Kappa project: DNA repair

Jean Krivine [1]

PPS lab. CNRS and Univ. Paris Diderot - UMR 7126

[1] Jean.Krivine@pps.univ-paris-diderot.fr

Introduction

In this project we want to use Kappa in order to test different models of DNA Base Excision Repair (BER) pathway. In passing we wish also to study the question of how a protein with a specific DNA interaction capability, such as a transcription factor, is able to (rapidly) find its target on DNA. Indeed if one considers that a DNA strand has the order of 10^9 base pairs, finding a particular sequence of a few nucleotides by colliding with DNA at random is rather unlikely. It has been argued [5] that enzymes with specific DNA interactions combine 3D search (diffusion on the cytoplasm with random encounter with DNA) with 1D diffusion along the DNA to find their substrate (see Fig. 1).

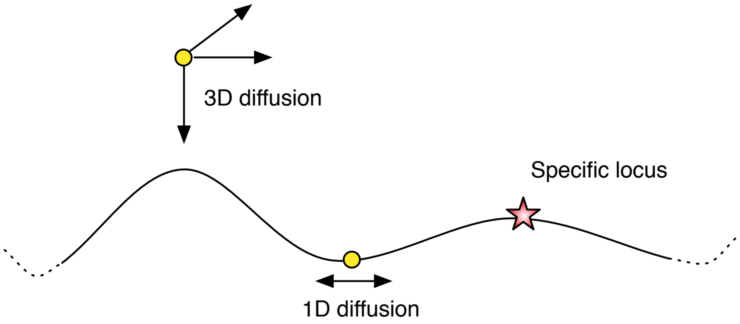

Figure 1: Two ways of finding a substrate for an enzyme with DNA binding capabilities.

We will model both 1D and 3D diffusion of the repair enzymes that are looking for T/G and U/G mismatches on DNA that arise after spontaneous hydrolytic deamination of (methylated) cytosine bases. Deamination corresponds to the chemical transformation of a cytosine into a uracil (or methylated cytosine into Thymine). Several enzymes are known to have the capacity to recognize and initiate the repair of mismatches, namely UNG, SMUG1, TDG and MBD4. In the present project we are interested in the modeling UNG (Uracil DNA Glycosylase) and TDG (Thymine DNA Glycosylase), the most active enzyme on T/G mismatches. Roughly the repair process is the following: TDG or UDG, probably after a combination of sliding and 3D diffusion, reaches a T/G or U/G mismatch on DNA. It anchors strongly to it and excises the faulty base, leaving an apurinic (AP) site: the DNA backbone with no nucleotide. The remaining part of the repair process is called *Base Excision Repair* and involves several other enzymes that are able to put a cytosine back in the AP site, re-glue the DNA backbone and reform the C:G pairing. Remethylation of the cytosine is performed by DNA methyl transferase (DNMTs) proteins.

There are three modeling challenges to raise here. The first one is of combinatorial nature: if one wishes to represent DNA sliding on a strand of size n_D, then one must take into account the fact all possible positions of the repair enzyme on DNA constitute a different molecular species, ie. there is the order of 2^{n_D} possible configurations. Also, if all the n_C cytosine bases present on a DNA may undergo a repair cycle of the form $5mC \to T \to AP \to C \to 5mC$, one must also consider the additional 3^{n_C} additional configurations of DNA. Essentially any approach which would try to enumerate (even on the fly) possible configuration would be doomed to fail. This first challenge is particularly well suited for Kappa, whose simulator has a cost per event that is independent of the number of possible molecular species the system may generate [4]. However using Kappa raises a second problem, inherent to the fact that space is abstracted away in this formalism. We will deal with this issue by showing one may use the structure link structure of DNA to model faithfully the distance between the enzyme on DNA and its substrates. Last, one has to face the question of running simulations on a realistic sample of DNA. We will see in the preliminary part of this project, that one may use KaSim to generate meaningful initial conditions.

The remaining of the project is structured as follows:

- Chapter 1 is dedicated to the construction of the DNA strands that will be the substrate of the study. A large part of this chapter was already studied in class.
- Chapter 2 incorporates a notion of DNA damage into the picture
- Chapter 3 describes the main part of the project where one seeks to model and study various hypothesis of repair mechanisms.

Chapter 1

Preliminaries: generating initial conditions

In this part of the project one is interested in generating a Kappa representation of a double DNA strand with a consistent distribution of base pairs.

