Enumeration Specification: Version 4

Brian Wolfe (wolfe@caltech.edu)

September 26, 2009

This document describes the enumerable reactions of a domain-based branch migration model and a method to efficiently explore and enumerate a useful subset of the space defined by these reactions. There are four basic reactions, with two subtypes of each reaction. These reactions are bind (1-1 and 2-1), open (1-1 and 1-2), 3-way branch migration (1-1 and 1-2), and 4-way branch migration (1-1 and 1-2). Where 1-1 indicates one molecule's self-reaction. 1-2 indicates a molecule splitting into 2 molecules as a result of a self reaction and 2-1 indicates two molecules reacting to become 1.

By assuming time-scale separation between bimolecular and unimolecular reactions the space of reactions can be quickly explored for all finite reaction graphs, an enumeration method is shown at the end of this document.

1 Description of the Terms

What follows is a list of the terms that will be used throughout the rest of this document to describe the complexes and reactions.

1.1 Domain

A domain is a tuple (s, l) where s is a label which specifies binding between domains and l is an integer specifying the length of the domain. Pairing can only occur between complementary domains. These are specified like s^*, the complement domain to the domain s. For most of this discussion, domains will be represented merely by a label, ignoring the length. If the length of a domain a is used it will be represented on $length(a)$. Note there is one special domain for the purpose of discussion, $(+, 0)$ which indicates a strand break. Strand breaks have no complement.

1.2 Sequence

Sequence: A sequence is an ordered list of domains. Two sequences $seq1$ and $seq2$ are equal if and only if the length of the sequences are equal and there exists a circular offset of $seq1$ s.t. $seq1[i] = seq2[i]$ for all $0 \leq i < n$ where n is the number of domains in seq1 and seq2.

e.g. sequence S = [a,b,c,b*,a*,c,+] is equal to sequence B = [b*,a*,c,+,a,b,c]

These sequences represent a single DNA strand consisting of 7 domains with 2 complementary sets of domains (the complement of c is not present) and one strand break. The canonical form is the form of S, where the strand break is placed at the end. If there are multiple strand breaks then there are multiple canonical forms, these can be assigned an order in an arbitrary manner.

1.3 Structure

Structure: A list of integers indicating the domain pairing in a complex, the symbol used here for the structure is T. The special symbol ∅ is used to indicate an unpaired domain. The special symbol + is used to indicate a strand break. The list of integers is equivalent to a dot-paren notation representation of pairing, the integers are used to make descriptions of the reactions more precise and to match with implementation details.

Otherwise, T[i] (the ith component of structure T) is the index of the domain which is bound to the ith domain

e.g. structure $T = [\emptyset, 3, \emptyset, 1, \emptyset, \emptyset, +]$

This structure is one of the valid structures for S given above. The domain at position 1 is paired to the domain at position 3 (0 based index). The final domain is a strand break.

1.4 Complex

Complex: A tuple (S, T) of a valid sequence structure pair. A valid complex must have valid pairing, be unpseudoknotted and be connected.

Should make these sections more exact.

1.4.1 Valid Pairing

A complex is only valid if all pairings are between complementary domains and pairings are symmetric, that is $T[i] = j \Rightarrow S[i] = S[j]^*$ and $T[j] = i$.

1.4.2 Unpseudoknotted

Complexes must be free of pseudoknots, that is, all paired domains must be well nested.

1.4.3 Connectedness

Each domain must be connected to another domain through a combination of pairings and adjacency of domains without strand breaks.

1.4.4 Rotation

A rotation of a complex is a circular offset of the domain and structure lists such that all pairings are preserved. (should make this statement more exact)

1.4.5 Equality

Two complexes are equal if and only if there is a rotation such that the structure and sequence are exactly identical.

2 Reaction Types

2.1 Bind

2.1.1 1-1 Binding

A 1-1 binding reaction is a reaction between two complementary, unpaired domains within a single complex that produces an unpseudoknotted product complex. The reaction is described below and shown in Fig. 1.

