Using Oracle Data Mining to Analyze Data

Purpose

This module shows you how to use Basic Local Alignment Search Tool (BLAST) to perform bioinformatics tasks.

Topics

This module will discuss the following topics:

Place the cursor on this icon to display all screenshots. You can also place the cursor on each icon to see only the screenshot associated with it.

Overview

Back to List

What is BLAST?

Sequence alignment is without doubt one of the most commonly performed bioinformatics tasks. The most widely used sequence alignment algorithm is BLAST (Basic Local Alignment Search Tool). BLAST is a method for rapid sequence comparison introduced in 1990 by Stephen Altschul. It is typically used to compare a query nucleotide or amino acid sequence against a database of sequences. Its success comes from its combination of speed, sensitivity and statistical assessment of the results.

BLAST is a heuristic method to find the high scoring locally optimal alignments between a query sequence and a database. The BLAST algorithm and family of programs rely on the statistics of gapped and un-gapped sequence alignments. The statistics allow the probability of obtaining an alignment with a particular score to be estimated.

Over the last few years, as more sequence data has become available as a result of large scale sequencing efforts, BLAST has started to be used for an increasing range of sequence alignment activities which include functional annotation, gene discovery, across-organism and across-species analysis, genome assembly and completion.

Oracle10 *g* BLAST Functions

A version of BLAST, which is very similar to NCBI BLAST 2.0, has been implemented in the database using Table Functions. This enables users to perform BLAST queries against data that is held directly inside an Oracle database. As the algorithms are implemented as table functions, parallel computation is intrinsically supported.

The five core variants of NCBI BLAST have been implemented:

- BLASTN compares a nucleotide query sequence against a nucleotide database.
- BLASTN compares an amino acid query sequence against a protein sequence database.

- BLASTX compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.
- TBLASTN compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands).
- TBLASTX compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database.

Two BLAST Table Functions have been implemented:

MATCH	Returns the following: - **q_seq_id** : identifier of the query sequence - **t_seq_id** : identifier of the matched (target) sequence (for example: the NCBI accession number) - **score** : score of the match - **value** : the expected value
ALIGN	Returns information about the alignment including the following: - **q_seq_id** : identifier of the query sequence - **t_seq_id** : identifier of the matched (target) sequence (for example: the NCBI accession number) - **pct_identity** : percentage of the query sequence that identically matches with the database sequence - **alignment_length** : the length of the alignment - **mismatches** : the number of base-pair mismatches between the query and database sequences - **gap_openings** : the number of gaps opened in gapped alignment - **gap_list** : the list of offsets where a gap is opened - **q_start** : the position where the gap alignment starts - **q_end** : the position where the gap alignment ends - **s_start** : the position where the database sequence alignment starts - **s_end** : the position where the database sequence alignment ends - **expect** : the expect value of the alignment - **score** : the score of the alignment

BLAST queries can be invoked directly using the SQL interface or through an application. The results of queries can also either be displayed using the SQL interface or through an application. When the SQL interface is used, the user has the flexibility to decide how the results will be displayed.

The introduction of similarity search functionality in the database means that users no longer have to export sequence data and transform it into a BLAST data set prior to doing similarity searches.

Complex BLAST Queries

As BLAST is implemented in the database, and can be invoked by SQL, it is possible to pre-process any queries as well as perform

any required post-processing. The ability to preprocess data, so that you are only performing the BLAST search on a subset of the data, means that queries should be highly performant. The post-processing capability means that it is now possible to integrate the output of a sequence similarity search with the data mining, text mining, and other analytical features of the database.

Prerequisites

Back to List

Before starting this module, you should have:

- ☐ **1.** Completed the Configuring Linux for the Installation of Oracle Database 10g lesson
- ☐ **2.** Completed the Installing the Oracle Database 10g on Linux lesson
- ☐ **3.** Completed the Postinstallation Tasks lesson. Note: The demo data needed for this lesson is installed in the Postinstallation Tasks lesson.
- ☐ **4.** Download and unzip blast.zip into your working directory (i.e. /home/oracle/wkdir)

Query Nucleotide Data

Back to Topic List

You will perform a **BLASTN** query against a human DNA database using the MATCH and ALIGN functions. Perform the following:

