Hands-on on DNA damage quantification

geant4-dna.org

The « moleculardna » advanced example

Contributors

Jeremy M.C. BROWN
Konstantinos CHATZIPAPAS
Paolo DONDERO
Milos DORDEVIC
Sébastien INCERTI
Mathieu KARAMITROS
Nathanael LAMPE
Dousatsu SAKATA
Wook-Geun SHIN
Ronny STANZANI
Hoang TRAN
Sara ZEIN

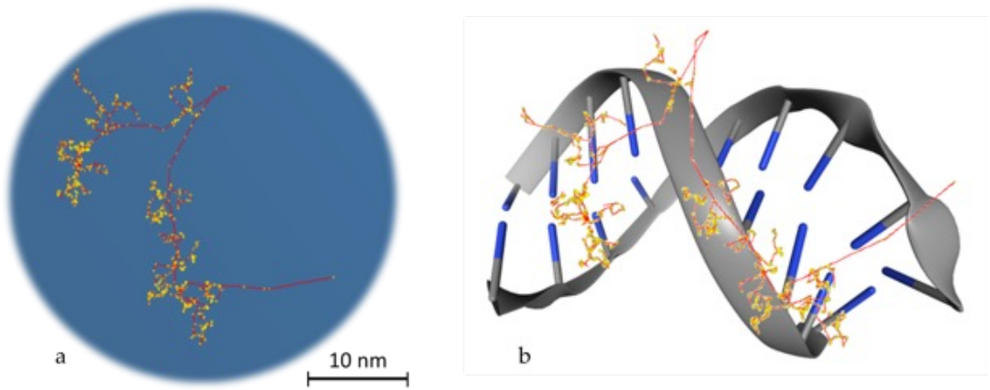
Geant4-DNA tutorial
Pohang Accelerator Laboratory, Republic of Korea
07/02/2025

Geant4 version 11.3
Released in December 2024

The « moleculardna » advanced example

- The « moleculardna » example is an **integrated simulation chain** that combines
  - Geant4-DNA **Physics** models
  - Geant4-DNA **Chemistry** models
  - Geant4-DNA DNA-scale **geometrical models** of biological targets

- This is a Geant4 advanced example and it is located in:
  `$G4INSTALL/examples/advanced/dna`

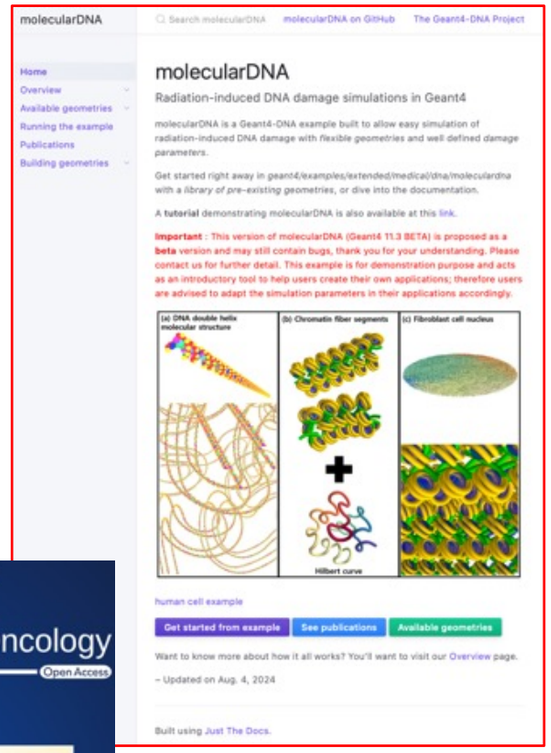


K. P. Chatzipapas, et al., Cancers 2020, 12(4), 799

The « moleculardna » advanced example

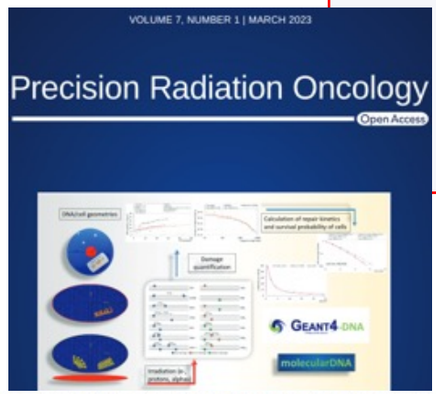
- This example aims to demonstrate how the Geant4-DNA toolkit can be used to quantify DNA damage induced by ionising irradiation
- Several types of **DNA geometries** are included
  - Chromatin fibers
  - Plasmid DNA
  - Bacterial DNA
  - Human cell DNA
- Both **direct and indirect** damage can be calculated, taking into account
  - Physics
  - Chemistry (physico-chemical and chemical stages)
  - Geometry
- The **complexity** of damage can be investigated
- **Repair and survival rate** can be calculated
- See dedicated web site: **http://moleculardna.org**

https://moleculardna.org



Web site

Review



Simulation of DNA damage using Geant4-DNA: an overview of the "molecularDNA" example application

Konstantinos P. Chatzipapas, Ngoc Hoang Tran, Milos Dordevic, Sara Zivkovic, Sara Zein, Wook-Geun Shin, Dousatsu Sakata, Nathanael Lampe, Jeremy M. C. Brown, Aleksandra Ristic-Fira, Ivan Petrovic, Ioanna Kyriakou, Dimitris Emfietzoglou, Susanna Guatelli, Sébastien Incerti
... See fewer authors

K. P. Chatzipapas, et al., Prec. Radiat. Oncol. 2023; 7: 4–14

and references therein

Geant4-DNA approach

MC Simulation Block

Physical stage
step-by-step modelling of physical interactions of incoming & secondary ionising radiation with biological medium (liquid water)

- Excited water molecules
- Ionised water molecules
- Solvated electrons

Physico-chemical/chemical stage
- Radical species production
- Diffusion
- Mutual chemical interactions

Geometrical models
DNA strands, chromatin fibres, chromosomes, whole cell nucleus, cells…
for the prediction of damage resulting from direct and indirect hits

DIRECT DNA damage

INDIRECT DNA damage

Prediction Block

Biological repair
Prediction of foci yields versus time using semi-empirical biological repair model (number of DSB, irreparable DSB fraction).

Biological endpoints are calculated using the number of DSB and their complexity.

