CHAPTER

6

Sequence Alignment and Similarity Searching in Genomic Databases: BLAST and FASTA*

OUTLINE

6.1 EVOLUTIONARY BASIS OF SEQUENCE ALIGNMENT

As discussed in Chapter 2, evolution is defined as "descent with modification" from a common ancestor. At the molecular level, the modification means changes in DNA and protein sequence, and corresponding changes in protein function. As mutations accumulate in sequences derived from an ancestral sequence, the derived sequences diverge from one another over time, but sections of the sequences may still retain enough similarity to allow identification of a common ancestry.

*The opinions expressed in this chapter are the author's own and they do not necessarily reflect the opinions of the FDA, the DHHS, or the Federal Government.

Evolutionary change in a sequence does not always have to be large; slight changes in certain crucial sections of a sequence can have profound functional consequences.

Expectedly, sequence comparison through sequence alignment is central to most bioinformatic analysis. It is the first step towards understanding the evolutionary relationship and the pattern of divergence between two sequences. The relationship between two sequences also helps predict the potential function of an unknown sequence, thereby indicating protein family relationship.

6.2 THREE TERMS—SEQUENCE IDENTITY, SEQUENCE SIMILARITY, AND SEQUENCE HOMOLOGY—AND THEIR PROPER USAGE

Sequence identity means the same residues being present at corresponding positions in two sequences being compared. For proteins, it means the same amino acids; for nucleic acids, it means the same bases.

Sequence similarity means similar residues being present at corresponding positions in the two sequences being compared. For nucleic acids, sequence similarity and sequence identity are the same. However, for proteins, sequence similarity involves amino acids with similar physicochemical and functional properties. For example, substitution of lysine and arginine by one another will be regarded as similar substitution because both are positively charged hydrophilic amino acids. Likewise, substitution of aspartic acid and glutamic acid by one another will be regarded as similar substitution because both are negatively charged hydrophilic amino acids. Substitution of asparagine by aspartic acid and substitution of glutamine by glutamic acid, or vice versa, are also regarded as similar substitutions. Substitution of isoleucine, leucine, and valine by one another will be regarded as similar substitutions because they have similar aliphatic hydrophobic side chains. Substitution of serine and threonine by one another is also regarded as similar substitution. Similar substitutions are also referred to as **conservative substitutions**[a]. A conservative amino acid substitution is not expected to disrupt the structural/functional attributes of the protein.

Sequence homology is an evolutionary term that has been misused the most in the literature to denote sequence similarity or identity. Sequences are called homologous if they have a common evolutionary origin—that is, if they are derived from a common ancestral sequence. So, sequences are either homologous or not homologous and there is no quantitation of homology. However, even now, expressions like "high homology," "significant homology," and even specifying a "% homology" are very widely used. Such usage has no reference to the evolutionary underpinning of the term **homology**. The root of the term homology goes back to the early evolutionary literature, where organs having similar structure and anatomical origin but performing different functions (hence morphologically different) were called homologous organs. Examples of homologous organs are bats' wings, whales' flippers, and human hands; these are all mammalian forelimbs that are morphologically different because they are adapted to perform different functions. Conversely, organs having different structure and anatomical origin but performing the same function (hence morphologically similar) were called analogous organs. Such a character state (analogous organs) shared by a set of species but not present in their common ancestor is also called **homoplasy**. Examples of analogous organs/homoplasy are bats' wings and butterflies' wings, and dolphins' flippers and sharks' fins. Homoplasy is the result of **convergent evolution** in which unrelated species develop similar morphological structures because of adaptation to the same or a similar environment.

In the case of nucleic acid or protein sequence, a high degree of identity/similarity usually suggests homology as well. However, conclusions about homology are largely conjecture because we cannot go back in time and test the sequence in the ancestor and the descendants. Therefore, it is the quantitative identity/similarity between the two sequences that is used to conclude whether the two sequences are homologous or not. For example, metallothionein-1 proteins in rat and mouse (61 amino acids in both) are 95% identical and 98% similar[b]. Rat and mouse diverged about 33 million years ago.[1] Therefore, based on the substitution of three amino acids in 33 million years, the substitution rate per site per year can be calculated, and it can be concluded with a great deal of certainty that rat and mouse metallothionein-1 were derived from a common ancestor, and have not changed much, probably because of functional constraints; hence, they are **homologous**. Homologous genes in different species performing the same function are called **orthologs**. So, the metallothionein-1 genes in rat and mouse are also orthologs. The problem in drawing conclusions on

[a]*Similar substitution* and *conservative substitution* refer to amino acid substitution in protein. *Synonymous substitution* and *nonsynonymous substitution* refer to nucleotide substitution in DNA. *Synonymous substitution* leads to no changes in amino acids in the encoded protein, while *nonsynonymous substitution* leads to changes in amino acids in the encoded protein.

[b]95% identity (58 identical amino acids; hence $(58/61) \times 100 = 95\%$ identity); 98% similarity (58 identical amino acids + 2 similar substitutions; hence $(60/61) \times 100 = 98\%$ similarity).

homology arises when the similarity between two sequences is low. Conclusions on homology, in this case, are drawn on a case-by-case basis. Two proteins can be considered homologous despite low similarity if one or more of the following conditions are met: (1) the similarity extends over a long stretch of sequence and is statistically significant; (2) despite low sequence similarity, the same pattern of identical and similar amino acid residues is seen in multiple sequences; or (3) the pattern of sequence similarity reflects the similarity between experimentally determined structures of the respective proteins, or at least corresponds to the known key elements of one such structure.[2]

6.3 SEQUENCE IDENTITY AND SEQUENCE SIMILARITY

Sequence identity and sequence similarity can be calculated based on the proportion of identical and similar amino acids, respectively:

$$\%\,\text{Identity(PID)} = (\#\,\text{of identical amino acids}/\#\,\text{of total amino acids}) \times 100; \tag{6.1}$$

$$\%\,\text{Similarity} = \{(\#\,\text{of identical amino acids} + \#\,\text{of similar substitutions})/\#\,\text{of total amino acids}\} \times 100 \tag{6.2}$$

In the above formulae, the denominator (# of total amino acids) can vary. For example, the denominator could be (1) the length of the shortest sequence, (2) the length of the longest sequence, (3) the mean length of the sequences, (4) the length of the aligned region (aligned positions excluding overhangs), etc. Therefore, PID is a rough measure and can be influenced by how it is calculated. However, because of the simplicity of calculation, PID is widely used.[3]

The pairwise alignment in Figure 6.1 (National Center for Biotechnology Information (NCBI) BLAST pairwise alignment; http://blast.ncbi.nlm.nih.gov/→check the "Align two or more sequences" link) and that in Figure 6.2 (EMBOSS Needle of the European Molecular Biology Laboratory's European Bioinformatics Institute (EMBL-EBI); http://www.ebi.ac.uk/Tools/psa/) show that there are 560 identical amino acids and 53 similar substitutions (making 560 + 53 = 613 similar amino acids) between the rlst-1a and mlst-1 proteins[c]. This makes the identity 81% and the similarity 88.7%. Note that the NCBI designates % similarity as % positive.

6.4 GLOBAL VERSUS LOCAL ALIGNMENT

A **global sequence-alignment** method aligns and compares two sequences along their entire length, and comes up with the best alignment that displays the maximum number of nucleotides or amino acids aligned. The algorithm that drives global alignment is the **Needleman–Wunsch algorithm**. A global alignment algorithm starts at the beginning of two sequences and adds gaps to each until the end of one is reached. *Global alignment works the best when the sequences are similar in character and length*. Because global alignment displays the best alignment between two sequences using the entire sequence, it may miss a small region of biological importance. This is a trade-off in global alignment.

Two of the available web servers for pairwise global alignment are **EMBL-EBI EMBOSS** (http://www.ebi.ac.uk/Tools/psa/), and **NCBI specialized BLAST** (look for the Global Sequence Alignment Tool link on the NCBI BLAST home page under Specialized BLAST; the URL is too long to include here). For EMBL-EBI EMBOSS, the page that appears by clicking the link provides separate options for protein and nucleotide global alignment. **EMBOSS Stretcher** uses a modification of the Needleman–Wunsch algorithm that allows larger sequences to be globally aligned; it also provides separate options for proteins and nucleic acids.

In contrast to global alignment, **local sequence alignment** is intended to find the most similar regions in two sequences being aligned. The algorithm that drives local alignment is the **Smith–Waterman algorithm**. A local alignment algorithm finds the region of highest similarity between two sequences and builds the alignment outward from this region. If there are multiple regions of very high similarity, the same principle applies. *Obviously, local alignment is useful for sequences that are not similar in character and length, yet are suspected to contain small regions of similarity, such as biologically important motifs.*

The global and local alignments involving two protein sequences that are significantly similar produce identical results. For example, running a global alignment using the Needleman–Wunsch algorithm or a local alignment using the Smith–Waterman algorithm (discussed below) for the rlst-1a and mlst-1 proteins produces identical results. Pairwise global alignment using both RNA (complementary DNA, or cDNA) and protein sequences can identify alternatively spliced variants. Figure 6.3 (EMBOSS Needle of the EMBL-EBI) shows that rlst-1c protein, which is an alternatively spliced form, lacks a segment of 33 amino acids that is present

[c]The original submission accession number of rlst-1a is AF208545 and that of mlst-1 is AB031959.

Query: rlst-1a; 687 amino acids (Accession #: AAF87098.1)
Sbjct: mlst-1; 689 amino acids (Accession #: BAB03272.1)

Score	Expect	Method	Identities	Positives	Gaps	Frame
1166 bits(3016)	0.0	Compositional matrix adjust	560/691(81%)	613/691(88%)	6/691(0%)	

```
Query  1    MDHTQQSRKAAEAQPSRSKQTRFCDGFKLFLAALSFSYICKALGGVVMKSSITQIERRFD  60
            MD TQ   KAA  QP RS++TR CDGF++FLAALSFSYICKALGGV+MKSSITQIERRFD
Sbjct  1    MDQTQHPSKAA--QPLRSEKTRHCDGFRIFLAALSFSYICKALGGVIMKSSITQIERRFD  58

Query  61   IPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGIGCFIMGIGSILTALPHFFMGY  120
            IPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIG GCFIMGIGSILTALPHFFMGY
Sbjct  59   IPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGTGCFIMGIGSILTALPHFFMGY  118

Query  121  YKYAKENDIGSLGNSTLTCFINQMTSPTGPSPEIVEKGCEKGLKSHMWIYVLMGNMLRGI  180
            Y+YA ENDI SL NSTLTC +NQ TS TG SPEI+EKGCEKG  S+ WIYVLMGNMLRGI
Sbjct  119  YRYATENDISSLHNSTLTCLVNQTTSLTGTSPEIMEKGCEKGSNSYTWIYVLMGNMLRGI  178

Query  181  GETPIVPLGISYLDDFAKEGHTSMHLGTLHTIAMIGPILGFIMSSVFAKIYVDVGYVDLN  240
            GETPIVPLG+SY+DDFAKEG++SM+LGTLHTIAMIGPILGFIMSSVFAK+YVDVGYVDL
Sbjct  179  GETPIVPLGVSYIDDFAKEGNSSMYLGTLHTIAMIGPILGFIMSSVFAKLYVDVGYVDLR  238

Query  241  SVRITPNDARWVGAWWLSFIVNGLLCITSSIPFFFLPKIPKRSQEERKNSVSLHAPKTDE  300
            SVRITP DARWVGAWWL FIVNGLLCI  SIPFFFLPKIPKRSQ+ERKNS SLH  KTDE
Sbjct  239  SVRITPQDARWVGAWWLGFIVNGLLCIICSIPFFFLPKIPKRSQKERKNSASLHVLKTDE  298

Query  301  EKKHMTNLTKQEEQDPSNMTGFLRSLRSILTNEIYVIFLILTLLQVSGFIGSFTYLFKFI  360
            +K  +TN T QE+Q P+N+TGFL SLRSILTNE YVIFLILTLLQ+S FIGSFTYLFKFI
Sbjct  299  DKNPVTNPTTQEKQAPANLTGFLWSLRSILTNEQYVIFLILTLLQISSFIGSFTYLFKFI  358

Query  361  EQQFGRTASQANFLLGIITIPTMATAMFLGGYIVKKFKLTSVGIAKFVFFTSSVAYAFQF  420
            EQQFG+TASQANFLLG+ITIPTMA+ MFLGGY++K+ KLT +GI KFVFFT+++AY F
Sbjct  359  EQQFGQTASQANFLLGVITIPTMASGMFLGGYLIKRLKLTLLGITKFVFFTTTMAYVFYL  418

Query  421  LYFPLLCENKPFAGLTLTYDGMNPVDSHIDVPLSYCNSDCSCDKNQWEPICGENGVTYIS  480
             YF L+CENK FAGLTLTYDGMNPVDSHIDVPLSYCNSDC CDKNQWEP+CGENGVTYIS
Sbjct  419  SYFLLICENKAFAGLTLTYDGMNPVDSHIDVPLSYCNSDCICDKNQWEPVCGENGVTYIS  478

Query  481  PCLAGCKSFRGDKKPNNTEFYDCSCISNS----GNNSAHLGECPRYKCKTNYYFYIILQV  536
            PCLAGCKSFRGDKK  N EFYDCSC+S S    GN+SA LGECPR KCKT YYFYI  QV
Sbjct  479  PCLAGCKSFRGDKKLMNIEFYDCSCVSGSGFQKGNHSARLGECPRDKCKTKYYFYITFQV  538

Query  537  TVSFFTAMGSPSLILILMKSVQPELKSLAMGFHSLIIRALGGILAPIYYGAFIDRTCIKW  596
             +SFFTA+GS SL+LIL++SVQPELKSL MGFHSL++R LGGILAP+YYGA IDRTC+KW
Sbjct  539  IISFFTALGSTSLMLILIRSVQPELKSLGMGFHSLVVRTLGGILAPVYYGALIDRTCMKW  598

Query  597  SVTSCGKRGACRLYNSRLFGFSYLGLNLALKTPPLFLYVVLIYFTKRKYKRNDNKTLENG  656
            SVTSCG RGACRLYNSRLFG  Y+GL++ALKTP L LYV LIY  KRK KRNDNK LENG
Sbjct  599  SVTSCGARGACRLYNSRLFGMIYVGLSIALKTPILLLYVALIYVMKRKMKRNDNKILENG  658

Query  657  RQFTDEGNPDSVNKNGYYCVPYDEQSNETPL  687
            R+FTDEGNP+ VN NGY CVP DE+++ETPL
Sbjct  659  RKFTDEGNPEPVNNNGYSCVPSDEKNSETPL  689
```