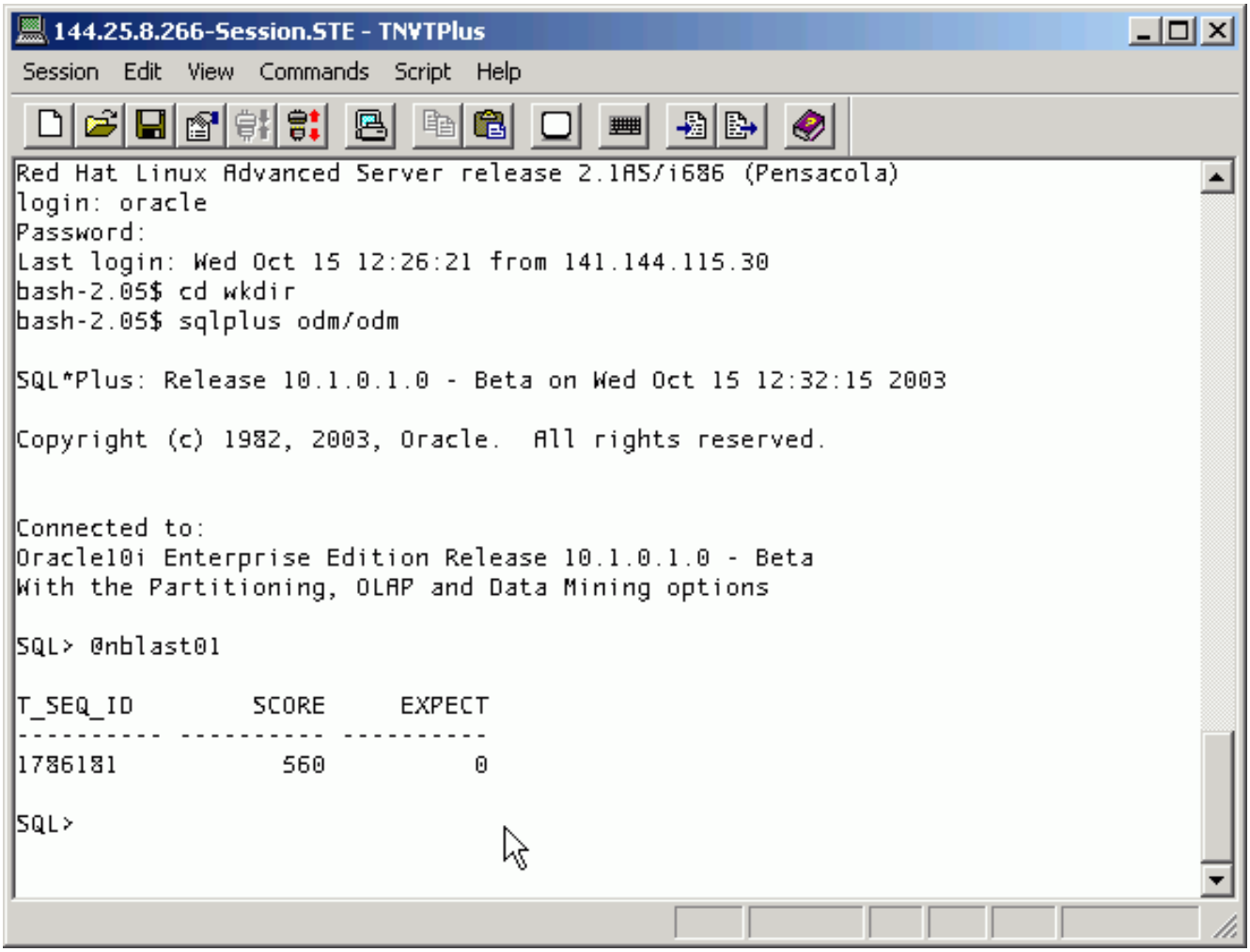
- ☐ **1.** You will perform a BLAST search of the given query sequence against a human DNA database and return the seq_id, score, and expect value of matches that score > 25. Open a terminal window and execute the following commands:

```
cd wkdir
sqlplus odm/odm
@nblast01
```

The query in the **`nblast01.sql`** script is as follows:

```
select *
 from TABLE(BLASTN_MATCH (
    (select sequence from ecoli_query), -- query_sequence
     CURSOR(SELECT seq_id, seq_data FROM ecoli10), -- seqdb_cursor
       1, -- subsequence_from
      -1, -- subsequence_to
       0, -- FILTER_LOW_COMPLEXITY
```

```
        0,  -- MASK_LOWER_CASE
       10,  -- EXPECT_VALUE
        0,  -- OPEN_GAP_COST
        0,  -- EXTEND_GAP_COST
        0,  -- MISMATCH_COST
        0,  -- MATCH_REWARD
       11,  -- WORD_SIZE
        0,  -- X_DROPOFF
        0 -- FINAL_X_DROPOFF))
 t where t.score > 25;
```