- Protein/enzyme kinetics
- DNA rejoining
- Cell survival

$t=0$ $t=10^{-15}s$ $t=10^{-9}\sim10^{-6}$ s

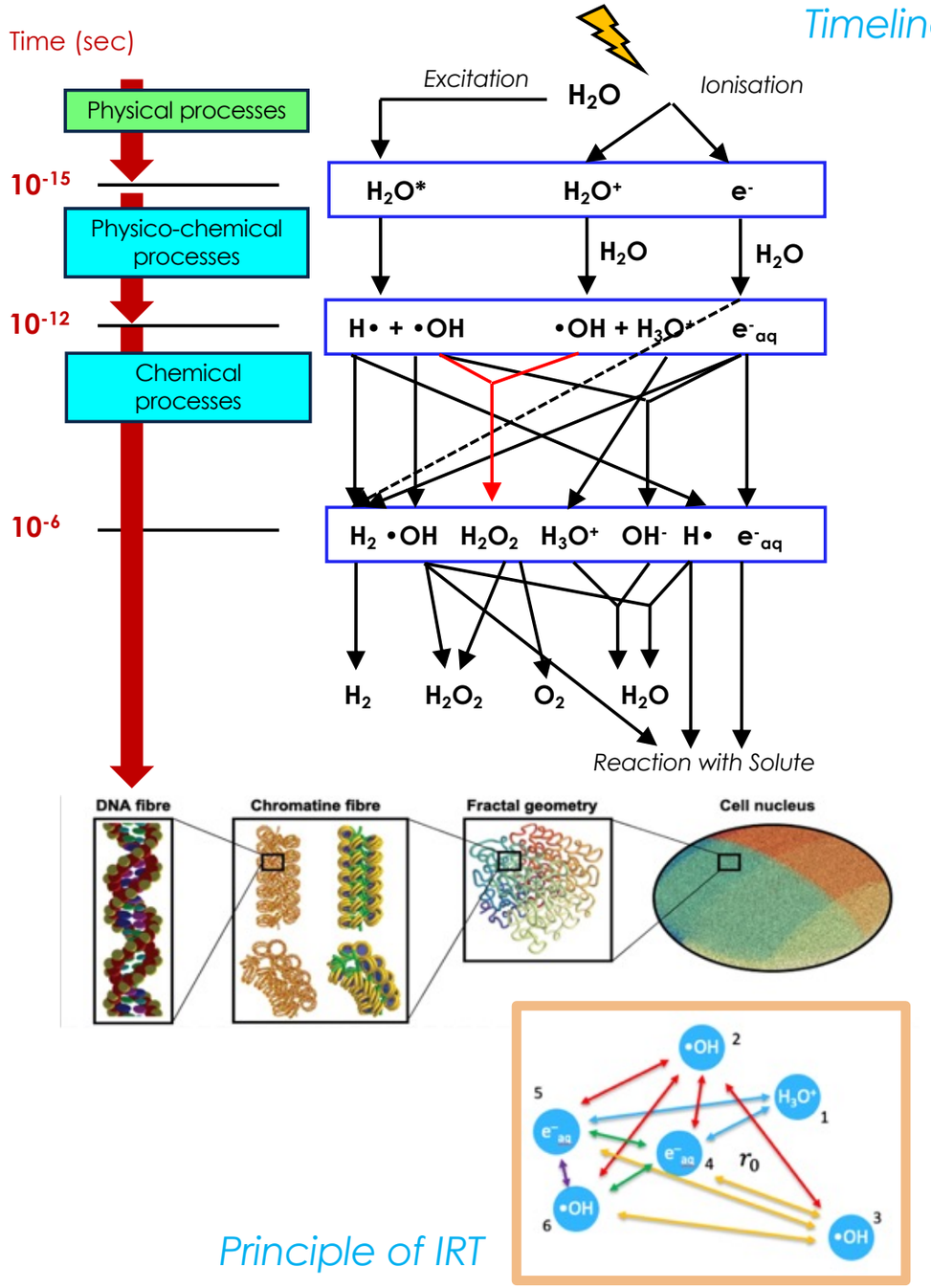
GEANT4-DNA

Physical stage

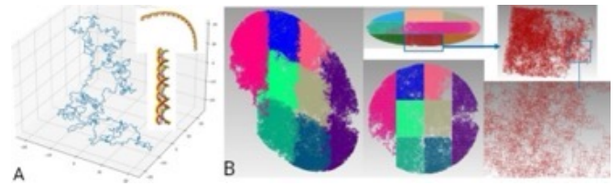
- Recommendation: use Geant4-DNA physics constructors
  - `G4EmDNAPhysics_option2`
  - `G4EmDNAPhysics_option4`
  - `G4EmDNAPhysics_option6`

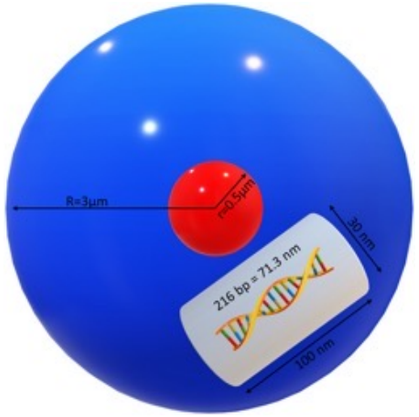
Chemical stage

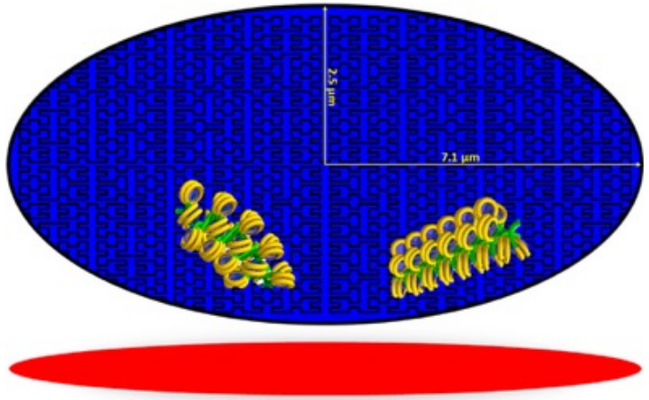
- Independent Reaction Times (« IRT ») approach
  - From the 1980's by Clifford, Green et al., widely used today
  - Iterative process where the approximation of « independent pairs » is assumed: calculates the reaction times between all possible pairs of reactive species, as if they were isolated.
  - No longer necessary to diffuse the molecular species and to calculate the possible reactions between the species at each time step.
  - A « synchronous » alternative **hybrid version** (« **IRT-sync** ») **is used**: it gives all spatio-temporal info on radicals, as it is required to combine with the DNA geometries.

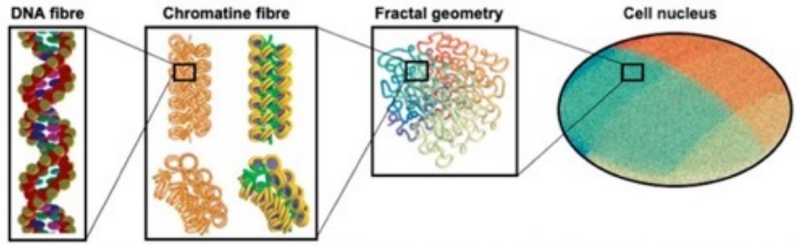


Examples of geometrical models created from the « fractalDNA » tool

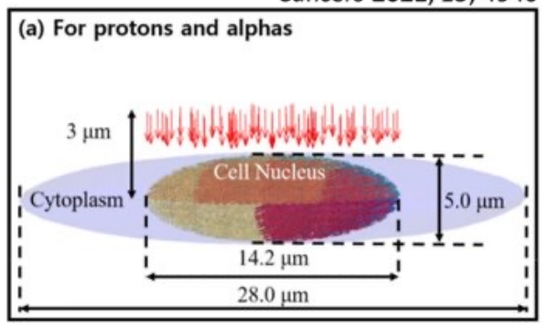


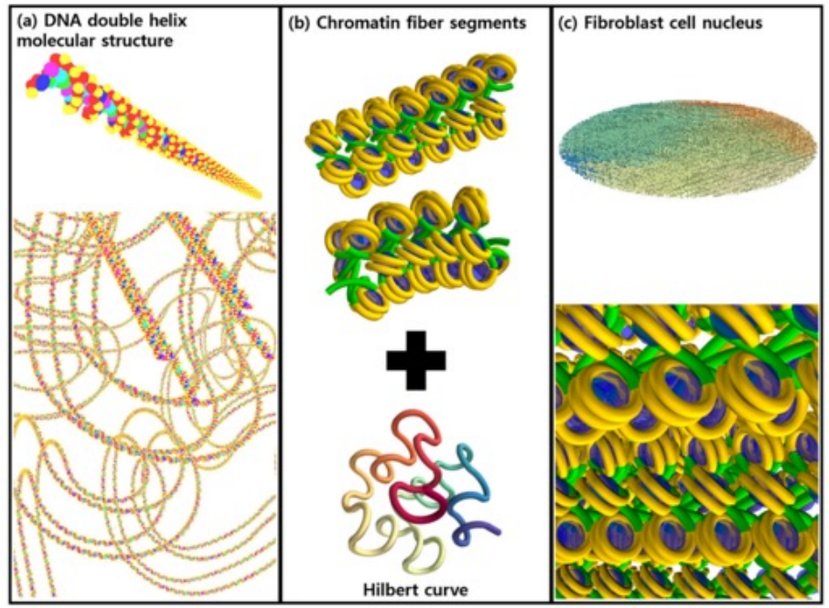






Cancers **2021**, *13*, 4940





- **The « fractalDNA » tool (by N. Lampe):** a Python package to make DNA geometries that can be joined together like jigsaw puzzles

https://pypi.org/project/fractaldna/

https://natl.github.io/fractaldna/

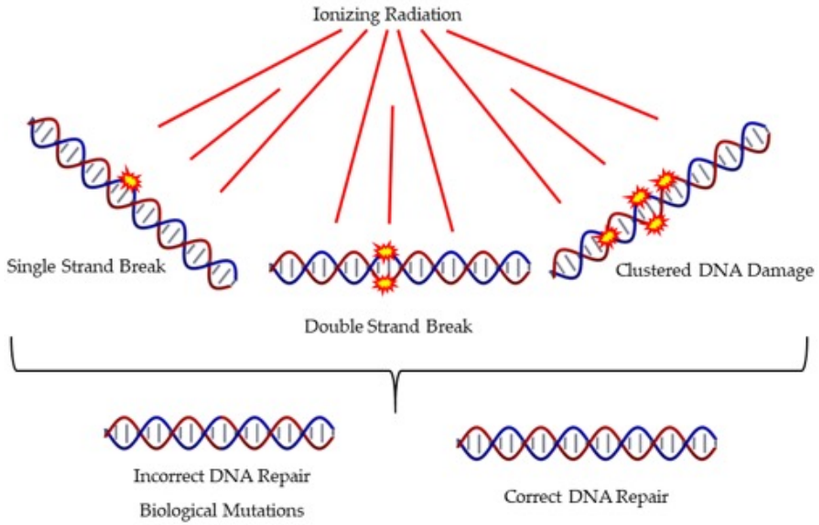
https://github.com/konhat88/complexDNA

DNA damage classification

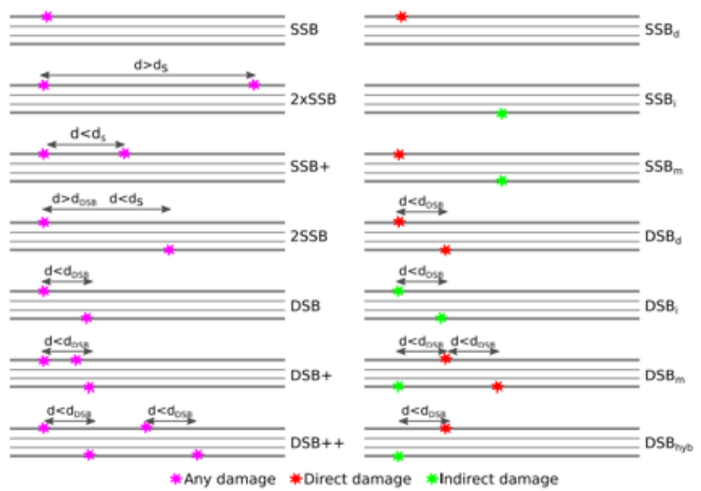
- **Direct damage**
  occurs when energy from physical processes is deposited near a DNA molecule.

