# `microPop`: Modelling microbial populations and communities in R

Helen Kettle*[1], Grietje Holtrop[2], Petra Louis[3], and Harry J. Flint[3]

[1]*Biomathematics and Statistics Scotland (BioSS), Kings Buildings, Edinburgh, EH9 3FD, UK.*

[2]*Biomathematics and Statistics Scotland (BioSS), The Rowett Institute, University of Aberdeen, Foresterhill, Aberdeen, AB25 2ZD, Scotland, UK.*

[3]*The Rowett Institute, University of Aberdeen, Foresterhill, Aberdeen, AB25 2ZD, Scotland, UK.*

May 19, 2017

***Keywords***— microbiota, colon, population dynamics, microbes, bacteria, rumen, gut, phytoplankton, methanogens, resource competition, microbial diversity, phages, virus, bacteriophage

## Summary

1. Microbial communities perform highly dynamic and complex ecosystem functions that impact plants, animals and humans. Here we present an R-package, `microPop`, which is a dynamic model based on a functional representation of different microbiota.

2. `microPop` simulates the deterministic dynamics and interactions of microbial populations by solving a system of ordinary differential equations

*Corresponding author: Helen.Kettle@bioss.ac.uk

which are constructed automatically based on a description of the system to be simulated.

3. Data frames for a number of microbial functional groups and default functions for rates of microbial growth, resource uptake, metabolite production are provided but can be modified or replaced by the user.

4. `microPop` can simulate growth in a single compartment (e.g. bio-reactor) or ‘compartments’ in series (e.g. human colon) or in a simple 1-d application (e.g. phytoplankton in a water column). Furthermore, a microbial functional group may contain multiple strains in order to study adaptation and diversity or parameter uncertainty. Also simple interactions between viruses (bacteriophages) and bacteria can be included in `microPop`.

# 1 Introduction

Microbial communities play a crucial role in bio-geochemical cycling and perform ecosystem functions important to plants, animals and humans. Building predictive models that link microbial community composition to function is a key emerging challenge in microbial ecology (Widder et al., 2016). Here we present `microPop`, an R package which is a mechanistic model using ordinary differential equations (ODEs) to predict the dynamics and interactions of microbial functional groups. The general equations for rates of change of a microbial functional group (MFG), with quantity $X$, growing on a resource, with quantity $R$, at time, $t$, can be expressed as,

$$\frac{dX(t)}{dt} = v_X^{in}(t)X^{in}(t) + G(t)X(t) - v_X^{out}(t)X(t) \quad (1)$$

$$\frac{dR(t)}{dt} = v_R^{in}(t)R^{in}(t) - \frac{G(t)X(t)}{Y} - v_R^{out}(t)R(t) \quad (2)$$

where $v_i^{in}$ and $v_i^{out}$ are the inflow and outflow to the system (units of inverse time) for microbes ($i = X$) and resources ($i = R$), and $X^{in}(t)$ and $R^{in}(t)$ are the incoming quantities of microbes and resources respectively. $G(t)$ is the

45 specific growth rate of microbes on the resource (units of inverse time) and can
46 be expressed in a variety of ways (see Appendix B in supp. info.). The second
47 term on the right hand side of Eq. 2 is the uptake rate of the resource due to
48 microbial growth where $Y$ is the yield i.e. the quantity of microbial growth per
49 unit of resource taken up.

50 When there are multiple resources and several microbial groups with multiple
51 strains then Eqs. 1 and 2 expand into a large system with multiple metabolic
52 pathways. This is where `microPop` is a useful tool. Rather than coding these
53 equations, the user simply gives a description of the system (using 2 data frames,
54 '`resourceSysInfo`' and '`microbeSysInfo`') and a data frame for each MFG and
55 these equations are constructed and solved by **microPopModel** (ODE solvers
56 are provided by the deSolve package (Soetaert et al., 2010)).

57 Data frames for a number of MFGs found in the human large intestine (e.g.
58 Bacteroides, Acetogens, Methanogens, Butyrate Producers, Lactate Producers
59 and so on) as described by Kettle et al. (2015) and the rumen (by Munoz-
60 Tamayo et al. (2016)) are included in the package (Table 1). These two sets of
61 microbial groups are not fundamentally different but rather a different approach
62 has been used to subdivide the microbiota. If the user simply wishes to use these
63 MFGs then microPop can be used 'off the shelf', however, any number of other
64 MFGs may also be added by the user by defining a data frame in the correct
65 format.

66 Since many of the required parameter values for the MFGs are not well
67 known it should be noted that the parameter values stated in the included
68 MFG data frames will almost certainly change with increasing knowledge and
69 in some cases can be interpreted as simply a 'best guess'. One way of coping
70 with this parameter uncertainty is addressed in our previous work (Kettle et al.,
71 2015) (and included in `microPop`) where we assigned multiple strains to each
72 MFG with stochastically-generated parameter values. The strains will compete
73 with each other; some will flourish, some will die out, and by the time a steady
74 state is reached a viable microbial community for the given environment will

have been created. By changing the seed for the random number generator in **micropopModel**, multiple viable communities can be created and ensemble statistics can be used to define the solution. Moreover, if only one strain per group is specified the user has a choice to either run the model with the parameter values specified in the data frame or to randomly generate its parameters. This might be useful, for example, for generating model output to represent samples from a number of volunteers.

Since microbial growth, resource uptake and metabolite production may be modelled in a number of ways, the choices behind `microPop`'s default growth and uptake rate functions are explained fully in the Appendix. All of these functions are contained in a list called '`rateFuncs`' (Table 2) and may be redefined by the user (see Appendix A in supp. info.), allowing `microPop` to be applied to a large number of different microbial ecosystems. Although very complex systems with multiple microbial groups and strains may be slow to run in R, we hope that the transparency and flexibility of the code and its accessibility will enable researchers to simulate fairly complex systems without taking on a large computing project. Section 2 gives some examples of what `microPop`can do, section 3 gives a brief description of how to use `microPop`; these sections can be read in either order depending on your preference.