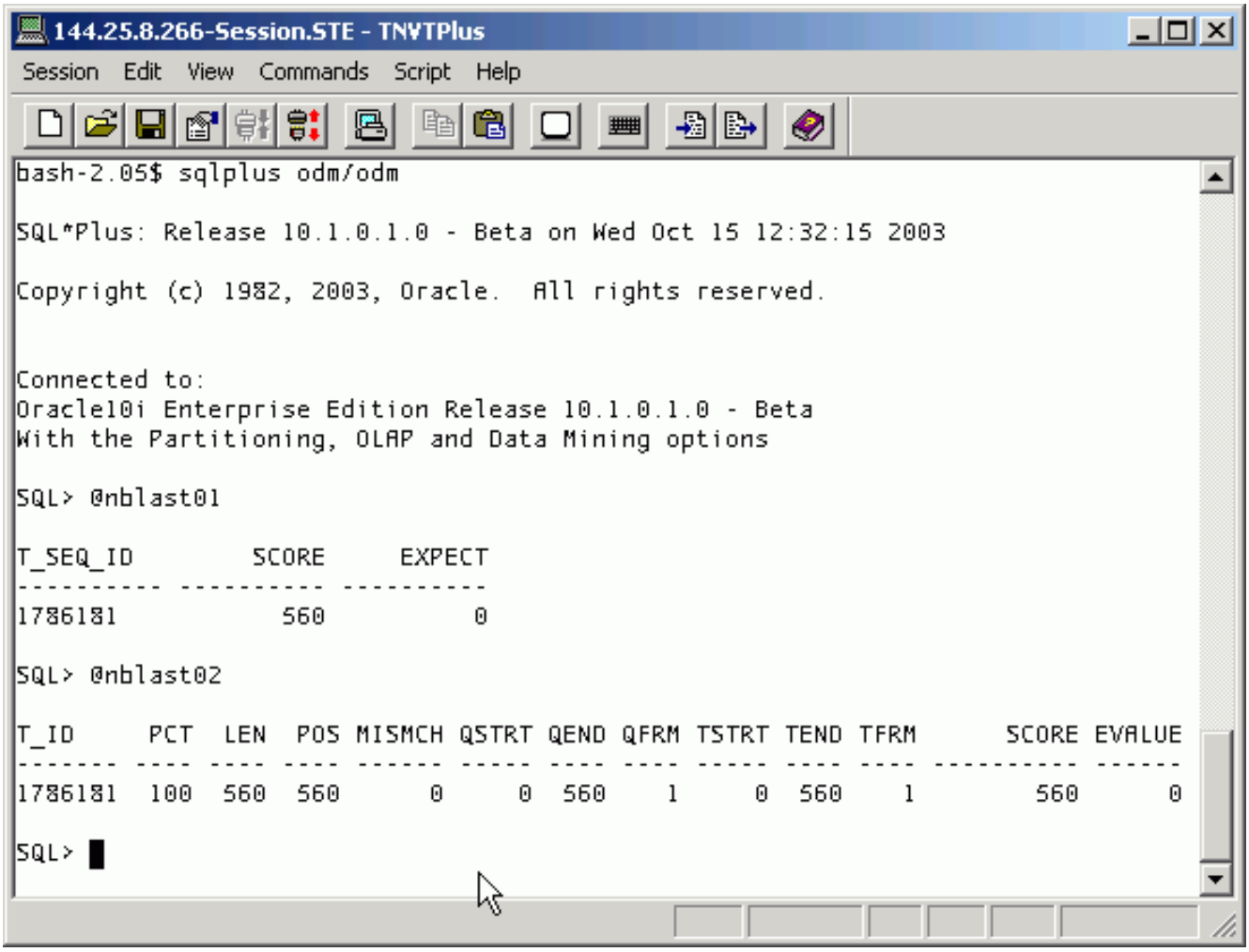
The output displays the sequence ID, expect value and score for all nucleotide sequences in the ecoli10 database that have a similarity score of greate than 25 compared to the input nucleotide query sequence.

☐ **2.** Now perform a BLAST search of the given query sequence against a human DNA database and return the full alignment information of matches that have an expect value score > 25. From your terminal window, execute the following commands:

```
@nblast02
```

The query in the **nblast02.sql** script is as follows:

```
select T_SEQ_ID, PCT_IDENTITY, ALIGNMENT_LENGTH,
  POSITIVES, MISMATCHES, Q_SEQ_START, Q_SEQ_END,
  Q_FRAME, T_SEQ_START, T_SEQ_END, T_FRAME,
  score, EXPECT
 from TABLE(BLASTN_ALIGN (
    (select sequence from ecoli_query), -- query_sequence
     CURSOR(SELECT seq_id, seq_data FROM ecoli10), -- seqdb_cursor
       1, -- subsequence_from
      -1, -- subsequence_to
       0, -- FILTER_LOW_COMPLEXITY
       0, -- MASK_LOWER_CASE
      10, -- EXPECT_VALUE
       0, -- OPEN_GAP_COST
       0, -- EXTEND_GAP_COST
       0, -- MISMATCH_COST
       0, -- MATCH_REWARD
      11, -- WORD_SIZE
       0, -- X_DROPOFF
       0 -- FINAL_X_DROPOFF))
 t where t.score > 25;
```



The results show the sequence ID, expect value, score and full alignment information for all nucleotide sequences in the ecoli10 database that have a sequence similarity score of greater than 25 compared to the input nucleotide query sequence.

