ITEM #242 - Sequence CCC Kernel: A Cross-Domain CCC Core for DNA Two-Ways, ACLM Calling Graph, and LLM Span Mining

Conversation: 降服万爪龙算法三大落地应用

20260119

Authors: Sizhe Tan & GPT-Obot

DBM-COT ITEM #242 (EN)
Sequence CCC Kernel: A Cross-Domain CCC Core for DNA Two-Ways, ACLM Calling Graph, and LLM Span Mining

Item ID: 240

Keywords: Sequence CCC, Two-Ways CCC, CCC Kernel API, Budget/StopRule, Signature Dedup, DNA Functional Signal, ACLM SOS, Calling Graph Highway, LLM Sentence/Span Mining, Differential Tree Distance

Status: Draft (Engineering-Ready MVP)

Scope: Core algorithm abstraction + minimal Java 8 drop-in skeleton + unit tests

0. Abstract

This item formalizes a Sequence CCC Kernel as a reusable, cross-domain algorithmic core that "tames the combinational explosion (the 'ten-thousandclaw dragon')" while extracting stable CCC (Common Concept Core) fragments. The kernel is designed to be shared by three major DBM application lines:

1. DNA Two-Ways CCC (functional signal detection; ITEM #234 as the key reference)

2. ACLM SOS Chains / Calling Graph (distance for differential trees and "highway paths" in calling graphs)

3. LLM (mining high-quality sentence/span CCC for training signal enhancement and inference-time routing/scoring)

We additionally provide an engineering-ready Java 8 MVP (core + 3 adapters) and JUnit4 tests validating: budget enforcement, two-ways consistency, and signature dedup stability.

1. Motivation: One Kernel, Three Natural Habitats

Sequence CCC is not a domain trick; it is a structural alignment-andconsensus kernel. The key recurring problem is:
Given two sequences under heavy combinational explosion, extract stable, reusable consensus fragments that can be voted, merged, routed, and reused in downstream pipelines.
This same issue appears in:

- DNA: motifs/functional signals submerged in large backgrounds
- ACLM: path search explosion in SOS chains / calling graphs
- LLM: token-level statistics are too local; we need robust span-level structural consensus

2. Core Idea: CCC as Fragment Consensus, Not a Single Match

The kernel does not output a single alignment. It outputs:

- a set of CCC fragments (contiguous matches in MVP, extendable to gapped/beam states later),

- plus evidence (explored states, stop-rule pruning, explain trace),

- plus a stable signature for fragment voting and dedup.

This representation is crucial for:

- Two-Ways mutual verification (DNA)
- “Highway” path extraction (ACLM) • Span mining and routing signals (LLM)

3. Kernel API (Domain-Agnostic)

The kernel is split into four layers.

3.1 Sequence & Element Layer

- SequenceView<E>: read-only access to an indexed sequence

- ElementOps<E>: domain-specific comparability and substitution/gap cost rules
  - DNA: A/C/G/T/N matching policy

  - ACLM: SOS token equality or softened equivalence later

    - LLM: token-id equality or future embedding-assisted equivalence

3.2 Fragment Layer

- CccFragment: (aStart,aEnd,bStart,bEnd,score,signature64)

- signature64 enables stable dedup and voting/merge across passes and pipelines.

3.3 Result & Evidence Layer

- CccResult: totalScore + fragments + evidence + optional attrs

- CccEvidence: exploredStates / prunedByStopRule / explainLines / bestPartialScore

3.4 Kernel Entrypoint

- SequenceCccKernel<E>.compute(a,b,ops,cfg)

- TwoWaysCccKernel orchestrates forward+reverse pass and merges fragments back into the original coordinate system.

4. Stop-Rule & Budget: First-Class "Dragon-Taming Controls"

To defeat long-tail combinational explosion, the kernel treats budget and stop-rule as first-class citizens:

- maxExploredStates: hard budget for expansions/comparisons
- stallStepsToStop: early stop on low-value exploration tail
- StopRule: a plug-in policy hook for S1/S2/S3 style stop systems

This design is consistent with DBM's principle: timely stop is part of correctness, not a "performance tweak".

