Automated proton beam model validation

A. J. Egan[1], D. Maes[1], R. Regmi[1], C. Bloch[2], J. Saini[1], P. Taddei[2], S. Bowen[2], T. Wong[1]
[1]SCCA Proton Therapy Center, Seattle, Washington, USA
[2]University of Washington School of Medicine, Seattle, Washington, USA

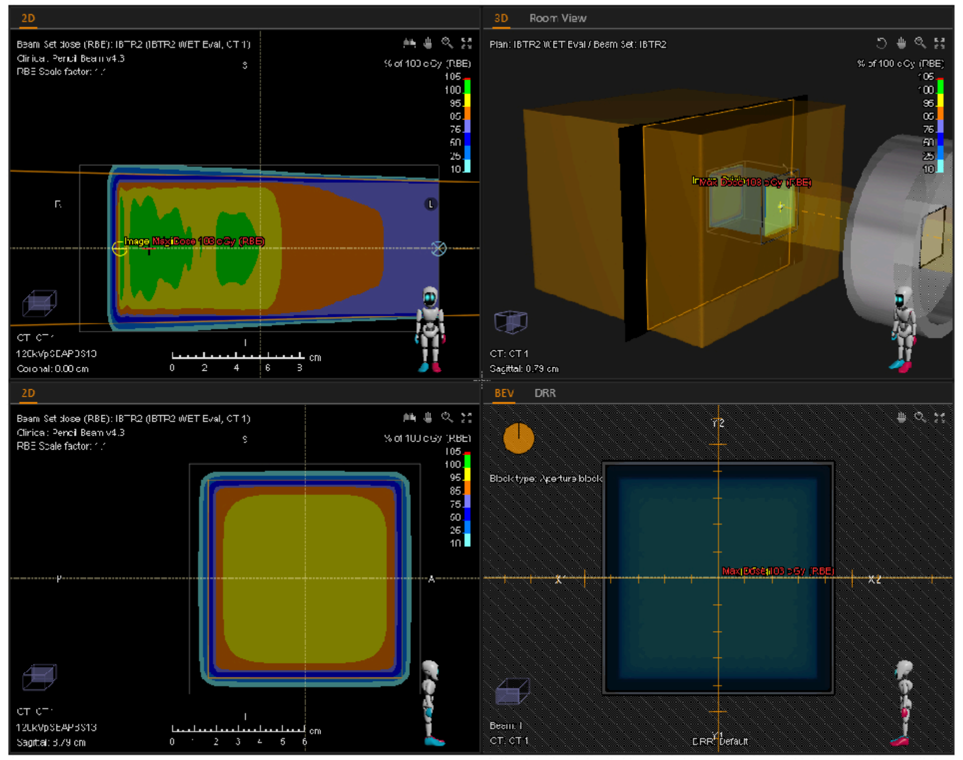
JULY 12–16 2020 VIRTUAL JOINT AAPM | COMP MEETING EASTERN TIME [GMT-4]

INTRODUCTION

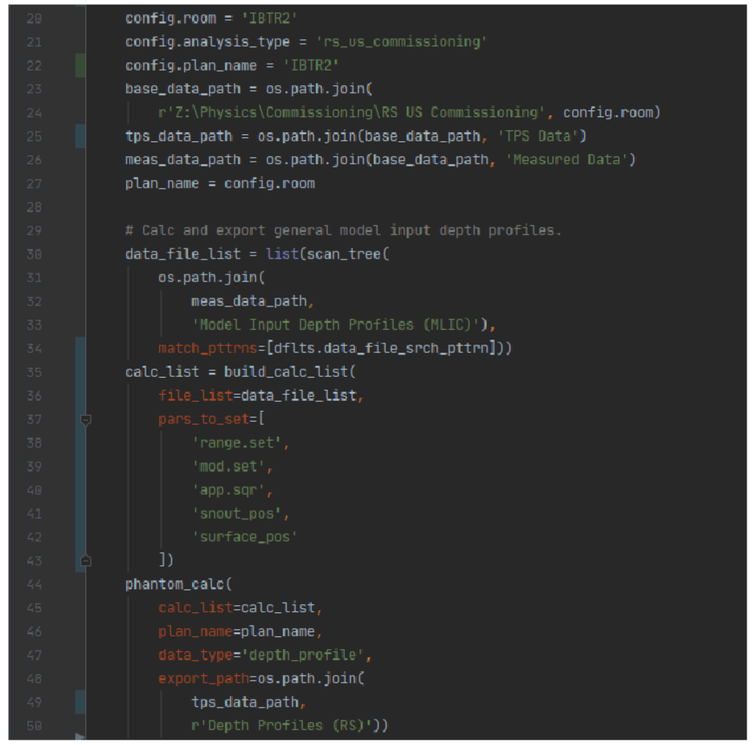
Manual beam model validation can be an error-prone and time-consuming process. To verify clinical viability, the model must first be tested across a wide range of beam and geometry parameters. For each individual calculation, dose-distribution data must be manually exported and then manually matched and compared to measured data, often in varied software. At the Seattle Proton Therapy Center (SPTC) an in-house suite of Python-based automation scripts have been developed to automate this process.