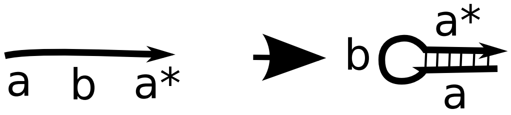

Figure 1: A 1-1 binding reaction, a single complex binds to itself.

$\forall$ complexes $C = (S, T)$ $\forall 0 \leq i, j < length(S)$ if:

$S[i] = S[j]^*$ Complementary

$T[i] = \emptyset$ Unpaired

$T[j] = \emptyset$ Unpaired

and

$\forall i < k < j$: No Pseudoknot Creation

$i < T[k] < j$

or $T[k] = +$

or $T[k] = \emptyset$

Then $\exists$ 1-1 binding reaction $R := [C] \rightarrow [C']$ where $C' = (S, T')$

$T'[n] = T[n] \forall n \neq i, j.$

$T'[i] = j, T'[j] = i.$

Connectedness is guaranteed as long as the original complex was connected.

2.1.2 2-1 Binding

$\forall$ pairs of complexes C_1 and C_2, generate all circular rotations of C_1 and C_2 that end in a strand break. Generate $C_{tot} = C_{1,i}, C_{2,j} = ([S_{1,i}, S_{2,j}], [T_{1,i}, T_{2,j}]) =$ the concatenation of the ith rotation of C_1 and the jth rotation of C_2.

If there exists a binding reaction (as defined above) that results in a connected complex, then there exists a bimolecular reaction $R := [C_1, C_2] \rightarrow [C'_{tot}]$,

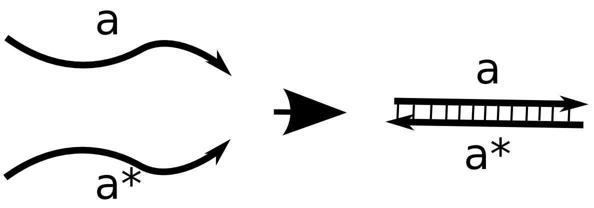

Figure 2: This shows the 2-1 binding reaction. Two complexes (in this case only single strands), come together to form a single complex by binding at unpaired domains.

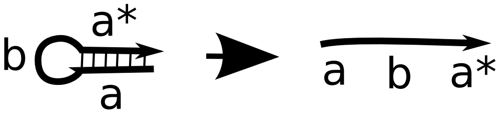

Figure 3: One to one open reaction. Note that this assumes that the length of the opening helix is below the threshold

where C'_{tot} is defined the same way as C' is above. In the current version, this is the only reaction which is defined to be slow. The source complexes for this type of reaction must be end-states, that is, they must be in a strongly connected component that has no outward leading "fast" edges. This is described further in the implementation section.

2.2 Open

Open1-1 reactions are present when a series of adjacent, paired domains with a total length less than a threshold (labelled Threshold11) completely dissociates and the result is still a connected complex. An example is shown in Fig. 3. Open1-2 reactions are present when a series of adjacent, paired domains with a total length less than a different threshold (labelled Threshold12) dissociates, leaving two distinct complexes. Note that these definitions are really only going to be valid/useful if $Threshold11 \geq Threshold12$. This is physically relevant because the entropy gain from releasing a strand is greater than the entropy gain from releasing a few domains, but retaining the connectedness. An example of this type of reaction is given in Fig. 4.

```
For each $i \in 0..n-1 : T[i] \neq \emptyset$:
    If $T[i-1] \neq T[i]+1$
    and $\exists k$ s.t. $T[i+k+1] \neq T[i+k]-1$ (this is the whole helix)
    and $\forall 1 \leq j \leq k$: $T[i+j] = T[i+j-1]+1$ (the helix is paired)
    and $\sum_{j=0}^{k} length(T[i+j]) < Threshold11$ (the helix is short enough to open)
```

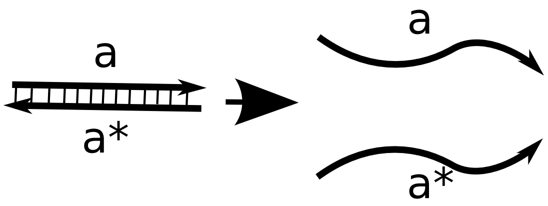

Figure 4: 1-2 open reaction. Note that this again assumes that the helix is below the threshold for opening reactions (which can be different from 1-1 open thresholds). Note that this can actually involve multiple domains as long as the total helix length is less than the threshold.