  In moleculardna, we associate damage either with a **strand** molecule (sugar or phosphate) or a **base** molecule.

- **Indirect damage**
  is scored when a chemical reaction leads to a strand break.



K. P. Chatzipapas, et al., Cancers 2020, 12(4), 799



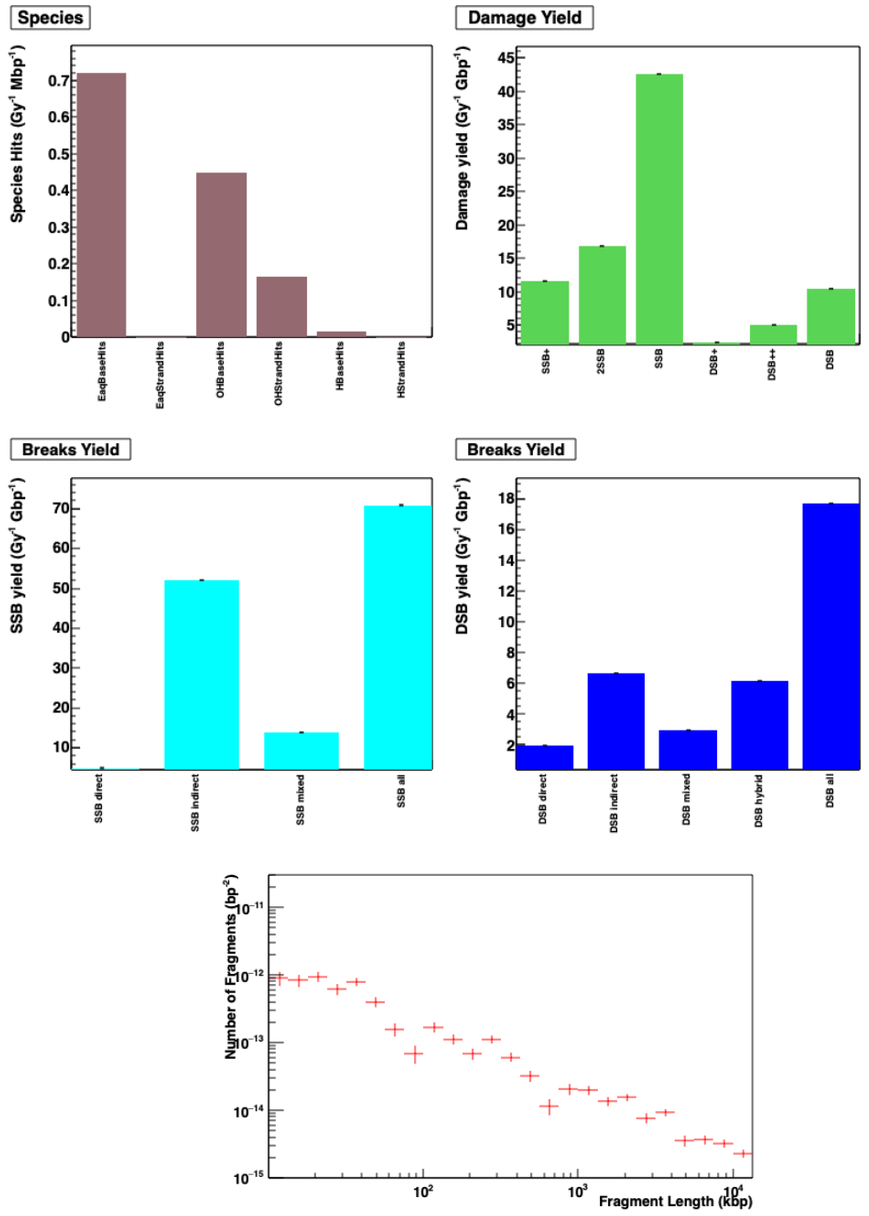
N. Lampe, PhD thesis, 2017

Results of the simulation: the ROOT macros

https://root.cern.ch

- **Species hits** (/Gy/Mbp)
  defined as the name of radical species together with the DNA reaction
  e.g. `EaqStrandHits` is e^-_{aq} + DNA backbone
- **Damage yield** (/Gy/Gbp)
  defined by DNA damage complexity
  (see classification scheme – previous slide)
- **Break yield** (/Gy/Gbp)
  shown for each break type
  (direct SSB, indirect SSB, DSB,…)
- **Fragment distribution of DNA**
  A fragment is part of DNA between two DSBs, with length equal to their separation distance.
- See more explanations at this page:
  https://geant4-dna.github.io/molecular-docs/docs/overview/results-and-analysis
  - Or moleculardna.org → Overview → Results and analysis

Graphical output of the ROOT macros



List of macro files & scripts (1/4)

Chromatin

Plasmid

Bacterium

Cell

- **Geant4 macro files** to **run the moleculardna simulation** (***.mac**)
  - `cylinders.mac`
    - Simulates a **parameter study** based on the publication of H. Nikjoo et al. Can be used for regression testing of the different parameters.
  - `plasmid.mac`
    - Simulates the irradiation of **10000 plasmids** (pBR322) in liquid water. The length of the DNA chain in each plasmids is **~4.4 kbp**.
  - `ecoli.mac`
    - The geometry of an ***E. coli* bacterium** has been modeled and can be used to simulate early damage induction by irradiation. The length of the DNA contained in the bacterial cell is **~4.63 Mbp**.
  - `human_cell.mac, human_cell_HTB177.mac, human_cell_MCF7.mac, human_cell_chromosomes.mac`
    - A **human fibroblast cell** has been modeled and is included in this mac file. The length of the DNA included in this cell is **~6.4 Gbp**.
    - One default model (keratinocyte), **two alternative models** for HTB177 (lung) and MCF7 (breast) cancer cell lines (see K. Chatzipapas et al.) and one version with **chromosomes** (see K. Chatzipapas et al.)
- **ROOT macro files** to **analyze damage** results (***.C**)
  - `cylinders.C`
    - Analysis of simulation results of the cylinders.mac file. Prints Break Source and Break Complexity frequency.
  - `plasmid.C`
    - Analysis of simulation results of the plasmid.mac file. Prints Number of damages per plasmid and damage distributions.
  - `ecoli.C`
    - Analysis of simulation results of the ecoli.mac file. Prints Species hits, Damage yield, Breaks yield and Fragments distribution.
  - `human_cell.C, human_cell_alphas.C, human_cell_chromosomes.C`
    - Analysis of simulation results of the human_cell.mac file. Prints Species hits, Damage yield, Breaks yield and Fragments distribution.
    - A specific version for the irradiation with alpha particles, of the geometries HTB177 and MCF7: `human_cell_alphas.C`
    - A specific version for the irradiation of the chromosome geometry: `human_cell_chromosomes.C`
- **Python extras**
  - `createSDD.py`
    - After the simulation, ROOT data can be converted to Standard DNA Damage (SDD) data format (see SDD format).
  - `repair_survival_models` directory
  - `phase_space` directory

List of macro files & scripts (2/4)

- **Late damage** estimation with dedicated Python scripts (*.py) (1/3): see [1][2][3]
  - The **repair model** by **O. Belov**, considers **4 repair pathways** as presented in the function:
    - non-homologous end-joining (NHEJ),
    - homologous recombination (HR),
    - single-strand annealing (SSA), and
    - alternative end-joining mechanism (Alt-NHEJ)

$$\frac{dN_0}{dt} = a(L)\frac{dD}{dt}N_{cDSB} - V_{NHEJ} - V_{HR} - V_{SSA} - V_{microSSA}$$

  - To use the script, the procedure is typical for Python scripts.
    In a terminal the user can type from the `repair_survival_models` directory:

```
python3 molecularDNArepair.py
```

  - **Several parameters** need to be defined:

```
iRootFile  = "/path/to/molecular-dna.root"
outputFile = "/path/to/molecularDNArepair.txt"
r3 = 7100*1e-09 * 2500*1e-09 * 7100*1e-09    # volume calculation, if ellipsoid cell
mass = 997 * 4 * 3.141592 * r3 / 3           # mass calculation, if ellipsoid cell
NBP = 6405886128      # length of the cellular DNA in base pairs
```


