

Hands-on on DNA damage quantification



geant4-dna.org

The « moleculedna » 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

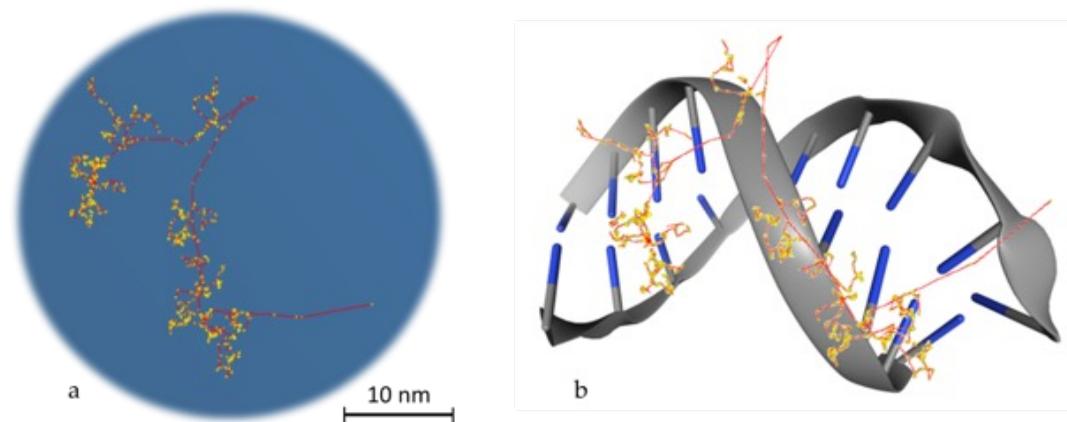
2026 Geant4-DNA International Tutorial
Princess Srisavangavadhana Faculty of Medicine,
Bangkok, Thailand

18-20/03/2026

Geant4 version 11.4
Released in December 2024

The « molecular dna » advanced example

- The « molecular dna » 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](#)

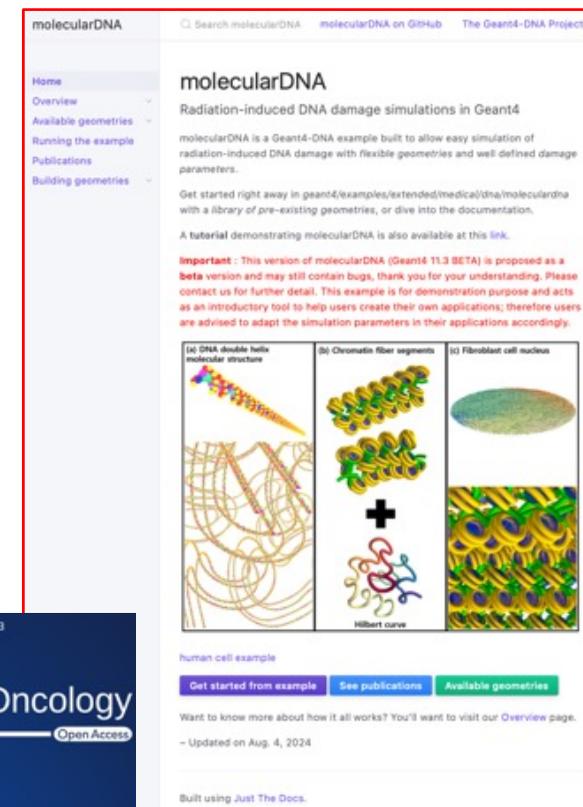


[K. P. Chatzipapas, et al., Cancers 2020, 12\(4\), 799](#)

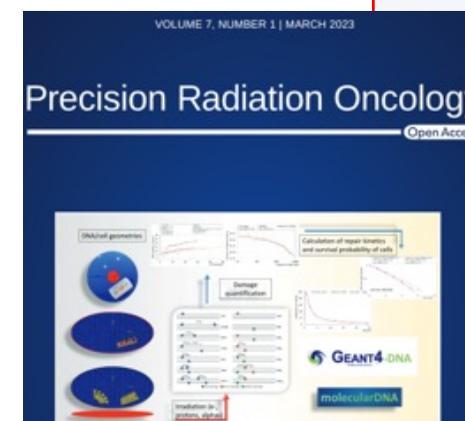
The « moleculardna » advanced example

- 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>



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-Gyun 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

