Algorithms for Enumerating Irreducible and Closed Autocatalytic Sets

Wim Hordijk *

Posted Date: 24 April 2026

doi: 10.20944/preprints202604.1787.v1

Keywords: chemical reaction networks; autocatalysis; RAF sets; origin of life

 doi:10.20944/preprints202604.1787.v1

Article

Algorithms for Enumerating Irreducible and Closed Autocatalytic Sets

Wim Hordijk

Independent Researcher, The Netherlands; wim@worldwidewanderings.net

Abstract

Autocatalytic sets are chemical reaction networks in which the molecules mutually catalyze each other's formation supported by an ambient food set. They are believed to have played an important role in the origin of metabolism and life, and have been studied extensively both theoretically and experimentally. Autocatalytic sets often consist of a hierarchical structure of smaller and smaller autocatalytic subsets. Of particular interest are irreducible autocatalytic sets and closed autocatalytic sets. Previously, it has been shown that finding *all* such autocatalytic subsets is, in principle, intractable. Here, several algorithms are presented to enumerate irreducible and closed autocatalytic sets, either exhaustively (but only practical in limited cases) or in the form of a random sample. Their implementation in a C++ program, made available as a GitHub repository, is then tested on instances of a computational model of chemical reaction networks known as the binary polymer model.

Keywords: chemical reaction networks; autocatalysis; RAF sets; origin of life

1. Introduction

An *autocatalytic set* is a chemical reaction network in which (1) each reaction is catalyzed by at least one of the molecules involved in the network, and (2) each molecule can be produced through a sequence of reactions from the network itself starting from a given food set. This food set is a subset of molecules that are assumed to be available from the environment.

The original concept of autocatalytic sets was introduced by Kauffman [1,2,3] in the context of the origin of life, as a metabolism-first alternative to the still dominant genetics-first paradigm. The general concept has been studied extensively, both theoretically as well as experimentally [4]. It was mathematically formalized as *Reflexively Autocatalytic and Food-generated* sets, or RAF sets [5–7]. Autocatalytic sets are indeed believed to have played an important role in the origin of metabolism and life [8–12].

An efficient (polynomial-time) algorithm for finding RAF sets in arbitrary chemical reaction networks was introduced earlier [6,13]. This algorithm finds the *maximal RAF*, or maxRAF, within a given reaction network, i.e., the union of all RAFs that might exist within that network. Indeed, it was shown that a maxRAF often consist of a hierarchical structure of smaller and smaller RAF subsets, or subRAFs [14]. Of particular interest are the *irreducible RAFs*, or iRAFs [15,16], and *closed RAFs*, or cRAFs [17,18].

Irreducible RAFs are RAF sets from which none of the reactions can be removed without losing the RAF property. In other words, they represent the minimal RAFs within a network. Closed RAFs are the dynamically stable subRAFs in a network, and have been shown to be the equivalent of *chemical organizations* [19]. Importantly, the existence of such subRAFs allow autocatalytic sets to be evolvable, at least to some extent [14,20–23]. However, for both types of subRAFs it has been shown that finding *all* of them within a maxRAF is, in general, intractable, as there may be exponentially many.

In practice, though, the number of iRAFs and cRAFs within a maxRAF is usually much smaller than the theoretical upper limit. Here we formally present various algorithms for enumerating iRAFs and cRAFs within a given RAF. We then test the performance of their implementations on a particular

 doi:10.20944/preprints202604.1787.v1

computational model of chemical reaction networks known as the binary polymer model. As expected, methods for exhaustively enumerating *all* iRAFs or cRAFs only work on relatively small instances, but heuristic methods for randomly *sampling* them work well for both iRAFs and cRAFs. A general-purpose implementation of these various algorithms in a C++ program is available as a GitHub repository (https://github.com/wimhor/RAF).

2. Autocatalytic Sets

2.1. RAF Sets

To formally define an autocatalytic set, first a *chemical reaction network* (CRN) is defined as a tuple $Q = \{X, R, C, F\}$, where:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of *molecule types*.
- $R = \{r_1, r_2, \ldots, r_m\}$ is a set of *reactions*. A reaction $r \in R$ is of the form

$$a_1 x_1 + \cdots + a_k x_k \to b_1 y_1 + \cdots + b_l y_l$$

  where the a_i and b_j are positive integers representing the reaction stoichiometry, and $x_i, y_j \in X$. Note that a reaction can also be bi-directional, where the reactants x_i and products y_j swap roles in case of the reverse direction.
- $C \subseteq X \times R$ is a set of *catalysis assignments*. A catalysis assignment is a pair $(x, r) : X \in X, r \in R$, indicating that molecule type x can catalyze reaction r. A reaction can have more than one catalyst, and a molecule type can catalyze more than one reaction.
- $F \subset X$ is a *food set* consisting of molecule types that are assumed to be available from the environment. In other words, they do not necessarily need to be produced by any of the reactions in the set R.

In addition, the following definitions are useful as well.

- $\rho(r) = \{x_1, \ldots, x_k\}$ is the *set of reactants* of a reaction r.
- $\pi(r) = \{y_1, \ldots, y_l\}$ is the *set of products* of a reaction r.
- $\rho(R') = \cup_{r \in R'} \rho(r)$ is the set of all reactants of all reactions $r \in R' \subseteq R$.
- $\pi(R') = \cup_{r \in R'} \pi(r)$ is the set of all products of all reactions $r \in R' \subseteq R$.
- $A_R(X') = \{r \in R : \rho(r) \subseteq X'\}$ is the set of *active reactions* for a set of molecules $X' \subseteq X$ relative to a set of reactions R, i.e., all reactions in R that have all their reactants in X'.
- $A_{R,C}(X') = \{r \in R : (\rho(r) \subseteq X') \wedge (\exists x \in X' : (x, r) \in C)\}$ is the set of *catalytically active reactions* for a set of molecules $X' \subseteq X$ relative to a set of reactions R and catalysis assignments C, i.e., all reactions in R that have all their reactants and at least one catalyst in X'.

