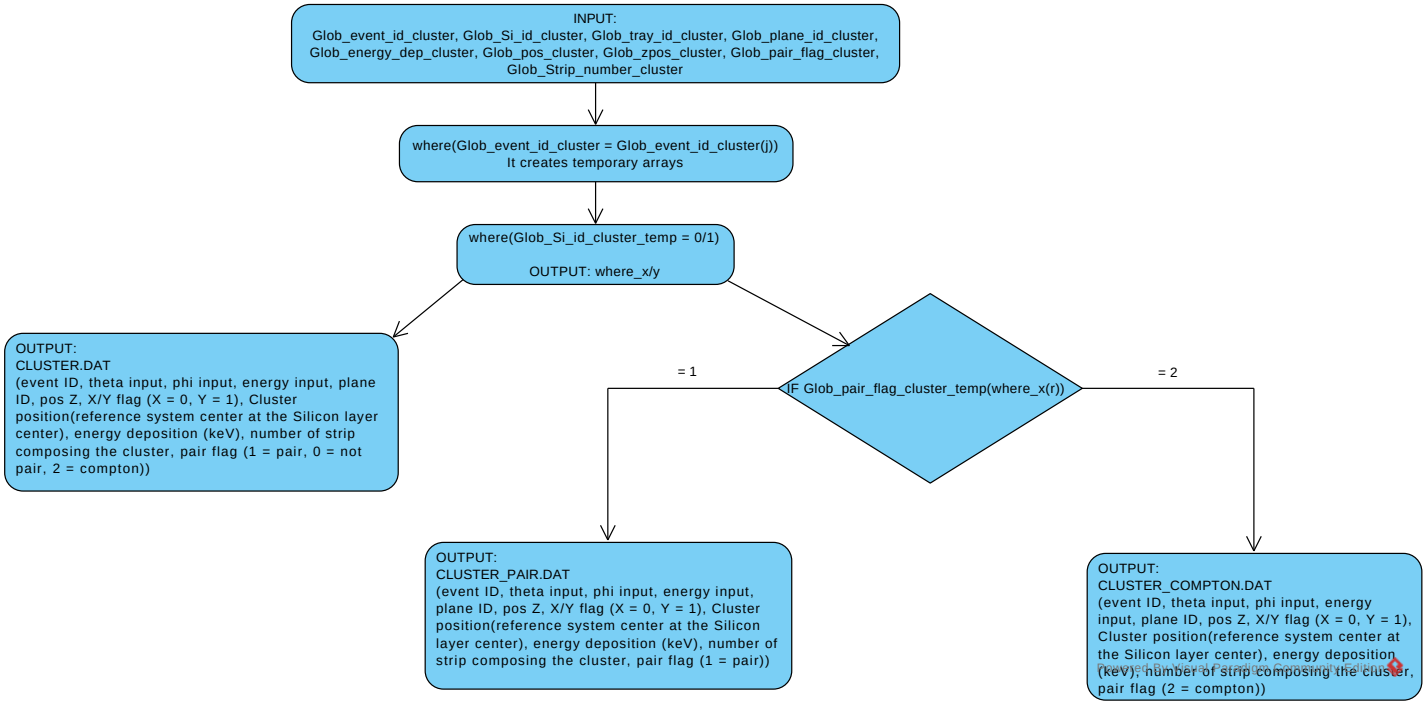
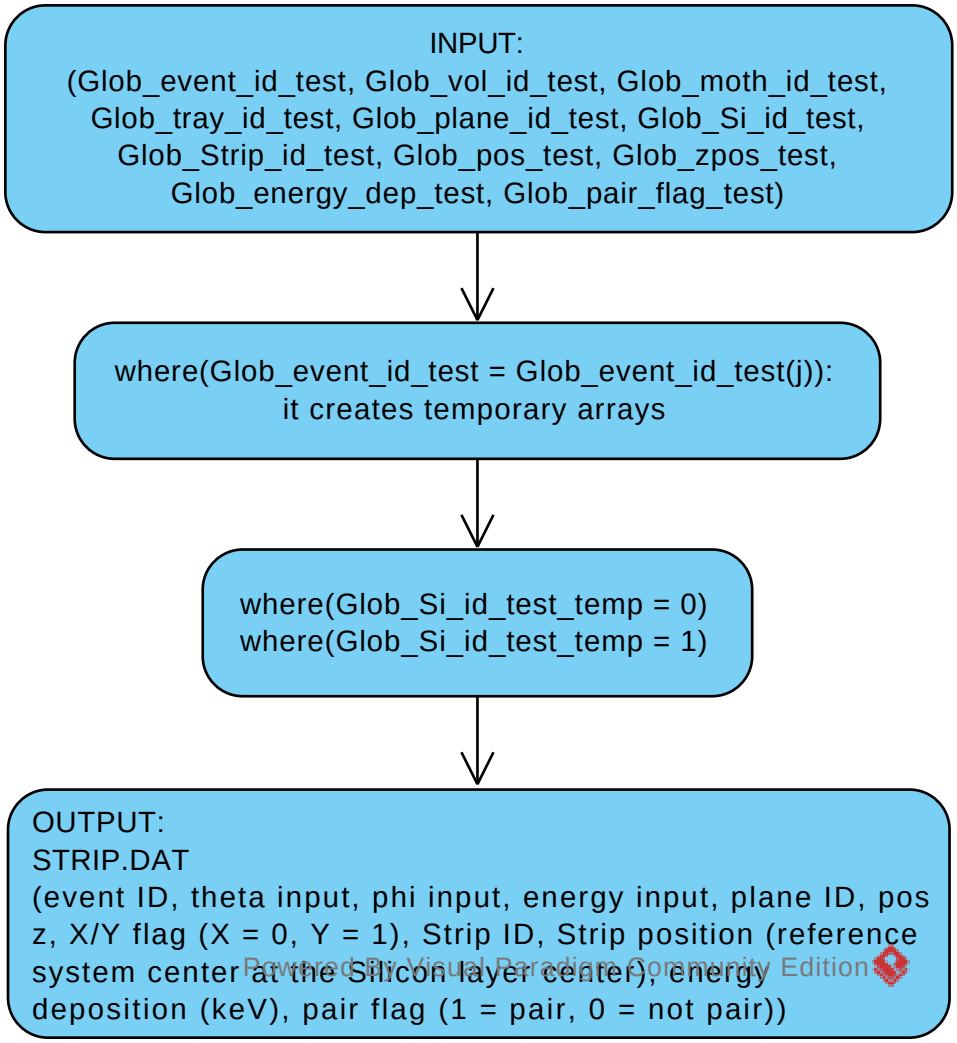
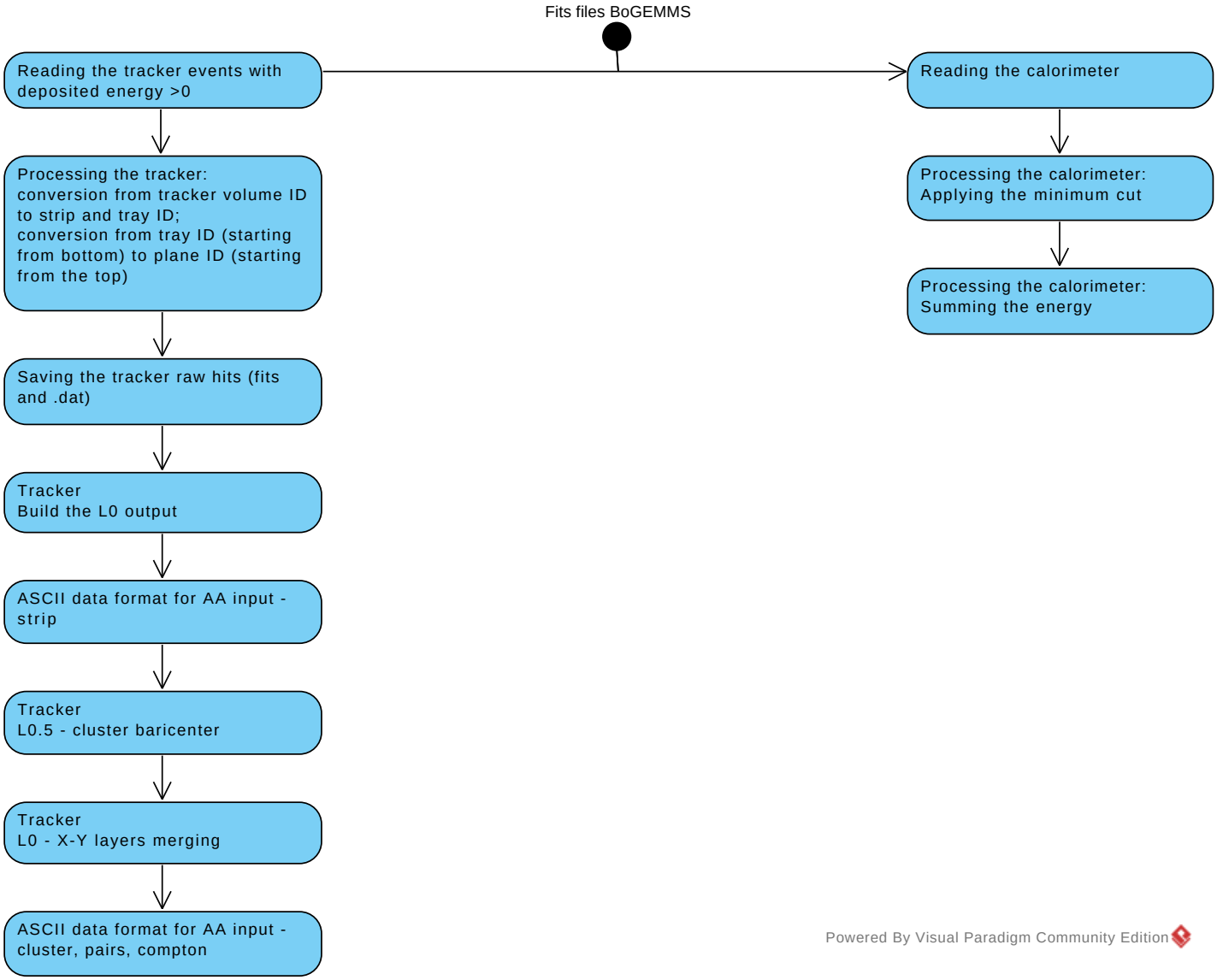