# 2 Example Applications

Here we give a flavour of how `microPop` can be used to simulate a wide range of microbial systems. For more information on these examples please refer to the vignette included with the package (**vignette('microPop')** in R). The scripts for all of these examples are included in the `microPop` package[1] and are intended to serve as a template for users when defining their own problems. They are also included in the supporting information file 'Scripts'. The name of the appropriate script is given in square brackets in each example heading

[1] The location of these files can be found by '`system.file("DemoFiles/ExampleFileName.R", package = "microPop"). It is also printed to screen when the script is run.`'

102 and they can be run in R (after **library(microPop)**) e.g. using **runMicroPopExample('human1')** (for the human1.R script (Section 2.1.1)). Most of the plots shown in this paper are automatically generated by microPop and can be tweaked using the '`plotOptions`' input list in **microPopModel**.

## 2.1 Modelling human gut microbiota

The microbial ecosystem in the human colon has been linked to numerous issues in human health. For example, its two important functions are harvesting extra energy from our food, thus warranting the name the "forgotten organ" (O'Hara and Shanahan, 2006), and aiding the development of our immune system (Chow et al., 2010). The following four examples are based on the model described by Kettle et al. (2015) which uses 10 different microbial groups to represent the microbial community in the human colon (Table 1). Here we use just three of these – Bacteroides, NoButyStarchDeg (starch degraders that do not produce butyrate) and Acetogens – to demonstrate some features of microPop. The information describing the inflows and outflows of each state variable for these scenarios is contained in the data frames '`resourceSysInfoHuman`' and '`microbeSysInfoHuman`' which are included with the package and are based on the system described by Kettle et al. (2015) and Walker et al. (2005). To look at these simply type '`resourceSysInfoHuman`' or '`microbeSysInfoHuman`' at the R prompt. Since these contain information on all 10 groups used in the full simulation by Kettle et al. (2015) the user can also use these when simulating the behaviour of any/all of the 10 groups.

### 2.1.1 Microbial growth in a constant environment [`human1.R`]

This is a simple example to show how **microPopModel** can be run using most of the default settings and intrinsic dataframes. In this scenario there is no limit on growth due to pH and the Bacteroides group dominate the system (Fig. 1).

128 
### 2.1.2 How does temporal pH change affect microbial growth? [`human2.R`]

129 In this scenario, pH changes from 5.5 to 6.5 halfway through the simulation. This
130 is implemented by altering **rateFuncs$pHFunc** and setting input argument
131 '`pHLimit=TRUE`'. Due to their preferred pH ranges (determined by '`pHcorners`'
132 in the data frames for each group) NoButyStarchDeg now dominate the first
133 half of the simulation, however when the pH rises to 6.5 Bacteroides regain
134 dominance (Fig. 2).

135 
### 2.1.3 How does spatial pH change affect microbial growth? [`human3.R`]

136 Here we approximate the pH change in sections of the human colon by defin-
137 ing the system as two compartments where the first one (at pH 5.5) flows into
138 the second (at pH 6.0). To simulate two compartments we add a loop to call
139 **microPopModel** twice. The first call simulates growth in the first compart-
140 ment over the whole of the simulation time. The results from this are then
141 used to provide the entry rates to the second compartment (using the function
142 **makeInputFromSoln**) in the second **microPopModel** call. The results (Fig.
143 3) show that NoButyStarchDeg dominate in first compartment (top row) and
144 Bacteroides begin dominating the second compartment but this changes due to
145 large inflow of NoButyStarchDeg from the previous compartment.

146 
### 2.1.4 How does microbial diversity affect response to pH? [`human4.R`]

147 Here we use the 'human2' example, where pH changes from 5.5 to 6.5 halfway
148 through the simulation, but include microbial diversity by assigning 5 strains
149 to each microbial group (via input argument, '`numStrains`'). We assume that
150 the strains within a microbial group have the same metabolic pathways i.e.
151 those specified in the group data frame, but diversity is incoporated by ran-
152 domly varying some of their growth parameters (based on Kettle et al. (2015)).
153 The extent of the variation, the parameters which are to be randomised and
154 whether trade-offs are required are all controlled via the '`strainOptions`' list.
155 Moreover, the user may also specify the parameter values for individual strains

using '`paramsSpecified`' and '`paramDataName`' also in this list (note, not all parameter values need to be specified - those that are specified will simply overwrite the randomly generated values). Fig. 4 a and b show the results for each strain.

Since there are multiple strains per group it is possible to examine how the mean group trait adapts over time using a biomass-weighted average at each time step:

$$\overline{x(t)} = \frac{\sum_i^n x_i m_i(t)}{\sum_i^n m_i(t)} \tag{3}$$

(Norberg et al., 2001) where $\overline{x(t)}$ is the average group trait at time $t$, $x_i$ is the trait value for strain $i$ and $m_i(t)$ is the mass of strain $i$ at time $t$. For example when pH changes, strains which prefer that new pH will flourish whilst others will be washed out. The centre of mass of the trapezoidal pH limitation can be computed using the function **pHcentreOfMass** and we define this one parameter as the pH trait. We can compute and plot the change in time of any of the stochastically-varying parameters/traits using the function **plotTraitChange** (e.g. Fig. 4 c shows the variation of the pH trait over time for each microbial group). For more details on phenotype adaptation please see Kettle et al. (2015) or Norberg et al. (2001).