1.1 The choice of DNA signature

The obvious way to go would be to use one agent per nucleotide A,T,G,C,5mC. However a better choice is to consider a generic agent N (for nulceotide) which will allow us to specify rules which apply to all types of nucleotides (see Figure 1.1). The signature of N is declared in KaSim by:

```
%agent : N(e3, o, i~A~T~G~C~5mC~U~AP, e5)
```

where e3 and e5 are respectively the 3' and 5' ends of the nucleotides, o is its "outer" face and i its "inner" face. Note that this latter has 5 possible internal states which will be used to specialize the nucleotide to an A,T,G,C,5mC (for methylated cytosine) or U (Uracil) base, or an apurinic site (AP).

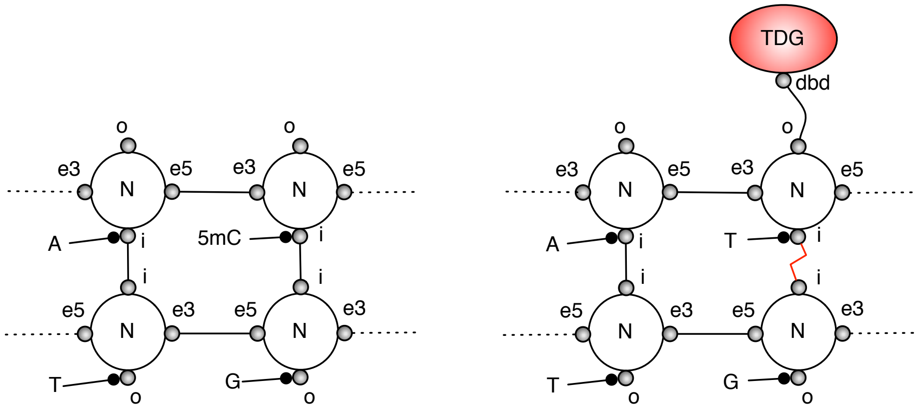

Figure 1.1: An AG fragment of DNA (left) and a TDG enzyme bound to a T/G mismatch (right) in Kappa.

1.2 Growing DNA

In a file `synth.ka` we put the signatures defined in the previous section, together with the following rules that will enable us to grow DNA strands:

```
'growAT'  N(i!0, e5), N(i!0, e3) ->
          N(i!0, e5!1), N(i!0, e3!2), N(e3!1, i~A!3), N(e5!2, i~T!3)@'kAT'
'growTA'  N(i!0, e5), N(i!0, e3) ->
          N(i!0, e5!1), N(i!0, e3!2), N(e3!1, i~T!3), N(e5!2, i~A!3)@'kAT'
'growCG'  N(i!0, e5), N(i!0, e3) ->
          N(i!0, e5!1), N(i!0, e3!2), N(e3!1, i~C!3), N(e5!2, i~G!3)@'kGC'
'growGC'  N(i!0, e5), N(i!0, e3) ->
          N(i!0, e5!1), N(i!0, e3!2), N(e3!1, i~G!3), N(e5!2, i~C!3)@'kGC'
```

Note that these rule may grow a new base pair provided there pre-exists in the mixture a base pair with a free 3':5' end. One solution would be to start from an initial state with such a pair. However a more elegant solution is seed the synthesis of DNA with the rules:

```
'seedAT' -> N(i~A!0), N(i~T!0) @ 'sAT'
'seedGC' -> N(i~G!0), N(i~C!0) @ 'sGC'
```

Recall that the convention adopted in KaSim is that agents that are created by a rule and whose interface is not fully specified, are completed using the signature of the agent[1].

In DNA, the ratio of GC pairs over AT ones is not evenly distributed. In the human genome the GC content (defined as $\frac{G+C}{A+T+G+C}$) is about 40%.

Question 1: Assuming that `'sAT'='kAT'`, what should be the value of `'kGC'` and `'sGC'` in order to respect this ratio?
Answer: The probability that rule r applies is $\alpha(r)/\sum_i \alpha(r_i)$, where $\alpha(r_i)$ is the activity of rule r_i defined as the number of matches of its left hand side, times its kinetic rate. Note that all the rules that introduce pairs of nucleotide have equal left hand sides. Therefore we can easily show that in order to respect the 40% ratio of GC content, it suffices to set $\alpha(\texttt{'growAT'})/\alpha(\texttt{'growGC'}) = \alpha(\texttt{'seedAT'})/\alpha(\texttt{'seedGC'}) = 1.5$.

