Package 'microPopGut'

February 23, 2022

Type Package

Title Process-Based Model of Microbes in Human Colon

Version 1.0

Date 2022-02-23

Description This model uses the microPop package to simulate the dynamics of microbes in the human colon. The colon is divided into three compartments (proximal, transverse and distal). The microbes are divided into 10 functional groups and 10 metabolites are modelled. Incoming substrate is divided into protein and carbohydrates (resistant starch and non-starch polysaccharides (NSP)) and can be fluctuating to simulate meals eaten. Bowel movements can also be simulated.

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

VignetteBuilder R.rsp

Depends microPop (>= 1.6), deSolve, stats, grDevices, graphics

Suggests testthat, rmarkdown, knitr, R.rsp

Collate 'computeSCFA.R' 'entryRateFuncMPG.R' 'growthLimFuncMPG.R' 'inflowFuncs.R' 'makeInflowMat.R' 'microPopGut.R' 'removalRateFuncMPG.R' 'pHLimFuncMPG.R' 'pHFuncMPG.R' 'newRateFunctions.R' 'plotMPG.R' 'settings.R' 'subsetFunc.R' 'verification.R'

NeedsCompilation no

Author Helen Kettle [aut, cre] (<https://orcid.org/0000-0002-6960-9733>)

Maintainer Helen Kettle <Helen.Kettle@bioss.ac.uk>

R topics documented:

`computeSCFA` *computeSCFA*

Description

calculates SCFA (mM) from model results

Usage

```
computeSCFA(out, SCFAnames, molarMass, waterName = "H2O")
```

Arguments

`out`	is the output of microPopGut()$solution[[n]]. It is a matrix of all model state variables (columns) at each output time (rows) for compartment n
`SCFAnames`	vector of strings which are the names of the SCFA
`molarMass`	is the molar mass of each resource
`waterName`	is the name of the water variable

Value

returns a matrix with SCFA in each column and rows over time

`entryRateFuncMPG` *entry Rate Function for microPopGut*

Description

Return the rate of entry to the system for any state variable

Usage

```
entryRateFuncMPG(varName, varValue, stateVarValues, time, inflowRate, parms)
```

Arguments

varName	(string) Name of state variable of interest (resource name or strain name)
varValue	(scalar) value of state variable of interest
stateVarValues	(named vector) values of all state variables
time	(scalar) time
inflowRate	(named vector) on inflow rates (specified in SysInfo files)
parms	List containing all system parameters

Value

(scalar) rate of entry (quantity per unit time) for any state variable

---

growthLimFuncMPG *growth rate limitation function for microPopGut*

Description

Returns the value of growthLim (must lie in interval [0,1] i.e. unitless) of strainName on varName which is used to scale the maximum growth rate Contains two options - one for essential resources and one for substitutable resources (based on Ballyk and Wolkowicz, 1993)

Usage

```
growthLimFuncMPG(
  strainName,
  groupName,
  pathName,
  varName,
  resourceValues,
  allSubType,
  strainHalfSat,
  stateVarValues,
  parms
)
```

Arguments

strainName	Name of the strain that is being looped through in the ODE solver
groupName	Name of microbial group that is being looped through in the ODE solver
pathName	Name of metabolic path (e.g. path1) that is being looped through in the ODE solver
varName	(string) Name of variable (resource) of interest
resourceValues	State vector of resources (with names)
allSubType	Vector of strings (with names corresponding to the resourceNames) which describes the type of each resource ('Rtype') - Rtypes are S (substitutable resource), Se (essential resource), Sb (booster resource), Sm (microbial resource), P (product) and Pb (biomass product)

`strainHalfSat`
: Vector (with names corresponding to the resourceNames) of half-saturation constants for the given strain. If resource is not a substrate for the given strain, the value is NA

`stateVarValues`
: State vector (resources and microbes) (with names)

`parms`
: list of parameter values

Value

scalar giving limitation on growth rate - must be >=0 and <=1

`makeSubstrateMat` *makeSubstrateMat this function calls mealsPattern(), mealComposition() and smallIntestine()*

Description

makeSubstrateMat this function calls mealsPattern(), mealComposition() and smallIntestine()

Usage

```
makeSubstrateMat(
  substrateNames,
  substrateMeans,
  num.days = 2,
  breakfast.start = 7,
  lunch.start = 13,
  dinner.start = 19,
  time.step.h = 0.1,
  meal.duration.h = 0.5,
  fluc.comp = TRUE,
  time.to.reach.colon.h = 7,
  gamma.mag = 1,
  RS.frac = 0.78,
  showPlot = TRUE,
  saveFig = FALSE,
  waterName = "H2O",
  waterMean = 1100
)
```

