Hands-On on DNA damage quantification

geant4-dna.org

The « moleculardna » extended example

Konstantinos CHATZIPAPAS et al.
LaTIM, Inserm, University of Brest, France
konstantinos.chatzipapas@univ-brest.fr

XI International Geant4 School
University of Pavia, Pavia, Italy
14-19 January, 2024

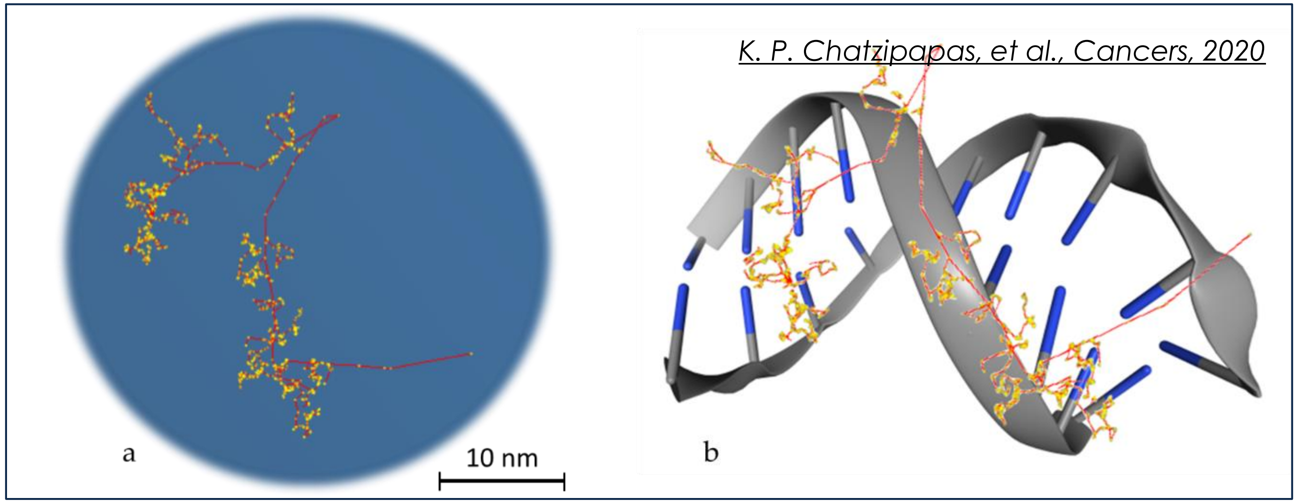
Geant4 version 11.2

Released in December 2023

The « moleculardna » extended example

- Try the « **moleculardna** » example, which **combines**
  - Geant4-DNA Physics models
  - Geant4-DNA Chemistry models &
  - Geant4-DNA DNA-scale geometrical models of biological targets

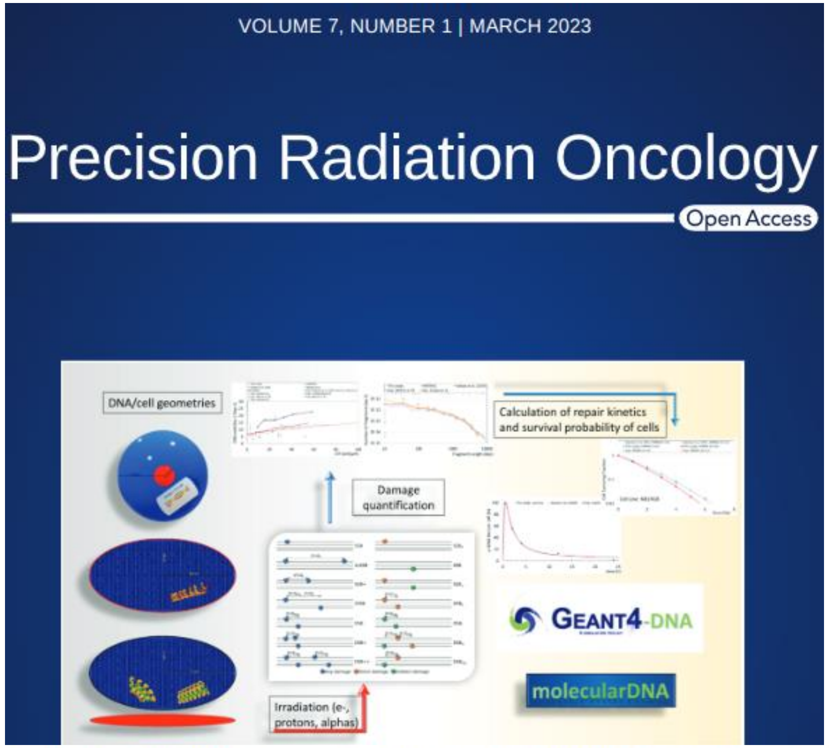
- This is a Geant4 **extended** example and it is located in
  `$G4INSTALL/examples/extended/medical/dna`



K. P. Chatzipapas, et al., Cancers, 2020

The « moleculardna » extended example

- This example aims to demonstrate how the Geant4-DNA toolkit can be used to **quantify DNA damage** induced by ionising irradiation
- Full macro file control of simulation (No C++ knowledge needed)
- Physical interaction and Chemistry models
- Several types of **DNA geometries** are included
  - Cylinders
  - Bacterial DNA
  - Human cells
- Both **direct and indirect** damage can be calculated, taking into account
  - Physics
  - Chemistry (physico-chemical and chemical stages)
- The **complexity of damage** can be investigated



VOLUME 7, NUMBER 1 | MARCH 2023

Precision Radiation Oncology

Open Access

DNA/cell geometries
Calculation of repair kinetics and survival probability of cells
Damage quantification
Irradiation (e-, protons, alphas)
GEANT4-DNA
molecularDNA

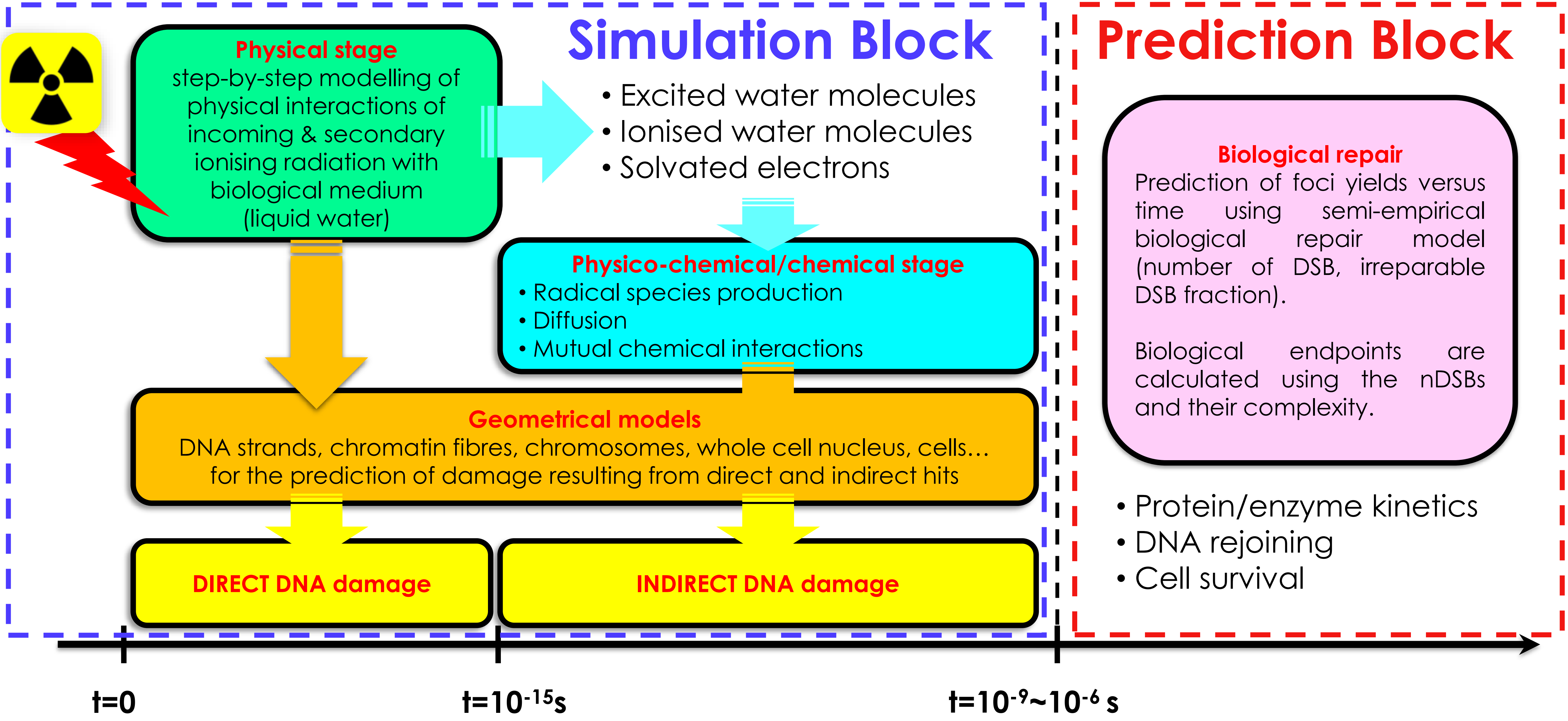
Simulation of DNA damage using Geant4-DNA: an overview of the "molecularDNA" example application

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 ^

https://doi.org/10.1002/pro6.1186
and references therein

molecularDNA approach



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 nDSBs and their complexity.