In KaSim variables are declared by:

```
%var : 'kGC' 1.0
%var : 'kAT' 1.5 * 'kGC'
%var : 'sGC' 1.0
%var : 'sAT' 1.5 * 'sGC'
```

Question 2: Given a state with n strands of growing DNA. What is the likelihood that that a given strand is the next one to grow?
Answer: Any strand has exactly two matches for the growing rules, one on the 3':5' end and the other on the 5':3' end. Therefore the probability that a given strand is the next to grow does not depend on its current length and is uniformly distributed. As a result one should expect that the n strands will have in average the same length.

Question 3: One wishes to favor growing strands over creating new ones. What can we do for this?
Answer: an easy solution is to increase the ratio `'kXY'/'sXY'`. For instance we can set `'kGC'` $= 100 *$ `'sGC'` and `'kAT'` $= 100 *$ `'sAT'`.

1.3 Controlling the number of strands

Instead of growing an undetermined number of DNA strands one may wish to restrict to *exactly* n strands. To do so, one needs to count how many

[1] whenever a site that has possible internal states are not specified in the created agent, it receives the value indicated first in the signature.

times the seed rules have been applied. We will use *tokens* to do this. We add the signature:

```
%token : Cpt
```

and patch the seed rule as:

```
'seedAT' -> N(i~A!0),N(i~T!0) | 1 : Cpt @ 'sAT'
'seedGC' -> N(i~G!0),N(i~C!0) | 1 : Cpt @ 'sGC'
```

which tells KaSim to create 1 token `Cpt` at each application of a seed rule. Now the total number of seed applications can be accessed by `|Cpt|` (the number of tokens `Cpt`). One may then use the perturbation:

```
%mod : |Cpt| = n do ($UPDATE 'sAT' 0. ; $UPDATE 'sGC' 0.)
```

which will turn off seeding new strands after n applications.

1.4 Checking the DNA production

In order to check that the GC content of the mixture is respected, one defines the following variables:

```
%var : 'C' N(i~C?)
%var : 'G' N(i~G?)
%var : 'N' N()
```

which enable us to define the observable:

```
%obs : 'GC content' ('C'+'G')/('N'+0.1)
```

Note that we add 0.1 to `'N'` in order to avoid an undetermined valued for `'GC content'` at the beginning of the simulation. We can check the production of our DNA strand using the command `KaSim -i synth.ka -e 5000 -p 1000` that should synthesize a dna strand of 5,000 base pairs (see Fig. 1.2).

In order to control visually that the pairing is correctly done, one may also wish to output the final state of the simulation of `synth.ka`. To do so, one adds the following declarations:

```
%def : "dotSnapshots" "yes"
%mod : 'N' = 50 do $SNAPSHOT < "dna_".[int]('N'/2) >
```

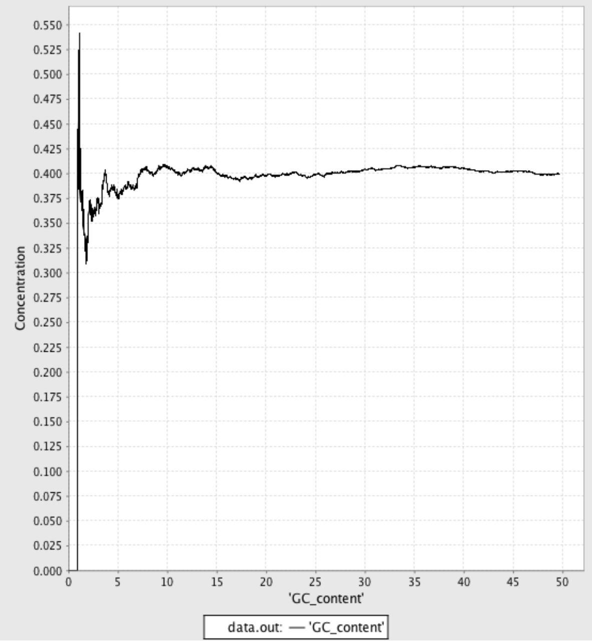

Figure 1.2: The evolution of the GC content of the DNA strand while it is being assembled.

The first line requiring snapshots to be outputted in the dot format[2], the second line requiring the production of a snapshot, that will be named `dna_25.dot` whenever the number of nucleotides has reached 50 (see Fig 1.3). Now in order to generate the initial condition of our simulations, it suffices to produce the Kappa encoding of a 10,000 bp DNA strand. To do so, one changes the perturbation above into:

```
%def : "dotSnapshots" "no"
%mod : 'N' = 20000 do $SNAPSHOT < "dna_".[int]('N'/2) >
```

[2] To be visualized with graphviz or any dot format compatible graph viewer.