Arguments

`substrateNames`
: a vector of substrate names

`substrateMeans`
: a named vector of substrate means, e.g. c(A=1,B=2,C=3)

`num.days`
: number of days simulation is over

`breakfast.start`
: time to start breakfast (24h clock), default is 7am

`lunch.start`
: time to start lunch (24h clock), default is 1pm

`dinner.start`	time to start dinner (24h clock), default is 7pm
`time.step.h`	the model time step converted from days to hours
`meal.duration.h`	length of time for a meal (in hours)
`fluc.comp`	LOGICAL. TRUE for fluctating meal composition
`time.to.reach.colon.h`	time from ingestion to reaching colon (hours)
`gamma.mag`	a number controlling the shape of the gamma function
`RS.frac`	Fraction of carbs that is RS
`showPlot`	produce plot of inflow
`saveFig`	save plot of inflow
`waterName`	string for the name of the water variable. Default is 'H2O'
`waterMean`	number for mean inflow of water to colon (g/d). Default is 1100 g/d

Value

a matrix for substrate inflow (g/d) for each substrate and water (cols) at each model time step (rows). The first col is time in days.

`mealComposition` *mealComposition takes step function from mealsPattern() (in hours) then adds in dietary composition for each meal. This is randomised using a gamma distribtion (controlled by scale argument)*

Description

mealComposition takes step function from mealsPattern() (in hours) then adds in dietary composition for each meal. This is randomised using a gamma distribtion (controlled by scale argument)

Usage

```
mealComposition(
  mealsTS,
  substrateNames,
  substrateMeans,
  scale,
  fluc.comp,
  num.days,
  waterName,
  showPlot = FALSE,
  saveFig = FALSE
)
```

Arguments

mealsTS	output from mealsPattern()
substrateNames	a vector of substrate names
substrateMeans	a named vector of substrate means, e.g. c(A=1,B=2,C=3)
scale	a named vector controlling the shape of the gamma function, e.g. c(A=0.5,B=1,C=1.5)
fluc.comp	LOGICAL. TRUE for fluctating meal composition
num.days	Scalar Number of days to simulate
waterName	string specifying the name of water e.g. 'H20', 'water' etc
showPlot	FALSE If TRUE will plot meal composition
saveFig	FALSE If TRUE image will be saved as eps

Value

output is a matrix with a column for each substrate over time

---

mealsPattern *mealsPattern*

Description

this function generates a step function which is 1 for eating time and 0 if not eating. This is then converted to have a mean of 1 over the model simulation time Note this function is in hours (unlike rest of package)

Usage

```
mealsPattern(
  num.days = 3,
  breakfast.start = 7,
  lunch.start = 13,
  dinner.start = 19,
  time.step.h = 1/60,
  meal.duration.h = 0.5,
  showPlot = FALSE
)
```

Arguments

num.days	number of days simulation is over
breakfast.start	time to start breakfast (24h clock), default is 7am
lunch.start	time to start lunch (24h clock), default is 1pm
dinner.start	time to start dinner (24h clock), default is 7pm
time.step.h	the model time step converted from days to hours
meal.duration.h	length of time for a meal (in hours)
showPlot	Logical produce plot of inflow (defaults to FALSE)

Value

is a 2 column matrix with time in hours (col 1) and meals (mean=1)