References:
[1] A quantitative model of the major pathways for radiation-induced DNA double-strand break repair, O. V. Belov et al. J Theor Biol., Feb 7;366:115-30, 2015: link
[2] Performance Evaluation for Repair of HSGc-C5 Carcinoma Cell Using Geant4-DNA, D. Sakata et al., Cancers, 13, p. 6046, 2021: link
[3] Simulation of DNA damage using Geant4-DNA: an overview of the "molecularDNA" example application, K. Chatzipapas et al. Prec Radiat Oncol. 1–11. 2023: link

List of macro files & scripts (3/4)

- **Late damage** estimation with dedicated Python scripts (*.py) (2/3) : see [1][2]
  - The **cell survival** model is based on the **two-lesion kinetics (TLK) model**. It includes kinetic processes of fast- and slow-DNA repair, and, based on lethal DNA damage, it can calculate the survival fraction (SF) of a cell population.
  - Mathematically, the **Survival Fraction** of cells is calculated using:

$$\mathrm{SF}(t) = \exp(-L_f(t)) = \exp\left(-\int_0^t (\beta_1\lambda_1 L_1(t) + \beta_2\lambda_2 L_2(t) + \gamma\eta[L_1(t) + L_2(t)]^2)dt\right)$$

    - **$L_1(t)$** is the number of lesions per cell in the fast- repair process at a given time **t** after the beginning of the irradiation.
    - **$L_2(t)$** is the number of lesions per cell in the slow- repair process at a given time **t**.
    - **$L_f(t)$** is the number of lethal lesions that may lead to cell death at time **t**.

  - **Repair probability** coefficients, represent the rate of rejoined lesions (λ and η):
    - $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, and $\boldsymbol{\eta}$ correspond to fast-, slow-, and binary- rejoining processes, respectively ($hour^{-1}$).

  - **Lethality probability** coefficients, represent the probability that a residual lesion may lead to cell death (β and γ):
    - $\boldsymbol{\beta_1}$, $\boldsymbol{\beta_2}$, and $\boldsymbol{\gamma}$ correspond to fast-, slow-, and binary- rejoining processes, respectively ($hour^{-1}$).

References:
[1] Two-lesion kinetic model of double-strand break rejoining and cell killing, Stewart RD. Radiat Res. 2001: link
[2] Simulation of DNA damage using Geant4-DNA: an overview of the "molecularDNA" example application, Chatzipapas et al. Prec Radiat Oncol. 1–11. 2023: link

List of macro files & scripts (4/4)

$$\mathrm{SF}(t) = \exp(-L_f(t)) = \exp\left(-\int_0^t (\beta_1\lambda_1 L_1(t) + \beta_2\lambda_2 L_2(t) + \gamma\eta[L_1(t) + L_2(t)]^2)dt\right)$$

- **Late damage** estimation with dedicated Python scripts (*.py) (3/3)
  - To use the tool, the procedure is typical for python scripts. In a terminal the user can type from the `repair_survival_models` directory:

    ```
    python3 molecularDNAsurvival.py
    ```

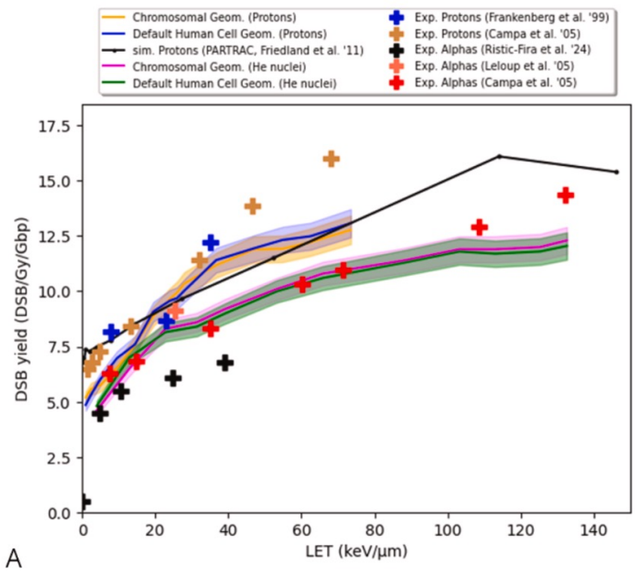
  - **Several parameters** need to be defined (indicative values):
    - `iRootFile  = "/path/to/molecular-dna.root"`
    - `outputFile = "/path/to/molecularDNAsurvival.txt"`
    - `r3 = 7100*1e-09 * 2500*1e-09 * 7100*1e-09  # volume calculation, if ellipsoid cell`
    - `mass = 997 * 4 * 3.141592 * r3 / 3  # mass calculation, if ellipsoid cell`
    - `NBP = 6405886128  # length of the cellular DNA in base pairs`
    - `cell  = "test"  # name of the cell`
    - `Lamb1 = 0.0125959  # indicative values of parameters` λ_1, λ_2, η, β_1, β_2, γ
    - `Lamb2 = 1`
    - `Eta   = 7.50595e-06`
    - `Beta1 = 0.0193207`
    - `Beta2 = 0`
    - `gamma = 0.189328`

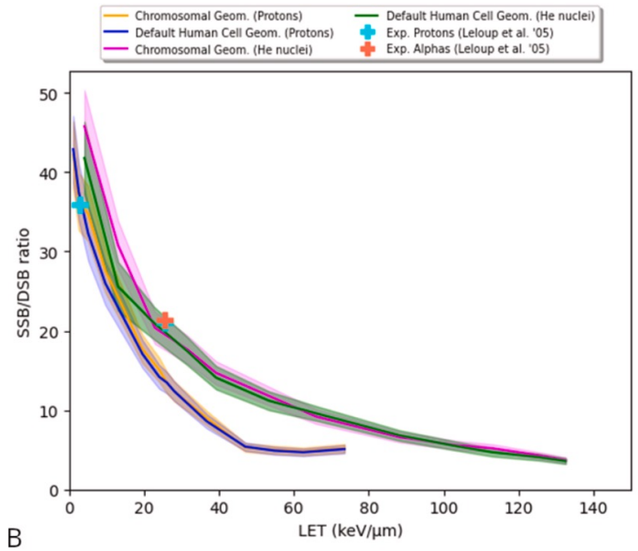
Examples of verification & validation

K. P. Chatzipapas, et al.,
Prec. Radiat. Oncol. 2023; 7: 4–14

DSB & SSB/DSB vs LET:

