sites

    foreach $raw ( @raw_tags )
    {
        if ( $raw =~ /$punc_r.*$punc$/ )
        {
            $true_tag = $raw;      # tag begins and ends with punctuation,
            $dir = "a";           # so we can't determine direction.
        }
        elsif ( $raw =~ /^($punc$/ )
        {
            $true_tag = $1;       # this is clearly a forward tag
            $dir = "f";
        }
        elsif ( $raw =~ /(^$punc(.*)$) )
        {
            $true_tag = $rc($1); # this is clearly a reverse tag
            $dir = "r";
        }
        else
        {
            $true_tag = $raw;
            $dir = "b"           # GGG direction marker
        }
        my $len = length( $true_tag );
        $len -= $punc_len if ( $dir eq 'a' || $dir eq 'b' );
        if ( $len >= $low_ext && $len <= $high_ext )
        {
            if ( ( ($dir eq 'a') && $opt_A ) ||
                ( ($dir eq 'b') && $opt_B ) ||
                ( ($dir eq 'f') && $opt_F ) ||
                ( ($dir eq 'r') && $opt_R ) )
            {
                printf "%-25.25s %s %d %s\n", $true_tag, $dir, $pos, $header;
            }
            $n_raw++;
            $pos += length( $raw ) + $anch_len;
        }
    }
}

```

punctuation → anchor enzyme sequence →

CATGGGGCCGCCCTATTTCATGAAATCGCGAATTCAAAGGCATGAA  
 ATCGAACATCATCATTGCCATGAAAGGATGTCATTGATATCGGCATGAA  
 TGGTTAATCAAGATGCCATGAAAATAAGGTCTGACGTCCATGAAGGGG  
 CGTTGGCAGGATGGCATGAAAGCCCCAAAGGCTTTTGCCTCATGTTCCGTCC  
 CATATTGCACCTTCATGCCAAAGAAGCTGGCTCTTTCATGTTTGCGGCA  
 GGCTAAATTCATGAAAGCAATACATCTGGTTCCATGAAAGAACCGAACCTC  
 CAGATCATGAAACAATCAGTACATCGCTGTATGCCCTCCAGGAAGCCTGT  
 TTTCATGAAATCTGGCTGTCGCCAATTTCATGACTGTAACCAATTTCCTTC  
 ATGGTCGCGCGCATACCTGCTTCATGAAAGCCCACATCCTGTCGCTACATG

**Three polarity cases:**

GTTGGGGCCGCCCTATTT

**Forward**

AAATCGCGAATTCAAAGC

**Reverse**

AATCTGGCTGTCGCCAATT

**Ambiguous - tag begins with AA.**  
**could be AATCTGGCTGTCGCCAA or**  
**AATTGGCGACAGCCAG**