- Protein/enzyme kinetics
- DNA rejoining
- Cell survival

t=0

t=10^{-15}s

t=10^{-9}~10^{-6} s

Physical stage

Recommendation: use Geant4-DNA physics constructors

- G4EmDNAPhysics_**option2**
- G4EmDNAPhysics_**option4**
- G4EmDNAPhysics_**option6**

```
/chem/activate true/false
```

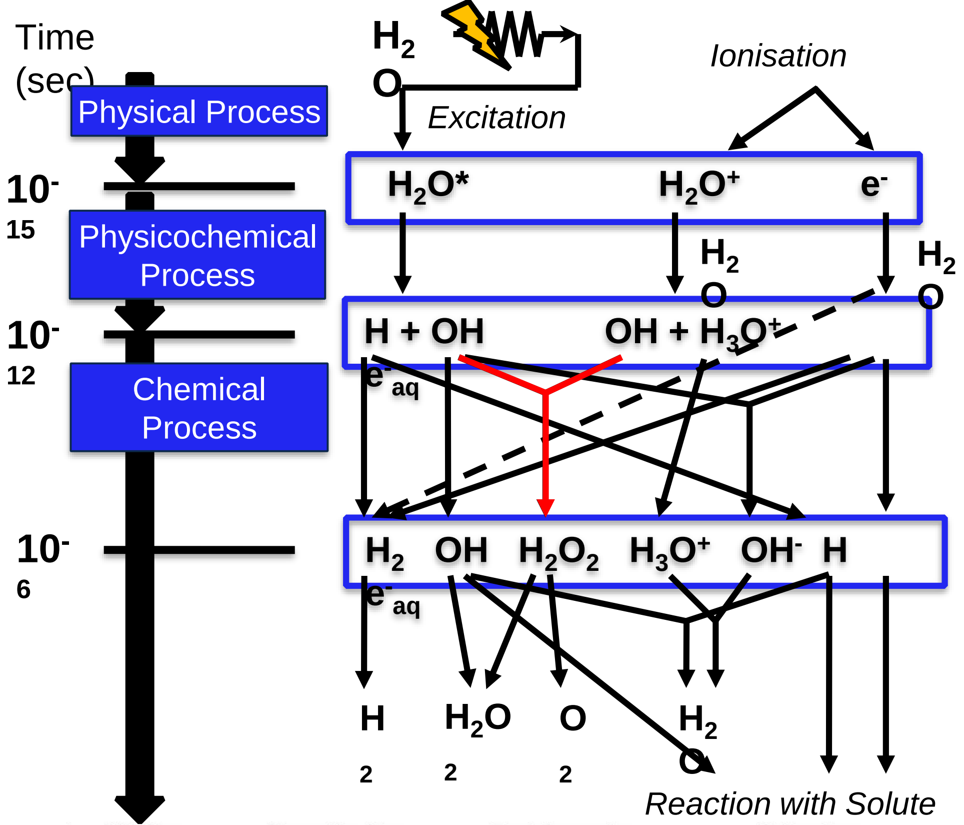
Physicochemical stage

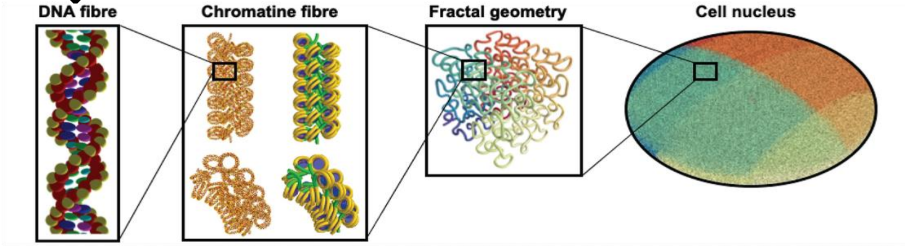
During this stage, free radicals are produced.

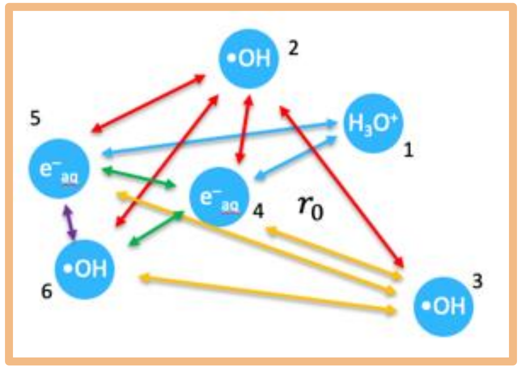
Chemical stage

Independent Reaction Times (« IRT ») approach

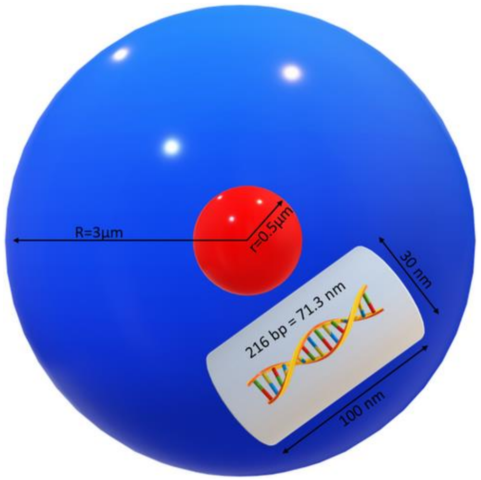
- From the 1980's by Clifford, Green et al., widely used today.
- Iterative process where the approximation of « independent pairs » is assumed: calculates the reaction times between all possible pairs of reactive species, as if they were isolated.
- No longer necessary to diffuse the molecular species 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.