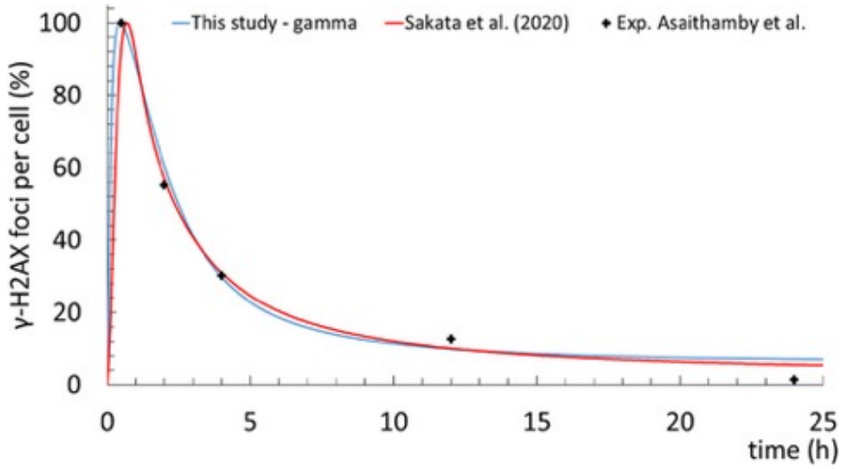
moleculardna & measurements

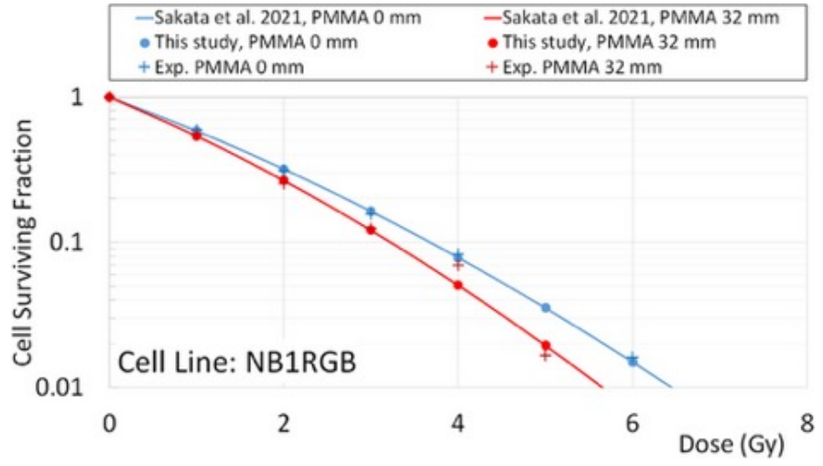




K. P. Chatzipapas, et al.,
Phys. Med. 127, 104839, 2024

Damage repair & cell survival: moleculardna & measurements





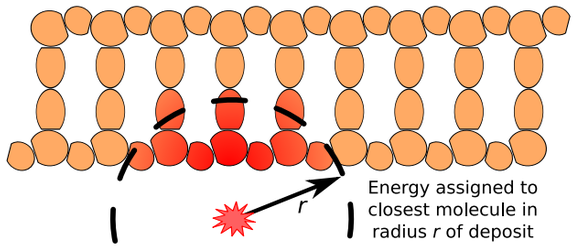
Important: parameter choice

Physics

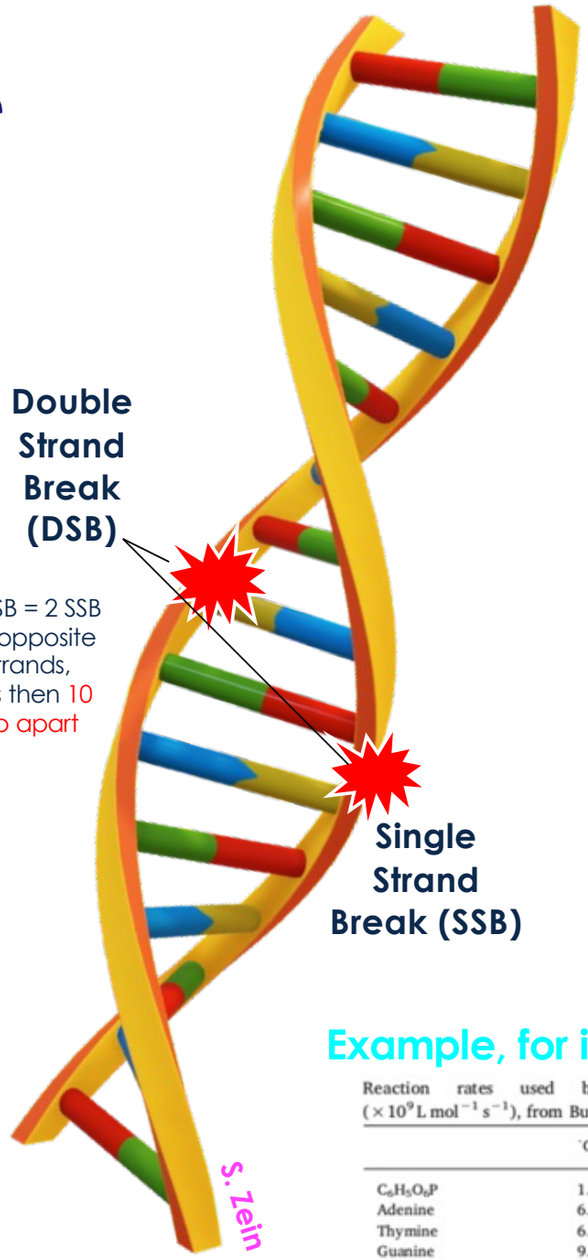
DIRECT damage induction

1. **Choice of G4DNA physics constructor**
2. **Volume for energy deposition scoring in DNA backbone (D or P molecule)**
3. **Probability of Single Strand Break induction in DNA backbone**
   - Threshold, linear…

Example, for item 2:



E.g. 2 A° for ~1 hydration layer



Chemistry

NON-DIRECT damage induction

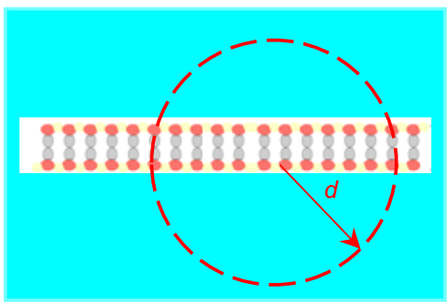
1. **Choice of G4DNA chemistry constructor**
   - Including reactions with DNA components
2. **Probability of non-direct SSB induction**
   - •OH on DNA backbone : e.g. 40.5 %
3. **Distance from DNA to kill radicals (mimic scavenging in cells)**
4. **Histones considered as full scavengers (in cells)**
5. **Radiolysis maximum time steps**
6. **Chemical stage end time**

Example, for item 1:

Reaction rates used between radicals and DNA components ($\times 10^9$ L mol^{-1} s^{-1}), from Buxton et al. [65].

	$^{\cdot}OH$	$H^{\cdot}$	e^-_{aq}
$C_6H_5O_8P$	1.8	0.029	0.01
Adenine	6.1	0.10	9.0
Thymine	6.4	0.57	18.0
Guanine	9.2	–	14.0
Cytosine	6.1	0.092	13.0

Example, for item 3:



https://moleculardna.org

Content of macro file: e.g. human_cell.mac

```
# Physics: choice of thermalization model
/process/dna/e-SolvationSubType Meesungnoen2002
#/process/dna/e-SolvationSubType Ritchie1994
#/process/dna/e-SolvationSubType Terrisol1990
```

Physics

```
# Verbosity: settings
/run/verbose 1
/control/verbose 1
```

```
# Chemistry: selection of IRT_syn
/process/chem/TimeStepModel IRT_syn
# Chemistry: end time of chemistry stage
/scheduler/endTime 5.0 ns
# Chemistry: set maximum allowed zero time steps
/scheduler/maxNullTimeSteps 10000000
```

Chemistry

```
# Geometry: size of World volume
/world/worldSize 50 um
# Geometry: shape of the cell
/cell/radiusSize 14 2.5 14 um

# Geometry: creation
# See https://geant4-dna.github.io/molecular-docs/docs/examples/bacterial-cell
#   - Side length for each placement
/dnageom/placementSize 75 75 75 nm
#   - Scaling of XYZ in fractal definition file
/dnageom/fractalScaling 75 75 75 nm
#   - Path to file that defines placement locations
/dnageom/definitionFile geometries/cube-centred-X-8.txt
#   - Set placement volumes
/dnageom/placementVolume turn geometries/turned_solenoid_750_withHistone.txt
/dnageom/placementVolume turntwist geometries/turned_twisted_solenoid_750_withHistone.txt true
/dnageom/placementVolume straight geometries/straight_solenoid_750_withHistone.txt

# Geometry: distance from base pairs at which radicals are killed
/dnageom/radicalKillDistance 9 nm

# Geometry: deposited energy accumulation range limit to start recording SBs from direct effects
/dnageom/interactionDirectRange 2.0 angstrom
```

Geometry

```
# Damage: model settings
/dnadamage/directDamageLower 5 eV
/dnadamage/directDamageUpper 37.5 eV

/dnadamage/indirectOHBaseChance 1.0
/dnadamage/indirectOHStrandChance 0.405
/dnadamage/inductionOHChance 0.0

/dnadamage/indirectHBaseChance 1.0
/dnadamage/indirectHStrandChance 0.0
/dnadamage/inductionHChance 0.0

/dnadamage/indirectEaqBaseChance 1.0
/dnadamage/indirectEaqStrandChance 0.0
/dnadamage/inductionEaqChance 0.0

# Analysis: add ellipsoid chromosomal region of interest, with the name "cell"
/chromosome/add cell ellipse 7100 2500 7100 0 0 0 nm 0 0 0
```

Damage

```
# Run: initialization
/run/initialize
# Run: print progress
/run/printProgress 10
```

Run

```
# Source geometry
/gps/pos/type Plane
/gps/pos/shape Circle
/gps/pos/centre 0 3000 0 nm
/gps/pos/rot1 0 0 1
/gps/pos/rot2 1 0 0
/gps/pos/radius 7100 nm
/gps/direction 0 -1 0
# Source particle
/gps/particle e-
# Source energy
/gps/energy 0.662 MeV

# Beam on
/run/beamOn 2
```

Shoot particles

UI commands of moleculardna (1/4) (FYI only)

Geometry related commands (1/2)

- **`/world/worldSize <s> <unit>`**
  Side length for the world.
- **`/dnageom/setVerbose <int>`**
  Print verbose debugging information related to the DNA geometry.
- **`/dnageom/definitionFile <filepath>`**
  Path to file that defines placement locations.
- **`/dnageom/placementVolume <name> <filepath> [<twist>]`**
  Set a placement volume, twist is an optional boolean parameter (written as true or false.
- **`/dnageom/fractalScaling <x> <y> <z> <unit>`**
  Scaling and units for the fractal along each axis.
- **`/dnageom/placementSize <x> <y> <z> <unit>`**
  Side length for each placement.
- **`/dnageom/checkOverlaps <bool>`**
  Check overlaps of molecules and fractal placements being placed for debugging.

UI commands of moleculardna (2/4) (FYI only)

Geometry related commands (2/2)

- **`/dnageom/setSmartVoxels <int>`**
  Change the amount of voxelisation in the Geant4 geometry optimisation for a faster simulation initialisation, but slower overall simulation (1 refers to maximal optimisation in initialisation).

- Chromosomes can be added to define regions of interest. For all chromosome types, a name is required. The x, y and z variables refer to the translation of the chromosome, and the optional rotations in x, y and z are Euler rotations.

  - **`/chromosome/add sphere <name> <rad> <x> <y> <z> <unit> [<rx> <ry> <rz>]`**
    Add a spherical chromosome with a specified radius.