Define T' as $T'[m] = \begin{cases} \emptyset & : i \le m \le i+k \\ T[m] & : otherwise \end{cases}$

If T' is connected, then $R = [(S,T)] \rightarrow [(S,T')]$ is a valid Open1-1 reaction.

If all the above constraints hold except that the total length of the domains is less than the constant $Threshold12$ and T' is disconnected, then $R = [(S,T)] \rightarrow [(S_1,T_1'),(S_2,T_2')]$ is a valid Open1-2 reaction. Where S_1, S_2, T_1', T_2' are defined by the split of the disconnected complex (S,T'). (should show the uniqueness of the split operation for these cases)

The open reactions could be unified better with the rates by removing the threshold constraint and creating a rate function which calculates the reaction rate based on length. This cutoff could then be unified with the fast-slow cutoff. This would result in three time scales, fast reactions, slow reactions, and reactions so slow that they are ignored. This would be worth implementing, and I shall attempt to provide notes for how I imagine it should be implemented.

2.3 3-way Branch Migration

A three-way branch migration is a displacement reaction where an unpaired string of domains replaces an adjacent double-stranded region which is fully complementary. Example of this type of reaction given in Fig. 5.

NOTE: In the current version of the code, all displacements occur 1 domain at a time. In the future, consider whether displacements should also be able to displace across nicked multiloops.

NOTE2: In the current version of the code 3-way and 4-way branch migration are evaluated in the same location (they are very similar) this may cause problems with later extensions. I documented the input-output of these reactions so that they could be rewritten in that case.

Where $C = (S,T)$ is a connected complex:

$\forall i, j, k$ s.t.:

$T[i] = j$ (There is a pair to displace)

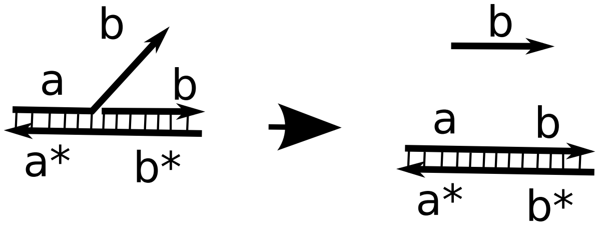

Figure 5: Three way branch migration example. In this example, the result is disconnected and so this is a 1-2 type reaction. Note also, unlike binding and open reactions, this reaction can be irreversible. The initiation rate of binding to the closed helix is assumed to be negligible at relevant time-scales. It can also be reversible if the displacing and displaced strand remain bound.

$S[i] = S[j]^* = S[k]^*$ (Correct pairing exists)

$T[k-1] = i + 1$ (Adjacency constraint)

or the analogous constraints for branch migration in the other direction.

$T' : T'[m] = \begin{cases} \emptyset & : m = j \\ i & : m = k \\ k & : m = i \\ T[m] & : otherwise \end{cases}$ If T' is connected, then $R = [(S,T)] \rightarrow [(S,T')]$ is a reaction in the reaction network.

If T' is disconnected, then $R' = [(S,T)] \rightarrow [(S_1, T'_1), (S_2, T'_2)]$ is a reaction in the network, with S_i, T'_i defined by the split of (S, T')

2.4 4-way

Four-way branch migrations rearrange four-arm junctions. All strands involved start and end completely base-paired. This is shown in Fig. 6.

