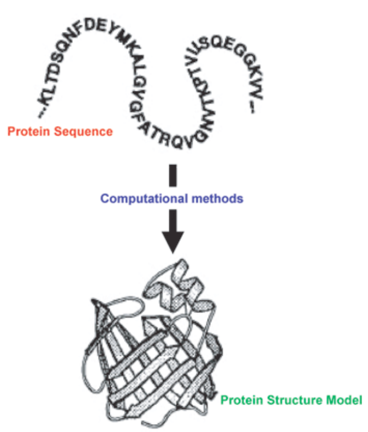
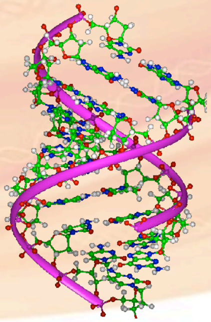


Bioinformatics Computational Methods 1 - BIOL 6308



Million years
3.4
5.1
Million years
7.0
2.2
H aactggtacg tggtcg tacgaactgg H
C aaccggttcg cggtcg ttcgaaccgg C
G aacctgttcg ctgtcg ttcgaacctg G
O atctggtacg tggtcg tacgatctgg O

November 21st 2013

http://155.33.203.128/cleslin/home/teaching6308F2013.php

Databases are Redundant

So how can I remove redundancy

Databases are Redundant

- Biological reasons
  - Some protein functions, or sequence motifs are more common than others
- Laboratory artifacts
  - Some protein families:
    - have been heavily investigated
    - others not
  - Mutagenesis studies create large and almost identical replications in the database
  - This bias is non-biological

Sequence Clustering

- Grouping related sequences based on some set thresholds such as length, % identity, composition etc
- % identity is the most commonly used criterion to remove redundant sequences in the databases
- Helps improve speed of database searches in the orders of magnitude with minimal loss of content
- General principle in clustering is pairwise alignment of sequences in all-to-all combination
  - **Reduce the overall size of the database**
  - **without removing any sequence information by only removing 'redundant' (or highly similar) sequences**
- Most commonly used tools are
  - **BLASTClust**
  - **cd-hit**
  - **Hobohm**
  - Skipredundant (EMBOSS TOOLS – Very Slow)
    - http://emboss.sourceforge.net/apps/release/6.6/emboss/apps/skipredundant.html

BLASTClust

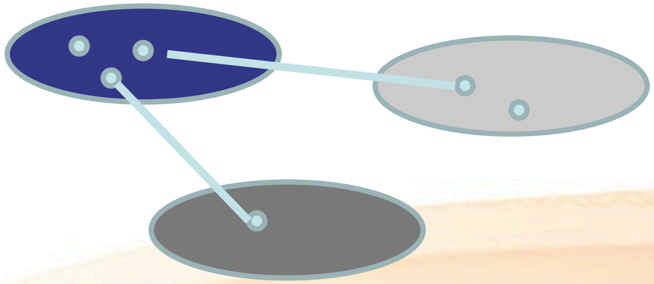
- Program within **standalone BLAST package** used to cluster either protein or dna sequences
- Begins with pairwise matches and places a sequence in a cluster if the sequence matches at least one sequence already in the cluster
- Proteins - BLASTp algorithm is used to compute the pairwise matches
- Nucleotide - MegaBLAST algorithm is used

- **BLAST score-based single-linkage clustering**
- All sequences in the database are compared pairwise in all-to-all combinations, based on the BLAST score

BLASTClust – Single Linkage Clustering Algorithm

- Similarity between closest neighbors meets a threshold
- “If A is related to B, and B is related to C, then A is related to C.”

Genomes



- BLASTClust default values were used:
  - Length coverage threshold = 0.9
  - Score coverage threshold (bit score / length if < 3.0, percentage of identities otherwise) = 1.75

What BLASTClust Does

- BLASTClust formats the input sequence to produce a temporary BLAST database
- Performs the clustering, and removes the database at completion
- Hence, there is no need to run formatdb in advance to use BLASTClust
- Output of BLASTClust **consists of a file, one cluster to a line, of sequence identifiers separated by spaces**
- Clusters are sorted from the largest cluster to the smallest

- Hundreds of times slower than CD-HIT

BLASTClust

- To produce the non-redundant set, one might use:

```
blastclust -i infile -o outfile -p F -L .9 -b T -S 95
```

- Sequences in "infile" will be clustered - results will be written to "outfile"
- Sequences are identified as:
  - nucleotide -p F
  - -p T protein (default)
- Two sequences will need to be 95% identical (-S 95) over an area covering 90% of the length (-L .9) of each sequence (-b T)
- Using "-b F" instead of "-b T" – enforce alignment length threshold on only one member of a sequence pair
- "S", used here to specify the percent identity
  - Can also be used to specify "score density."
    - Equivalent to the BLAST score divided by the alignment length
    - If "S" is given as a number between 0 and 3, it is interpreted as a score density threshold; otherwise it is interpreted as a percent identity threshold.

CD-HIT

CD-HIT - Cluster Database at High Identity with Tolerance

- Clustering sequence DB requires all-by-all comparisons
  - **Time consuming**
  - Many methods use **BLAST to compute the all vs. all similarities**
  - Difficult cluster large DBs
- Program (cd-hit) takes:
  - Fasta format protein file as input
  - Produces a set of 'non-redundant' representative sequences as output
  - Outputs a cluster file
    - documenting the sequence 'groups' for each nr sequence representative
- Produces a set of closely related protein/dna familes from a given fasta sequence database

http://www.bioinformatics.org/cd-hit/

CD-HIT Algorithm

- CD-HIT skips many pairwise sequence alignments with short word filter
- **Greedy incremental clustering algorithm method**
- Sequences are first sorted in order of decreasing length
- The longest one becomes representative of the first cluster
- Then, each remaining sequence is compared to the representatives of existing clusters
- If the similarity with any representative is above a **given threshold**, it is grouped into that cluster
- Otherwise, a new cluster is defined with that sequence as the representative

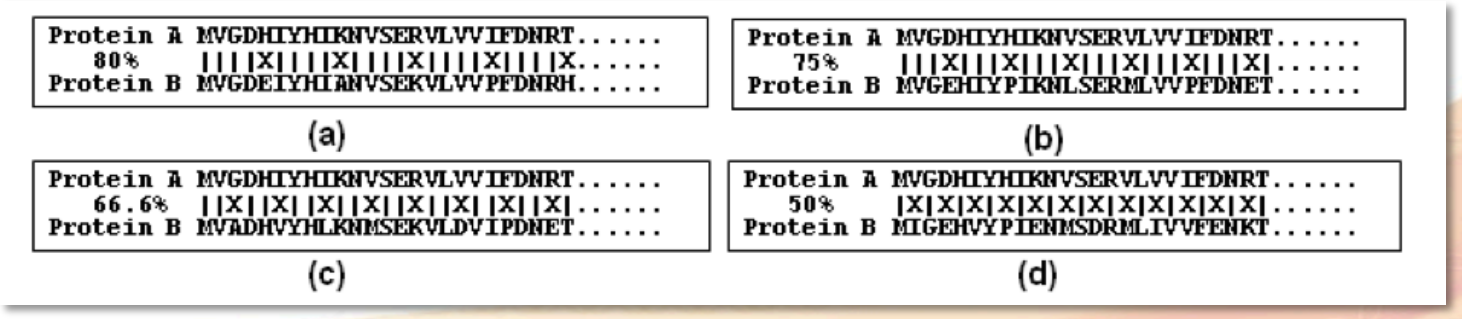
http://www.bioinformatics.org/cd-hit/cd-hit-user-guide.pdf

Short Word Filter Works

- Two proteins, with a certain sequence identity, must have at least a specific number of identical dipeptides, tripeptides and etc
- e.x.
  - For two sequences to have 85% identity over a 100 residue window
  - They have to have at least 70 identical dipeptides, 55 identical tripeptides, and 25 identical pentapeptides
- CD-HIT skips most pairwise alignments because it knows that the similarity of two sequences is below certain threshold by **simple word counting**

Algorithm Limitations (1)

- When mismatches are evenly distributed along the alignment, the numbers of common short words are minimal



```
Protein A MVGDHIYHIKNVSERVLVVIFDNRT......
   80%    ||||X||||X||||X||||X||||X......
Protein B MVGDEIYHIANVSEKVLVVPFDNRH......
```

(a)

```
Protein A MVGDHIYHIKNVSERVLVVIFDNRT......
   75%    |||X|||X|||X|||X|||X|||X|......
Protein B MVGEHIYPIKNLSERMLVVPFDNET......
```

(b)

```
Protein A MVGDHIYHIKNVSERVLVVIFDNRT......
  66.6%   ||X||X||X||X||X||X||X||X|......
Protein B MVADHVYHLKNMSEKVLDVIPDNET......
```

(c)

```
Protein A MVGDHIYHIKNVSERVLVVIFDNRT......
   50%    |X|X|X|X|X|X|X|X|X|X|X|X|......
Protein B MIGEHVYPIENMSDRMLIVVFENKT......
```

(d)

- Short word filtering is limited to certain clustering thresholds
- **Evenly distributed mismatches** are shown in alignments with 80%, 75%, 66.67% and 50% sequence identities (above)
- The # of common pentapeptides in (a), tetrapeptides in (b), tripeptides in (c), and dipeptides in (d) can be zero

Biological Sequences Are Not Lines of Random Letters

- Proteins usually have more:
  - Conserved regions
  - Diverse regions as the result of **specific constraints of evolution**
- Situations (previous slide) very rare in the real world
  - The actual number of **common short words** is much **higher** than in the **worst case scenarios**
- Large scale statistical analysis on short words conducted
  - Even at 70% identity, sequences still have statistically significant number of common pentapeptides
  - Current CD-HIT is based on this short word statistics
- But the short word filters are still limited to certain thresholds
- Reasonable limits of clustering thresholds for pentapeptide, tetrapeptide, tripeptide and dipeptide are ~70%, 60%, 50% and 40%, respectively

Algorithm Limitations (2)

- Introduced by the **greedy incremental clustering**
- Let say, there are two clusters:
  - Cluster #1
    - has A, X and Y
    - where A is the representative
  - Cluster #2
    - has B and Z
    - where B is the representative
- Even if **Y** is more similar to **B** than to **A**, it will still be in cluster #1, simple because **Y** first hit **A** during clustering process

Running the CD-HIT - Proteins

```
# cd-hit -i InfluenzaA_Human_Nov_16_2012.MP.prot.fasta -o
   InfluenzaA_Human_Nov_16_2012.MP.prot.fasta.cdhit.c99.fasta -c 0.99 -n
   5 -M 3000
```