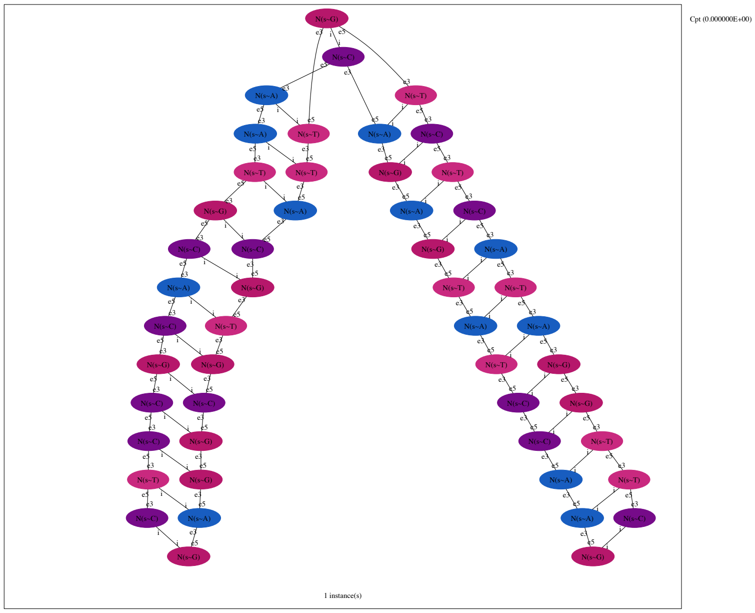

Figure 1.3: A snapshot of the dna strand of 25 bp. Note that each base is assigned its own color.

1.5 The Kappa file `synth.ka`

```
## Chapter 2: generating initial conditions

## Section 2.1 Agent signatures
%agent: N(e3,o,i~A~T~G~C~5mC~U~AP,e5)
%agent: UDG(dbd)

## Section 2.2 Growing DNA

#Setting the rate in order to respect the GC content ratio
%var: ’kGC’ 1
%var: ’kAT’ 1.5 * ’kGC’
%var: ’sGC’ 1
%var: ’sAT’ 1.5 * ’sGC’

#Initial state: creating the first base pair.
’seedAT’ -> N(i~A!0),N(i~T!0) | 1:Cpt  @ ’sAT’
```

```
’seedGC’ -> N(i~G!0),N(i~C!0) | 1:Cpt @ ’sGC’

#Growing rules
’growAT’ N(i!0,e5),N(i!0,e3) -> \
    N(i!0,e5!1),N(i!0,e3!2),N(e3!1,i~A!3),N(e5!2,i~T!3) @ ’kAT’
’growTA’ N(i!0,e5),N(i!0,e3) -> \
    N(i!0,e5!1),N(i!0,e3!2),N(e3!1,i~T!3),N(e5!2,i~A!3) @ ’kAT’
’growGC’ N(i!0,e5),N(i!0,e3) -> \
    N(i!0,e5!1),N(i!0,e3!2),N(e3!1,i~G!3),N(e5!2,i~C!3) @ ’kGC’
’growCG’ N(i!0,e5),N(i!0,e3) -> \
    N(i!0,e5!1),N(i!0,e3!2),N(e3!1,i~C!3),N(e5!2,i~G!3) @ ’kGC’

## Section 2.3 Controlling the number of strands

#Token that will count the number of strands
%token: Cpt
%init: Cpt <- 0

%mod: |Cpt| = 1 do ($UPDATE ’sAT’ 0.0 ; $UPDATE ’sGC’ 0.0)

## Section 2.4 Checking the DNA production

%var: ’C’ N(i~C?) #The number of C nucleotides
%var: ’G’ N(i~G?) #The number of G nucleotides
%var: ’N’ N() #The total number of nucleotides

%obs: ’GC content’ (’C’+’G’)/(’N’+ 0.01)

# Uncomment the next line to have snapshot as a dot format
# %def: "dotSnapshots" "yes"
%mod: [E]=[Emax] do $SNAPSHOT <"dna_".[int](’N’/2)>
```

Chapter 2

TDG sliding and deamination

2.1 The Kappa files

We should have now a Kappa file `dna_10000.ka` which contains the Kappa encoding of a 10,000 bp DNA strand generated at the end of Section 1.3. We will not mention anymore the `synth.ka` file that we used to generate `dna_10000.ka`.

