PET-SORTEO[*]
Simulation Package Manual

Anthonin Reilhac
McConnell Brain Imaging Centre, WB2B
Montreal Neurological Institute
3801 rue Université
Montréal, Qc, Canada
email: anthonin@bic.mni.mcgill.ca

March 11, 2003

[*] **PET-S**imulator **O**f **R**ealistic **T**ridimensional **E**mitting **O**bjects, SORTEO means random drawing in spanish

PET SORTEO is a Monte Carlo-based simulation tool able to generate realistic PET projections data (emission, transmission) given a scanner definition and an emitting and/or attenuation map(s) definition. The present document explains how to use this simulation tool. People are strongly encouraged to read published papers related to the simulator [,,,,].

Three majors steps are involved in the process of producing simulated projections using the simulator **PET-SORTEO**. First, the user must create a text file describing the simulation to be done with regard to the scanner geometry and physical properties, the acquisition configuration, the emission and attenuation map definitions etc ... To ease this step, a GUI is (will be) available (soon) that automatically generates this text file for standard acquisition from the interface entries. However, writing this file manually with any regular text editor is possible. This last mean gives you access to the complete set of keyword for the simulation configuration whereas the GUI restricts the setting. Then, the command `CompilProtocol` processes this text file and generates a binary protocol file. This binary file is not portable to other platforms and you must make sure to *compile* the protocol on the kind of platform that will be used for the computation. Finally, the command `sorteo` takes this protocol file as single argument and generates the projections. These three processes are represented in figure 1.

Besides, two others commands, related to the sorteo package happen to be usefull: `ShowProtocol` and `ResetProtocol`. Both act directly on the binary protocol file and allows for displaying (ShowProtocol) and reseting (ResetProtocol) the protocol data. The last command is usefull for generating multiple realizations of the same acquisition description.

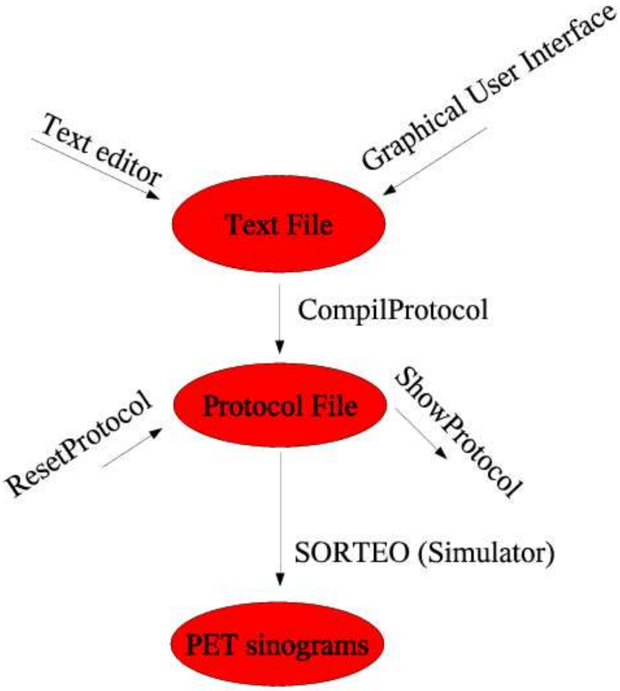

Figure 1: General functioning

1 Description of the main commands

1.1 CompilProtocol

```
 Usage CompilProtocol
      -t protocol text file (input)
      -p protocol binary file (output)
      -s show scanner list
      -r show radioelement list
      -m show medium list
      -v verbose mode
      -h help
```

The main purpose of this command is to generate a binary protocol file (name passed by -p option) from a protocol text file (passed by -t option) The protocol text file consists of a succession of paramerters set via predefined keywords. See section 2 for editing and/creating this text file. Options -{s, r, m} respectively returns the list of the available scanners, radioelements and materials (for attenuation definition). This may be usefull when one wants to edit manually the simulation text file.