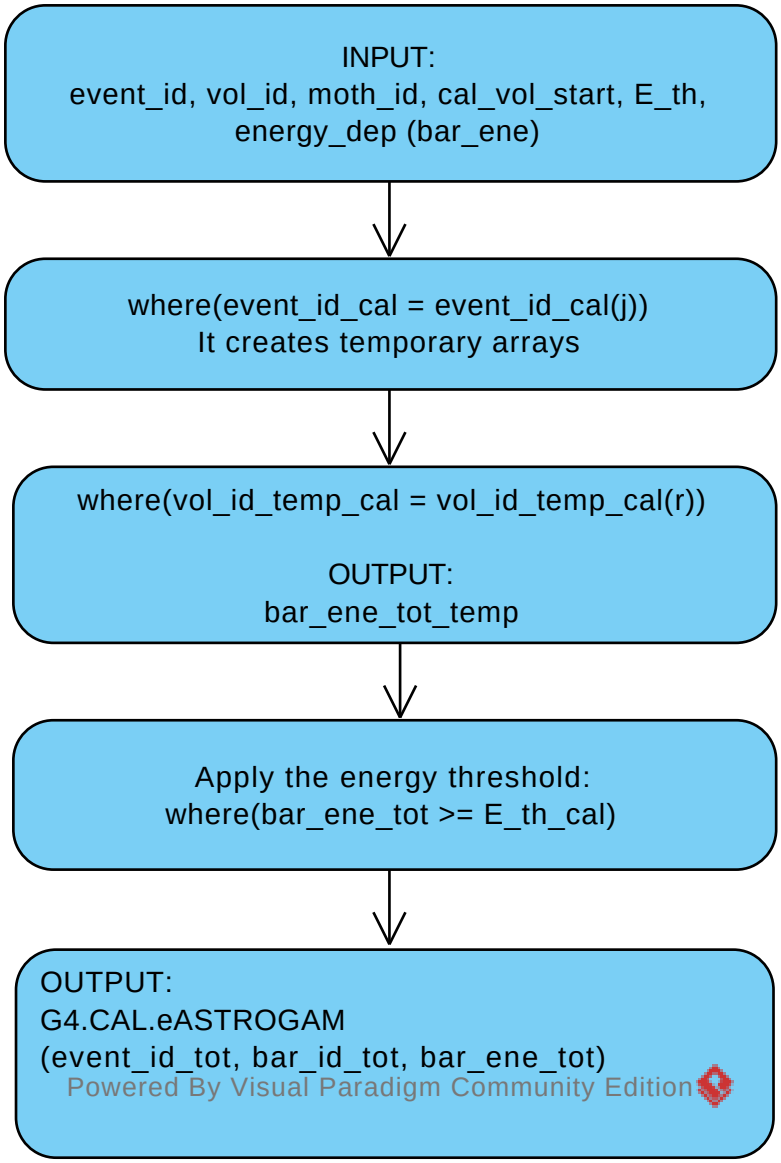
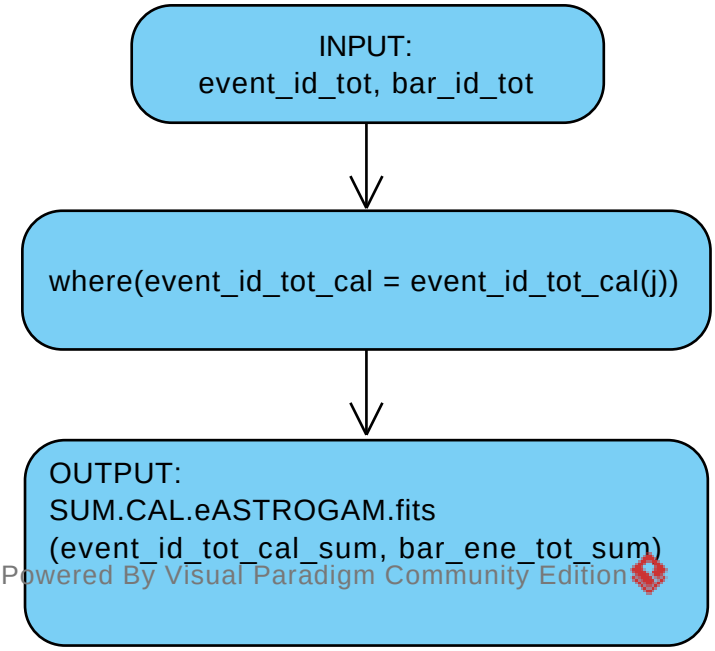
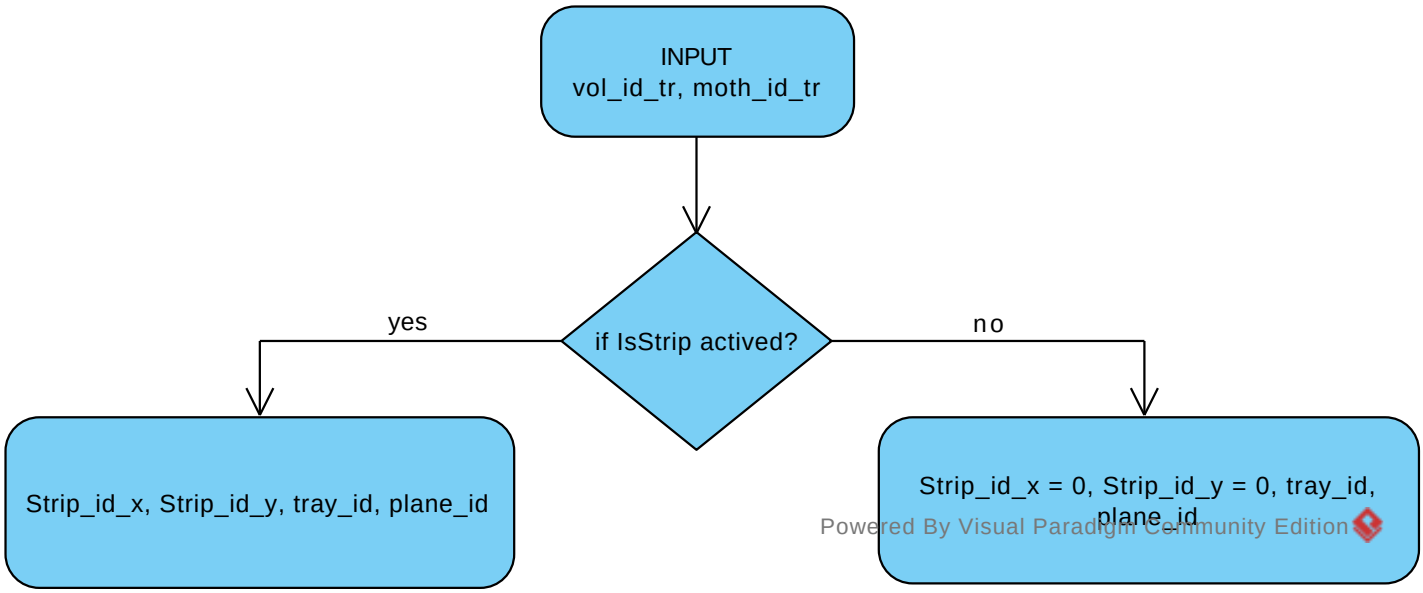