FIGURE 6.1 **Pairwise alignment of rlst-1a and mlst-1 proteins using NCBI BLAST.** NCBI BLAST pairwise alignment shows that these two proteins share 81% identity but 88.7% similarity. The similar amino acids are highlighted in gray; many of these are hydrophobic amino acids, charged polar amino acids, and neutral polar amino acids. In the NCBI BLAST pairwise alignment format, the identical amino acids and similar substitutions between the query and the subject sequences are in the middle; and similar substitutions are indicated by a + sign.

```
# Aligned_sequences: 2
# 1: rlst-1a (Accession #: AAF87098.1)
# 2: mlst-1  (Accession #: BAB03272.1)
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 691
# Identity:     560/691 (81.0%)
# Similarity:   613/691 (88.7%)
# Gaps:           6/691 ( 0.9%)
# Score: 2986.0

rlst-1a           1 MDHTQQSRKAAEAQPSRSKQTRFCDGFKLFLAALSFSYICKALGGVVMKS     50
                    ||.||...||  |||.||::||.||||::||||||||||||||||:|||
mlst-1            1 MDQTQHPSKA--AQPLRSEKTRHCDGFRIFLAALSFSYICKALGGVIMKS     48

rlst-1a          51 SITQIERRFDIPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGIG    100
                    ||||||||||||||||||||||||||||||||||||||||||||||||.|
mlst-1           49 SITQIERRFDIPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGTG     98

rlst-1a         101 CFIMGIGSILTALPHFFMGYYKYAKENDIGSLGNSTLTCFINQMTSPTGP    150
                    ||||||||||||||||||||:||.||||.||.||||||.:||.||.||.
mlst-1           99 CFIMGIGSILTALPHFFMGYYRYATENDISSLHNSTLTCLVNQTTSLTGT    148

rlst-1a         151 SPEIVEKGCEKGLKSHMWIYVLMGNMLRGIGETPIVPLGISYLDDFAKEG    200
                    ||||:|||||||..|:.||||||||||||||||||:||:|||||||
mlst-1          149 SPEIMEKGCEKGSNSYTWIYVLMGNMLRGIGETPIVPLGVSYIDDFAKEG    198

rlst-1a         201 HTSMHLGTLHTIAMIGPILGFIMSSVFAKIYVDVGYVDLNSVRITPNDAR    250
                    ::||:|||||||||||||||||||||:|||||||||.||||||.|||
mlst-1          199 NSSMYLGTLHTIAMIGPILGFIMSSVFAKLYVDVGYVDLRSVRITPQDAR    248

rlst-1a         251 WVGAWWLSFIVNGLLCITSSIPFFFLPKIPKRSQEERKNSVSLHAPKTDE    300
                    |||||||.|||||||..|||||||||||||||:|||||.|||..||||
mlst-1          249 WVGAWWLGFIVNGLLCIICSIPFFFLPKIPKRSQKERKNSASLHVLKTDE    298

rlst-1a         301 EKKHMTNLTKQEEQDPSNMTGFLRSLRSILTNEIYVIFLILTLLQVSGFI    350
                    :|..:||.|.||:|.|:|:||||.|||||||.|||||||||||:|.||
mlst-1          299 DKNPVTNPTTQEKQAPANLTGFLWSLRSILTNEQYVIFLILTLLQISSFI    348

rlst-1a         351 GSFTYLFKFIEQQFGRTASQANFLLGIITIPTMATAMFLGGYIVKKFKLT    400
                    |||||||||||||:||||||||||:||||||||.||||||::|:.|||
mlst-1          349 GSFTYLFKFIEQQFGQTASQANFLLGVITIPTMASGMFLGGYLIKRLKLT    398

rlst-1a         401 SVGIAKFVFFTSSVAYAFQFLYFPLLCENKPFAGLTLTYDGMNPVDSHID    450
                    .:||.||||||:::||.|...||.|:||||.||||||||||||||||||
mlst-1          399 LLGITKFVFFTTTMAYVFYLSYFLLICENKAFAGLTLTYDGMNPVDSHID    448

rlst-1a         451 VPLSYCNSDCSCDKNQWEPICGENGVTYISPCLAGCKSFRGDKKPNNTEF    500
                    ||||||||||.||||||||:||||||||||||||||||||||..|.||
mlst-1          449 VPLSYCNSDCICDKNQWEPVCGENGVTYISPCLAGCKSFRGDKKLMNIEF    498

rlst-1a         501 YDCSCISNS----GNNSAHLGECPRYKCKTNYYFYIILQVTVSFFTAMGS    546
                    |||||:|.|    ||:||.||||||.||||.|||||..||.:||||||||
mlst-1          499 YDCSCVSGSGFQKGNHSARLGECPRDKCKTKYYFYITFQVIISFFTALGS    548

rlst-1a         547 PSLILILMKSVQPELKSLAMGFHSLIIRALGGILAPIYYGAFIDRTCIKW    596
                    .||:|||::|||||||||.|||||::|.|||||||:||||.|||||:||
mlst-1          549 TSLMLILIRSVQPELKSLGMGFHSLVVRTLGGILAPVYYGALIDRTCMKW    598

rlst-1a         597 SVTSCGKRGACRLYNSRLFGFSYLGLNLALKTPPLFLYVVLIYFTKRKYK    646
                    ||||||.||||||||||..|:|||::||||.|.|||.|||..|||.|
mlst-1          599 SVTSCGARGACRLYNSRLFGMIYVGLSIALKTPILLLYVALIYVMKRKMK    648

rlst-1a         647 RNDNKTLENGRQFTDEGNPDSVNKNGYYCVPYDEQSNETPL    687
                    |||||.|||||:|||||||:.||.|||.|||.||:::||||
mlst-1          649 RNDNKILENGRKFTDEGNPEPVNNNGYSCVPSDEKNSETPL    689
```

FIGURE 6.2 **Pairwise global alignment of rlst-1a and mlst-1 proteins using EMBL-EBI EMBOSS.** EMBOSS Needle (Needleman–Wunsch algorithm) shows that these two proteins share 81% identity but 88.7% similarity. The similar amino acids are highlighted in grey.

```
# Aligned_sequences: 2
# 1: rlst-1a (Accession #: AAF87098.1)
# 2: rlst-1c (Accession #: AAF87099.1)
# Matrix: EBLOSUM62
# Gap_penalty: 12
# Extend_penalty: 2
#
# Length: 687
# Identity:     653/687 (95.1%)
# Similarity:   654/687 (95.2%)
# Gaps:          33/687 ( 4.8%)
# Score: 3388

rlst-1a           1 MDHTQQSRKAAEAQPSRSKQTRFCDGFKLFLAALSFSYICKALGGVVMKS     50
                    ||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c           1 MDHTQQSRKAAEAQPSRSKQTRFCDGFKLFLAALSFSYICKALGGVVMKS     50

rlst-1a          51 SITQIERRFDIPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGIG    100
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c          51 SITQIERRFDIPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGIG    100

rlst-1a         101 CFIMGIGSILTALPHFFMGYYKYAKENDIGSLGNSTLTCFINQMTSPTGP    150
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         101 CFIMGIGSILTALPHFFMGYYKYAKENDIGSLGNSTLTCFINQMTSPTGP    150

rlst-1a         151 SPEIVEKGCEKGLKSHMWIYVLMGNMLRGIGETPIVPLGISYLDDFAKEG    200
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         151 SPEIVEKGCEKGLKSHMWIYVLMGNMLRGIGETPIVPLGISYLDDFAKEG    200

rlst-1a         201 HTSMHLGTLHTIAMIGPILGFIMSSVFAKIYVDVGYVDLNSVRITPNDAR    250
                    ||||||                              :||||||||||
rlst-1c         201 HTSMHL----------------------------------DSVRITPNDAR    217

rlst-1a         251 WVGAWWLSFIVNGLLCITSSIPFFFLPKIPKRSQEERKNSVSLHAPKTDE    300
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         218 WVGAWWLSFIVNGLLCITSSIPFFFLPKIPKRSQEERKNSVSLHAPKTDE    267

rlst-1a         301 EKKHMTNLTKQEEQDPSNMTGFLRSLRSILTNEIYVIFLILTLLQVSGFI    350
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         268 EKKHMTNLTKQEEQDPSNMTGFLRSLRSILTNEIYVIFLILTLLQVSGFI    317

rlst-1a         351 GSFTYLFKFIEQQFGRTASQANFLLGIITIPTMATAMFLGGYIVKKFKLT    400
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         318 GSFTYLFKFIEQQFGRTASQANFLLGIITIPTMATAMFLGGYIVKKFKLT    367

rlst-1a         401 SVGIAKFVFFTSSVAYAFQFLYFPLLCENKPFAGLTLTYDGMNPVDSHID    450
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         368 SVGIAKFVFFTSSVAYAFQFLYFPLLCENKPFAGLTLTYDGMNPVDSHID    417

rlst-1a         451 VPLSYCNSDCSCDKNQWEPICGENGVTYISPCLAGCKSFRGDKKPNNTEF    500
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         418 VPLSYCNSDCSCDKNQWEPICGENGVTYISPCLAGCKSFRGDKKPNNTEF    467

rlst-1a         501 YDCSCISNSGNNSAHLGECPRYKCKTNYYFYIILQVTVSFFTAMGSPSLI    550
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         468 YDCSCISNSGNNSAHLGECPRYKCKTNYYFYIILQVTVSFFTAMGSPSLI    517

rlst-1a         551 LILMKSVQPELKSLAMGFHSLIIRALGGILAPIYYGAFIDRTCIKWSVTS    600
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         518 LILMKSVQPELKSLAMGFHSLIIRALGGILAPIYYGAFIDRTCIKWSVTS    567

rlst-1a         601 CGKRGACRLYNSRLFGFSYLGLNLALKTPPLFLYVVLIYFTKRKYKRNDN    650
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||
rlst-1c         568 CGKRGACRLYNSRLFGFSYLGLNLALKTPPLFLYVVLIYFTKRKYKRNDN    617

rlst-1a         651 KTLENGRQFTDEGNPDSVNKNGYYCVPYDEQSNETPL    687
                    |||||||||||||||||||||||||||||||||||||
rlst-1c         618 KTLENGRQFTDEGNPDSVNKNGYYCVPYDEQSNETPL    654
```