`-i InfluenzaA_Human_Nov_16_2012.MP.prot.fasta` is the filename of input

`-o InfluenzaA_Human_Nov_16_2012.MP.prot.fasta.cdhit.c99.out` is output

`-c 0.99` means 99% identity,

clustering threshold 5 is the size of word `-n 5`

`-M` (max memory in MB)

Choose of word size:

- -n 5 for thresholds 0.7 ~ 1.0
- -n 4 for thresholds 0.6 ~ 0.7
- -n 3 for thresholds 0.5 ~ 0.6
- -n 2 for thresholds 0.4 ~ 0.5

```
total seq: 13570
longest and shortest : 272 and 11
Total letters: 3297473
Sequences have been sorted

Approximated minimal memory consumption:
Sequence        : 4M
Buffer          : 1 X 10M = 10M
Table           : 1 X 65M = 65M
Miscellaneous   : 0M
Total           : 81M

Table limit with the given memory limit:
Max number of representatives: 4000000
Max number of word counting entries: 364837619

comparing sequences from          0  to      13570
..........    10000  finished         97  clusters
...
    13570  finished        154  clusters

Apprixmated maximum memory consumption: 81M
writing new database
writing clustering information
program completed !

Total CPU time 0.97
```

data can be found here /data/METHODS/Fall/LECT15/

Running the CD-HIT – DNA/RNA

```
# cd-hit-est -i InfluenzaA_Human_Nov_16_2012.MP.nt.fasta -o
   InfluenzaA_Human_Nov_16_2012.MP.nt.fasta.cdhit.c99.fasta -c 0.99 -n 8 -M
   3000 -r 1
```

`-i InfluenzaA_Human_Nov_16_2012.MP.nt.fasta` is the filename of input

`-o InfluenzaA_Human_Nov_16_2012.MP.nt.fasta.cdhit.c99.out` is output

`-c 0.99` means 99% identity,

clustering threshold 5 is the size of word `-n 5`

`-M` (max memory in MB)

`-r 1` or `0`, default 0, if set to 1, comparing both strand (++, +-)

Choose of word size:

- -n 8,9,10 for thresholds 0.9 ~ 1.0
- -n 7 for thresholds 0.88 ~ 0.9
- -n 6 for thresholds 0.85 ~ 0.88
- -n 5 for thresholds 0.80 ~ 0.85
- -n 4 for thresholds 0.75 ~ 0.80

```
total seq: 13983
longest and shortest : 1121 and 23
Total letters: 13055533
Sequences have been sorted

Approximated minimal memory consumption:
Sequence        : 14M
Buffer          : 1 X 12M = 12M
Table           : 1 X 1M = 1M
Miscellaneous   : 0M
Total           : 28M

Table limit with the given memory limit:
Max number of representatives: 4194304
Max number of word counting entries: 371378763

comparing sequences from          0  to      13983
..........    10000  finished        188  clusters
...
    13983  finished        297  clusters

Apprixmated maximum memory consumption: 31M
writing new database
writing clustering information
program completed !

Total CPU time 13
```

data can be found here /data/METHODS/Fall/LECT15/

Other Algorithms in CD-HIT

- **cd-hit-2d** same short word filtering and index table used for comparing two data sets (proteins) and reporting the matches between 2 datasets over a certain similarity   threshold
- **choose wordsizes as you would for cd-hit**

```
cd-hit-2d -i Neisseria.meningitidis.NC_010120.1.CDS.protein.fasta  -i2
    Neisseria.gonorrhoeae.NC_002946.2.CDS.protein.fasta -c 0.9 -n 5 -o Neisseria.shared.prot.fasta
```

- `-i` and `-i2` are inputs
- `Neisseria.shared.prot.fasta` is the output
- `-c 0.9`, means 90% identity,
- `-n 5` the comparing threshold 5 is the size of word

data can be found here /data/METHODS/Fall/LECT15/

ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Escherichia_coli_K_12_substr__MG1655_uid57779/NC_000913.faa

Other Algorithms in CD-HIT

- **cd-hit-est-2d** same as CD-HIT-2D but for nucleotides
- **choose wordsizes as you would for cd-hit-est**

```
cd-hit-est-2d -i Neisseria.meningitidis.NC_010120.1.CDS.dna.fasta -i2
    Neisseria.gonorrhoeae.NC_002946.2.CDS.dna.fasta  -c 0.90 -r 1 -n 10 -o Neisseria.shared.dna.fasta
```

- `-i` and `–i2` are inputs
- `Neisseria.shared.dna.fasta` is the output
- `-c 0.9`, means 90% identity,
- `-n 10` the comparing threshold 10 is the size of word

data can be found here /data/METHODS/Fall/LECT15/

UniRef

- Clustered sets of sequences from UniProt and selected UniParc records
- Provides complete coverage of sequence space
- While hiding redundant sequences (but not their descriptions) from view
- UniRef100 database combines **identical sequences** and **sub-fragments with 11 or more residues** (from any organism) into **a single UniRef entry**
  - Displaying the sequence of a representative protein
  - The accession numbers of all the merged entries
  - Links to the corresponding UniProtKB and UniParc records

http://www.uniprot.org/help/uniref

UniRef

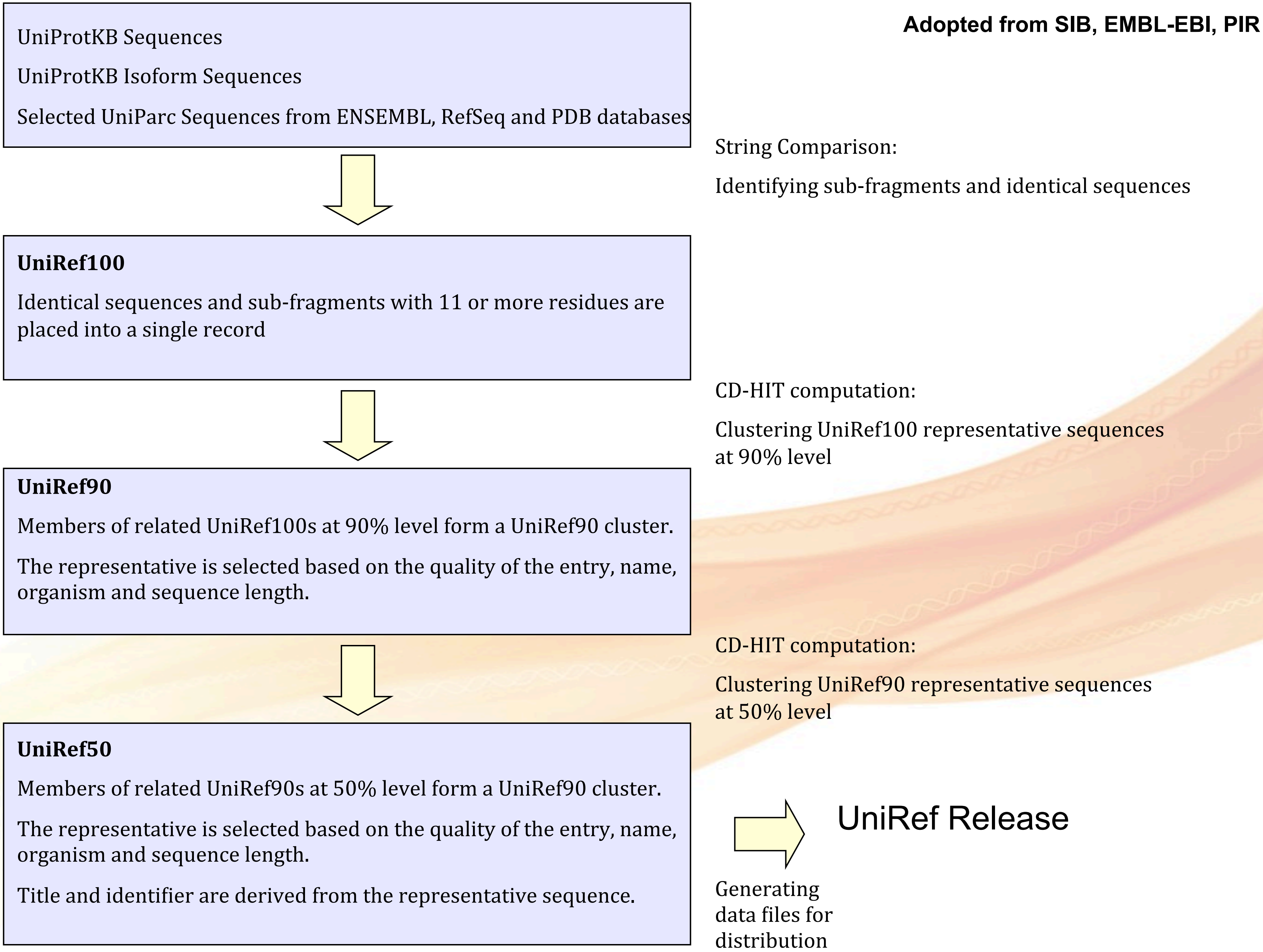
- Each cluster is composed of sequences that have at least 90% or 50% sequence identity, respectively, to the longest sequence (UniRef seed sequence)
- UniRef90 and UniRef50 yield a database size reduction of approximately 40% and 65%, respectively, providing for significantly faster sequence searches
- UniRef90 and UniRef50 yield a database size reduction of approximately 58% and 79%, respectively
- Getting the data http://www.ebi.ac.uk/uniref/

http://www.uniprot.org/help/uniref

UniRef - Representative

- The sequences are ranked as follows:
  - Quality of the entry: member entries from UniProtKB/Swiss-Prot are preferred
  - Meaningful name (entries with names that do not contain words such as hypothetical, probable, etc. are preferred)
  - Organism (entries from model organisms preferred)
  - Length of the sequence (longest sequence preferred)

http://www.uniprot.org/help/uniref



Adopted from SIB, EMBL-EBI, PIR

UniProtKB Sequences

UniProtKB Isoform Sequences

Selected UniParc Sequences from ENSEMBL, RefSeq and PDB databases

String Comparison:

Identifying sub-fragments and identical sequences

UniRef100

Identical sequences and sub-fragments with 11 or more residues are placed into a single record

CD-HIT computation:

Clustering UniRef100 representative sequences at 90% level

UniRef90

Members of related UniRef100s at 90% level form a UniRef90 cluster.

The representative is selected based on the quality of the entry, name, organism and sequence length.

CD-HIT computation:

Clustering UniRef90 representative sequences at 50% level

UniRef50

Members of related UniRef90s at 50% level form a UniRef90 cluster.

The representative is selected based on the quality of the entry, name, organism and sequence length.

Title and identifier are derived from the representative sequence.