Web site

Geant4-DNA approach



Physical stage

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

MC Simulation Block

- 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

$t=0$

$t=10^{-15}s$

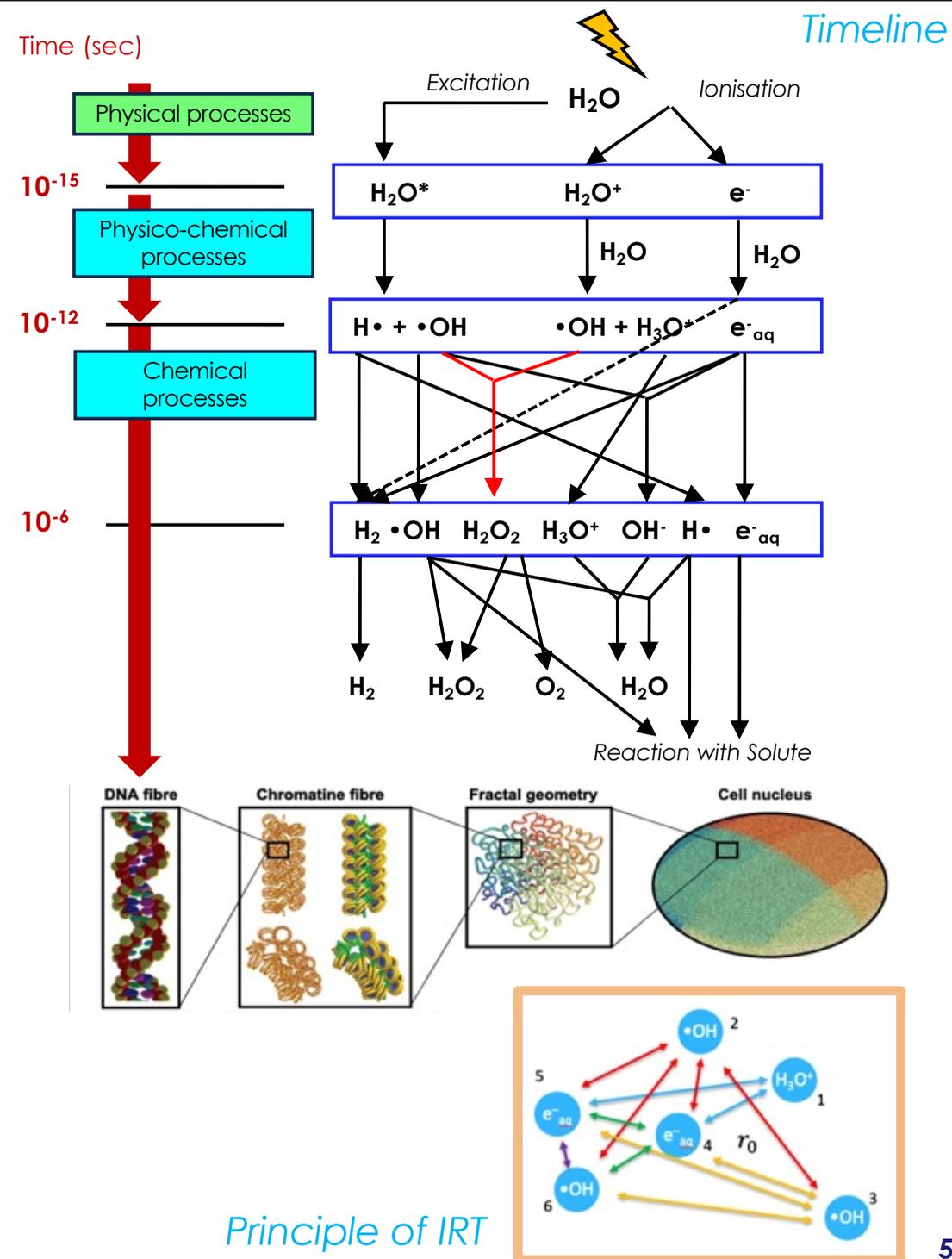
$t=10^{-9}\sim 10^{-6}s$

Physical stage

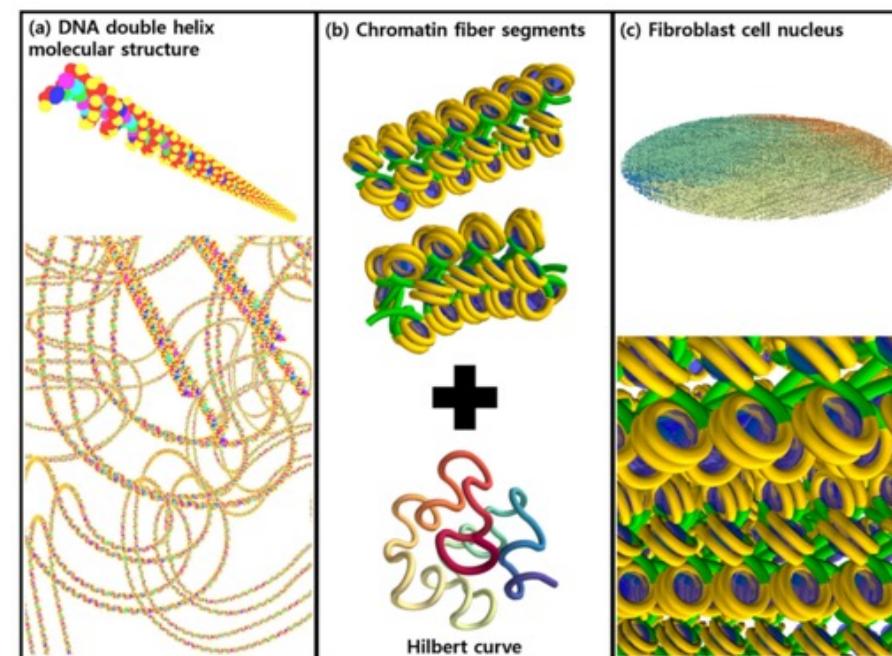
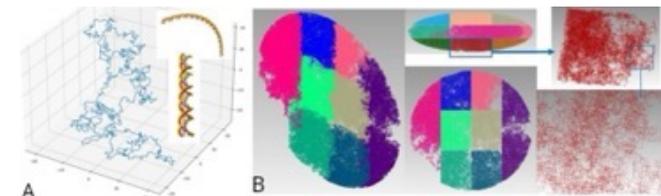
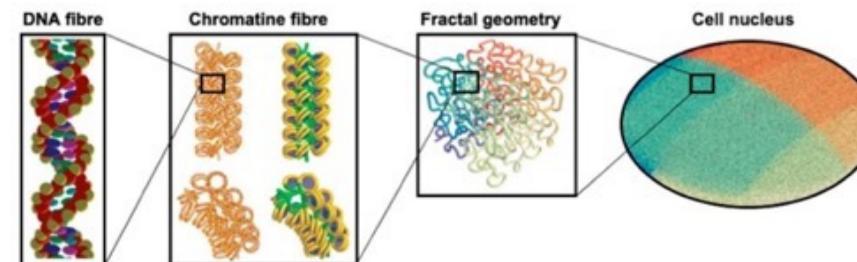
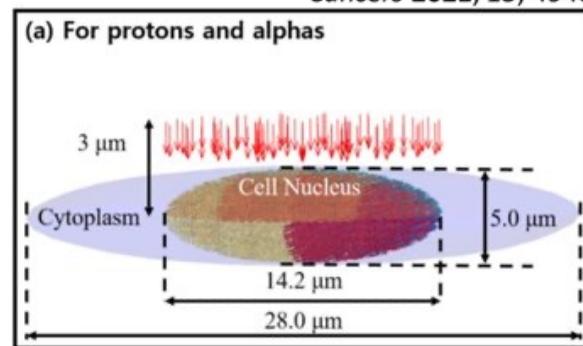
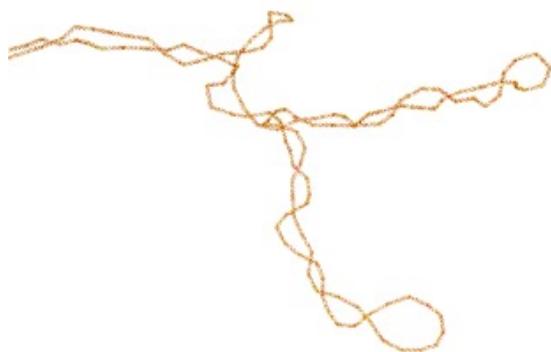
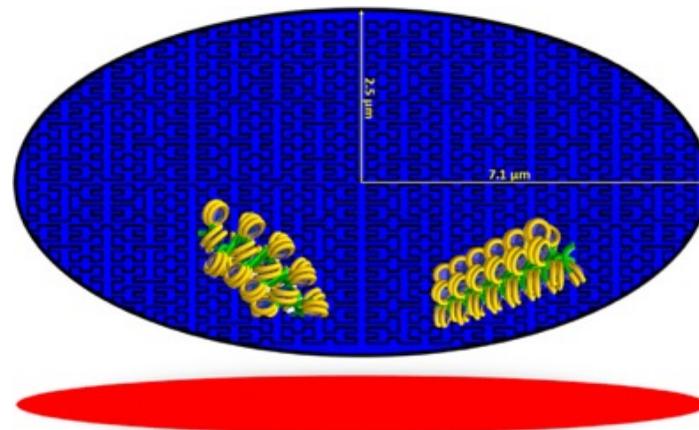
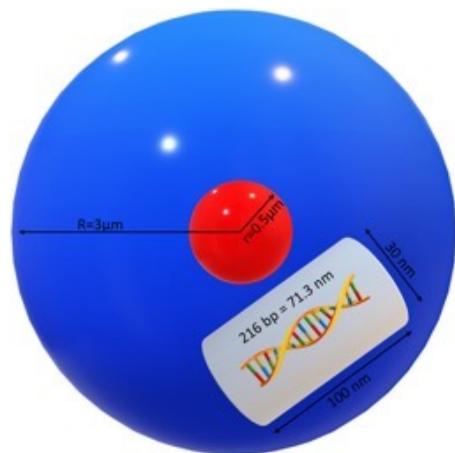
- Recommendation: use **Geant4-DNA physics constructors**
 - `G4EmDNAPhysics_option2`
 - `G4EmDNAPhysics_option4`
 - `G4EmDNAPhysics_option6`

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



- 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/konxat/complexDNA>

DNA damage classification

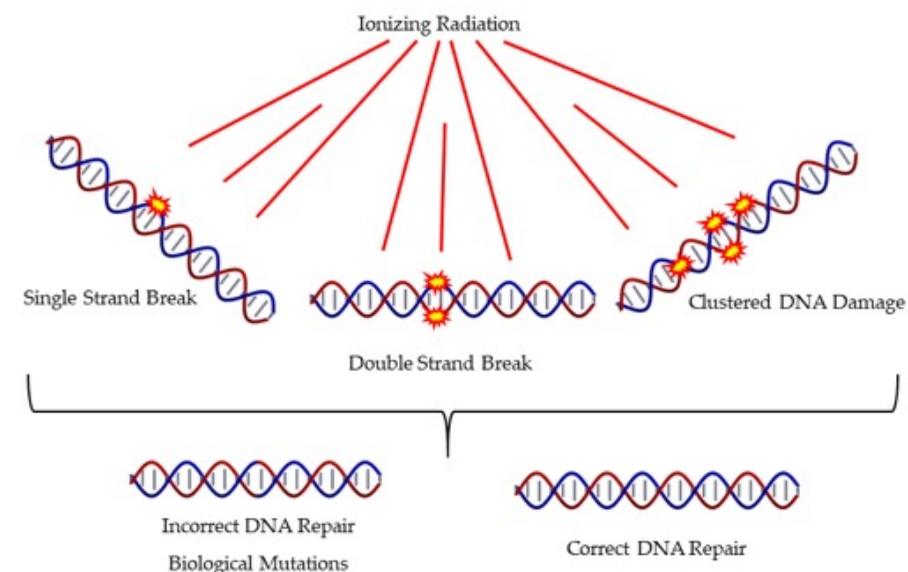
- Direct damage**

occurs when energy from physical processes is deposited near a DNA molecule.