INPUT:
Glob_event_id_cluster, Glob_Si_id_cluster, Glob_tray_id_cluster, Glob_plane_id_cluster, Glob_energy_dep_cluster, Glob_pos_cluster, Glob_zpos_cluster, Glob_pair_flag_cluster, Glob_Strip_number_cluster
where(Glob_event_id_cluster = Glob_event_id_cluster(j))
It creates temporary arrays
where(Glob_Si_id_cluster_temp = 0/1)
OUTPUT: where_x/y
OUTPUT:
CLUSTER.DAT
(event ID, theta input, phi input, energy input, plane ID, pos Z, X/Y flag (X = 0, Y = 1), Cluster position(reference system center at the Silicon layer center), energy deposition (keV), number of strip composing the cluster, pair flag (1 = pair, 0 = not pair, 2 = compton))
IF Glob_pair_flag_cluster_temp(where_x(r))
= 1
= 2
OUTPUT:
CLUSTER_PAIR.DAT
(event ID, theta input, phi input, energy input, plane ID, pos Z, X/Y flag (X = 0, Y = 1), Cluster position(reference system center at the Silicon layer center), energy deposition (keV), number of strip composing the cluster, pair flag (1 = pair))
OUTPUT:
CLUSTER_COMPTON.DAT
(event ID, theta input, phi input, energy input, plane ID, pos Z, X/Y flag (X = 0, Y = 1), Cluster position(reference system center at the Silicon layer center), energy deposition (keV), number of strip composing the cluster, pair flag (2 = compton))

