MATLAB FUNCTIONS

model_msk.m

INFO: Implementation of the MSK model. This function simulates the dynamic response of a tumor to a radiotherapy schedule by tracking cell populations across the three specific oxygenation compartments defined in the MSK model.

INPUTS:
- *schedule*: treatment schedule doses, 2xn matrix, where n is the number of fraction. The first row constains the doses per fraction, and the second row constaing the treatment delivery time (h) of each fraction.
- *params*: parameters of the MSK model

OUTPUTS:
- *t_array*: an array of specific time points recorded during the simulation
- *N_v_hist*: an array with total number of viable surviving cells at each time point in *t_array*
- *D_hist*: an array recording the cumulative radiation dose delivered at each time point in *t_array*

calculate_EQD2_msk.m

INFO: This function calculates the equivalent dose in 2 Gy fractions (EQD2) for various radiotherapy schedules. It first simulates a standard reference schedule (2 Gy/fraction every 24 hours) using the model_msk.m function to construct a baseline spline interpolant, which maps the logarithm of surviving tumor cells to the accumulated dose. Then, it evaluates each user-provided schedule, calculates its final surviving cell count, and uses the baseline interpolant to find the mathematically equivalent dose had the treatment been delivered in standard 2 Gy fractions.

INPUT:
- *doses:* an array where each element represents the dose per fraction for a specific treatment schedule.
- *schedules*: cell array where each cell contains a 1D vector of administration times (in hours) corresponding to a specific treatment schedule.
- *params:* an array of 11 radiobiological parameters required by the MSK model.

OUTPUT:
- *EQD2_hist_array:* a 1D array containing the calculated final EQD2 values for each corresponding schedule provided in the input.

optimise_tcp_logistic_reox.m

INFO: Implementation of a Simulated Annealing optimization algorithm to find the best-fitting parameters for a logistic Tumor Control Probability (TCP) model using a maximum likelihood methed.

INPUT:
- *A:* cell array containing the experimental data.

- *par0:* starting point for the model parameters.
- *ind_opt:* indices of the specific parameters that we want to optimize (some will stay fixed).
- *Constraint:* a 2x13 matrix with the lower and upper bounds for each parameter.
- *cell_spline & v_k:* pre-computed spline functions to approximate binomial probabilities (in the case of large patient cohorts).
- *runs*: the number of cooling steps and the number of iterations per temperature step ([1000, 1000] typically).
- *T0 & dT*: initial temperature and the cooling rate

OUTPUT:
- *par_best:* best-fitting parameters.
- *cost_best*: minimum cost of the objective function.
- *EQD2_best*: EQD2 values for each schedule calculated using the best-fitting parameters.

MATLAB WORKSPACES

best_fits.mat

INFO: workspace containing the results of the best fits for LR and IR

Variables:

- **A_IR:** experimental data for Intermediate Risk patients. Each of the 6 columns contains: dose per fraction; number of fractions; overal treatment time (days); number of patients; vector with the treatment delivery times of each fraction (hors)
- **A_LR**: as above Low Risk patients
- **cell_spline_IR & V_K_IR**: pre-computed spline functions to approximate binomial probabilities for large_patients cohorts in A_IR
- **cell_spline_LR & V_K_LR**: pre-computed spline functions to approximate binomial probabilities for large_patients cohorts in A_LR
- **constraint_IR_Zharinov**: matrix with the lower and upper bounds for each parameter for IR dataset
- **constraint_LR_Zharinov**: matrix with the lower and upper bounds for each parameter for LR dataset
- **cost_IR_optimo:** minimum cost achieved for IR dataset
- **cost_LR_optimo:** minimum cost achieved for LR dataset
- **EQD2_IR_optimo:** EQD2 values for each schedule calculated using the best-fitting parameters for IR.
- **EQD2_LR_optimo:** EQD2 values for each schedule calculated using the best-fitting parameters for LR.
- **indice_optimizar:** indices of the specific parameters that we want to optimize (some will stay fixed). In our case [1, 2, 7, 9, 10, 12] which corresponds to [D50, gamma50, km, alpha/beta, OER_I, f_pro]
- **par_IR_optimo:** best-fitting parameters for LR.
- **par_LR_optimo:** best-fitting parameters for IR.

ic.mat

INFO: workspace containing the results of the calculation of confidence intervals using the profile likelihood methodology

Variables:

- **coste_LR/IR_*parameter*:** vector with the minimum costs found for the parameter values found in valor_LR/IR_parameter.

- **valor_LR/IR_*parameter:*** vector with the fixed values of that parameter used to obtain the confidence intervals

- **ic_IR/LR_*parameter*:** vectors with the confidence interval results for each parameter both for LR and IR datasets -> [lower limit of 95% confidence interval, optimal value, upper limit of 95% confidence interval]. Obtained through interpolation of the values reported in the above variables.

kcross.mat

INFO: workspace containing the results of the k-cross validation

Variables:

- **kcross_results_IR/kcross_results_LR:** matlab structure containing the results for the 50 kcross validations for IR and LR. Fields saved for each validation are:
  - idx_train: index of IR experimental points used for training
  - params: optimal parameters found
  - coste_training: minimun cost value obtained for the training subdataset
  - p_value_training: p-value obtained from the χ2 test for the training subdataset
  - coste_validacion: cost value obtained evaluating the optimal parameters (*params)* in the validation subdataset
  - p_value_validacion: p-value obtained from the χ2 test for the validation subdataset