The local alignment algorithm has identified exactly the subsequence match that we identified from our previous semiglobal alignment. When working with long sequences of many thousands, or even millions, of nucleotides, local alignment methods can identify subsequence matches that would be impossible to find using global or semiglobal alignments.

Database Searches

While sequence alignments can be an invaluable tool for comparing two known sequences, a far more common use of alignments is to search through a database of many sequences to retrieve those that are similar to a particular sequence. If, for example, we had identified a region of the human genome that we believe is a previously unidentified gene, we might compare our putative gene with the millions of other sequences in the GenBank database at the National Center for Biological Information (NCBI). The search results, consisting of other sequences that align well with (and thus are similar to) our sequence, might give us an indication of the functional role of our newfound gene along with valuable clues regarding its regulation and expression and its relationship to similar genes in humans and other species.

In performing database searches, the size and sheer number of sequences to be searched (at the time of the writing of this text, there were more than 13 million sequences in GenBank) often precludes the obvious and direct approach of aligning a query sequence with each sequence in the database and returning the sequences with the highest alignment scores. Instead, various indexing schemes and heuristics must be used to speed the search process. Many of the commonly used database search algorithms are not guaranteed to produce the best match from the database, but rather have a high probability of returning most of the sequences that align well with the query sequence. Nevertheless, the efficiency of these tools in finding sequences similar to a query sequence from the vast repositories of available sequence data has made them invaluable tools in the study of molecular biology.

BLAST and Its Relatives

One of the most well known and commonly used tools for searching sequence databases is the BLAST algorithm, introduced by S. Altschul *et al.* in the early 1990s. The original BLAST algorithm searches a sequence database for maximal ungapped local alignments. In other words, BLAST finds subsequences from the database that are similar to subsequences in the query sequence. Several variations of the BLAST algorithm are available for searching protein or nucleotide sequence databases using protein or nucleotide query sequences. To illustrate the basic concepts of BLAST searches, we will discuss the BLASTP algorithm, which searches for protein sequence matches using PAM or BLOSUM matrices to score the ungapped alignments.

To search a large database efficiently, BLASTP first breaks down the query sequence into **words**, or subsequences of a fixed length (4 is the default word length). All possible words in the query sequence are calculated by sliding a window equal in size to the word length over the query sequence. For example, a protein query sequence of AILVPTV would produce four different words: AILV (4 characters, starting with the first character), ILVP (starting with the second character), LVPT, and VPTV. Once all of the words in the query sequence have been determined, words composed mostly of common amino acids will be discarded. The sequences in the database are then searched for occurrences of the search words. Each time a word match is found in the database, the match is extended in both directions from the matching word until the alignment score falls below a given threshold. Since the alignment is ungapped, the extension only involves adding additional residues to the matching region and recalculating the score according to the scoring matrix. The choice of the threshold value for continuing the extension is an important search parameter, because it determines how likely the resulting sequences are to be biologically relevant homologs of the query sequence. Figure 2.11 shows a simplified overview of the BLASTP search process for a simple polypeptide sequence.