  - **`/chromosome/add cyl <name> <rad> <height> <x> <y> <z> <unit> [<rx> <ry> <rz>]`**
    Add a cylindrical chromosome with a specified height and radius.

  - **`/chromosome/add rod <name> <rad> <height> <x> <y> <z> <unit> [<rx> <ry> <rz>]`**
    Add a rod-shaped chromosome. This is a cylinder of a specified height, with two hemispherical end caps. The radius of the cylinder and end caps is specified.

  - **`/chromosome/add ellipse <name> <sx> <sy> <sz> <x> <y> <z> <unit> [<rx> <ry> <rz>]`**
    Add an ellipsoidal chromosome, with semi-major axes <sx> <sy> and <sz>.

- **`/chromosome/plotData <filename>`**
  Save a scatter plot (x,y,z data points) of all chromosome positions.

UI commands of moleculardna (3/4) (FYI only)

Damage related commands (1/2)

- **`/dnageom/interactionDirectRange <d> <unit>`**
  Distance from DNA molecule at which energy deposits count towards DNA damage.
- **`/dnageom/radicalKillDistance <d> <unit>`**
  Distance from DNA at which to stop tracking radicals.
- **`/dnadamage/directDamageLower <d> <unit>`**
  Minimum Energy required for an SSB.
- **`/dnadamage/directDamageUpper <d> <unit>`**
  Maximum energy required for an SSB to occur.
- **`/dnadamage/indirectOHBaseChance <d>`**
  Chance ∈ [0,1] of a •OH damaging a base.
- **`/dnadamage/indirectOHStrandChance <d>`**
  Chance ∈ [0,1] of a •OH damaging a sugar-phosphate moiety.
- **`/dnadamage/inductionOHChance <d>`**
  Chance ∈ [0,1] of a reaction between a base and •OH yielding a strand break.

UI commands of moleculardna (4/4) (FYI only)

Damage related commands (2/2)

- **`/dnadamage/indirectHBaseChance <d>`**
  Chance ∈ [0,1] of a H• damaging a base.
- **`/dnadamage/indirectHStrandChance <d>`**
  Chance ∈ [0,1] of a H• damaging sugar-phosphate moiety.
- **`/dnadamage/inductionHChance <d>`**
  Chance ∈ [0,1] of a reaction between a base and H• yielding a strand break.
- **`/dnadamage/indirectEaqBaseChance <d>`**
  Chance ∈ [0,1] of a e^-_{aq} damaging a base.
- **`/dnadamage/indirectEaqStrandChance <d>`**
  Chance ∈ [0,1] of a e^-_{aq} damaging sugar-phosphate moiety.
- **`/dnadamage/inductionEaqChance <d>`**
  Chance ∈ [0,1] of a reaction between a base and e^-_{aq} yielding a strand break.
- **`/scheduler/endTime <d> <unit>`**
  End time of the simulation (related to the chemical part).

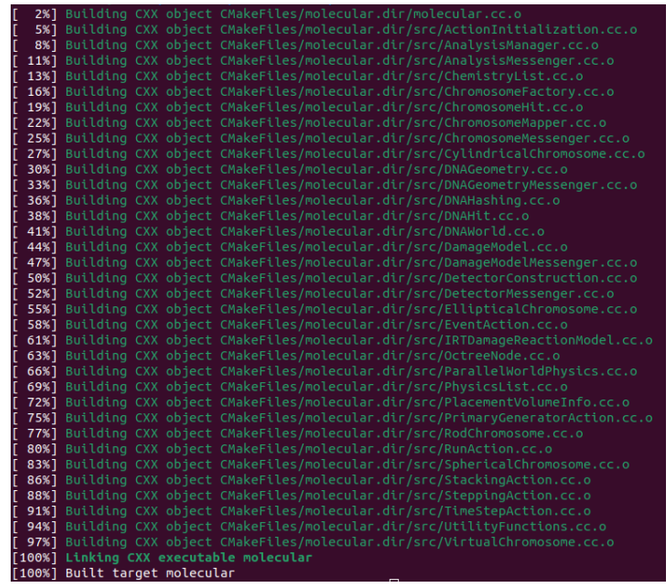
Hands-on

Hands-on practice with the « moleculardna » advanced example

- **Copy** the moleculardna advanced example to your local directory, **create your build** directory and **compile** moleculardna
- How do do this?

```
cd
cp -R $G4EXAMPLES/advanced/dna/moleculardna .
cd moleculardna
mkdir build
cd build
cmake ..
make
```

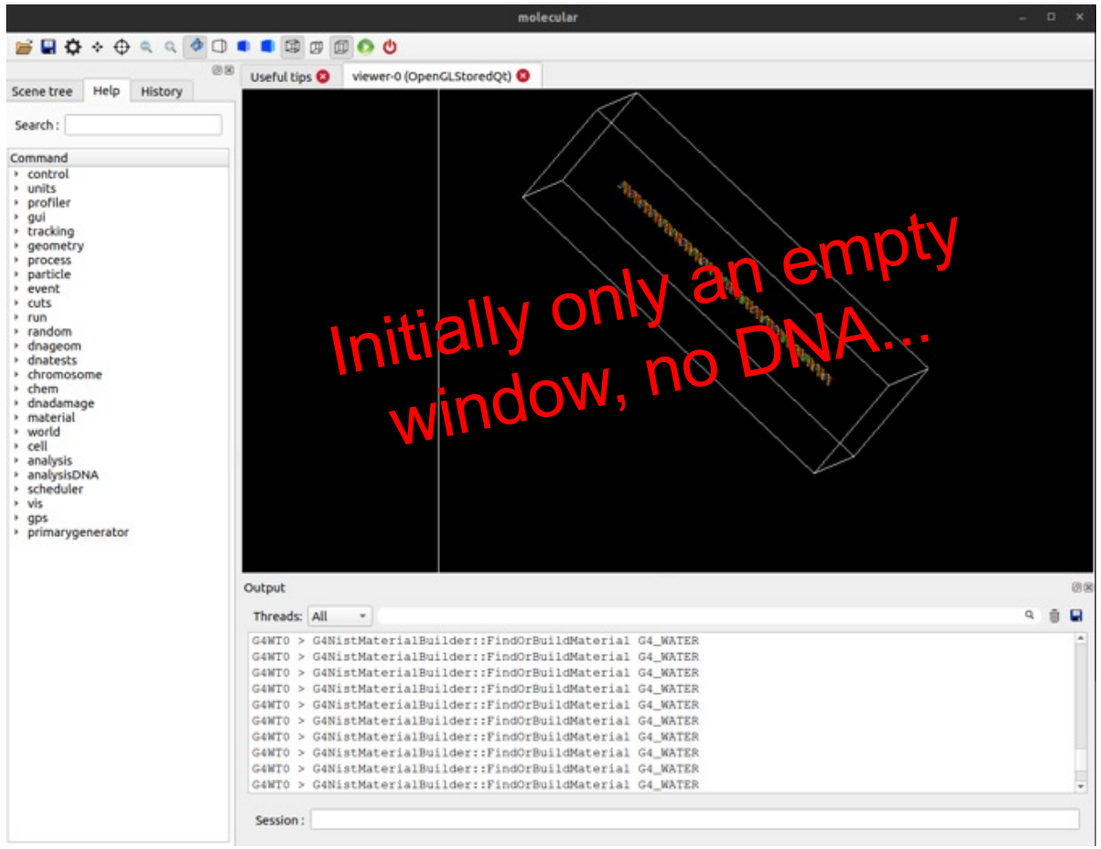
`make -jN` if you have N cores



Note: download of geometry files might be slow

Hands-on practice with the « moleculardna » advanced example

- Run moleculardna in **interactive mode** with GUI commands
  - `./molecular`
- Using `/control/execute` command, you can run any example in interactive mode
  - See next slides
- Be careful on the RAM used…



Hands-on practice with the « moleculardna » advanced example

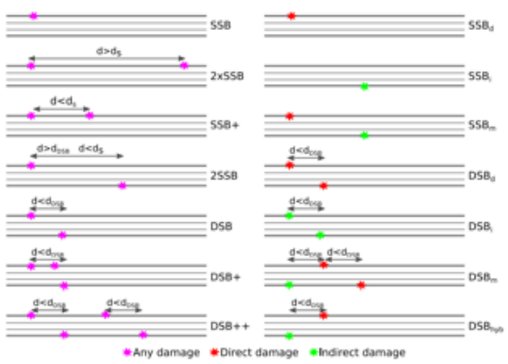
- Run moleculardna in **batch mode** using a macro file
  - You can run for 1000 electrons of 4.5 keV
    - use `gedit` to edit the macro file...
  - `./molecular -m cylinders.mac -p 4 -t 1`
  - No visualization by default
  - Results are saved in the `molecular-dna_t0.root` file