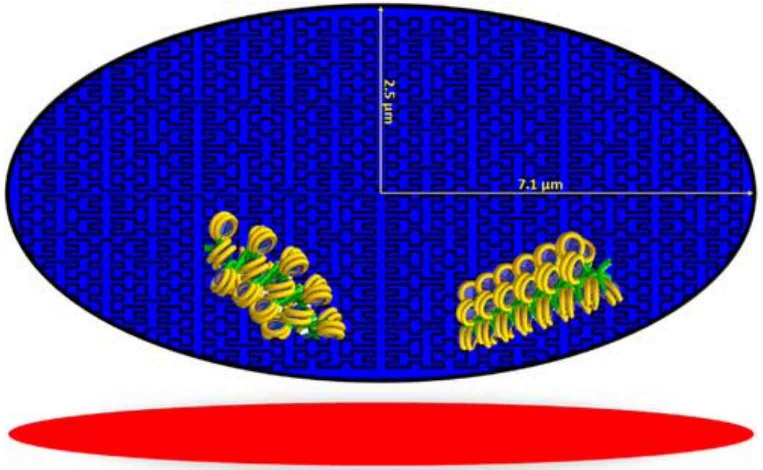


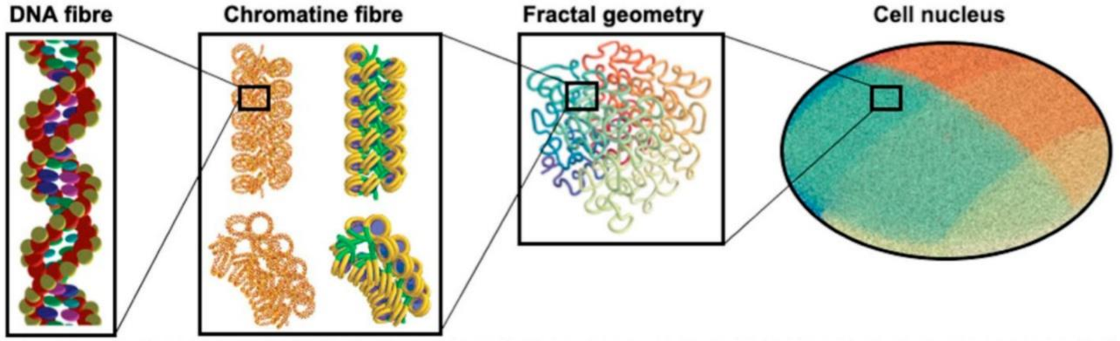


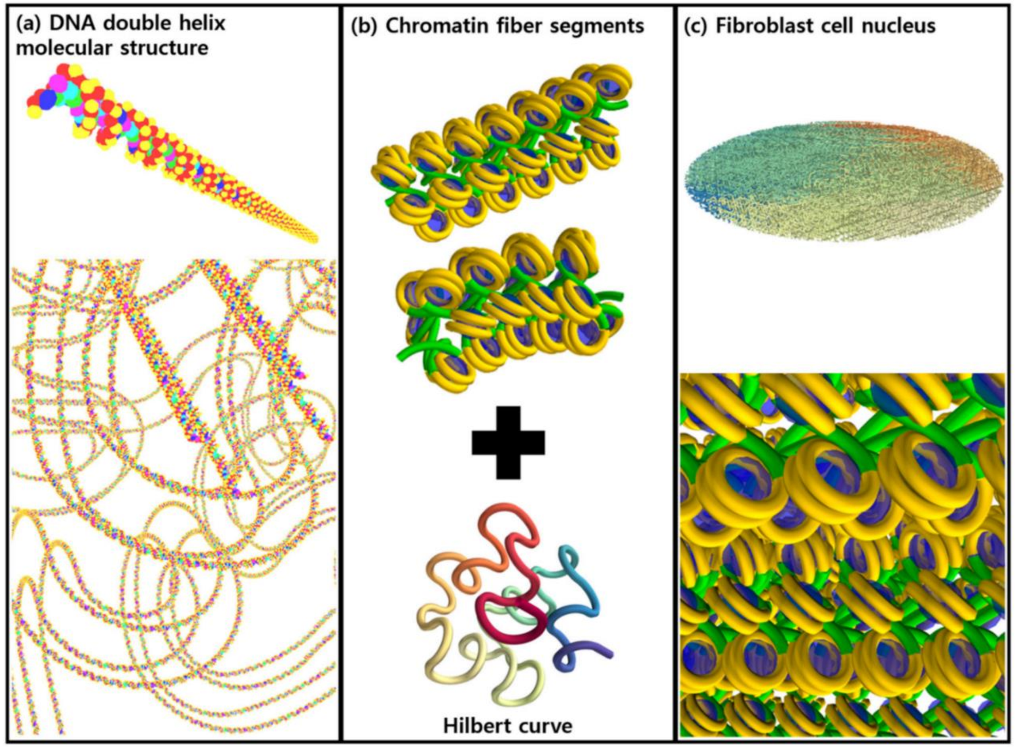


Examples of geometrical models created from the « fractalDNA » tool

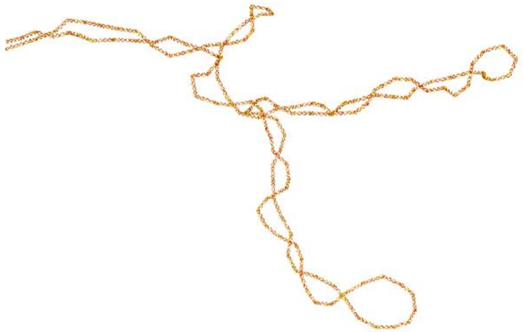




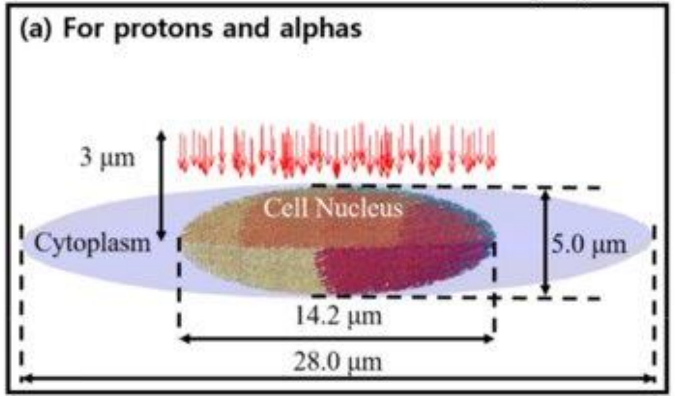




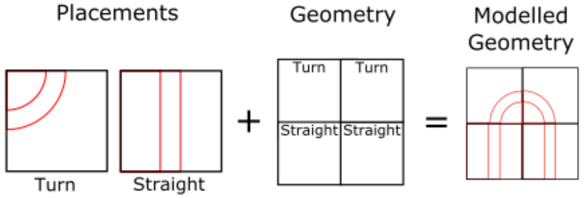
(a) DNA double helix molecular structure
(b) Chromatin fiber segments
(c) Fibroblast cell nucleus
Hilbert curve



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



The «fractalDNA» tool (by N. Lampe): open-source Python package to build DNA geometries that can be joined together like jigsaw puzzles. Users can build their own geometries based on provided examples.



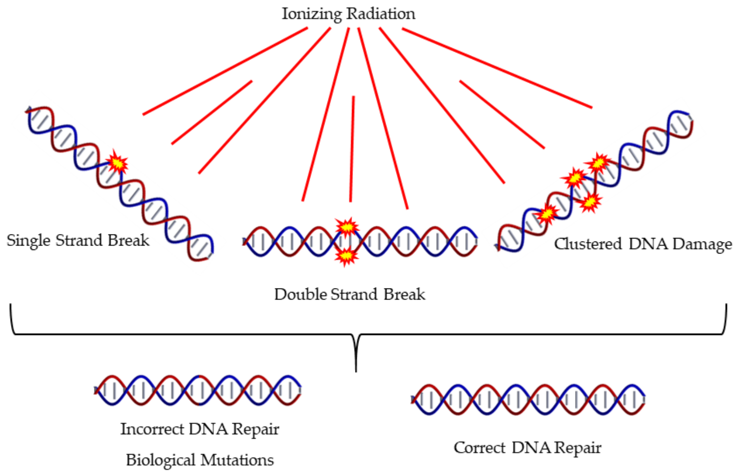
https://pypi.org/project/fractaldna/
http://natl.github.io/fractaldna/

DNA damage classification

- **Direct damage**
  occurs when energy from ionizing particles is deposited near a DNA.

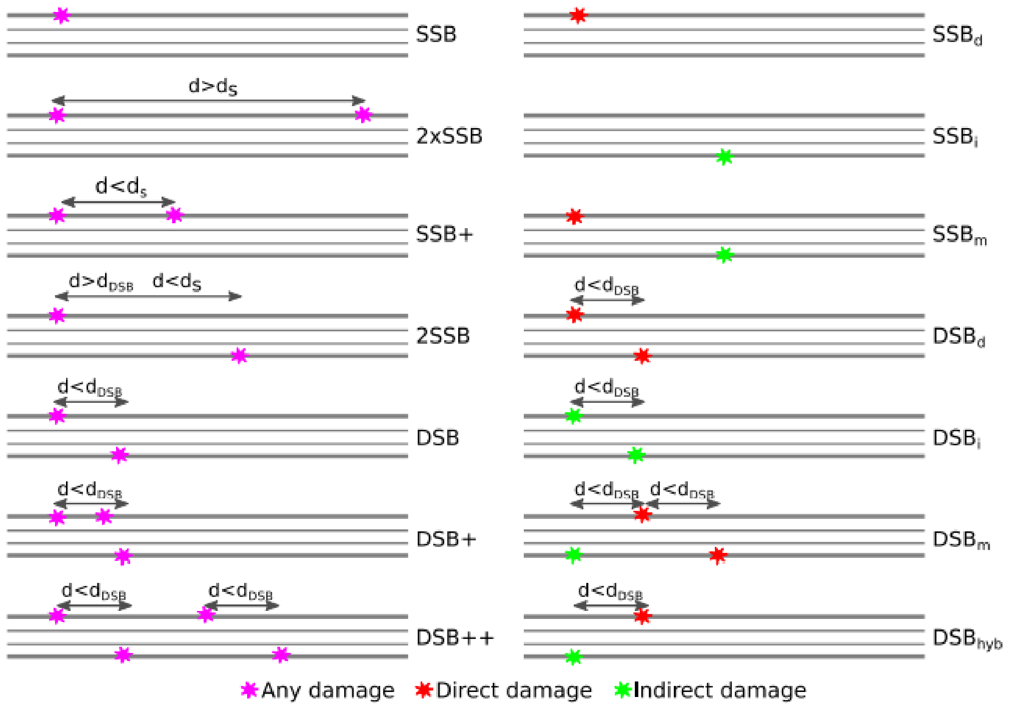
  In molecularDNA, we associate damage either with a **strand** molecule (sugar or phosphate) or a **base** molecule.

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



K. P. Chatzipapas et al. Cancers, 2020

Classification scheme

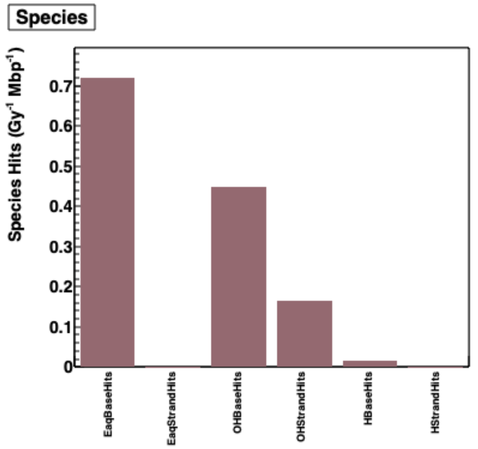


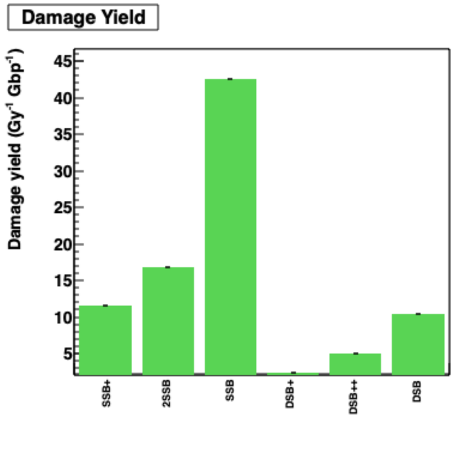
N. Lampe, PhD thesis, 2017

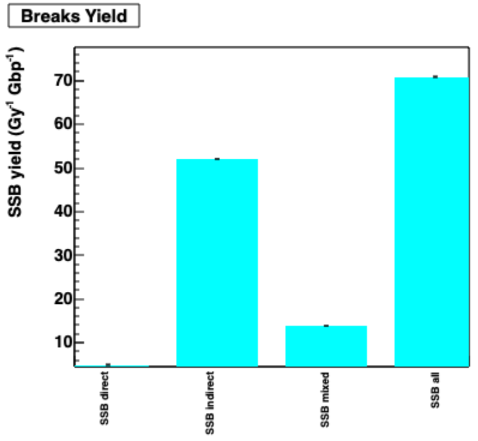
Results: ROOT output file

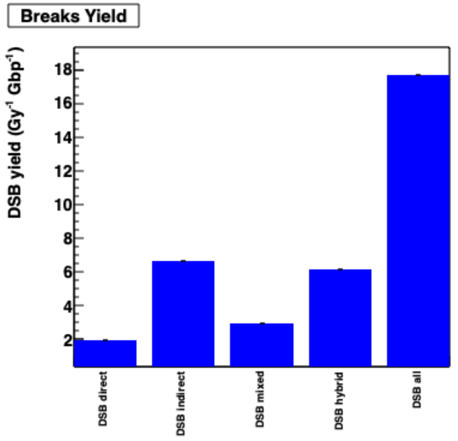
https://root.cern.ch

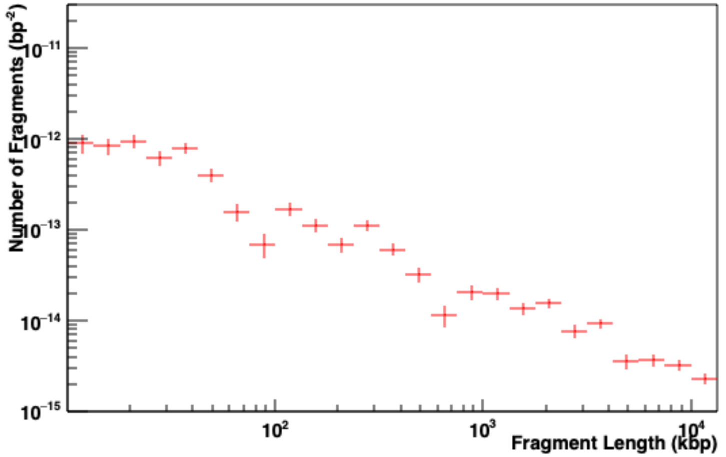
- **Species hits** (/Gy/Mbp)
  defined as the name of radical species together with the DNA reaction
  e.g. EaqStrandHits is e^{-}_{aq} + DNA backbone
- **Damage yield** (/Gy/Gbp)
  defined by DNA damage complexity
  (see classification scheme – previous slide)
- **Break yield** (/Gy/Gbp)
  shown for 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 https://geant4-dna.github.io/molecular-docs/docs/overview/results-and-analysis