The following of this chapter is dedicated to the Kappa encoding of protein-DNA interactions in a new Kappa file called `damage.ka`.

2.2 Deamination

Spontaneous cytosine deamination occurs in a cell at a quite slow rate, evaluated at $k_{deam} = 7.0 * 10^{-13}$ deamination event per second [7]. Since there are about 10^9 cytosines per genome (in a human cell) that makes roughly 3 deamination events per genome per hour. It is clear that at the level of a Kappa simulation, this would require both a very large model and a very long time in order to observe just one deamination event. Instead we will later consider a “pulse” of deamination and study how fast the system recovers.

The rules for deamination can be written:

```
'5mc Deamination' N(i~5mC?) -> N(i~T?) @ 'k_deam'
'C Deamination' N(i~C?) -> N(i~U?) @ 'k_deam'
```

Question 1: In the above rule, it is visible that a mismatch U/G is not represented by an explicit break of the bond between both nucleotides on the *i* site (the rule preserves the link if it exists). Define a Kappa variable that counts the number of correctly paired nucleotides.
Answer: We first need to define a variable, say `'bp'`, that will count the total number of base pairs:

```
%var: 'n' N()
%var: 'bp' 'n'/2
```

Now the number `'correct'` of correctly paired nucleotides is given by:

```
%var: 'nUG' N(i~U!_)
%var: 'nTG' N(i~T!1),N(i~G!1)
%var: 'nAP' N(i~AP!_)
%var: 'damaged' 'nUG'+'nTG' + 'nAP'
%var: 'correct' 'bp'-'damaged'
```

To obtain a vertical deamination pulse, one defines the following variables and perturbations:

```
%var: 'deam_percent' 0.05 # deamination ratio
%var: 'nCG' N(i~C!_)
%var: 'n5mCG' N(i~5mC!_)

#Pulse start
%mod: [E]=1000 do $UPDATE 'k_deam' [inf]

#Pulse stop
%mod: 'damaged' > 'deam_percent' * ('nC'+'5mC'+'damaged') do\
    $UPDATE 'k_deam' 0.0
```

which turns on the pulse after a 1000 events (to let the system drift away from initial conditions) and shuts down deamination when the number of mismatches has reached 5% of the initial number of cytosines (methylated or not). Note that KaSim understands infinite values, represented by the pre-defined constant `[inf]`.

2.3 Glycosylases non specific interactions

We introduce now our DNA Glycosylases, the Thymine DNA Glycosylase (TDG) and the Uracil DNA Glycosylase (UNG for the human version, sometimes called UDG). For the modeling of these enzymes in Kappa we suggest the following basic signature (which can be expanded on the need later on):

```
%agent : DG(dbd, act~U~T)
```

We will use the dbd (DNA binding domain) site of TDG and UNG for binding and sliding on DNA. The act site denotes the nature of the activity of the enzyme: either Uracil or Thymine excision.

There is no explicit notion of space in Kappa: agents are supposed to evolve in a perfectly mixed fluid where, everything being equal, agents have a uniform probability to meet. Consider the rules:

```
'DG bind' N(o), DG(dbd) -> N(o!1), DG(dbd!1) @ 'k_on'
'DG unbind' DG(dbd!_) -> DG(dbd) @ 'k_off'
```

Assuming here that both types of enzyme have the same affinity with non specific DNA.

Question 1: Given a simulation with 100 DG agents and the 10,000 bp DNA. What is the proportion of unbound DG's at steady state? What should be the value of K_D so that half of DG agents are bound on DNA at steady state?
Answer: The rate at which free DG's (d) bind on available nucleotides (n) is given by $n * d * k_{on}$. Now the rate at which bound DG's ($\bar{d}$) disconnect from DNA is $\bar{d} * k_{off}$. At steady state we have $n * d * k_{on} = \bar{d} * k_{off}$ and therefore we derive $\frac{k_{off}}{k_{on}} = K_D = \frac{n*d}{\bar{d}}$. So the ratio of free DG's over bound DG's at steady state is K_D/n which is roughly $10^{-4} * K_D$. Thus in order to set the ratio to 1 (as many free DG's than bound DG's) one should have $K_D = 10^4$ (a k_{off} 10,000 times faster than k_{on}).