`microPopGut` *microPopGut runs the human colon version of microPop*

Description

microPopGut runs the human colon version of microPop

Usage

```
microPopGut(
  numDays,
  time.step,
  transitTime,
  microbeNames,
  microbeNames.short,
  numStrains = 1,
  absorption = c(water = 3, SCFA = 9.6),
  pH.pars = list(pH.fixed = c(5.7, 6.2, 6.6), fix.pH = FALSE, pHLimit = TRUE),
  colon.pars = list(vol.litres = c(0.41, 0.98, 1.63), compartmentNames = c("prox
    "trans", "dist")),
  bowel.movements = list(BM.duration.h = 15/60, frac.distal.emptied = 0.9, BMpd
    start.BM.time = list(7, c(7, 19), c(7, 15, 21))),
  init = list(C = 1, P = 1, B = 10, Acetate = 0.3606, Propionate = 0.1482, Butyr
    0.1762, W = 100),
  inflow = list(C = 50, P = 10, W = 1100, RS.frac = 0.78),
  meals = list(seed = 1, fluc.inflow = FALSE, fluc.subst.comp = FALSE, plotInflo
    FALSE, saveInflowFig = FALSE, breakfast.start = 7, lunch.start = 13, dinner.
    19, meal.duration.h = 0.5, time.to.reach.colon.h = 7, gamma.mag = 1),
  mucin = list(mucin.rate = 5, RS.frac.mucin = 0.5, mucin.degrader = c("Bacteroi
    "NoButyFibreDeg"), halfSat.mucin = 0.5, P.frac.mucin = 0.2, C.frac.mucin = 0
  odeFunc = microPop::derivsDefault,
  oneStrainRandomParams = FALSE,
  microbeMolarMass = 113,
  bacCutOff = 1e-14,
  plotOptions = list(yLabel = "Concentration (g/L)", xLabel = "Time", plotFig =
    sumOverStrains = FALSE, saveFig = FALSE, figType = "eps", figName = "microPo
  odeOptions = list(atol = 1e-06, rtol = 1e-06, method = "lsoda"),
  strainOptions = list(randomParams = c("halfSat", "yield", "maxGrowthRate",
    "pHtrait"), seed = 1, distribution = "uniform", percentTraitRange = 0, maxPH
    0, applyTradeOffs = FALSE, tradeOffParams = NULL, paramsSpecified = FALSE,
    paramDataName = NULL),
  checkingOptions = list(checkMassConv = FALSE, balanceTol = 0.01, reBalanceStoi
    FALSE, stoiTol = 0.1, checkForNegs = TRUE, negTol = -0.01, checkStoichiomBal
    TRUE),
  ...
)
```

Arguments

`numDays`	number of days over which to run simulation
`time.step`	simulation time step (in days)
`transitTime`	time taken to travel through colon (d)
`microbeNames`	named vector with names of microbial data frames
`microbeNames.short`	named vector with shortened names for plotting
`numStrains`	scalar if all groups have the same number of strains or a named vector specifying the number of strains in each microbial group.
`absorption`	named vector with specific absorption rates (/d) for water and SCFA. Default is c('water'=3,'SCFA'=9.6)
`pH.pars`	list controlling pH in the model. Default is list(pH.fixed=c(5.7,6.2,6.6),fix.pH=FALSE,pHLimit=TRU • pH.fixed=c(5.7,6.2,6.6) pH values in prox, trans and distal compartments - used if fix.pH is TRUE • fix.pH=FALSE Set to TRUE to keep pH fixed. If FALSE, pH varies with SCFA concentrations • pHLimit=TRUE If TRUE then microbial growth is pH limited
`colon.pars`	list of parameters describing colon. Default is list(vol.litres=c(0.41,0.98,1.63),compartmentNames=c(
`bowel.movements`	List of parameters for modelling bowel movements. Default is list(BM.duration.h=15/60,frac.distal.en BMpd=0,start.BM.time=list(7,c(7,19),c(7,15,21)) • BM.duration.h time for bowel movement in hours • frac.distal.emptied fraction of the distal colon that is emptied by bowel movemen • BMpd number of bowel movements per day • start.BM.time list of times of bowel movements for 1, 2, or 3 BM per day
`init`	list of initial values for carbohydrates (C), protein (P), microbes (B), Acetate, Propionate and Butyrate and water (W) in grams. Default values are: list(C=1,P=1,B=10,Acetate=0.3(W=100). SCFA initial values are based on concentrations of 60 mM, 20 mM, 20 mM for acetate, propionate and butyrate. The microbial mass is split evenly between the microbial groups. The initial conditions are same in each model compartment.
`inflow`	list specifying the inflow of substrates (C, P) and water (W) in g/d, to the colon. Default values are: list(C=50,P=10,W=1100,RS.frac=0.78). The RS.frac is the fraction of RS in the carbohydrate i.e. RS/(RS+NSP) assuming RS+NSP=C
`meals`	list of parameters describing meals. Default is list(seed=1,fluc.inflow=FALSE,fluc.subst.comp=FALS • seed seed for random number generator used for stochastic meal composition only (fluc.subst.comp=TRUE) • fluc.inflow FALSE if incoming substrate is constant i.e. TRUE means meals • fluc.subst.comp FALSE change to TRUE for substrate composition varying randomly • plotInflow FALSE change to TRUE to plot the incoming substrate • saveInflowFig FALSE Change to TRUE to save inflow figs • breakfast.start time breakfast is eaten (24 h clock) • lunch.start time lunch is eaten (24 h clock) • dinner.start time dinner is eaten (24 h clock)

- meal.duration.h time taken to eat a meal (in hours)
- time.to.reach.colon.h time taken to move through stomach and small intestine etc (h)
- gamma.mag (positive number) this determines the level of variation in diet fluctuations (1.1 will give bigger fluctuations, 0.9 smaller).