List of macro files & scripts (1/4)

- **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, as well as for simulating a population of plasmids.
  - 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
    - 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) and **two alternative models** for HTB177 (lung) and MCF7 (breast) cancer cell lines (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.
  - ecoli.C
    - Analysis of simulation results of the ecoli.mac file. Prints ***Species hits***, ***Damage yield***, ***Breaks yield*** and ***Fragments distribution***.
  - human_cell.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

- Extras
  - createSDD.py
    - After the simulation, ROOT data can be converted to ***Standard DNA Damage*** (SDD) data format (see SDD format).
  - **repair_survival_models** folder
  - **phase-space** directory including information of incident particle sources

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:

```
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, Belov OV, 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, 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 SF of a cell population.
  - Mathematically, the **Survival Fraction** (SF) of cells is calculated using:

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

    - ***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*):
    - **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* $^{-1}$).

[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)

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

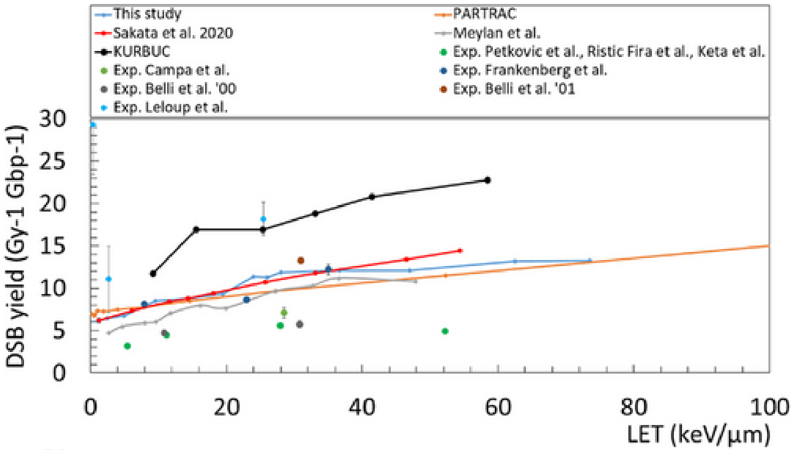
- **Late damage** estimation with dedicated Python scripts (*.py) (3/3)
  - To use the tool, the procedure is typical for python scripts. In a terminal the user can type :

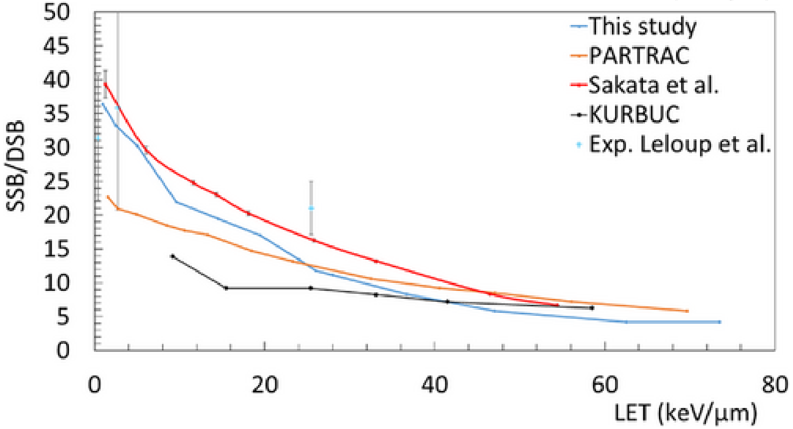
    `python3 molecularDNAsurvival.py`

  - **Several parameters** need to be defined (<u>indicative</u> 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` $\lambda_1, \lambda_2, \eta, \beta_1, \beta_2, \gamma$
    - `Lamb2 = 1`
    - `Eta   = 7.50595e-06`
    - `Beta1 = 0.0193207`
    - `Beta2 = 0`
    - `gamma = 0.189328`

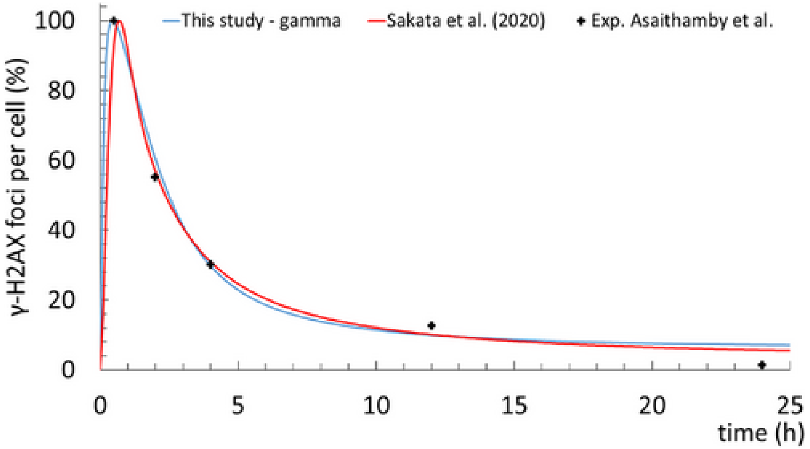
Examples of verification & validation

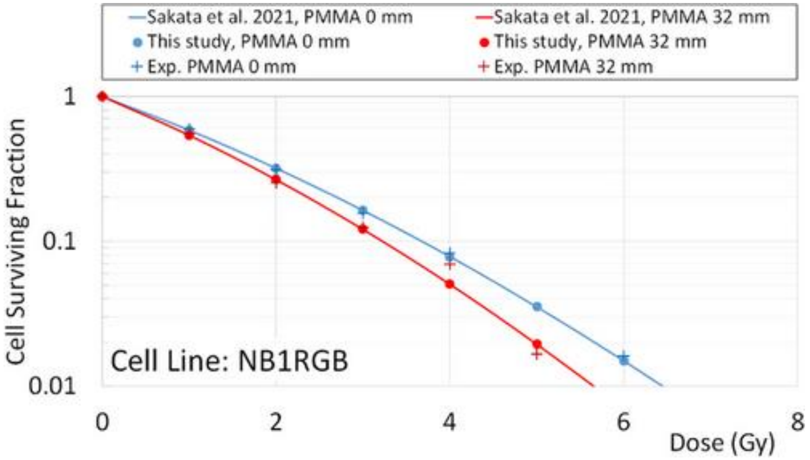
Proton irradiation: MC tools & measurements



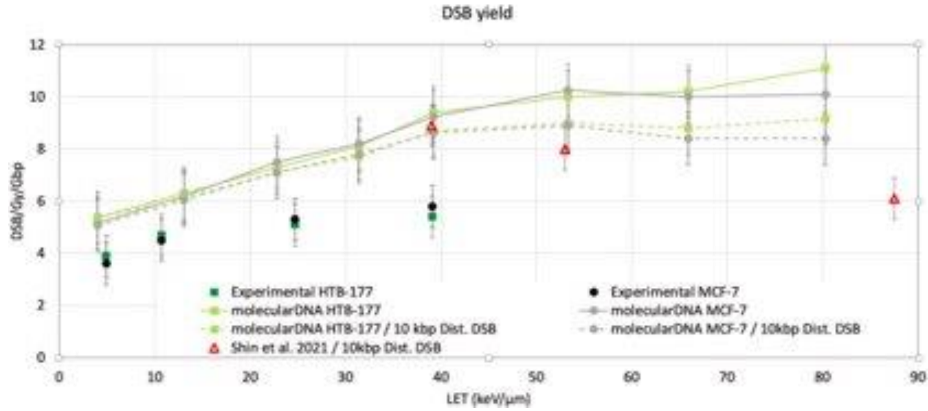


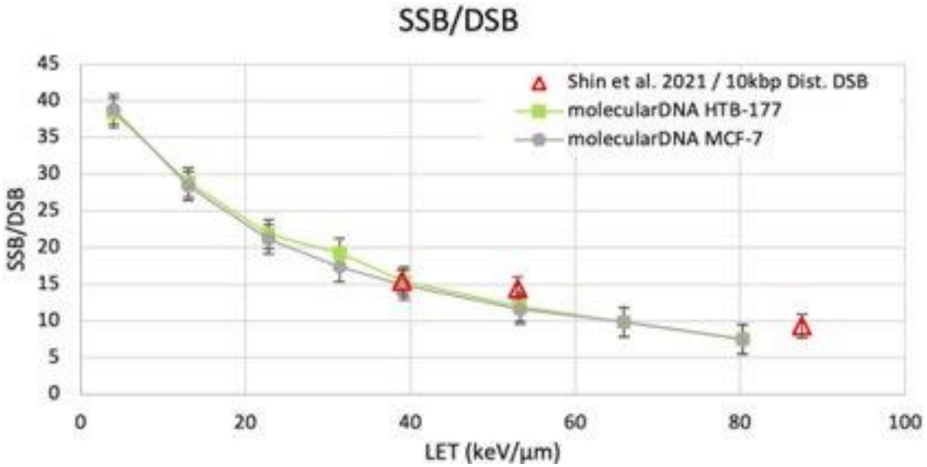
Damage repair and cell survival: moleculardna & measurements

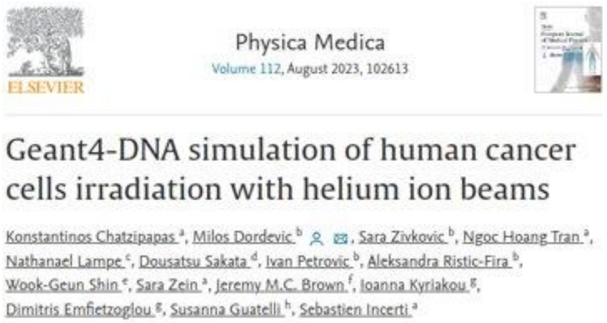




Alpha irradiation: moleculardna & measurements







Physica Medica

Volume 112, August 2023, 102613

ELSEVIER

Geant4-DNA simulation of human cancer cells irradiation with helium ion beams

Konstantinos Chatzipapas [a], Milos Dordevic [b], Sara Zivkovic [b], Ngoc Hoang Tran [a], Nathanael Lampe [c], Dousatsu Sakata [d], Ivan Petrovic [b], Aleksandra Ristic-Fira [b], Wook-Geun Shin [e], Sara Zein [a], Jeremy M.C. Brown [f], Ioanna Kyriakou [g], Dimitris Emfietzoglou [g], Susanna Guatelli [h], Sebastien Incerti [a]

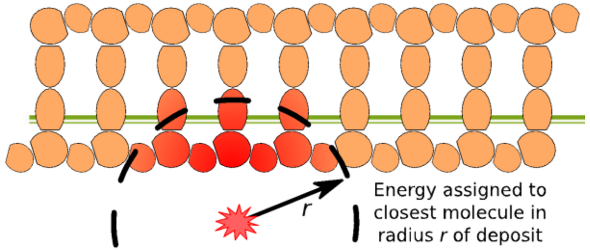
Important: parameter choice

Physics

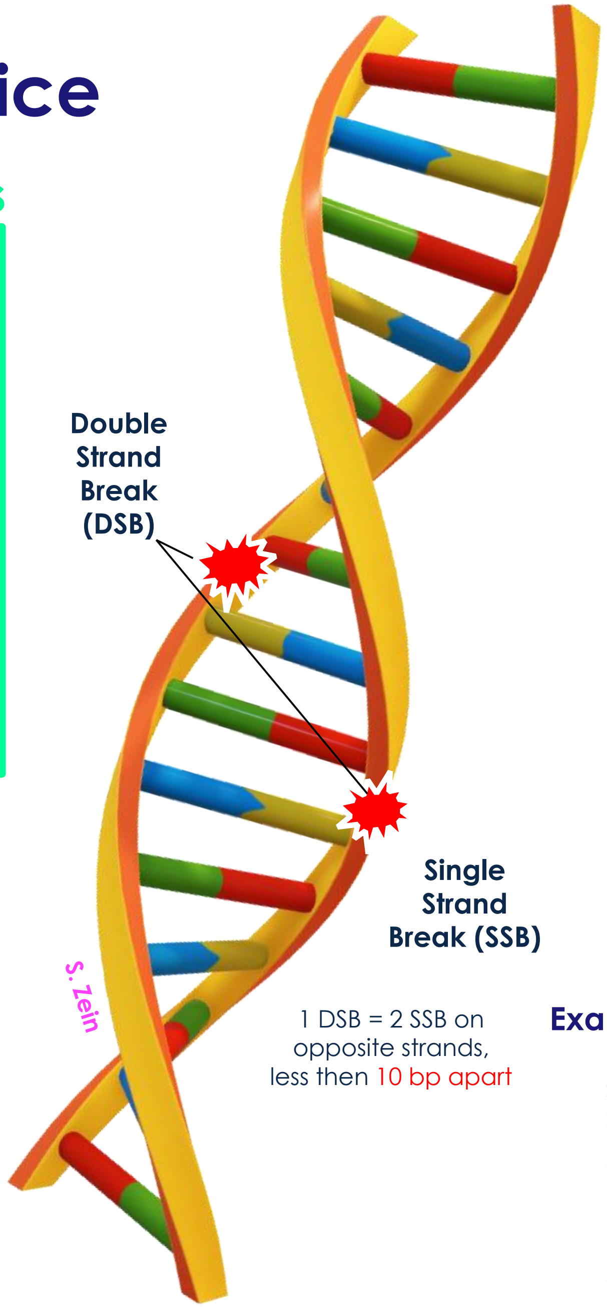
DIRECT damage induction

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. 6 A° for ~3 hydration sh.



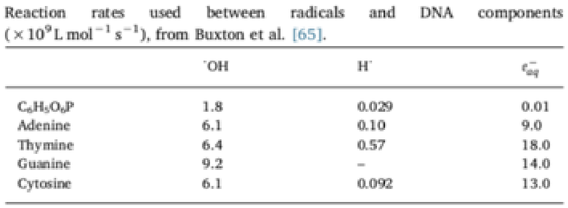
1 DSB = 2 SSB on opposite strands, less then 10 bp apart

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**

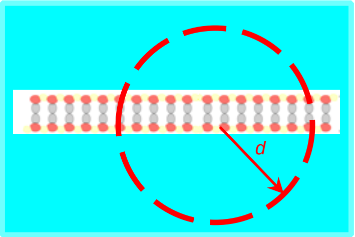
Example, for item 1. :



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

	˙OH	H˙	e^-_{aq}
$C_6H_5O_6P$	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. :



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

Physics