Query Amino Acid Data

Back to Topic List

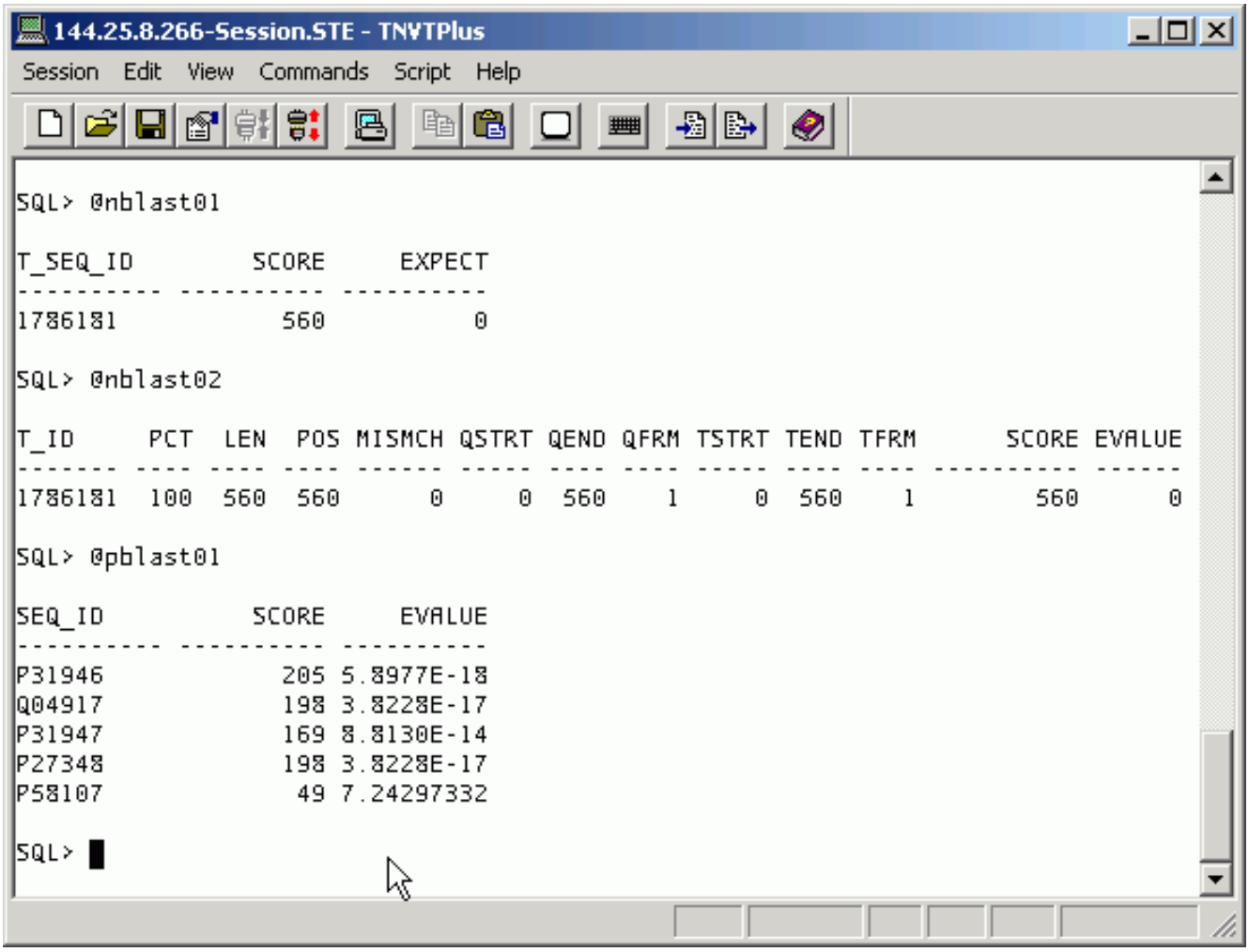
You will perform a **BLASTP** query against all human proteins in SwissProt using the MATCH and ALIGN functions. Perform the following:

☐ **1.** Perform a BLAST search of the given query sequence against all human proteins in SwissProt and return the seq_id, score, and expect value of matches that score > 25. From your terminal window, execute the following commands:

@pblast01

The query in the **pblast01.sql** script is as follows:

```
Select T_SEQ_ID, score, EXPECT as evalue

 from TABLE(BLASTP_MATCH (

    (select sequence from query_db), -- query_sequence

     CURSOR(SELECT seq_id, seq_data

      FROM swissprot

      WHERE organism = 'Homo sapiens (Human)'), -- seqdb_cursor

       1, -- subsequence_from

      -1, -- subsequence_to

       0, -- FILTER_LOW_COMPLEXITY

       0, -- MASK_LOWER_CASE
 'BLOSUM62', -- SUB_MATRIX

      10, -- EXPECT_VALUE

       0, -- OPEN_GAP_COST

       0, -- EXTEND_GAP_COST

      11, -- WORD_SIZE

       0, -- X_DROPOFF

       0 -- FINAL_X_DROPOFF))

 t where t.score > 25;
```



The output displays the sequence ID, expect value and score for all human amino acid sequences in the Swiss-Prot database that have a similarity score of greater than 25 compared to the input amino acid query sequence.