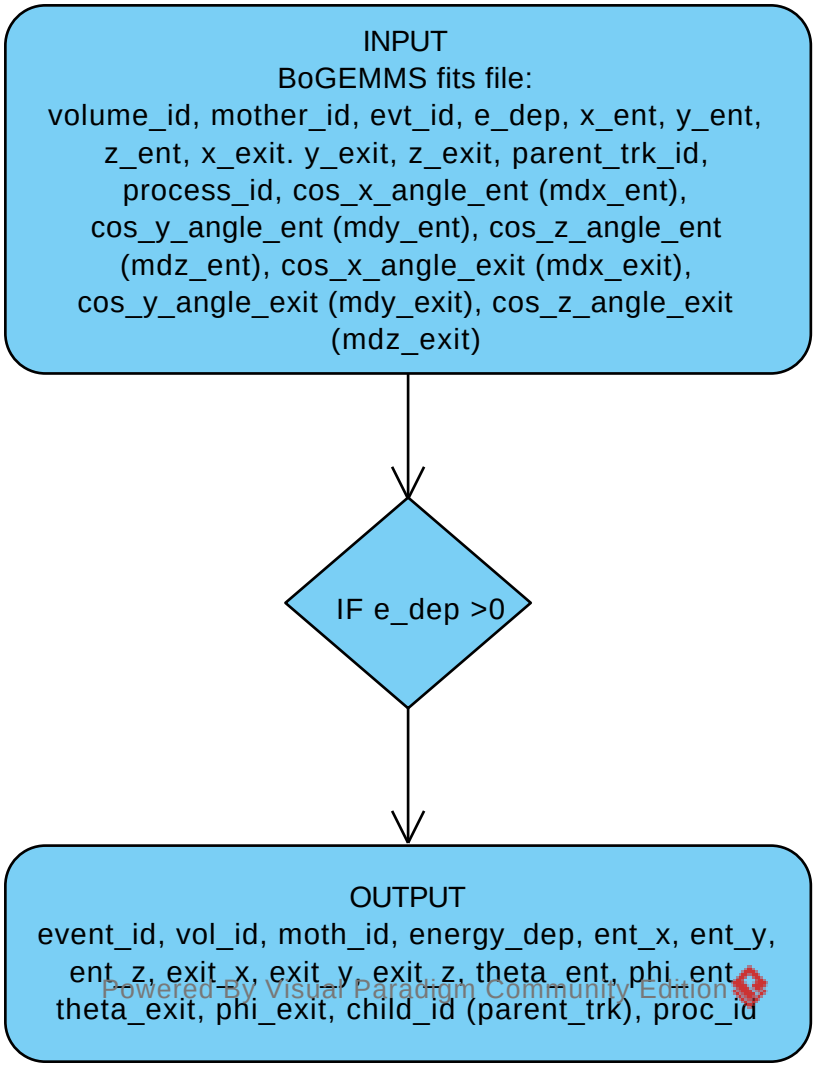

INPUT:
(Glob_event_id_test, Glob_vol_id_test, Glob_moth_id_test, Glob_tray_id_test, Glob_plane_id_test, Glob_Si_id_test, Glob_Strip_id_test, Glob_pos_test, Glob_zpos_test, Glob_energy_dep_test, Glob_pair_flag_test)
where(Glob_event_id_test = Glob_event_id_test(j)):
it creates temporary arrays
where(Glob_Si_id_test_temp = 0)
where(Glob_Si_id_test_temp = 1)
OUTPUT:
STRIP.DAT
(event ID, theta input, phi input, energy input, plane ID, pos z, X/Y flag (X = 0, Y = 1), Strip ID, Strip position (reference system center at the Silicon layer center), energy deposition (keV), pair flag (1 = pair, 0 = not pair))
Powered By Visual Paradigm Community Edition