Where $C = (S,T)$ is a connected complex:

$\forall i, j, m, n$ such that:

$T[i] = j$

$T[m] = n$

$S[i] = S[m]$

$T[i-1] = n + 1$

$T[j+1] = m - 1$

$\exists R = [(S,T)] \rightarrow [(S,T')]$ where:

$\forall a \neq i, j, m, n$: $T'[a] = T[a]$

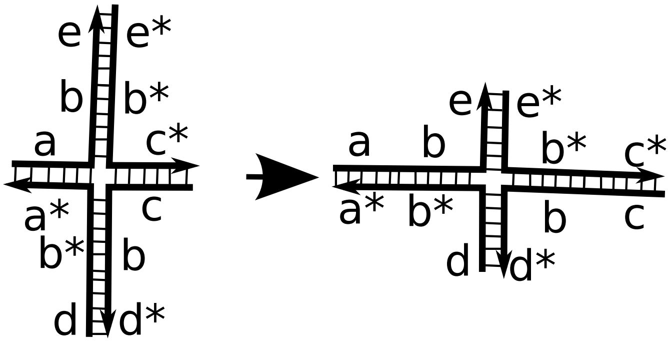

Figure 6: A 1-1 4-way branch migration. Four-way branch migrations occur at junctions like the center of this reaction. Note that this type of reaction can also be reversible or irreversible depending on whether the displacement reaches the end of the helices. 1-2 4-way branch migrations are also possible (for instance if d and e helices were not present).

$$T'[i] = n$$
$$T'[j] = m$$
$$T'[m] = j$$
$$T'[n] = i$$

R is in set of reactions iff T' is connected and is of type 4-way1-1
$R' = [(S, T)] \rightarrow [(S_1, T'_1), (S_2, T'_2)]$ (with the split complex product) is a reaction in the network otherwise and is of type 4-way1-2

3 Reaction Space

The reaction space of a given system consists of all complexes C that can be formed from the initial species using these 8 reaction types and an infinite quantity of initial reactants, and all reactions which connect these complexes.

In order to limit the space of explored reactions, reactions are time-scale separated into fast and slow reactions. Slower reactions (e.g. intermolecular interactions) are assumed to occur only occur between substrates which have already reached equilibrium through the fast reactions.

In the current code version, this is implemented such that bind2-1 reactions are only considered between complexes which are strongly connected by fast reactions without any outward fast edges.

In the general version of this, each reaction could be assigned rates (or given rate functions, so that the rate could depend on the length/composition of a domain). A threshold for intermediate timescales is given, and all possible reactions are evaluated in the step to find next slow reactions and the next fast reactions. The rate cutoff is then used to select only fast or only slow reactions for each purpose.

I use strongly connected neighborhoods (SCCs) to distinguish between groups of states and how each of them can interact. Complexes that are in a strongly connected neighborhood without outward fast edges are called fast end-states. Complexes which do not belong to such a strongly connected component are called transition states. The reasoning behind this is that each fast-strongly connected neighborhood will quickly transition between all states in the SCC through the fast reactions. If there is an outward reaction from the strongly connected component, then that means there is an irreversible reaction from the strongly connected component, thus any complex which enters the SCC will quickly transition out of the SCC through that reaction. If there is no outward edge, then any complex in the SCC is trapped there until a slow reaction occurs.

Should maybe have a figure to show this?

I will not justify further this approximation to reality.

4 Algorithm

4.1 The Sets of Complexes

There are several sets of complexes which are used during the enumeration process.

The first set is the set of end-state complexes. These complexes have already had all self-reactions enumerated and are end-states from the fast-enumeration process (they are members of fast SCCs with no outward edges). These complexes have also had all reactions between other members of this set enumerated. This set will be written as E in the following sections.

The second set consists of transition-state complexes. These complexes have also had all self-reactions enumerated, but they are not fast end-states. This means that they will not be considered as reactants for any slow reaction. If there are any bi-molecular fast reactions defined, then all such cross-reactions would need to be enumerated before adding the state to this set. It seems like that sort of a consideration has the potential to get complicated (because the other reactants for the fast reaction may not have been enumerated yet, and because of the uncertain concentrations of the other reactants), but since all bimolecular reactions are slow in the considered cases, this concern shall be ignored. This set will be denoted T.

The third set is unenumerated slow reactants. This is the set of end-state complexes that have been generated as a result of the latest iterations that have not yet had any bimolecular reactions considered. This set will be called S.