2.4 1D diffusion

We now want to take into account 1D diffusion of DG's on the DNA strand. There are several possibilities: either a "random walk" (DG's would slide at each step in either the 3' or 5' direction) or a "ballistic" model in which DG's would only slide in a given direction (say 3').

To represent DG's DNA sliding (in the 3' direction) it suffices to make sure that the nucleotide which is 3' to the current location of DG is free:

```
'slide 3'  DG(dbd!1), N(o!1, s3!2), N(o, s5!2) -> \
           DG(dbd!1), N(o, s3!2), N(o!1, s5!2) @ 'k_slide'
```

Note that the reversible version of the above rule gives exactly the non biased random walk:

```
'slide 35'  DG(dbd!1),N(o!1,s3!2),N(o,s5!2) <-> \
            DG(dbd!1),N(o,s3!2),N(o!1,s5!2) @ 'k_slide','k_slide'
```

Question 1: One wishes to extend 1D diffusion with a form of 2D diffusion where a DG could move from one strand to the other. Write the corresponding Kappa rule(s).
Answer: It suffices to make the DG follow the link connecting opposite *i* sites on DNA:

```
'slide Watson-Crick'  DG(dbd!1),N(o!1,i!2),N(o,i!2) -> \
                      DG(dbd!1),N(o,i!2),N(o!1,i!2) @ 'k_slide2D'
```

Note that this rule being purely symmetric, one should not add its backward version.

Chapter 3

Modeling DNA repair steps

In this chapter we are interested in introducing the various enzymatic steps that complete the deamination/repair cycles $C \rightarrow U \rightarrow C$ and potentially $C \rightarrow 5mC$. At the end of this chapter we would like to be able to study various hypothesis concerning the way enzymatic events coordinate each others for the repair steps.

3.1 Cleaving the faulty base

Upon discovery of a mismatch on DNA, DG's anchor on the mismatch, possibly undergoing a conformational change. The faulty base (either U or T) is then captured in the catalytic pocket of the DG and eventually excised, leaving an apurinic site in its stead (AP). Importantly the DG remains tightly bound to the AP site after the excision, yielding a form of *inhibition* of the DG. Note: TDG enzymes may excise both T/G and U/G mismatches although with a much slower rate that UNG for the latter mismatch. UNG seems to be essentially specialized in U/G repair. The generic reaction is of the type:

$$\texttt{DG.DNA}[\downarrow_{\texttt{n}}] \overset{\texttt{k\_flip}}{\rightleftarrows} \texttt{DG.DNA}[\uparrow^{\texttt{n}}] \overset{\texttt{k\_chem}}{\rightarrow} \texttt{DG.DNA}[\texttt{ap}] \overset{\texttt{k\_off}'}{\rightleftarrows} \texttt{DG} + \texttt{DNA}[\texttt{ap}]$$

where $\downarrow_n$ and $\uparrow^n$ represent the faulty nucleotide (either U or T) respectively pointing downward or in the catalytic pocket of the DG. Since `k_off'` is extremely low when DG is anchored on an AP site (and therefore rate limiting) several papers have worked on identifying `k_cat` the rate at which DG performes both the flipping and the irreversible excision of the base. It turns out that this rate is quite slow and highly context dependent for TDG [13] and faster for UNG [9].

Importantly AP sites are cytotoxic for the cell. One may view the low k_off' as a way to protect the DNA by hiding the AP site with the DG.

Milestone 1: Write a complete model up-to base excision with initial condition containing DG's of type UNG and TDG. The initial amount of DG's should be reasonable with respect to the size of a real DNA they maintain in a real cell. You should study how fast mismatches are discovered (using as an observable the DG's bound to a mismatch) depending on the nature of the mismatch (either U/G or T/G), the way the sliding mechanism of the DG was modeled and any meaningful parameter.

3.2 Base Excision Repair

In this part we are interested in studying under which circumstances deaminated (methylated) cytosines are repaired. To do so we need to incorporate into the model some further actors that intervene in the Base Excision Repair (BER) pathway:

$$\mathtt{DNA}[\mathtt{ap/G}] \rightarrow^{\mathtt{APE1}} \mathtt{DNA}]\mathtt{ap/G}] \rightarrow^{\mathtt{POL}\beta} \mathtt{DNA}\cdot\mathtt{C}:\mathtt{G}] \rightarrow^{\mathtt{LIG3}} \mathtt{DNA}[\mathtt{C}:\mathtt{G}]$$

1. APE1, for AP^2 endonuclease 1, cleaves the phosphodiesteric bond of the AP site 5' of the damage, creating a single strand break on DNA: DNA]ap/G].