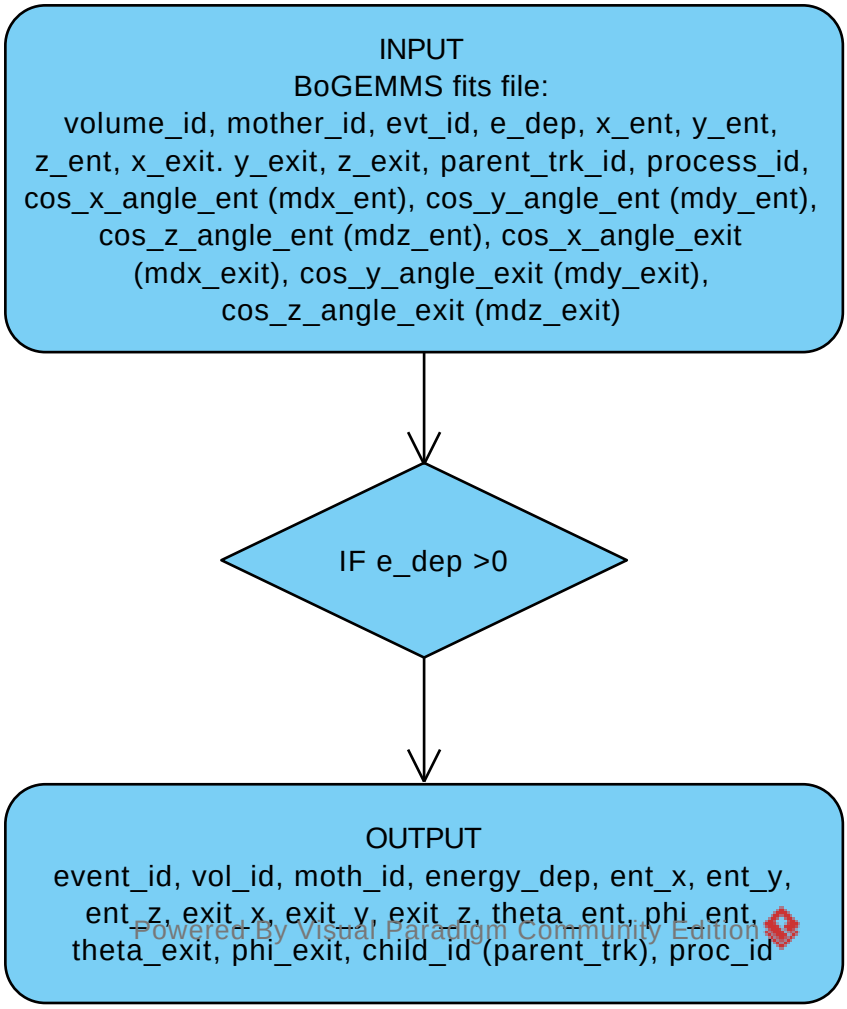

Fits files BoGEMMS
Reading the tracker events with deposited energy >0
Processing the tracker:
conversion from tracker volume ID to strip and tray ID;
conversion from tray ID (starting from bottom) to plane ID (starting from the top)
Saving the tracker raw hits (fits and .dat)
Tracker
Build the L0 output
ASCII data format for AA input - strip
Tracker
L0.5 - cluster baricenter
Tracker
L0 - X-Y layers merging
ASCII data format for AA input - cluster, pairs, compton
Reading the calorimeter
Processing the calorimeter:
Applying the minimum cut
Processing the calorimeter:
Summing the energy
Powered By Visual Paradigm Community Edition

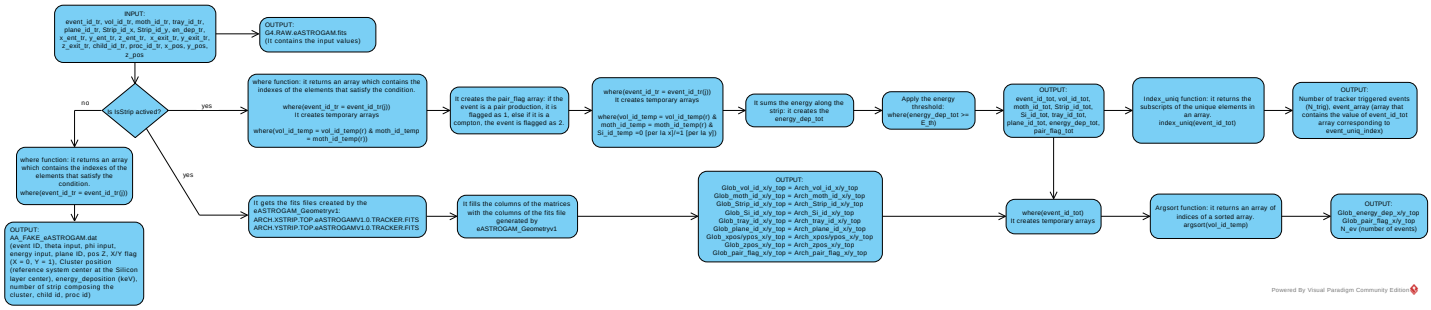

INPUT:
event_id, vol_id, moth_id, cal_vol_start, E_th,
energy_dep (bar_ene)
where(event_id_cal = event_id_cal(j))
It creates temporary arrays
where(vol_id_temp_cal = vol_id_temp_cal(r))
OUTPUT:
bar_ene_tot_temp
Apply the energy threshold:
where(bar_ene_tot >= E_th_cal)
OUTPUT:
G4.CAL.eASTROGAM
(event_id_tot, bar_id_tot, bar_ene_tot)
Powered By Visual Paradigm Community Edition