The fourth set is the current neighborhood. The current neighborhood is the directed graph of fast reactions which have been enumerated from the most recently considered complex. This set is examined to determine end-states for a given starting complex. This set will be called N.

The fifth set is the set of the products of slow reactions, B. These are completely unenumerated complexes. They will be added to F and the local neighborhood of fast reactions and complexes will be enumerated.

The final set is the unenumerated fast reactants. This is the temporary set of reactants created during the enumeration of fast reactions. This will be denoted F.

4.2 Enumeration of Complexes and Reactions

The enumeration of fast-connected complexes generates all complexes reachable from the current complex, and examines the connectedness between them to determine the corresponding end-states. This is the inner loop of the enumeration process.

```
E = []   # End-state complexes
T = []   # Transition complexes
S = []   # complexes without bimolecular interactions enumerated
N = []   # Local fast-reaction neighborhood
F = []   # unenumerated fast-reaction complexes
B = starting reactants
         # unenumerated bimolecular products.
R = []   # List of reactions

# First, make all the initial fast reactions.

While B not empty:
  remove b from B
  F = [b]
  While F not empty:
    f = F.pop()
    N.add(f)
    Rfast = fast_reactions(f)
    For product of Rfast:
      if product is a new complex (not in T,E,N,F,S)
        add product to F
      if product in B:
        add product to F, remove from B
      else:
        change pointer of Rfast to existing instance
  SCCs = Tarjans(N, Rfast)
  for SCC in SCCs:
    if SCC has no fast outward edges:
      add all elements of SCC to S
    else:
      add all elements of SCC to T
  Add all reactions to R
```

```
While S not empty:
  s = S.pop()
  E.add(s)
  R = []
  For complex in E:
    R.add(slow_reactions(s,complex))
  B = products of all reactions in R
  For b in B:
    remove b from B
    F = [b]
    While F not empty:
      f = F.pop()
      N.add(f)
      Rfast = fast_reactions(f)
      For product of Rfast:
        if product is a new complex (not in T,E,N,F,S):
          add product to F
        if product in B:
          remove it from B, add it to F
        else:
          change the pointer of Rfast to already existing instance
    SCCs = Tarjans(N, Rfast)
    for SCC in SCCs:
      if SCC has no outward edges:
        add all elements of SCC to S
      else:
        add all elements of SCC to T

Return E,T and R
```

where *generateReactions*(e) returns a set of all possible reactions involving e.

This loop can be truncated by ignoring nodes that were reached from other source nodes, this is just a further check on the membership condition, the current complex can't be in any fully enumerated set (either transition or end-state). During the strongly connected component determination, this would be replaced with a dummy node, which would have to be an end state. The end-states could then be looked up from the previously determined end-states.

5 Figures that might be useful

- Figure of the space of complexes/reactions
- Figure of the end-states

6 Other Discussions

6.1 Pruning useless branches

In toe-hold mediated reactions (such as see-saw gates, for instance) the reaction graph is populated with a number of states which have no useful products, that is to say, there is only a single fast-SCC and it has a fast output reaction which is just the reverse of the input reaction.

Seung Woo and I discussed pruning the reaction space to remove these useless branches by reconstructing the fast-reaction SCCs (using Tarjan's algorithm). If, for any fast-SCC, the input SCC was equal to the output SCC and there was no end state, then the branch could be pruned with no change to the overall pathway of the system.

6.2 Polymer Enumeration

Simple polymer enumeration seems possible with a few extensions to the given setup. The algorithm already has place for trimming and consolidating reactions. The setup currently searches through breadth-first, so it always finds the shortest pathway to a complex first. If you backtrace through the reaction you can keep track of all domains which needed to interact to get from complex A to complex B and what inputs were needed. If analogous domains are available in the same relative structure on B (i.e. if those reactions could occur again on B to lead to a new complex C), then this would imply the formation of a polymer, I think.

Obviously, this needs more thought, but it seems like it should fit in the same framework.

7 Contact and Source Code

Seung Woo Shin should have all the source code for this project, I assume it is part of the Winfree svn repository at this point. Example source code and domain specification files should also be available in the same directory. If not, email me and I can provide them and any further details on usage and implementation.