```
/process/dna/e-SolvationSubType Meesungnoen2002
#/process/dna/e-SolvationSubType Ritchie1994
#/process/dna/e-SolvationSubType Terrisol1990
```

```
/run/verbose 1
/control/verbose 1

/world/worldSize 50 um
/cell/radiusSize 14 2.5 14 um
```

Chemistry

```
/scheduler/endTime 5.0 ns
/scheduler/maxNullTimeSteps 10000000
```

endTime aims to simulate the radical scavenging that is considered as the radical time life. Beyond this time point radicals are scavenged by the biological medium

Geometry (1)

```
/dnageom/radicalKillDistance 9 nm
/dnageom/interactionDirectRange 3.5 angstrom
```

Geometry (2)

```
/dnageom/placementSize 75 75 75 nm
/dnageom/fractalScaling 75 75 75 nm
/dnageom/definitionFile geometries/cube-centred-X-8.txt
/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
```

Damage

```
/dnadamage/directDamageLower 5 eV
/dnadamage/directDamageUpper 37.5 eV

/dnadamage/indirectOHBaseChance 1.0
/dnadamage/indirectOHStrandChance 0.405
/dnadamage/inductionOHChance 0.00

/dnadamage/indirectHBaseChance 1.0
/dnadamage/indirectHStrandChance 0.0
/dnadamage/inductionHChance 0.00

/dnadamage/indirectEaqBaseChance 1.0
/dnadamage/indirectEaqStrandChance 0.0
/dnadamage/inductionEaqChance 0.00
```

Geometry (3)

```
/chromosome/add cell ellipse 7100 2500 7100 0 0 0 nm 0 0 0
#/chromosome/add cell sphere 3000 0 0 0 nm
```