`mucin` list of parameters describing mucins. Default is list(mucin.rate=5,RS.frac.mucin=0.5,mucin.degrader=

- mucin.rate rate of mucin (g/d) secreted into colon
- RS.frac.mucin fraction of C in mucin that is RS
- mucin.degrader microbes that break down mucin
- halfSat.mucin half saturation constant (g/l of B) for mucin breakdown
- P.frac.mucin fraction of mucin that is protein
- C.frac.mucin fraction of mucin that is carbohydrate

`odeFunc` The function the ODE solver in microPopModel() will use - the default is derivsDefault provided by the package but if the user wants to make significant changes a new ODE function file can be used. See ?derivsDefault

`oneStrainRandomParams`
Logical to allow randomization of parameters even if there is only one strain. The default is FALSE which means that if numStrains=1 then the group params are used; if numStrains>1 then the parameters are automatically randomised according to info given in strainOptions. If oneStrainRandomParams=TRUE then even if there is only one strain its parameters will be randomised according to info given in strainOptions.

`microbeMolarMass`
Scalar. Mass of 1 mole of microbes - default is 113g/mol (Batstone et al., 2002)

`bacCutOff` Scalar. Amount of bacteria below which the bacteria are considered to have left the system and can't grow, default =1e-14. If this is set to zero then bacteria will always be able to grow again as zero is never reached.

`plotOptions` List containing instructions for plotting: Default is list(plotFig=TRUE, sumOverStrains=FALSE, saveFig=FALSE, figType='eps', figName='microPopFig', yLabel='Concentration (g/L)', xLabel='Time').
To turn off plot generation set plotFig=FALSE. If there are multiple strains these are all plotted if sumOverStrains=FALSE, otherwise they will be summed over each group. To save plot, saveFig=TRUE, figType (format) can be 'eps','png', 'pdf' or 'tiff' and is specified in figType (string), the name is figName (string) to which the string 'Microbes' or 'Resources' will be added for the respective plots.

`odeOptions` List containing instructions for the ODE solver ('deSolve'). Default: list('atol'=1e-6,'rtol'=1e-6,'method'='lsoda'). See ?ode for more details.

`strainOptions`
List containing instructions for specifying strain parameters. Default: list(randomParams=c('halfSat', 'yield', 'maxGrowthRate', 'pHtrait'), seed=1, distribution='uniform', percentTraitRange=0, maxPHshift=0., applyTradeOffs=FALSE, tradeOffParams=NULL, paramsSpecified=FALSE, paramDataName=NULL).

- randomParams (vector) specifying which parameters need to be stochastically generated.
- seed (number) seed for random number generator.
- distribution (string) - either 'uniform' or 'normal' specifying the shape of the distribution from which to draw the random strain parameters.

- percentTraitRange (number) this is the percentage either side of the group parameter value which the strain parameter may range e.g. if percentTraitRange=10 then range is 0.9x to 1.1x for group mean x.
- maxPHshift (number) pH units to range over.
- applyTradeOffs (logical) to trade off good' and bad' parameter values.
- tradeOffParams (vector of two strings) - parameters to trade off against each other. Note that pHtrait can not be traded off as whether this trait is good or bad depends on the environmental pH.
- paramsSpecified (logical) TRUE if strain parameters are read in from a file (whose name is specified in paramDataName). The file must have colnames c(strainName, paramName, paramVal, paramUnit, resource,path) and where strainName is in format 'groupName.i' where i is the strain number.

`checkingOptions`

(List) Default is list(checkMassConv=FALSE, balanceTol=1e-2, reBalanceStoichiom=FALSE, stoiTol=0.1, checkForNegs=TRUE, negTol=-1e-2).