## 2.2 Methane production from rumen microbiota [`rumen.R`]

Methane production from fermentation of feed by ruminant livestock contributes significantly to greenhouse gas production by agriculture (Cottle et al., 2011). Here we use `microPop` to model fermentation in the rumen, based on a simplified version of the model by Munoz-Tamayo et al. (2016), to provide a basic demonstration of how a mechanistic model may potentially aid in the design of diet strageties which reduce greenhouse emissions. The construction of this model is significantly different to the human colon model in Section 2.1 in several ways. Firstly, and most importantly, there are no substitutable resource; all resources are essential (see Appendix B.1 for an explanation of the different types of re-

183 source) and microbial growth is included explicitly in the group stoichiometries
184 (the groups involved are sugar-utilisers ($X_{su}$), amino-acid utilisers ($X_{aa}$) and
185 hydrogen utilisers ($X_{h2}$); included data frames '`Xh2`', '`Xsu`' and '`Xaa`'). Secondly,
186 hydrolysis is treated as a separate process such that polymer substrates must
187 be hydrolised to soluble sugars and amino acids before they are available for
188 microbial uptake. Thirdly, dead microbial cells are recycled into polymers.

189 For demonstration purposes we have simplifed the original model by Munoz-
190 Tamayo et al. (2016) as follows: we consider only constituents dissolved in the
191 rumen fluid (thereby removing gas transfer from the fluid fluid to the rumen head
192 space), we have removed carbon chemistry (we only consider dissolved inorganic
193 carbon) and we have removed the calculation of pH from acid-base reactions.
194 Also, we use units of mass rather than moles. Fig. 5 shows a schematic diagram
195 of the system and notation of state variables (figure caption) [2].

196 Since polymers are not used directly by any of the microbial groups (and are
197 therefore not mentioned in the MFG data frames) they will not be automati-
198 cally added as state variables by **microPopModel**. Thus to include hydrolysis
199 we add $Z_{nsc}$, $Z_{ndf}$ and $Z_{pro}$ to the microPop data frame for $X_{su}$. We then
200 explicitly state the parameters needed for hydrolysis and recycling of dead cells
201 into polymers as these are not included in the input files. Furthermore, **re-**
202 **movalRateFunc** is redefined to include the reduction rate for polymers and
203 the **entryRateFunc** includes the equivalent increase for soluble sugars ($S_{su}$)
204 and soluble amino acids ($S_{aa}$). Similarly the death of microbial cells is included
205 in **removalRateFunc** and the increase in polymers from the dead cells is in-
206 cluded in **entryRateFunc**.

207 Using the same settings as Munoz-Tamayo et al. (2016), we investigate how
208 increasing the initial concentrations of the feed polymers, $Z_{nsc}$, $Z_{ndf}$ and $Z_{pro}$,
209 affects the concentration of methane in the rumen ($S_{ch4}$). Thus we set the initial
210 polymer concentrations at 1 g/l and then increase each one in turn to 20 g/l
211 (Fig. 6). Increasing $Z_{ndf}$ and $Z_{pro}$ leads to increasing methane concentrations

[2] `microPop` code for the original (unsimplified model) is available on request for academic purposes.

212 as expected, however, the second row in Fig. 6 shows that, somewhat counter-
213 intuitively, the amount of methane produced decreases as initial concentrations
214 of $Z_{nsc}$ increases over a threshold between 15-20 g/l. $S_{IC}$ (soluble inorganic car-
215 bon) and $S_{h_2}$ (soluble hydrogen, not shown) both increase with $Z_{nsc}$, therefore
216 the cause of this appears to be the decrease in ammonia ($S_{nh3}$) (third column
217 in Fig. 6) which rapidly falls to zero for high initial values of $Z_{nsc}$. This is
218 because $Z_{nsc}$ is hyrolysed at a much faster rate (0.2 $h^{-1}$) than $Z_{ndf}$ (0.05 $h^{-1}$)
219 so increased $Z_{nsc}$ leads to increased $S_{su}$ and rapid growth of $X_{su}$ and hence
220 rapid uptake of $S_{nh3}$. The depletion of $S_{nh3}$ inhibits the growth of $X_{h2}$ and
221 thus mitigates methane production in this simple model example.

## 222 2.3 At what depth do phytoplankton grow best? [`phyto.R`]

223 Here we show how `microPop` can be used in a simple 1D application to in-
224 vestigate the depth at which phytoplankton blooms occur, given their growth
225 relies on nutrients welling up from below and sun light entering from above.
226 By simulating the competing growth of three different (theoretical) microbial
227 groups we show how the groups form a vertical assemblage based on their dif-
228 ferent requirements for light and nutrient (loosely based on work by Kettle and
229 Merchant (2008)). The light level at depth, $z$ (m), is given by,

$$\exp(-k_L z), \tag{4}$$

230 where $k_L$ is the light attenuation coefficient (we use $k_L$=0.5 $m^{-1}$). Nutrient
231 upwelling is modelled by assuming that the inflow of nutrient, $I_N$ (g $l^{-1}d^{-1}$),
232 increases with depth such that

$$I_N = v_N z \tag{5}$$

233 where $v_N$ is the gradient of inflow rate of nutrient with depth (g $l^{-1}d^{-1}m^{-1}$).
234 To define this system in microPop we consider nutrient to be the only re-

235 source since light is not depleted through microbial use and therefore does not
236 need to be included as a state variable. Nutrient upwelling is incorporated via
237 **entryRateFunc** and light limitation via **extraGrowthLimFunc** (the output
238 from this function is used to scale the maximum growth rate in a similar way to
239 **pHLimFunc**). There is no wash out rate for resources but we set a small wash
240 out rate for the phytoplankton of 0.005 $d^{-1}$ (see 'systemInfoMicrobesPhyto.csv')
241 to represent death rate.