Given a CRN Q and a subset of molecule types $X' \subset X$, the *closure* $\mathrm{cl}_R(X')$ of X' relative to R is the minimal subset W of X that contains X' and that further satisfies the condition that for each reaction $r \in R$:

$$\rho(r) \subseteq W \implies \pi(r) \subseteq W.$$

Informally, $\mathrm{cl}_R(X')$ is X' together with all molecule types that can be produced from X' by repeated applications of reactions from R (regardless of whether a reaction is catalyzed or not), until no new molecule types are produced. In case of a bi-directional reaction, the reverse direction (with $\rho(r)$ and $\pi(r)$ swapped) also needs to be checked. Note that the closure of a set X' can be computed efficiently, as will be shown below.

Similarly, the *catalytic closure* $\mathrm{cl}_{R,C}(X')$ of X' relative to R is the minimal subset W of X that contains X' and that further satisfies the condition that for each reaction $r \in R$:

$$(\rho(r) \subseteq W) \wedge (\exists x \in W : (x, r) \in C) \implies \pi(r) \subseteq W.$$

Informally, $\mathrm{cl}_{R,C}(X')$ is X' together with all molecule types that can be produced from X' by repeated applications of *catalyzed* reactions from R (i.e., at least one catalyst needs to be present for the reaction

 doi:10.20944/preprints202604.1787.v1

to happen), until no new molecule types are produced. Again, in case of a bi-directional reaction, the reverse direction needs to be checked as well. An efficient method for constructing the catalytic closure will be presented below.

Now, given a CRN $Q = \{X, R, C, F\}$ and a subset of reactions $R' \subseteq R$, R' is:

1. *Reflexively Autocatalytic* (RA) if for each reaction $r \in R'$ there is at least one molecule type in the closure of the food set (relative to R') that can catalyze reaction r, and
2. *Food-generated* (F) if for each reaction $r \in R'$ all its reactants are in the closure of the food set (relative to R').

A subset of reactions R' is *Reflexively Autocatalytic and Food-generated* (RAF) if both conditions hold [6]. Note that for bi-directional reactions, if the products are in the closure of the food set, then so are the reactants, so there is no need to check the reverse direction.

More formally, given a CRN $Q = \{X, R, C, F\}$ and a subset $R' \subset R$, R' is a RAF set if for each $r \in R'$:

1. $\exists x \in \mathrm{cl}_{R'}(F) : (x, r) \in C$, and
2. $\rho(r) \subseteq \mathrm{cl}_{R'}(F)$.

In words, a subset of reactions R' is a RAF set if for each of its reactions at least one possible catalyst and all reactants are in the closure of the food set (relative to R').

Figure 1 shows an example with $X = \{f_1, \ldots, f_4, p_1, \ldots, p_6\}$, $F = \{f_1, \ldots, f_4\}$, $R = \{r_1, \ldots, r_6\}$, and $C = \{(p_1, r_2), (p_2, r_1), (p_4, r_3), (p_4, r_4), (p_6, r_5)\}$. The graph representation used is common for chemical reaction networks, with dots denoting molecule types and boxes denoting chemical reactions [24], and with catalysis added as dashed gray arrows.

The subset of reactions $R' = \{r_1, r_2, r_3, r_4\}$, contained within the blue polygon, is a RAF set within the full network. Note that reaction r_6 has no catalysts at all, and can thus not be part of a RAF set. Furthermore, this reaction produces molecule type p_6, which is the only possible catalyst for reaction r_5, which can thus not be part of a RAF set either.

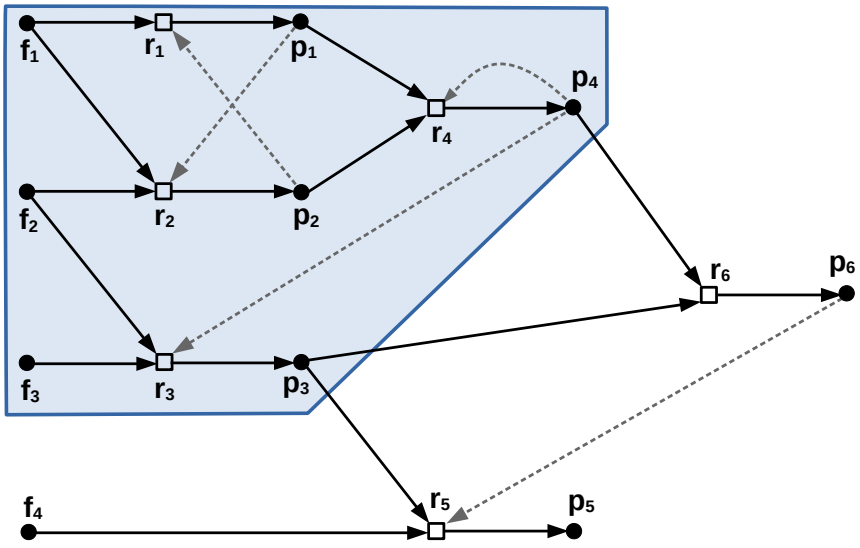

Figure 1. An example CRN containing a RAF set (within the blue polygon).

2.2. CAF Sets

The more restricted notion of a *Constructively Autocatalytic and Food-generated* set (CAF) is a RAF set $R' \subseteq R$ with the additional condition that the reactions in R' can be ordered in such a way that for each reaction $r_i \in R'$ in this ordering, each reactant and at least one catalyst is produced by some earlier reaction $r_j \in R', j < i$, or otherwise is present in the food set F [25]. More formally, a CAF

 doi:10.20944/preprints202604.1787.v1

$R' \subset R$ is defined as $R' = A_{R,C}(\mathrm{cl}_{R,C}(F))$, i.e., the catalytically active set of the catalytic closure of the food set (which may be empty, in which case there is no CAF in R).

The RAF in Figure 1 is not a CAF, nor does it contain one. For example, molecule types p_1 and p_2 catalyze each other's formation, so one of the reactions r_1 and r_2 has to happen spontaneously (i.e., uncatalyzed) at least once before the RAF set can be realized dynamically. Chemical reactions can always happen spontaneously, as a catalyst primarily speeds up the *rate* at which that reaction happens. So, although the RAF set is present *topologically* (i.e., in the network representation), there may be a certain waiting time before it is realized *dynamically*. But once one of these two reactions happens spontaneously, the other reaction can proceed catalyzed and, in turn, create the necessary catalyst for the first reaction. In contrast, a CAF requires at least one catalyst to be in the food set.