```
/run/initialize
/run/printProgress 10

# 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 Energy
/gps/particle  e-
/gps/energy 0.662 MeV

#/analysisDNA/fileName 50MeV
```

Output file name

```
/run/beamOn 2
```

UI commands of moleculardna (1/4)

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)

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)

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)

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).

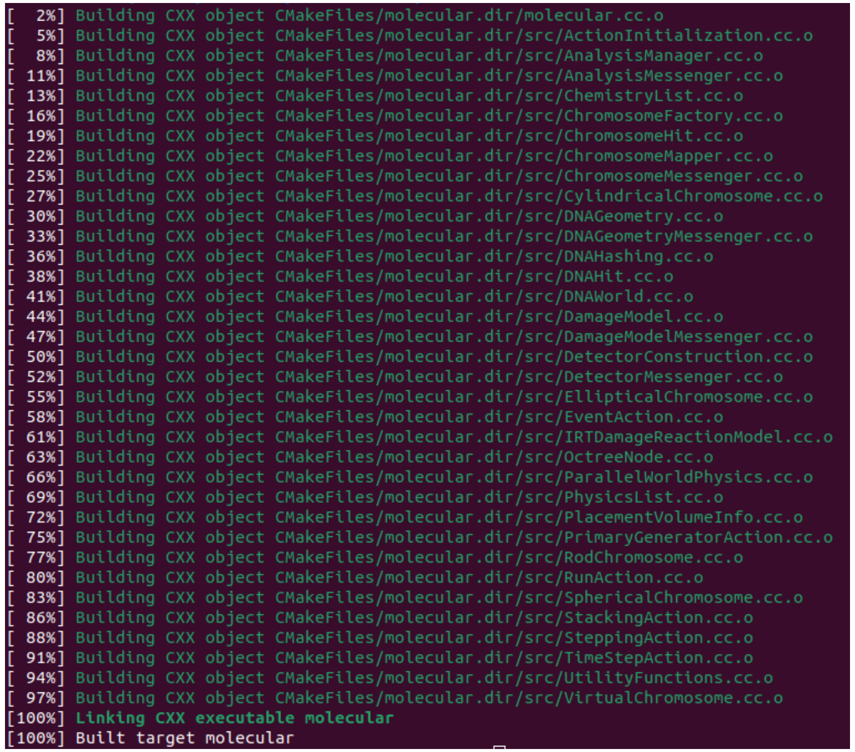
Hands-on

Hands-on practice with the « moleculardna » extended example

- **Copy** the moleculardna extended example to your local directory, **create** your build directory and **compile** moleculardna
  - `cd`
  - `cp -R $G4EXAMPLES/examples/extended/medical/dna/moleculardna .`
  - `cd moleculardna`
  - `mkdir build`
  - `cd build`
  - `cmake ..`
  - `make`

`make -jN` if you have N cores

The example needs internet to download geometry library and be installed



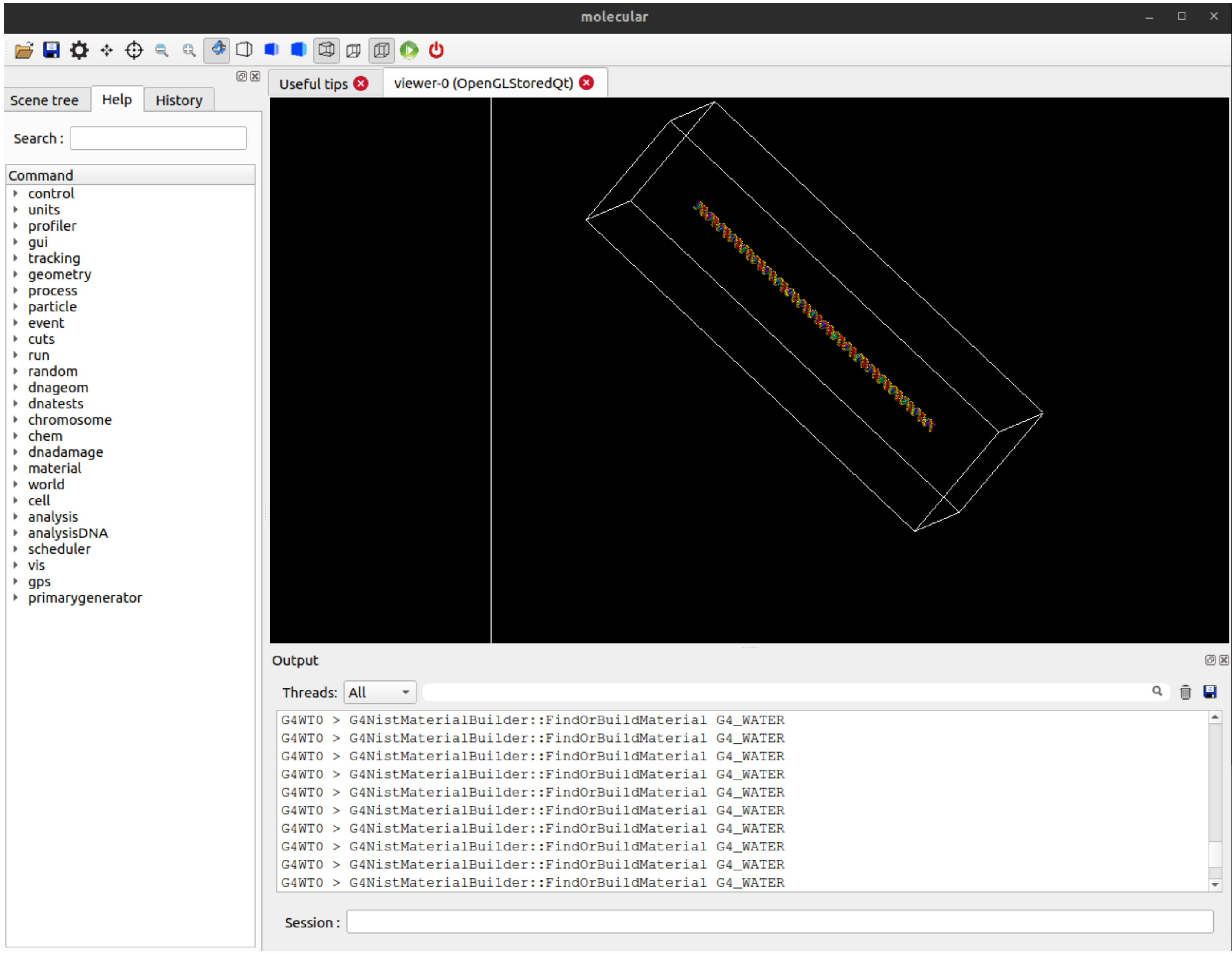
```
[  2%] Building CXX object CMakeFiles/molecular.dir/molecular.cc.o
[  5%] Building CXX object CMakeFiles/molecular.dir/src/ActionInitialization.cc.o
[  8%] Building CXX object CMakeFiles/molecular.dir/src/AnalysisManager.cc.o
[ 11%] Building CXX object CMakeFiles/molecular.dir/src/AnalysisMessenger.cc.o
[ 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 « moleculardna » extended example

- Run moleculardna in **interactive mode** with GUI commands
  - `./molecular`

- Using `/control/execute` command, you can run any example in interactive mode.

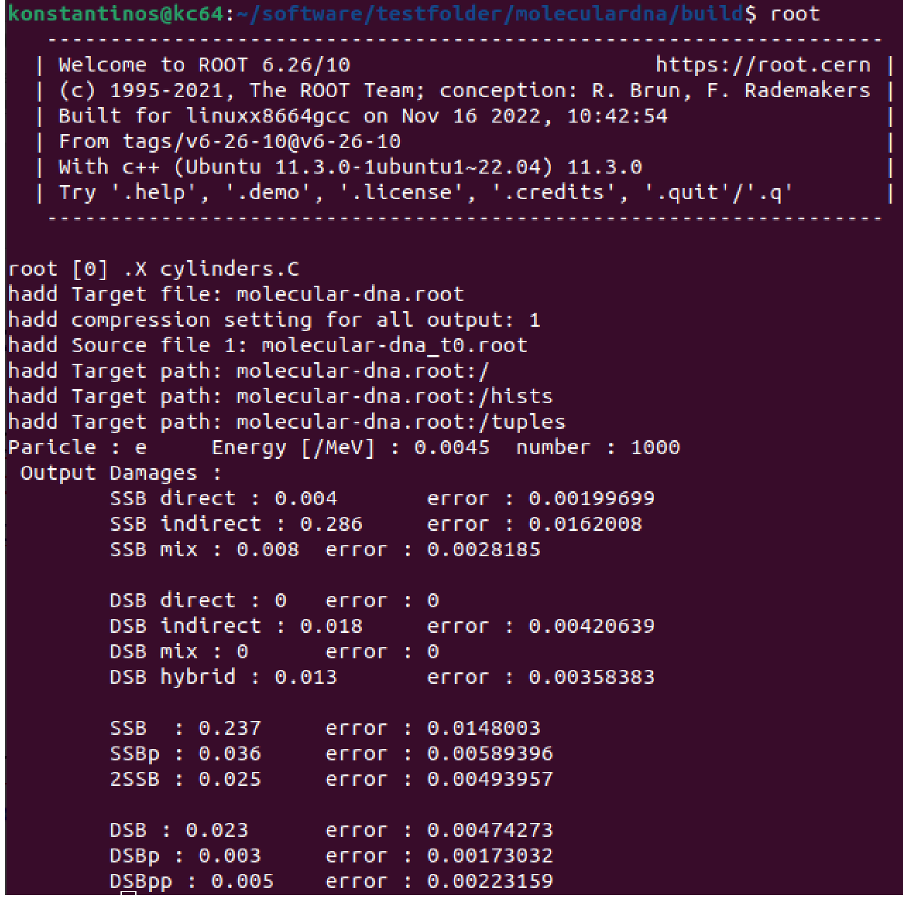
- Be careful on the RAM used.



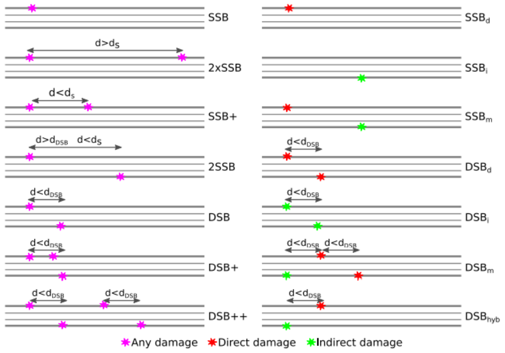
```
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
G4WT0 > G4NistMaterialBuilder::FindOrBuildMaterial G4_WATER
```

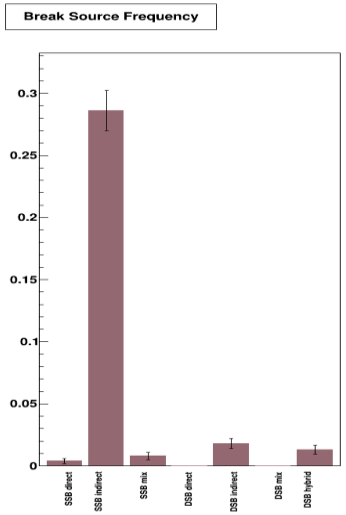
Hands-on practice with the « moleculardna » extended example