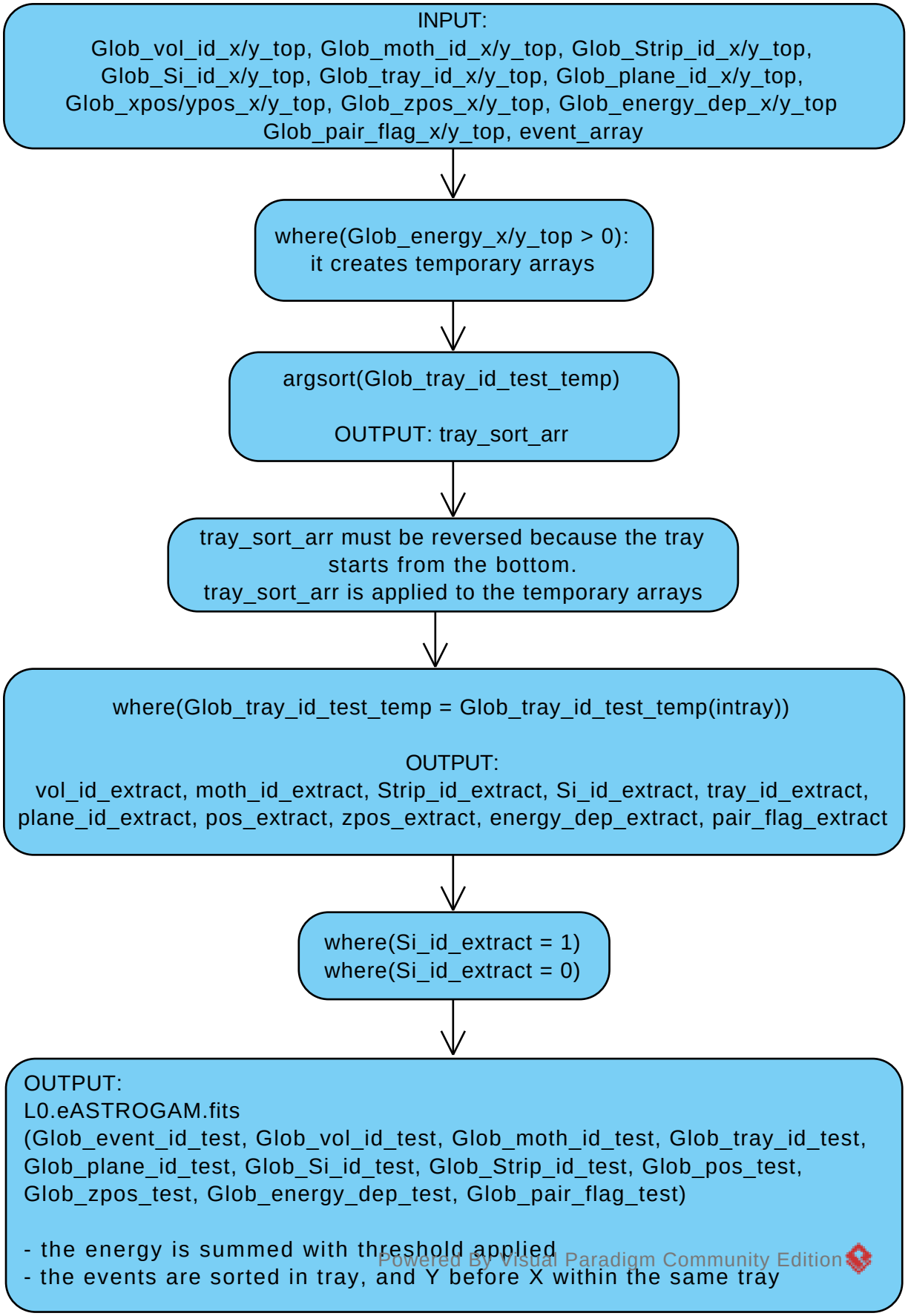

INPUT:
event_id_tot, bar_id_tot
where(event_id_tot_cal = event_id_tot_cal(j))
OUTPUT:
SUM.CAL.eASTROGAM.fits
(event_id_tot_cal_sum, bar_ene_tot_sum)
Powered By Visual Paradigm Community Edition

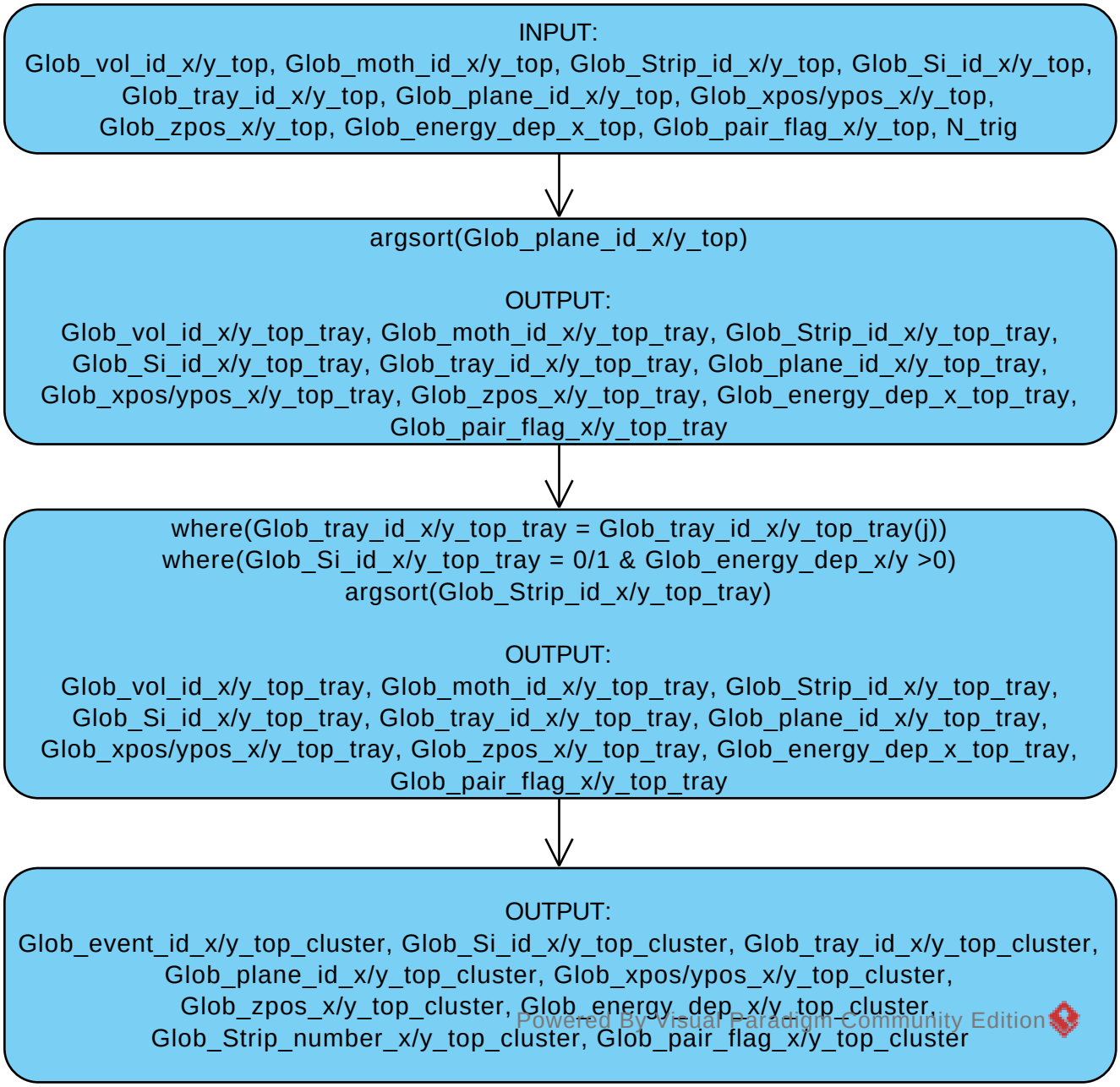

INPUT
vol_id_tr, moth_id_tr
if IsStrip actived?
yes
no
Strip_id_x, Strip_id_y, tray_id, plane_id
Strip_id_x = 0, Strip_id_y = 0, tray_id, plane_id
Powered By Visual Paradigm Community Edition