242 We divide a depth of 20 m into 1 m layers and run `microPop` for each layer
243 for a simulation time of 3 months. The simulation begins with phytoplankton
244 spread evenly through the depth of the water column (e.g. this may occur
245 after vertical mixing caused by high winds). Thereafter there is no mixing
246 (calm conditions) and the phytoplankton are stationary in the water but grow
247 at different rates according to the light and nutrient levels at that particular
248 depth. In Case 1 (when running **runMicroPopExample('phyto')** the user
249 will be prompted to enter a case number) we simulate the growth of just one
250 phytoplankton group. Fig. 7a shows how the magnitude and depth of the bloom
251 changes.

252 In Case 2 we add in 2 more groups, all with the same starting concentration.
253 The 3 groups have different requirements for nutrient and light as determined
254 by their half saturation values for nutrient and light ($K_N$ and $K_L$ respectively).
255 Fig. 7b shows how over time the groups occupy different levels in the water
256 column.

## 257 2.4 Bacteriophages and resistance [`phages.R`]

258 Although not the main intended use of `microPop`, bacteriophages (viruses which
259 attack bacteria) can be included in `microPop` in a simplistic way. In this exam-
260 ple we consider 2 (theorectical) groups of bacteria (called Bacteria1 and Bac-
261 teria2) and 2 bacteriophages called Virus1 and Virus2. Both bacteria have the
262 same substrate (nutrient) and the same parameters with the only difference that
263 Bacteria2 has a higher maximum growth rate than Bacteria1. Virus1 attacks

264 Bacteria1, and Virus2 attacks Bacteria2. The two viruses have the same pa-
265 rameter values and differ only in their choice of host cell (bacterial group). We
266 consider a simple system with a constant dilution rate of 0.1 $d^{-1}$. All variables
267 have a starting value of 1 and the only inflow is nutrient.

268 In order to infect a host cell, the bacteriophage attaches itself to the bacterial
269 cell wall and then injects its genetic material into the host cell, causing the host
270 cell to eventually die and release a large number of new phage particles. To
271 model this within microPop we make some simplifying assumptions. Firstly,
272 since one phage attacks one bacterial cell, the 'consumption' rate does not follow
273 a Monod Equation but it is more like a predator-prey model where the rate of
274 change of the number of cells of the virus, $V$, due to viral attack on $B$ bacterial
275 cells is

$$\frac{dV}{dt} = \alpha V B \tag{6}$$

276 where $\alpha$ is the specific reproduction rate (number of new virus cells made from
277 one viral cell per bacterial cell per day). Within `microPop` we put the max-
278 GrowthRate of V1 on B1 equal to $\alpha$ and redefine **growthLimFunc** so the
279 'limitation' is now simply $B$ rather than the Monod equation (this is multiplied
280 by $V$ later in **derivsDefault**). The rate of change of the number of bacterial
281 cells due to death by virus attack is

$$\frac{dB}{dt} = -\frac{\alpha}{Y} V B \tag{7}$$

282 where $Y$ is the yield i.e. the number of new virus cells per bacterial cell (note
283 $\alpha = Yb$ where $b$ is the binding rate (units of $V^{-1}d^{-1}$)).

284 We now consider bacterial mutations by incorporating mutation of Bacteria1
285 to a resistant strain via **entryRateFunc** so that a fraction of the Bacteria1
286 population is converted to a resistant group ('resistantBacteria1', $B_1^R$) per day
287 ($f_B$). Thus the rate of change of $B_1^R$ due to mutation is

$$\frac{dB_1^R}{dt} = f_B B_1 \tag{8}$$

288 and the loss rate from $B_1$ is the negative of this (also included via **entryRate-**
289 **Func**). This model is a very simplified version of that of Cairns et al. (2009)
290 where we have removed the time delay and the infected stage.

291 We run microPop for 4 different system scenarios (when running **runMi-**
292 **croPopExample('phages')** the user will be prompted to choose from case 1
293 to 4); the results are shown in Fig. 8. To begin with we look at the system with-
294 out viruses and see the two bacteria competing for nutrient, since Bacteria2 has
295 the highest growth rate it dominates the system (case 1; Fig. 8a). We now add
296 in Virus2 which attacks Bacteria2 allowing Bacteria1 to dominate the system
297 causing Bacteria2, and hence Virus2, to die out (case 2; Fig. 8b). If we now
298 add in Virus1, so that we have both bacterial groups and both viral groups, we
299 see more complex dynamics emerge (case 3; Fig. 8c). In the fourth case we add
300 in random mutations from Bacteria1 to resistantBacteria1 which is resistant to
301 Virus1 and therefore survives at the expense of the other bacterial groups (case
302 4; Fig. 8d).

## 303 3 Running `microPop`

304 As previously mentioned the main function in the package is **microPopModel**
305 and this is used to run a simulation. The input arguments to this function are
306 used to completely define the system and its output is a list containing two
307 elements - one is the solution to the ODEs i.e. a matrix of the values of the
308 state variables over time ('`solution`') and the other is a list containing all of
309 the information used to produce the solution ('`parms`'). In the simplest case,
310 the user need only specify 4 of the input arguments to **microPopModel** (the
311 others have defaults) these 4 are:

- 312 '`microbeNames`' - a vector of the names of the microbial groups in your
313 system, e.g. c('Bacteroides','Methanogens'). Note that a data frame with
314 the same name must be available for each group specified.
- 315 '`times`' - a vector defining the time sequence at which output is required,

316 e.g. seq(0,10,0.1).

317 • '`resourceSysInfo`' - this is a data frame or the name of a csv file describ-
318 ing the inflow, outflow, start values and molar masses of the substrates and
319 products associated with the microbial groups specified in microbeNames.
320 See **help(resourceSysInfo)** for details.