2. POLβ, for Polymerase β, inserts the correct nucleotide (a cytosine in our case) and removes the remaining damaged backbone of the DNA, resulting in a DNA · C : G] configuration.

3. LIG3, for DNA Ligase 3, may then seal the newly repaired base pair and reform an undamaged DNA DNA[G : G].

Milestone 2: Write the signature of the new agents and the abstract rules for their interaction with DNA. One should model the steps: APE1 binds and interacts with the substrate DNA[ap/G] (1), POLβ binds and transforms DNA]ap/G] (2) and eventually LIG3 operates on DNA[C:G] (3). This may imply modifying the signature of N agents in order to model accurately the substrates (1), (2) and (3). As a first step, one may model simple binding/chemical modification/unbinding rule. Where the binding rules for steps (1), (2) and (3) simply test that the corresponding substrate is free. The aim of the next two sections is to make these steps more efficient.

An additional protein seems to play an important role in the BER pathway. The X-ray Cross Complementing protein 1 (XRCC1) is known to act as a scaffold for POLβ and LIG3. An interesting part of the project would be to understand the rôle of such a scaffolding protein in two different approaches that model the way the DG's activity is coordinated with BER (see the next two sections).

3.3 The *Passing the baton* model

The most currently believed model concerning BER is that APE1 dislodges the DG whenever it is bound to the AP site on DNA. APE1 would then recruit POLβ whose catalytic activity on DNA would in turn dislodge APE1. Possibly via the scaffolding protein XRCC1, POLβ would be replaced by LIG3 to finish the repair. This model is called *passing the baton* [14, 12].

Milestone 3: Write refinements of the rules of Milestone 2 in order to sequentialize the repair process, in a passing the baton (PTB) manner. Note that only DGs exhibit a product inhibition mechanism (they do not dissociate from their substrate) so the PTB should be modeled by playing on kinetic rates of the chemical reactions (`A.S -> A+S' @ k and B.A.S -> B.S' + A @ k' >> k` to model the passing the baton between enzymes `A` and `B` operating resp. on substrates `S` and `S'`).

3.4 The *repairozome* hypothesis

It has also been suggested that BER is implemented by a *preassembled* complex structured around the DG and containing all the necessary agents of BER [1]. In this model the DG would accumulate partners for BER during its 3D diffusion. Upon discovery of the mismatch, most of the enzyme necessary for BER would already be present and the PTB phase would occur inside the complex.

Milestone 4: Implement a repairozome variant of BER and compare the repair speed between both models. An interesting observable is the amount of unbound AP sites during time (these AP sites being cytotoxic, one can imagine that the nature would select strongly against repair mechanisms which leave AP sites visible for too long).

3.5 The remethylation issue

Finally we would like to understand when (re)methylation of cytosine may occur at the end of BER. Our final agent is a DNA methyl-transferase (DNMT). DNMTs are sorted among two main families: the so called *maintenance* methylation enzymes and the *de novo* methyl transferases. Maintenance DNMTs are believed to maintain methylation marks that occur on CpG contexts. Thanks to the symmetrical nature of CpG contexts, if a cytosine on the Watson strand has lost its methylation marker, these enzymes will re-establish the methyl group in the Crick strand is methylated. DNMT1 is the main representative of this family of methyl-transferase. DNMT3(a) is a methyl transferase that has little activity by itself but can methylate cytosines. The discovery of its affinity with TDG [10] is an indication that DNMT3 could be brought to a particular cytosine that has just been repaired from a T/G mismatch.

Milestone 5: Add cytosine methylation to the initial state of the simulation in order to have between 5% and 10% of cytosine methylated on DNA (you can use a separate Kappa File `methyl.ka` for this, that you only use to methylate DNA). Then add DNMT3/TDG complex formation (you may try several hypothesis: the complex can form during 3D diffusion of TDG or only when TDG is bound to the AP site) and imagine some rule(s) that induce remethylation via DNMT3 at the end of BER is DNMT3 is still (indirectly) connected to the newly repaired base. In particular it would be interesting to understand what should be the relative copy number of DNMTs vs. TDG so that the repair is always complete.