UniRef Release

Generating data files for distribution

UniRef100, 90, 50

- **Generated by placing "UniRef100_" prefix before UniProtKB accession or UniParc identifier of the representative UniProt or UniParc entry, e.g. "UniRef100_Q8WZ42" or "UniRef100_UPI0000000F90"**

- UniRef90 cluster titles and identifiers are derived from the representative UniRef100 entry
  - **The UniRef90 identifier is generated by replacing "UniRef100_" prefix of the representative with "UniRef90_". e.g. "UniRef90_Q8WZ42"**
- UniRef50 cluster titles and identifiers are derived from the representative UniRef90 entry
  - **The UniRef50 identifier is generated by replacing "UniRef100_" prefix of the representative with "UniRef50_". e.g. "UniRef50_Q10466"**

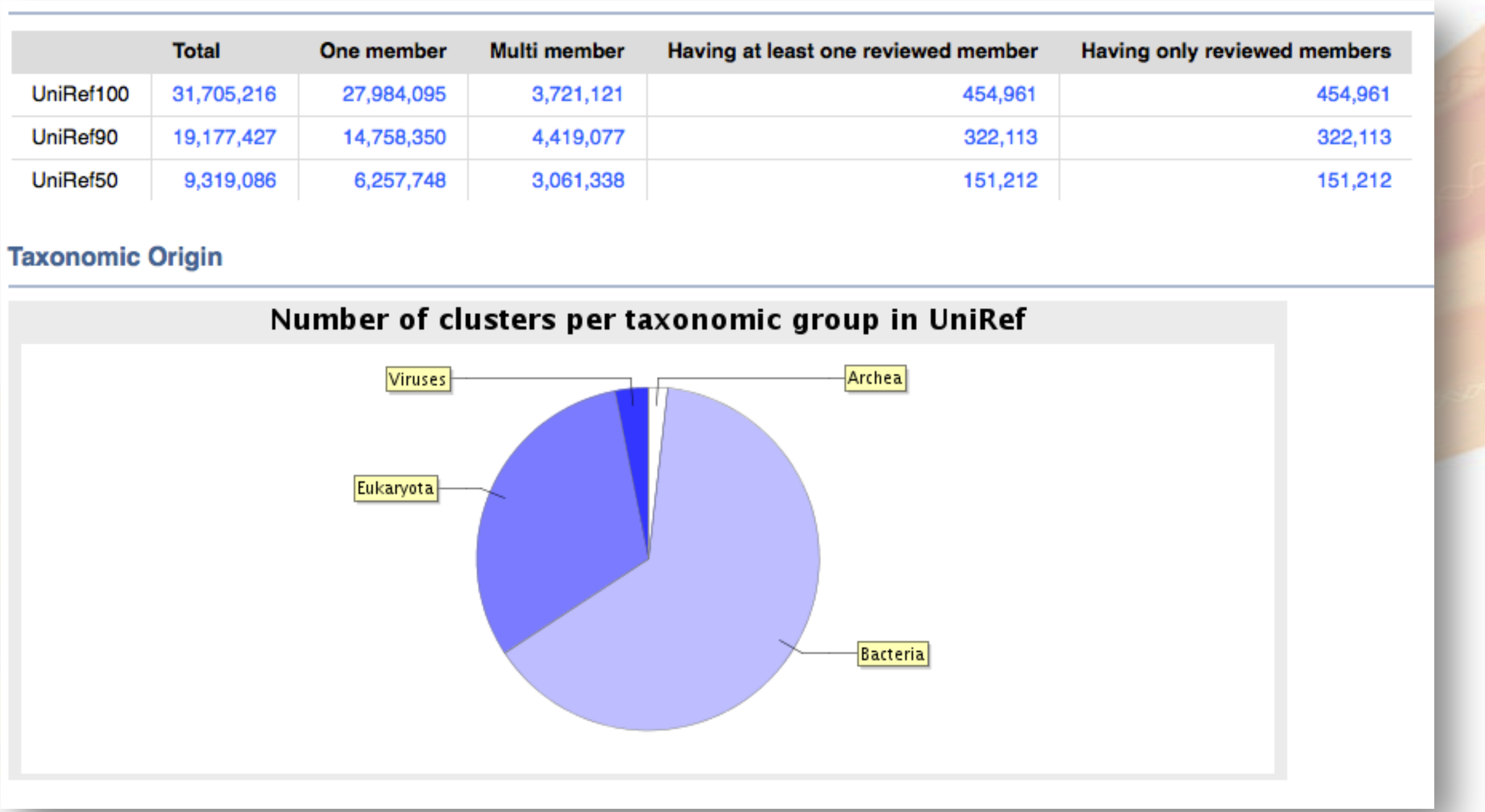
http://www.uniprot.org/help/uniref

UniRef

- Speeding up similarity search
- Reducing bias in homology searches by providing more even sequence space
- Using the clusters for family classification
- Using the clusters to annotate EST and other sequence databases
- Using the clusters to check the consistency of UniProtKB annotations

UniRef Release Statistics

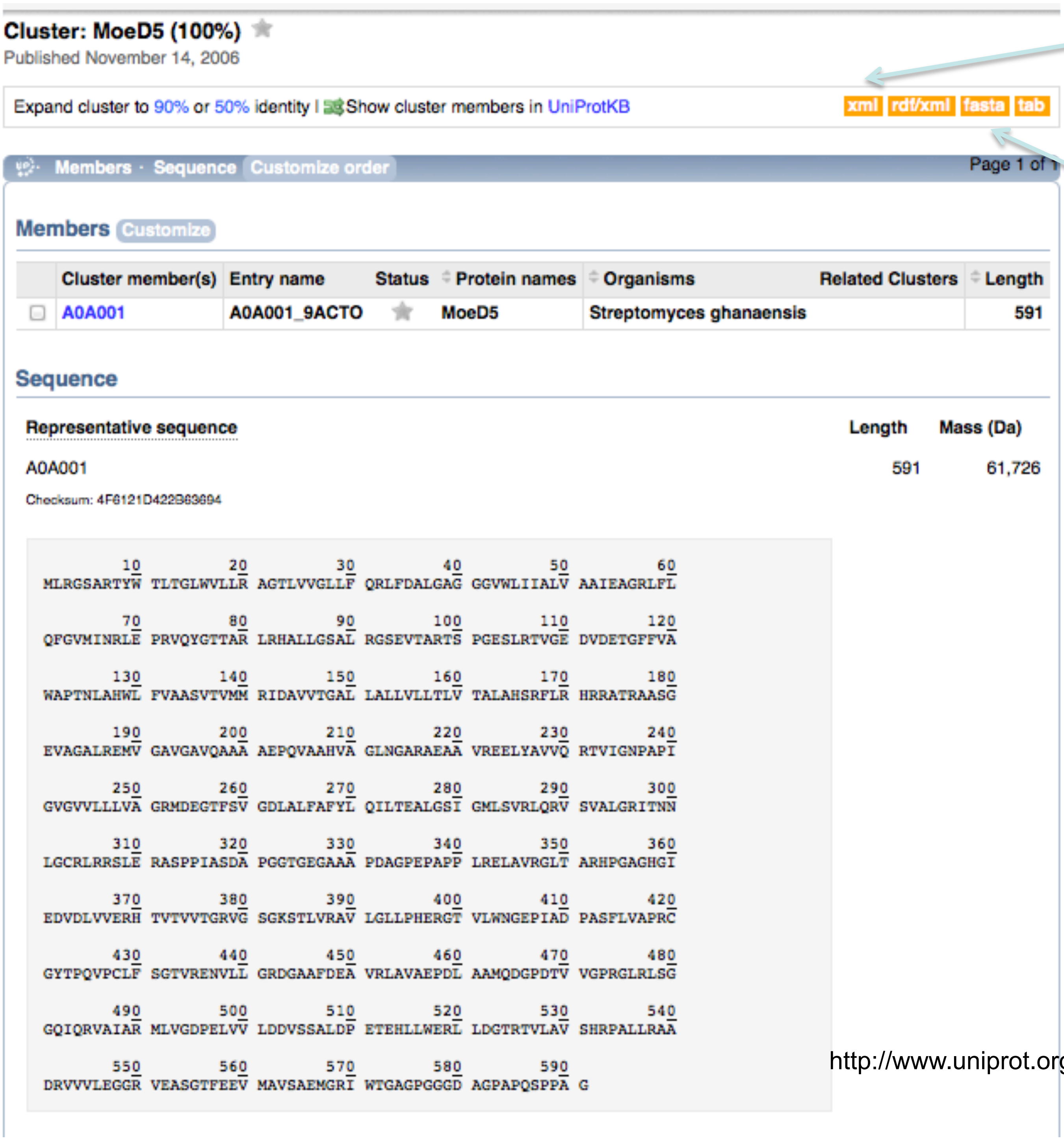
- http://www.uniprot.org/statistics/UniRef



	Total	One member	Multi member	Having at least one reviewed member	Having only reviewed members
UniRef100	31,705,216	27,984,095	3,721,121	454,961	454,961
UniRef90	19,177,427	14,758,350	4,419,077	322,113	322,113
UniRef50	9,319,086	6,257,748	3,061,338	151,212	151,212

Taxonomic Origin

Number of clusters per taxonomic group in UniRef



Cluster: MoeD5 (100%)

Published November 14, 2006

Expand cluster to 90% or 50% identity | Show cluster members in UniProtKB

Members

	Cluster member(s)	Entry name	Status	Protein names	Organisms	Related Clusters	Length
□	A0A001	A0A001_9ACTO	★	MoeD5	Streptomyces ghanaensis		591

Sequence

Representative sequence	Length	Mass (Da)
A0A001	591	61,726

Checksum: 4F6121D422B63694

```
        10         20         30         40         50         60
MLRGSARTYW TLTGLWVLLR AGTLVVGLLF QRLFDALGAG GGVWLIIALV AAIEAGRLFL

        70         80         90        100        110        120
QFGVMINRLE PRVQYGTTAR LRHALLGSAL RGSEVTARTS PGESLRTVGE DVDETGFFVA

       130        140        150        160        170        180
WAPTNLAHWL FVAASVTVMM RIDAVVTGAL LALLVLLTLV TALAHSRFLR HRRATRAASG

       190        200        210        220        230        240
EVAGALREMV GAVGAVQAAA AEPQVAAHVA GLNGARAEAA VREELYAVVQ RTVIGNPAPI

       250        260        270        280        290        300
GVGVVLLLVA GRMDEGTFSV GDLALFAFYL QILTEALGSI GMLSVRLQRV SVALGRITNN

       310        320        330        340        350        360
LGCRLRRSLE RASPPIASDA PGGTGEGAAA PDAGPEPAPP LRELAVRGLT ARHPGAGHGI

       370        380        390        400        410        420
EDVDLVVERH TVTVVTGRVG SGKSTLVRAV LGLLPHERGT VLWNGEPIAD PASFLVAPRC

       430        440        450        460        470        480
GYTPQVPCLF SGTVRENVLL GRDGAAFDEA VRLAVAEPDL AAMQDGPDTV VGPRGLRLSG

       490        500        510        520        530        540
GQIQRVAIAR MLVGDPELVV LDDVSSALDP ETEHLLWERL LDGTRTVLAV SHRPALLRAA

       550        560        570        580        590
DRVVVLEGGR VEASGTFEEV MAVSAEMGRI WTGAGPGGGD AGPAPQSPPA G
```