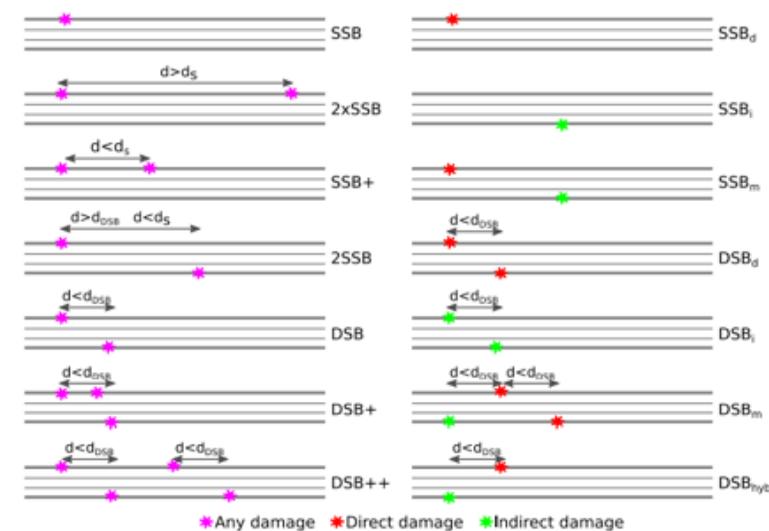
In molecular dna, we associate damage either with a **strand** molecule (sugar or phosphate) or a **base** molecule.

- Indirect damage**

is scored when a chemical reaction leads to a strand break.



[K. P. Chatzipapas, et al., Cancers 2020, 12\(4\), 799](#)

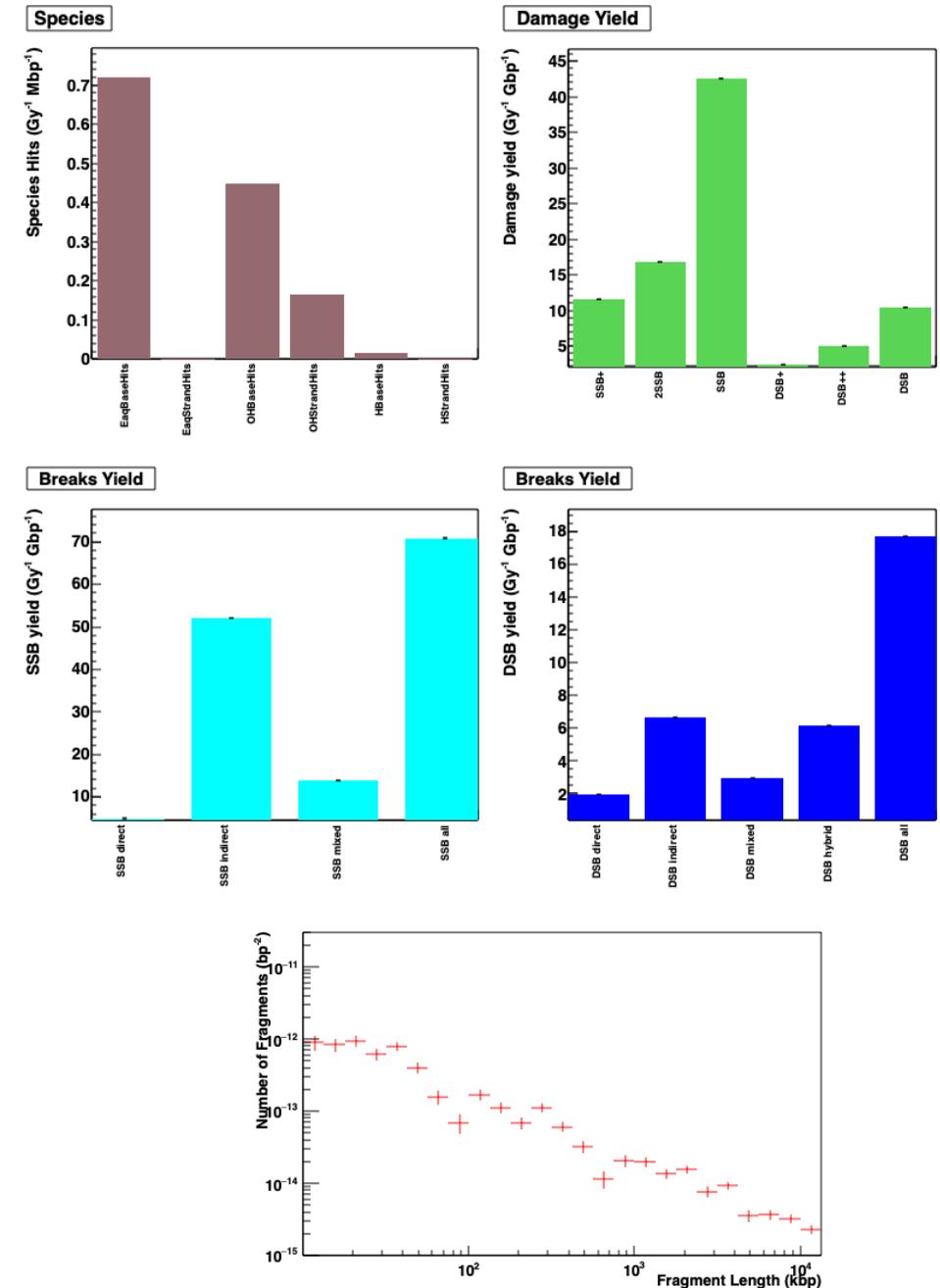


[N. Lampe, PhD thesis, 2017](#)

Results of the simulation: the ROOT macros

- **Species hits** (/Gy/Mbp)
defined as the name of radical species together with the DNA reaction
e.g. **EaqStrandHits** is $e^-_{aq} + \text{DNA backbone}$
- **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:

$$SF(t) = \exp(-L_f(t)) = \exp\left(-\int_0^t (\beta_1\lambda_1L_1(t) + \beta_2\lambda_2L_2(t) + \gamma\eta[L_1(t) + L_2(t)]^2)dt\right)$$

- $L_1(t)$ 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):
 - λ_1 , λ_2 , and η 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):
 - β_1 , β_2 , and γ 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)