FIGURE 6.3 **Pairwise global alignment of rlst-1a and rlst-1c proteins using EMBL-EBI EMBOSS.** EMBOSS Needle (Needleman–Wunsch algorithm) shows that the rlst-1c protein is an alternatively spliced form missing a 33-amino-acid segment that is present in the rlst-1a protein (highlighted).

```
Sequence format is CLUSTAL (CLUSTAL 2.1 Multiple Sequence Alignments)
Sequence 1: rlst-1a      687 aa (Accession #: AAF87098.1)
Sequence 2: rlst-1c      687 aa (Accession #: AAF87099.1)

rlst-1a         MDHTQQSRKAAEAQPSRSKQTRFCDGFKLFLAALSFSYICKALGGVVMKSSITQIERRFD 60
rlst-1c         MDHTQQSRKAAEAQPSRSKQTRFCDGFKLFLAALSFSYICKALGGVVMKSSITQIERRFD 60
                ************************************************************

rlst-1a         IPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGIGCFIMGIGSILTALPHFFMGY 120
rlst-1c         IPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGIGCFIMGIGSILTALPHFFMGY 120
                ************************************************************

rlst-1a         YKYAKENDIGSLGNSTLTCFINQMTSPTGPSPEIVEKGCEKGLKSHMWIYVLMGNMLRGI 180
rlst-1c         YKYAKENDIGSLGNSTLTCFINQMTSPTGPSPEIVEKGCEKGLKSHMWIYVLMGNMLRGI 180
                ************************************************************

rlst-1a         GETPIVPLGISYLDDFAKEGHTSMHLGTLHTIAMIGPILGFIMSSVFAKIYVDVGYVDLN 240
rlst-1c         GETPIVPLGISYLDDFAKEGHTSMHL--------------------------------D 207
                ************************                                :

rlst-1a         SVRITPNDARWVGAWWLSFIVNGLLCITSSIPFFFLPKIPKRSQEERKNSVSLHAPKTDE 300
rlst-1c         SVRITPNDARWVGAWWLSFIVNGLLCITSSIPFFFLPKIPKRSQEERKNSVSLHAPKTDE 267
                ************************************************************

rlst-1a         EKKHMTNLTKQEEQDPSNMTGFLRSLRSILTNEIYVIFLILTLLQVSGFIGSFTYLFKFI 360
rlst-1c         EKKHMTNLTKQEEQDPSNMTGFLRSLRSILTNEIYVIFLILTLLQVSGFIGSFTYLFKFI 327
                ************************************************************

rlst-1a         EQQFGRTASQANFLLGIITIPTMATAMFLGGYIVKKFKLTSVGIAKFVFFTSSVAYAFQF 420
rlst-1c         EQQFGRTASQANFLLGIITIPTMATAMFLGGYIVKKFKLTSVGIAKFVFFTSSVAYAFQF 387
                ************************************************************

rlst-1a         LYFPLLCENKPFAGLTLTYDGMNPVDSHIDVPLSYCNSDCSCDKNQWEPICGENGVTYIS 480
rlst-1c         LYFPLLCENKPFAGLTLTYDGMNPVDSHIDVPLSYCNSDCSCDKNQWEPICGENGVTYIS 447
                ************************************************************

rlst-1a         PCLAGCKSFRGDKKPNNTEFYDCSCISNSGNNSAHLGECPRYKCKTNYYFYIILQVTVSF 540
rlst-1c         PCLAGCKSFRGDKKPNNTEFYDCSCISNSGNNSAHLGECPRYKCKTNYYFYIILQVTVSF 507
                ************************************************************

rlst-1a         FTAMGSPSLILILMKSVQPELKSLAMGFHSLIIRALGGILAPIYYGAFIDRTCIKWSVTS 600
rlst-1c         FTAMGSPSLILILMKSVQPELKSLAMGFHSLIIRALGGILAPIYYGAFIDRTCIKWSVTS 567
                ************************************************************

rlst-1a         CGKRGACRLYNSRLFGFSYLGLNLALKTPPLFLYVVLIYFTKRKYKRNDNKTLENGRQFT 660
rlst-1c         CGKRGACRLYNSRLFGFSYLGLNLALKTPPLFLYVVLIYFTKRKYKRNDNKTLENGRQFT 627
                ************************************************************

rlst-1a         DEGNPDSVNKNGYYCVPYDEQSNETPL 687
rlst-1c         DEGNPDSVNKNGYYCVPYDEQSNETPL 654
                ***************************
```

FIGURE 6.4 **Pairwise alignment of rlst-1a and rlst-1c proteins using DDBJ ClustalW.** Analysis using the multiple alignment program ClustalW (DDBJ). The result is the same as that depicted in Figure 6.3. The missing 33-amino-acid segment in rlst-1c is highlighted. *(DDBJ; http://clustalw.ddbj.nig.ac.jp/)*

in rlst-1a protein, which is the full-length form.[4] The pairwise alignment can also be performed using a multiple alignment program, such as ClustalW (DNA Data Bank of Japan (DDBJ); http://clustalw.ddbj.nig.ac.jp/); the result of the analysis is the same (Figure 6.4). *Note that the alignments in Figures 6.1 through 6.4 have been performed using tools from NCBI, EMBL-EBI, and DDBJ in order to provide visual display of different output formats for marking identical amino acids and similar amino acids.*

6.5 PAIRWISE AND MULTIPLE ALIGNMENT

As the name suggests, pairwise alignment aligns two nucleic acid or two protein sequences to find the best match. Multiple alignment performs the same function using more than two sequences. The purpose of alignment is to identify regions of similarity that may have structural, functional, and evolutionary

TABLE 6.1 Online Pairwise Alignment Tools Using the Smith–Waterman Algorithm

Online Tool	URL
PIR SSEARCH	http://pir.georgetown.edu/pirwww/search/pairwise.shtml[5]
NCBI specialized BLAST	bl2seq resource; look for the Align link on the NCBI BLAST home page under Specialized BLAST
SIM	http://web.expasy.org/sim/
LALIGN*	http://www.ch.embnet.org/software/LALIGN_form.html

*The LALIGN program is William Pearson's, and it implements the algorithm of X. Huang and W. Miller.[6]

consequences. Figures 6.1 through 6.4 are examples of pairwise alignment.

Some widely used online pairwise alignment tools use local alignment strategy (Smith–Waterman algorithm) and are shown in Table 6.1.

The NCBI BLAST pairwise alignment tool, SIM, and LALIGN not only show the overall alignment of the two sequences, but will also display, as separate output, multiple matching subsegments between the two sequences being aligned. For example, Figure 6.5 shows the alignment of the partial sequence of mlst-1 and moatp-2 proteins[d] using LALIGN (http://www.ch.embnet.org/software/LALIGN_form.html), which is also accessible from the EMBL-EBI page (http://www.ebi.ac.uk/Tools/psa/lalign/). A hypothetical sequence "THATISGREATANDFANTASTIC" was added at the beginning of the mlst-1 protein and the end of the moatp-2 protein. The two resulting sequences were then aligned using LALIGN and NCBI BLAST pairwise alignment. Both LALIGN (Figure 6.5) and NCBI BLAST pairwise alignment (Figure 6.6) produced an overall alignment of the two input sequences, and also reported the matching subsegment in these two sequences, which is the added hypothetical sequence. Therefore, these tools are very useful in finding various motifs and conserved sequences between two proteins being compared.

Multiple sequence alignments are useful in identifying conserved sequence segments across the sequences being aligned. Such conserved regions across multiple sequences usually indicate an evolutionary relationship. For an unknown protein, for example, such conserved sequence segments identified through multiple alignment can be used in conjunction with other information to predict functionally important and evolutionarily conserved motifs within the proteins. Multiple alignment is also needed for the construction of phylogenetic trees. Figure 6.7 shows multiple alignment of five transporter proteins (partial sequence used) from mouse and rat using DDBJ ClustalW. The T-Coffee, CBRC (Computational Biology Research Center at the National Institute of Advanced Industrial Science and Technology, Japan) MAFFT, and EMBL-EBI MUSCLE all use ClustalW, so the output format is similar. NCBI COBALT has a very different output format. Multiple alignment is frequently done using Clustal programs, such as **ClustalW** and more recently **Clustal Omega**. Clustal Omega is a scaled-up version that enables thousands of sequences to be aligned. In order to perform multiple alignment, the ClustalW algorithm goes through a number of steps, as follows: it calculates all possible pairwise alignments of the input sequences; computes the score of each alignment, where the score reflects the distance between the two sequences; creates a dendrogram (guide tree) based on the matrix of the distance; and uses the dendrogram as the basis to perform multiple alignment, where closely related pairs of sequences are aligned first.

Multiple alignment programs can also be used to run pairwise alignment. Some online multiple alignment tools are shown in Table 6.2. Sequence input needs to be in FASTA or other formats.

6.6 ALIGNMENT ALGORITHMS, GAPS, AND GAP PENALTIES

An algorithm is a step-by-step procedure that utilizes a finite number of instructions for automated reasoning and the calculation of a function. The algorithm that drives global alignment is the Needleman–Wunsch algorithm, and the algorithm that drives local alignment is the Smith–Waterman algorithm. Both these algorithms are examples of **dynamic programming**. Dynamic programming is a method for solving complex problems by breaking them down into simpler subproblems. In the case of sequence alignment, dynamic programming involves setting up a two-dimensional matrix in which one sequence is listed vertically and the other sequence is listed horizontally; then calculating the scores, one row at a time. For example, a match can be given a 1, a mismatch a 0, and a gap a −1. A 100% perfect alignment will produce a diagonal straight line (with a negative slope) spanning from the top left to bottom right. If the alignment is not perfect, gaps are introduced in the matrix. For the sequence represented horizontally, gaps are introduced vertically, and for the sequence represented vertically, gaps are introduced

[d]The original submission accession number of mlst-1 is AB031959 and that of moatp-2 is AB031814. Partial sequence for each entry is used to save space.

```
LALIGN output
   mlst-1 (BAB03272.1; partial sequence)
   moatp-2 (BAB12445.1; partial sequence).
A hypothetical sequence THATISGREATANDFANTASTIC was added to the beginning
of mlst-1 protein and the end of moatp-2 protein


 49.1% identity in 212 aa overlap (31-241:2-210); score:  681 E(10000): 1.2e-61

               40        50        60        70        80        90
mlst-1 SKAAQPLRSEKTRHCDGFRIFLAALSFSYICKALGGVIMKSSITQIERRFDIPSSISGLI
       .:. . . .. .:  . .. :: ::. .:. :.:.:. :.: .::::::.: ::.:. :::
moatp- GKSEKEVATHGVRCFSKIKAFLLALTCAYVSKSLSGTYMNSMLTQIERQFGIPTSVVGLI
              10        20        30        40        50        60

              100       110       120       130       140
mlst-1 DGGFEIGNLLVIVFVSYFGSKLHRPKLIGTGCFIMGIGSILTALPHFFMGYYRYATEN-D
       .:.::::::::.:.::::::.::::: .::.:: .::.: .: ..::::.:: :.: :
moatp- NGSFEIGNLLLIIFVSYFGTKLHRPIMIGVGCAVMGLGCFLISIPHFLMGRYEYETTILP
              70        80        90       100       110       120

     150       160       170       180       190       200
mlst-1 ISSLHNSTLTCLVNQTTSLTGTSPEIMEKGCEKGSNSYTWIYVLMGNMLRGIGETPIVPL
        :.: .....:  :.: .:   .:      : :  .:  :::::.::..::.::::::.::
moatp- TSNLSSNSFVCTENRTQTL---KPTQDPTECVKEMKSLMWIYVLVGNIIRGMGETPIMPL
             130       140          150       160       170

     210       220       230       240
mlst-1 GVSYIDDFAKEGNSSMYLGTLHTIAMIGPILG
       :.:::.::::  :: .:.: :.:   :::..:
moatp- GISYIEDFAKSENSPLYIGILETGMTIGPLIG
      180       190       200       210

-------------------------------------------------------------------------------

 100.0% identity in 23 aa overlap (1-23:219-241); score:  144 E(10000): 1.5e-07

               10        20
mlst-1 THATISGREATANDFANTASTIC
       ::::::::::::::::::::::::
moatp- THATISGREATANDFANTASTIC
      220       230       240

-------------------------------------------------------------------------------

 40.0% identity in 15 aa overlap (117-131:95-109); score:   50 E(10000): 4.3e+02

        120       130
mlst-1 LIGTGCFIMGIGSIL
       ..: :::...:  .:
moatp- VMGLGCFLISIPHFL
          100
```

FIGURE 6.5 **LALIGN pairwise comparison.** LALIGN output of pairwise comparison of mlst-1 (BAB03272.1; partial sequence) and moatp-2 (BAB12445.1; partial sequence) each containing the hypothetical sequence "THATISGREATANDFANTASTIC." LALIGN produces an overall alignment of two protein sequences and also finds matching subsegments shared by these two input sequences. Note that in LALIGN the identities are reported by two dots and similar substitutions are reported by one dot.

horizontally, and the alignment is determined by a traceback step. The basic sequence alignment method is the dot matrix or dot plot method. In this method, two sequences being compared are written in the vertical and horizontal axes of the matrix. Then each residue is scanned and each match is given a dot; mismatches are left blank. When enough dots are lined up, they are connected (Figure 6.8).