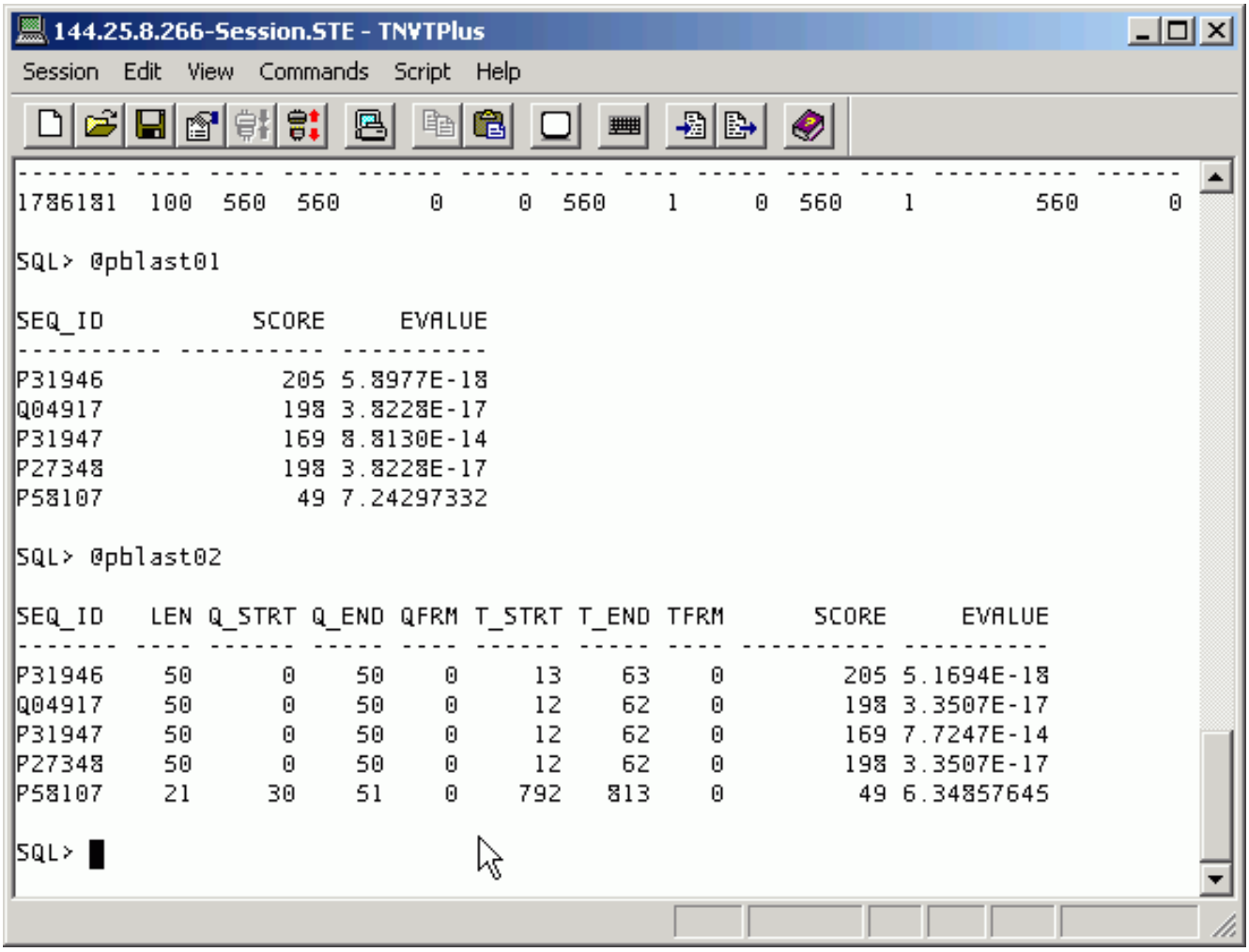
☐ **2.** Perform a BLAST search of the given query sequence against all of the human proteins in SwissProt created after 01-Jan-90 and return the full alignment information of matches with an expect value score > 25. From your terminal window, execute the following commands:

```
@pblast02
```

The query in the **`pblast02.sql`** script is as follows:

```
Select T_SEQ_ID, ALIGNMENT_LENGTH,
  Q_SEQ_START, Q_SEQ_END, Q_FRAME, T_SEQ_START,
  T_SEQ_END, T_FRAME, score, EXPECT as evalue
 from TABLE(BLASTP_ALIGN (
```

```
    (select sequence from query_db), -- query_sequence

     CURSOR(SELECT seq_id, seq_data

      FROM swissprot

      WHERE organism = 'Homo Sapiens (Human)' AND

            creation_date > '01-Jan-90'), -- seqdb_cursor

       1, -- subsequence_from

      -1, -- subsequence_to

       0, -- FILTER_LOW_COMPLEXITY

       0, -- MASK_LOWER_CASE
 'BLOSUM62', -- SUB_MATRIX

      10, -- EXPECT_VALUE

       0, -- OPEN_GAP_COST

       0, -- EXTEND_GAP_COST

       3, -- WORD_SIZE

       0, -- X_DROPOFF

       0 -- FINAL_X_DROPOFF))

t where t.score > 25;
```



The results show the sequence ID, expect value, score and full alignment information for all human amino acid sequences in the Swiss-Prot database that have a sequence similarity score of greater than 25 compared to the input amino acid query sequence.