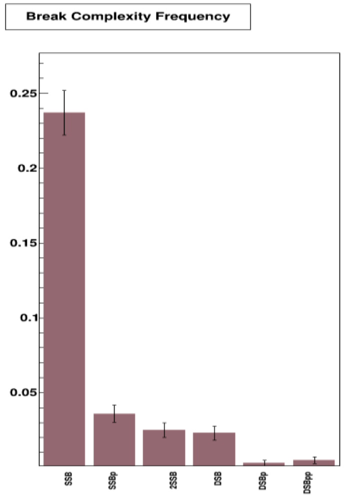
- Run moleculardnas in **batch mode** using a macro file
  - `./molecular -m cylinders.mac -p 4 -t 1`
  - You can run for 1000 electrons of 4.5 keV
  - No visualization by default
  - Results are saved in molecular-dna_t0.root file
- Results can be analyzed using ROOT (depending on the version these commands my differ)
  - `root`
  - ROOT is already installed on your Geant4 virtual machine
  - `.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
Pariicle : 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
```

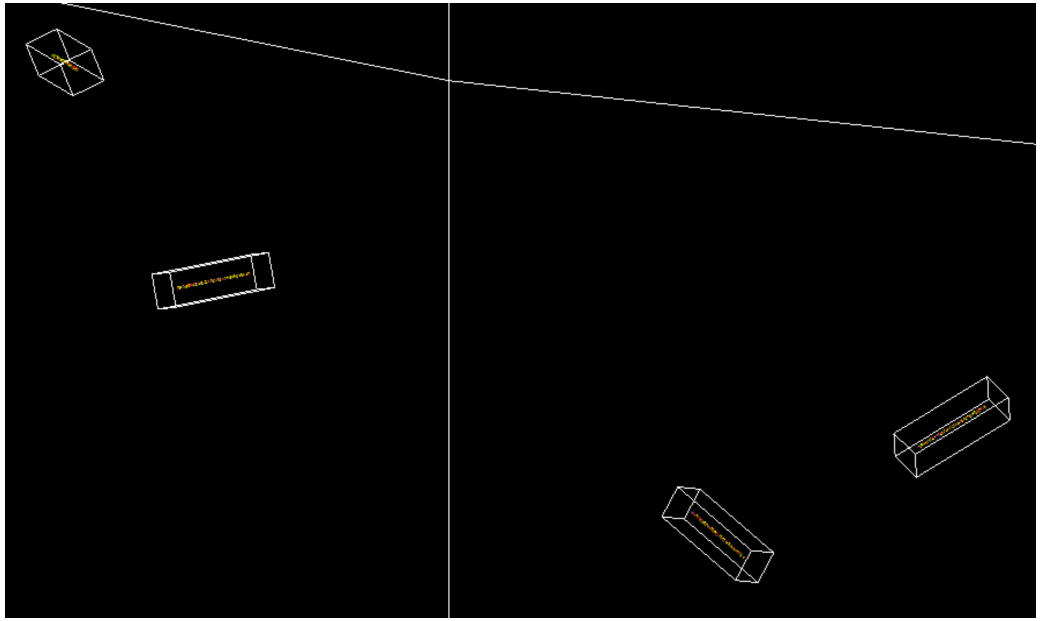


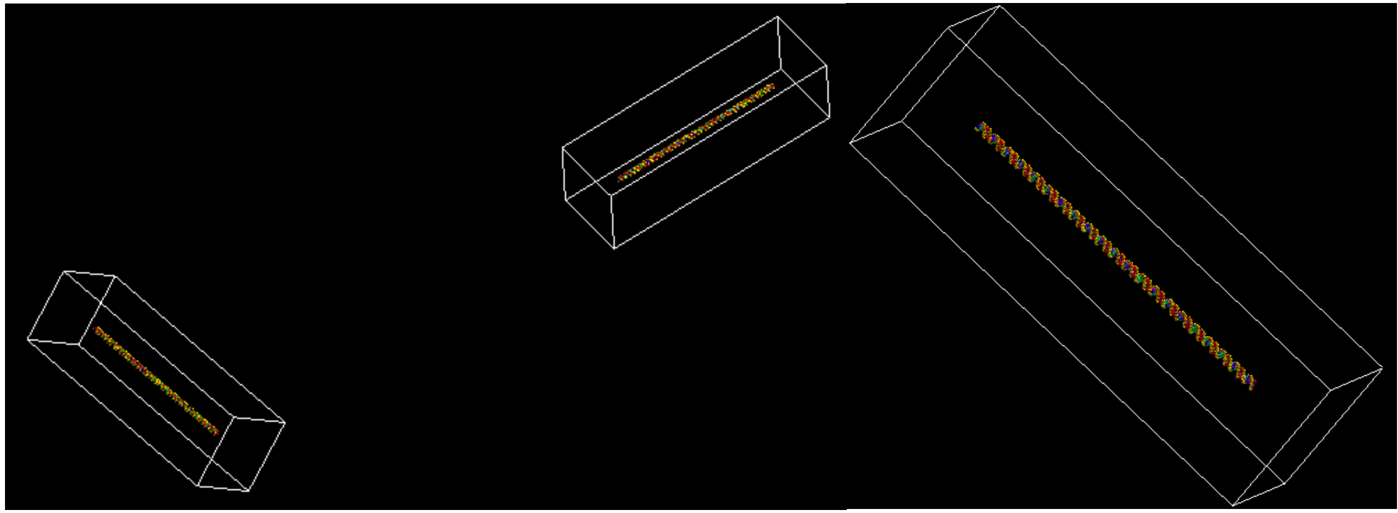




Hands-on practice with the « moleculardna » extended example

1. Open molecular.cc
   - Go to line 115 and comment it
2. Open DetectorConstruction.cc
   - Go to line 51 and enable « DNAWorld »
     - G4bool useParallelPhysicsWorld = true;
3. Open PhysicsList.cc
   - Go to line 78 and enable « DNAWorld »
     - G4bool useParallelPhysicsWorld = true;
4. Recompile moleculardna (Do `make`)
5. Create a copy of prisms200k_r3000.txt
6. Open prisms200k_r3000.txt:
   - Keep only the first 50 prisms (0-49)(up-to line 51).
7. Open cylinders.mac:
   - Comment the last line: `/run/beamOn`
   - Un-comment line /control/execute vis.mac
   - Replace prisms200k_r3000.txt, to the new one and save
8. In the terminal type `./molecular`
9. In the Qt window type: `/control/execute cylinders.mac`
10. Observe DNA geometry !





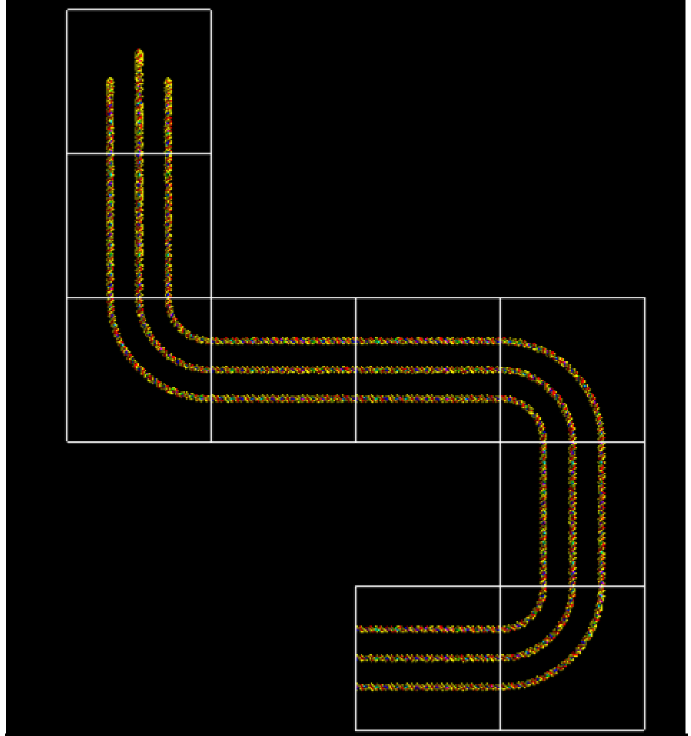
Hands-on practice with the « moleculardna » extended example

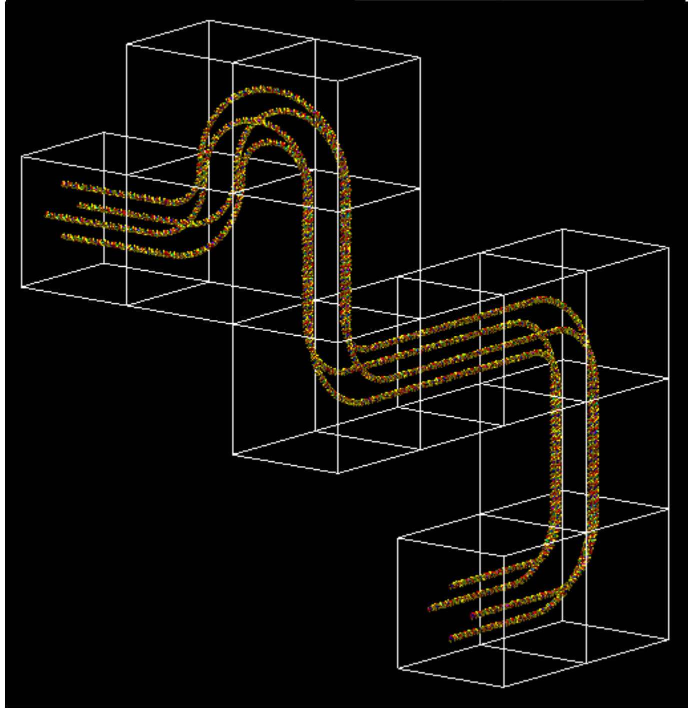
- Using 4strands_50nm_straight.txt and 4strands_50nm_turn.txt, create a new geometry like the one shown in the following images.
  Use also cylinders.mac as a starting point.

- 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
```

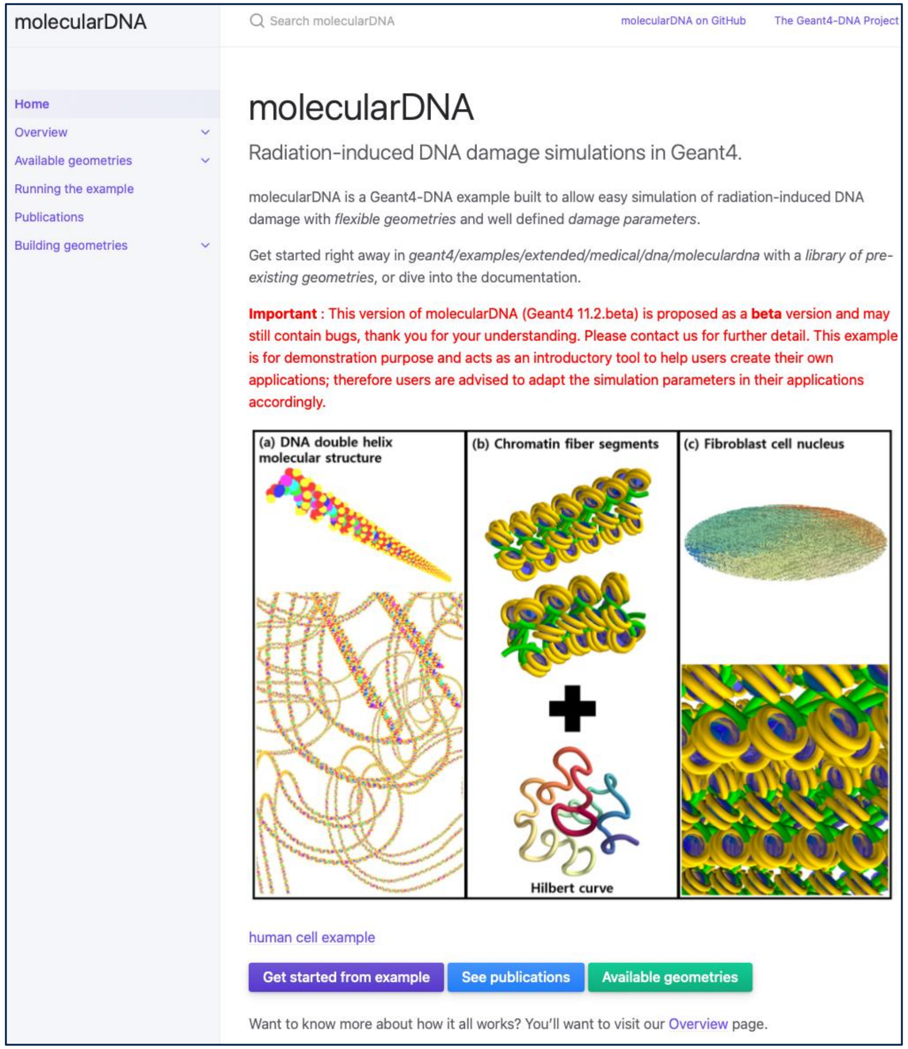




Documentation

moleculardna documentation
https://geant4-dna.github.io/molecular-docs/

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



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*.

