Release Notes - Dose Calculation

Author	Ryan Neph
Version	0.7.0
Date	2018 March 15

Features:

- Support for collimator rotation angle selection in beamlist file
- **Significant** speed improvements enabled by further optimizing memory management during beamlet dose unpacking, sparsification, and write-to-file operations.
- New sparse beamlet dose *write-strategy* introduced in `dosecalc-beamlet` that enables storing per-beamlet dose in independent files, linked to a single entry-point file.
- Support for reading the new beamlet dose file format using matlab function `read_dose_data.m` .
- Added option to only produce metadata structs in matlab beamlet dose file parsing `read_dose_data.m`
- Early support for constructing `--fmaps=` compatible fluence map and beam angle files via `fmo_io.py` command line interface

Interaction Changes:

- Changes to the beamlet dose output (h5) file (old and new version should be compatible with latest data loading scripts in `<rs4pi>/dosecalc_gpu/extern/`) including the addition of the following metadata fields:
  - in *calc_meta*:
    - kernel_extent_cm
    - penumbra_cm
    - rev_voxelsize_cm
  - in *beamlet_meta*:
    - perc_nonzero (computed with respect to calc_bbox volume)
    - perc_dropped (nonzeros whose value was below *sparsity_threshold*)
    - perc_reducedrop (if M->A matrix is requested, describes nonzeros that don't appear in any defined structures/ROIs)
- Beamlist file now accepts an optional collimator rotation angle in degrees. Check `<rs4pi>/dosecalc_gpu/doc/schema_examples/example_beamlist.txt` for details. This version is compatbile with beamlist files (without collimator angle specification) that were created for v0.6.0+.
- New options are available as command line arguments to `dosecalc-beamlet` that can improve speeds on certain hardware configurations (default settings should be acceptible in most cases):
  - `--write-strategy=["central"|"perbeamlet"]` - change the way beamlet dose is written to disk (experimental: perbeamlet performs per-beamlet writing and links these external files to the main h5 file)
  - `--sparsify-streams=<int>` - enables concurrency in beamlet dose unpacking and device to host memory transfers. default value is 4. This is independent of the number of threads to use, and only marginally increases the CPU-side memory overhead.
  - `--srworkers=<int>` - sets the number of independent CPU threads to dedicate for asynchronous beamlet sparsification and M-matrix to A-matrix conversion (aka reduction; if enabled). This significantly affects

performance and should be at least equal to the number of GPUs in use (if there are CPU cores available).
    - `--w2fworkers=<int>` - sets the number of independent CPU threads to dedicate to asynchronous sparse beamlet dose write-to-file operations. This significantly affects performance and should be at least equal to the number of GPUs in use (if there are CPU cores available) (in "central" write_strategy mode, this is always set to 1).
    - 3rd party language support (matlab/python) scripts have been moved from /treatmentPlanning, to /dosecalc/extern/
- Changes to `dosecalc-preprocess` command line and config arguments:
    - `--revvoxsize=<float>` has been added for quick setting of rev (convolution) voxel size
    - `--delr=<float>` & `config:"delr": <float>` have been removed - now set automatically

Bug-fixes:

- fixed fluence map rotation when collimator rotation is assumed to be 0 degrees (won't have a big influence on the dose output)
- fixed data shape permutation in dosecalc-beam output causing *Dose.h5* data shape to be incorrectly set as XYZ instead of ZYX
- fixed issue where penumbra was interpreted as #voxels instead of distance in cm in `dosecalc-beam`
- fixed long read times in `open_masks.m` matlab external script by using array views instead of data re-ordering
- removed convolution ray early stopping condition that produced dose artifacts for *targets* with large "holes"
- fixed issue where sometimes the temp directory is not cleared during dosecalc-preprocess execution
- fixed dose kernel value interpolation shift (by 0.5 indices; won't have a big influence on the dose output)

Questions & Concerns:

Please contact the package maintainer: Ryan Neph <ryanneph@ucla.edu> with any error reports or questions.