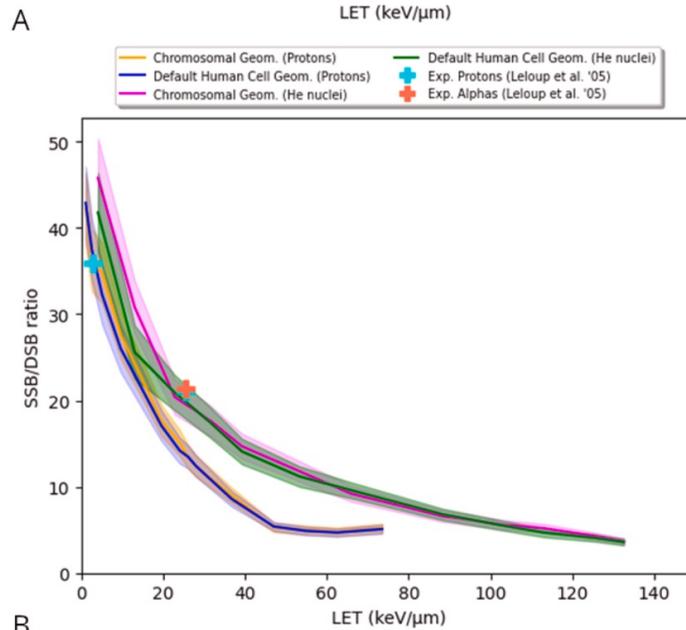
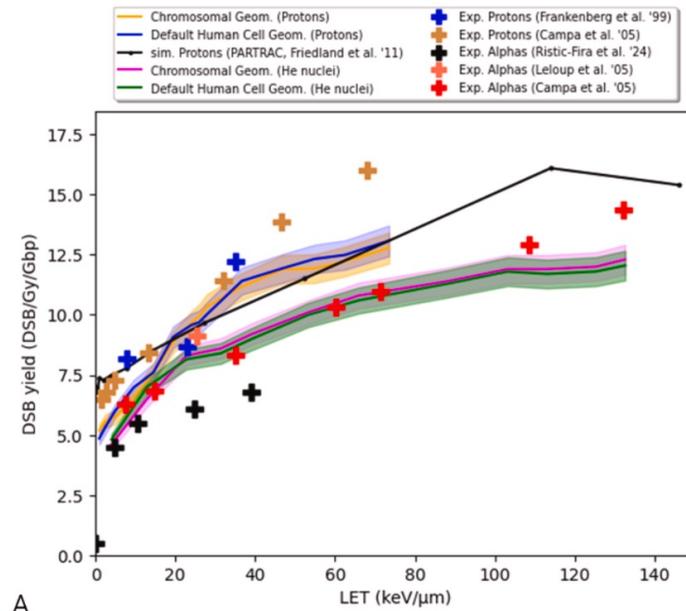
$$SF(t) = \exp(-L_f(t)) = \exp\left(-\int_0^t (\beta_1\lambda_1L_1(t) + \beta_2\lambda_2L_2(t) + \gamma\eta[L_1(t) + L_2(t)]^2)dt\right)$$

■ **Late damage** estimation with dedicated Python scripts (*.py) (3/3)

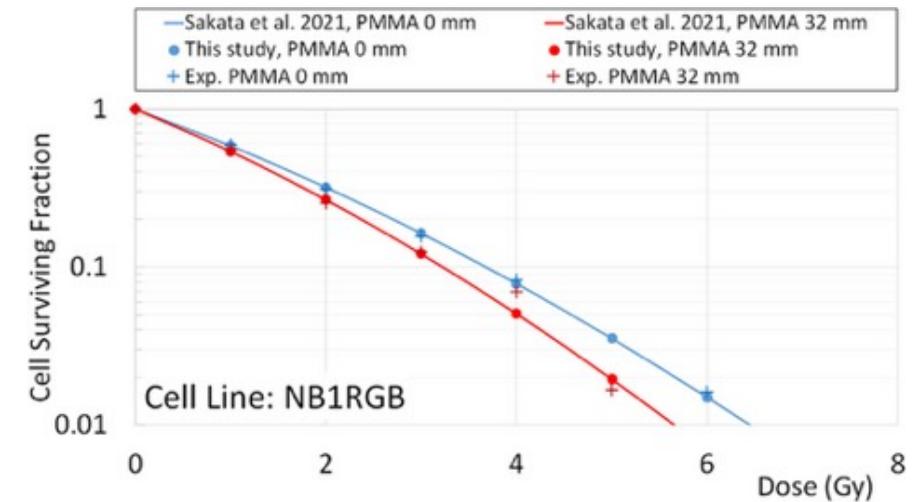
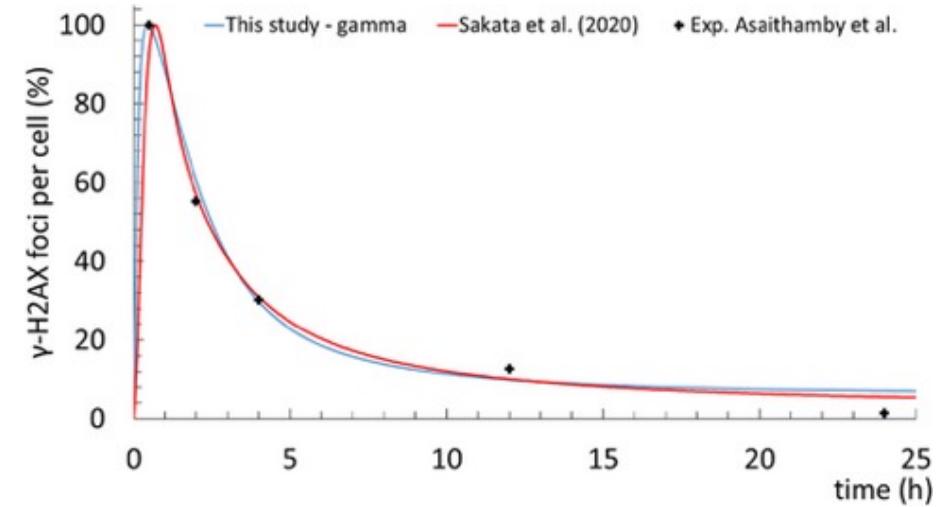
- To use the tool, the procedure is typical for python scripts. In a terminal the user can type from the `repair_survival_models` directory:
`python3 molecularDNAsurvival.py`
- **Several parameters** need to be defined (indicative values):
 - `iRootFile` = `"/path/to/molecular-dna.root"`
 - `outputFile` = `"/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 $\lambda_1, \lambda_2, \eta, \theta_1, \theta_2, \gamma$
 - `Lamb2` = `1`
 - `Eta` = `7.50595e-06`
 - `Beta1` = `0.0193207`
 - `Beta2` = `0`
 - `gamma` = `0.189328`

Examples of verification & validation

DSB & SSB/DSB vs LET:
 molecular dna & measurements



Damage repair & cell survival: molecular dna & measurements



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



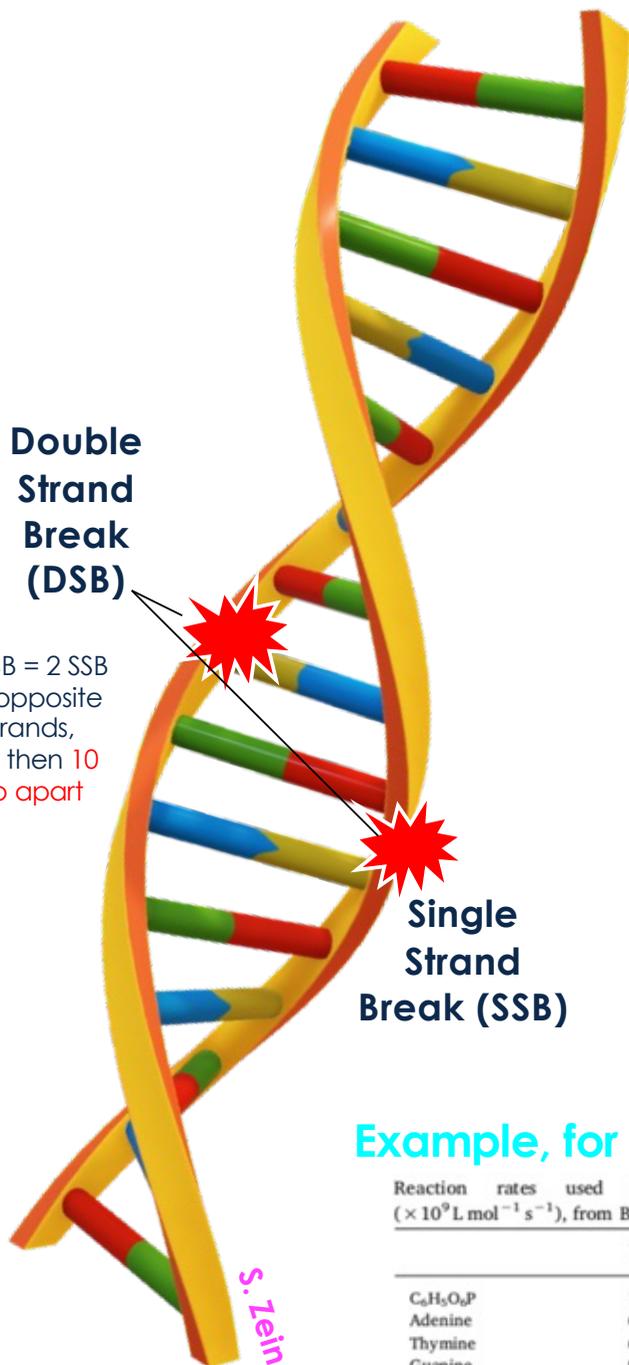
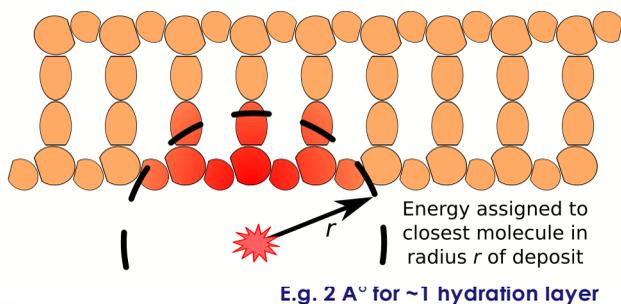
Important: parameter choice