321 • '`microbeSysInfo`' - this is a data frame or the name of a csv file describing
322 the inflow, outflow and start values of the microbial groups specified in
323 microbeNames. See **help(microbeSysInfo)** for details.

324 Fig. 9 shows this in detail using the example given in **help(microPopModel)**.
325 Details of all the input arguments can be found via the function help and in the
326 vignette included with the package.

# 327 Supporting Information

328 • **Appendix 1** Equations and information on '`rateFuncs`'.

329 • **Appendix 2** R Scripts for the Example Applications in section 2.

# 330 Acknowledgements

331 We thank the Scottish Goverment's Rural and Environment Science and Ana-
332 lytical Services Division (RESAS) for funding this research. Also many thanks
333 to Rafael Munoz-Tamayo for sharing his matlab code for the rumen model.

# 334 References

335 Cairns, B.J.., Timms, A.R., Jansen, V.A.A.., Connerton, I.F.., and Payne,
336 R.J.H. (2009). Quantitative models of *in vitro* bacteriophage-host dynam-
337 ics and their application to phage therapy. *PLoS Pathogens*, 5:1–10.

338 Chow, J., Lee, S.M., Shen, Y., Khosravi, A., and Mazmanian, S.K. (2010).
339 Host-bacterial symbiosis in health and disease. *Advances in Immunology*,
340 107:243–274.

341 Cottle, D.J., Nolan, J.V., and Weidemann, S.G. (2011). Ruminant enteric
342 methane mitigation: a review. *Animal Production Science*, 51:491–514.

343 Kettle, H., Louis, P., Holtrop, G., Duncan, S.H., and Flint, H.J. (2015). Mod-
344 elling the emergent dynamics and major metabolites of the human colonic
345 microbiota. *Environmental Microbiology*, 17:16151630.

346 Kettle, H. and Merchant, C.J. (2008). Modeling ocean primary production: Sen-
347 sitivity to spectral resolution of attenuation and absorption of light. *Progress*
348 *in Oceanography*, 78:135–146.

349 Munoz-Tamayo, R., Giger-Reverdin, S., and Sauvant, D. (2016). Mechanistic
350 modelling of in vitro fermentation and methane production by rumen micro-
351 biota. *Animal Feed*, 220:1–21.

352 Norberg, J., Swaney, D.P., Dushoff, J., Lin, J., Casagrandi, R., and Levin,
353 S.A. (2001). Phenotypic diversity and ecosystem functioning in changing
354 environments: a theoretical framework. *Proceedings of the National Academy*
355 *of Sciences*, 98:11376–11381.

356 O'Hara, A.M. and Shanahan, F. (2006). The gut flora as a forgotten organ.
357 *EMBO Reports*, 7:688–693.

358 Soetaert, K., Petzoldt, T., and Woodrow Setzer, R. (2010). Solving differential
359 equations in r: package desolve. *Journal of Statistical Software*, 33:1–25.

360 Walker, A.W., Duncan, S.H., McWilliam Leitch, E.C., Child, M.W., and Flint,
361 H.J. (2005). pH and peptide can radically alter bacterial populations and
362 short-chain fatty acid ratios within microbial communities from the human
363 colon. *Applied Environmental Microbiology*, 71:3692–3700.

364 Widder, S., Allen, R.J., Pfeiffer, T., Curtis, T.P., Wiuf, C., Sloan, W.T.,
365 Cordero, O.X., Brown, S.P., Momeni, B., Shou, W., Kettle, H., Flint, H.J,
366 Haas, A., Laroche, B., Kreft, J.-U., Rainey, P., Freilich, S., Schuster, S.,
367 Milferstedt, K., van der Meer, J., Grosskopf, T., Huisman, J., Free, A., Pi-
368 cioreanu, C., Quince, C., Klapper, I., Labarthe, S., Smets, B., Wang, H.,
369 and Soyer, O. (2016). Challenges in microbial ecology: building predictive
370 understanding of community function and dynamics. *The ISME Journal*,
371 10:2557–2568.

Table 1: Microbial functional groups (MFGs) included in microPop. The first ten groups are described by Kettle et al. (2015), the last three are described by Munoz-Tamayo et al. (2016). It should be noted that these two sets of microbial groups are not fundamentally different but rather a different approach has been used to subdivide the microbiota. To see these data frames simply type in the group name at the R prompt. Users should be aware that the parameter values given in these data frames will almost certainly change with increasing knowledge of gut microbiota and in some cases are simply a 'best guess'. The csv files for these groups can be found using '`system.file('extdata/groupname.csv',package='microPop')`' where 'groupname' is any one of the entries in the first column below.

| MFG (Kettle et al., 2015) | Description | Examples |
|---|---|---|
| Bacteroides | Acetate-propionate-succinate group | Bacteroides spp. |
| NoButyStarchDeg | Non-butyrate-forming starch degraders | Ruminococcaceae related to Ruminococcus bromii. Might also include certain Lachnospiraceae. |
| NoButyFibreDeg | Non-butyrate-forming fibre degraders | Ruminococcaceae related to Ruminococcus albus, Ruminococcus flavefaciens. Might also include certain Lachnospiraceae. |
| LactateProducers | Lactate producers | Actinobacteria, especially Bifidobacterium spp, Collinsella aerofaciens |
| ButyrateProducers1 | Butyrate Producers | Lachnospiraceae related to Eubacterium rectale, Roseburia spp. |
| ButyrateProducers2 | Butyrate Producers | Certain Ruminococcaceae, in particular Faecalibacterium prausnitzii |
| PropionateProducers | Propionate producers | Veillonellaceae e.g. Veillonella spp., Megasphaera elsdenii |
| ButyrateProducers3 | Butyrate Producers | Lachnospiraceae related to Eubacterium hallii, Anaerostipes spp. |
| Acetogens | Acetate Producers | Certain Lachnospiraceae, e.g. Blautia hydrogenotrophica |
| Methanogens | Methanogenic archaea | Methanobrevibacter smithii |
| MFG (Munoz-Tamayo et al., 2016) | Description | Examples |
| Xsu | Sugar utilizers | |
| Xaa | Amino acid utilizers | |
| Xh2 | Hydrogen utilizers | Methanobrevibacter smithii |

