Computational Molecular Biology and Bioinformatics

CORNETO

Malay Bhattacharyya

Associate Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

August, 2025

1. Introduction

2. The CORNETO Method
   - Mathematical preliminaries
   - The unified approach
   - An illustrative example

3. References

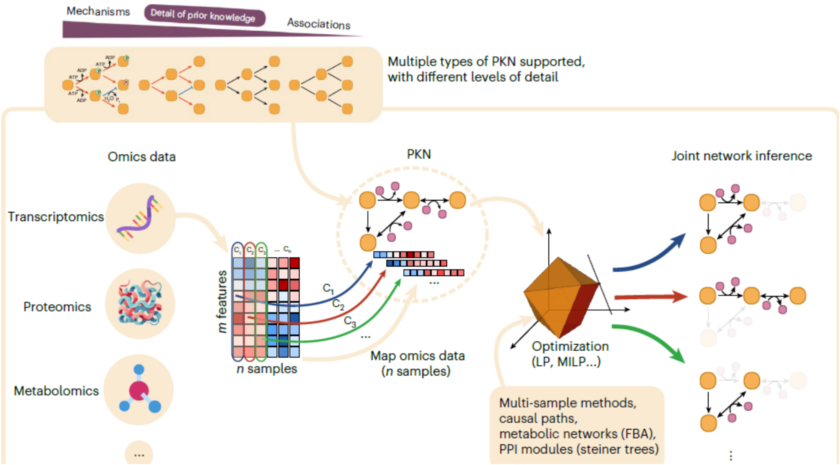
What is CORNETO?

CORNETO is a unified mathematical framework that generalizes a broad range of techniques for learning biological networks using omics data and prior knowledge [1].

Advantages:

- All methods are implemented consistently with a common vocabulary and API;
- It identifies and reuses common components across different methods, streamlining development; and
- It provides exact algorithmic implementations, thereby allowing solvers to find optimal or suboptimal solutions.

Overview of CORNETO's framework



Mathematical preliminaries

Mixed-integer Programming: It is a type of constrained optimization problem that involves decision variables that can be both integer and continuous. A mixed-integer programming problem can be defined as follows:

$$\min_{x \in \mathbb{R}^n, y \in \mathbb{Z}^m} f(x, y) \quad \text{or} \quad \max_{x \in \mathbb{R}^n, y \in \mathbb{Z}^m} f(x, y)$$

subject to

$$g_i(x, y) \leq 0, i = 1, 2, \ldots, k, \quad \text{and} \quad h_j(x, y) = 0, j = 1, 2, \ldots, l,$$

where $f : \mathbb{R}^n \times \mathbb{Z}^m \rightarrow \mathbb{R}$ is the objective function, which involves both real and integer decision variables.
If the functions f, g_i and h_j are linear, the problem is termed as a mixed-integer linear programming (MILP) problem. If all variables are continuous, the problem is a linear programming (LP) problem.

Mathematical preliminaries

Network flow problem on digraphs: It aims to find the vector of flows that satisfies the flow capacity constraints and flow conservation constraints.

The canonical form of the network flow problem is formulated as an LP, which is expressed as the following minimization problem:

$$\min c^T x \quad \text{subject to} \quad Ax = b, 0 \leq x \leq x_{\max},$$

where A denotes the vertex–edge incidence matrix, b represents the net flow entering or leaving each vertex (sources having negative values, sinks having positive values, otherwise zeros).

For network inference problems (that not only aim to find the values of certain parameters but also determine the structure of the network), this can be extended and formulated as an MILP.

The core framework

We redefine network inference by transforming it into a network flow-based problem, exploiting the mixed-integer optimization framework commonly used in such problems. This enables joint inference by utilizing multiple samples simultaneously, allowing the model to improve network inference by borrowing information across all samples.

Outline:

1. Apply a mapping function ϕ on omics data to produce an annotated hypergraph.
2. Perform a graph transformation ψ for pruning the hypergraph.
3. Add an objective function $f()$ and a set of variables and constraints.
4. Cast the entire problem as a mixed-integer network-flow formulation.

Apply a mapping function

Integrating experimental data with biological prior knowledge is an effective alternative, particularly when samples are limited.

Starting from omics data $\mathbf{D} \in \mathbb{R}^{m \times n}$ and a context-free hypergraph (Prior knowledge networks, abbreviated as PKNs) $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, the mapping $\phi : (\mathcal{H}, \mathbf{D}) \to \mathcal{H}'$ projects measurements onto $\mathcal{H}$, producing an annotated hypergraph $\mathcal{H}'$.

Perform a graph transformation

The graph transformation $\psi : \mathcal{H}' \rightarrow \mathcal{H}''$ applies preprocessing strategies such as pruning unreachable vertices or irrelevant edges and, most importantly, inserting source and sink edges to enable the injection and extraction of flows in a method-specific manner. This transformation constitutes the first step in reformulating a given method into a flow-based problem.

For example, in metabolic networks, these edges may already be present in the PKN (for example, importer and exporter reactions) and, thus, may not be required to guarantee that flows can traverse the network.

Add an objective function

Applying ϕ and ψ to each sample yields a collection of transformed hypergraphs, which we merge into a union hypergraph $\mathcal{H}_u$.

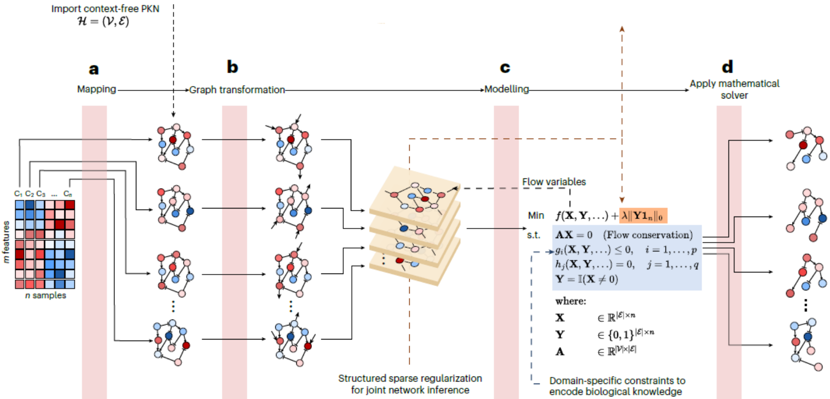
Flows and indicator variables model the selection of graph structures (paths, trees, direct acyclic graphs or other combinatorial objects), with flows $\mathbf{X}$ and edge indicators $\mathbf{Y}$ determined automatically.

Mixed-integer network-flow formulation

Finally, the problem is formulated as an MILP on directed hypergraphs.

Similar to multi-commodity network flows, which involve multiple resources (commodities) moving through a network simultaneously with shared constraints, we extend the formulation to support multiple simultaneous flows. In the context of the framework, each sample or condition is associated with a flow, and each flow can be subjected to different constraints.

An illustrative example



References

1. Rodriguez-Mier, P., Garrido-Rodriguez, M., Gabor, A. and Saez-Rodriguez, J., Unifying multi-sample network inference from prior knowledge and omics data with CORNETO. Nature Machine Intelligence, 7:1168-1186, 2025.
All code is publicly available on:
`https://github.com/saezlab/corneto`