1.2 ShowProtocol

```
Usage ShowProtocol -p protocol_filename [options]
      - 1st order options:
                   -E display informations concerning the emission volume.
                      possible 2nd order option: r.
                   -A display informations concerning the attenuation volume.
                      possible 2nd order option: r.
                   -F display informations about the involved files.
                      possible 2nd order options: e, t and b.
                   -S display informations about the scanner geometrical and physical parameters.
                   -G display informations about the acquisition parameters.
                      possible 2nd order options: e, t and b.
                   -C display informations about the event counts.
                      possible 2nd order options: e, t, b, f and r.
                   -B display probabilities derived from the simulation of the singles.
                      possible 2nd order options:  e, t, b and s.
                   -P display informations about the processes.
                   -I display the seed values used by the random generator.
 note: if no 1st order option is used, ShowProtocol displays the whole set of data.
      - 2nd order options:
                   -e restrict the displayed informations to the emission data.
                   -t restrict the displayed informations to the transmission data.
                   -b restrict the displayed informations to the blank data.
                   -f <frame_number> restrict the displayed informations to frame_number.
                   -r don't show informations about each emitting and/or attenuating regions.
                   -s don't show single rates per block.
      - miscl:
                   -l refresh cycle (sec)
                   -h help
This function displays informations from a binary protocol file.
The options allows you to restrict the display to specific components of a protocol file.
This function can be also usefull to monitor a running simulation (option -P, -C ..).
```

1.3 ResetProtocol

```
Usage: ResetProtocol  -p protocol_file [options]
                      [-p protocol file name]
                      [-s <unsigned long> new seed value for the random generator]
                      [-K keep the old seed value]
                      [-T get a new seed value from time]
                      [-S don't reinitialize values obtained from the singles computation]
                      [-h help]
This function resets a binary protocol file with the scope of regenarating data with the
same simulation description.
```

1.4 sorteo

```
Usage :sorteo protocol_file processor_number
```

This is the basic command to launch a simulation. For multiprocess simulations, the command must be launched manually on each processor/machine by specifying the process number. This tedious step may be automatized via a script.

2 Creating the protocol text file

2.1 Manually ...

The simulation text file si composed of 3 majors parts, the header, the core and the footer. The header and footer parts are not interpreted by the CompilProtocol command and the user can use them to write comments. The complete simulation description is given by the core part which is composed of a list of keyword associated to values. Comments can however be inserted within the core part by starting the line by the pound character #. The core part starts (and the header ends ..) when the first occurence of the keyword `protocol_begin` is encoutered, and stretches till the first occurence of the `protocol_end` keyword. The following example shows the basic structure of a simulation text file.

```
this is the header part
You can put whatever you want here.

protocol_begin
#the core of the simulation file begins at this point.
#comments are still possible with the # character
#this can be very usefull when you want hide a keyword, exemple:
#keyword value
#please, no space at the begining of a line neither blank line.
#the following will be interpreted with the CompilProtocol command:
keyword1 value1
keyword2 value2
...
protocol_end
The footer starts here.
Once again, you can put any kind of comments here.
```

For the simulation of an emission scan, the minimum pieces of information required are:

- the filename of the 3D emission map
- the list of emitting region labels with their associated tac definition
- the frame duration (a minimum of one frame is registered)
- the output filename (sinogram or list mode file).

In this case, default values are assumed:

- The scanner is the first of the scanner list (The Ecat Exact HR+ with the current distribution). Use CompilProtocol -s to have knowledge of the available scanners. The keyword scanner may be used to select another scanner from the scanner database. Experimented user may partially modify or completly define a scanner by using the keyword set_scanner associated to the list of field values (see below).
- The scanner operates with the 3d acquisition default parameters. This default bahavior may be overriden by passing the new acquisition parameters via the keyword configuration.
- Default sensitivity parameters provided by the scanner database are considered. These values may be switched by passing the new senstivity values via the keyword sensitivity
- There is no attenuation map (μ map). A volume giving the 3d attenuating media distribution may be passed by the keyword volume associated to the flag attenuation. Related attenuation regions must be set with the keyword aregion.
- The computation will be done over a single processor. The user can set the number of processor over which the computation will be done via the keyword process_number