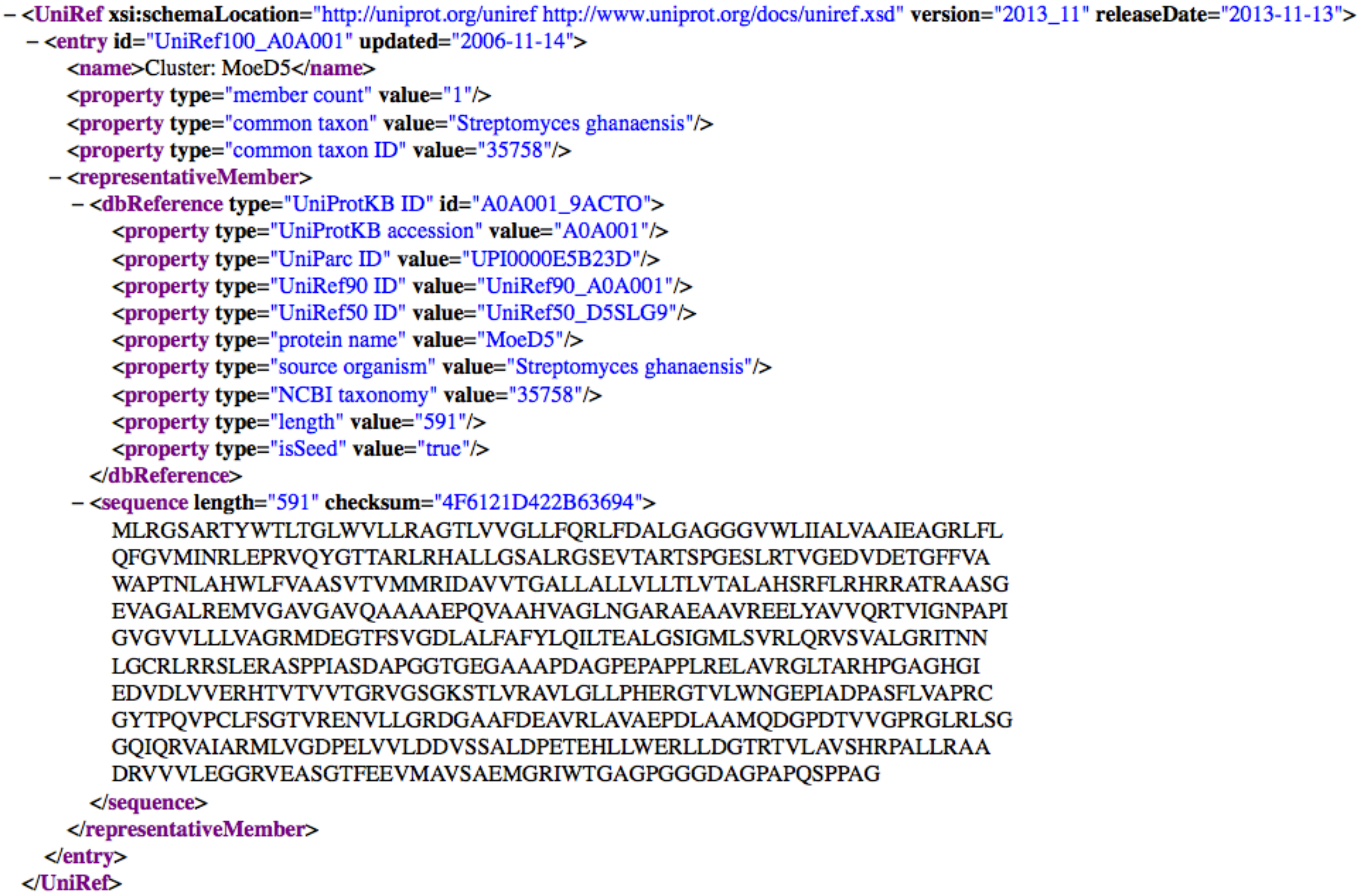
http://www.uniprot.org/uniref/UniRef100_A0A001

Fasta

```
>UniRef100_A0A001 MoeD5 n=1 Tax=Streptomyces ghanaensis RepID=A0A001_9ACTO
MLRGSARTYWTLTGLWVLLRAGTLVVGLLFQRLFDALGAGGGVWLIIALVAAIEAGRLFL
QFGVMINRLEPRVQYGTTARLRHALLGSALRGSEVTARTSPGESLRTVGEDVDETGFFVA
WAPTNLAHWLFVAASVTVMMRIDAVVTGALLALLVLLTLVTALAHSRFLRHRRATRAASG
EVAGALREMVGAVGAVQAAAAEPQVAAHVAGLNGARAEAAVREELYAVVQRTVIGNPAPI
GVGVVLLLVAGRMDEGTFSVGDLALFAFYLQILTEALGSIGMLSVRLQRVSVALGRITNN
LGCRLRRSLERASPPIASDAPGGTGEGAAAPDAGPEPAPPLRELAVRGLTARHPGAGHGI
EDVDLVVERHTVTVVTGRVGSGKSTLVRAVLGLLPHERGTVLWNGEPIADPASFLVAPRC
GYTPQVPCLFSGTVRENVLLGRDGAAFDEAVRLAVAEPDLAAMQDGPDTVVGPRGLRLSG
GQIQRVAIARMLVGDPELVVLDDVSSALDPETEHLLWERLLDGTRTVLAVSHRPALLRAA
DRVVVLEGGRVEASGTFEEVMAVSAEMGRIWTGAGPGGGDAGPAPQSPPAG
```

http://www.uniprot.org/uniref/UniRef100_A0A001.fasta

XML

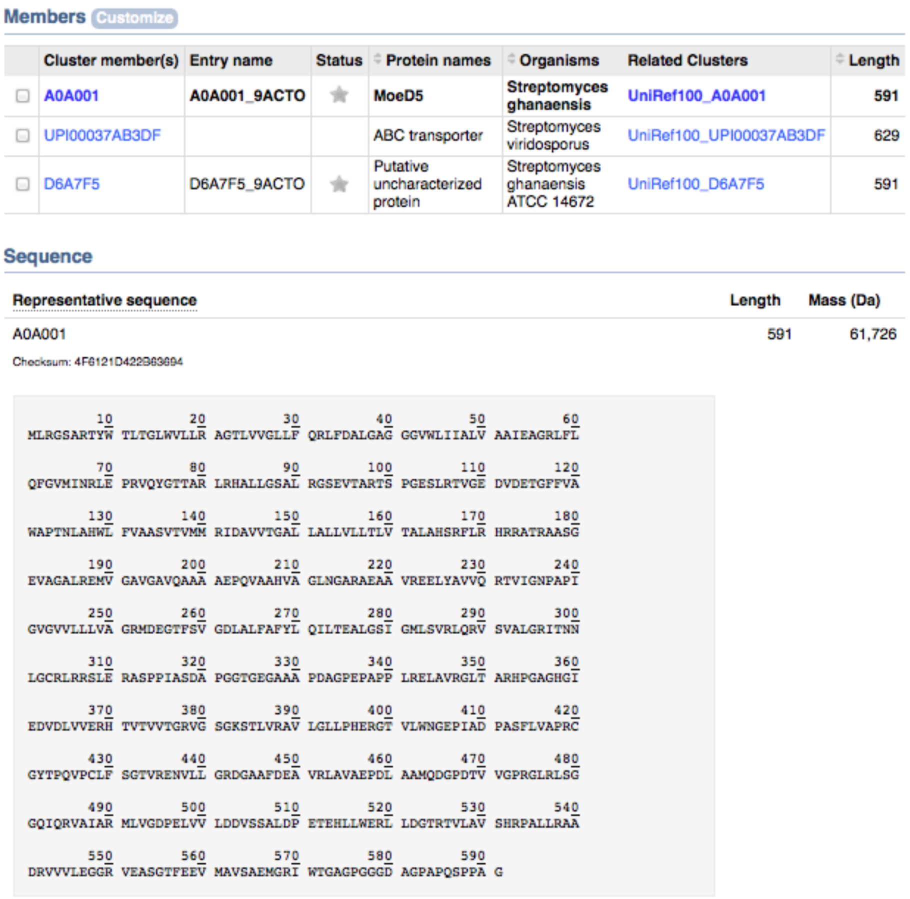


```
– <UniRef xsi:schemaLocation="http://uniprot.org/uniref http://www.uniprot.org/docs/uniref.xsd" version="2013_11" releaseDate="2013-11-13">
  – <entry id="UniRef100_A0A001" updated="2006-11-14">
      <name>Cluster: MoeD5</name>
      <property type="member count" value="1"/>
      <property type="common taxon" value="Streptomyces ghanaensis"/>
      <property type="common taxon ID" value="35758"/>
    – <representativeMember>
      – <dbReference type="UniProtKB ID" id="A0A001_9ACTO">
          <property type="UniProtKB accession" value="A0A001"/>
          <property type="UniParc ID" value="UPI0000E5B23D"/>
          <property type="UniRef90 ID" value="UniRef90_A0A001"/>
          <property type="UniRef50 ID" value="UniRef50_D5SLG9"/>
          <property type="protein name" value="MoeD5"/>
          <property type="source organism" value="Streptomyces ghanaensis"/>
          <property type="NCBI taxonomy" value="35758"/>
          <property type="length" value="591"/>
          <property type="isSeed" value="true"/>
        </dbReference>
      – <sequence length="591" checksum="4F6121D422B63694">
          MLRGSARTYWTLTGLWVLLRAGTLVVGLLFQRLFDALGAGGGVWLIIALVAAIEAGRLFL
          QFGVMINRLEPRVQYGTTARLRHALLGSALRGSEVTARTSPGESLRTVGEDVDETGFFVA
          WAPTNLAHWLFVAASVTVMMRIDAVVTGALLALLVLLTLVTALAHSRFLRHRRATRAASG
          EVAGALREMVGAVGAVQAAAAEPQVAAHVAGLNGARAEAAVREELYAVVQRTVIGNPAPI
          GVGVVLLLVAGRMDEGTFSVGDLALFAFYLQILTEALGSIGMLSVRLQRVSVALGRITNN
          LGCRLRRSLERASPPIASDAPGGTGEGAAAPDAGPEPAPPLRELAVRGLTARHPGAGHGI
          EDVDLVVERHTVTVVTGRVGSGKSTLVRAVLGLLPHERGTVLWNGEPIADPASFLVAPRC
          GYTPQVPCLFSGTVRENVLLGRDGAAFDEAVRLAVAEPDLAAMQDGPDTVVGPRGLRLSG
          GQIQRVAIARMLVGDPELVVLDDVSSALDPETEHLLWERLLDGTRTVLAVSHRPALLRAA
          DRVVVLEGGRVEASGTFEEVMAVSAEMGRIWTGAGPGGGDAGPAPQSPPAG
        </sequence>
      </representativeMember>
    </entry>
  </UniRef>
```

http://www.uniprot.org/uniref/UniRef100_A0A001.xml

Clusters

90%



Members Customize

	Cluster member(s)	Entry name	Status	Protein names	Organisms	Related Clusters	Length
□	A0A001	A0A001_9ACTO	★	MoeD5	Streptomyces ghanaensis	UniRef100_A0A001	591
□	UPI00037AB3DF			ABC transporter	Streptomyces viridosporus	UniRef100_UPI00037AB3DF	629
□	D6A7F5	D6A7F5_9ACTO	★	Putative uncharacterized protein	Streptomyces ghanaensis ATCC 14672	UniRef100_D6A7F5	591