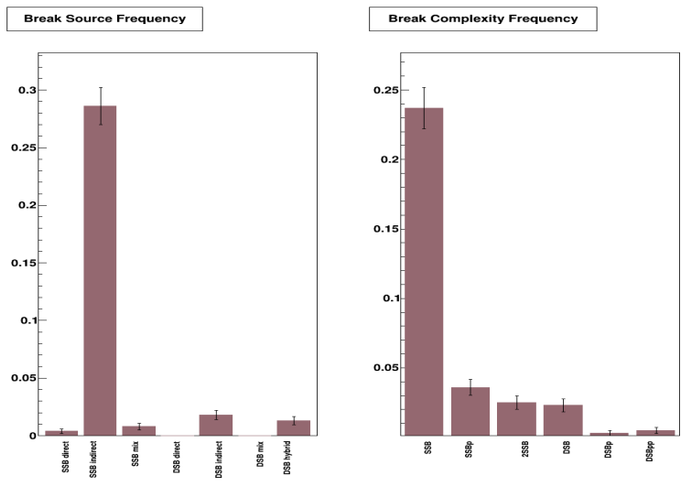
Physics

Threads

- Results can be analyzed using ROOT (depending on the version these commands my differ)
  - ROOT is already installed on your Geant4 virtual machine
    - `root`
  - Within ROOT
    - `.X cylinders.C`
  - Histograms will be plotted

```
konstantinos@kc64:~/software/testfolder/moleculardna/build$ root
   ------------------------------------------------------------------
  | Welcome to ROOT 6.26/10                        https://root.cern |
  | (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for linuxx8664gcc on Nov 16 2022, 10:42:54                 |
  | From tags/v6-26-10@v6-26-10                                      |
  | With c++ (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0                  |
  | Try '.help', '.demo', '.license', '.credits', '.quit'/'.q'       |
   ------------------------------------------------------------------

root [0] .X cylinders.C
hadd Target file: molecular-dna.root
hadd compression setting for all output: 1
hadd Source file 1: molecular-dna_t0.root
hadd Target path: molecular-dna.root:/
hadd Target path: molecular-dna.root:/hists
hadd Target path: molecular-dna.root:/tuples
Paricle : e     Energy [/MeV] : 0.0045  number : 1000
 Output Damages :
        SSB direct : 0.004       error : 0.00199699
        SSB indirect : 0.286     error : 0.0162008
        SSB mix : 0.008  error : 0.0028185

        DSB direct : 0   error : 0
        DSB indirect : 0.018     error : 0.00420639
        DSB mix : 0      error : 0
        DSB hybrid : 0.013       error : 0.00358383

        SSB  : 0.237     error : 0.0148003
        SSBp : 0.036     error : 0.00589396
        2SSB : 0.025     error : 0.00493957

        DSB : 0.023      error : 0.00474273
        DSBp : 0.003     error : 0.00173032
        DSBpp : 0.005    error : 0.00223159
```



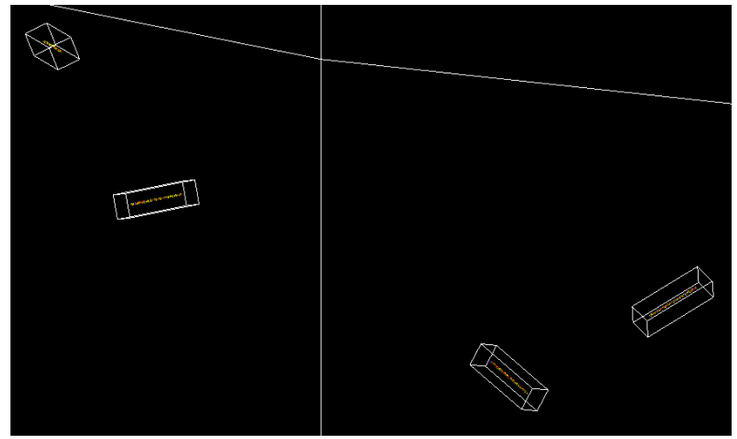


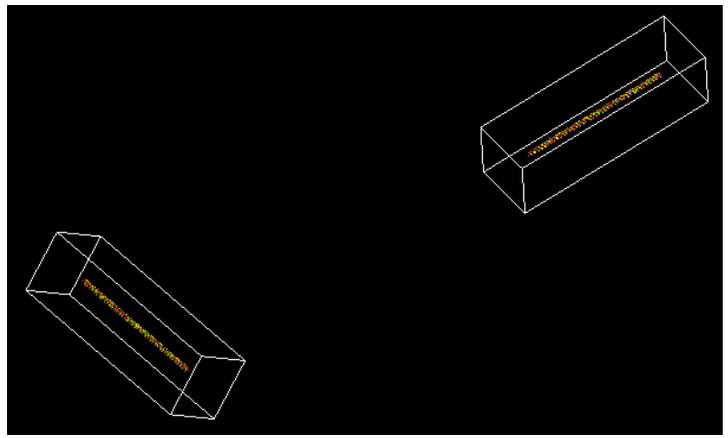
Hands-on practice with the « moleculardna » advanced example

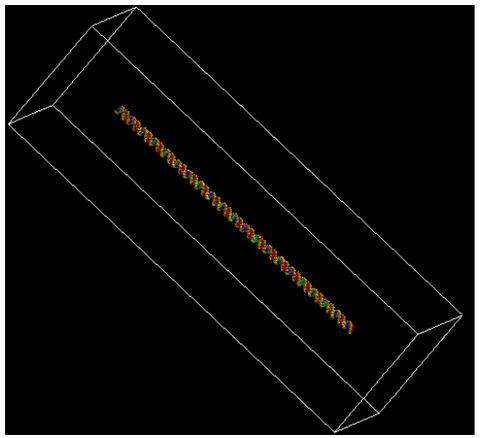
1. Open `DetectorConstruction.cc`
   - Go to line 53 and enable « DNAWorld »

     ```
     G4bool useParallelPhysicsWorld = true;
     ```
2. Open `PhysicsList.cc`
   - Go to line 73 and enable « DNAWorld »

     ```
     G4bool useParallelPhysicsWorld = true;
     ```
3. Recompile moleculardna (do `make`)
4. In **geometries** dir., create a copy of prisms200k_r3000.txt
5. Open prisms200k_r3000.txt:
   - Keep only the first 50 prisms (0-49) (up-to line 51 strictly)
6. Open cylinders.mac:
   - Comment the last line: `/run/beamOn`
   - Un-comment line `/control/execute vis.mac`
   - And save
7. In the terminal type `./molecular`
8. In the Qt window type:
   `/control/execute cylinders.mac`
9. Observe the DNA geometry !



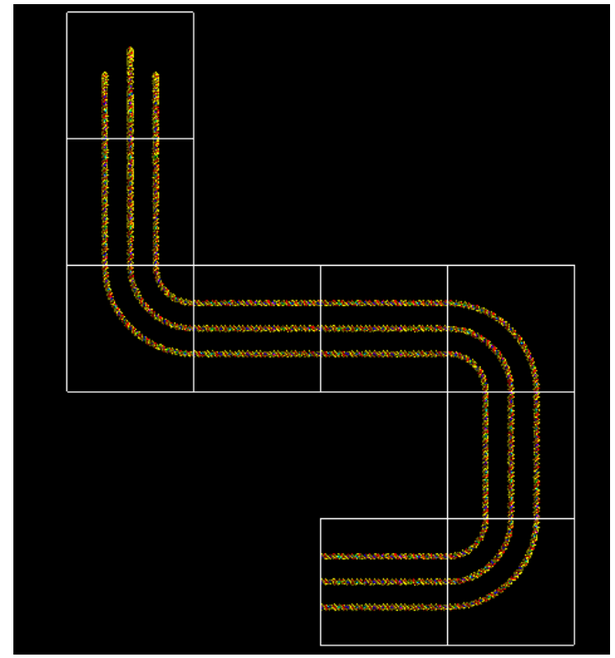


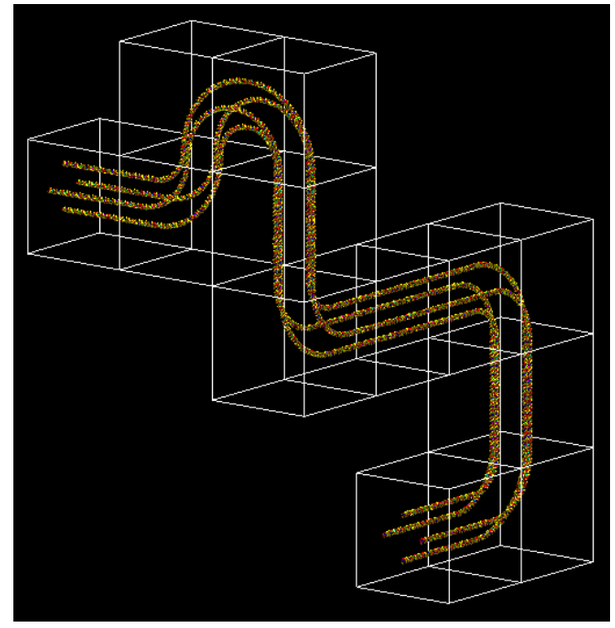


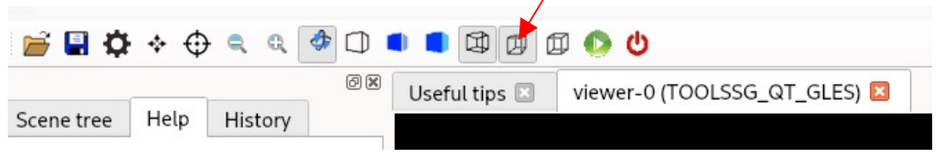
Hands-on practice with the « moleculardna » advanced example