The following core example may be used as a template for most of the simulations of emission data.

```
protocol_begin
# remove the # for setting another scanner
# by default the scanner is the 1st friom the scanner list
#scanner Ecat_Exact_HR+
#
#
# Use the next line for customized configuration
#configuration emission manual 9,22,0,1,1,328,650,144,1
#
#
# Use the next line for customized sensitivity parameters
#sensitivity emission 0.94,1,0.96,0.89,2000
#
#
# This is a 3 10-mn-frame acqisition
# each frame is separated by 30s
frame 600 30
frame 600 30
frame 600 30
#
#
# setting the emission volume
# 2 emitting regions are defined:
volume emission em_brain.v
eregion 1 white_matter 4 C-11 Bq/cc
tac 0 0
tac 20 4000
tac 100 5000
tac 1890 3000
eregion 2 gray_matter 2 F-18 nCi/ml
tac 0 0
tac 1890 90
#
#
# Setting the aatenuation map
# 4 attenuating regions are defined:
volume attenuation em_brain.v
aregion 1 R1 Water
aregion 2 R2 Water
aregion 3 R3 Lead
aregion 4 R4 Air
#
#
# sinogram of the emission data
sinogram emission sinogram.S
#
#
# Number of process, by default one.
# the following line to set a different number
# process_number 20
protocol_end
```

scanner

The scanner may be set with the scanner keyword.

scanner configuration

Then, the configuration parameters for emission mode are set with the line:

```
configuration emission manual 9,22,0,1,1,328,650,144,1
```

This line set the following parameters:

- span to 9
- maximum ring difference to 22
- septa state to 0 (retracted)
- transaxial interleaving to True (= radial sampling x2 and angular sampling / 2).
- Angular compression to 1 (Compression x2)
- Lower Level Discriminator to 328 KeV

- Upper Level Discriminator to 650 KeV
- number of radial crystals (in the FOV)

By default these parameters are set for 3d acquisition mode. Two dimensional parameters may be also used: configuration emission default2d.

scanner block sensitivity fit parameters

Scanner sensitivity with regard to the scattered and unscattered radiation may be adjusted using the following command line:

> sensitivity emission 0.94,1,0.96,0.89,2000

which fixes:

- the global sensitivity to 0.94
- the sensitivity with regard to the scattered gamma to 1.0
- the sensitivity with regard to the unscattered gamma to 0.96
- the threshold efficiency to 0.89
- the detected singles from background activity (imperfectly shielded rod source).

frame definition

`frame duration offset` is being used for the frame definition. The offset is optional. The simulator may be set for dynamic acquisition by consecutively defining each frame.

emission map

Here the emitting filename is passed by the keyword volume associated to the flag emission. Volume must be in Ecat7 format (Ecat native format) with the .v extension. Emitting labels are passed and defined via the keyword eregion associated to the label value, the region name the number of time-activity points (for the tac definition), the radioelement and the activcity unit related to the tac definition. The last two arguments are however optional and F-18 and Bq/cc are assumed as defaults. Each *eregion* command line must be followed by a list of time-activity points values passed by the keyword tac. This association *eregion/tac* must form a single block with no comment or other commands inserted in between. The number of time-activity points must be strictly equal to the one passed on the eregion command line. Each list of time-activity values has the following form:

```
tac t1 A1   #1st point
tac t2 A2   #2nd
...
tac tn An #last point
```

There is a little twist here: t_1 must alway be equal to 0 and t_n to the total scan duration. However, the command `CompilProtocol` will complain if these values are not set properly. Figure 2 shows the relation bewteen the numerical volume and the emitting region definitions. In this example, the numerical model is made of 4 differents labels (keyed from 1 to 4). Only region 1 and 2 are defined as emitting region. The others (region 3 and 4) are then automatically set to non-emitting. Figure 3 depicts the relation between the frame definition (offset and duration) and each region tac definition.