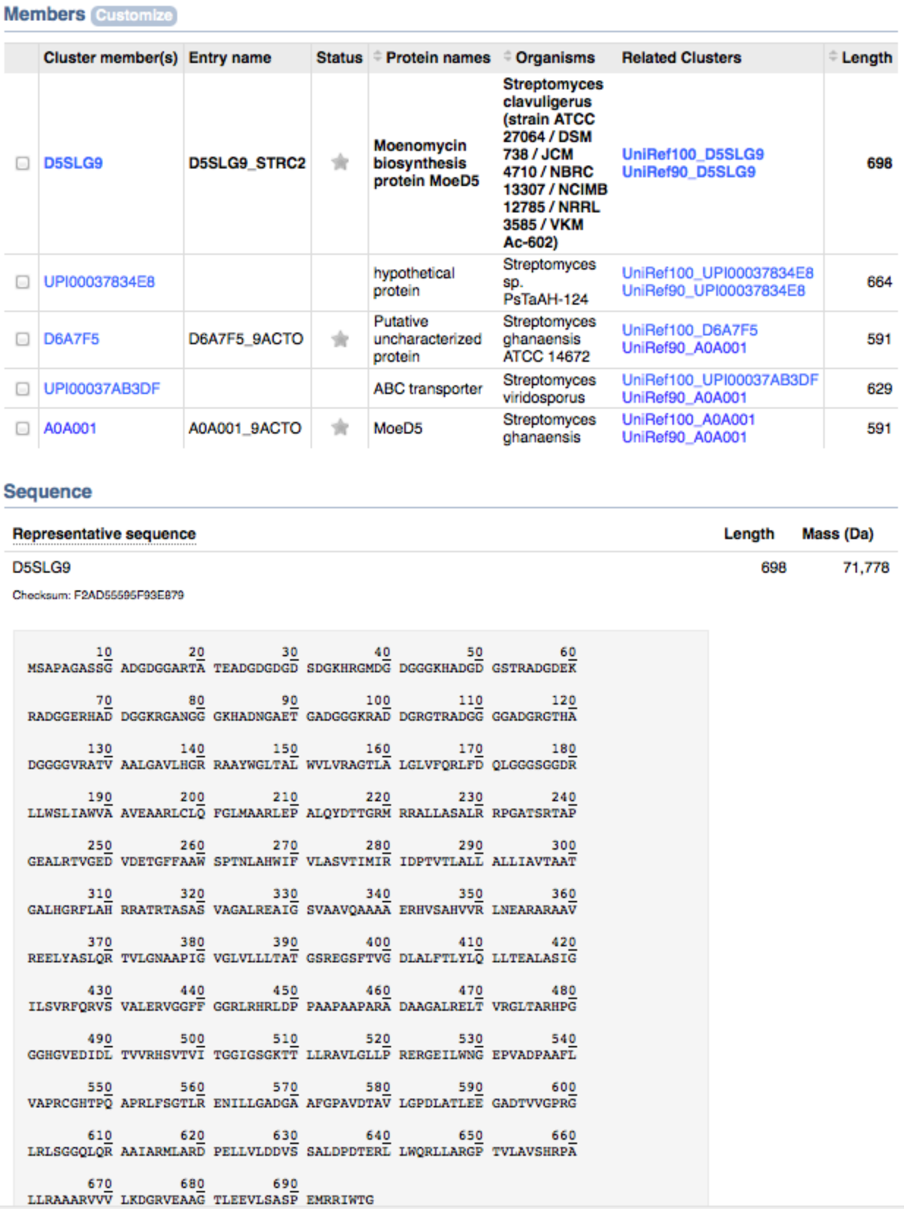
Sequence

Representative sequence	Length	Mass (Da)
A0A001	591	61,726

Checksum: 4F6121D422B63694

```
        10         20         30         40         50         60
MLRGSARTYW TLTGLWVLLR AGTLVVGLLF QRLFDALGAG GGVWLIIALV AAIEAGRLFL

        70         80         90        100        110        120
QFGVMINRLE PRVQYGTTAR LRHALLGSAL RGSEVTARTS PGESLRTVGE DVDETGFFVA

       130        140        150        160        170        180
WAPTNLAHWL FVAASVTVMM RIDAVVTGAL LALLVLLTLV TALAHSRFLR HRRATRAASG

       190        200        210        220        230        240
EVAGALREMV GAVGAVQAAA AEPQVAAHVA GLNGARAEAA VREELYAVVQ RTVIGNPAPI

       250        260        270        280        290        300
GVGVVLLLVA GRMDEGTFSV GDLALFAFYL QILTEALGSI GMLSVRLQRV SVALGRITNN

       310        320        330        340        350        360
LGCRLRRSLE RASPPIASDA PGGTGEGAAA PDAGPEPAPP LRELAVRGLT ARHPGAGHGI

       370        380        390        400        410        420
EDVDLVVERH TVTVVTGRVG SGKSTLVRAV LGLLPHERGT VLWNGEPIAD PASFLVAPRC

       430        440        450        460        470        480
GYTPQVPCLF SGTVRENVLL GRDGAAFDEA VRLAVAEPDL AAMQDGPDTV VGPRGLRLSG

       490        500        510        520        530        540
GQIQRVAIAR MLVGDPELVV LDDVSSALDP ETEHLLWERL LDGTRTVLAV SHRPALLRAA

       550        560        570        580        590
DRVVVLEGGR VEASGTFEEV MAVSAEMGRI WTGAGPGGGD AGPAPQSPPA G
```

50%



Members Customize

	Cluster member(s)	Entry name	Status	Protein names	Organisms	Related Clusters	Length
□	D5SLG9	D5SLG9_STRC2	★	Moenomycin biosynthesis protein MoeD5	Streptomyces clavuligerus (strain ATCC 27064 / DSM 738 / JCM 4710 / NBRC 13307 / NCIMB 12785 / NRRL 3585 / VKM Ac-602)	UniRef100_D5SLG9 UniRef90_D5SLG9	698
□	UPI00037834E8			hypothetical protein	Streptomyces sp. PsTaAH-124	UniRef100_UPI00037834E8 UniRef90_UPI00037834E8	664
□	D6A7F5	D6A7F5_9ACTO	★	Putative uncharacterized protein	Streptomyces ghanaensis ATCC 14672	UniRef100_D6A7F5 UniRef90_A0A001	591
□	UPI00037AB3DF			ABC transporter	Streptomyces viridosporus	UniRef100_UPI00037AB3DF UniRef90_A0A001	629
□	A0A001	A0A001_9ACTO	★	MoeD5	Streptomyces ghanaensis	UniRef100_A0A001 UniRef90_A0A001	591

Sequence

Representative sequence	Length	Mass (Da)
D5SLG9	698	71,778

Checksum: F2AD55595F93E879

```
        10         20         30         40         50         60
MSAPAGASSG ADGDGGARTA TEADGDGDGD SDGKHRGMDG DGGGKHADGD GSTRADGDEK

        70         80         90        100        110        120
RADGGERHAD DGGKRGANGG GKHADNGAET GADGGGKRAD DGRGTRADGG GGADGRGTHA

       130        140        150        160        170        180
DGGGGVRATV AALGAVLHGR RAAYWGLTAL WVLVRAGTLA LGLVFQRLFD QLGGGSGGDR

       190        200        210        220        230        240
LLWSLIAWVA AVEAARLCLQ FGLMAARLEP ALQYDTTGRM RRALLASALR RPGATSRTAP

       250        260        270        280        290        300
GEALRTVGED VDETGFFAAW SPTNLAHWIF VLASVTIMIR IDPTVTLALL ALLIAVTAAT

       310        320        330        340        350        360
GALHGRFLAH RRATRTASAS VAGALREAIG SVAAVQAAAA ERHVSAHVVR LNEARARAAV

       370        380        390        400        410        420
REELYASLQR TVLGNAAPIG VGLVLLLTAT GSREGSFTVG DLALFTLYLQ LLTEALASIG

       430        440        450        460        470        480
ILSVRFQRVS VALERVGGFF GGRLRHRLDP PAAPAAPARA DAAGALRELT VRGLTARHPG

       490        500        510        520        530        540
GGHGVEDIDL TVVRHSVTVI TGGIGSGKTT LLRAVLGLLP RERGEILWNG EPVADPAAFL

       550        560        570        580        590        600
VAPRCGHTPQ APRLFSGTLR ENILLGADGA AFGPAVDTAV LGPDLATLEE GADTVVGPRG

       610        620        630        640        650        660
LRLSGGQLQR AAIARMLARD PELLVLDDVS SALDPDTERL LWQRLLARGP TVLAVSHRPA

       670        680        690
LLRAAARVVV LKDGRVEAAG TLEEVLSASP EMRRIWTG
```

Hobohm Clustering

Selection of representative protein data sets
UWE HOBOHM, MICHAEL SCHARF, REINHARD SCHNEIDER,
AND CHRIS SANDER