5. Cross-Domain Mappings

5.1 DNA (Two-Ways CCC)

- Purpose: functional signal detection under noisy background
- Two-Ways CCC: forward + reverse mutual verification reduces false positives
- Output fragments: signal candidates and evidence for hypothesis export

5.2 ACLM (Differential Tree Distance + Calling Graph "Highways")

- CCC fragments define a sequence-structural distance used for differential routing/fabric
- Recurrent CCC fragments across chains naturally form "highway-like" calling graph paths
- Benefit: better search allocation, reduced oscillation, improved explainability

5.3 LLM (Span/Sentence CCC)

- CCC fragments mined from corpora can yield higher-quality spans than word/bigram heuristics
- CCC density and stability can be used for:

◦ training sample selection and weighting

◦ inference-time routing/branching

◦ scoring and compute allocation (structural signal for resource dispatch)

6. Engineering Deliverable: Java 8 Drop-in MVP Skeleton

We provide a minimal, compilable module with:

- core/: kernel interfaces, config, hashing, sliding-window MVP miner, twoways orchestrator
- dna/: DNA adapter (DnaOps, optional reverse-complement helpers)
- aclm/: ACLM token adapter (AclmOps)
- llm/: LLM token adapter (LlmTokenOps)

The MVP mines contiguous-match CCC fragments with:

- deterministic sorting
- signature-based dedup
- bounded explain trace
- budget/stop-rule enforcement

7. Unit Tests (JUnit4)

The MVP includes 3 tests: 1.

Budget enforcement

Asserts explored states hit the configured cap and stop-rule prunes.

2. Two-Ways consistency

   Uses a reversal-invariant case; verifies best CCC score is stable and signature is deterministic across runs.

3. Signature dedup stability

   Uses repeated motifs; ensures duplicated fragments collapse into one per signature and top fragment remains stable.

8. Next Extensions (Non-Breaking)

The API is stable; future upgrades can be internal:

- replace sliding-window miner with beam-state CCC assembly
- add demotion/drop filters for extreme repetition terms
- introduce soft-match costs (synonyms/equivalence classes)
- integrate differential-tree embedding and calling-graph “highway” scoring as post-processors

9. Conclusion

Sequence CCC Kernel unifies multiple DBM fronts under a single structuralconsensus core:

- DNA Two-Ways CCC gains a reusable kernel with explicit budget governance
- ACLM gains a principled distance engine and a mechanism to discover “highway” calling paths
- LLM gains span-level CCC signals for data selection, routing, and scoring

This item establishes the engineering foundation to scale CCC from a single algorithm into a cross-domain structural intelligence primitive.

DBM-COT ITEM #242 (中文)

Sequence CCC Kernel：跨 DNA Two-Ways、ACLM Calling Graph 与 LLM Span Mining 的统一 CCC 内核

Item ID：240

关键词：Sequence CCC、Two-Ways CCC、CCC Kernel API、Budget/Stop-Rule、Signature 去重、DNA 功能信号、ACLM SOS、Calling Graph 高速公路、LLM 句段抽取、差分树距离

状态：Draft（工程可用 MVP）

范围：算法内核抽象 + Java 8 小可编译骨架 + 单元测试

0. 摘要

本 ITEM 固化一套可跨域复用的 Sequence CCC Kernel：它以“降服万爪龙（组合爆炸）”为前提，在预算与停机规则约束下稳定析出 CCC（Common Concept Core）片段共识。该内核天然适配 DBM 三条关键战线：

1）DNA Two-Ways CCC（功能信号检测；ITEM #234 为关键参考）

2）ACLM SOS Chains / Calling Graph（差分树距离计算与 Calling Graph “高速公路”路径）

3）LLM（抽取高质量 sentence/span 级 CCC，用于训练信号增强与推理时路由/打分）

同时，本 ITEM 给出 Java 8 可编译 MVP 工程骨架（core + 3 个 adapter），并提供 JUnit4 测试验证：budget 生效、two-ways 一致性、signature 去重稳定。

1. 动机：一套内核，三个自然栖息地

Sequence CCC 不是某领域的“技巧”，而是一个结构级对齐与共识内核。反复出现的共性难题是：

在强组合爆炸条件下，如何提取稳定、可复用、可投票、可合并的结构共识片段（CCC fragments），并将长尾探索纳入可控预算。

该问题在三域中同构出现：

- DNA：功能信号被噪声背景淹没
- ACLM：SOS 链与 Calling Graph 的路径组合爆炸

- LLM：token 统计单位过于局部，需要句段级稳定结构共识

2. 核心思想：CCC 是“片段共识”，不是单一匹配

Kernel 输出不是单条对齐，而是：

- 一组 CCC fragments（MVP 先做连续片段，后续可扩展 beam/gap）
- 一份 evidence（探索规模、停机剪枝、解释链）
- 一个 稳定 signature（用于跨 pass 的投票、合并、去重）