Query Translated Data

Back to Topic List

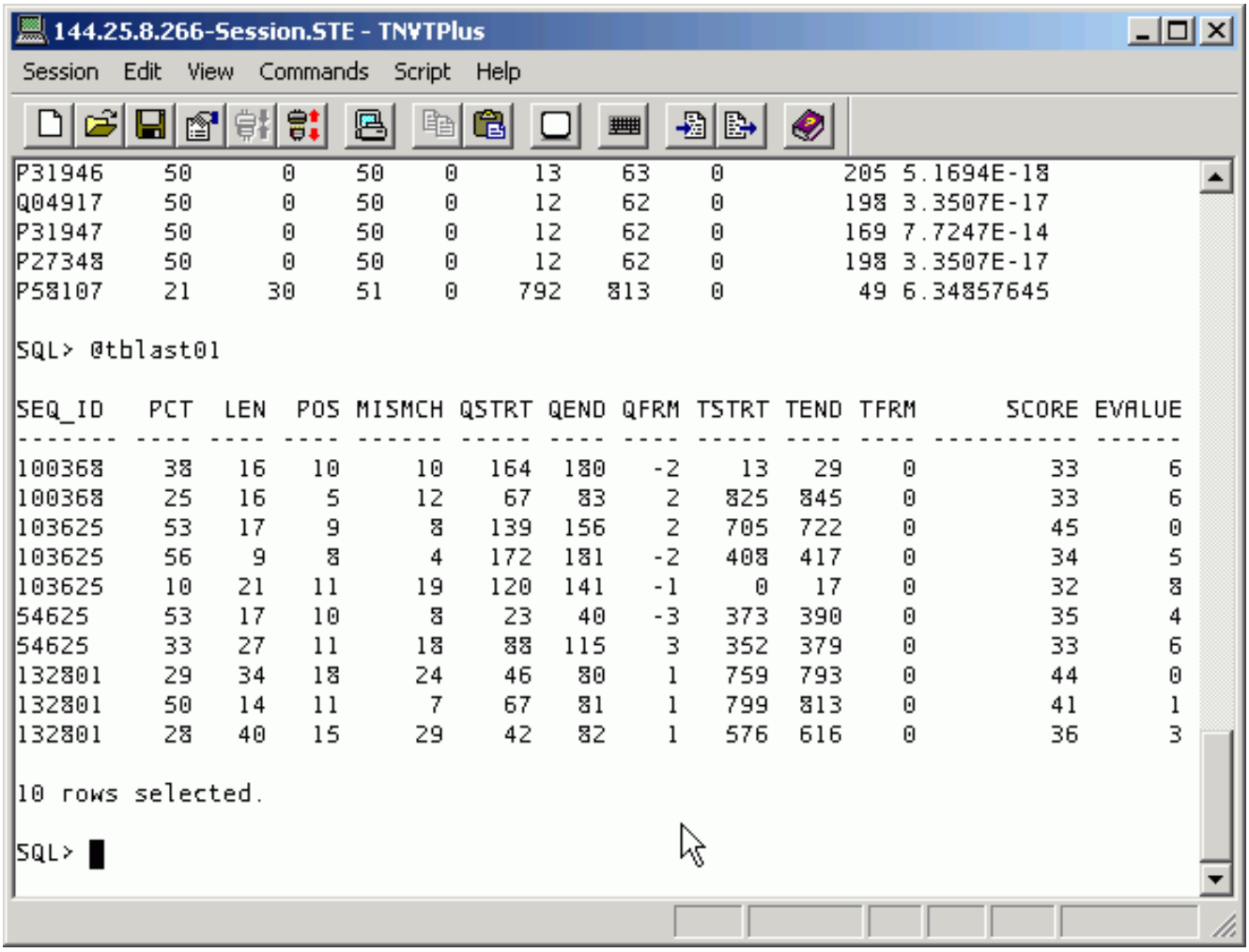
You will perform **TBLAST** queries using the ALIGN function. Perform the following:

☐ **1.** Perform a BLAST search of the given nucleotide query sequence against an amino acid database and return the full alignment information of matches with expect values that score > 25. From your terminal window, execute the following commands:

@tblast01

The query in the **tblast01.sql** script is as follows:

```
Select T_SEQ_ID, PCT_IDENTITY, ALIGNMENT_LENGTH,
  POSITIVES, MISMATCHES, Q_SEQ_START, Q_SEQ_END,
  Q_FRAME, T_SEQ_START, T_SEQ_END, T_FRAME,
  score, EXPECT
 from TABLE(TBLAST_ALIGN (
    (select sequence from ecoli_query), -- query_sequence
     CURSOR(SELECT seq_id, seq_data FROM prot_db), -- seqdb_cursor
          1, -- subsequence_from
         -1, -- subsequence_to
  'blastx', -- TRANSLATION_TYPE
          1, -- GENETIC_CODE
          0, -- FILTER_LOW_COMPLEXITY
          0, -- MASK_LOWER_CASE
  'BLOSUM62', -- SUB_MATRIX
        10, -- EXPECT_VALUE
          0, -- OPEN_GAP_COST
          0, -- EXTEND_GAP_COST
          3, -- WORD_SIZE
          0, -- X_DROPOFF
          0 -- FINAL_X_DROPOFF))
 t where t.score > 25;
```



The results show the sequence ID, expect value, score and full alignment information for all amino acid sequences in the protdb database that have a sequence similarity score of greater than 25 compared to the input nucleotide query sequence.