2.3. Irreducible RAF Sets

In the example in Figure 1, the set $R' = \{r_1, r_2, r_3, r_4\}$ (within the blue outline) is the maxRAF for the given network R. However, this maxRAF contains two (nested) subRAFs: $R_1 = \{r_1, r_2, r_4\}$ and $R_2 = \{r_1, r_2\}$, which are both RAF sets themselves.

Since the maxRAF contains at least one proper subRAF, it is "reducible": One or more reactions can be removed while still leaving a (smaller) RAF set. Similarly, R_1 contains a proper subRAF (R_2), and is therefore also reducible. However, R_2 is *irreducible* (or *minimal*), as it contains no proper subRAF. More formally, a RAF R is irreducible if for each $r \in R$ the set $R \setminus \{r\}$ does not contain a RAF.

Previously it was shown that in principle there can be exponentially many irreducible RAFs, or iRAFs, within any RAF [14]. In other words, if a RAF consists of n reactions, there can be $\mathcal{O}(2^n)$ iRAFs contained within it. This means that enumerating all of them is, in principle, intractable for large RAFs.

2.4. Closed RAF Sets

A RAF $R' \subset R$ is *closed* in R if $A_{R,C}(F \cup \pi(R')) = R'$. In other words, if a RAF $R' \subset R$ is closed, there are no reactions in $R \setminus R'$ for which all reactants and at least one catalyst are either the product of some reaction within R' or are in the food set. Note that both the maxRAF and a CAF in a given reaction network R are (by definition) a closed RAF, or cRAF. However, a closed RAF may contain smaller closed RAFs as subsets, besides the CAF (if one is present).

Although the maxRAF in the example in Figure 1 does not contain a CAF, it does contain one smaller cRAF. The minimal RAF (iRAF) $R_2 = \{r_1, r_2\}$ is also a closed RAF (cRAF). Given R_2, neither r_3 nor r_4 can happen catalyzed. However, the subRAF $R_1 = \{r_1, r_2, r_4\}$ is not closed, as r_3 can now also happen catalyzed.

Dynamically, for a given closed RAF (say R_2) to "expand" into its containing closed RAF (in this case the maxRAF R'), one or more reactions will need to happen spontaneously first. In the example above, given R_2, reaction r_4 will need to happen spontaneously, at which point the full maxRAF R' can come into existence.

Earlier it was shown that closed RAFs are the equivalent of chemical organizations [18]. As such, cRAFs are the dynamically stable subsets of a maxRAF. However, since there can be exponentially many organizations in a given reaction network, finding all of them is, in principle, intractable [19,26].

2.5. The Binary Polymer Model

A computational model of reaction networks based on polymer chemistry was introduced by Kauffman [2] to argue about the conditions under which autocatalytic sets emerge. In this model, the molecules are represented by bit strings, i.e., strings of 0s and 1s, up to a maximum length n. The food molecules are all bit string up to a small length t. The reactions consist of a *ligation* (gluing two bit strings together into a longer one) and corresponding *cleavage* (cutting a bit string into two smaller parts), with the constraint that no bit strings longer than n can be produced. Finally, catalysis is assigned randomly, with a fixed probability p that any given bit string can catalyze any given (bi-directional) ligation-cleavage reaction. Formally, an instance of the binary polymer model forms a CRN $Q = \{X, R, C, F\}$, where:

 doi:10.20944/preprints202604.1787.v1

- $X = \{0,1\}^{\leq n}$
- $R = \{x_i + x_j \leftrightarrow x_i \cdot x_j : (x_i, x_j \in X) \wedge (|x_i \cdot x_j| \leq n)\}$
- $\mathbb{P}[(x \in X, r \in R) \in C] = p$
- $F = \{0,1\}^{\leq t}$

The emergence and structure of autocatalytic sets have been studied in various computational models of chemical reaction networks, including the binary polymer model [2,4,7,27–32]. However, such sets have also been constructed in real experimental systems in the lab [33–38]. Moreover, they have been shown to exist in the metabolic networks of prokaryotes [9,10,12].

The example in Figure 1 is small and simple enough to find the various RAF sets by eye (or determine their absence). However, for larger systems (such as metabolic networks) this is clearly not convenient. Fortunately, though, there are *efficient algorithms* for detecting various RAF sets in arbitrary chemical reaction networks.

3. RAF Algorithms

3.1. The Main RAF Algorithm

Given a CRN $Q = \{X, R, C, F\}$, an efficient (polynomial-time) algorithm exists for deciding whether Q contains a RAF set or not. This algorithm was introduced in earlier work [6,13], but often in a rather descriptive way. However, the subRAF enumeration algorithms presented below depend crucially on the main RAF algorithm. Therefore, it is presented formally in Algorithm 2 below. The RAF algorithm itself, though, depends on computing the closure $\mathrm{cl}_R(F)$ of the food set F relative to the reaction set R. This closure computation is presented first, in Algorithm 1.

Algorithm 1 ComputeClosure (F, R)

```
W = F
molsAdded = true
while (molsAdded) do
  molsAdded = false
  for all (r ∈ R) do
    if ((ρ(r) ⊆ W) ∧ (π(r) ⊄ W)) then
      W = W ∪ π(r)
      molsAdded = true
    end if
  end for
end while
Return W
```

As noted above, in case of a bidirectional reaction, the reverse direction (i.e., with the roles of $\rho(r)$ and $\pi(r)$ swapped) also needs to be checked. Computing the closure of the food set is computationally the most expensive step in the main RAF algorithm, which roughly works as follows. Starting with the full set of reactions $R' = R$, the RAF algorithm repeatedly computes the closure of the food set relative to the current reaction set R', and then removes from R' all reactions that do not have all of their reactants and at least one catalyst in this closure. This is repeated until no more reactions can be removed.

If upon termination of the algorithm R' is non-empty, then R' is the unique *maximal RAF* (or maxRAF) contained in Q, i.e., the union of all possible RAF subsets in R. This may be one connected RAF set, or consist of more than one disconnected subsets. If R' is empty, Q does not contain a RAF set.

