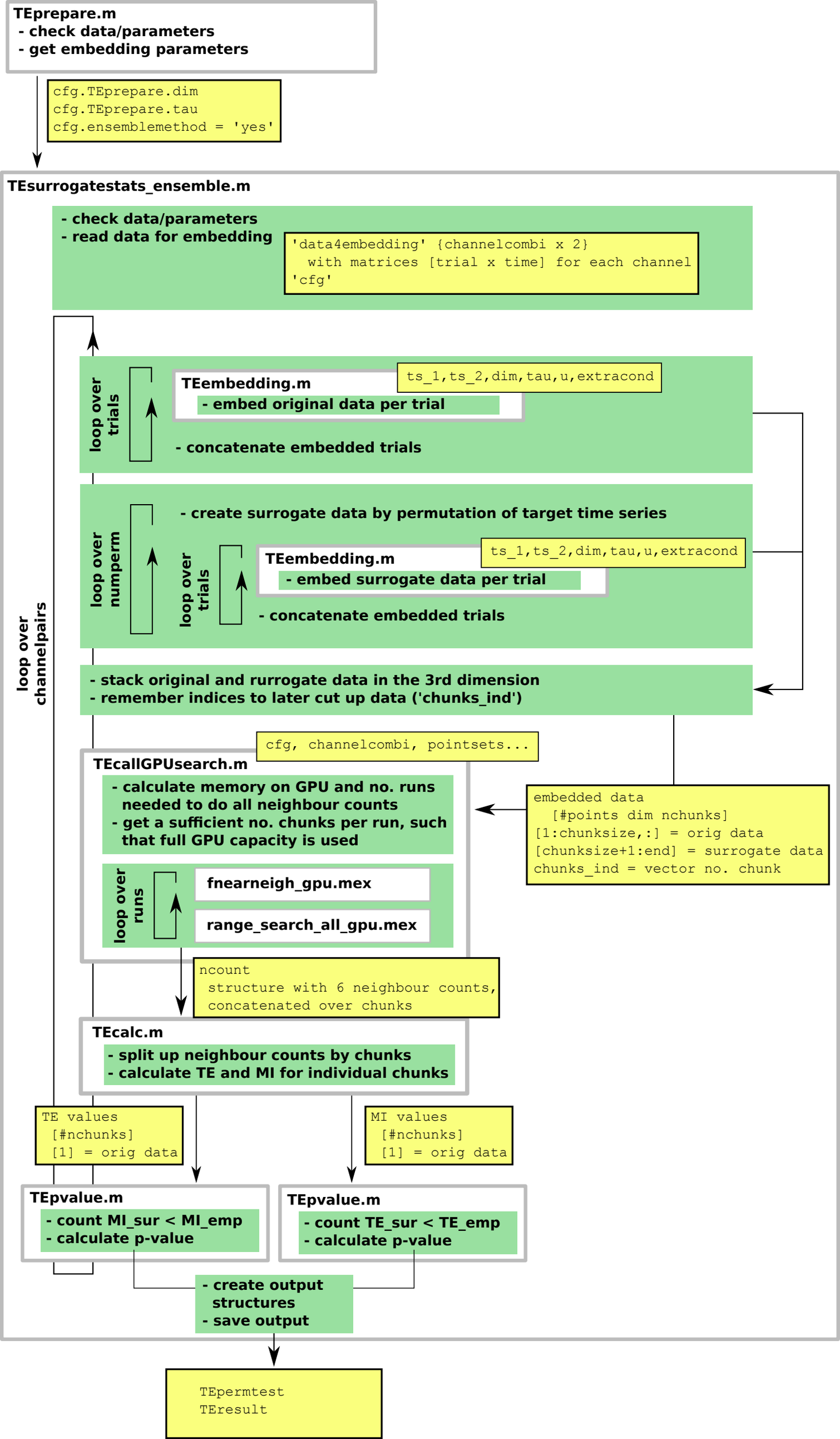
TEprepare.m
- check data/parameters
- get embedding parameters
cfg.TEprepare.dim
cfg.TEprepare.tau
cfg.ensemblemethod = 'yes'
TEsurrogatestats_ensemble.m
- check data/parameters
- read data for embedding
'data4embedding' {channelcombi x 2}
with matrices [trial x time] for each channel
'cfg'
loop over channelpairs
loop over trials
TEembedding.m
- embed original data per trial
ts_1,ts_2,dim,tau,u,extracond
- concatenate embedded trials
loop over numperm
- create surrogate data by permutation of target time series
loop over trials
TEembedding.m
- embed surrogate data per trial
ts_1,ts_2,dim,tau,u,extracond
- concatenate embedded trials
- stack original and rurrogate data in the 3rd dimension
- remember indices to later cut up data ('chunks_ind')
cfg, channelcombi, pointsets...
TEcallGPUsearch.m
- calculate memory on GPU and no. runs needed to do all neighbour counts
- get a sufficient no. chunks per run, such that full GPU capacity is used
loop over runs
fnearneigh_gpu.mex
range_search_all_gpu.mex
embedded data
[#points dim nchunks]
[1:chunksize,:] = orig data
[chunksize+1:end] = surrogate data
chunks_ind = vector no. chunk
ncount
structure with 6 neighbour counts,
concatenated over chunks
TEcalc.m
- split up neighbour counts by chunks
- calculate TE and MI for individual chunks
TE values
[#nchunks]
[1] = orig data
MI values
[#nchunks]
[1] = orig data
TEpvalue.m
- count MI_sur < MI_emp
- calculate p-value
TEpvalue.m
- count TE_sur < TE_emp
- calculate p-value
- create output structures
- save output
TEpermtest
TEresult