In both global and local alignment, the final output is given an **alignment score**. **Gaps** have to be introduced to improve the alignment. The reason gaps are introduced is because one of the sequences may have gained or lost sequence characteristics (insertion−deletion) during evolution that did not happen with the other sequence. However, the number of gaps is kept to a minimum to keep the

```
Query: mlst-1  (BAB03272.1; partial sequence)
Sbjct: moatp-2 (BAB12445.1; partial sequence)
```

Range 1: 3 to 210 Graphics

Score	Expect	Method	Identities	Positives	Gaps	Frame
209 bits(533)	2e-71	Compositional matrix adjust	104/211(49%)	143/211(67%)	4/211(1%)	

```
Query  32   KAAQPLRSEKTRHCDGFRIFLAALSFSYICKALGGVIMKSSITQIERRFDIPSSISGLID  91
            K+ + + +   R     + FL AL+ +Y+ K+L G  M S +TQIER+F IP+S+ GLI+
Sbjct  3    KSEKEVATHGVRCFSKIKAFLLALTCAYVSKSLSGTYMNSMLTQIERQFGIPTSVVGLIN  62

Query  92   GGFEIGNLLVIVFVSYFGSKLHRPKLIGTGCFIMGIGSILTALPHFFMGYYRYATEN-DI  150
            G FEIGNLL+I+FVSYFG+KLHRP +IG GC +MG+G  L ++PHF MG Y Y T
Sbjct  63   GSFEIGNLLLIIFVSYFGTKLHRPIMIGVGCAVMGLGCFLISIPHFLMGRYEYETTILPT  122

Query  151  SSLHNSTLTCLVNQTTSLTGTSPEIMEKGCEKGSNSYTWIYVLMGNMLRGIGETPIVPLG  210
            S+L +++  C  N+T +L    P      C K   S  WIYVL+GN++RG+GETPI+PLG
Sbjct  123  SNLSSNSFVCTENRTQTL---KPTQDPTECVKEMKSLMWIYVLVGNIIRGMGETPIMPLG  179

Query  211  VSYIDDFAKEGNSSMYLGTLHTIAMIGPILG  241
            +SYI+DFAK  NS +Y+G L T   IGP++G
Sbjct  180  ISYIEDFAKSENSPLYIGILETGMTIGPLIG  210
```

Range 2: 219 to 241 Graphics

Score	Expect	Method	Identities	Positives	Gaps	Frame
50.4 bits(119)	4e-12	Compositional matrix adjust	23/23(100%)	23/23(100%)	0/23(0%)	

```
Query  1    THATISGREATANDFANTASTIC  23
            THATISGREATANDFANTASTIC
Sbjct  219  THATISGREATANDFANTASTIC  241
```

FIGURE 6.6 **NCBI BLAST pairwise alignment.** The two partial sequences depicted in Figure 6.5 were also aligned using NCBI BLAST pairwise alignment. Like LALIGN, NCBI BLAST pairwise alignment also produces an overall alignment of two protein sequences, and also finds matching subsegments shared by these two sequences. The hypothetical sequence "THATISGREATANDFANTASTIC" has been identified as a subsegment of 100% identity between the two proteins.

alignment meaningful; otherwise an artificially high alignment score can be obtained even when the two sequences are not related. The **gap penalty** value is subtracted from the gross alignment score to obtain the final alignment score (alignment score and scoring matrix are discussed in the next section). *The insertion of no more than 1 gap per 20 amino acid residues is ideal but that is not possible in most cases*. For each gap opened, a **gap-opening penalty** value is assigned, and for each gap extended, a **gap-extension penalty** value is assigned. A gap-opening penalty is always much higher than a gap-extension penalty. Often, a default value of -10 for a gap-opening penalty and -1 for a gap-extension penalty are used. However, these values can be different and can also be adjusted by the user. This type of differential penalty for gap opening and gap extension is called **affine gap penalty**. There are other types of gap penalties, such as constant gap penalty, linear gap penalty, and proportional gap penalty, but for all practical purposes affine gap penalty is the most relevant for sequence alignment. Affine gap penalty is calculated as follows:

$$G_t = G_o + G_e \times L_n, \tag{6.3}$$

where G_t = total gap penalty, G_o = gap-opening penalty, G_e = gap-extension penalty, and L_n = length of the extension gaps. For any given block of gaps, L_n = # of total gaps $-$ 1, because the first gap is the opening, the rest in the block are extensions.

When running an alignment, it is better to use the default value with the default matrix. This is because there is no rule for setting the best gap-opening and -extension penalty values for a given pair of sequences being compared; thus, changing the gap-opening and -extension penalty values may influence the nature of the alignment. For example, setting gap-opening and -extension penalty values that are a lot higher than the default values creates alignments that contain fewer internal gaps and more end gaps; also local alignments containing gaps may be split into several shorter alignments.

```
Sequence format is CLUSTAL (CLUSTAL 2.1 Multiple Sequence Alignments
Sequence 1: moatp-2      287 aa (Accession #: BAB12445.1; partial sequence)
Sequence 2: moatp-5      287 aa (Accession #: AAG60350.1; partial sequence)
Sequence 3: moatp-1      287 aa (Accession #: BAB12444.1; partial sequence)
Sequence 4: rlst-1a      287 aa (Accession #: AAF87098.1; partial sequence)
Sequence 5: mlst-1       287 aa (Accession #: BAB03272.1; partial sequence)

CLUSTAL 2.1 multiple sequence alignment

moatp-2         MGKSE--------KEVATHGVRCFSKIKAFLLALTCAYVSKSLSGTYMNSMLTQIERQFG 52
moatp-5         MGEPG--------KRVGIHRVRCFAKIKVFLLALIWAYISKILSGVYMSTMLTQLERQFN 52
moatp-1         MEETE--------KKVATQEGRFFSKMKVFLMSLTCAYLAKSLSGVYMNSMLTQIERQFG 52
rlst-1a         MDHTQQSRKAAEAQPSRSKQTRFCDGFKLFLAALSFSYICKALGGVVMKSSITQIERRFD 60
mlst-1          MDQTQHPSKA--AQPLRSEKTRHCDGFRIFLAALSFSYICKALGGVIMKSSITQIERRFD 58
                * ..         :    .  *    :: ** :*  :*:.* *.*. *.: :**:**:*.

moatp-2         IPTSVVGLINGSFEIGNLLLIIFVSYFGTKLHRPIMIGVGCAVMGLGCFLISIPHFLMGR 112
moatp-5         ISTSIVGLINGSFEMGNLLVIVFVSYFGTKLHRPIMIGVGCAVMGLGCFIISLPHFLMGR 112
moatp-1         IPTSVVGFITGSFEIGNLLLIVFVSYFGRKLHRPIIIGVGCVVMGLGCFLMASPHFLMGR 112
rlst-1a         IPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGIGCFIMGIGSILTALPHFFMGY 120
mlst-1          IPSSISGLIDGGFEIGNLLVIVFVSYFGSKLHRPKLIGTGCFIMGIGSILTALPHFFMGY 118
                *.:*: *:* *.**:****:*:****** ***** :** ** :**:*.:: : ***:**

moatp-2         YEYETTILPTSNLSSNSFVCTENRTQTLK---PTQDPTECVKEMKSLMWIYVLVGNIIRG 169
moatp-5         YEYETTISPTSNLSSNSFLCVENRSQTLK---PTQDPAECVKEIKSLMWIYVLVGNIIRG 169
moatp-1         YKYETTISPTSNLSSNSFLCIENRTQTLK---PTQDPTECVKEIKSLMWIYVLIGNTMRG 169
rlst-1a         YKYAKEND-IGSLGNSTLTCFINQMTSPTGPSPEIVEKGCEKGLKSHMWIYVLMGNMLRG 179
mlst-1          YRYATEND-ISSLHNSTLTCLVNQTTSLTGTSPEIMEKGCEKGSNSYTWIYVLMGNMLRG 177
                *.* .     ..* ..:: *  *:  : .   *      * *  :*  *****:** :**

moatp-2         MGETPIMPLGISYIEDFAKSENSPLYIGILETGMTIGPLIGLLLGSSCANIYVDTGSVNT 229
moatp-5         IGETPIMPLGISYIEDFAKSENSPLYIGILEVGKMIGPILGYLMGPFCANIYVDTGSVNT 229
moatp-1         IGETPIMPLGISYIEDFAKSENSPLYIGILEMGKIVGPIIGLLLGSFFARVYVDIGSVNT 229
rlst-1a         IGETPIVPLGISYLDDFAKEGHTSMHLGTLHTIAMIGPILGFIMSSVFAKIYVDVGYVDL 239
mlst-1          IGETPIVPLGVSYIDDFAKEGNSSMYLGTLHTIAMIGPILGFIMSSVFAKLYVDVGYVDL 237
                :*****:***:**::****. ::.:::* *.    :**::* ::..  *.:*** * *:

moatp-2         DDLTITPTDTRWVGAWWIGFLVCAGVNILTSIPFFFFPKTLLKEGLQ 276
moatp-5         DDLTITPTDTRWVGAWWIGFLVCAGVNVLTSIPFFFFPKTLPKEGLQ 276
moatp-1         DDLTITPTDTRWVGAWWIGFLVCAGVNILTSIPFFFFPKTLPKKELQ 276
rlst-1a         NSVRITPNDARWVGAWWLSFIVNGLLCITSSIPFFFL---------- 276
mlst-1          RSVRITPQDARWVGAWWLGFIVNGLLCIICSIPFFFLPK-------- 276
                 .: *** *:*******:.*:* . : :  ******:
```

FIGURE 6.7 **Multiple alignment using ClustalW from DDBJ.** Five transporters from rat and mouse have been aligned. Identical amino acids are indicated by a star (*), whereas similar substitutions are indicated by a colon (:). To save space, only the first 287 amino acids from each transporter have been used for the alignment.

TABLE 6.2 Online Multiple Alignment Tools

Online Tool	URL
COBALT (NCBI)	http://www.ncbi.nlm.nih.gov/tools/cobalt/cobalt.cgi?link_loc = BlastHomeLink[7]
ClustalW (DDBJ)	http://clustalw.ddbj.nig.ac.jp/index.php?lang = en[8,9]
MAFFT (CBRC)	http://mafft.cbrc.jp/alignment/server/[10]
MUSCLE (EMBL-EBI)	http://www.ebi.ac.uk/Tools/msa/muscle/[11]
T-Coffee	http://www.tcoffee.org/Projects/tcoffee/. Then click any of the server links on this page, such as http://www.tcoffee.org/ and from there the type of alignment program needed for analysis[12]

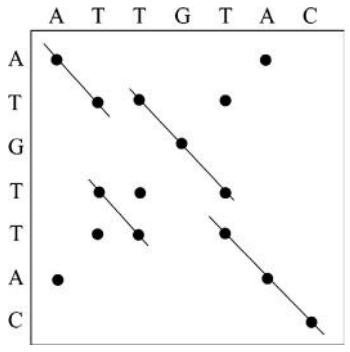

FIGURE 6.8 **Comparison of two sequences using dot matrix or dot plot.**

6.7 SCORING MATRIX, ALIGNMENT SCORE, AND STATISTICAL SIGNIFICANCE OF SEQUENCE ALIGNMENT

A raw alignment score can be calculated based on the following simple formula:

$$S = \Sigma_i + \Sigma_m - G_t, \tag{6.4}$$

where S = raw score, Σ_i = total score for identities, Σ_m = total score for mismatches, and G_t = total gap penalty.

For both nucleic acids and proteins, the alignment score is calculated using a **scoring matrix**. A scoring matrix is a set of values representing the likelihood of one residue being substituted by another during sequence divergence through evolution. This is why the scoring matrix is also known as the **substitution matrix**.

A scoring matrix for comparing DNA sequences can be simple because there are only four nucleotides and the mutation frequencies are assumed to be equal (the Jukes and Cantor assumption). A high positive score (e.g. 5) is assigned for a match and a low negative score (e.g. −4) for a mismatch, thus creating a simple model. However, the frequency of transition mutations (purine replaced by purine or pyrimidine replaced by pyrimidine) is higher than transversion mutations (purine replaced by pyrimidine or vice versa). To deal with this differential mutation frequency, sophisticated statistical models have been developed by Kimura and others. For generating a DNA sequence-alignment score, the simple scoring matrix is still used, such as the NUC4.2 and NUC4.4 DNA scoring matrices. These matrices can be obtained from the NCBI (ftp://ftp.ncbi.nih.gov/blast/matrices/).

Scoring matrices for amino-acid substitutions are more complex, reflecting the similarity of physicochemical properties, as well as the likelihood of one amino acid being substituted by another at a particular position in homologous proteins. The scoring matrices for proteins are 20×20 matrices. Two well-known types of scoring matrices for proteins are PAM and BLOSUM.