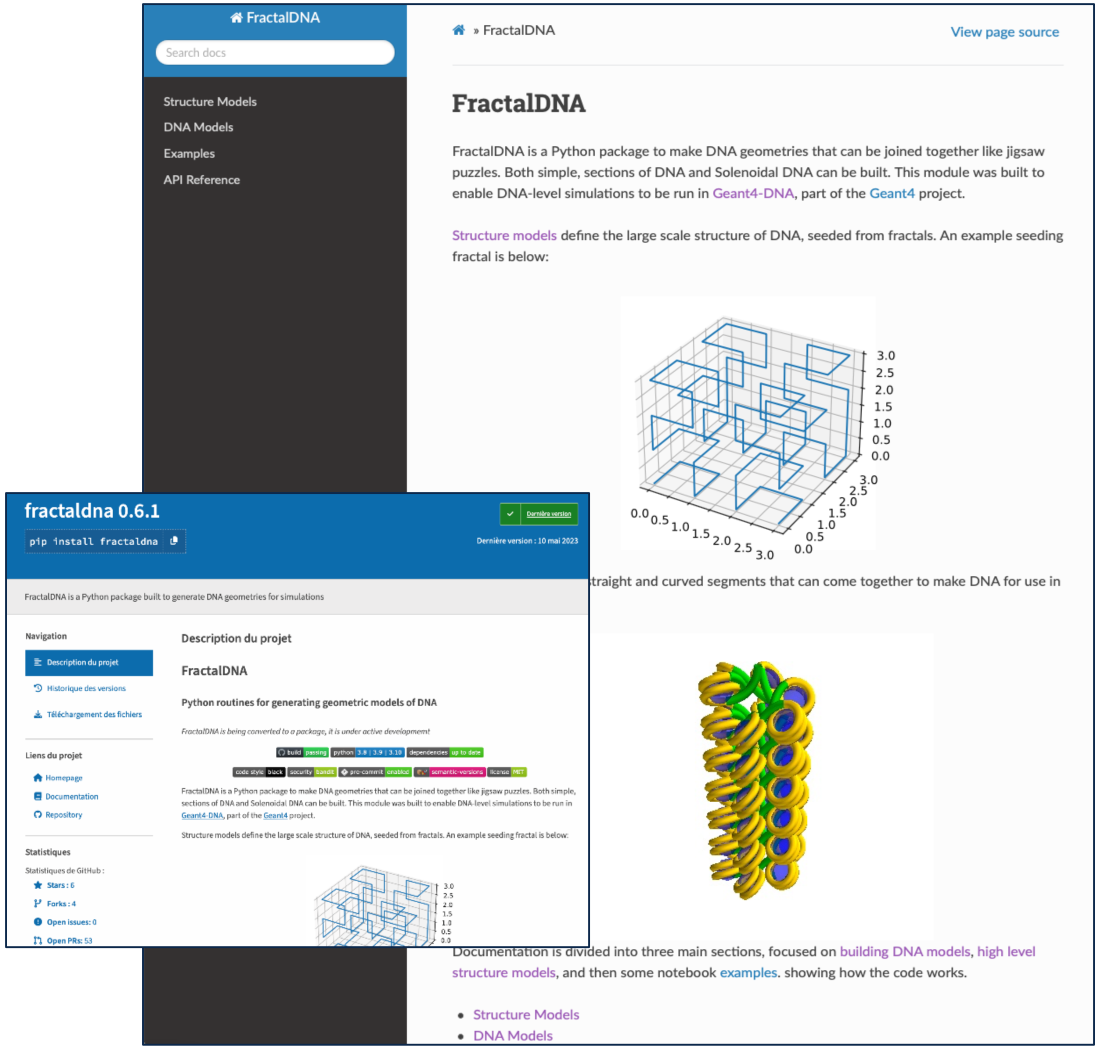
Get started right away in *geant4/examples/extended/medical/dna/moleculardna* with a *library of pre-existing geometries*, or dive into the documentation.

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

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



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:

traight and curved segments that can come together to make DNA for use in

Documentation is divided into three main sections, focused on building DNA models, high level structure models, and then some notebook examples. showing how the code works.

- Structure Models
- DNA Models

fractaldna 0.6.1

```
pip install fractaldna
```

Dernière version : 10 mai 2023

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

Statistiques de GitHub :
Stars : 6
Forks : 4
Open issues: 0
Open PRs: 53

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:

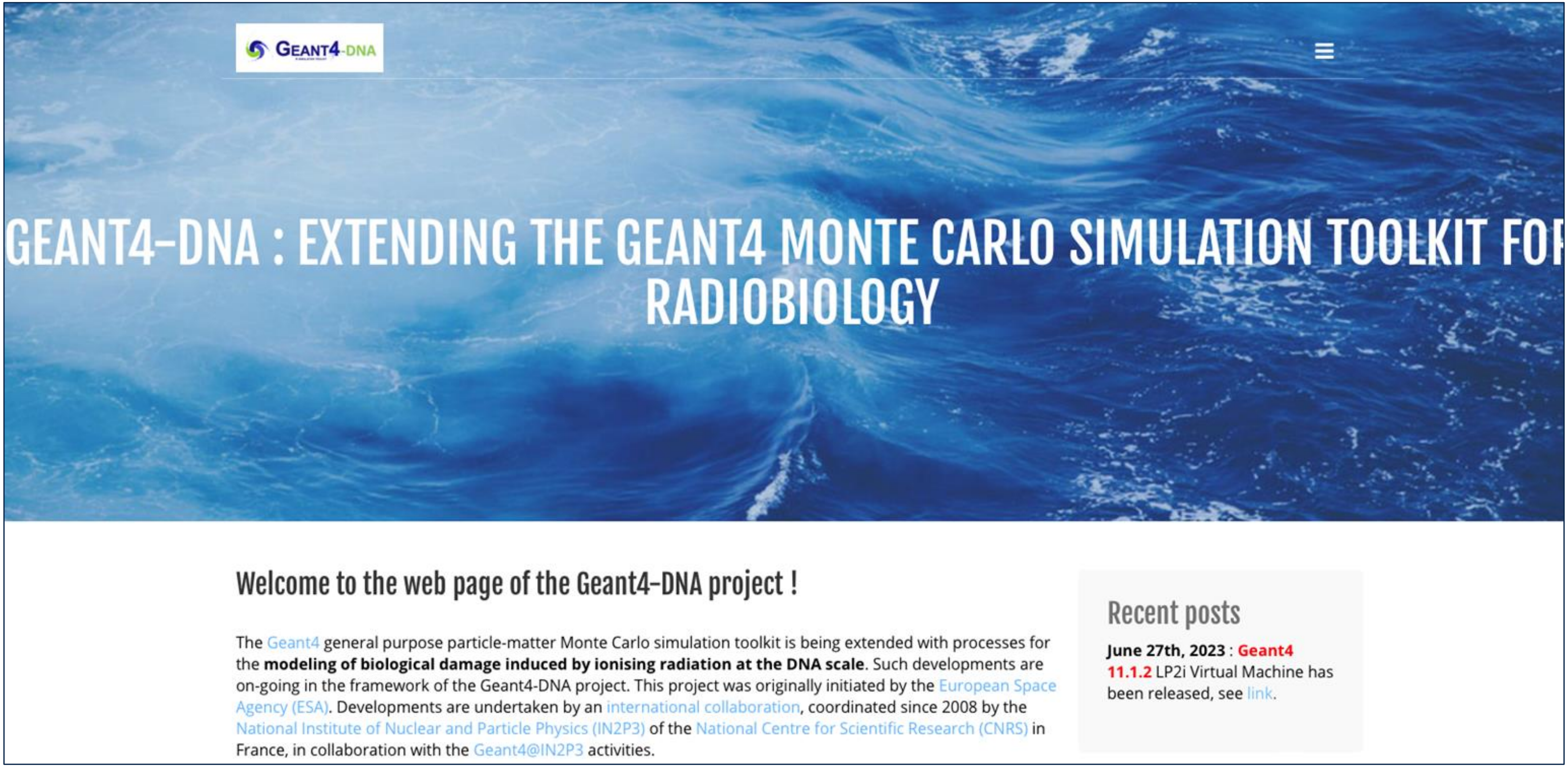
Main publications

- Geant4-DNA simulation of human cancer cells irradiation with helium ion beams, K. Chatzipapas et al., Phys. Med. 112 (2023) 102613 (link)
- 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)
- 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)
- 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)

http://geant4-dna.org
http://geant4.in2p3.fr

More information

http://geant4-dna.org



GEANT4-DNA : EXTENDING THE GEANT4 MONTE CARLO SIMULATION TOOLKIT FO
RADIOBIOLOGY

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.

Recent posts

June 27th, 2023 : **Geant4 11.1.2** LP2i Virtual Machine has been released, see link.