METHODS

These tools were initially developed to assist in the commissioning of IBA uniform scanning [Ion Beam Applications, Louvain-Neuve, Belgium] for the RayStation treatment planning system [RaySearch Laboratories, Stockholm, Sweden]. The RayStation scripting interface was used in conjunction with Python code to automate beam model parameter variation, dose calculation, and data export. Parameter set lists can either be scraped from previously measured data file names or specified manually. Comparison data searching/matching, device-specific data file reading (e.g. multi-layer ion-chamber or planar array), data-type-specific reformatting (e.g. noise smoothing or curve normalization), and finally analysis (e.g. gamma comparison) is all fully automated requiring just minutes of user-specific configuration. Validation results are documented in pdf printing of comparison plots and tables, and analysis metrics across multiple comparison instances are compiled in csv files for easy review in spreadsheet software.



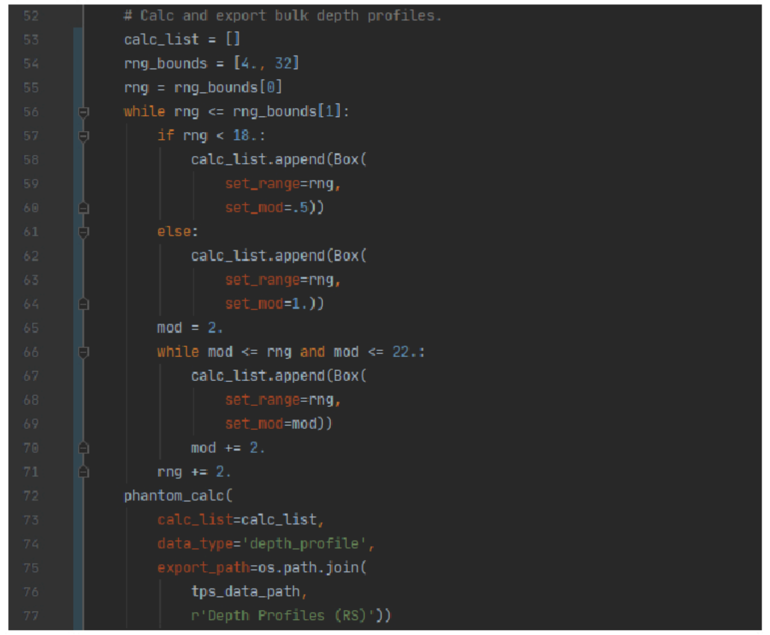
A simple geometry water phantom with a simple beam set and beam must be initially manually configured in RayStation. Square and circular apertures are automatically build to physical specifications as illustrated in the methods section.



```python
config.room = 'IBTR2'
config.analysis_type = 'rs_us_commissioning'
config.plan_name = 'IBTR2'
base_data_path = os.path.join(
    r'Z:\Physics\Commissioning\RS US Commissioning', config.room)
tps_data_path = os.path.join(base_data_path, 'TPS Data')
meas_data_path = os.path.join(base_data_path, 'Measured Data')
plan_name = config.room

# Calc and export general model input depth profiles.
data_file_list = list(scan_tree(
    os.path.join(
        meas_data_path,
        'Model Input Depth Profiles (MLIC)'),
    match_pttrns=[dflts.data_file_srch_pttrn]))
calc_list = build_calc_list(
    file_list=data_file_list,
    pars_to_set=[
        'range.set',
        'mod.set',
        'app.sqr',
        'snout_pos',
        'surface_pos'
    ])
phantom_calc(
    calc_list=calc_list,
    plan_name=plan_name,
    data_type='depth_profile',
    export_path=os.path.join(
        tps_data_path,
        r'Depth Profiles (RS)'))
```

In approximately 30 lines of configuration code the Python-based automation tool can be set to search a directory of appropriately named measured data and then then automatically specify, calc, and export those beam model scenarios.



```python
# Calc and export bulk depth profiles.
calc_list = []
rng_bounds = [4., 32]
rng = rng_bounds[0]
while rng <= rng_bounds[1]:
    if rng < 18.:
        calc_list.append(Box(
            set_range=rng,
            set_mod=.5))
    else:
        calc_list.append(Box(
            set_range=rng,
            set_mod=1.))
    mod = 2.
    while mod <= rng and mod <= 22.:
        calc_list.append(Box(
            set_range=rng,
            set_mod=mod))
        mod += 2.
    rng += 2.
phantom_calc(
    calc_list=calc_list,
    data_type='depth_profile',
    export_path=os.path.join(
        tps_data_path,
        r'Depth Profiles (RS)'))
```