6.7.1 PAM Matrices

PAM (point accepted mutation—that is, accepted point mutation—also called percent accepted mutation) matrices were first developed by Margaret Dayhoff and colleagues in 1978 and hence are also known as Dayhoff PAM matrices. A PAM represents a substitution of one amino acid by another that has been fixed by natural selection because either it does not alter the protein function or it is beneficial to the organism. *In a PAM1 matrix, which is the original PAM matrix generated, a PAM unit is an evolutionary time over which 1% of the amino acids in a sequence are expected to undergo accepted mutations, resulting in 1% sequence divergence.* Construction of a PAM1 matrix begins with alignment of the full-length sequences, reconstruction of the phylogenetic tree, and determination of the ancestral sequences for the internal nodes of the tree (see Chapter 9 for a description of the phylogenetic tree). Each computed ancestral sequence is then used to calculate the number and frequency of substitutions in the sequences along each branch arising from the node. The values in the matrix represent the probability that the amino acid in a column will be replaced by the amino acid in row in a given evolutionary time (1 PAM unit in a PAM1 matrix). From the computed probability, the percent probability can be determined. A PAM1 matrix is often displayed after multiplying each entry by 10,000.

The relationship between % amino acid substitution and the number of PAM units is not linear; thus, the above definition applies only when the divergence between two sequences is low. As the divergence increases beyond ~20%, this relationship falls apart. For example, a 100-PAM-unit divergence does not mean 100% substitution. A 100-PAM-unit divergence can be achieved by substituting ~55% of the amino acid residues, and a 200-PAM-unit divergence can be achieved by substituting ~75% of the amino acid residues. The PAM1 matrix was built by aligning closely related protein sequences (71 protein families) that had at least 85% sequence identity.

Subsequently, in order to deal with protein sequences that are more diverged and distantly related, other PAM matrices, such as PAM100 and PAM250, were generated. These later PAM matrices were generated by multiplying the PAM1 matrix by itself hundreds of times. For example, the PAM250 matrix can be obtained by multiplying the PAM1 matrix by itself 250 times over. Figure 6.9 shows the PAM250 substitution matrix. The values in the matrix are **log odds scores** (see Box 6.1).

6.7.1.1 PET91 Matrix

At the time PAM matrices were developed, the number of available protein sequences and the amount of protein family information as well as the knowledge of protein three-dimensional structure were limited. Obviously, PAM matrices could be prone to certain inherent flaws, such as (1) the assumption that each amino acid in a sequence is equally mutable, (2) multiplying a PAM1 matrix n number of times to obtain a PAMn matrix can amplify any error in the original matrix, and (3) the amino-acid-residue profiles of the proteins used to generate a PAM matrix do not

PAM250 matrix

Ala	A	2																	
Arg	R	-2	6																
Asn	N	0	0	2															
Asp	D	0	-1	2	4														
Cys	C	-2	-4	-4	-5	12													
Gln	Q	0	1	1	2	-5	4												
Glu	E	0	-1	1	3	-5	2	4											
Gly	G	1	-3	0	1	-3	-1	0	5										
His	H	-1	2	2	1	-3	3	1	-2	6									
Ile	I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5								
Leu	L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6							
Lys	K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5						
Met	M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6					
Phe	F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9				
Pro	P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6			
Ser	S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2		
Thr	T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	
Trp	W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17
Tyr	Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10
Val	V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

FIGURE 6.9 **A PAM250 substitution matrix made by writing the amino acids in alphabetical order.**

necessarily represent the residue profiles of all protein families.

Jones et al.[13] updated the PAM matrix by taking into account 2621 families of sequences (>16,000 homologous protein sequences) from the Swiss-Prot database. The sequences were clustered at 85% identity level as was done in the original PAM matrix, and the raw mutation frequency matrix was processed in a similar way as in the PAM matrix. This updated PAM matrix is called the PET91 matrix (PET91 = pair exchange table for year 1991). Thus, PET91 takes into account the substitutions that were poorly represented in the original Dayhoff matrix. The overall character of PAM and PET91 matrices is similar.

Each PAM matrix is designed to be used for comparing sequences that are evolutionarily diverged by a specific number of PAM units—that is, by a specific length of evolutionary time. The suffix (number) with PAM indicates evolutionary distance; the greater the number, the greater is the distance. For example, the PAM120 matrix is ideal for comparing sequences that have diverged by 120 PAM units during evolution. Assuming $\sim 10^7$ years (10 million years) as a PAM unit of evolutionary time, 120 PAM units of evolutionary time will correspond to 120×10^7, or 1200 million years. The higher the PAM suffix (number), the better it is in aligning more divergent sequences. PAM matrices have been developed based on the Markovian evolutionary model. The Markovian evolutionary model is the application of the Markov model to predict the probability of the state of a variable over evolutionary time, such as the probability of occurrence of an amino acid at a particular position in a protein sequence. For protein evolution, the Markov model can look at a long sequence of amino acids and analyze the likelihood that an amino acid will substituted by another. The Markov model assumes that each substitution is an independent, "memoryless" process.

6.7.2 BLOSUM

BLOSUM will be referred to as BLOSUM matrix here. BLOSUM (blocks substitution matrices) scoring matrices were proposed by Steven Henikoff and Jorja Henikoff in 1992.[14] BLOSUM represents an alternative set of scoring matrices, which are widely used in sequence-alignment algorithms. Like PAM, BLOSUM matrices are also log-odds matrices. BLOSUM matrices were developed based on multiple alignment of 500 groups of related protein sequences, which yielded > 2000 blocks of conserved amino-acid patterns. Blocks are ungapped multiple sequence alignments corresponding to the most conserved regions of the proteins involved. Henikoff and Henikoff used their BLOCKS database of trusted alignments. In each multiple alignment, the sequences showing similar % identity were clustered into groups and averaged. Using these groups, the substitution frequencies for all pairs of amino acids were calculated and the matrix was developed. Therefore, the blocks of ungapped multiple sequence alignments, which are the cornerstone of BLOSUM matrices, reveal the evolutionary relationship between proteins. The BLOCKS database was developed to host these multiple sequence alignments that reveal the blocks. By 1996, there were $\sim$3000 blocks reported, based on 770 protein families.[15] Different BLOSUM matrices differ in the % sequence identity used in clustering. Therefore, BLOSUM62

BOX 6.1

PROBABILITY, ODDS, LOG-ODDS, SCORING MATRIX

Probability is a measure of how often an event may occur, whereas **odds** is a measure based on the probability that an event may ever occur. Odds is the ratio of probabilities.

1. Probability of event X = # of events X/# of all possible events
   (e.g. when a die is rolled, the probability that the die will land with the six-side up is 1/6. In this case, the probability of the alternative event—that is, the probability *against* the die landing with the six-side up—is 5/6)
2. Odds of event X = probability of event X/probability of the alternative event (i.e. probability *against* event X)
   (e.g. in the above example, the odds of the die landing with the six-side up is the ratio of the two probabilities—that is, (1/6) ÷ (5/6) = 1/5).

In the case of amino-acid substitution (mutation), the odds of substitution means the ratio of the probability that one specific amino acid is preferentially substituted by another specific amino acid during evolution to the probability that such substitution is random. By assigning a score (odds score) to all possible pairs of amino-acid substitution, a scoring matrix can be obtained. Substitution matrices are scoring matrices that use the logarithm of the odd score, called the **log-odds score**. Use of the log-odds score instead of the odds score (which is the ratio of probabilities) allows for addition of the scores instead of multiplication of the probabilities. All algorithms for sequence comparison use some kind of scoring scheme.

If the substitution of two residues i and j is considered, the mathematical logic for the calculation of log-odds will be as follows:

1. The probability that i and j are aligned based on their evolutionary relationship of substitution is $P_e = f_i \times f_{ji}$ (f_i = frequency of residue i and f_{ji} = frequency of residue j substituting for i).
2. The probability that i and j are aligned by random chance is $P_r = f_i \times f_j$ (f_i = frequency of residue i and f_j = frequency of residue j).
3. Hence, the odds = $P_e/P_r = (f_i \times f_{ji})/(f_i \times f_j) = f_{ji}/f_j$.
4. Log odds = log (f_{ji}/f_j).
5. If (f_{ji}/f_j) = 1, then log (f_{ji}/f_j) = 0. This means that the odds of i and j being aligned based on their evolutionary relationship of substitution is the same as that by random chance.
6. If (f_{ji}/f_j) > 1, then log (f_{ji}/f_j) = positive. This means that the odds of i and j being aligned based on their evolutionary relationship of substitution is greater than by random chance.
7. If (f_{ji}/f_j) < 1, then log (f_{ji}/f_j) = negative. This means that the odds of i and j being aligned based on their evolutionary relationship of substitution is lower than even by random chance.

Therefore, a negative log-odds score means that the cost of such substitution to the protein structure and function is high, and normally such substitutions are not encouraged by natural selection. For example, the PAM250 matrix shows that the likelihood of valine being substituted by isoleucine, another hydrophobic amino acid, is higher (4) than by any one of the four hydrophilic and charged amino acids—arginine, lysine, aspartic acid, and glutamic acid (−2 for each one).

means that the sequences used to create this matrix have approximately 62% identity. Substitution frequencies weigh more heavily by protein sequences having less than 62% identity. Therefore, BLOSUM62 is useful for aligning and scoring proteins that show less than 62% identity. Shown below is an example of an ungapped multiple alignment. The conserved amino acids are shaded for identification.

```
GSFEIGNLLLII
GSFEMGNLLVIV
GSFEIGNLLLIV
GGFEIGNLLVIV
GGFEIGNLLVIV
```

Henikoff and Henikoff tested the performance of hierarchical multiple alignment of three serine proteases using BLOSUM45, BLOSUM62, BLOSUM80, PAM120, PAM160, and PAM250 matrices. All BLOSUM matrices performed better than PAM matrices; the number of residues misaligned was three to five times lower when BLOSUM matrices were used compared to PAM matrices. BLOSUM62 performed slightly better than BLOSUM45 and BLOSUM80. The reader is urged to read an excellent short primer by Sean Eddy on how the BLOSUM62 matrix was developed.[16]

BLOSUM62 matrix

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Ala	A	4																			
Arg	R	-1	5																		
Asn	N	-2	0	6																	
Asp	D	-2	-2	1	6																
Cys	C	0	-3	-3	-3	9															
Gln	Q	-1	1	0	0	-3	5														
Glu	E	-1	0	0	2	-4	2	5													
Gly	G	0	-2	0	-1	-3	-2	-2	6												
His	H	-2	0	1	-1	-3	0	0	-2	8											
Ile	I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

FIGURE 6.10 **BLOSUM62 substitution matrix made by writing the amino acids in alphabetical order.**

For a PAM matrix, the higher the suffix number, the better it is in dealing with evolutionarily distant protein alignment, and the lower the suffix number, the better it is in dealing with evolutionarily closer protein alignment. In contrast, for BLOSUM matrices, the suffix numbering system is the opposite of PAM matrices; hence, the higher the suffix number, the better it is in dealing with evolutionarily closer protein alignment. In their publication, Henikoff and Henikoff drew equivalence between different PAM and BLOSUM matrices based on relative entropy[e]. For BLOSUM matrices, relative entropy increases nearly linearly with increasing clustering percentage. Based on relative entropy, Henikoff and Henikoff concluded the following:

PAM250 ≈ BLOSUM45 (relative entropy 0.4 bit)
PAM120 ≈ BLOSUM80 (relative entropy 1 bit)
PAM160 ≈ BLOSUM62 (relative entropy 0.7 bit).

BLOSUM62 is the most widely used amino-acid scoring matrix (including by BLAST algorithms) for scoring amino-acid alignment for database searches (discussed below). Figure 6.10 shows a BLOSUM62 matrix. The NCBI FTP site from where various nucleic-acid and protein scoring matrices can be downloaded is ftp://ftp.ncbi.nih.gov/blast/matrices/.

To summarize, PAM and BLOSUM matrices can be compared as follows:

1. PAM matrices are constructed based on an evolutionary model—that is, from the estimation of mutation rates through constructing phylogenetic trees and inferring the ancestral sequence—but BLOSUM matrices are constructed based on direct observation of ungapped multiple alignment-driven sequence relationships. *Thus, PAM matrices are often used for reconstructing phylogenetic trees, whereas BLOSUM matrices are suitable for local sequence alignments*.
2. PAM matrix construction involves global alignment of the full-length sequences consisting of both conserved and diverged regions, but BLOSUM matrix construction involves local sequence alignment of conserved sequence blocks. Additionally, when Henikoff and Henikoff compared the two equivalent matrices PAM160 and BLOSUM62, they found that BLOSUM62 is less tolerant to hydrophilic-amino-acid substitution, but more tolerant to hydrophobic-amino-acid substitution than PAM160. Also, for rare amino acids, such as cysteine and tryptophan, BLOSUM62 is typically more tolerant to mismatches than PAM160.

[e]Relative entropy (also known as Kullback–Leibler divergence) is a measure of the difference between two states or two probability distributions P1 and P2. For example, P1 could be the frequency of occurrence of an amino acid at a given position in a multiple alignment relative to the background frequency, P2, of a random sample. Thus, in the context of sequence alignment, relative entropy can be calculated to determine sequence conservation relative to the background, and it is measured as the average information per residue pair in bit units. When relative entropy is 0, the target (or observed) distribution of pair frequencies is the same as the background (or expected) distribution. Relative entropy increases as two distributions become more distinguishable. An online tool for the calculation of relative entropy within sequence alignment blocks is H-BLOX, which can be accessed at http://gecco.org.chemie.uni-frankfurt.de/h-blox/hblox.html.

