

Introduction to pyCOT

Tomás Veloz

1 Introduction

PyCOT is a library to apply chemical organization theory for modeling and analysis purposes. First algorithms aimed to compute organizations and self-maintaining sets were developed by the group of Peter Dittrich between 2007 and 2010. The idea of developing a library was conceived by Tomas Veloz, Pedro Maldonado and Alejandro Bassi in Santiago, Chile, 2017. While Bassi and Maldonado explored how to perform calculations in efficient manners in R and C respectively, the first development in python was pyRN between 2020 and 2022 as part of the Templeton grant "Origins of Goal Directedness", and is available in github¹. That preliminary version contains some of the important elements of pyCOT, but the complex handling of data structures implied lack in versatility, so Tomas Veloz rebuilt a version of the library using a more efficient class structure, introducing the notion of subRN and supRN that can inherit maximal information, and adding several functions that permit COT analysis, in an effort to integrate and complete the scattered previous developments.

2 Understanding the pyCOT object

A general COT object of analysis is a part of a reaction network $(\mathcal{M}, \mathcal{R})$ where $(\mathcal{M}, \mathcal{R})$ is a set of species and $(\mathcal{M}, \mathcal{R})$ is a set of reactions. Note that every pair $(X \subset \mathcal{M}, R \subset \mathcal{R})$ are in principle a possible choices for a part of the reaction network. This implies that some routines might require exploration of exponential combinations. For this reason, efficiency is important.

Previous work reveals we encounter three levels of analysis: Relational, Stoichiometric and Dynamical. Interestingly, these require different (and increasingly complex) mathematical structures to define properties.

¹<https://github.com/pmaldona/pyRN>

Relational properties require set-theoretical operations such as union, intersection, containment. Stoichiometric properties require linear algebraic manipulations. Dynamical properties require simulations.

2.1 Identification of species and reactions

For this reason, the basic structure of the pyCOT object identifies species and reactions in three different ways:

- SpBt/RnBt: Bitarray identification
- SpVec/RnVec: Vector (numpy.array) identification
- SpStr/RnStr: List of strings (species/reaction names) identification

The library is built in a way that relational calculations are made at the bitarray identification and algebraic calculations are made at the np.array identification. This helps to optimize the speed at which operations are made.

By default, we work in notebooks at the 'Str' identification, which is easier to identify by eye, but functions generally operate in the other have functions that can exchange one representation into another to easily manipulate