Table 2: Top section of table: Functions contained in the list rateFuncsDefault (further details on these functions are included in the Appendix (supp. info.)). Bottom section of table: other functions in microPop. To get help on the inputs and outputs of these functions use **help(functionName)** in R using the function names below.

| Function name | Description |
|---|---|
| **entryRateFunc** | Rate of entry of each state variable to system at time $t$ |
| **removalRateFunc** | Rate of exit of each state variable from system at time $t$ |
| **pHFunc** | pH value at time $t$ |
| **pHLimFunc** | pH limit on growth (varies between 0 and 1 for a given pH value) |
| **extraGrowthLimFunc** | Another limit on growth (default value is 1 i.e. no limit). This is included to allow the user to add in any kind of growth limitation as its output is used to scale the maxGrowthRate value) |
| **growthLimFunc** | This scales the maximum growth value (value between 0 and 1) |
| **combineGrowthLimFunc** | Combining growth on multiple resources |
| **uptakeFunc** | Uptake of resource due to microbial growth |
| **productionFunc** | Production of metabolites resulting from microbial growth |
| **combinePathsFunc** | Combining the results of growth on multiple metabolic pathways |
| **createDF** | Creates a data frame from a .csv file |
| **derivsDefault** | Describes the ODEs; called by **ode** |
| **getGroupName** | Returns the name of the group from the strain name |
| **makeInflowFromSoln** | Returns the exit rate of each state variable (matrix[time,variable]) |
| **microPopModel** | Simulates growth of microbial populations (main function) |
| **pHcentreOfMass** | Finds the mean pH weighted by the pH limitation |
| **plotTraitChange** | Plots the average group trait over time (when there are multiple strains per group) |
| **runMicroPopExample** | Used to run the scripts for the examples described in Section 2 |

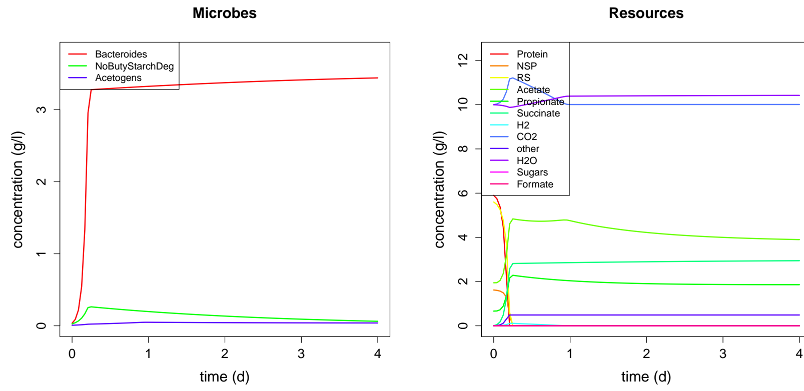


Figure 1: Human Colon Application (human1)

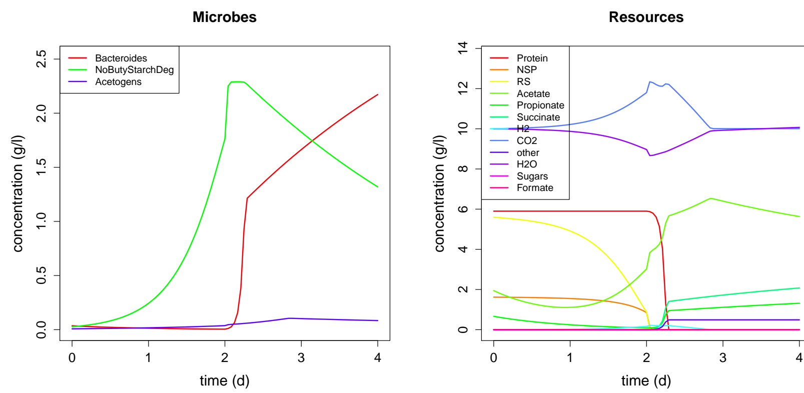


Figure 2: Human Colon Application (human2) - pH changes from 5.5 to 6.5 halfway through the simulation.

