# [第15周--ch02 string] 高级算法大模型辅助学习


## 赵乐天 -2024244014-计算机科学与技术


---

**Q01: global alignment and local alignment**

- **prompt**:

  ```
  What are global alignment and local alignment, and how do they differ in
  application?
  ```

- **model_name**: **GPT-4-turbo**
- **model_version**: **v2**
- **response**:

  ```
  Global alignment aims to align sequences along their entire lengths, which
  is suitable for sequences of similar length with small differences, such as
  homologous genes. Needleman-Wunsch is a commonly used algorithm for this
  purpose.
  Local alignment focuses on aligning subsequences within longer sequences,
  identifying regions of high similarity. This approach is particularly
  effective for sequences with significant differences or varying lengths,
  such as identifying conserved regions in genomes. The Smith-Waterman
  algorithm is typically used for local alignment.
  ```

  ---

**Q02: Dynamic programming**

- **prompt**:

  ```
  Good answer, and what is the main role of dynamic programming in sequence
  alignment?
  ```

- **model_name**: **GPT-4-turbo**
- **model_version**: **v2**
- **response**:

```
Dynamic programming is used to compute an optimal alignment score matrix
through a recursive process, enabling efficient identification of the best
alignment. In global alignment (Needleman-Wunsch), it ensures alignment
across the entire sequence, while in local alignment (Smith-Waterman), it
identifies the most similar subsequences. Dynamic programming guarantees
optimal solutions, making it a cornerstone of sequence alignment algorithms.
```

---

#### Q03: Smith-Waterman algorithm

- **prompt**:

```
Right, and why does the Smith-Waterman algorithm include "0" as an option in
the recurrence relation?
```

- **model_name**: **GPT-4-turbo**
- **model_version**: **v2**
- **response**:

```
The inclusion of "0" allows the alignment to terminate at the point where
the optimal subsequence ends, rather than forcing alignment across the
entire sequences. This ensures that the algorithm focuses only on highly
similar regions and ignores unrelated parts, making it ideal for local
alignment in divergent sequences.
```

#### Q04: MSA

- **prompt**:

```
Nice answer, and how does progressive alignment reduce computational
complexity in multiple sequence alignment (MSA)?
```

- **model_name**: **GPT-4-turbo**
- **model_version**: **v2**
- **response**:

```
Progressive alignment reduces computational complexity by aligning the most
similar sequence pairs first, guided by a phylogenetic tree or a distance
matrix, and incrementally building the MSA. This avoids the computationally
prohibitive task of optimizing the alignment of all sequences
simultaneously. Techniques like UPGMA or neighbor-joining are commonly used
to construct the guide tree.
```

---

#### Q05: multiple sequence alignment

- **prompt**

```
Good answer, and how can deep learning improve the accuracy of multiple
sequence alignment?
```

- **model_name**: **GPT-4-turbo**
- **model_version**: **v2**
- **response**:

```
Deep learning can enhance MSA by leveraging models like Transformers or
convolutional neural networks to learn patterns from large biological
sequence datasets. These models can predict conserved regions, refine
scoring matrices, and improve gap penalties. Techniques such as self-
supervised learning can utilize unlabeled sequence data to extract
biologically relevant features, leading to more accurate and context-aware
alignments.
```

---