Attenuation map

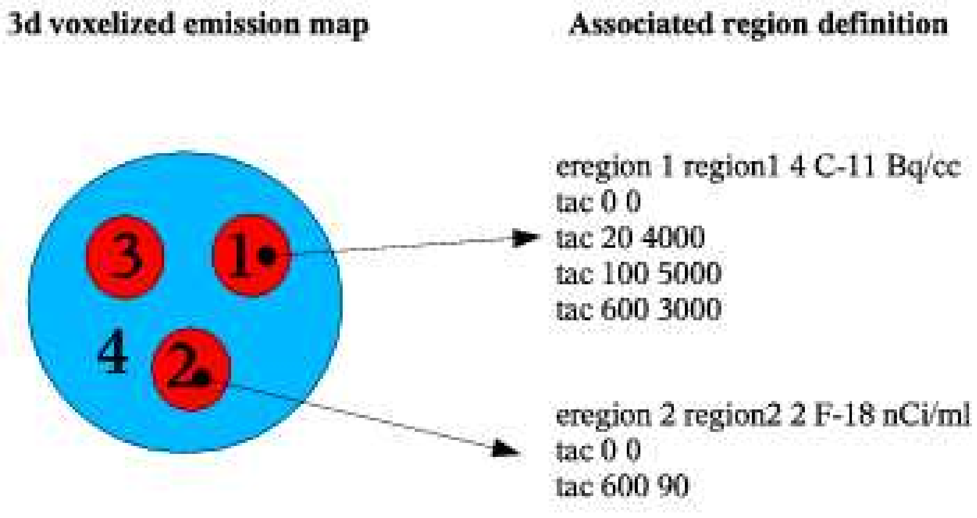
3d voxelized emission map
Associated region definition
3
1
4
2
eregion 1 region1 4 C-11 Bq/cc
tac 0 0
tac 20 4000
tac 100 5000
tac 600 3000
eregion 2 region2 2 F-18 nCi/ml
tac 0 0
tac 600 90

Figure 2: Emitting map definition

The procedure for defining the μ map is very close to the emissin map definition. The Ecat7 file is passed via the volume keyword with the attenuation flag. Then, attenuating regions setup with a a line of the form:

```
aregion label name medium (see the medium list)
```

If a region from the 3D volume does not find a corresponding definition within the *aregion* list, then it is considered as non attenuating.
The command line `CompilProtocol -m` lists all the available medium.

sinogram output file

Output sinograms are saved in the Ecat7 format (.S extension). The sinogram filenames are passed with the `sinogram` keyword with one of the associated flag: total (default), scatter, random or delayed. For a specific mode, assigning a name for the total events (unscatter + scatter + randoms - delayed) is mandatory; however, assigning a name for the scatter, random and delayed is optional. Alternative mode value for the sinogram are **transmission** and **blank**. In order to simulate a specific mode, the user has only to give a name to the correponmding sinogram (total).

splitting the computation over several processes

By default the binary protocol file is built for a computation over a single processor. This behavior can be overriden by setting the number of processor via the `process_number`.

The Complete keyword list

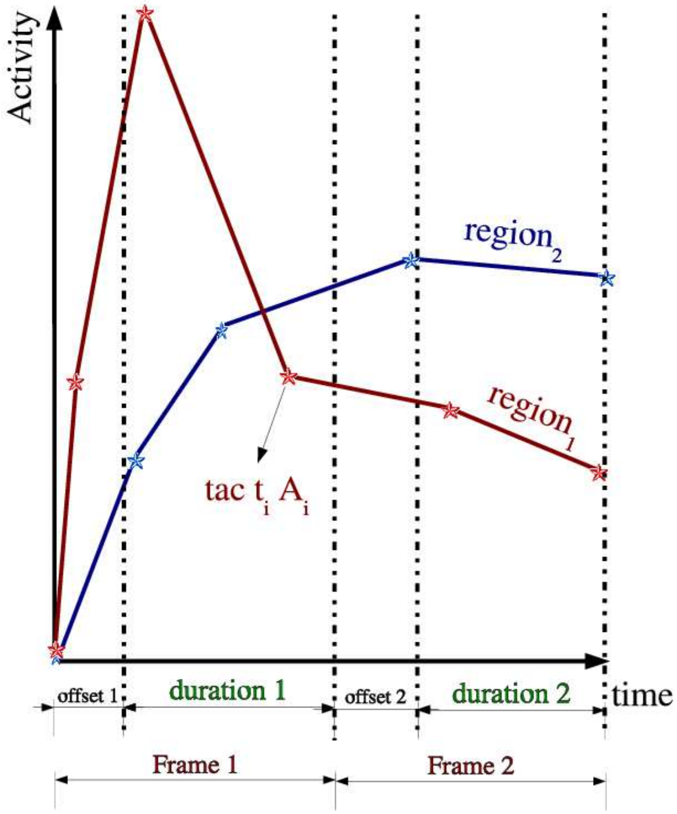

Figure 3: Emitting map definition

This simple example should be sufficient for most of the simulation description file. However, the `CompilProtocol` command allows for evoluated acquisition description involving for instance scanner geometry definition. The interested user is invinted to find the complete list of keyword below, associated to a short explanation.