Most bioinformatics analysis tools provide users with a default matrix, but the default matrix may not be the most suitable matrix for the user's need. Therefore, it is important to be mindful about the utility of a specific matrix for a specific purpose. There are essentially three levels of similarity-searching alignments: that of closely related sequences, that of divergent sequences, and that of sequences intermediate between the closely related and divergent sequences. Both PAM and BLOSUM matrices can be used for this purpose. The following example shows the PAM−BLOSUM matrix equivalence, and their preferred use:

PAM100 ≈ BLOSUM90 (for less divergent proteins)
PAM120 ≈ BLOSUM80
PAM160 ≈ BLOSUM62 } (for most other proteins)
PAM200 ≈ BLOSUM52
PAM250 ≈ BLOSUM45 (for more divergent proteins)

In general, BLOSUM matrices are widely used for detecting local alignments. BLOSUM62 is the most frequently used matrix for detecting the majority of weak protein similarities, and BLOSUM45 is very suitable for detecting long and weak alignments.

While aligning unknown sequences, if one wants to use the most appropriate matrix based on how similar the sequences are, one has to first try multiple matrices and then use the one that gives the highest ungapped alignment score.

6.7.3 Scoring Sequence Alignment and Statistical Significance of Sequence Alignment

The calculation of alignment scores involves addition of the match/mismatch values from the matrix for every nucleotide base or amino acid residue involved in the alignment to obtain a gross alignment score. Then the total gap penalty is calculated. The total gap penalty value is subtracted from the gross alignment score value to obtain the final alignment score. The terminal gaps may or may not be penalized, depending on the program used. For example, in local alignment (Smith−Waterman algorithm), a terminal gap penalty does not make sense, whereas in global alignment (Needleman−Wunsch algorithm), a terminal gap penalty may be applied depending on the program.

Different alignments should not be directly compared based on their raw score (S). For example, a not-so-good long alignment may get a higher S than a very good short alignment. Thus, different alignments should only be compared after normalization. This is achieved by determining the statistical significance of the score.

The statistical significance of the raw score, S, of an alignment is assessed to determine whether the observed alignment is specific or could be the result of random chance. This is done by creating many random sequences of the same length from one of the two aligned sequences by shuffling the sequence and running the alignment again. Typically this reshuffling and realignment process is repeated 200 times or more. Each alignment using these random sequences produces an alignment score (s). These scores ($s_1 \ldots s_n$) are plotted to generate a distribution pattern, a threshold of significance is set, and the original score (S) is compared against this distribution. If the S is located at one end of the distribution (extreme value distribution) that means that the alignment is not likely to be produced by random chance.

6.7.3.1 P-*Value*

The ***P*-value** of an alignment represents the **probability** of obtaining a score $\geq S$ by chance. For example, if the P-value is 10^{-5}, it means that the probability of obtaining an alignment with a score $\geq S$ is 1 out of 10^5. Thus, different alignments can be compared based on their P-values. The P-value ranges from 0 to 1; the closer it is to 0, the better is the alignment.

6.7.3.2 Z-*Score*

In the statistical sense, Z is the distance between S and the mean of scores obtained using randomized sequences. The Z-score is calculated by repeating the reshuffling and realignment process, as described above, and noting the raw score (s) of each alignment using the randomized sequences ($s_1 \ldots s_n$). The mean ($\bar{x}$) and the standard deviation (σ) of $s_1 \ldots s_n$ are calculated and from these the Z-score of the target alignment can be determined.

The calculation of the Z-score assumes that the alignment of the shuffled random sequences shows a normal distribution. Hence, the farther the alignment raw score S is away from the $\bar{x}$ of $s_1 \ldots s_n$, the more likely it is to be significant. In a statistical sense, the Z-score reflects the extent to which S is an outlier from the population. A $Z = 5$ means the S is 5σ above the $\bar{x}$ of $s_1 \ldots s_n$. By convention, a $Z > 7$ indicates a significant alignment and it is likely that the two sequences being aligned are homologs; it also indicates that the alignment of the two sequences likely reflects the alignment of structurally and functionally related amino acid residues of the proteins. Another interpretation of the Z-score is as follows[17]:

$Z > 20$: two sequences are definitely homologous (Family)
Z between 10 and 20: two sequences most likely homologous (Family/Superfamily)
Z between 6 and 8: two sequences are less likely to be homologous
$Z < 6$: not significant.

PRSS (current version PRSS3; http://www.ch.embnet.org/software/PRSS_form.html)[18] is freely available web-based software that can be used to evaluate the significance of a protein or DNA sequence-similarity score. PRSS compares two sequences and calculates the optimal similarity scores, and then repeatedly shuffles the second sequence, and calculates optimal similarity scores using the Smith–Waterman algorithm. An extreme value distribution (EVD) is then fit to the shuffled-sequence scores. In the PRSS output, the left-most column represents the normalized similarity scores; and the E ( ) column on the right represents the number of sequences expected to achieve the score in the first column.

6.7.3.3 E-*Value*

This is particularly relevant in relation to sequence-similarity searching using BLAST and FASTA, which are discussed later in this chapter. The ***E*-value** is the **expectation** value that indicates the number of alignments with a score $\geq S$ that one can expect to find by chance in a database of size N. Hence, the E-value is dependent on the database size and the query length. The closer the E-value to 0, the better is the alignment. For $E < 1e-2$ $(=1\times10^{-2}=0.01)$, $P \approx E$. *The* E*-value is the most widely used measure for estimating the quality of sequence alignment—that is, the extent of sequence similarity*.

The typical threshold for the E-value when judging homology, particularly using BLAST, is $E \leq 1e-5$ $(=1\times10^{-5})$, and the lower the value, the better it is. For BLAST (both nucleotide and protein), the default E-value is set at 10 in the **Expect threshold** box under **Algorithm parameters** (lower left corner of the BLAST home page). This means that 10 matches are expected to be found merely by chance, according to the stochastic model of Karlin and Altschul (1990).[19] It also means that the BLAST output will not report any alignment with an E-value greater than 10. Obviously, when the E-value is increased from the default value of 10, a larger number of chance matches will be reported. In contrast, lowering the default value makes the search more stringent and fewer chance matches are reported. The default E-value should be increased if searching for short sequence matches, because setting a lower E-value will automatically exclude the short matches as spurious and these will not be reported. In such cases, the default value in the "Expect threshold" box can be manually changed. *Alternatively, the nucleotide and protein BLAST programs of the NCBI automatically adjust the* E*-value* if the query, either nucleotide or amino acid, is of length 30 or less.

6.7.3.4 *Bit Score*

The **bit score** (S') is a normalized raw score expressed in *bits*; it is an estimate of the search space one has to search through—that is, the number of sequence pairs one has to score—before one can come across a raw alignment score $\geq S$, by chance.

For example, a bit score of 30 means that, on average, one has to score 2^{30} $(=1$ billion) sequence pairs before one will come across a score $\geq S$, by chance. Usually, good alignments produce a bit score > 50. *It should be emphasized that the bit score is dependent on sequence length, and short sequences may not produce high bit scores despite very high identity*.

To summarize the utility of the statistical estimates of sequence alignment in simple terms, **the better the alignment (e.g. homologous sequences), the lower the *P*- and *E*-values, and the higher the *Z*- and bit scores.**

6.8 DATABASE SEARCHING WITH THE HEURISTIC VERSIONS OF THE SMITH–WATERMAN ALGORITHM—BLAST AND FASTA

Alignment programs that use dynamic programming algorithms, such as the Needleman–Wunsch and Smith–Waterman algorithms, require long processing times, particularly when searching a huge database. In order to circumvent this computational limitation, heuristic methods have been developed. A **heuristic** method (algorithm) estimates the best solution without considering every possible outcome; thus, a heuristic method does not guarantee to find the best solution, but finds good solutions, and thereby has high speed and is time efficient. Two examples of heuristic methods are the Basic Local Alignment Search Tool (BLAST) and FAST-All (FASTA). FASTA is pronounced "fast A". It stands for "FAST-All" because it is an extension of "FAST-P" for proteins and "FAST-N" for nucleotides; therefore, FASTA works with all alphabets associated with proteins and nucleic acids.

6.8.1 BLAST and its Utility

Currently, the most widely used heuristic algorithm is BLAST, developed by Altschul and colleagues.[20] The BLAST algorithm allows a DNA or protein **query** sequence to be compared with sequences in the database. The main idea behind BLAST searching is that homologous sequences are likely to contain a short, high-scoring similarity region, called a **word** or **hit** (**W**). Each word (hit) gives a **seed** that triggers

the alignment and BLAST tries to extend on both sides of the seed. The word size—i.e. the length of the seed—may vary. For nucleotides (**blastn**), the default word size is 11 and the smallest word size is 7; for proteins (**blastp**), the default word size is 3 and the smallest word size is 2. For **megablast** (highly similar sequences), the default word size is 28 and the smallest word size is 16 for nucleotides. These parameters can be adjusted by clicking "**Algorithm parameters**" in the lower left corner of the BLAST page. For a nucleic-acid sequence alignment, the seed should match completely in order to trigger the alignment; for proteins, the match may or may not be exact. In order to create an alignment, the BLAST algorithm breaks the query sequence into short subsequences. Typically, BLAST is designed to find local regions of similarity, but can be expected to run about two orders of magnitude faster than the Smith–Waterman algorithm. An important parameter governing the sensitivity of BLAST searches is the length of the initial words (hits).

Database searching is done for various reasons, such as finding relationships between the query sequence and other sequences in the databases, understanding the likely function of a sequence, identifying regulatory elements, understanding genome evolution, or assisting in sequence assembly. In designing probes and primers, the selected nucleic acid sequence is compared with other sequences in the database to determine the specificity and uniqueness of the selected sequence. Therefore, a BLAST search can help determine the identity of nucleic acid and protein sequences, reveal whether these sequences represent new genes and proteins, discover variants of existing genes and proteins, discover potential orthologs and paralogs of a sequence, determine whether a gene or protein is present in other organisms, or determine whether a nucleic acid sequence is expressed.

In a BLAST search, the sequence that is subject to comparison is termed the **query**. This query sequence is subjected to BLAST search against all sequences in the database. The search retrieves all sequences showing similarity with the query sequence. These sequences are called **subject** (or **target**).

6.8.2 Various BLAST Programs for Analysis

At the NCBI, there are several BLAST resources, which can be grouped as **basic BLAST** and **specialized BLAST**.

Basic BLAST offers a few options, such as **blastn** (searches a nucleotide database using a nucleotide query), **blastp** (searches a protein database using a protein query), **blastx** (searches a protein database using a translated nucleotide query), **tblastn** (searches a translated nucleotide database using a protein query), and **tblastx** (searches a translated nucleotide database using a translated nucleotide query).

Specialized BLAST provides many specialized/advanced options, such as Primer-BLAST, trace archives, conserved domains, conserved domain architecture, gene expression profile (GEO), immunoglobulin search (IgBLAST), single nucleotide polymorphism (SNP) flank search, vector contamination screening (vecscreen), Align, PubChem BioAssay search, searching SRA transcript and genomic libraries, Multiple Alignment Tool, Global Sequence Alignment Tool, or searching the RefSeqGene database.

For a detailed description of each of these different BLAST programs and their use, refer to the NCBI reference resource (http://blast.ncbi.nlm.nih.gov/).

6.8.2.1 Megablast, Blastn, and Discontinuous Megablast

Currently, the nucleotide BLAST program offers three options for searching sequences for hits in the database with different degrees of similarity. These are megablast, blastn, and discontinuous megablast.

Megablast is optimized for highly similar sequences. It efficiently finds long alignments between highly similar (>95%) sequences, and thus is the best tool to find the identical match to the query sequence. The default word size is 28 and the lowest word size is 16.

Blastn is optimized for somewhat similar sequences. The reason blastn is more sensitive than megablast is because it uses a shorter default word size (11). Because of this, blastn is better than megablast at finding alignments to related nucleotide sequences from other organisms. Reducing the word size from 11 (default) to 7 (lowest) increases the **sensitivity** of search—that is, increases the number of positive hits.

Discontinuous megablast is optimized for more dissimilar sequences. Instead of using the exact word match as seed for an alignment extension, discontinuous megablast uses a noncontiguous word within a longer window of template. As a result, discontinuous megablast using the same size of the initial hit is even more sensitive and efficient than standard blastn using the same word size.