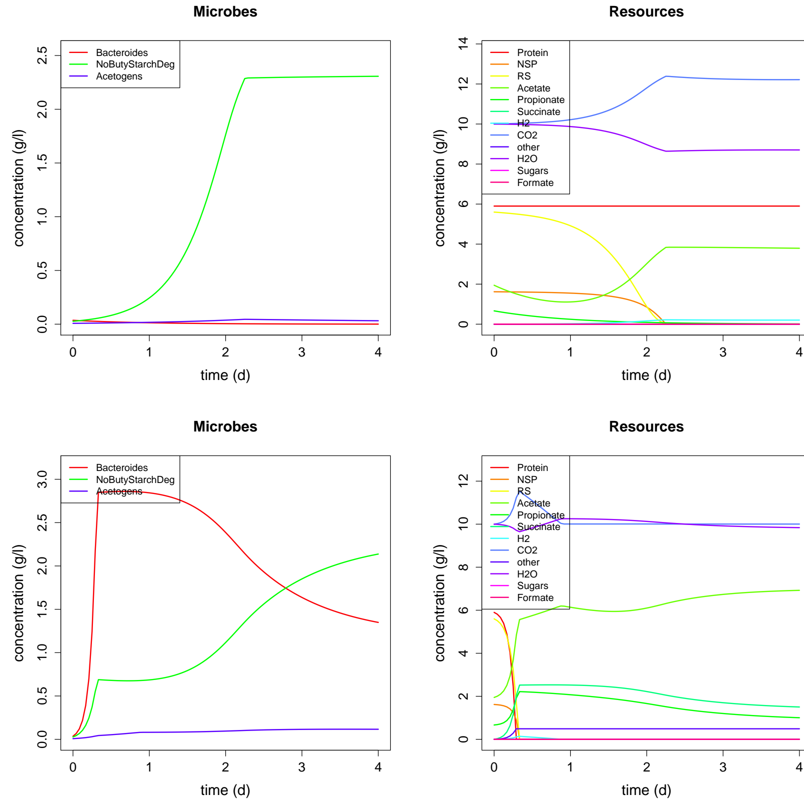


Figure 3: Human Colon Application (human3) - two compartments with different pH. Top row: first compartment (pH 5.5), bottom row: second compartment (pH 6.0).

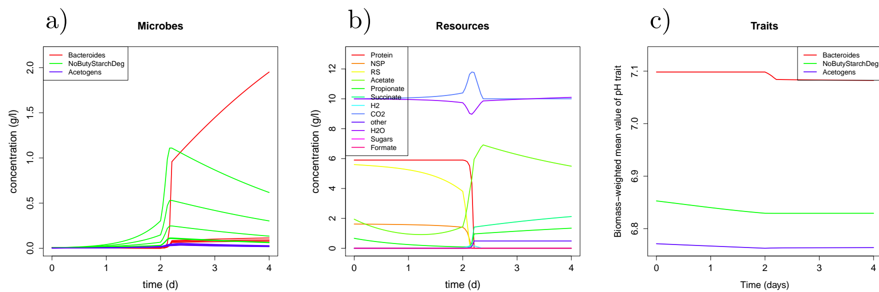


Figure 4: Human Colon Application (human4) - five strains per group. Note, to plot the sum of the strains in each group set '`plotOptions$sumOverStrains = TRUE`'). c) shows the mean value of the pH trait for each of the three groups computed using Eq 3.

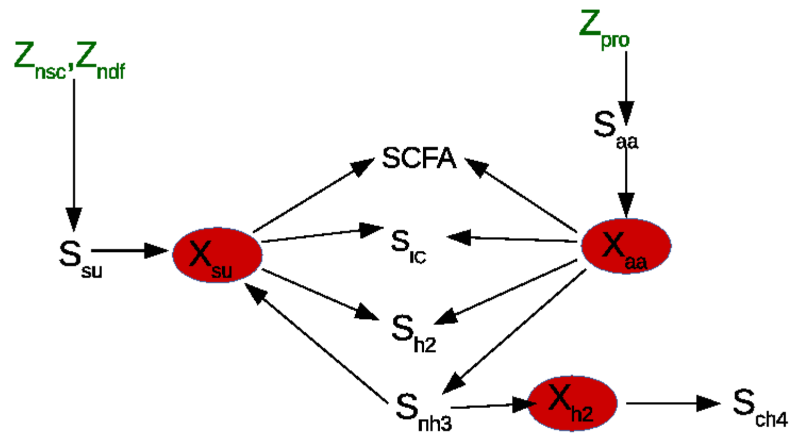


Figure 5: The rumen system based on the model by Munoz-Tamayo et al. (2016) consists of polymers (non-structural carbohydrates ($Z_{nsc}$), cell wall carbohydrates ($Z_{ndf}$) and proteins ($Z_{pro}$) which are hydrolised to the soluble components: sugars ($S_{su}$) and amino acids ($S_{aa}$). The microbial groups are sugar-utilisers ($X_{su}$), amino-acid utilisers ($X_{aa}$) and hydrogen utilisers ($X_{h2}$). They convert their respective substrates to short chain fatty acids, SCFA, (acetate, butyrate and propionate), hydrogen ($S_{h2}$), ammonia ($S_{nh3}$), inorganic carbon ($S_{IC}$) and methane ($S_{ch4}$). Dead microbial cells are recycled to the polymer compartments (arrows not shown).