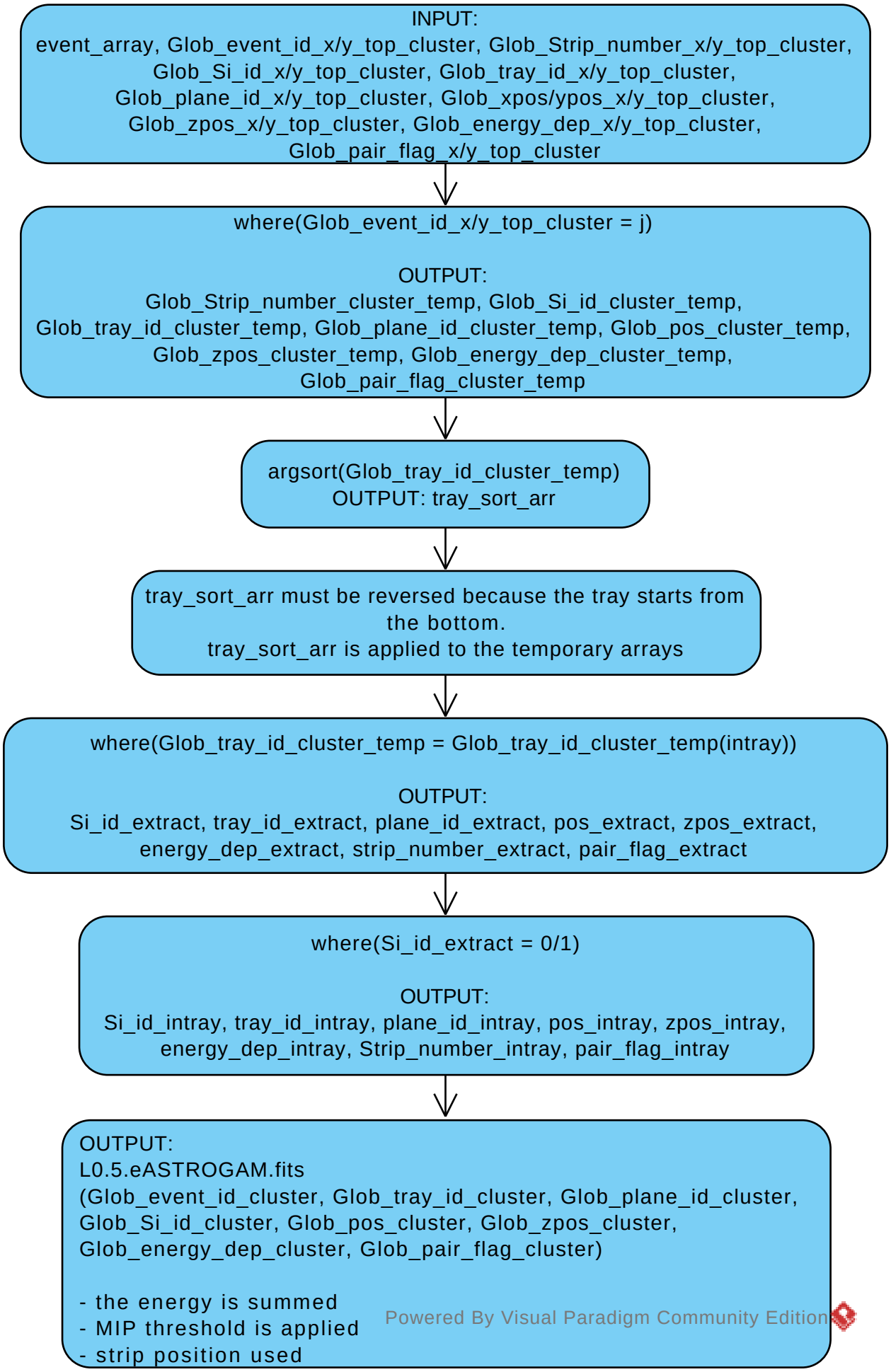

INPUT
BoGEMMS fits file:
volume_id, mother_id, evt_id, e_dep, x_ent, y_ent, z_ent, x_exit. y_exit, z_exit, parent_trk_id, process_id, cos_x_angle_ent (mdx_ent), cos_y_angle_ent (mdy_ent), cos_z_angle_ent (mdz_ent), cos_x_angle_exit (mdx_exit), cos_y_angle_exit (mdy_exit), cos_z_angle_exit (mdz_exit)
IF e_dep >0
OUTPUT
event_id, vol_id, moth_id, energy_dep, ent_x, ent_y, ent_z, exit_x, exit_y, exit_z, theta_ent, phi_ent, theta_exit, phi_exit, child_id (parent_trk), proc_id
Powered By Visual Paradigm Community Edition