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:



1 DSB = 2 SSB on opposite strands, less than 10 bp apart

Example, for item 1:

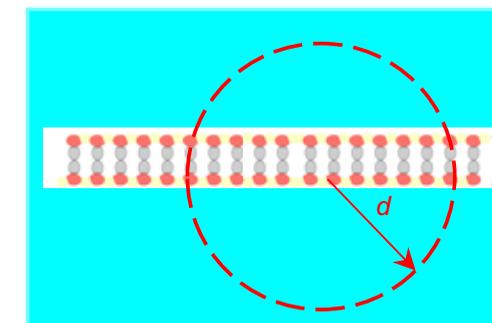
Reaction rates used between radicals and DNA components ($\times 10^9 \text{ L mol}^{-1} \text{ s}^{-1}$), from Buxton et al. [65].

	$\cdot\text{OH}$	$\text{H}\cdot$	$\epsilon_{\text{aq}}^{\cdot}$
$\text{C}_5\text{H}_2\text{O}_2\text{P}$	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

NON-DIRECT damage induction

1. Choice of G4DNA chemistry constructor
 - Including reactions with DNA components
2. Probability of non-direct SSB induction
 - $\cdot\text{OH}$ 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 3:



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 1000000
```

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

```
# 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/indirectEqBaseChance 1.0
/dnadamage/indirectEqStrandChance 0.0
/dnadamage/inductionEqChance 0.0
```

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

```
# 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
```

Shoot particles

```
# Beam on
/run/beamOn 2
```

UI commands of molecular dna (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 molecular dna (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 molecular dna (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 $\in [0,1]$ of a $\bullet\text{OH}$ damaging a base.
- `/dnadamage/indirectOHStrandChance <d>`
Chance $\in [0,1]$ of a $\bullet\text{OH}$ damaging a sugar-phosphate moiety.
- `/dnadamage/inductionOHChance <d>`
Chance $\in [0,1]$ of a reaction between a base and $\bullet\text{OH}$ yielding a strand break.

UI commands of molecular dna (4/4) (FYI only)

Damage related commands (2/2)

- **/dnadamage/indirectHBaseChance <d>**
Chance $\in [0,1]$ of a H^\bullet damaging a base.
- **/dnadamage/indirectHStrandChance <d>**
Chance $\in [0,1]$ of a H^\bullet damaging sugar-phosphate moiety.
- **/dnadamage/inductionHChance <d>**
Chance $\in [0,1]$ of a reaction between a base and H^\bullet yielding a strand break.
- **/dnadamage/indirectEaqBaseChance <d>**
Chance $\in [0,1]$ of a e^-_{aq} damaging a base.
- **/dnadamage/indirectEaqStrandChance <d>**
Chance $\in [0,1]$ of a e^-_{aq} damaging sugar-phosphate moiety.
- **/dnadamage/inductionEaqChance <d>**
Chance $\in [0,1]$ 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 « molecular dna » advanced example

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

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

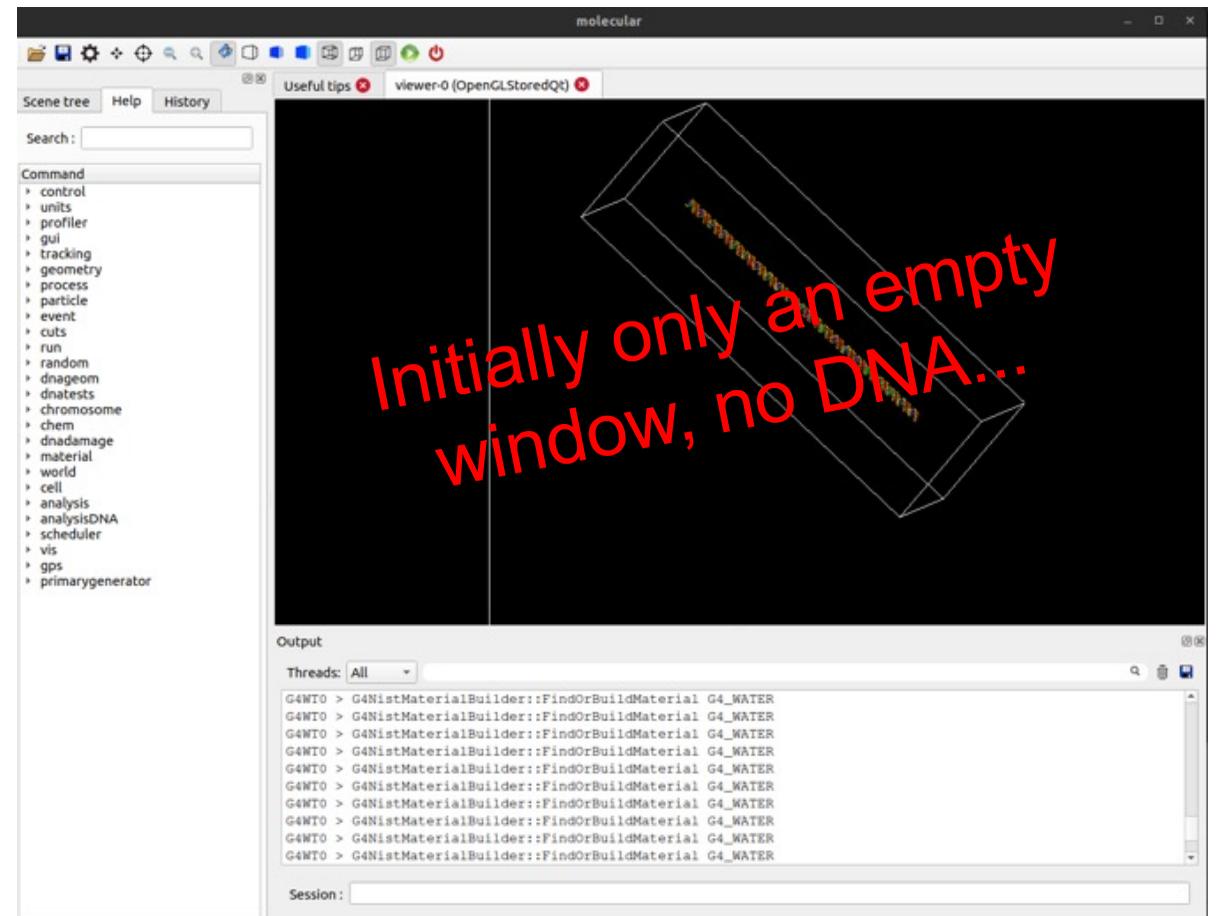
← **make -jN** if you have **N** cores

Note: download of geometry files might be slow

