

More Biopython

SeqRecord

- ▶ `.annotations`: dictionary of fields
- ▶ `.features`: list of positional annotations (`SeqFeatures`)
- ▶ Mainly Genbank
- ▶ Information on using them is scant

SeqFeature

- ▶ location
- ▶ type
- ▶ strand
- ▶ id
- ▶ ref
- ▶ sub_features

Example

mRNA complement (<49223..49300,49780..>50208)
/gene="F28B23.12"

location <49223..49300,49780..>50208

type mRNA

strand -1

ref None

ref_db None

sub_features None

Positions

“Fuzzy”

exact e.g. 210

before e.g. <210

after e.g. >210

within inclusive e.g. (210..230)

between exclusive e.g. 210^230

Locations

Can construct locations from positions.

```
>>> from Bio import SeqFeature
>>> start_pos = SeqFeature.AfterPosition (50)
>>> end_pos = SeqFeature.BetweenPosition(80, 113)
>>> my_locn = SeqFeature.FeatureLocation(start_pos, end_pos)
>>> exact_locn = SeqFeature.FeatureLocation (50, 80)
```

Extracting locations

```
>>> print my_location
[>50:(80^113)]
>>> my_location.start
<Bio.SeqFeature.AfterPosition instance at 0x101d7164>
>>> print my_location.start
>50
>>> print my_location.end
(80^113)
>>> my_location.nofuzzy_start
50
>>> my_location.nofuzzy_end
80
```

Alignments

- ▶ Reading - there's several ways!
- ▶ Several alignment classes, see `Align` and `Clustal`
- ▶ Functions for aligning

```
from Bio import SeqIO
handle = open ("test.aln")
for record in SeqIO.parse (handle, "clustal") :
    print record.id
```

```
al = Clustalw.parse_file ("test.aln")
```

Aligning

Requires physical file (write to /tmp?)

```
from Align.Clustalw import MultipleAlignCL
aligner = MultipleAlignCL ('opuntia.fasta')
aligner.set_output ('out.aln')
alignment = Clustalw.do_alignment (aligner)
```

Alignment properties

- ▶ Basically a list of SeqRecords
- ▶ `.get_all_seqs()`
- ▶ `.get_alignment_length()`
- ▶ `.get_seq_by_num (<n>)`
- ▶ `.get_column (<col>)`
- ▶ `.add_sequence (descriptor, sequence, start, end, weight)`

Summary information

- ▶ Generate AlignInfo objects from alignment
- ▶ `.dumb_consensus (<threshold>, <ambiguous-symbol>)`
- ▶ `.pos_specific_score_matrix (<consensus>, <chars_to_ignore>)`
- ▶ `.information_content()`

```
from Bio.Align import AlignInfo
summary = AlignInfo.SummaryInfo (alignment)
```

Resources

- ▶ [BioPython cookbook](#)
- ▶ [Loading Genbank files](#)
- ▶ [BioPython wiki on SeqIO](#)