- checkMassConv=TRUE checks for mass conservation in the ODE solver with a tolerance of 'balanceTol' (default is FALSE).
- reBalanceStoichiom will check the mass balance of the stoichiometries on every metabolic path and rebalance if these are not conserving mass within a tolerance of stoiTol (a warning message will be issued). Rebalancing will only affect the final solution if the pathway contains only essential resources (Rtype 'Se') and microbial biomass is a product (Rtype 'Pb').
- checkForNegs If TRUE the function checkSolution is called and the solution for each variable, x, is checked for negative values that are greater in magnitude than negTol*max(x). If negative values occur then the solution is incorect and either the problem is incorrectly specified or the tolerances in the ODE solver need to be smaller.

`...` Add your own inputs

Value

The output is a list containing lists called 'solution', 'params' and myPars. These lists have 3 components (one for each model compartment (i.e. the proximal, transverse and distal colon). Solution[[n]] (for n=1,2,or 3 depending on compartment) is a matrix where rows are points in time and the columns are the state variables. params[[n]] contains all the information needed to run the model. Use names(out$params[[1]]) to see contents.

`newRateFuncs` *newRateFuncs for microPopGut*

Description

List of functions that are used by the ODE solver these functions can be changed by the user but all must be listed.

Usage

```
newRateFuncs
```

Format

An object of class `list` of length 11.

Details

rateFuncsDefault=list(pHFunc=pHFuncDefault, pHLimFunc=pHLimFuncDefault, extraGrowthLimFunc=extraGrowthLimFuncDefault, growthLimFunc=growthLimFuncDefault, combineGrowthLimFunc=combineGrowthLimFuncDefault, uptakeFunc=uptakeFuncDefault, productionFunc=productionFuncDefault, combinePathsFunc=combinePathsFuncDefault, massBalanceFunc=massBalanceFuncDefault, entryRateFunc=entryRateFuncDefault, removalRateFunc=removalRateFuncDefault)

note that in these functions, the parms list is intrinsic to microPop whereas the myPars list is defined by the inputs to microPopGut() and then added to parms

---

`pHFuncMPG` *pH Function for microPopGut*

Description

Return the value of pH in pH units

Usage

```
pHFuncMPG(time, parms, stateVarValues)
```

Arguments

`time`	(scalar). The current time point in the ODE solver.
`parms`	List which contains all information required by the ODE solver
`stateVarValues`	State vector (resources and microbes) (with names)

Value

(scalar) pH at the given time

`pHLimFuncMPG` *pH Limitation Function for microPopGut*

Description

Return the value of pHLim (must lie in interval [0,1])

Usage

```
pHLimFuncMPG(strainName, groupName, pH, parms)
```

Arguments

strainName	Name of the strain that is being looped through in the ODE solver
groupName	Name of microbial group that is being looped through in the ODE solver
pH	(scalar). The current pH value.
parms	List of all parameters

Value

(scalar) pH limitation (0 to 1)

---

plotMPG *plotMPG*

Description

Basic plot of the model results

Usage

```
plotMPG(m.out)
```

Arguments

m.out	is the output of microPopGut(). It is a list containing the following objects: solution, params, myPars, etc

---

removalRateFuncMPG *Removal Rate Function for microPopGut*

Description

Return the rate of removal of any state variable from the system This is called in the ODE derivs func

Usage

```
removalRateFuncMPG(varName, varValue, stateVarValues, time, washOut, parms)
```

Arguments

varName	(string) Name of state variable of interest (this is group name or a resource name - NOT a strain name)
varValue	(scalar) value of state variable of interest
stateVarValues	(named vector) values of all state variables
time	(scalar) time
washOut	(named vector) of wash out rates (per unit time) of groups and resources (specified in SysInfo files)
parms	List containing all system parameters

Value

(scalar) rate of removal (quantity per unit time) for the state variable varName

---

`smallIntestine` *smallIntestine put output from mealComposition(), combined with a time vec (col 1) through a one compartment ODE model (representing stomach and small intestine)*

Description

smallIntestine put output from mealComposition(), combined with a time vec (col 1) through a one compartment ODE model (representing stomach and small intestine)

Usage

```
smallIntestine(meals, time.to.reach.colon.h, showPlot = FALSE)
```

Arguments

`meals`	output from mealComposition()
`time.to.reach.colon.h`	time in hours taken for food to reach the entrance to the colon
`showPlot`	Default is FALSE

`verification` *verification*

Description

Prints a summary of the model results averaged over the time period start.av to fin.av

Usage

```
verification(m.out, start.av, fin.av)
```

Arguments

`m.out`	is the output of microPopGut(). It is a list containing the following objects: solution, params, myPars, etc
`start.av`	if the time to start averaging over
`fin.av`	if the time to finish averaging over

Index