A straightforward computational complexity analysis of the RAF algorithm gives a worst-case running time of $\mathcal{O}(|X||R|^3)$. However, the efficiency of the algorithms as presented in Algorithm 2 can still be improved on. With some additional book-keeping, such as keeping track of which reactions have already been applied during the closure computation, the worst-case running time of the overall algorithm can be reduced somewhat. In practice, the *average* running time on random instances of the binary polymer model turns out to be sub-quadratic [6].

 doi:10.20944/preprints202604.1787.v1

Algorithm 2 RAF (X, R, C, F)

$R' = R$
`reacsRemoved = true`
while (`reacsRemoved`) **do**
  `reacsRemoved = false`
  `clF` = ComputeClosure (F, R')
  for all $(r \in R')$ **do**
    if $((\nexists x \in$ `clF` $: (x, r) \in C) \vee (\rho(r) \not\subset$ `clF`$))$ **then**
      $R' = R' \setminus \{r\}$
      `reacsRemoved = true`
    end if
  end for
end while
Return R'

3.2. The CAF Algorithm

The closure computation algorithm above can easily be adjusted to find CAF sets, as presented in Algorithm 3. In this case, rather than just checking whether all of a reaction's reactants are already included in W, it also needs to be checked whether at least one catalyst is included in W. In addition, it is necessary to keep track of which reactions were used to expand the set W.

Algorithm 3 CAF (X, R, C, F)

$S = \emptyset$
$W = F$
`reacsAdded = true`
while (`reacsAdded`) **do**
  `reacsAdded = false`
  for all $(r \in R)$ **do**
    if $((\rho(r) \subset W) \wedge (\exists x \in W : (x, r) \in C) \wedge (r \notin S))$ **then**
      $W = W \cup \pi(r)$
      $S = S \cup \{r\}$
      `reacsAdded = true`
    end if
  end for
end while
Return S

The set S returned by the algorithm, if non-empty, is the unique (maximal) CAF set contained in the CRN Q. If S is empty, there is no CAF set.

3.3. Irreducible RAF Algorithms

Previously, an outline for a method to find *all* iRAFs was presented but not yet implemented [15,39]. The main idea behind this method is based on considering the Cartesian product

$$V = R_1 \times R_2 \times \cdots \times R_n = \{r_1, r_2, \ldots, r_n : R_1 \in R_1, r_2 \in R_2, \ldots, r_n \in R_n\}$$

of all iRAFs $R_i, i = 1, \ldots, n$ found so far in a given RAF R. If for all $v \in V$ the set $R \setminus v$ does not contain a non-empty RAF set, then all iRAFs have been found. Otherwise, if there is at least one $v \in V$ for which the set $R \setminus v$ contains a non-empty RAF subset, a new iRAF R_{n+1} can be found within that subset, and the procedure is repeated. The method is presented formally in Algorithm 4. Besides from the main RAF algorithm, it also depends on a procedure rndiRAF for finding *some* (random) iRAF in a given RAF, which will be presented formally below.

 doi:10.20944/preprints202604.1787.v1

Algorithm 4 iRAFs (X, R, C, F)

R_1 = rndiRAF (X, R, C, F)
$I = \{R_1\}$
$n = 1$
`allFound = false`
while (`!allFound`) **do**
  `allFound = true`
  for all $(v = \{r_1, \ldots, r_n\} : R_i \in R_i, i = 1, \ldots, n)$ **do**
    R' = RAF $(X, R \setminus v, C, F)$
    if $(R' \neq \emptyset)$ **then**
      $n = n + 1$
      R_n = rndiRAF (X, R', C, F)
      $I = I \cup R_n$
      `allFound = false`
      `break`
    end if
  end for
end while
Return I

Upon return, the set I will contain all iRAFs within a given RAF R. However, the Cartesian product V is likely to become prohibitively large in practice for any instance with more than just a few relatively small iRAFs.

Finding a *random* iRAF within a given RAF R, though, can be done efficiently by repeatedly removing a random reaction and then calling the main RAF algorithm on the reduced reaction set, until a minimal RAF is found. This method, presented formally in Algorithm 5, can thus be used to produce a random *sample* of iRAFs within a given RAF R by calling it a given number of times, and saving the unique iRAFs found within that sample.

Algorithm 5 rndiRAF (X, R, C, F)

shuffle(R) # randomly shuffle the reactions in R
`index` $= 1$
while (`index` $\leq |R|$) **do**
  $r = R$[`index`]
  R' = RAF $(X, R \setminus \{r\}, C, F)$
  if $(R' == \emptyset)$ **then**
    `index` $=$ `index` $+ 1$
  else
    $R = R'$
  end if
end while
Return R

This random method can be applied to maxRAFs of any size and with any number of iRAFs. Of course it will not be known whether *all* iRAFs are found within a given sample, or how many more still might be found with a larger sample size. However, it can provide a useful statistical insight into the sizes and structures of iRAFs also within larger RAFs.

3.4. Closed RAF Algorithms

Earlier it was shown that a closed RAF is the equivalent of a chemical organization within the maxRAF, and that the closed RAFs within a maxRAF can thus be obtained by computing the organizations within that maxRAF [18]. In principle, though, computing *all* organizations within a given chemical reaction network is intractable. However, a decomposition theorem for organizations suggests a more efficient algorithm [40], which was recently made available in an implementation

 doi:10.20944/preprints202604.1787.v1

called `pyCOT` (https://github.com/tveloz/pyCOT). This implementation can thus be used to obtain all closed RAFs within a maxRAF, at least for moderately sized RAFs.

Note, though, that the reverse is not always true. An organization within a maxRAF does not always correspond to a closed RAF. In particular, if a maxRAF does not contain a CAF then there will be an organization consisting of just the food set F, but there are no RAF reactions associated with it. Therefore, such an organization would represent an "empty" closed RAF. In other words, there can be more organizations than closed RAFs, but these additional organizations can be easily checked to see if they consist of just the food set and thus be discarded.

Alternatively, the algorithm for finding a random iRAF can be easily adapted to find random cRAFs. However, whereas an iRAF cannot contain a smaller iRAF, a cRAF can contain one or more smaller cRAFs. Therefore, the method for finding random cRAFs, presented formally in Algorithm 6, may return any number of (nested) cRAFs. If the algorithm would stop as soon as the first closed RAF is found, its closed subRAFs (if any) would never be seen. However, note that it may also return none, even if there are cRAFs present.