6.8.2.2 Searching for Short, Nearly Exact Matches

For searching short nucleotide-sequence matches, algorithm parameters can be manually adjusted as follows: select blastn→select the non-redundant (nr) nucleotide database (unless a specific database is needed)→select "Somewhat similar sequences (blastn)"→click on

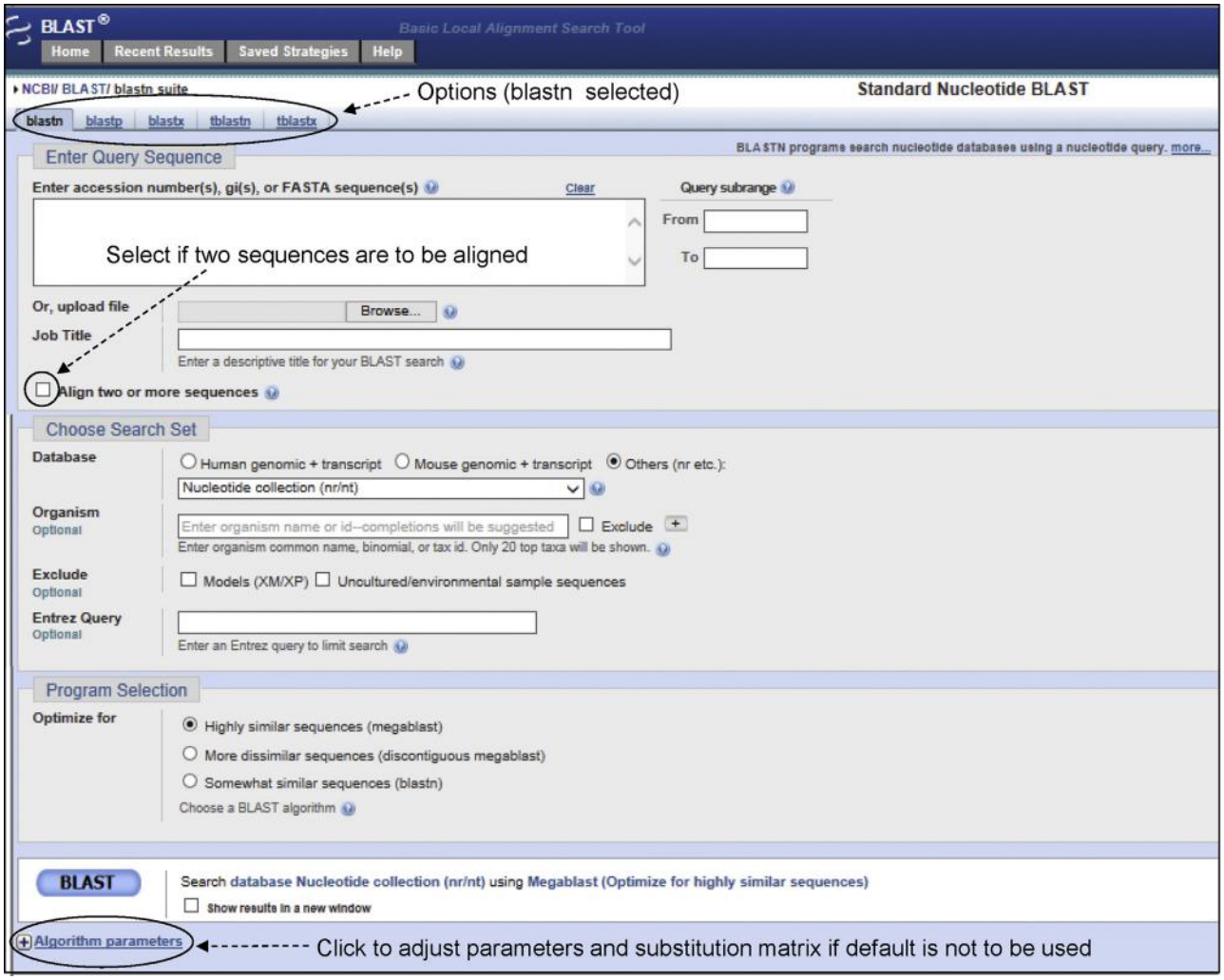

FIGURE 6.11 **NCBI BLAST home page of nucleotide blast.** By clicking the tabs at the top (circled), other BLAST tools can be obtained. For regular BLAST, the sequence can be entered in plain text format. For pairwise alignment, the small box (indicated by an arrow) can be checked and a second box appears where the other sequence can be entered. The "Algorithm parameters" can be clicked and the default setting can be changed.

"Algorithm parameters"→check the short queries box→filter[f] setting to remain off→select the word size 7→change expect threshold to 1000 (or as necessary). For searching short protein-sequence matches, algorithm parameters can be manually adjusted as follows: select blastp→select the non-redundant (nr) protein database (unless a specific database is needed)→check the short queries box→filter setting to remain off→select the word size 2→change expect threshold to 10000 (or as necessary)→select PAM30 as the scoring matrix. The query needs to be at least twice the word size. Theoretically therefore, a query of four amino acid residues should be searchable, but at least five residues are recommended.[21] Figure 6.11 shows a partial screenshot of the BLAST home page. *Alternatively, the nucleotide and protein BLAST programs of NCBI automatically adjust the E-value if the query, either nucleotide or amino acid, is of length 30 or less.*

[f]Because sequence-similarity searching aims to detect sequences that indicate structural and/or functional similarity, a sequence filter is used to remove low-complexity regions during similarity searching. Examples of low-complexity regions are repeat sequences (e.g. polyA tails, nucleotide sequences like AAAATTAAAAAT, proline-rich regions, amino-acid sequences like GGGGKDKKKKDD), compositionally biased sequences etc. that are naturally abundant in most sequences. If low-complexity regions are not removed, then the sequence alignment may produce artificially high scores that would not be a true reflection of homology. Blastn filters low-complexity nucleotide sequences with the DUST algorithm, and blastp filters low-complexity amino-acid sequences with the SEG or XNU algorithms. Low-complexity nucleotide sequence is substituted by "N" (e.g. NNNNNNN), whereas low-complexity amino-acid sequence is substituted by "X" (e.g. XXXXXXX), and removed from the search.

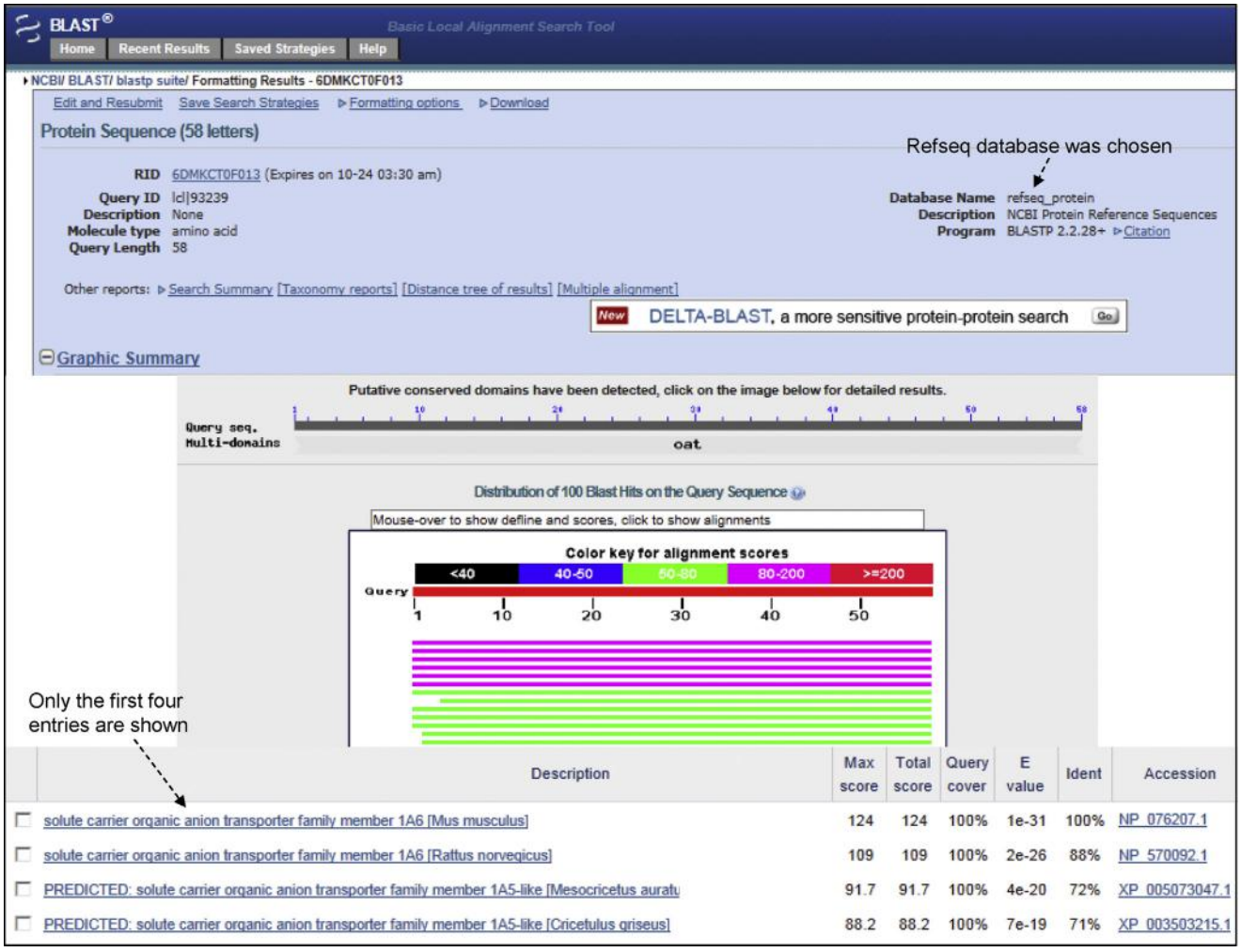

FIGURE 6.12 **Result of the BLAST analysis of Slco1a6.** The screenshot was captured in three different pieces (the upper, middle and lower segments), which are put together in the figure. A 58-amino-acid segment was used for BLAST (blastp). The RefSeq protein database was chosen to minimize the number of redundant hits. Alternatively, the Swiss-Prot could be chosen to obtain non-repetitive specific hits. The result shows on the top that putative conserved domains have been detected. These are the Kazal domain and the MFS domain. Refer to Chapter 8 for a more detailed discussion on this topic. From the analysis, only the first four entries are shown. From the BLAST hit diagram, a specific line can be clicked to get to the alignment. The color key for alignment score is self explanatory.

6.8.2.3 *Suggested BLAST* E-*Value Cut-Off*

For nucleic-acid-based search, the suggested threshold (minimum significant hit) for the *E*-value is ≤ 1e-6 ($=10^{-6}$), and a sequence identity of ≥ 70%. For protein-based search, the suggested threshold for the *E*-value ≤ 1e-4 ($=10^{-4}$), with a sequence identity of ≥ 35%[g]. However, typically for protein-based homology search, the threshold used is $E \leq$ 1e-5 ($=10^{-5}$), and the lower it is, the better. For example, an *E*-value of 1e-25 ($=10^{-25}$) will indicate a clear homology.

It should be borne in mind that the E-value is influenced by the query length. A moderately good alignment involving two very long sequences will produce a higher E-value than an extremely good alignment involving two smaller sequences.

6.8.3 Typical Basic BLAST Output

Figure 6.12 shows the result of a BLAST search. A 58-amino-acid segment was searched in the NCBI database using BLAST. In order to tailor the search to

[g]It has been reported that protein pairs with similar structure and function are likely to have > 35% sequence identity[22]. The author analyzed more than a million sequence alignments between protein pairs of known structures and noted that sequence alignments could unambiguously distinguish between protein pairs of similar and non-similar structure when the pairwise sequence identity was > 40% for long alignments. The signal, however, became blurred when the sequence identity was between 20 and 35%; this 20–35% range was termed the **twilight zone** of sequence identity.