aregion label name medium

> **label** (integer) label's value
>
> **name** (string) name of the region
>
> **medium** (string) one element from the medium list

configuration mode [options]

> **mode** **<u>emission</u>** or **<u>transmission</u>**
>
> **options** for:
>
> - **<u>emission</u>** (default2d,<u>default3d</u>,manual ...). If a scanner has been pick up from the databse, then the default acquisition is 3D. If scanner geometry is modified, the default values might be no longer viable and the user may have to define them manually. The `CompilProtocol` command will complain anyway!

- **transmission** (defaulttx, manual ...)

manual for

- **emission** span,mrd,septa,transinterleaving,mash,LLD,ULD,transFOV ,Delayedsubstraction
- **transmission** span,mrd,septa, transinterleaving,mash,LLD,ULD,transFOV ,Delayedsubstraction,window, rod_source_activity, duration, radioname

span (integer) .....

mrd (integer) ...

septa (0 — 1) 0 ⟺ (retracted — extended).

transinterleaving (0 — 1) ⟺ (False — True)

mash (integer) angular compression value (i.e. 1 $\mapsto$ x2, 2 $\mapsto$ x4 ...)

LLD (integer) Lower Level Discriminator (KeV).

ULD (integer) Upper Level Discriminator (KeV).

transFOV (integer) Number of radial crystal per projection.

Delayedsubstraction (0 — 1) ⟺ (False — True).

window (integer) Number of radial bin of the mask during transmission (rod winfowing).

rod_source_activity (double) total rod source activity (Bq)

duration (double) transmission/blank scan duration (sec)

radioname (string) an element from the radioelement list.

deadtime mode boolean

mode **emission** or **transmission** or **blank**

boolean true or false, default is true

eregion label_value region_name num_tac_points radioelement (set by default to the 1st element of the list) unit (defaut is 1st element).

tac time, activity

label_value (integer) labels's value

region_name (string) name of the region

num_tac_points (integer) number of (time,activity) pairs.

radioelement optional, (string) radioelement name (see the list), default is F-18

unit optional, **bq-cc** or **nci-ml** default is bq-cc

time (double) the time in second at this point

activity (double) the activity at this point.

flip flag flipx,flipy,flipz (default is no flip)

flag (emission — attenuation).

flipx,y,z (0 — 1) ⟺ (False — True)

frame duration offset

duration (double) the frame duration

offset optional, (double) the offset between the end of the former frame and the begining of this one.

initseed value (unsigned int), by default, initseed is set with the `time() functions`.

lmf mode filename, where mode = **emission** or **transmission** or **blank**. Write the output in list mode format.

one_sino_per_layer 4 sinograms are outputed (only with dual crystal acquisition)

offset flag offsetx,offsety,offsetz (default is 0.,0.,0.).

- **flag** (emission — attenuation)
- **offsetx,y,z** (double) the offset from the volume to the scanner (center to center).

process_number number (integer), default is 1.

process_weight proc_i weight

- **proc**$_i$ processor number
- **weight** relative weight of the process (by default, all the processes have the same weight).

refresh_cycle mode time

- **mode** **emission** or **transmission** or **blank**
- **time** (double) sampling step in sec for the singles related phenomena in sec. (default is 1.)

scanner scanner name. By default the first scanner on scanner list is selected (the Ecat Exact HR$^+$ with the current distribution). Use `CompilProtocol -s` to get the complete list of scanners.

sensitivity mode ϵ_G,ϵ_S,ϵ_U,ϵ_d, [singlebg - just for emission]

