SAPIENZA
UNIVERSITÀ DI ROMA

AI-driven analysis of molecular pathways

Facoltà di Ingegneria dell'informazione, informatica e statistica
Informatica

Ionuț Cicio
ID number 2048752

Advisor	Co-Advisor
Prof. Mancini	Prof. Tronci

Academic Year 2024/2025

AI-driven analysis of molecular pathways
Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author’s email: cicio.2048752@studenti.uniroma1.it

Abstract

Contents

Chapter 1

Introduction

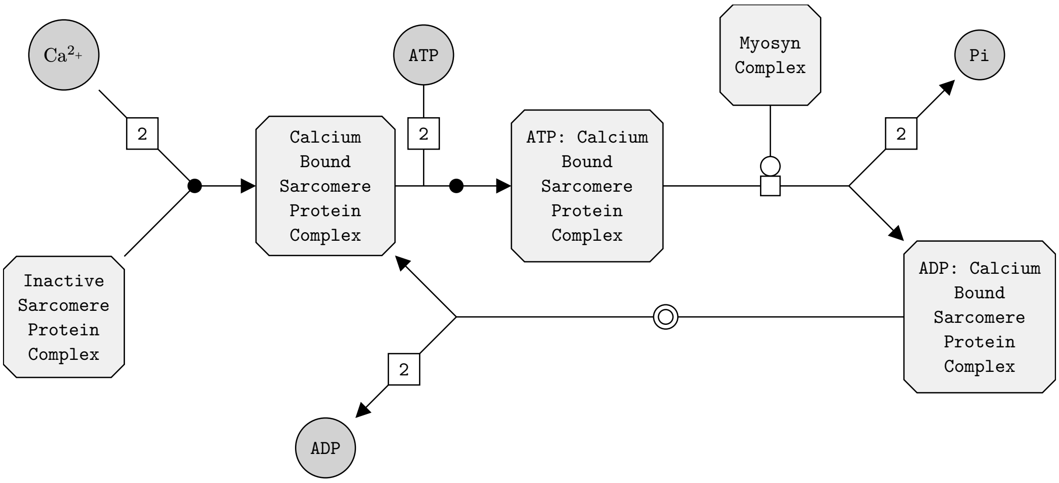
Biological systems such as organisms, cells, or biomolecules are highly organized in their structure and function, [2]. [Such structures can help study the and compare biological functions of different organisms]. [In order to understand, formalize and abstract these structures some kind of modeling is needed, be it mathematical or not (what not?)].

In this report is presented a proof of concept which generates biological models by using the concept of reachability and databases of biochemical reactions.

[A fundamental component of bioinformatics is data integration, (a problem is) i.e. partial information in distributed databases is needed]. [One such data source is Reactome [1], which is a qualitative network database] REACTOME is an open-source, open access, manually curated and peer-reviewed pathway database. [The goal with this work is to use qualitative data in the Reactome database to generate quantitative models, and use BBO (cite something) techinques to "validate" (validate is not good, find something else) these models]

[Validating complex biological models is a computationally intensive task (cite something) thus HPC clusters are required in order.] Such clusters may not always be fully available (i.e. there are multiple users and multiple experiments running on the cluster + there are some limits on job running times), so in order to better distribute the load of the validation task some infrastructure work is needed.

1.1 Computational modeling of biological systems


Ca^{2+}
ATP
Myosyn Complex
Pi
2
2
2
Calcium Bound Sarcomere Protein Complex
ATP: Calcium Bound Sarcomere Protein Complex
Inactive Sarcomere Protein Complex
ADP: Calcium Bound Sarcomere Protein Complex
2
ADP

Striated muscle contraction is a process whereby force is generated within striated muscle tissue, resulting in a change in muscle geometry, or in short, increased force being exerted on the tendons. Force generation involves a chemo-mechanical energy conversion step that is carried out by the actin/myosin complex activity, which generates force through ATP hydrolysis. Striated muscle is a type of muscle composed of myofibrils, containing repeating units called sarcomeres, in which the contractile myofibrils are arranged in parallel to the axis of the cell, resulting in transverse or oblique striations observable at the level of the light microscope. Here striated muscle contraction is represented on the basis of calcium binding to the troponin complex, which exposes the active sites of actin. Once the active sites of actin are exposed, the myosin complex bound to ADP can bind actin and the myosin head can pivot, pulling the thin actin and thick myosin filaments past one another. Once the myosin head pivots, ADP is ejected, a fresh ATP can be bound and the energy from the hydrolysis of ATP to ADP is channeled into kinetic energy by resetting the myosin head. With repeated rounds of this cycle the sarcomere containing the thin and thick filaments effectively shortens, forming the basis of muscle contraction.

1.2 Qualitative network models augmentation

1.2.1 Qualitative network models

TODO: here cite reactome, put in "notes" the UML used later for the query etc...

1.3 Outline

Chapter 2

Preliminaries

We denote by $\mathbb{R}, \mathbb{R}_{0+}, \mathbb{Z}, \mathbb{N}$ the sets of, respectively, real, non- negative real, integer, and non-negative integer numbers.

The terms "physical entity" and "species" are used interchangibly in this document, since Reactome (TODO: cite reactome document model / glossary) uses "PhysicalEntity" to reference (TODO: biological species?) and SBML (TODO: cite SBML documentation) uses "Species".

Reactome uses "ReactionLikeEvent" to refer to generic reactions (TODO: cite glossary and use glossary to tell different types of reactions)

Definition 1 (Biological network). A biological network G is a tuple (S, R, E, σ) where

- $S = U \cup X \cup Y$ is the set of species of the biological network, where
  - U is the set of input species
  - Y is the set of output species
  - X is the set of other species in the network
- R is the set of reactions in the biological network
- $E \subseteq S \times R$

Chapter 3

Quantitative model generation

3.1 Reachability

3.2 Scenario definition

(TODO: define what is an expansions, why do we need a scenario, etc...it might be important to study what a subsection which contains two specific species behaves)

(TODO: define what is a Pathway, and what a Pathway is in terms of Reactome)

(TODO: maybe do a chapter about Reactome, or something simpler before about the generation, maybe about "Reachability", for reachability you need a definition of a network)

Definition 2 (Biological scenario). A

A scenario is defined by

- a set of physical entities from which to start the expansions
- a set ot pathways to which to limit (constraint?) reactions to
- a max depth for recursion (/reachability) (TODO: a max depth in terms of nodes in the path, not in the number of reactions in the path, for that apoc is needed)
- a set of physical entities to exclude from reachability
- a partial order of the species

(as per figure ... of UML etc...)

Chapter 4

Satisfiability problem definition

Chapter 5

Optimization architecture

When analyzing the scalability of a parallel algorithm on a HPC cluster, an interesting problem is the one of trying to predict how would the algorithm scale on a cluster with a higher degree of parallelism compared to the one available for experiments. This document presents one possible way to make this kind of analysis when the computation is asynchronous and the sequence of values in the computation depends on the state of an orchestrator. OpenBox, a system design for generalized black-box optimization [1], will be the main case study for this type of systems.

5.1 OpenBox

OpenBox is an efficient open-source system designed for solving gener alized black-box optimization (BBO) problems. It can be used either as a Standalone python package or Online BBO service [1].

5.2 Orchestrator Worker infrastructure

OpenBox is an efficient open-source system designed for solving generalized black-box optimization (BBO) problems. It can be used either as a Standalone python package or Online BBO service @open-box.

OpenBox has a great support for bayesian optimization, so that will be the main subject of the analysis @open-box-automatic-algorithm-selection.

5.3 Scalability analysis

Chapter 6

Experiments