3.6 References

Some papers that describe interesting facts and kinetic informations:

- DNA Glycosylases: [13] (TDG kinetics), [9] (UNG kinetics).
- Base excision repair: [11] (some interesting copy number for initial conditions), [6] (Kinetic rates for TDG excision and interaction with APE1), [8] (some details about the steps of BER), [1] (the repairosome hypothesis)
- Remethylation: [10] (DNMT and TDG interactions), [2] (Review of the DNMTs), [3] (discussion on how DNMTs find their substrate by

either 1D diffusion or brought by another enzyme –for instance TDG), [15] (kinetic rates for DNMTs activity).

Bibliography

[1] Mansour Akbari, Marit Otterlei, Javier Peña-Diaz, Per Arne Aas, Bodil Kavli, Nina B. Liabakk, Lars Hagen, Kohsuke Imai, Anne Durandy, Geir Slupphaug, and Hans E. Krokan. Repair of U/G and U/A in DNA by UNG2-associated repair complexes takes place predominantly by short-patch repair both in proliferating and growth-arrested cells. *Nucleic Acids Research*, 32(12):5486–98, 2004.

[2] Timothy H. Bestor. The DNA methyltransferases of mammals. *Human Molecular Genetics*, 9(16):2395–2402, 2000.

[3] Taiping Chen, Naomi Tsujimoto, and En Li. The PWWP domain of Dnmt3a and Dnmt3b is required for directing DNA methylation to the major satellite repeats at pericentric heterochromatin. *Mol. Cell. Biol.*, 24(20):9048–9058, 2004.

[4] Vincent Danos, Jérome Féret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In *Proceedings of APLAS'07:* 5^{t}*h ASIAN symposium on programming languages and systems*, volume 4807 of *LNCS*, pages 139–157, 2007. Invited paper.

[5] C. Dinant, Höfer T Luijsterburg MS, G. von Bornstaedt, W. Vermeulen, AB. Houtsmuller, and R. van Driel. Assembly of multiprotein complexes that control genome function. *J Cell Biol*, 185(1):21–26, 2009.

[6] Megan E. Fitzgerald and Alexander C. Drohat. Coordinating the initial steps of base excision repair. *Journal of Biological Chemistry*, 47:32680, 32690 2008.

[7] LA Frederico, TA Kunkel, and BR Shaw. A sensitive genetic assay for the detection of cytosine deamination: determination of rate constants and the activation energy. *Biochemistry*, 29(10):2532–7, 1990.

[8] Muralidhar L. Hedge, Tapas K. Hazra, and Sankar Mitra. Early steps in the DNA base excision/single-strand interruption repair parthway in mammalian cells. *Cell Research*, 18:27–47, 2008.

[9] Yu Lin Jiang and James T. Stivers. Reconstructing the substrate for uracil DNA glycosylase: Tracking the transmission of binding energy in catalysis. *Biochemistry*, 40:7710–9, 2001.

[10] Ya-Qiang Li, Ping-Zhu Zhou, Xiu-Dan Zheng, Colum P. Walsh, and Guo-Liang Xu. Association of Dnmt3a and thymine DNA glycosylase links DNA methylation with base-excision repair. *Nucleic Acids Research*, 35(2), 2007.

[11] Clifford D. Mol, David J. Hosfield, and John A. Tainer. Abasic site recognition by two apurinic/apyrimidic endonuclease families in DNA base excision repair: the 3' ends justify the means. *Mutation Research*, 460:211–219, 2000.

[12] Clifford D. Mol, Tadahide Izumi, Sankar Mitra, and John A. Tainer. DNA-bound structures and mutants reveal abasic DNA binding by APE1 DNA repair and coordination. *Nature*, 403, 2000.

[13] Michael T. Morgan, Matthew T. Bennett, and Alexander C. Drohat. Excision of 5-halogenated uracils by human thymine DNA glycosylase: robust activity for DNA contexts other than CpG. *Journal of Biological Chemistry*, 282(38):27578–27586, 2007.

[14] Jason L. Parsons, Irina I. Dianova, Sarah L. Allinson, and Grigory L. Dianov. DNA polymerase β promotes recruitment of DNA ligase IIIα-XRCC1 to sites of base excision repair. *Biochemistry*, 44:10613–19, 2005.

[15] Isao Suetake, Junko Miyazaki, Chikako Murakami, Hideyuki Takeshima, and Shoji Tajima. Distinct enzymatic properties of recombinant mouse DNA methyltransferases Dnmt3a and Dnmt3b. *Journal of Biological Chemistry*, 133:737–744, 2003.