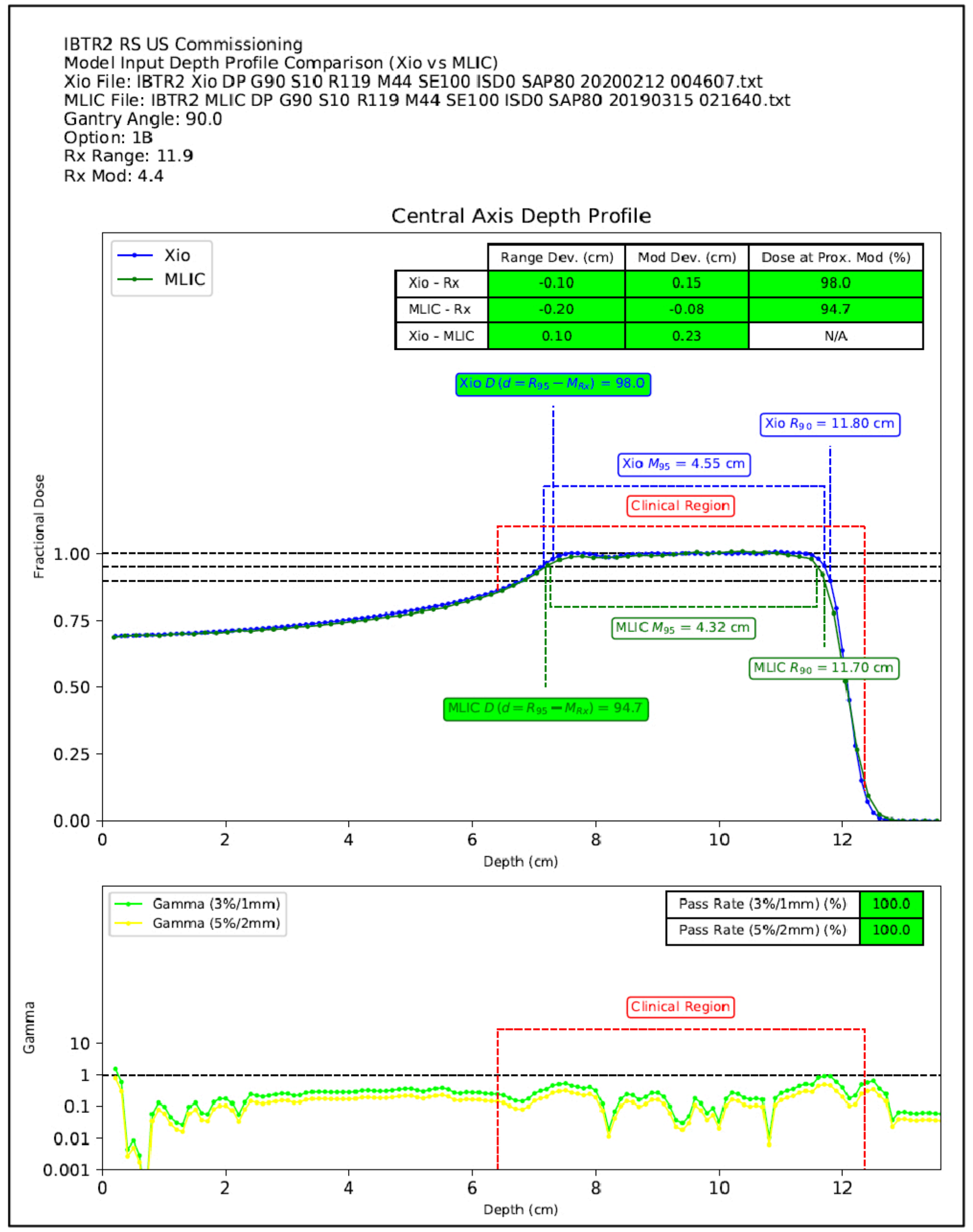
Similarly a calculation list of specific parameter sets can be specified using simple for loops.

CONTACT INFORMATION

Alexander J Egan, PhD, DABR
Email: alex.egan@seattleprotons.org
Mobile: 484-955-7849

RESULTS

For this specific model validation instance, plan calculation and export required 30 minutes of user configuration and approximately 8 hours of unattended calculation time. Analysis of this data set required 30 minutes of configuration and approximately 1 hour of unattended calculation time.



	Range Dev. (cm)	Mod Dev. (cm)	Dose at Prox. Mod (%)
Xio - Rx	-0.10	0.15	98.0
MLIC - Rx	-0.20	-0.08	94.7
Xio - MLIC	0.10	0.23	N/A

For each exported data curve, in-house developed analysis software automatically searches and matches measured and calculated data, constructs pdf output plots (example SOBP comparison shown above) as well as compiles batch-wise spreadsheet outputs of key analysis metrics.