该表示法直接支撑：

- DNA 的 Two-Ways 互证
- ACLM 的“高速公路”主干识别
- LLM 的句段抽取与路由/调度信号

3. Kernel API（领域无关抽象）

API 分为四层。

3.1 序列与元素层

- SequenceView<E>：只读序列视图
- ElementOps<E>：领域匹配规则与代价策略
  - DNA：A/C/G/T/N 匹配策略
  - ACLM：SOS token 等价/弱等价（后续可软化）
  - LLM：token-id 等价（后续可引入 embedding 辅助）

3.2 片段层

- CccFragment：(aStart,aEnd,bStart,bEnd,score,signature64)
- signature64：保证稳定去重与投票合并

3.3 结果与证据层

- CccResult：totalScore + fragments + evidence + 可选 attrs
- CccEvidence：exploredStates / prunedByStopRule / explainLines / bestPartialScore

3.4 Kernel 入口

- SequenceCccKernel<E>.compute(a,b,ops,cfg)
- TwoWaysCccKernel：负责 forward + reverse 的协调、坐标映射与合并去重

4. Stop-Rule 与 Budget：降服万爪龙的“正当控制系统”

要打败长尾爆炸，Kernel 将 budget 与 stop-rule 视为第一等公民：

- maxExploredStates：硬预算上限
- stallStepsToStop：无收益长尾早停
- StopRule：可插拔策略钩子（可演进到你常用的 S1/S2/S3）

这与 DBM 的一贯原则一致：及时收手是正确性的一部分，而非性能优化附属品。

5. 跨域映射

5.1 DNA（Two-Ways CCC）

- 目的：在噪声背景下检测功能信号
- Two-Ways：正向 + 反向互证降低假阳性
- 输出：CCC fragments 作为候选信号与证据链，用于假设包导出

5.2 ACLM（差分树距离 + Calling Graph 高速公路）

- CCC fragments 定义序列结构距离，支撑差分树 routing fabric
- 重复出现的 CCC 片段会自然形成 Calling Graph 的“高速公路主干”
- 收益：更优资源分配、更少震荡、更强可解释性

5.3 LLM（Sentence/Span CCC）

- 从语料中挖出高质量 span，优于 word/bigram 级启发式
- CCC 的密度与稳定性可用于：
  - 训练样本选择与权重
  - 推理时路由/分支 ◦ 打分与计算资源调度

  (结构信号驱动)

6. 工程交付：Java 8 可编译 Drop-in MVP 该

MVP 工程包含：

- core/：接口、配置、hash、滑窗挖掘器、two-ways orchestrator
- dna/：DNA adapter（含可选 reverse-complement 工具）
- aclm/：ACLM token adapter
- llm/：LLM token adapter

内核特性：

- deterministic 排序

- signature 去重

- explain trace 有界输出 • budget/stop-rule 强制生效

7. 单元测试（JUnit4）

提供 3 个测试： 1）Budget

生效

断言 explored states 触顶，且 stop-rule 发生剪枝。

2）Two-Ways 一致性

使用反转不变案例，验证 佳 CCC 分数与 signature 跨运行稳定一致。

3）Signature 去重稳定重复 motif 多次出现时，片段按 signature 收敛为一个，且 top fragment 稳定。

8. 下一步扩展（不破坏 API）

API 设计为稳定层，后续升级可全部发生在内部实现：

- 用 beam-state CCC assembly 替换滑窗挖掘

- 加入极端重复项过滤（drop/demote）

- 引入软匹配代价（同义/等价类）

- 增加差分树 embedding 与 calling graph highway scoring 的后处理器

9. 结论

Sequence CCC Kernel 将 DBM 的多个关键前沿统一到一个结构共识原语之上：

- DNA Two-Ways CCC 获得可复用内核与明确预算治理

- ACLM 获得原则化距离引擎与“高速公路”路径发现机制
- LLM 获得句段级 CCC 信号用于样本筛选、路由与打分

本 ITEM 给出了从方法论到工程 MVP 的完整闭环，为 CCC 从单一算法升级为跨域 Structural Intelligence Primitive 奠定基础。