Hobohm Version 1

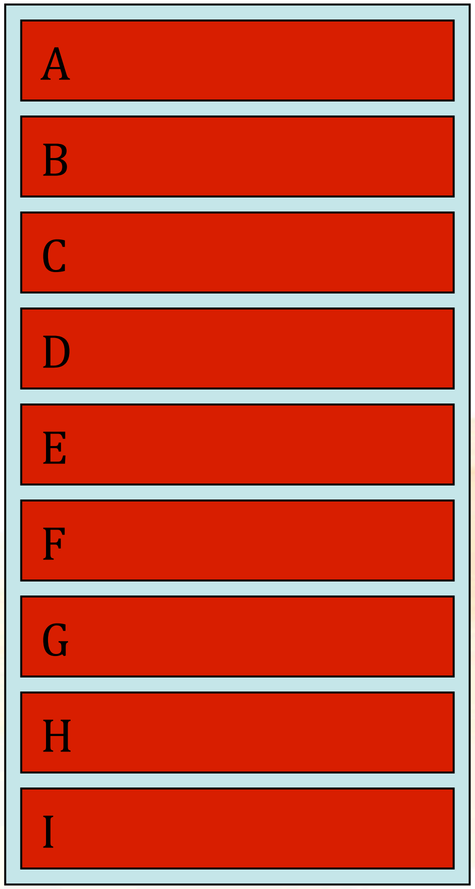
- Takes an sorted list of sequences as input (can be length, resolution of structure, etc)
- From the top of the list
  - Sequences are placed on an accepted list or
  - Discarded depending on whether they are similar
    - Do share more than X% identify to any member on the accepted list or not.
- This procedure is repeated for all sequences in the list
- **After the Hobohm reduction, the pairwise similarity in the accept list has a maximum given by the threshold used to generate it**

- This method is also used for the construction of the BLOSUM matrices normally used by BLAST (need sequence weighting)
  - The most commonly used clustering threshold is 62%

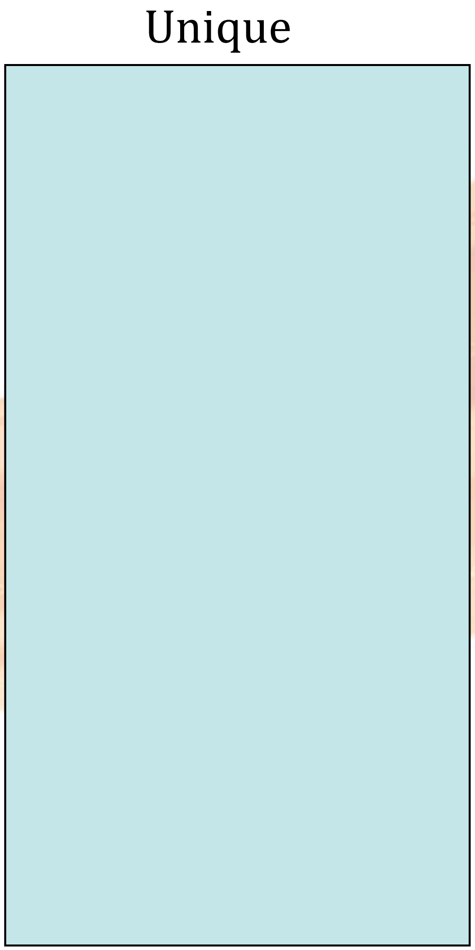
http://search.cpan.org/~brunov/String-Cluster-Hobohm-0.112890/lib/String/Cluster/Hobohm.pm

Hobohm Version 1

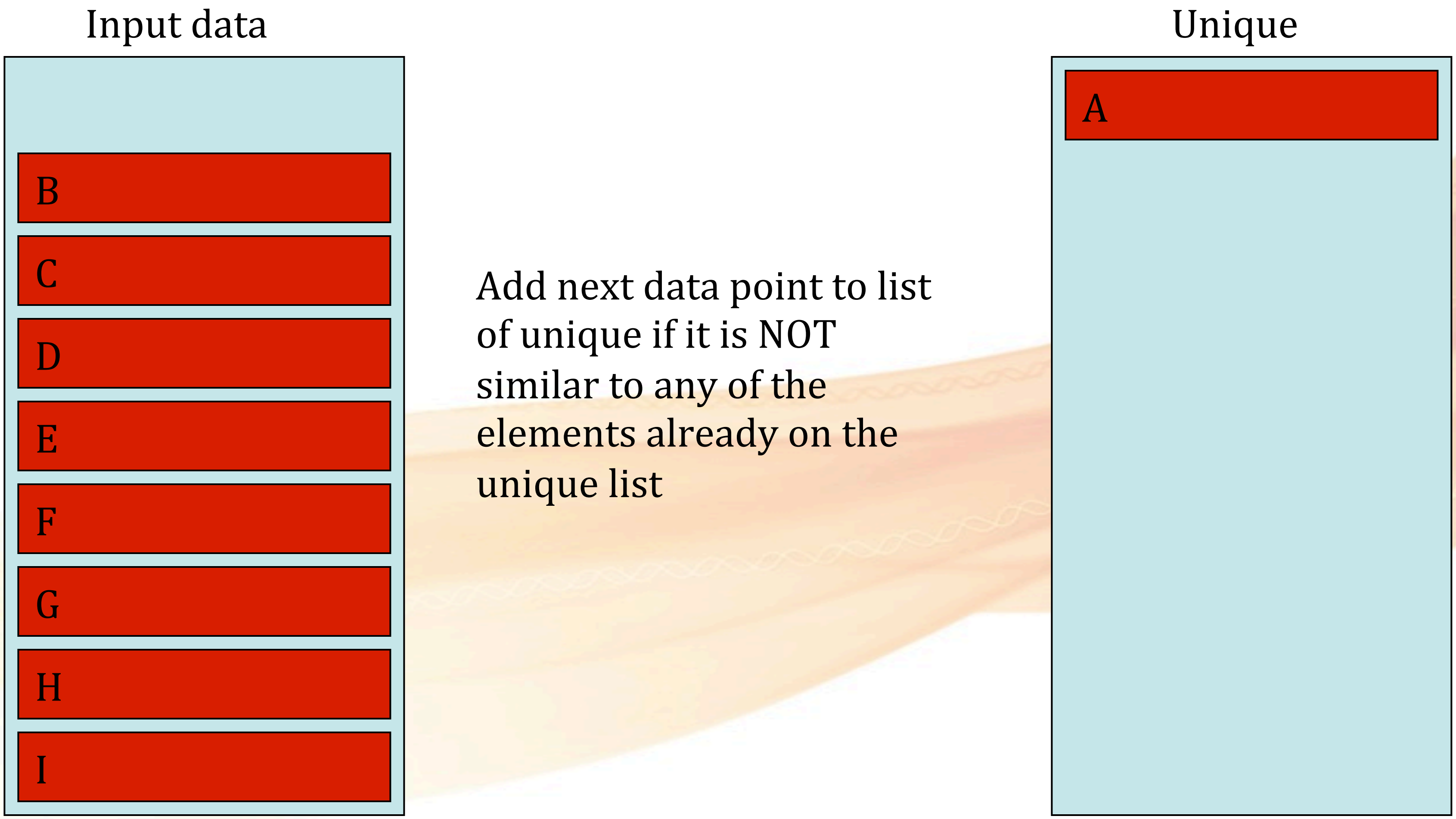
Input data - length in descending order to generate an ordered sequence set S



Add next data point to list of unique if it is NOT similar to any of the elements already on the unique list



