GAMMORA User Guide

GAMMORA was written by:
Jeremy LESTE
Tony YOUNES
Maxime CHAUVIN
Luc SIMON
Institut Universitaire du Cancer de Toulouse
Centre de Recherches en Cancerologie de Toulouse
Corresponding author: simon.luc@iuct-oncopole.fr

Abstract

This document describes how to install and how to use the GAMMORA package. GAMMORA is a tool that produces macro scripts for GATE v9.0 (or later) of the VARIAN TrueBeam STx linac. It includes the sources (6FF, 6FFF, 10FF and 10FFF), the geometry of the head, the ability to insert a CT, etc. GAMMORA is able to read DICOM RT PLAN to produce dynamic macros including the gantry and MLC motion.

Contents

1 Installation

1.1 Installation of vGATE

The best way to start GATE and GAMMORA for any kind of operating system is to use vGATE. vGATE is a virtual machine (VM) including Ubuntu, Geant4, GATE, code, etc. All is already compiled and installed properly (but sources are still inside!). vGate can be run on any OS and is similar to a computer in your computer. Of course, advanced users can download and compile GATE themselves.

- Install virtual box on your system (or any other VM tool): https://www.virtualbox.org/
- You also have to install from the Download page of the Virtual Box website the **VirtualBox 6.1.16 Oracle VM VirtualBox Extension Pack**

- Download vGate v9.0:
  http://www.opengatecollaboration.org/node/93
  The .ova file is quite large (8 Gb), place it in a safe directory. It can be deleted, once the VM is imported.
- Launch virtual box
- In the main menu: **File** → **Import a Virtual machine**
- Choose the downloaded .ova file and click **Open**, then **Continue**
- Click **Import** and wait several minutes
- When the import is over, select the VM in the left part, and click on **Settings**
- Several parameters can be changed but here are the two main ones.
- Under the **System** tab, select the quantity of RAM that you need for your VM. Select a value a least equal to 4000 Mo and no more than 80% of the maximum of your machine. Recommended value 10000 Mo.
- Under the **Shared folders** tab, add a folder that will be shared by your real machine and your VM. Enable the option: **Auto-Mount**. In the VM, the directory will be mounted to the path: /media/sf_shareVB/
- Close the **settings** windows and click on the green arrow to start your vGate machine (if needed, password is the word: *virtual*)
- Congratulations, you are probably in a Ubuntu environment. Right click on the desktop to open a terminal.

Maybe you will encounter the two following classic issues.

- For french user in particular, it may be possible that the keyboard is not properly selected. To select your keyboard: **Menu** (bottom left corner) → **control Center** → **keyboard**. Then select the keyboard under the **Layouts** tab
- For macOS users, the following operation must also be done. In the top menu of macOS (with the apple icon) Select **Devices** → **insert guest addition cd images** . Then, in a terminal type the following commands:
  - `cd /media/gate/VBox_GAs_5.2.18/`
  - `sudo sh VBoxLinuxAdditions.run`

    (password: virtual)
  - `sudo reboot`

    (the VM restarts...)
- In the terminal if you type

  `Gate`

  you should be in the Gate environment. Then:

  `exit`

  to quit.

1.2 Installation of GAMMORA

GAMMORA is a github repository. To clone it in your VM is very easy. To go in your home directory:

```
cd
```

Then clone the package with the following command:

```
git clone https://github.com/uhqd/GAMMORA.git
```

It creates a GAMMORA/ directory on your computer, in the directory where you ran this last command.

1.3 What's in the GAMMORA repository?

1.3.1 gaga-phsp/

This directory contains the *gaga* phase space (phsp) files of the TrueBeam for four photon energy modes: 6FF, 10FF, 6FFF and 10FFF. These phsp are made with *gaga*. It consists in a pre-trained Generative Adversarial Neural (GAN) network. The training has been performed by the GAMMORA team, using the IAEAphsp files, provided by Varian for its customer. These pre-trained GAN can generate any number of particles for your simulation.

Since Gate v9.0, it is possible to compile Gate with the libtorch library (already done in the VM). Thus, Gate is able to read any particle, generated with these gaga-phsp.

Each energy is composed of two files:

- .json: this contains the pytorch parameters
- .pt: the pre-trained GAN

Reference of *gaga* : Generative adversarial networks (GAN) for compact beam source modelling in Monte Carlo simulations Sarrut et al. PMB 2019 64(21)

gaga is available **here** on github.

1.3.2 my_first_GATE_macro/

GAMMORA creates GATE macros. This is a simple example of GATE macro that uses our gaga-phsp files. See the section 2.1 to try this example.

2 How to use it

2.1 Hello world!

Here we are going to run a first example of GATE macro, very similar to those created by GAMMORA. From GAMMORA directory, in your terminal, type:

```
cd my_first_GATE_macro/
Gate --qt mac/main.mac
```

The Gate QT visualization should appear and a simple Monte-Carlo simulation is running.

It consists in a gaga-phsp source of photon that turns and irradiate a phantom. There is not yet any elements of the linac head (jaws, MLC,...).

Close it when it is over and let's inspect the code using:

```
code mac/main.mac
```

In GATE, the commented line started with a #

Look at these commands and the commented lines to understand at least the structure of the macro.

The simulation is dynamic. It is made of several *run*. Each run can be seen as a new simulation.

The duration of each individual run are in the file:

`data/myTime.timeslices`

The number of primary particles for each run is in the file:

`data/primary.dat`

The position of the gantry at each run is in the file:

`data/gantryMovement.placements`

It is not possible to prepare these files manually for a real case, where gantry, collimator, MLC and even the patient can move during the simulation. This is the role of GAMMORA.