Algorithm 6 rndcRAF (X, R, C, F)

```
shuffle(R)                    # randomly shuffle the reactions in R
S = ∅
index = 1
while (index ≤ |R|) do
  r = R[index]
  R' = RAF (X, R \ {r}, C, F)
  if (R' == ∅) then
    index = index + 1
  else
    if closed(R') then
      S = S ∪ R'
    end if
    R = R'
  end if
end while
Return S
```

All algorithms presented in this section are implemented in a C++ program that is available as a GitHub repository at https://github.com/wimhor/RAF. It is based on an earlier implementation that was primarily written for research purposes and therefore not suitable for public release. However, the current implementation is a cleaned-up version with several new methods added (in particular for finding all iRAFs and for sampling cRAFs). It also includes a program for generating random instances of the binary polymer model. The next section presents results of testing and comparing these various RAF algorithms on the binary polymer model.

4. Results

4.1. Probability of RAF Sets

First, the implementation of the main RAF algorithm is tested (Algorithm 2). For each combination of a given maximum polymer length $n \in \{4, 5, \ldots, 12\}$ and a level of catalysis $f \in \{0.5, 0.6, \ldots, 2.0\}$, 1000 instances of the binary polymer model were generated using a uniform catalysis distribution and with $t = 2$ and $p = f/|R|$. The main RAF algorithm was then applied to each of these instances, and the probability P of a RAF is calculated as the fraction of instances that contain a RAF set. "Trivial" RAF sets consisting of one or two reactions that happen, by chance, to have all their reactants and at least one catalyst in the food set are discarded. In other words, only RAF sets R' with $|R'| > 2$ were counted.

Although the RAF algorithm is efficient (polynomial time) in the size of the reaction network $|R|$, this size grows exponentially with increasing maximum polymer length n in the binary polymer model.

 doi:10.20944/preprints202604.1787.v1

Therefore, the program was run on a high performance computing cluster so all 16 values for f could be run in parallel for each value of n. The results are shown in Figure 2. An interactive version that can be rotated and zoomed is available at https://worldwidewanderings.net/General/RAF/Pn.html.

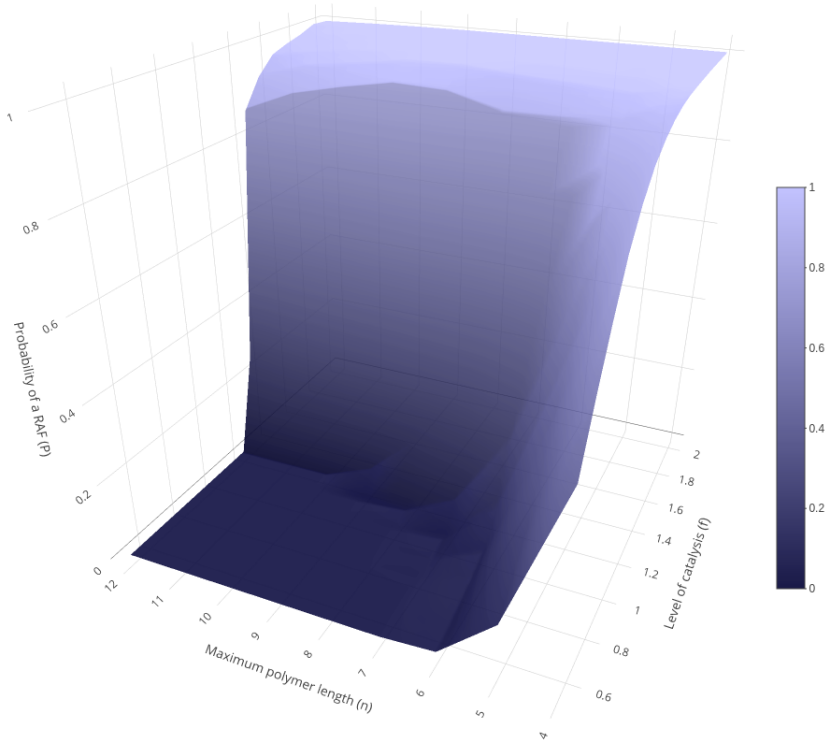

Figure 2. The probability P of finding RAF sets in random instances of the binary polymer model with various values for n and f. See https://worldwidewanderings.net/General/RAF/Pn.html for an interactive version.

These results agree very closely with the original results of Hordijk and Steel [6], which also have been confirmed with other independent (but not publicly available) implementations [16,41]. Most notably, the transition from no RAFs ($P = 0$) to almost always RAFs ($P \sim 1$) becomes steeper with increasing n, and the level of catalysis f where this transition happens growths slowly (linearly) with increasing n. Having thus verified the implementation of the main RAF algorithm, enumerating subRAFs is considered next.

4.2. Enumerating iRAFs

To test the implementation of the algorithms for enumerating irreducible RAFs, 100 instances of the binary polymer model with $n = 5$, $t = 2$, and $p = 0.0055$ were created. For each of these instances, a random sample of size 10000 of iRAFs was then generated (using Algorithm 5). Furthermore, on each instance the Cartesian product method for finding *all* iRAFs was applied (Algorithm 4). This was also run on a high performance computing cluster, allowing a maximum of 48 hours of compute time for each instance. The implementation of this method writes out iRAFs as it finds them, so even if the program does not finish within the allowed amount of time, at least some number of iRAFs will have been found.

Figure 3 shows the number of iRAFs found in the random sample of size 10000 against the number found by the Cartesian product ("all") method after 48hrs. The diagonal line indicates where they would be equal. For readability, the dots are displaced by a small random amount along the horizontal axis (otherwise multiple dots would overlap).

 doi:10.20944/preprints202604.1787.v1

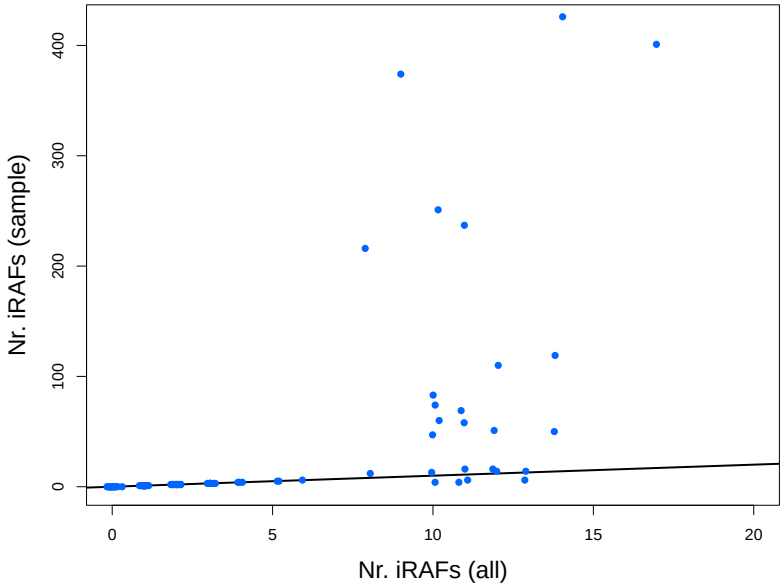

Figure 3. The number of iRAFs found in a random sample of size 10000 vs. the number found with the "find all" method.