```
[ 2%] Building CXX object CMakeFiles/molecular.dir/molecular.cc.o
[ 5%] Building CXX object CMakeFiles/molecular.dir/src/ActionInitialization.cc.o
[ 8%] Building CXX object CMakeFiles/molecular.dir/src/AnalysisManager.cc.o
[ 11%] Building CXX object CMakeFiles/molecular.dir/src/AnalysisMessenger.cc.o
[ 13%] Building CXX object CMakeFiles/molecular.dir/src/ChemistryList.cc.o
[ 16%] Building CXX object CMakeFiles/molecular.dir/src/ChromosomeFactory.cc.o
[ 19%] Building CXX object CMakeFiles/molecular.dir/src/ChromosomeHit.cc.o
[ 22%] Building CXX object CMakeFiles/molecular.dir/src/ChromosomeMapper.cc.o
[ 25%] Building CXX object CMakeFiles/molecular.dir/src/ChromosomeMessenger.cc.o
[ 27%] Building CXX object CMakeFiles/molecular.dir/src/CylindricalChromosome.cc.o
[ 30%] Building CXX object CMakeFiles/molecular.dir/src/DNAGeometry.cc.o
[ 33%] Building CXX object CMakeFiles/molecular.dir/src/DNAGeometryMessenger.cc.o
[ 36%] Building CXX object CMakeFiles/molecular.dir/src/DNAHashing.cc.o
[ 38%] Building CXX object CMakeFiles/molecular.dir/src/DNAHit.cc.o
[ 41%] Building CXX object CMakeFiles/molecular.dir/src/DNAWorld.cc.o
[ 44%] Building CXX object CMakeFiles/molecular.dir/src/DamageModel.cc.o
[ 47%] Building CXX object CMakeFiles/molecular.dir/src/DamageModelMessenger.cc.o
[ 50%] Building CXX object CMakeFiles/molecular.dir/src/DetectorConstruction.cc.o
[ 52%] Building CXX object CMakeFiles/molecular.dir/src/DetectorMessenger.cc.o
[ 55%] Building CXX object CMakeFiles/molecular.dir/src/EllipticalChromosome.cc.o
[ 58%] Building CXX object CMakeFiles/molecular.dir/src/EventAction.cc.o
[ 61%] Building CXX object CMakeFiles/molecular.dir/src/IRTDamageReactionModel.cc.o
[ 63%] Building CXX object CMakeFiles/molecular.dir/src/OctreeNode.cc.o
[ 66%] Building CXX object CMakeFiles/molecular.dir/src/ParallelWorldPhysics.cc.o
[ 69%] Building CXX object CMakeFiles/molecular.dir/src/PhysicsList.cc.o
[ 72%] Building CXX object CMakeFiles/molecular.dir/src/PlacementVolumeInfo.cc.o
[ 75%] Building CXX object CMakeFiles/molecular.dir/src/PrimaryGeneratorAction.cc.o
[ 77%] Building CXX object CMakeFiles/molecular.dir/src/RodChromosome.cc.o
[ 80%] Building CXX object CMakeFiles/molecular.dir/src/RunAction.cc.o
[ 83%] Building CXX object CMakeFiles/molecular.dir/src/SphericalChromosome.cc.o
[ 86%] Building CXX object CMakeFiles/molecular.dir/src/StackingAction.cc.o
[ 88%] Building CXX object CMakeFiles/molecular.dir/src/SteppingAction.cc.o
[ 91%] Building CXX object CMakeFiles/molecular.dir/src/TimeStepAction.cc.o
[ 94%] Building CXX object CMakeFiles/molecular.dir/src/UtilityFunctions.cc.o
[ 97%] Building CXX object CMakeFiles/molecular.dir/src/VirtualChromosome.cc.o
[100%] Linking CXX executable molecular
[100%] Built target molecular
```

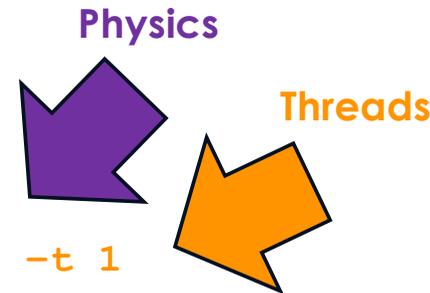
Hands-on practice with the « molecular dna » advanced example

- Run molecular dna 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



```

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 linuxx86_64gcc 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'
-----

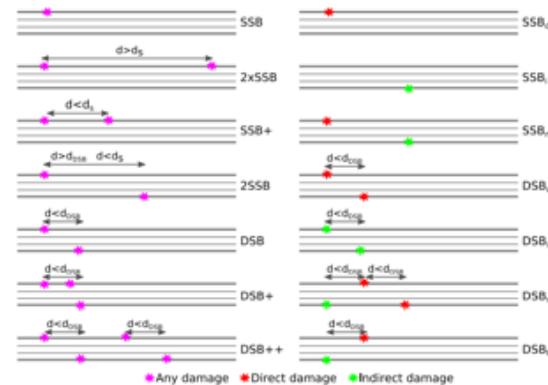
root [0] .X cylinders.C
hadd Target file: molecular-dna.root
hadd compression setting for all output: 1
hadd Source file 1: molecular-dna_t0.root
hadd Target path: molecular-dna.root:/
hadd Target path: molecular-dna.root:/hists
hadd Target path: molecular-dna.root:/tuples
Particle : e      Energy [/MeV] : 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

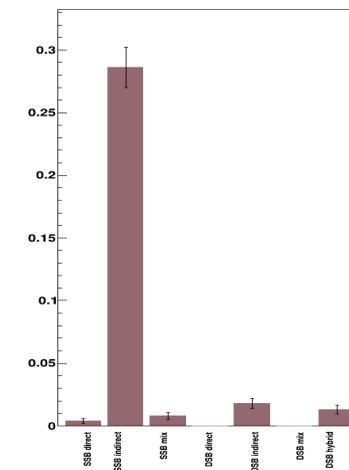
SSB : 0.237          error : 0.0148003
SSBp : 0.036         error : 0.00589396
2SSB : 0.025         error : 0.00493957

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

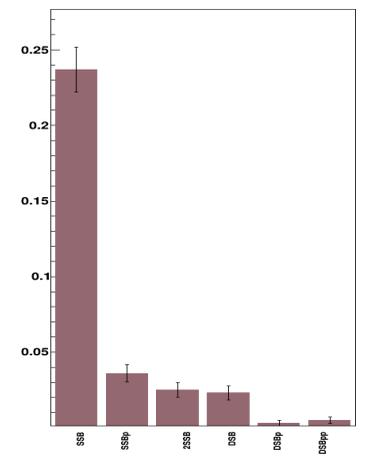
- Results can be analyzed using ROOT (depending on the version these commands may differ)
 - ROOT is already installed on your Geant4 virtual machine
 - `root`
 - Within ROOT
 - `.X cylinders.C`
 - Histograms will be plotted