INPUT:
event_id_tr, vol_id_tr, moth_id_tr, tray_id_tr, plane_id_tr, Strip_id_x, Strip_id_y, en_dep_tr, x_ent_tr, y_ent_tr, z_ent_tr, x_exit_tr, y_exit_tr, z_exit_tr, child_id_tr, proc_id_tr, x_pos, y_pos, z_pos
OUTPUT:
G4.RAW.eASTROGAM.fits
(It contains the input values)
is IsStrip actived?
no
yes
yes
where function: it returns an array which contains the indexes of the elements that satisfy the condition.
where(event_id_tr = event_id_tr(i))
OUTPUT:
AA_FAKE_eASTROGAM.dat
(event ID, theta input, phi input, energy input, plane ID, pos Z, X/Y flag (X = 0, Y = 1), Cluster position (reference system center at the Silicon layer center), energy_deposition (keV), number of strip composing the cluster, child id, proc id)
where function: it returns an array which contains the indexes of the elements that satisfy the condition.
where(event_id_tr = event_id_tr(i))
It creates temporary arrays
where(vol_id_temp = vol_id_temp(r) & moth_id_temp = moth_id_temp(r))
It creates the pair_flag array: if the event is a pair production, it is flagged as 1, else if it is a compton, the event is flagged as 2
where(event_id_tr = event_id_tr(i))
It creates temporary arrays
where(vol_id_temp = vol_id_temp(r) & moth_id_temp = moth_id_temp(r) & Si_id_temp =0 [per la x]/=1 [per la y])
It sums the energy along the strip: it creates the energy_dep_tot
Apply the energy threshold:
where(energy_dep_tot >= E_th)
OUTPUT:
event_id_tot, vol_id_tot, moth_id_tot, Strip_id_tot, Si_id_tot, tray_id_tot, plane_id_tot, energy_dep_tot, pair_flag_tot
Index_uniq function: it returns the subscripts of the unique elements in an array.
index_uniq(event_id_tot)
OUTPUT:
Number of tracker triggered events (N_trig), event_array (array that contains the value of event_id_tot array corresponding to event_uniq_index)
It gets the .fits files created by the eASTROGAM_Geometryv1:
ARCH.XSTRIP.TOP.eASTROGAMV1.0.TRACKER.FITS
ARCH.YSTRIP.TOP.eASTROGAMV1.0.TRACKER.FITS
It fills the columns of the matrices with the columns of the fits file generated by eASTROGAM_Geometryv1
OUTPUT:
Glob_vol_id_x/y_top = Arch_vol_id_x/y_top
Glob_moth_id_x/y_top = Arch_moth_id_x/y_top
Glob_Strip_id_x/y_top = Arch_Strip_id_x/y_top
Glob_Si_id_x/y_top = Arch_Si_id_x/y_top
Glob_tray_id_x/y_top = Arch_tray_id_x/y_top
Glob_plane_id_x/y_top = Arch_plane_id_x/y_top
Glob_xpos/ypos_x/y_top = Arch_xpos/ypos_x/y_top
Glob_zpos_x/y_top = Arch_zpos_x/y_top
Glob_pair_flag_x/y_top = Arch_pair_flag_x/y_top
where(event_id_tot)
It creates temporary arrays
Argsort function: it returns an array of indices of a sorted array.
argsort(vol_id_temp)
OUTPUT:
Glob_energy_dep_x/y_top
Glob_pair_flag_x/y_top
N_ev (number of events)
Powered By Visual Paradigm Community Edition




INPUT:
Glob_vol_id_x/y_top, Glob_moth_id_x/y_top, Glob_Strip_id_x/y_top, Glob_Si_id_x/y_top,
Glob_tray_id_x/y_top, Glob_plane_id_x/y_top, Glob_xpos/ypos_x/y_top,
Glob_zpos_x/y_top, Glob_energy_dep_x_top, Glob_pair_flag_x/y_top, N_trig
argsort(Glob_plane_id_x/y_top)
OUTPUT:
Glob_vol_id_x/y_top_tray, Glob_moth_id_x/y_top_tray, Glob_Strip_id_x/y_top_tray,
Glob_Si_id_x/y_top_tray, Glob_tray_id_x/y_top_tray, Glob_plane_id_x/y_top_tray,
Glob_xpos/ypos_x/y_top_tray, Glob_zpos_x/y_top_tray, Glob_energy_dep_x_top_tray,
Glob_pair_flag_x/y_top_tray
where(Glob_tray_id_x/y_top_tray = Glob_tray_id_x/y_top_tray(j))
where(Glob_Si_id_x/y_top_tray = 0/1 & Glob_energy_dep_x/y >0)
argsort(Glob_Strip_id_x/y_top_tray)
OUTPUT:
Glob_vol_id_x/y_top_tray, Glob_moth_id_x/y_top_tray, Glob_Strip_id_x/y_top_tray,
Glob_Si_id_x/y_top_tray, Glob_tray_id_x/y_top_tray, Glob_plane_id_x/y_top_tray,
Glob_xpos/ypos_x/y_top_tray, Glob_zpos_x/y_top_tray, Glob_energy_dep_x_top_tray,
Glob_pair_flag_x/y_top_tray
OUTPUT:
Glob_event_id_x/y_top_cluster, Glob_Si_id_x/y_top_cluster, Glob_tray_id_x/y_top_cluster,
Glob_plane_id_x/y_top_cluster, Glob_xpos/ypos_x/y_top_cluster,
Glob_zpos_x/y_top_cluster, Glob_energy_dep_x/y_top_cluster,
Glob_Strip_number_x/y_top_cluster, Glob_pair_flag_x/y_top_cluster
Powered By Visual Paradigm Community Edition


INPUT:
event_array, Glob_event_id_x/y_top_cluster, Glob_Strip_number_x/y_top_cluster,
Glob_Si_id_x/y_top_cluster, Glob_tray_id_x/y_top_cluster,
Glob_plane_id_x/y_top_cluster, Glob_xpos/ypos_x/y_top_cluster,
Glob_zpos_x/y_top_cluster, Glob_energy_dep_x/y_top_cluster,
Glob_pair_flag_x/y_top_cluster
where(Glob_event_id_x/y_top_cluster = j)
OUTPUT:
Glob_Strip_number_cluster_temp, Glob_Si_id_cluster_temp,
Glob_tray_id_cluster_temp, Glob_plane_id_cluster_temp, Glob_pos_cluster_temp,
Glob_zpos_cluster_temp, Glob_energy_dep_cluster_temp,
Glob_pair_flag_cluster_temp
argsort(Glob_tray_id_cluster_temp)
OUTPUT: tray_sort_arr
tray_sort_arr must be reversed because the tray starts from
the bottom.
tray_sort_arr is applied to the temporary arrays
where(Glob_tray_id_cluster_temp = Glob_tray_id_cluster_temp(intray))
OUTPUT:
Si_id_extract, tray_id_extract, plane_id_extract, pos_extract, zpos_extract,
energy_dep_extract, strip_number_extract, pair_flag_extract
where(Si_id_extract = 0/1)
OUTPUT:
Si_id_intray, tray_id_intray, plane_id_intray, pos_intray, zpos_intray,
energy_dep_intray, Strip_number_intray, pair_flag_intray
OUTPUT:
L0.5.eASTROGAM.fits
(Glob_event_id_cluster, Glob_tray_id_cluster, Glob_plane_id_cluster,
Glob_Si_id_cluster, Glob_pos_cluster, Glob_zpos_cluster,
Glob_energy_dep_cluster, Glob_pair_flag_cluster)
- the energy is summed
- MIP threshold is applied
- strip position used
Powered By Visual Paradigm Community Edition