There is a clear distinction between maxRAFs with only a small number of iRAFs (up to six), and those with a larger number. With up to six iRAFs, the "find all" method is able to indeed find all of them within the given amount of compute time, but the random sample always finds all of them as well.

In contrast, for maxRAFs with a larger number of iRAFs, the "find all" method never finds more than 20 iRAFs within the allowed 48hrs of compute time, whereas the random sample (which takes only a few minutes) often finds many more. Interestingly, though, there are four cases where the random sample found fewer iRAFs. It turns out that these are maxRAFs that contain multiple "trivial" iRAFs of just one reaction. Due to the way the random sample is performed (i.e., removing random reactions from the RAF), small iRAFs are much more likely to be found than larger ones, which may actually be missed entirely.

To check the impact of the sample size, smaller random samples of sizes 100 and 1000 were also generated on the same 100 instances of the binary polymer model. Figure 4 shows the number of iRAFs found in each of these samples against the size of the maxRAF they are part of.

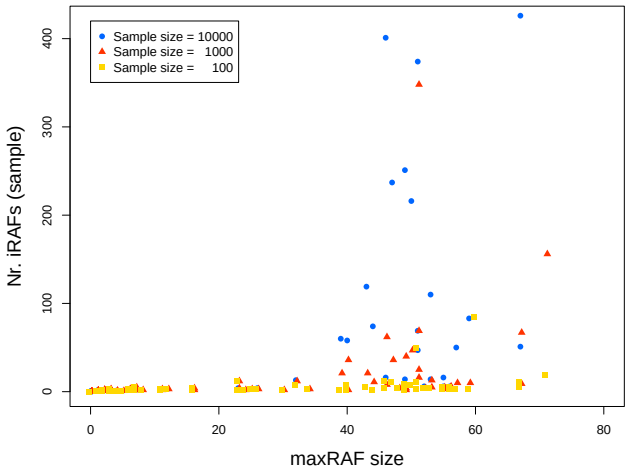

Figure 4. The number of iRAFs found in random samples of various sizes vs. the size of the maxRAF they are part of.

 doi:10.20944/preprints202604.1787.v1

There is also a clear distinction here, in this case between relatively small maxRAFs (less than 40 reactions) and larger ones. With up to 40 reactions in the maxRAF, the number of iRAFs is quite small, generally less than 10. However, for maxRAFs with 40 or more reactions there can be any number of iRAFs, up to several hundreds. Furthermore, as can be expected, larger samples tend to find a larger number of distinct iRAFs.

4.3. Enumerating cRAFs

To test the implementation of the algorithms for enumerating closed RAFs, the same 100 instances of the binary polymer model as in the previous subsection were used. For each of these instances, a random sample of size 10000 of cRAFs was generated (using Algorithm 6). Furthermore, for each instance all organizations within the resulting maxRAF were computed using the `pyCOT` program. This was again run on a high performance computing cluster, also allowing a maximum of 48 hours of compute time for each instance. Since `pyCOT` returns the results only at the very end when the program finishes, there are no results if the program does not finish within the allowed amount of time.

Figure 5 shows the number of cRAFs found in the random sample of size 10000 against the number of organizations found by the `pyCOT` program after 48hrs. The diagonal line indicates where they would be equal. For readability, the dots are again displaced by a small random amount along the horizontal axis.

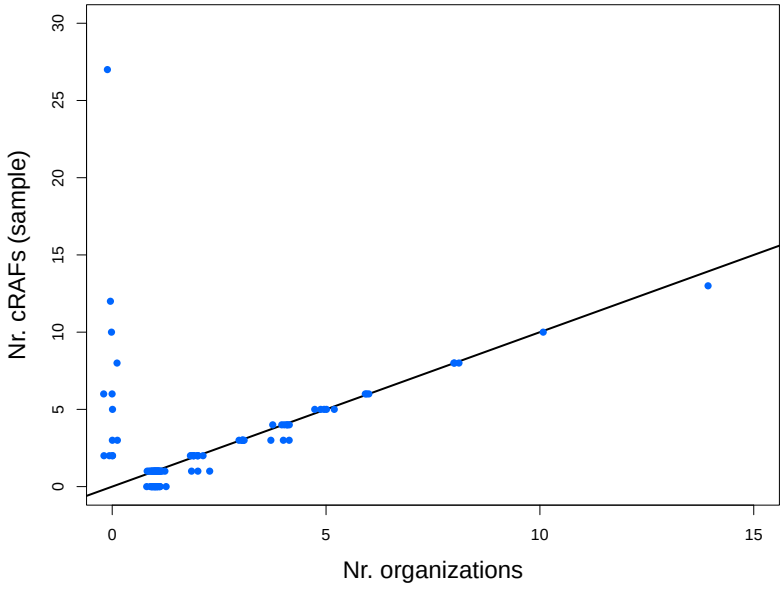

Figure 5. The number of cRAFs found in a random sample of size 10000 vs. the number of organizations found with the `pyCOT` program.

First note that the `pyCOT` program did not finish on 13 of the 100 instances, even after 48hrs. These instances are represented by the dots in the column on the left of the figure at zero organizations (i.e., no result). However, the random samples found anywhere from just a few to close to 30 cRAFs in these cases. Moreover, the sampling method only took several minutes to complete.

In those cases where the `pyCOT` program did return a result, it never found more than 15 organizations. The random sampling method found either the same number of cRAFs (blue dots on the diagonal), or one less (blue dots just below the diagonal). As mentioned earlier, the latter are cases where the maxRAF does not contain a CAF, resulting in an additional organization consisting of just the food set that does not correspond to a cRAF.