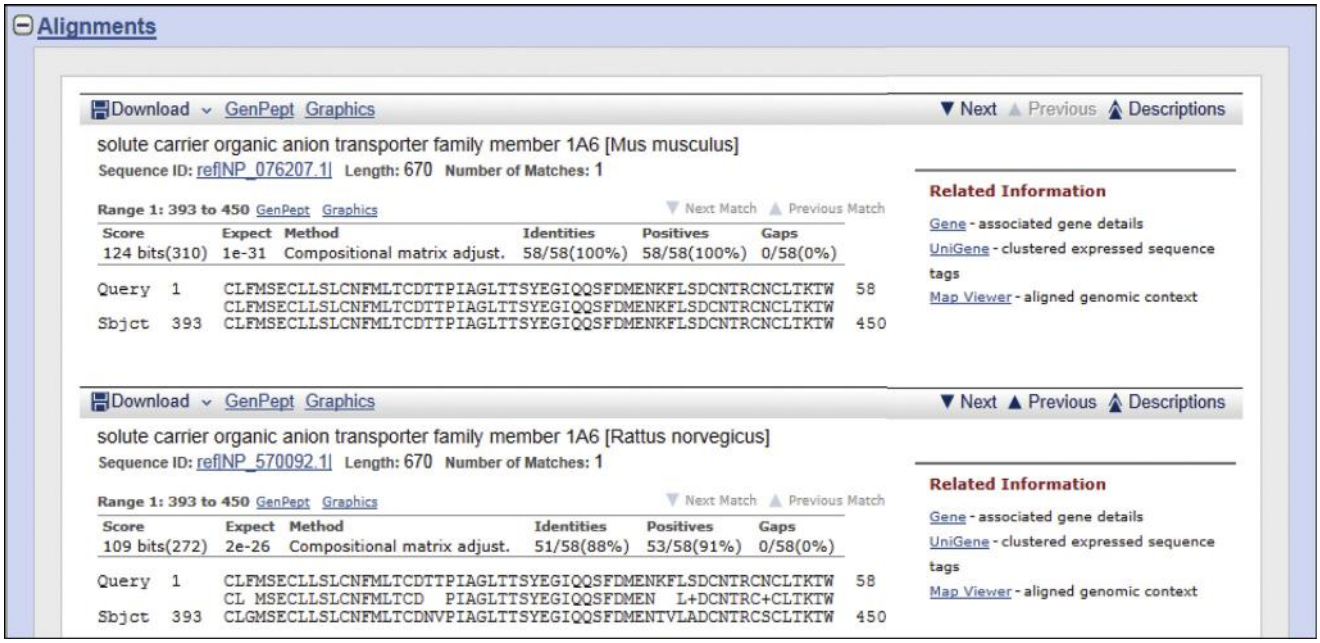

FIGURE 6.13 **The details of two alignments from Figure 6.12.** In the alignment, the upper sequence is the **query** sequence (the sequence submitted for search) and the lower sequence is the **subject** sequence (from the database); the identities and the similarities are in the middle. The number of amino acids showing identity/similarity is indicated; **identities** indicate identical amino acids between the query and subject sequences whereas **positives** indicate identical amino acids plus similar amino acids at the corresponding positions. Similar substitutions are indicated by a + sign. Each individual alignment also provides direct link to the original sequence in the database. If the subject sequence is from an organism whose whole genome is known and sequenced, the alignment also provides links to the Gene and Map Viewer databases, indicated on the right-hand side.

reduce the amount of less relevant output, the organism (*Mus musculus*) and the database (RefSeq protein database) were chosen on the BLAST home page. The search returns many entries; the highest similarity was (predictably) with mouse Slco1b2 protein (Refseq ID NP_065241). In the output, the subject sequences are listed from the highest similarity at the top to progressively lower similarities going down the list, as depicted by the bit score (score) and the *E*-value. The bit scores are listed from the highest value at the top to progressively lower values going down the list, whereas the *E*-values are listed from lowest value at the top to increasingly higher values going down the list. The detailed alignments are shown in Figure 6.13.

6.8.3.1 Searching for Distantly Related Proteins—PSI-BLAST

Many homologous proteins have similar three-dimensional structure, but in pairwise alignment they may not show significant sequence similarity. Therefore, regular protein BLAST (blastp) is not useful in identifying these proteins. Position-Specific Iterative BLAST (PSI-BLAST) is designed to detect weak relationships between the query sequence and other sequences in the database that are not necessarily detectable by standard BLAST searches. When a new genome is sequenced, PSI-BLAST can be used to identify the homology of the predicted protein products. The procedure of PSI-BLAST involves the following steps:

First step in PSI-BLAST involves standard protein–protein BLAST using the default substitution matrix, such as BLOSUM62. The input protein sequence is compared to proteins in the database to generate similarity hits. The high-scoring hits (default threshold *E*-value = < 0.005) are used to generate a multiple alignment. The original query sequence serves as the template to drive the multiple alignment. PSI-BLAST analyzes the alignments position by position and assigns a score to every position. If the amino acid residue is highly conserved at a particular position, that residue is assigned a high positive score, and others are assigned high negative scores. At weakly conserved positions, all residues receive scores near zero. Using these scores, a **profile** or **position-specific scoring matrix** (**PSSM**) is built. In the next iteration of BLAST search, this PSSM replaces the substitution matrix used in the previous iteration of BLAST search; thus more proteins are identified using this PSSM. The newly identified proteins are then incorporated in the multiple alignment to create a new PSSM, which replaces the previous one. This process is repeated (iterative) until no new

proteins are found. In each repetition, a new PSSM is generated, which replaces the old one and is used for the new round of search. The PSI-BLAST output looks like regular BLAST output.

Because of the nature of the algorithm, the main source of error in PSI-BLAST is the corruption of the profile (PSSM). In other words, for reasons unrelated to true homology/functional characteristics (e.g. amino-acid compositional bias), a position-specific amino acid may be wrongly identified as a conserved residue and assigned a high score. That position in the profile will then adversely influence the next iteration to identify more related proteins. Repeated iteration will amplify the error corrupting the subsequent profiles. There are several ways to address this problem, such as filtering out compositionally biased regions using a filtering algorithm, lowering the *E*-value from the default 0.005, or visually inspecting each output and applying judgment to discard the hits that appear spurious.

6.8.3.2 Searching for Pattern Hit—PHI-BLAST

Many proteins contain signature sequences (motifs) that are characteristics of a protein family. These signature sequences are part of important structural or functional domains. Pattern-hit-initiated (PHI)-BLAST is designed to search the database for proteins that are significantly related to the query sequence and also contain a pattern. In other words, PHI-BLAST searches for significantly similar sequences to both a query sequence and a signature. This dual requirement is supposed to reduce the number of database hits that contain the pattern but are likely to have no true homology to the query.

6.8.4 BLAT

Blast-like alignment tool (BLAT) has been discussed in the context of the University of California Santa Cruz (UCSC) Genome browser in Chapter 5. Also refer to Figure 5.32 for BLAT output. Therefore, the discussion here will be brief. BLAT is an alignment tool like BLAST, but it is structured differently. BLAT is commonly used to map the location of a query sequence in the genome, or to determine the exon structure of an mRNA. DNA BLAT works well within humans and primates, while protein BLAT works well for terrestrial vertebrates and even earlier organisms for conserved proteins.

6.8.5 FASTA

FASTA was developed for rapid biological-sequence comparison.[23] It was derived as a more sensitive and versatile program from its predecessor program FASTP, which was developed by the same authors 3 years earlier for rapid protein-sequence comparison. Like BLAST, FASTA also allows the user to compare a DNA or protein query sequence against a large database. FASTA searches for matching sequence patterns called ***k*-tuples** (***ktup***), which are akin to the "words" (W) in BLAST. The *ktup* length is usually user defined (e.g. defining *ktup* = 6 for a search involving DNA sequence will prompt the algorithm to use 6 nucleotides as the matching sequence pattern for the search). The FASTA search strategy involves searching for words of length *ktup* common to the query and target sequences. Using *ktup*, FASTA builds a local alignment. Finally, FASTA scores this alignment and provides the output as a list of sequences similar to the query in descending order. *The default ktup is 2 for amino acids and 6 for nucleotides; hence, the default window size in FASTA is smaller than that in BLAST.*

TABLE 6.3 Web-Based FASTA Servers

FASTA Server	URL
GenomeNet, Japan	http://www.genome.jp/tools/fasta/
EMBL-EBI	http://www.ebi.ac.uk/Tools/sss/fasta/
University of Virginia	http://fasta.bioch.virginia.edu/fasta_www2/fasta_list2.shtml

Some web-based FASTA servers are provided in Table 6.3.

6.8.5.1 Comparison of BLAST and FASTA

BLAST and FASTA are both heuristic algorithms that perform database searches to find sequences related to a query sequence. However, there are some differences between the two:

1. BLAST begins a search by looking for matches that include exact matches and conservative substitutions; FASTA begins a search by looking at exact matches.
2. BLAST scans a larger window size than FASTA; hence, FASTA may produce better coverage for homologs.
3. BLAST may produce multiple best-scoring alignments (also called **high-scoring segment pairs** or **HSPs**) from the same sequence; FASTA returns only one alignment from one sequence.
4. BLAST automatically masks low-complexity regions; FASTA does not employ such automatic masking. Therefore, if the query sequence has non-unique segments, such as repeats, compositionally biased segments, etc., FASTA search may return alignments with artificially high scores.
5. For a given sequence search, the BLAST output is larger than that of FASTA.
6. For a given sequence search, BLAST is faster than FASTA.

6.9 SEQUENCE COMPARISON, SYNTENY, AND MOLECULAR EVOLUTION

Comparative genomics is the study of the evolutionary relationships between the genes and genomes of different species. Comparative genomic studies are helpful in elucidating the structure, function, and evolution of genomic elements and sequence features that influence various aspects of genome biology. From the macro to the micro scale, the similarity between two genomic sequences can be studied at the level of the whole genome, at the level of chromosomal segments, and also at the level of specific genomic markers. This is because the genomes of the descendants of a common ancestor are likely to preserve at least some of the same genes in the same order. A chromosomal segment that has been inherited from the common ancestor during evolution without a major rearrangement of the order of genes is called a **syntenic block** (or **synteny block**). Syntenic blocks contain specific non-repetitive genomic markers that are in the same order and orientation in the genomes being compared. These genomic markers could be protein-coding genes, RNA-coding genes, noncoding sequences, pseudogenes, etc., and are called **syntenic anchors** (or **synteny anchors**).[24] In other words, syntenic blocks are composed of syntenic anchors present in consecutive order. Genes within a syntenic block are likely to be orthologous. While comparing two genomes, the overall sequence similarity can be enhanced if the genomes are segmented into syntenic blocks. For example, approximately 40% of the human genome can be aligned with the mouse genome, but over 90% of mouse and human genomes can be segmented into blocks of conserved synteny. Comparison of mouse chromosome 16 with the human genome shows regions of conserved synteny with human chromosomes 3, 8, 12, 16, 21, and 22. A total of 11,822 syntenic anchors map to chromosome 16; the mean length and identity of these anchors are 198 bp and 88.1%, respectively. Over 50% of these anchors are in runs of at least 128 in a row in the same order and orientation between mouse chromosome 16 and the human chromosomes sharing blocks of conserved synteny.[24] Charting the blocks of conserved synteny creates a **synteny map**, which shows the large-scale evolutionary relationships between genomes that are related through a common ancestor, but have diverged during evolution. Shared genomic synteny and shared protein functions can be used to enhance the identification of orthologous gene pairs.[25]

References

1. Nei M, et al. *Proc. Natl. Acad Sci USA* 2001;**98**:2497–502.
2. Koonin EV, Galperin MY. *Sequence-evolution-function: computational approaches in comparative genomics*. Boston, MA: Kluwer; 2003. p. 25–49
3. Raghava GPS, Barton GJ. *BMC Bioinformatics* 2006;**7**:415.
4. Choudhuri S, et al. *Biochem Biophys Res Commun* 2000;**274**:79–86.
5. Wu CH, et al. *Nucl. Acids Res* 2002;**30**:35–7.
6. Huang X, Miller W. *Adv Appl Math* 1991;**12**:337–57.
7. Papadopoulos JS, Agarwala R. *Bioinformatics* 2007;**23**:1073–9.
8. Larkin MA, et al. *Bioinformatics* 2007;**23**:2947–8.
9. Thompson JD, et al. *Nucl Acids Res* 1994;**22**:4673–80.
10. Katoh K, et al. *Nucl Acids Res* 2002;**30**:3059–66.
11. Edgar RC. *Nucl Acids Res* 2004;**32**:1792–7.
12. Notredame C, et al. *J Mol Biol* 2000;**302**:205–17.
13. Jones DT, et al. *Comput Appl Biosci* 1992;**8**:275–82.
14. Henikoff S, Henikoff JG. *Proc Natl Acad Sci USA* 1992;**89**:10915–9.
15. Pietrokovski S, et al. *Nucl Acids Res* 1996;**24**:197–200.
16. Eddy SR. *Nat Biotechnol* 2004;**22**:1035–6.
17. Kim D, et al. *Protein Eng* 2003;**16**:641–50.
18. Pearson WR. *Meth Enzymol* 1996;**266**:227–58.
19. Karlin S, Altschul SF. *Proc Natl Acad Sci USA* 1990;**87**:2264–8.
20. Altschul SF, et al. *J Mol Biol* 1990;**215**:403–10.
21. NCBI. *NCBI BLAST Help*. Available online at: <http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD = Web&PAGE_TYPE = BlastDocs&DOC_TYPE = ProgSelectionGuide>; 2013.
22. Rost B. *Protein Eng* 1999;**12**:85–94.
23. Pearson WR, Lipman DJ. *Proc Natl Acad Sci USA* 1988;**85**:2444–8.
24. Mural, et al. *Science* 2002;**296**:1661–71.
25. Zheng, et al. *Bioinformatics* 2005;**21**:703–10.