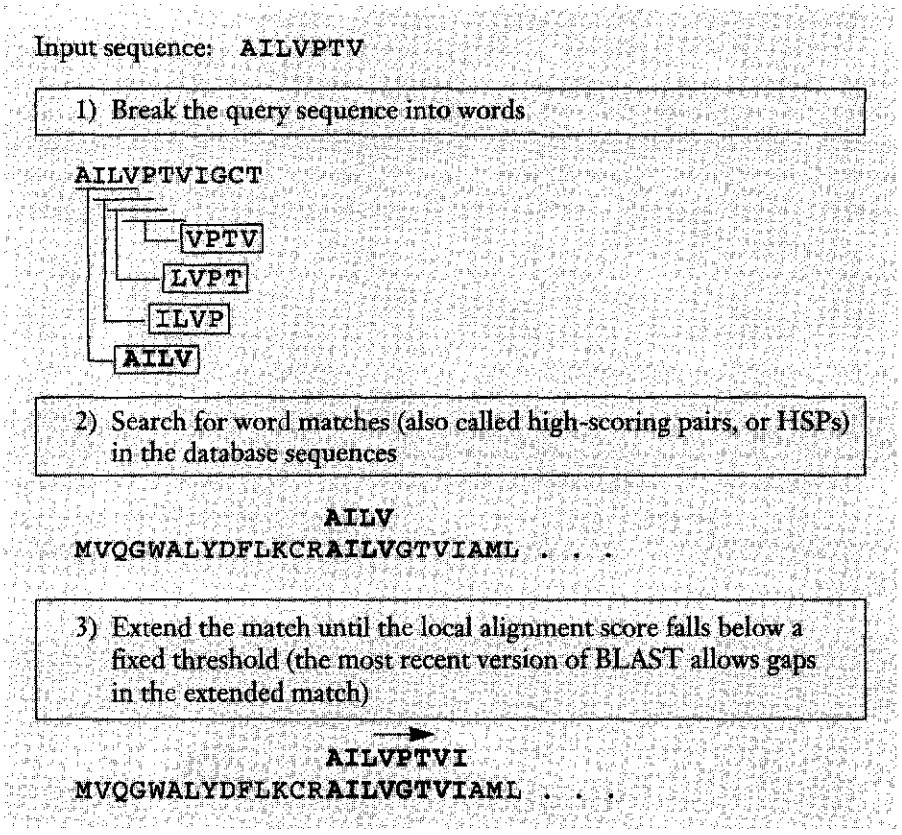

FIGURE 2.11 *Overview of the BLASTP search process.*

Numerous sequence alignment and database search algorithms have been developed for various specific types of sequence searches. As mentioned previously, BLASTP searches protein sequence databases for polypeptide sequences. Other variations of BLAST, including BLASTN and BLASTX, allow searching of nucleotide sequence databases and translating from nucleotide sequences to protein sequences prior to searching, respectively. BLAST 2.0, the most recent version of BLAST, inserts gaps to optimize the alignment. PSI-BLAST, another member of the BLAST family, summarizes results of sequence searches into **position-specific scoring matrices**, which are useful for protein modeling and structure prediction.

FASTA and Related Algorithms

The FASTX algorithms are another commonly used family of alignment and search tools. FASTA and its relatives perform gapped local alignments between sequences. Since FASTX searches perform several detailed comparisons between the query sequence and each sequence in the database, FASTX searches generally require significantly more execution time than the BLAST searches. However, the FASTX algorithms are considered by some to be more sensitive than BLAST, particularly when the query sequence is repetitive.

As with BLAST searches, a FASTA search begins by breaking the search sequence into words. For genomic sequences a word size of 4 to 6 nucleotides is generally used, while 1 to 2 residues are generally used for polypeptides. Next, a table is constructed for the query sequence showing the locations of each word within the sequence. For example, consider the amino acid sequence `FAMLGFIKYLPGCM`. For a word size of 1, the following table would be constructed:

Word	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
Pos.	2	13			1	5		7	8	4	3		11							9
					6	12				10	14									

In this table, the column for phenylalanine (F) contains entries 1 and 6 because F occurs in the first and sixth positions of the query sequence.

To compare this sequence to a target sequence, we construct a second table that compares the amino acid positions in the target sequence with the query sequence. For the target sequence `TGFIKYLPGACT` this table would appear as follows:

1	2	3	4	5	6	7	8	9	10	11	12
T	G	F	I	K	Y	L	P	G	A	C	T
	3	-2	3	3	3	-3	3	-4	-8	2	
	10	3				3		3			

Consider position 2, a glycine (G) residue. Looking at the table for the query sequence, we can quickly see that glycines are present in positions 5 and 12 of the

query sequence. The distance between 5 and 12 and the position of the first glycine in the target sequence (position 2) produces the two entries 3 and 10. For the second glycine, in position 9, we likewise subtract 9 from 5 and 12, obtaining entries –4 and 3. Amino acids, such as threonine (T), that are not found in the query sequence are not included in this table.

Note the large number of instances of the distance 3 in the second table. This suggests that by offsetting the target sequence by 3, we might obtain a reasonable alignment between the two sequences. In fact, we would obtain the following:

```
FAMLGFIKYLPGCM
   |||||||
  TGFIKYLPGACT
```

By comparing the offset tables for two sequences, areas of identity can be found quickly. Once these areas are found, they are joined to form larger sequences, which are then aligned using a full Smith-Waterman alignment. However, because the alignment is constrained to a known region of similar sequence, FASTA is much faster than performing a complete dynamic programming alignment between the query sequence and all possible targets.

Alignment Scores and Statistical Significance of Database Searches

While a database search will always produce a result, the sequences found cannot be assumed to be related to the search sequence without more information. The primary indicator of how similar the search results are to a query sequence is the alignment score. Alignment scores, however, vary among the different database search algorithms, and are not, of themselves, a sufficient indicator that two sequences are related. Given a database search result with an alignment score S, an appropriate question to ask is "Given a set of sequences *not related* to the query sequence (or even random sequences), what is the probability of finding a match with alignment score S simply by chance?" To answer this question, database search engines generally provide a P score or an E score along with each search result. While they answer slightly different questions, the two scores are closely related, and often have very similar values. Given a database result with an alignment score S, the E score is the expected number of sequences of score $> = S$ that would be found by random chance. The P score is the probability that *one or more* sequences of score $> = S$ would have been found randomly. Low values of E and P indicate that the search result was unlikely to have been obtained by random chance, and thus is likely to bear an evolutionary relationship to the query sequence. While E values of 10^{-3} and below are often considered indicative of statistically significant results, it is not uncommon for search algorithms to produce matches with E values on the order of 10^{-50}, indicating a very strong likelihood of evolutionary relationship between the query sequence and the search results.