To check the impact of the sample size, smaller random samples of sizes 100 and 1000 were again generated on the same 100 instances of the binary polymer model. Figure 6 shows the number of cRAFs found in each of these samples against the size of the maxRAF they are part of.

 doi:10.20944/preprints202604.1787.v1

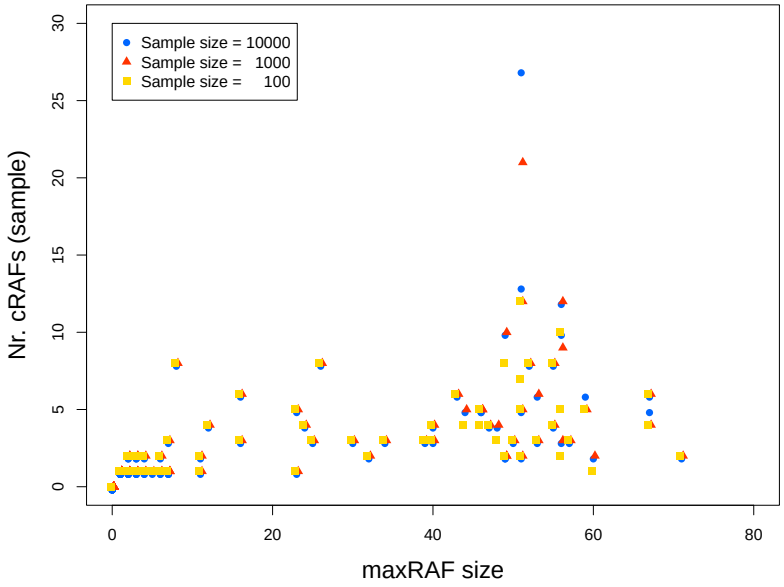

Figure 6. The number of cRAFs found in random samples of various sizes vs. the size of the maxRAF they are part of.

For cRAFs there is not such a clear distinction between relatively small maxRAFs and larger ones as there is for iRAFs. For the majority of maxRAFs, the number of cRAFs is at most 10 regardless of the size of the maxRAF. There are only a few exceptions. Furthermore, the sample size does not seem to make a significant difference, besides from those same few exceptions.

5. Conclusions

This article has provided a concise but formal description of the various RAF definitions and algorithms. Although several of these were already introduced in earlier work, they have become somewhat scattered throughout the literature over the years, and were sometimes presented in a rather descriptive way rather than more formally.

An implementation of all RAF algorithms presented here in a C++ program is available as a GitHub repository (https://github.com/wimhor/RAF). This repository is still a work in progress, and will be improved and expanded over time. But as the results in this article show, the implementations work as intended, and although only applied to the binary polymer model here, are suitable for more general use. For example, the existence, distribution, and structure of subRAFs such as iRAFs and cRAFs in metabolic networks could be investigated, something that so far has not been done.

The importance of such subRAFs lies in the fact that their existence makes autocatalytic sets *evolvable*. Interestingly, in the very same year that the concept of autocatalytic sets was originally introduced [1], a very similar idea was proposed independently by Nobel laureate Manfred Eigen, also in the context of proteins [42]. Eigen then immediately discards the idea again, claiming such autocatalytic networks of proteins are not evolvable because they lack genetic specification. However, as Vasas et al. [20] conclude, if a reaction network contains many "viable cores" (roughly the equivalent of iRAFs), it can be evolvable through a form of "compositional inheritance" rather than genetic inheritance. This form of compositional inheritance has also been studied in related models [28,32].

The results presented here show that, in general, there can indeed be many such iRAFs, up to hundres even in moderately-sized maxRAFs. As expected, though, enumerating *all* iRAFs or cRAFs in a given maxRAF is, in practice, limited to instances with relatively few such subRAFs of relatively small sizes. However, randomly sampling either iRAFs or cRAFs can be done very efficiently, even for large maxRAFs and large sample sizes, and produces meaningful results. A more detailed investigation of the sizes and structures of these subRAFs is beyond the scope of the current article, and will be left for future work.

 doi:10.20944/preprints202604.1787.v1

Funding: This work was partially funded by the MATOMIC project (https://www.sdu.dk/en/forskning/matomic).