Hobohm Version 1



Input data

B

C

D

E

F

G

H

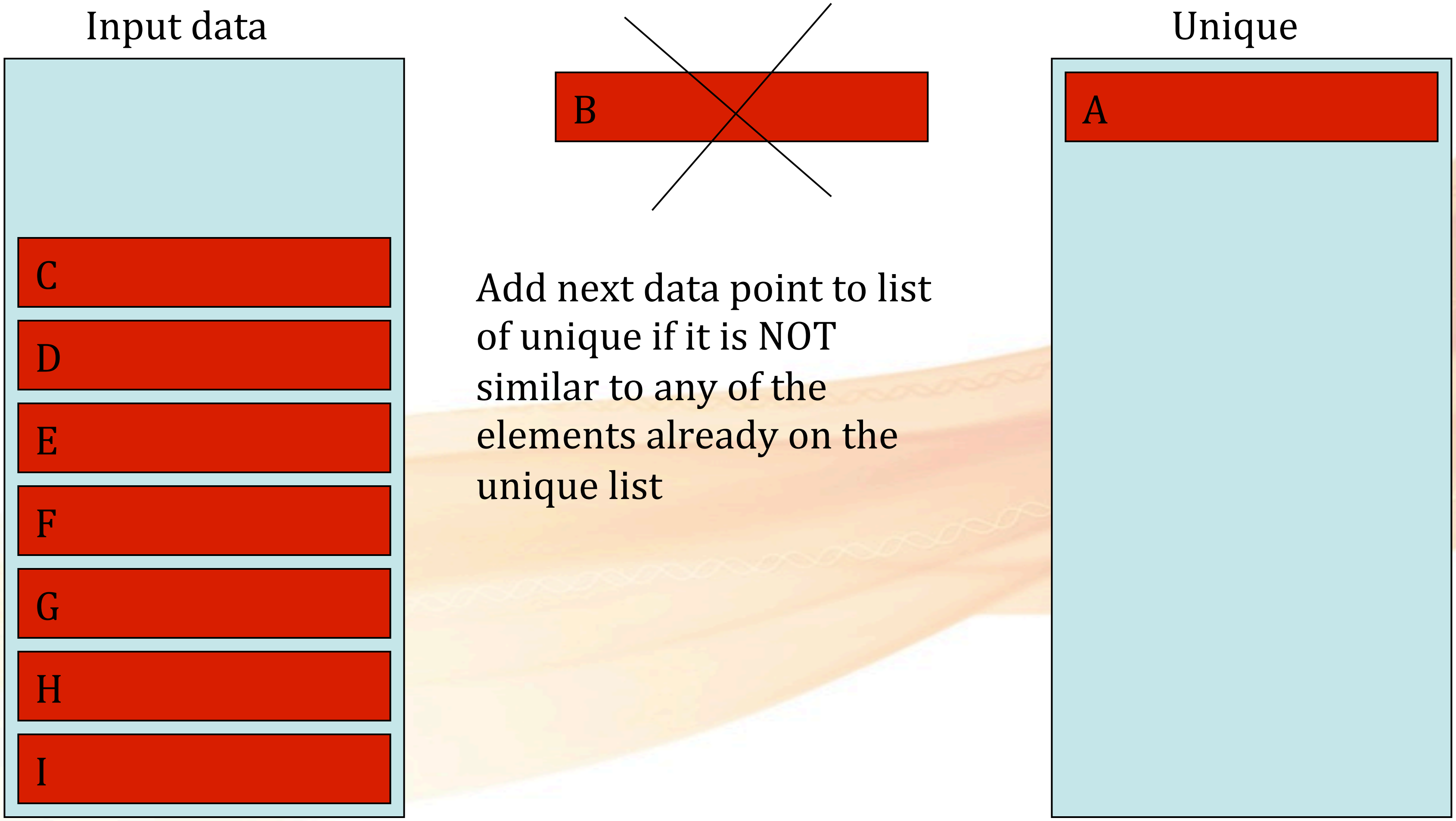
I

Unique

A

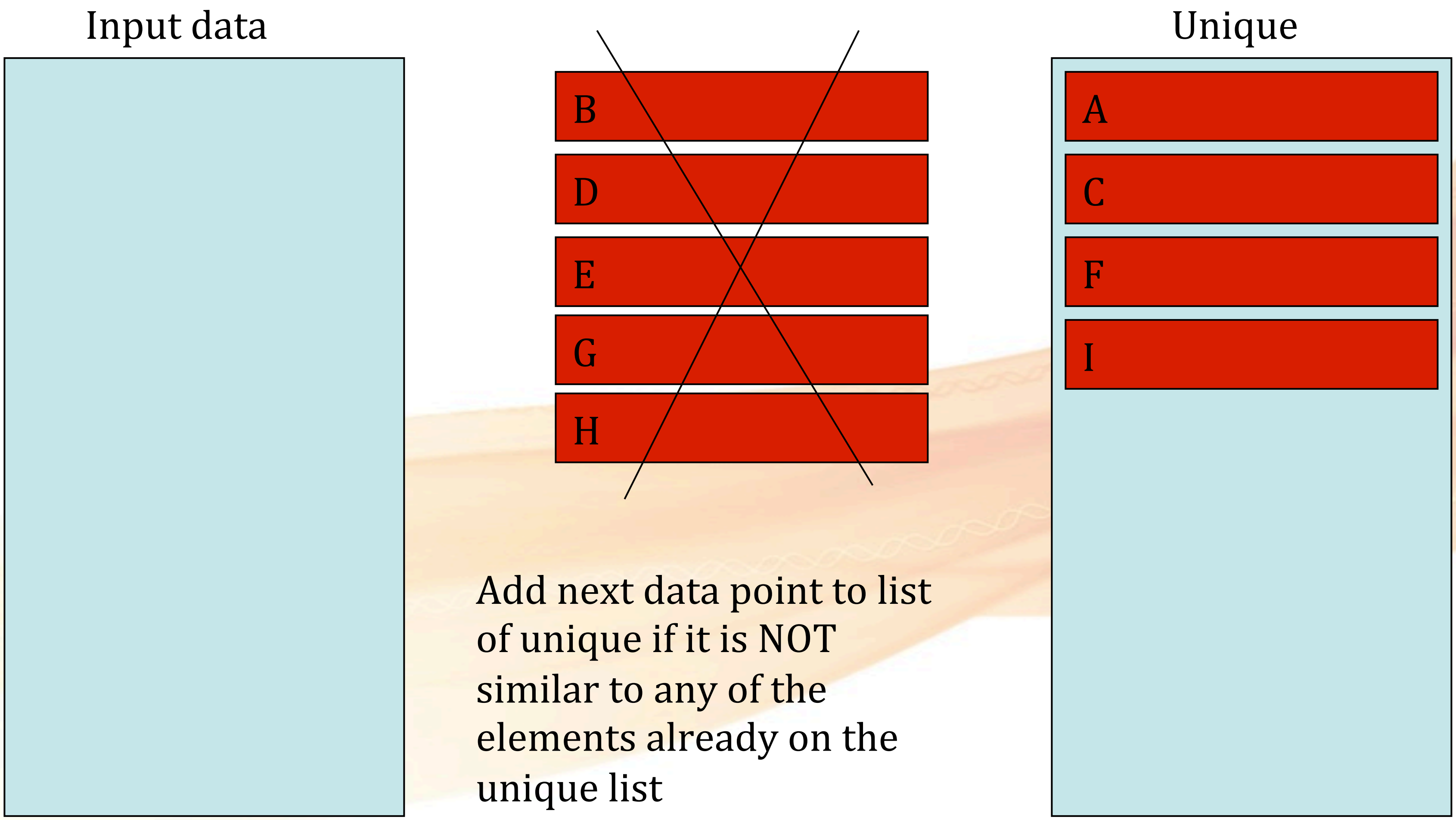
Add next data point to list of unique if it is NOT similar to any of the elements already on the unique list

Hobohm Version 1



Add next data point to list of unique if it is NOT similar to any of the elements already on the unique list

Hobohm Version 1



Add next data point to list of unique if it is NOT similar to any of the elements already on the unique list

Need only to align sequences against the Unique list!

Hobohm-2

- Align all-against-all
  - `(N * (N-1) ) / 2` comparisons
- Make similarity matrix D (N*N)
  - With value 1 if it's similar to j, otherwise 0
  - **Similar here is a threshold you can change**
- While data points have more than one neighbor
  - Remove data point S with most nearest neighbors

Hobohm-2

D:

	A	B	C	D	E	F	G	H	I
A	1	1	1	0	0	0	0	0	0
B	1	1	1	0	0	0	0	1	1
C	1	1	1	0	0	0	0	0	0
D	0	0	0	1	1	1	1	1	1
E	0	0	0	1	1	1	1	1	1
F	0	0	0	1	1	1	0	0	1
G	0	0	0	1	1	0	1	1	1
H	0	1	0	1	1	0	1	1	1
I	0	1	0	1	1	1	1	1	1

Make similarity matrix N*N

Hobohm-2

D:

		A	B	C	D	E	F	G	H	I	N
	A	1	1	1	0	0	0	0	0	0	3
	B	1	1	1	0	0	0	0	1	1	5
	C	1	1	1	0	0	0	0	0	0	3
	D	0	0	0	1	1	1	1	1	1	6
	E	0	0	0	1	1	1	1	1	1	6
	F	0	0	0	1	1	1	0	0	1	4
	G	0	0	0	1	1	0	1	1	1	5
	H	0	1	0	1	1	0	1	1	1	6
S	I	0	1	0	1	1	1	1	1	1	7

Find point S with the largest number of similarities

Hobohm-2

D:

	A	B	C	D	E	F	G	H	I	N
A	1	1	1	0	0	0	0	0	0	3
B	1	1	1	0	0	0	0	1	1	5
C	1	1	1	0	0	0	0	0	0	3
D	0	0	0	1	1	1	1	1	1	6
E	0	0	0	1	1	1	1	1	1	6
F	0	0	0	1	1	1	0	0	1	4
G	0	0	0	1	1	0	1	1	1	5
H	0	1	0	1	1	0	1	1	1	6
I	0	1	0	1	1	1	1	1	1	7

D:

	A	B	C	D	E	F	G	H	N
A	1	1	1	0	0	0	0	0	3
B	1	1	1	0	0	0	0	1	4
C	1	1	1	0	0	0	0	0	3
D	0	0	0	1	1	1	1	1	5
E	0	0	0	1	1	1	1	1	5
F	0	0	0	1	1	1	0	0	3
G	0	0	0	1	1	0	1	1	4
H	0	1	0	1	1	0	1	1	5

Remove point S with the largest number of similarities, and update N counts

Hobohm-2 (repeat this)

D:

	A	B	C	D	E	F	G	H	N
A	1	1	1	0	0	0	0	0	3
B	1	1	1	0	0	0	0	1	4
C	1	1	1	0	0	0	0	0	3
D	0	0	0	1	1	1	1	1	5
E	0	0	0	1	1	1	1	1	5
F	0	0	0	1	1	1	0	0	3
G	0	0	0	1	1	0	1	1	4
H	0	1	0	1	1	0	1	1	5

D:

	A	B	C	E	F	G	H	N
A	1	1	1	0	0	0	0	3
B	1	1	1	0	0	0	1	4
C	1	1	1	0	0	0	0	3
E	0	0	0	1	1	1	1	4
F	0	0	0	1	1	0	0	2
G	0	0	0	1	0	1	1	3
H	0	1	0	1	0	1	1	4

Remove point S with the largest number of similarities

Hobohm-2 (until N=1 for all)

D:

	A	B	C	D	E	F	G	H	I	N
A	1	1	1	0	0	0	0	0	0	3
B	1	1	1	0	0	0	0	1	1	5
C	1	1	1	0	0	0	0	0	0	3
D	0	0	0	1	1	1	1	1	1	6
E	0	0	0	1	1	1	1	1	1	6
F	0	0	0	1	1	1	0	0	1	4
G	0	0	0	1	1	0	1	1	1	5
H	0	1	0	1	1	0	1	1	1	6
I	0	1	0	1	1	1	1	1	1	7

=>

D:

	C	F	H	N
C	1	0	0	1
F	0	1	0	1
H	0	0	1	1

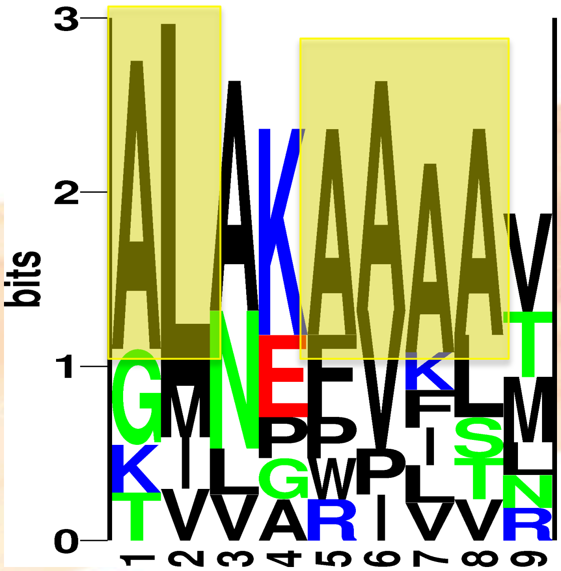
Unique list is C, F, H

Two Hobohm Algorithms?

