Item #244 - Taming the Sequence “Thousand-Claw Dragon”: A Non-Combinatorial Path to Sequence Distance and CCC

(Sequence CCC without Mandatory BTP)

Conversation: 降服万爪龙算法三大落地应用

20260120

**Authors: Sizhe Tan
& GPT-Obot**

DBM-COT ITEM #244
Taming the Sequence “Thousand-Claw Dragon”:
A Non-Combinatorial Path to Sequence Distance and CCC

(Sequence CCC without Mandatory BTP)

Abstract

A common misconception in sequence analysis is that once tolerance (slop, mismatch, or fuzziness) is introduced, combinatorial search (e.g., BTP-style exhaustive path exploration) becomes inevitable.

This item demonstrates that such a conclusion is structurally false for sequence data.

By strictly separating different kinds of tolerance, and by aligning algorithmic design with the principle of Minimal Evolution Threshold, we show that most practical Sequence Distance and Sequence CCC problems can be solved efficiently without resorting to combinatorial search.

1. The False Premise: “Sequence + Tolerance ⇒ Combinatorial Explosion”

The so-called “Thousand-Claw Dragon” fear in sequence CCC arises from a hidden assumption:
Any form of tolerance necessarily introduces branching search paths.
This assumption conflates structural tolerance with search freedom. In reality, these are orthogonal concepts.

2. Three Strictly Separated Types of ToleranceDBM distinguishes three fundamentally different tolerance mechanisms:

2.1 Single-Element Placeholder (*)

- Matches exactly one token
- No skipping, no length variation
- Deterministic, position-preserving

This is structural expressiveness, not tolerance in the combinatorial sense.

2.2 Within-Token Slop (Character-Level)

- Bounded mismatch inside a token
- Implemented via Hamming or bounded edit distance
- Deterministic, local, no branching

This models minor evidence variation, not structural uncertainty.

2.3 Between-Token Slop (Position-Level)

- Token misalignment, gaps, or shifts
- The only category that *can* introduce combinatorial risk

Crucially, DBM does not treat this slop as a search dimension by default.

3. Two-Phases Search: Slop as Scoring, Not Branching

Even when between-token slop is required, DBM does not immediately resort to BTP.

Phase 1 — Tolerant Token Occurrence Search

A DBM-regex-like mechanism is used:

- strict token matching
- single-element *
- bounded within-token slop

This phase rapidly constrains the candidate space.

Phase 2 — One-Pass Directional Scan

Within the constrained candidate windows:

- slop is accumulated as penalty
- a single directional scan is performed
- early exit applies when budget is exceeded

No path branching.
No backtracking.
No combinatorial explosion.

Only if this process fails does DBM escalate to BTP as a last-resort fallback.

4. Sequence CCC without Mandatory BTP

Under this framework:

- Sequence Distance
- Pairwise Unaligned AND matches
- N-way Sequence CCC

can all be computed using:

1. Deterministic occurrence finding
2. Vote aggregation across sequences
3. One-pass penalty accumulation
4. Top-K CCC extraction

BTP is no longer a default requirement for sequence data.

5. Alignment with Human Cognition

This algorithmic structure is not accidental.
It aligns closely with human cognitive computation:

5.1 Minimal Evolution Threshold

The human brain does not evolve for unnecessary combinatorial reasoning.
Complex search is expensive and used only when survival-critical.
DBM mirrors this by keeping combinatorial search off the main path.

5.2 Positional Insensitivity of Minor Evidence

Human reasoning tolerates:

- small ordering deviations
- local mismatches
- weak evidence aggregation

Minor evidence affects confidence, not branching structure.

This directly corresponds to:

- within-token slop
- slop as penalty, not path
- vote-based CCC extraction

6. Implications

- Sequence CCC is not inherently exponential
- Most real-world sequence problems belong to a linear or near-linear regime
- BTP remains essential—but only for rare, extreme cases

The “Thousand-Claw Dragon” is not eliminated by force, but by structural understanding.

7. Conclusion

For sequence data, the perceived inevitability of combinatorial explosion is a misconception.
By respecting structural constraints, separating tolerance categories, and adopting a two-phases evaluation strategy, DBM provides a practical, cognitively aligned, and computationally efficient foundation for Sequence Distance and CCC.

The dragon is not slain—it is tamed.

DBM-COT ITEM #244（中文版）

驯服 Sequence 的“万爪龙”：

一种非组合爆炸的 Sequence Distance 与 CCC 路径

摘要

在序列分析领域，一个长期存在的误解是：

一旦引入容错（slop / mismatch / fuzzy），就不可避免地要进入组合搜索（如 BTP）。

本文指出：这一结论对于 Sequence 数据在结构上是错误的。

通过严格区分不同层级的容错机制，并遵循 小进化门槛原理，DBM 展示了：

绝大多数实际的 Sequence Distance 与 Sequence CCC 问题，都可以在不使用组合搜索的前提下高效解决。

1. 错误前提：Sequence + 容错 ⇒ 必然爆炸

所谓“万爪龙”的恐惧，来源于一个隐含假设：只要允许容错，就必须引入分支搜索。
这是将结构表达能力与搜索自由度混为一谈。
二者在 Sequence 问题中并不等价。

2. 三类严格区分的容错机制

2.1 单元素占位符 *

- 精确匹配一个 token
- 不跳跃、不变长
- 完全确定性这是结构表达增强，不是搜索容错。

2.2 Token 内部容错（within-token slop）

- 字符级有限错配
- 有界、局部、确定
- 不产生分支它对应的是枝节证据的不确定性。

2.3 Token 之间容错（between-token slop）

- 顺序偏移、间隔
- 唯一可能引入组合风险的类别但 DBM 并不默认将其转化为搜索维度。

3. Two-Phases Search：把 slop 变成评分，而不是路径

即便需要 token 之间的容错，DBM 仍不直接进入 BTP。

第一阶段：DBM-regex 类容错 token 搜索

- 严格 token
- 单元素 *
- token 内 slop

快速收缩候选空间。

第二阶段：单向 One-Pass Scan

在候选窗口中：

- slop 累计为罚分
- 顺序扫描
- 超预算即停

没有分支，没有回溯。
只有在失败时，才升级为 BTP 兜底。

4. 不必然依赖 BTP 的 Sequence CCC

在该体系下：

- Sequence Distance
- Pairwise AND Match
- N-way Sequence CCC

均可通过以下流程完成：

1. 确定性 occurrence 搜索

2. 多序列投票叠加

3. 单向罚分扫描

4. Top-K CCC 提取

BTP 不再是 Sequence 的默认解法。

5. 与人类认知的深度一致性

5.1 小进化门槛原理

人脑不会为不常用、非生死攸关的问题进化组合搜索能力。

DBM 同样避免在主流程中使用高复杂度算法。

5.2 对枝节证据顺序的不敏感

人类推理中：

- 小错不致命
- 顺序不绝对
- 多弱证据可累积

这正是：

- within-token slop
- slop 作为罚分
- 投票式 CCC 的认知原型。

6. 结论

Sequence 问题并非天生指数复杂。

通过结构化容错、两阶段评估与确定性扫描，DBM 为 Sequence Distance 与 CCC 提供了一条可解释、可扩展、符合认知的主干算法路径。

万爪龙不是被消灭的，
而是被驯服的。