Acknowledgments: Thanks go to Mike Steel (University of Canterbury, Christchurch, New Zealand) for helpful comments and discussions, Tomas Veloz (Vrije Universiteit Brussels, Belgium) for his invitation to participate in a workshop which led to the use of `pyCOT` here, Peter Stadler for hosting the author's research visit in the Bioinformatics Group of the University of Leipzig, Germany, and SURF (www.surf.nl) for the support in using the Dutch national supercomputer Snellius.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Kauffman, S.A. Cellular homeostasis, epigenesis and replication in randomly aggregated macromolecular systems. *Journal of Cybernetics* **1971**, *1*, 71–96.
2. Kauffman, S.A. Autocatalytic sets of proteins. *Journal of Theoretical Biology* **1986**, *119*, 1–24.
3. Kauffman, S.A. *The Origins of Order*; Oxford University Press: Oxford, 1993.
4. Hordijk, W. A history of autocatalytic sets. *Biological Theory* **2019**, *14*, 224–246.
5. Steel, M. The emergence of a self-catalysing structure in abstract origin-of-life models. *Applied Mathematics Letters* **2000**, *3*, 91–95.
6. Hordijk, W.; Steel, M. Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *Journal of Theoretical Biology* **2004**, *227*, 451–461.
7. Hordijk, W.; Steel, M. Chasing the tail: The emergence of autocatalytic networks. *BioSystems* **2017**, *152*, 1–10.
8. Hordijk, W.; Hein, J.; Steel, M. Autocatalytic sets and the origin of life. *Entropy* **2010**, *12*, 1733–1742.
9. Sousa, F.L.; Hordijk, W.; Steel, M.; Martin, W.F. Autocatalytic sets in *E. coli* metabolism. *Journal of Systems Chemistry* **2015**, *6*, 4.
10. Xavier, J.C.; Hordijk, W.; Kauffman, S.A.; Steel, M.; Martin, W.F. Autocatalytic chemical networks at the origin of metabolism. *Proceedings of the Royal Society B* **2020**, *287*, 20192377.
11. Hordijk, W.; Steel, M.; Kauffman, S. Autocatalytic sets arising in a combinatorial model of chemical evolution. *Life* **2022**, *12*, 1703.
12. Xavier, J.C.; Kauffman, S.A. Small-molecule autocatalytic networks are universal metabolic fossils. *Philosophical Transactions of the Royal Society A* **2022**, *380*, 20210244.
13. Hordijk, W.; Smith, J.I.; Steel, M. Algorithms for detecting and analysing autocatalytic sets. *Algorithms for Molecular Biology* **2015**, *10*, 15.
14. Hordijk, W.; Steel, M.; Kauffman, S. The structure of autocatalytic sets: Evolvability, enablement, and emergence. *Acta Biotheoretica* **2012**, *60*, 379–392.
15. Steel, M.; Hordijk, W.; Smith, J. Minimal autocatalytic networks. *Journal of Theoretical Biology* **2013**, *332*, 96–107.
16. Hordijk, W.; Hasenclever, L.; Gao, J.; Mincheva, D.; Hein, J. An investigation into irreducible autocatalytic sets and power law distributed catalysis. *Natural Computing* **2014**, *13*, 287–296.
17. Smith, J.I.; Steel, M.; Hordijk, W. Autocatalytic sets in a partitioned biochemical network. *Journal of Systems Chemistry* **2014**, *5*, 2.
18. Hordijk, W.; Steel, M.; Dittrich, P. Autocatalytic sets and chemical organizations: Modeling self-sustaining reaction networks at the origin of life. *New Journal of Physics* **2018**, *20*, 015011.
19. Dittrich, P.; Speroni di Fenizio, P. Chemical Organization Theory. *Bulletin of Mathematical Biology* **2007**, *69*, 1199–1231.
20. Vasas, V.; Fernando, C.; Santos, M.; Kauffman, S.; Sathmáry, E. Evolution before genes. *Biology Direct* **2012**, *7*, 1.
21. Hordijk, W.; Steel, M. Conditions for evolvability of autocatalytic sets: A formal example and analysis. *Origins of Life and Evolution of Biospheres* **2014**, *44*, 111–124.
22. Hordijk, W.; Naylor, J.; Krasnogor, N.; Fellermann, H. Population dynamics of autocatalytic sets in a compartmentalized spatial world. *Life* **2018**, *8*, 33.
23. Ameta, S.; Arsène, S.; Foulon, S.; Saudemont, B.; Clifton, B.E.; Griffiths, A.D.; Nghe, P. Darwinian properties and their trade-offs in autocatalytic RNA reactionnetworks. *Nature Communications* **2021**, *12*, 842.
24. Temkin, O.N.; Zeigarnik, A.V.; Bonchev, D. *Chemical Reaction Networks: A Graph-Theoretical Approach*; CRC Press: Boca Raton, FL, USA, 1996.

 doi:10.20944/preprints202604.1787.v1

25. Mossel, E.; Steel, M. Random biochemical networks: The probability of self-sustaining autocatalysis. *Journal of Theoretical Biology* **2005**, *233*, 327–336.
26. Heylighen, F.; Beigi, S.; Veloz, T. Chemical organization theory as a general modeling framework for self-sustaining systems. *Systems* **2024**, *12*, 111.
27. Farmer, J.D.; Kauffman, S.A.; Packard, N.H. Autocatalytic replication of polymers. *Physica D* **1986**, *22*, 50–67.
28. Serra, R.; Villani, M. *Modelling Protocells*; Springer: Berlin, 2017.
29. Serra, R.; Villani, M. Template-based catalysis and the emergence of collectively autocatalytic systems. *Entropy* **2026**, *28*, 184.
30. Giri, V.; Jain, S. The origin of large molecules in primordial autocatalytic reaction networks. *PLoS ONE* **2012**, *7*, e29546.
31. Fellermann, H.; Tanaka, S.; Rasmussen, S. Sequence selection by dynamical symmetry breaking in an autocatalytic binary polymer model. *Physical Review E* **2017**, *96*, 062407.
32. Lancet, D.; Zidovetzki, R.; Markovitch, O. Systems protobiology: Origin of life in lipid catalytic networks. *Journal of the Royal Society Interface* **2018**, *15*, 20180159.
33. Sievers, D.; von Kiedrowski, G. Self-replication of complementary nucleotide-based oligomers. *Nature* **1994**, *369*, 221–224.
34. Kim, D.E.; Joyce, G.F. Cross-catalytic replication of an RNA ligase ribozyme. *Chemistry & Biology* **2004**, *11*, 1505–1512.
35. Lincoln, T.A.; Joyce, G.E. Self-sustained replication of an RNA enzyme. *Science* **2009**, *323*, 1229–1232.
36. Ashkenasy, G.; Jegasia, R.; Yadav, M.; Ghadiri, M.R. Design of a directed molecular network. *PNAS* **2004**, *101*, 10872–10877.
37. Vaidya, N.; Manapat, M.L.; Chen, I.A.; Xulvi-Brunet, R.; Hayden, E.J.; Lehman, N. Spontaneous network formation among cooperative RNA replicators. *Nature* **2012**, *491*, 72–77.
38. Arsène, S.; Ameta, S.; Lehman, N.; Griffiths, A.D.; Nghe, P. Coupled catabolism and anabolism in autocatalytic RNA sets. *Nucleic Acids Research* **2018**, *46*, 9660–9666.
39. Huson, D.; Xavier, J.C.; Steel, M. Self-generating autocatalytic networks: Structural results, algorithms and their relevance to early biochemistry. *Journal of the Royal Society Interface* **2024**, *21*, 20230732.
40. Veloz, T.; Reynaert, B.; Rojas-Camaggi, D.; Dittrich, P. A decomposition theorem in chemical organizations. In Proceedings of the 11th European Conference on Artificial Life, ECAL 2011. MIT Press Journals, 2011.
41. Filisetti, A.; Villani, M.; Damiani, C.; Graudenzi, A.; Roli, A.; Hordijk, W.; Serra, R. On RAF sets and autocatalytic cycles in random reaction networks. *Communications in Computer and Information Science* **2014**, *445*, 113–126.
42. Eigen, M. Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften* **1971**, *58*, 465–523.