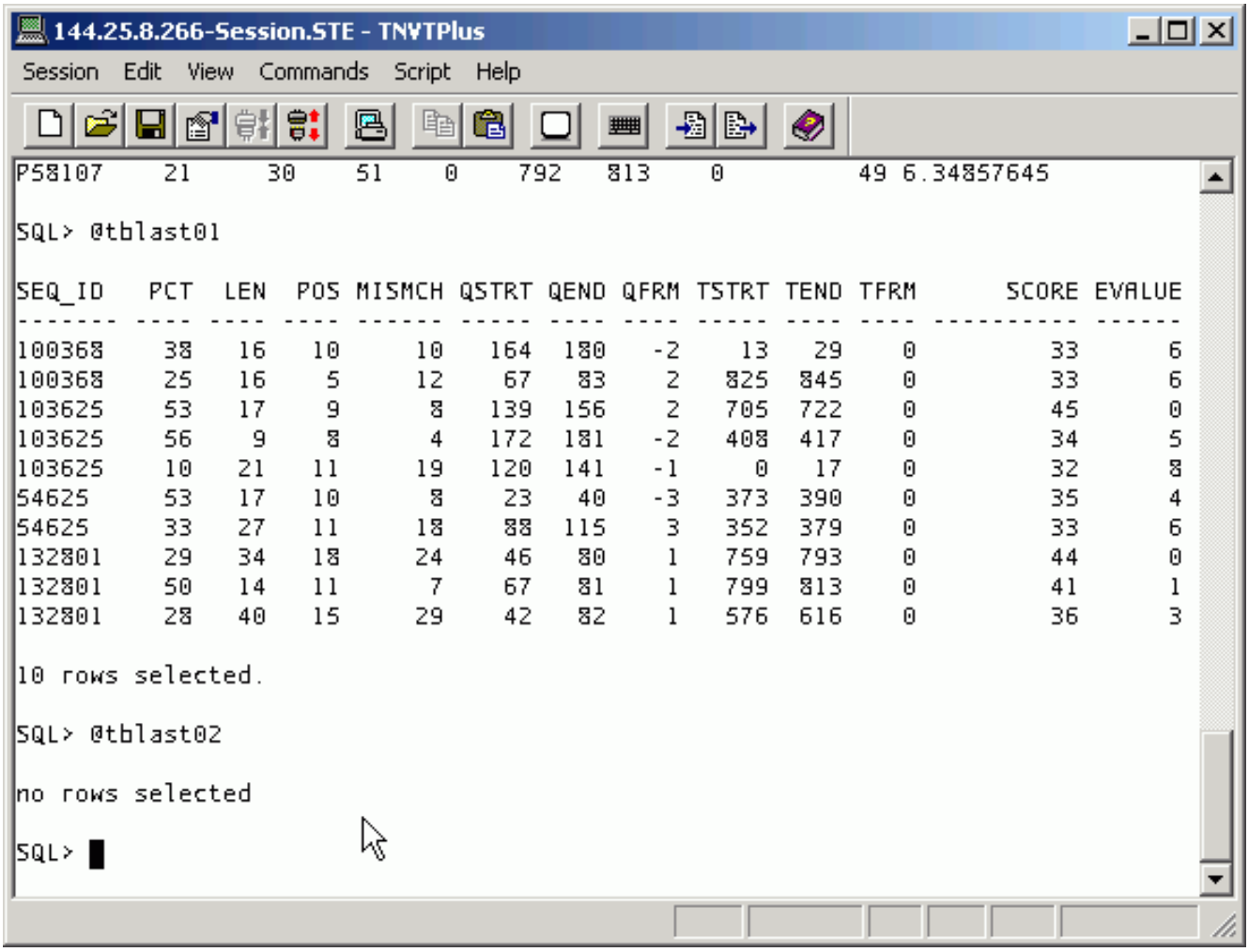
☐ **2.** Perform a BLAST search of an amino acid query sequence against a translated nucleotide database and return the full alignment information of matches with an expect value > 25. From your terminal window, execute the following commands:

`@tblast02`

The query in the **`tblast02.sql`** script is as follows:

```
Select *
 from TABLE(TBLAST_ALIGN (
    (select sequence from query_db), -- query_sequence
     CURSOR(SELECT seq_id, seq_data FROM ecoli10), -- seqdb_cursor
```

```
           1, -- subsequence_from
          53, -- subsequence_to
    'blastn', -- TRANSLATION_TYPE
           1, -- GENETIC_CODE
           0, -- FILTER_LOW_COMPLEXITY
           0, -- MASK_LOWER_CASE
    'BLOSUM62', -- SUB_MATRIX
          10, -- EXPECT_VALUE
           0, -- OPEN_GAP_COST
           0, -- EXTEND_GAP_COST
           3, -- WORD_SIZE
           0, -- X_DROPOFF
           0 -- FINAL_X_DROPOFF))
  t where t.score > 25;
```



The results show the sequence ID, expect value, score and full alignment information for all nucleotide sequences in the ecoli10 database that have a sequence similarity score of greater than 25 compared to the input amino acid sequence.