- Using **`4strands_50nm_straight.txt`** and **`4strands_50nm_turn.txt`**, create a **new geometry** like the one shown in the following images.
- Use **`cylinders.mac`** as a starting point.
- The following lines are given to help you…

```
/dnageom/placementSize 50 50 50 nm
/dnageom/fractalScaling 1 1 1 nm
/dnageom/definitionFile geometries/newGeometry.txt
/dnageom/placementVolume straight geometries/4strands_50nm_straight.txt
/dnageom/placementVolume turn geometries/4strands_50nm_turn.txt


# IDX TYPE POS_X POS_Y POS_Z EUL_PSI EUL_THETA EUL_PHI
0 straight 0 0 0 0 0 0
1 turn 0 0 50 0 0 1.5708
2 straight 0 50 50 1.5708 0 0
3 turn 0 100 50 0 1.5708 1.5708
```

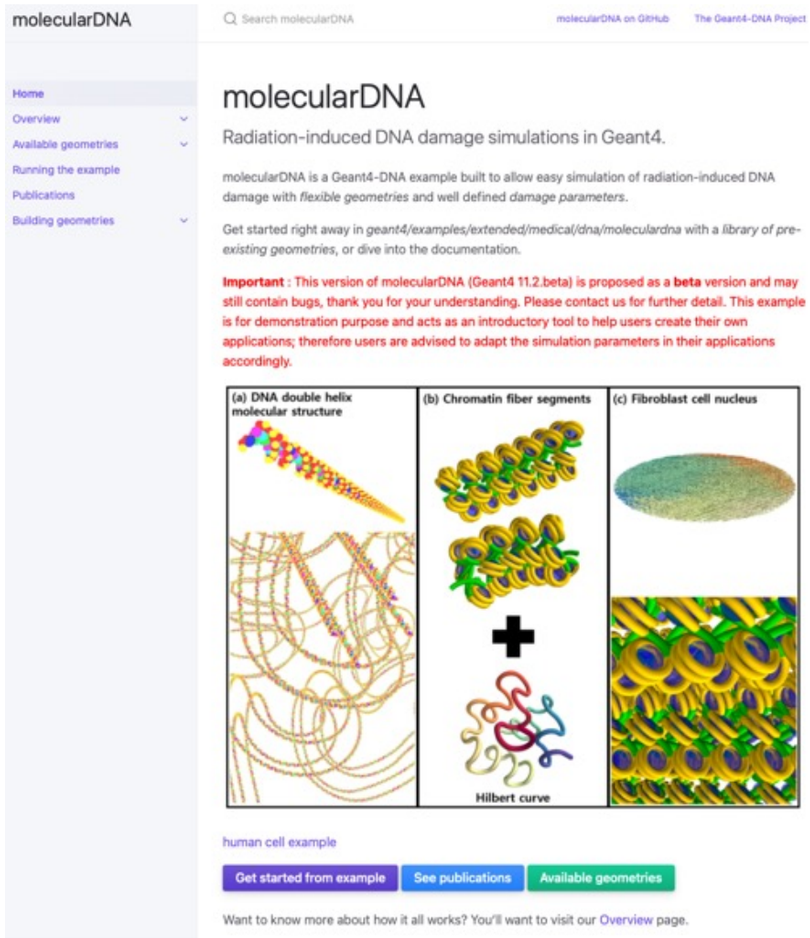




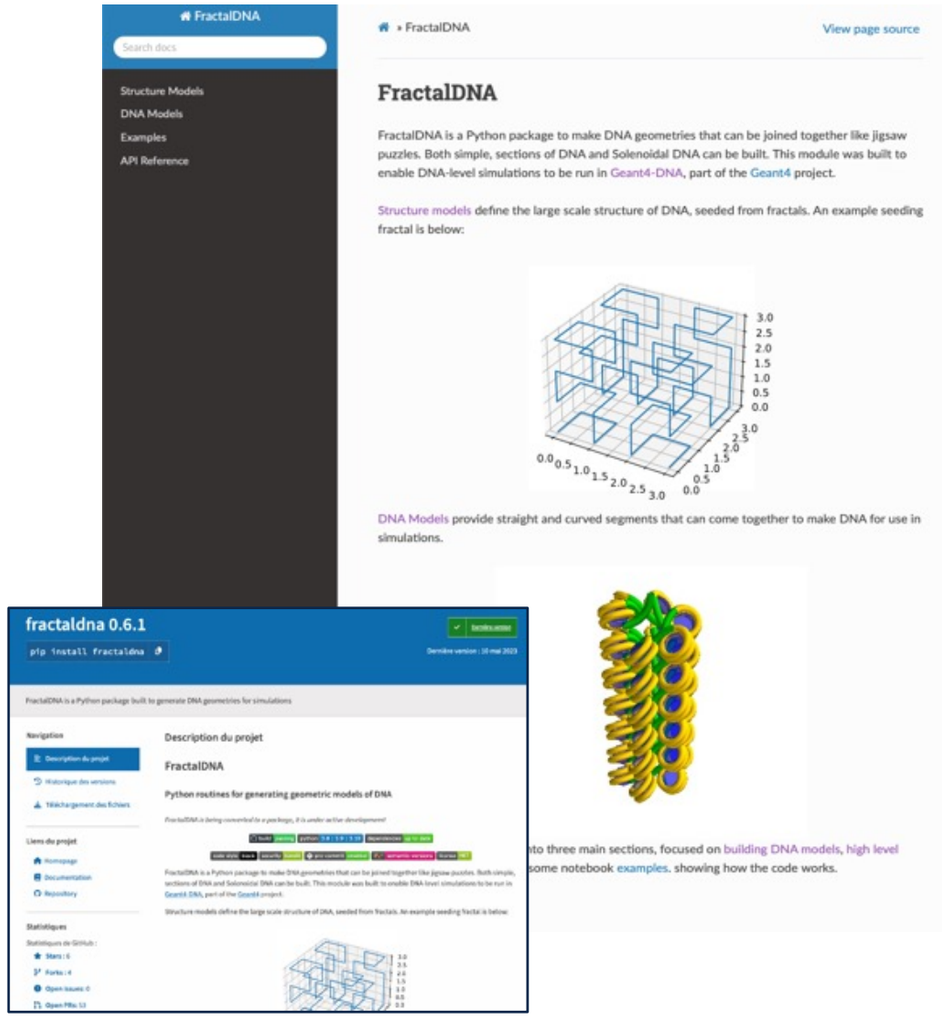


Documentation

moleculardna documentation
https://moleculardna.org



FractalDNA documentation
https://pypi.org/project/fractaldna/
https://natl.github.io/fractaldna/



Publications

https://moleculardna.org
https://geant4-dna.org/

- **Development of a novel computational technique to create DNA and cell geometrical models for Geant4-DNA**,
  K. Chatzipapas et al., Phys. Med. 127 (2024) 104389 (link)
- **Geant4-DNA simulation of human cancer cells irradiation with helium ion beams**,
  K. Chatzipapas et al., Phys. Med. 112 (2023) 102613 (link)
- **Prediction of DNA rejoining kinetics and cell survival for V79 cells using Geant4-DNA**,
  D. Sakata et al., Phys. Med. 105 (2023) 102508 (link) (corrigendum)
- **Simulation of DNA damage using Geant4-DNA: an overview of the "molecularDNA" example application**,
  K. Chatzipapas et al., Prec. Radiat. Oncol. (2023) 1–11 (link)
- **Performance Evaluation for Repair of HSGc-C5 Carcinoma Cell Using Geant4-DNA**,
  D. Sakata et al., Cancers 13 (2021) 6046 – (link)
- **A Geant4-DNA evaluation of radiation-induced DNA damage on a human fibroblast**,
  W.-G. Shin et al., Cancers 13 (2021) 4940 (link)
- **Fully integrated Monte Carlo simulation for evaluating radiation induced DNA damage and following repair using Geant4-DNA**,
  D. Sakata et al., Sc. Rep. 10 (2020) 20788 (link)
- **Evaluation of early radiation DNA damage in a fractal cell nucleus model using Geant4-DNA**,
  D. Sakata et al., Phys. Med. 62 (2019) 152-157 (link)
- **Mechanistic DNA Damage Simulations in Geant4-DNA Part 2: Electron and Proton Damage in a Bacterial Cell**,
  N. Lampe et al., Phys. Med. 48 (2018) 146-155 (link)) (corrigendum)
- **Mechanistic DNA Damage Simulations in Geant4-DNA Part 1: A parameter study in a simplified geometry**,
  N. Lampe et al., Phys. Med. 48 (2018) 135-145 (link)

Thank you for your attention