- Hobohm-2 (greedy)
  - Unbiased
  - Slow (O^2)
  - Focuses on lonely sequences
  - Example from exercise
    - 1000 Sequences alignment
    - Hobohm-2: 2 hours
- Hobohm-1
  - Biased by the original list
  - Fast (0)
  - Focuses on populated sequence areas
  - Example
    - 1000 Sequences
    - Hobohm-1: 12 seconds
- Hobohm2 in general gives more sequences than Hobohm1

Why Do Would We Need Sequence Weighting?
Raw Sequence Counting

- We could use the raw sequence
- Problems just mentioned are now more apparent
  - Where is this evident?
- The first 5 sequences in the alignment are very similar, and may reflect a sampling bias, rather than an actual amino acids bias in the binding motif
- What could we do?
- We need a way to weight the seqeuences



ALAKAAAAM
ALAKAAAAN
ALAKAAAAR
ALAKAAAAT
ALAKAAAAV
GMNERPILT
GILGFVFTM
TLNAWVKVV
KLNEPVLLL
AVVPFIVSV

Sequence Weighting

- Poor or biased sampling of sequence space
- In such a situation, one would therefore like to down-weight identical or almost identical sequences
- Example P1
  - P_A = **2/6**
  - $P_G = 2/6$
  - $P_T = P_K = 1/6$
  - $P_C = P_D = \ldots P_V = 0$

ALAKAAAAM
ALAKAAAAN
ALAKAAAAR
ALAKAAAAT
ALAKAAAAV
GMNERPILT
GILGFVFTM
TLNAWVKVV
KLNEPVLLL
AVVPFIVSV

Similar sequences
Weight 1/5

Sequence Weighting

- Different methods can be used to **weight sequences**
- One method is to cluster sequences
  - How to define clusters?
    - Hobohm algorithms (version 1 & 2)
      - Slow when data sets are large
      - Computation time increases as the square of the number of sequences (depending on the similarity between the sequences)
- The other method is a Heuristic (Henikoff method)
  - Less accurate, but but is sound for PSSM
  - Fast, as the computation time only increases linearly with the number of sequences

Clustering by the Hobohm 1

- End Results of Hobohm clustering:
- After clustering each peptide **k** in a cluster is assigned a weight:
  - $w_k = 1/N_c$
    - where Nc is the number of sequences in the cluster that contains peptide k
    - When the a.a frequencies are calculated, each a.a in sequence k is weighted by w_k
- In this example the first 5 peptides will form one cluster, and each of these sequences thus contribute with a weight of (1/5) to the probability matrix
- The frequency of A at position p1 will then be p_{1A} 2/6 = 0.33 as opposed to 6/10 = 0.6 found when using the raw sequence counts
  - **This is how weighting works**

Similar sequences Weight 1/5

Peptide	Weight
ALAKAAAAM	0.20
ALAKAAAAN	0.20
ALAKAAAAR	0.20
ALAKAAAAT	0.20
ALAKAAAAV	0.20
GMNERPILT	1.00
GILGFVFTM	1.00
TLNAWVKVV	1.00
KLNEPVLLL	1.00
AVVPFIVSV	1.00

How is Clustering Used in BLOSUM

- To reduce multiple contributions to amino acid pair frequencies from the most closely related members of a family, sequences are clustered within blocks and each cluster is weighted as a single sequence in counting pairs
- This is done by specifying a clustering percentage in which sequence segments that are identical for at least that percentage of amino acids are grouped together
- When a.a. pair frequencies are calculated, each a.a. in sequence k is weighted by w_k
- For example, a BLOSUM62 matrix is calculated from protein blocks such that if two sequences are more than 62% identical, then the contribution of these sequences is weighted to sum to one
- In this way the contributions of multiple entries of closely related sequences is reduced

BLOSUM Paper Example

- Column consisting of 9A residues and 1S residue (9A-1S column)
  - 36 possible AA pairs, 9 AS or SA pairs and no SS pairs
- After clustering , 8 of the 9 sequences with A in the 9A-1S column are clustered
- Then contribution of this column to the frequency table is equivalent to that of a 2A-1S column, which contributes 2AS pairs

Henikoff and Henikoff

The Heuristic Way of Determining Weights

- **Henikoff and Henikoff**
- A method to represent the diversity at a position is to:
  - **Award each different residue an equal share of the weight**
  - **Then to divide up that weight equally among the sequences sharing the same residue**
- So if in a position of a multiple alignment:
  - **r** different residues are represented
    - A residue represented in only one sequence contributes a score of 1/**r** to that sequence
    - Whereas a residue represented in **s** sequences contributes a score of 1/**rs** to each of the **s** sequences
- For each sequence, the contributions from each position are summed to give a sequence weight

Sequence Weighting

- Heuristics - weight on sequence **k** at position **p**

$$w_{kp} = \frac{1}{r \cdot s}$$

  - Where **r** is the number of different amino acids in the column **p**, and **s** is the number occurrence of a.a. **a** in that column
- Weight of sequence **k** is the sum of the weights over all positions

$$w_k = \sum_p w_{kp} = \sum_p \frac{1}{r_p \cdot s_p}$$

Example

$$w_{kp} = \frac{1}{r \cdot s}$$

r is the number of different amino acids in the column **p**, and **s** is the number occurrence of a.a. **a** in that column

Peptide	Weight
ALAKAAAAM	0.41
ALAKAAAAN	0.50
ALAKAAAAR	0.50
ALAKAAAAT	0.41
ALAKAAAAV	0.39
GMNERPILT	1.36
GILGFVFTM	1.46
TLNAWVKVV	1.27
KLNEPVLLL	1.19
AVVPFIVSV	1.51

End Results

Example (Weight on Each Sequence)

$$w_{kp} = \frac{1}{r \cdot s}$$

r is the number of different amino acids in the column **p**, and **s** is the number occurrence of a.a. **a** in that column

W_{11}= 1/(4*6) = 0.042
W_{12}= 1/(4*7) = 0.036
W_{13}= 1/(4*5) = 0.050
W_{14}= 1/(5*5) = 0.040
W_{15}= 1/(5*5) = 0.040
W_{16}= 1/(4*5) = 0.050
W_{17}= 1/(6*5) = 0.033
W_{18}= 1/(5*5) = 0.040
W_{19}= 1/(6*2) = 0.083
Sum = 0.414

Peptide
ALAKAAAAM
ALAKAAAAN
ALAKAAAAR
ALAKAAAAT
ALAKAAAAV
GMNERPILT
GILGFVFTM
TLNAWVKVV
KLNEPVLLL
AVVPFIVSV

Example (Weight on Each Sequence)

$$w_{kp} = \frac{1}{r \cdot s}$$

r is the number of different amino acids in the column **p**, and **s** is the number occurrence of a.a. **a** in that column

W_{11}= 1/(4*6) = 0.042
W_{12}= 1/(4*7) = 0.036
W_{13}= 1/(4*5) = 0.050
W_{14}= 1/(5*5) = 0.040
W_{15}= 1/(5*5) = 0.040
W_{16}= 1/(4*5) = 0.050
W_{17}= 1/(6*5) = 0.033
W_{18}= 1/(5*5) = 0.040
W_{19}= 1/(6*2) = 0.083
Sum = 0.414

Peptide	Weight
ALAKAAAAM	0.41
ALAKAAAAN	0.50
ALAKAAAAR	0.50
ALAKAAAAT	0.41
ALAKAAAAV	0.39
GMNERPILT	1.36
GILGFVFTM	1.46
TLNAWVKVV	1.27
KLNEPVLLL	1.19
AVVPFIVSV	1.51

Example (Weight on Each Column)

$$w_{kp} = \frac{1}{r \cdot s}$$

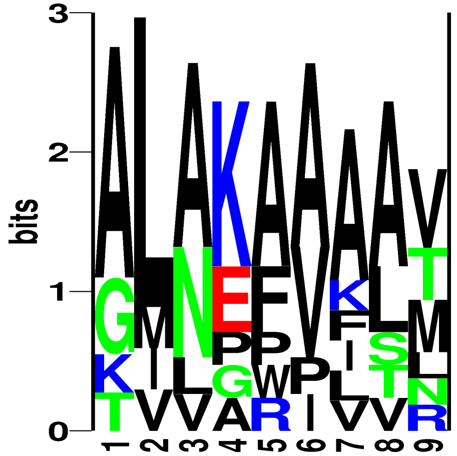
r is the number of different amino acids in the column **p**, and **s** is the number occurrence of a.a. **a** in that column

$W_{11} = 1/(4*6) = 0.042$
$W_{21} = 1/(4*6) = 0.042$
$W_{31} = 1/(4*6) = 0.042$
$W_{41} = 1/(4*6) = 0.042$
$W_{51} = 1/(4*6) = 0.042$
$W_{61} = 1/(4*2) = 0.125$
$W_{71} = 1/(4*2) = 0.125$
$W_{81} = 1/(4*1) = 0.250$
$W_{91} = 1/(4*1) = 0.250$
$W_{101} = 1/(4*6) = 0.042$
Sum = 1.000

Peptide	Weight
ALAKAAAAM	0.41
ALAKAAAAN	0.50
ALAKAAAAR	0.50
ALAKAAAAT	0.41
ALAKAAAAV	0.39
GMNERPILT	1.36
GILGFVFTM	1.46
TLNAWVKVV	1.27
KLNEPVLLL	1.19
AVVPFIVSV	1.51
Sum =	9.00

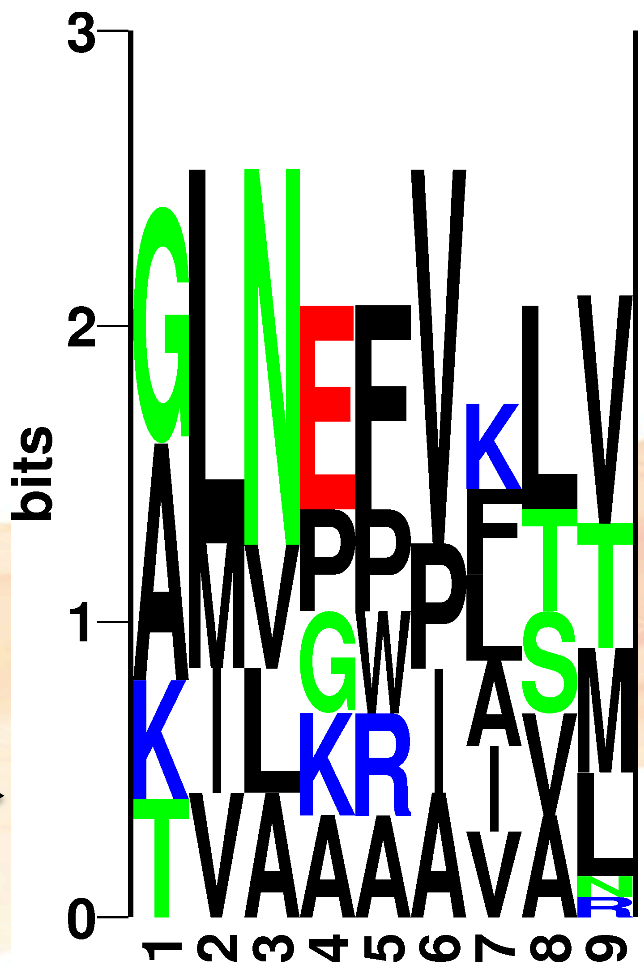
Sum of weights for all sequences is hence L (=9)

Sequence Weighting



Raw Sequence Counting

From the figure it is apparent that the strong alanine bias in the motif has been removed



With Sequence Weighting

ALAKAAAAM
ALAKAAAAN
ALAKAAAAR
ALAKAAAAT
ALAKAAAAV
GMNERPILT
GILGFVFTM
TLNAWVKVV
KLNEPVLLL
AVVPFIVSV

Better, but still some work to do