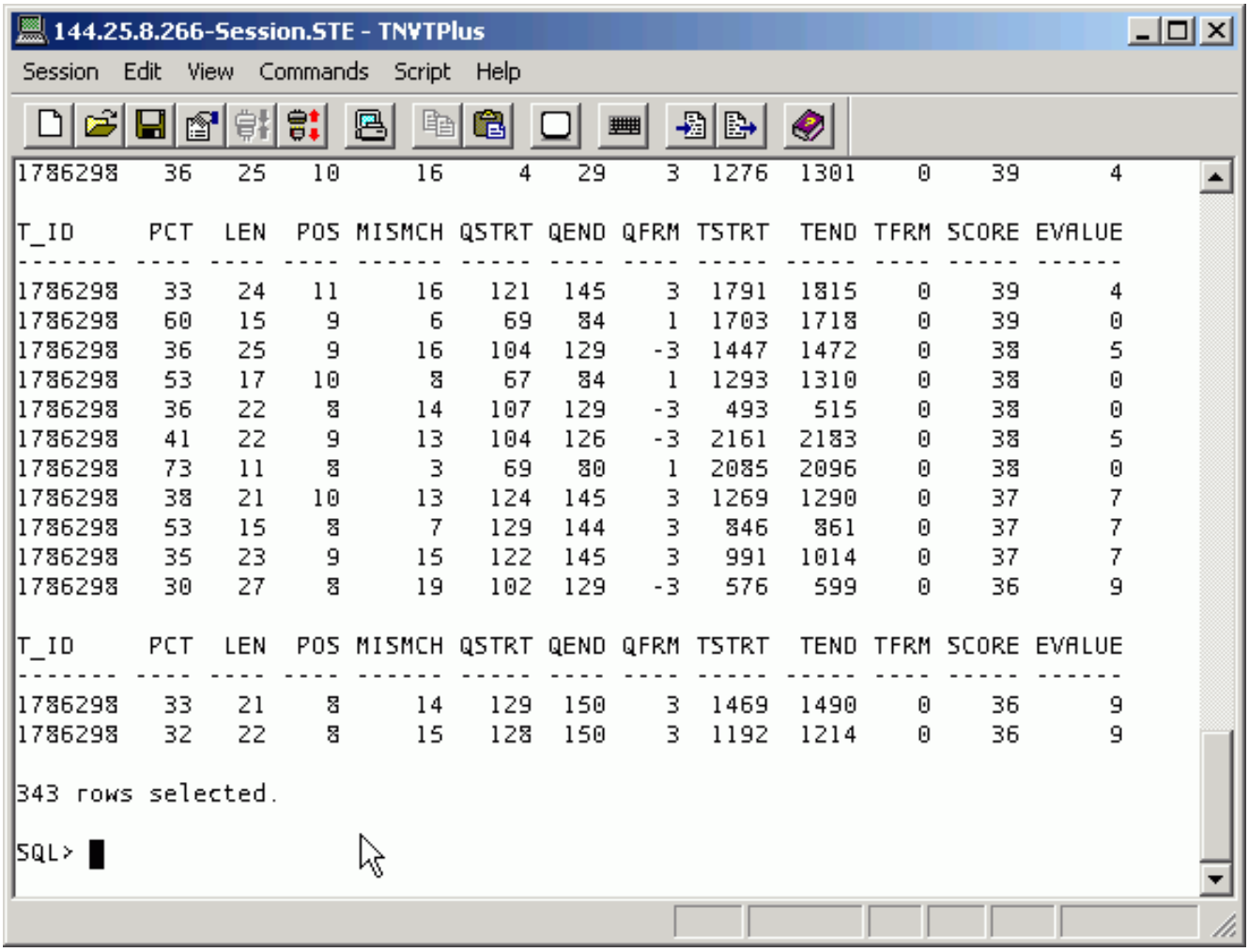
☐ **3.** Perform a BLAST search of the given nucleotide query sequence against a nucleotide database and return the full amino acid alignment information of matches with expect values that score > 25. From your terminal window, execute the following commands:

```
@tblast03
```

The query in the **`tblast03.sql`** script is as follows:

```
select T_SEQ_ID, PCT_IDENTITY, ALIGNMENT_LENGTH,
  POSITIVES, MISMATCHES, Q_SEQ_START, Q_SEQ_END,
  Q_FRAME, T_SEQ_START, T_SEQ_END, T_FRAME,
  score, EXPECT
```

```
from TABLE(TBLAST_ALIGN (
   (select sequence from ecoli_query), -- query_sequence
    CURSOR(SELECT seq_id, seq_data FROM ecoli10), -- seqdb_cursor
        1, -- subsequence_from
       53, -- subsequence_to
 'blastx', -- TRANSLATION_TYPE
        1, -- GENETIC_CODE
        0, -- FILTER_LOW_COMPLEXITY
        0, -- MASK_LOWER_CASE
 'BLOSUM62', -- SUB_MATRIX
      10, -- EXPECT_VALUE
        0, -- OPEN_GAP_COST
        0, -- EXTEND_GAP_COST
        3, -- WORD_SIZE
        0, -- X_DROPOFF
        0 -- FINAL_X_DROPOFF))
t where t.score > 25;
```




```
1786298   36   25   10     16     4   29    3  1276  1301    0    39      4

T_ID     PCT  LEN  POS MISMCH QSTRT QEND QFRM TSTRT  TEND TFRM SCORE EVALUE
------- ---- ---- ---- ------ ----- ---- ---- ----- ----- ---- ----- ------
1786298   33   24   11     16   121  145    3  1791  1815    0    39      4
1786298   60   15    9      6    69   84    1  1703  1718    0    39      0
1786298   36   25    9     16   104  129   -3  1447  1472    0    38      5
1786298   53   17   10      8    67   84    1  1293  1310    0    38      0
1786298   36   22    8     14   107  129   -3   493   515    0    38      0
1786298   41   22    9     13   104  126   -3  2161  2183    0    38      5
1786298   73   11    8      3    69   80    1  2085  2096    0    38      0
1786298   38   21   10     13   124  145    3  1269  1290    0    37      7
1786298   53   15    8      7   129  144    3   846   861    0    37      7
1786298   35   23    9     15   122  145    3   991  1014    0    37      7
1786298   30   27    8     19   102  129   -3   576   599    0    36      9

T_ID     PCT  LEN  POS MISMCH QSTRT QEND QFRM TSTRT  TEND TFRM SCORE EVALUE
------- ---- ---- ---- ------ ----- ---- ---- ----- ----- ---- ----- ------
1786298   33   21    8     14   129  150    3  1469  1490    0    36      9
1786298   32   22    8     15   128  150    3  1192  1214    0    36      9

343 rows selected.

SQL>
```

The results show the sequence ID, expect value, score and full alignment information for all translated nucleotide sequences in the ecoli10 database that have a sequence similarity score of greater than 25 compared to the input translated nucleotide query sequence.

Place the cursor on this icon to hide all screenshots.