Break Source Frequency

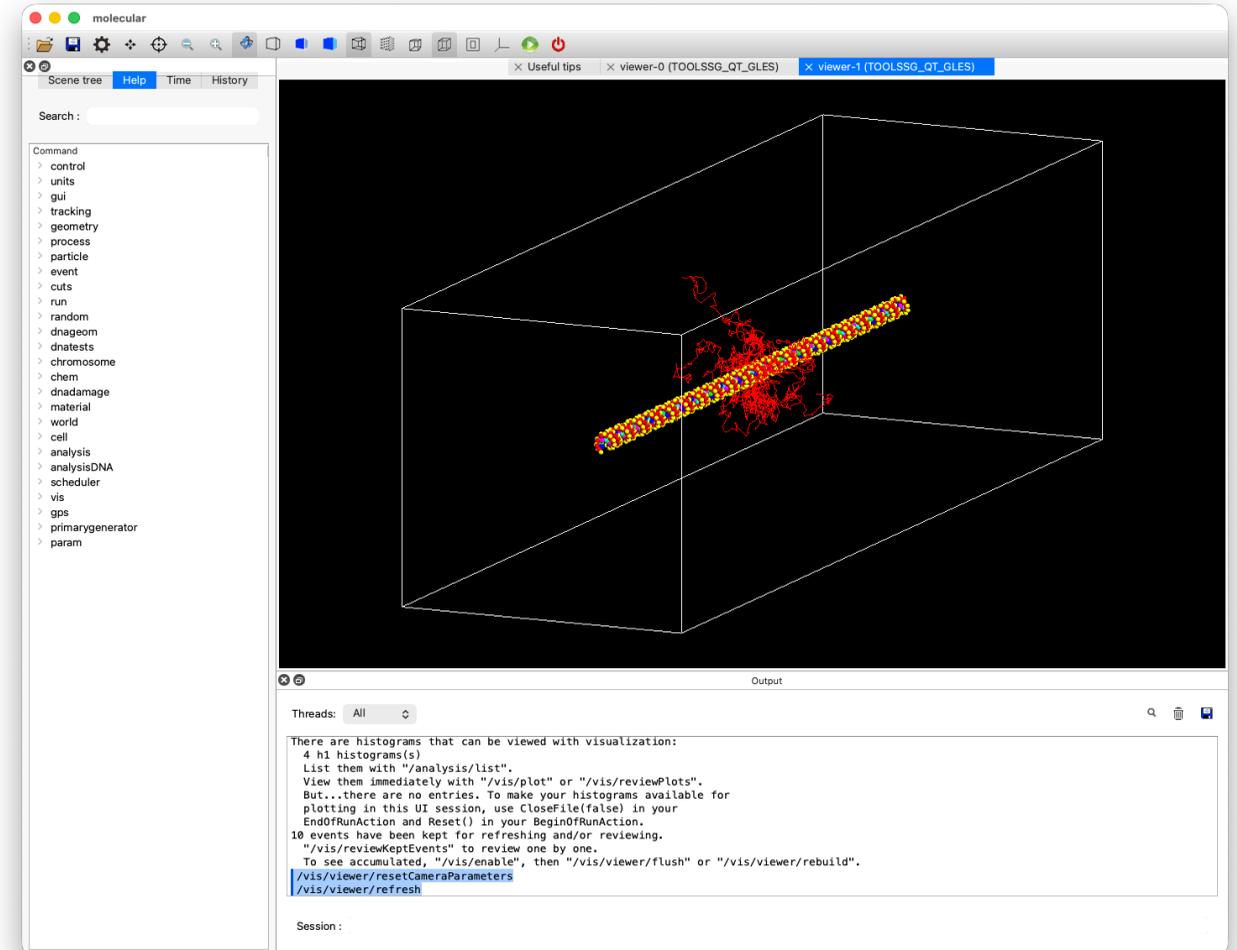


Break Complexity Frequency



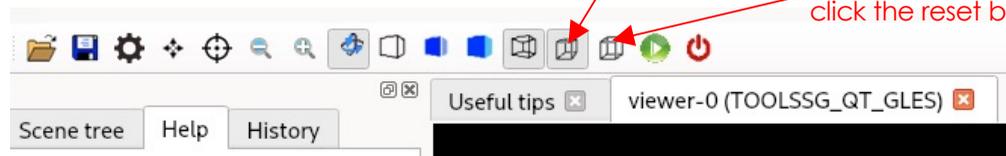
Hands-on practice with the « moleculardna » advanced example

1. In the terminal type: `./molecular -t 1 -v 1`
2. In the Qt window type: `/control/execute fiber.mac`
3. Observe the DNA geometry, using the mouse to zoom in & out!



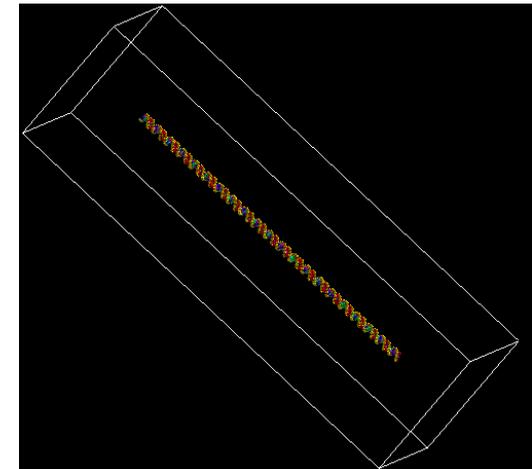
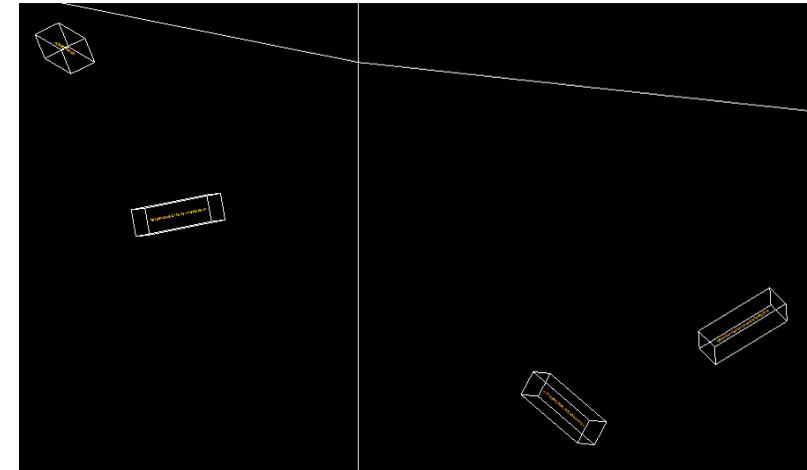
Enable perspective on Qt viewer to see your geometry

If it does not show anything, click the reset button



Hands-on practice with the « molecular dna » advanced example

1. In the **geometries** dir., create a copy of **prisms200k_r3000.txt**
2. Open your copy of **prisms200k_r3000.txt** (e.g. **prisms200k_r3000_c.txt**):
 - Keep only the first 20 prisms (0-19)(up-to line 21 strictly)
3. Open **cylinders.mac**:
 - Comment the last line: **/run/beamOn**
 - Un-comment line **/control/execute vis.mac**
 - And save
4. In the terminal type: **./molecular -t 1 -v 1**
5. In the Qt window type:
/control/execute cylinders.mac
6. Observe the DNA geometry, using the mouse to zoom in & out!



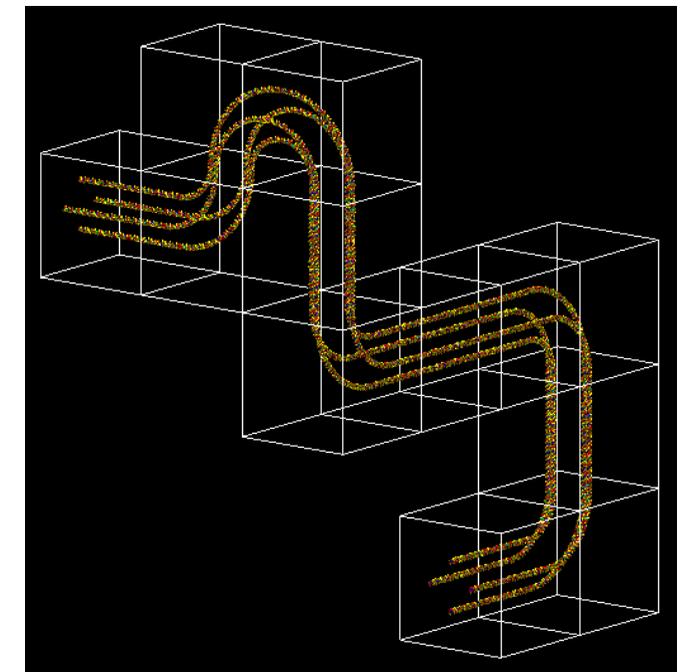
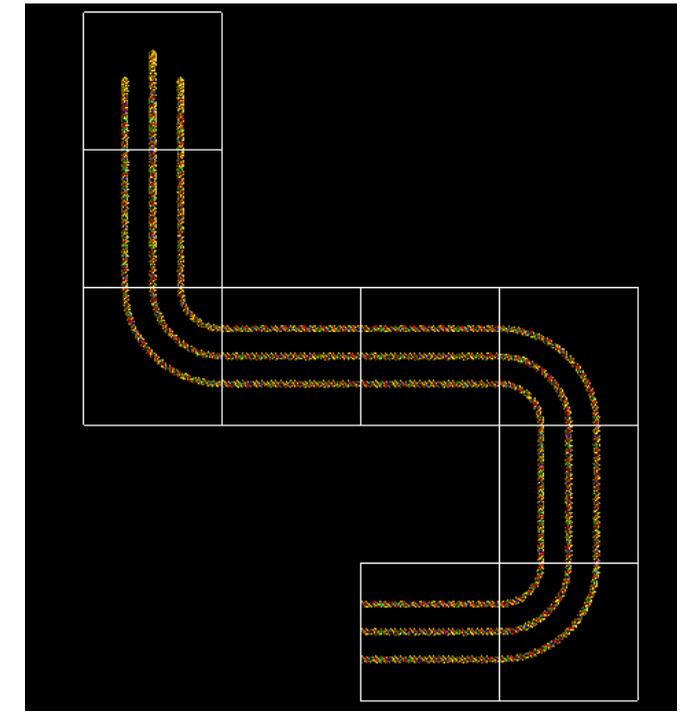
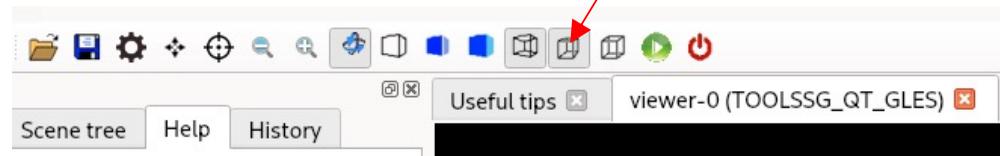
Hands-on practice with the « moleculedna » advanced example

