Codling Moth Pseudo Code

Hossein Noorazar *

Algorithm 1: Generate CMPOP files

Input : raw_file, params, start_year, end_year, lower=10, upper=31.11
Output: CMPOP_files (minus some columns such as ClimateGroup, latitude, etc.)

```
prepare_time_stuff(start_year, end_year);   //Nyears, Nrecords, NofVariables, Years, ind
create_ymdvalues (nYears, Years, leap.year)   // Generate Calendar
readbinarydata_addmdy(input_file, Nrecords, Nofvariables, ymd, ind)
add_dd_cumdd(data, lower, upper)   // Calculate daily and cumulative gdd
add day of year from 1 to 365/366
CodlingMothRelPopulation(params, data)   // compute relative population
append relative population to the columns of the data:  (tmax, tmin, dd, cum_dd,
cum_dd_F)
rename some of the columns:  "SumEgg", "SumLarva", "SumPupa", "SumAdult",
"dayofyear", "year", "month", "day"
CodlingMothPercentPopulation(params, data)   // Compute percentage population
append percentage pop.  to the rest of the data
return data
```

Algorithm 2: Generate CM files (prepareDataCMPOP)

Input : input_data, param, start_year, end_year, lower=10, upper=31.11
Output: CM files

```
prepare_time_stuff(start_year, end_year);   //Nyears, Nrecords, NofVariables, Years, ind
create_ymdvalues (nYears, Years, leap.year)   // Generate Calendar
readbinarydata_addmdy(input_file, Nrecords, Nofvariables, ymd, ind)
add_dd_cumdd(data, lower, upper)   // Calculate daily and cumulative gdd
add day of year from 1 to 365/366
add cumulative DD in celsius to data
compute relative population and append it to the data.
(CodlingMothRelPopulation(params, metdata))
compute percentage population and append it to the data.
(CodlingMothPercPopulation(params, metdata))
compute generations of adults and larva of all 4 generations by the beginning of
each month and append it to the data.
compute emergence and diapause and append it to the data.
compute when the 25%, 50% and 75% of generations are hit.
return CM_file
```

In the above algorithm stage_gen_toiterate is 16, 4 generations of eggs, 4 generations of larva, 4 of pupaes, 4 of adults.

*h.noorazar@wsu.edu

Algorithm 3: Codling Moth Percentage Population (CodlingMothPercPopulation)

Input : `params, metdata`
Output: `percentage_population`

`read parameters and generate an empty data frame with columns (dayofyear, year, month, Cum_dd_F)`
for $k = 1, \ldots,$ *stage_gen_toiterate* **do**
  relnum ← pweibull(data[Cum_dd_F], shape=params[i, 3], scale= params[i,4])
  add proper column name such as perc_egg_gen_1
end
generate columns (PercEgg, PercLarva, PercPupa, PercAdult)
for *all columns of data frame* **do**
  allrelnum$PercEgg[allrelnum[Cum_dd_F] > params [i,5] & allrelnum[Cum_dd_F] <= params [i,6]] ← allrelnum[allrelnum[Cum_dd_F] > params [i, 5] & allrelnum[Cum_dd_F] <= params [i,6], columnnumber]
end
return **allrelnum**

Algorithm 4: Codling Moth Relative Population (CodlingMothRelPopulation)

Input : `params, metdata`
Output: `relative_population`

choose a subset of data (day_of_year, month, year, cumdd_F)
for *all of the stages* **do**
  relnum ← dweibull(metdata[cumdd_F], shape= params[i, 3], scale= params[i, 4]) * 10000
  attach it to the data with proper name.
end
for *all stages such as eggs* **do**
  data [SumEgg] = data[EggGen1] + data[EggGen2] + data[EggGen3] + data[EggGen4]
end
return **data**

Algorithm 5: Add cumulative Degree Days (add_dd_cumdd)

Input : `metdata, lower, upper`
Output: `metdata with additional columns`

Compute the degree days and cumulative degree days according those 6 type of relations between tmin, tmax, lower and upper temps.
return **data**

Algorithm 6: (generate_vertdd)

Input : `combined_CMPOP, lower_temp = 4.5, upper_temp = 24.28`
Output: `metdata with additional columns`

generate vertical degree days
group by long, lat, climate scenario, climate group, year to generate cumulative vert. DD
generate 3 new columns for three type of apples (`pnorm(data[vert_cumdd_F], mean = 495.51, sd = 42.58, lower.tail = TRUE)`) return **data**

Algorithm 7: Diapause, absolute and relative population (diapause_abs_rel)

Input : `combined_CMPOP.RDS`
Output: `Absolute and relative population of diapause`

1 Look at the code please.

Algorithm 8: Diapause, absolute and relative population (diapause_abs_rel)

Input : `combined_CMPOP.RDS`
Output: `Absolute and relative population of diapause`

1 Look at the code please.

Algorithm 9: Bloom (bloom)

Input : `vertdd_combined_CMPOP_.RDS`
Output: `bloom`

1 Look at the code please