- **mode** **emission** or **transmission**
- ϵ_G (double between 0. and 1.) Global sensibility (default defined in the scanner data base)
- ϵ_S (double between 0. and 1.) Block sensibility for scattered singles (default defind in the scanner data base)
- ϵ_U (double between 0. and 1.) Block sensibility for unscattered singles (default defined in the scanner data base)
- ϵ_d (double between 0. and 1.) Thresholding efficiency, (default defined in the scanner data base)
- **singlebg** (double) detected block single rates from the rod source (default define in the scanner data base)

set_scanner field value where filed:

- **modele** (integer) modele number
- **lower_crystal_block_dim_h** (double) the heigth in cm of the lower block of crystal in the detection module
- **upper_crystal_block_dim_h** (double) the heigth in cm of the upper block of crytal in the detection module
- **block_dim_trans** (double) transverse dimension in cm of the block module (shield included)
- **block_dim_z** (double) axial dimension in cm of the block module (shield included)
- **shield_dim_trans** (double) width (in cm) of the sheet of metal in the transverse direction (shield)
- **shield_dim_z** (double) width (in cm) of the sheet of metal in the axial direction (shield)
- **num_crystal_unit_trans** (integer) number of crystal unit per block in the transverse direction
- **num_crystal_unit_z** (integer) number of crystal uinit per block in the axial direction

saw_cut_dim_trans (double) dimension in cm of the gap between the crystal units in the transverse dimension

saw_cut_dim_z (double) dimension in cm of the gap between the crystal units in the axial dimension

saw_cut_h (double) air gap depth in cm

block_shield_material (see the material list) material of the block shield

lower_crystal_name (see the material list) material of the lower crystal

upper_crystal_name (see the material list) material of the upper crystal

block_gap_material (see the material list) material filling the gap between crystal units

lower_crystal_energetic_resolution (double) energetic resolution of the lower crystal

upper_crystal_energetic_resolution (double) energetic resolution of the upper crystal

lower_crystal_spatial_resolution (double,double) block spatial resolution in cm in the transverse and axial direction for the lower crystal

upper_crystal_spatial_resolution (double, double) block spatial resolution in cm in the transverse and axial direction for the upper crystal

thau_block (double) integration cycle in second at the block level

num_block_in_plane_trans (integer) number of blocks per plane in the transerve direction

num_block_in_plane_z (integer) number of blocks per plane in the axial direction

distance_between_blocks_trans (double) air gap dimension in cm between 2 blocks in the transverse direction

distance_between_blocks_z (double) air gap dimension in cm between 2 blocks in axial direction

air_gap_between_plane (double) air gap dimension in cm between 2 planes of blocks

num_side (integer) number of planes

num_bucket_per_ring (integer) number of buckets within a ring

num_bucket_ring (integer) number of bucket rings

thau_bucket (double) bucket integration period in second.

in_septa_radius (double) septa inner radius in cm.

out_septa_radius (double) septa outer radius in cm.

septa_material (see the material list) septa material

septa_thickness (double) thickness of the septa in cm.

shield_radius (double) radius in cm of the scanner ends shields

rod_source_radius (double) orbit in cm of the rotating rod sources

rod_source_length (double) length in cm of the rotating rod sources

rod_source_omega_rotation (double) rotation per minute of the rod sources

thresh_trig (double) triggering energetic level for gamma integration, in keV

thau_coincidence (double) coincidence window in sec.

thau_processor (double) processing clock cycle of a single processor in sec

num_processor (integer) number of CPU to process the events

storage_bandwidth (double) storage bandwidth (in Hz)

num_theta_sampling_tx_per_block (integer) number of virtual block subdivisions for the singles rates computation in transmission mode.

eff_detection_radius (double) scanner effective radius in cm including gamma penetration

depth_of_interaction_crystal1 (double) gamma penetration in crystal1 in cm.

depth_of_interaction_crystal2 (double) gamma penetration crystal2 in cm.

sinogram mode filename [options]

mode **emission** or **transmission** or **blank**

options total, scatter, random, delayed

Attention: for any mode, `total` tells the simulator to simulate the corresponding mode, `scatter, random, delayed` are optional and useless if the total is not set.

volume flag filename, where flag is emission or attenuation

2.2 Using the Graphical Interface