- 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
```

Enable perspective on Qt viewer
to see your geometry



Documentation

moleculardna documentation
<https://moleculardna.org>

moleculardna

Search moleculardna

moleculardna on GitHub The Geant4-DNA Project

moleculardna

Radiation-induced DNA damage simulations in Geant4.

moleculardna is a Geant4-DNA example built to allow easy simulation of radiation-induced DNA damage with flexible geometries and well defined damage parameters.

Get started right away in `geant4/examples/extended/medical/dna/moleculardna` with a library of pre-existing geometries, or dive into the documentation.

Important : This version of moleculardna (Geant4 11.2.beta) is proposed as a **beta** version and may still contain bugs, thank you for your understanding. Please contact us for further detail. This example is for demonstration purpose and acts as an introductory tool to help users create their own applications; therefore users are advised to adapt the simulation parameters in their applications accordingly.

(a) DNA double helix molecular structure

(b) Chromatin fiber segments

(c) Fibroblast cell nucleus

Hilbert curve

human cell example

Get started from example See publications Available geometries

Want to know more about how it all works? You'll want to visit our Overview page.

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

FractalDNA

Search docs

Structure Models
DNA Models
Examples
API Reference

FractalDNA

FractalDNA is a Python package to make DNA geometries that can be joined together like jigsaw puzzles. Both simple, sections of DNA and Solenoidal DNA can be built. This module was built to enable DNA-level simulations to be run in `Geant4-DNA`, part of the `Geant4` project.

Structure models define the large scale structure of DNA, seeded from fractals. An example seeding fractal is below:

DNA Models provide straight and curved segments that can come together to make DNA for use in simulations.

fractaldna 0.6.1

pip install fractaldna

FractalDNA is a Python package built to generate DNA geometries for simulations

Navigation

- Description du projet
- Historique des versions
- Téléchargement des fichiers

Liens de projet

- Homepage
- Documentation
- Repository

Statistiques

Statistiques de GitHub :

- Stars : 6
- Forks : 4
- Open Issues : 0
- Open PRs : 1

Description du projet

FractalDNA

Python routines for generating geometric models of DNA

FractalDNA is being converted to a package, it is under active development

FractalDNA is a Python package to make DNA geometries that can be joined together like jigsaw puzzles. Both simple, sections of DNA and Solenoidal DNA can be built. This module was built to enable DNA level simulations to be run in `Geant4-DNA`, part of the `Geant4` project.

Structure models define the large scale structure of DNA, seeded from fractals. An example seeding fractal is below:

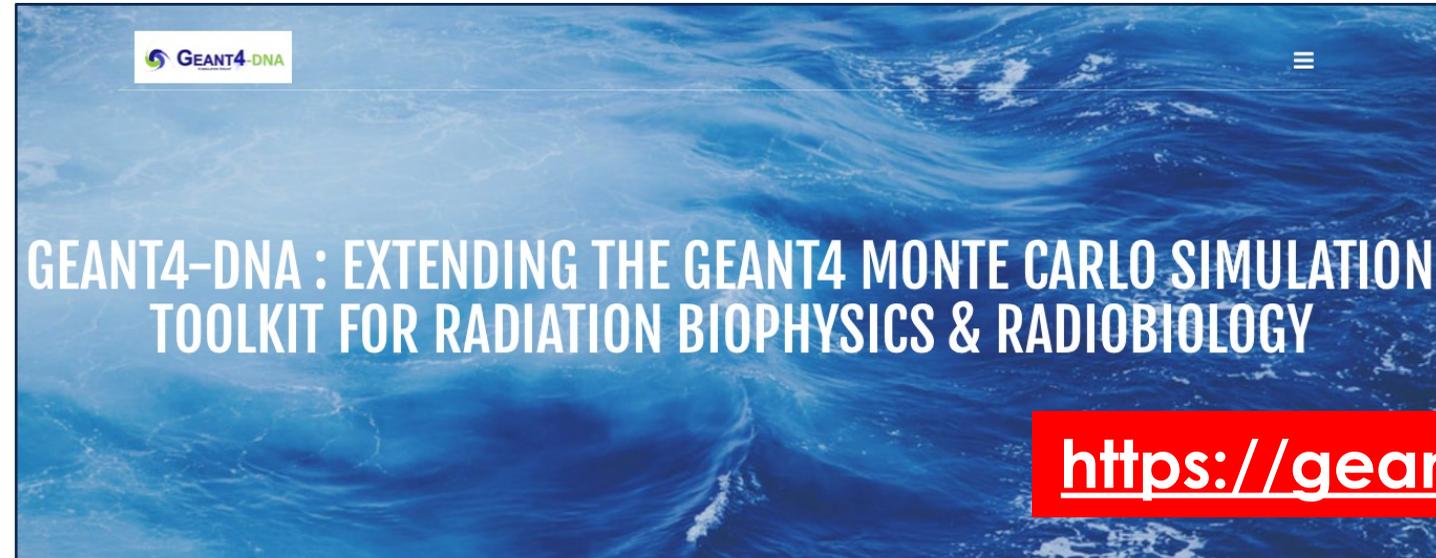
to three main sections, focused on building DNA models, high level some notebook examples, showing how the code works.

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



<https://geant4-dna.org/>

Welcome to the web page of the Geant4-DNA project !

The [Geant4](#) general purpose particle-matter Monte Carlo simulation toolkit is being extended with processes for the **modeling of biological damage induced by ionising radiation at the DNA scale**. Such developments are on-going in the framework of the Geant4-DNA project. This project was originally initiated by the [European Space Agency \(ESA\)](#). Developments are undertaken by an [international collaboration](#), coordinated since 2008 by the [National Institute of Nuclear and Particle Physics \(IN2P3\)](#) of the [National Centre for Scientific Research \(CNRS\)](#) in France, in collaboration with the [Geant4@IN2P3](#) activities.

Once published, all developments are freely accessible in **full open access** through the [Geant4 toolkit](#) or through our [Geant4 Virtual Machine](#).

Last posts

Dec. 20, 2024: Geant4-DNA in the [Winter 2024 issue of the EFOMP European Medical Physics News](#).

Dec. 17, 2024: The [Geant4 11.3.0 LP2i Virtual Machine](#) with [AlmaLinux 9](#) has been released, see [link](#).

You may follow us on [X](#) and [Bluesky](#)