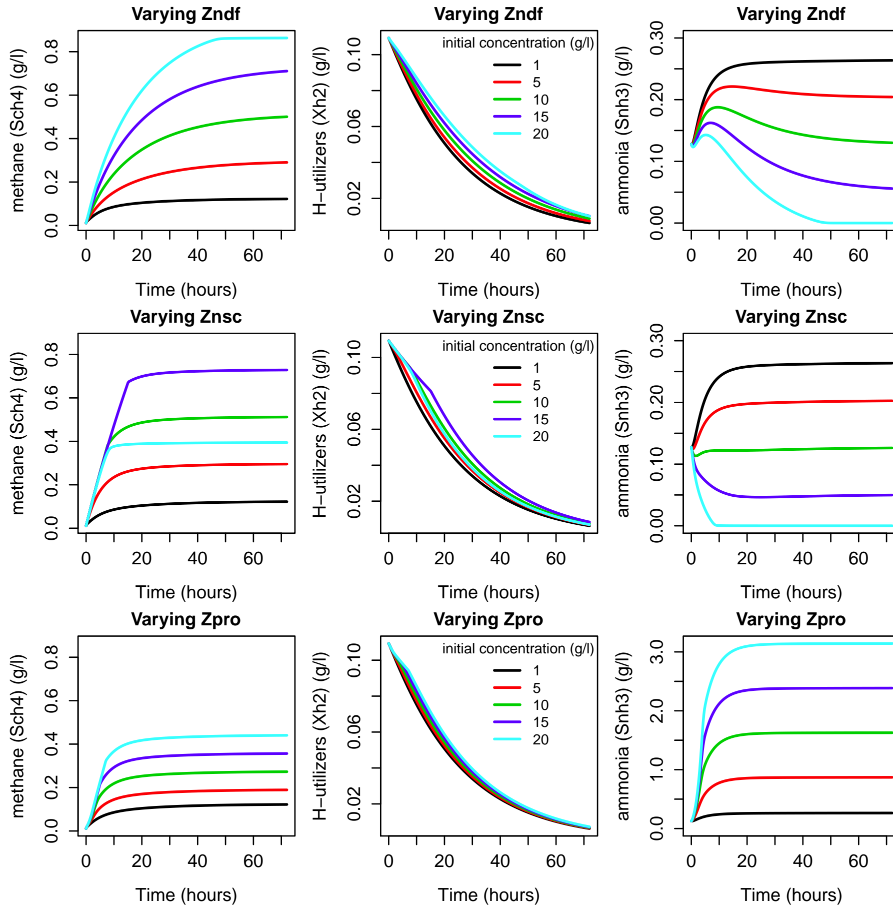


Figure 6: Methane concentration in the rumen for initial concentrations between 1 and 20 g/l (legend in centre column) of the feed polymers $Z_{nsc}$, $Z_{ndf}$ and $Z_{pro}$ (while one polymer concentration is changed the other two are held at 1 g/l). Note change in scale for $S_{nh3}$ for $Z_{pro}$.

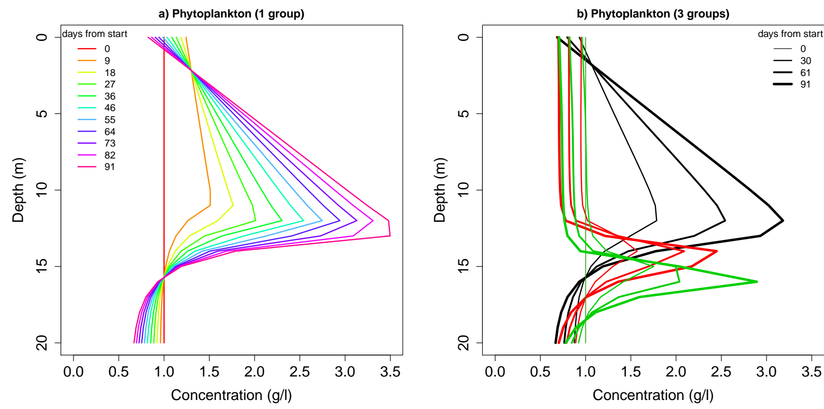


Figure 7: a) Concentration of Phyto1 at 9 day intervals when it is the only group present ($K_N$=1e-6 g $l^{-1}$, $K_L$=0.8 light units). b) Concentration of all three groups at monthly intervals with Phyto1 in black ($K_N$=1e-6 g $l^{-1}$, $K_L$=0.8 light units), Phyto2 in red ($K_N$=1e-4 g $l^{-1}$, $K_L$=0.4 light units) and Phyto3 in green ($K_N$=1e-2 g $l^{-1}$, $K_L$=0.2 light units).

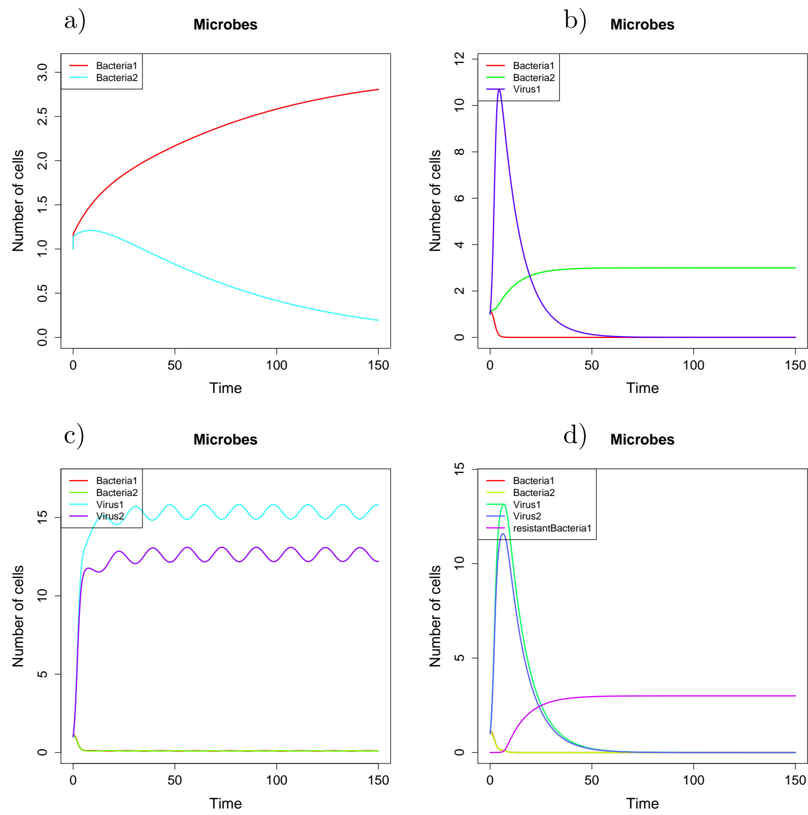


Figure 8: a) Case 1: Two bacterial groups compete for one substrate (Nutrient); no viruses present. b) Case 2: A virus (Virus1) which attacks Bacteria1 is added to the system. c) Case 3: As in b) but a virus (Virus2) which attacks Bacteria2 is also added to the system. d) Case 4: As in c) but Bacteria1 randomly mutates into a group (at a rate $f_B B_1(t)$ where $f_B = 0.001$ d $^{-1}$) which is identical to Bacteria1 apart from it has resistance to Virus1.

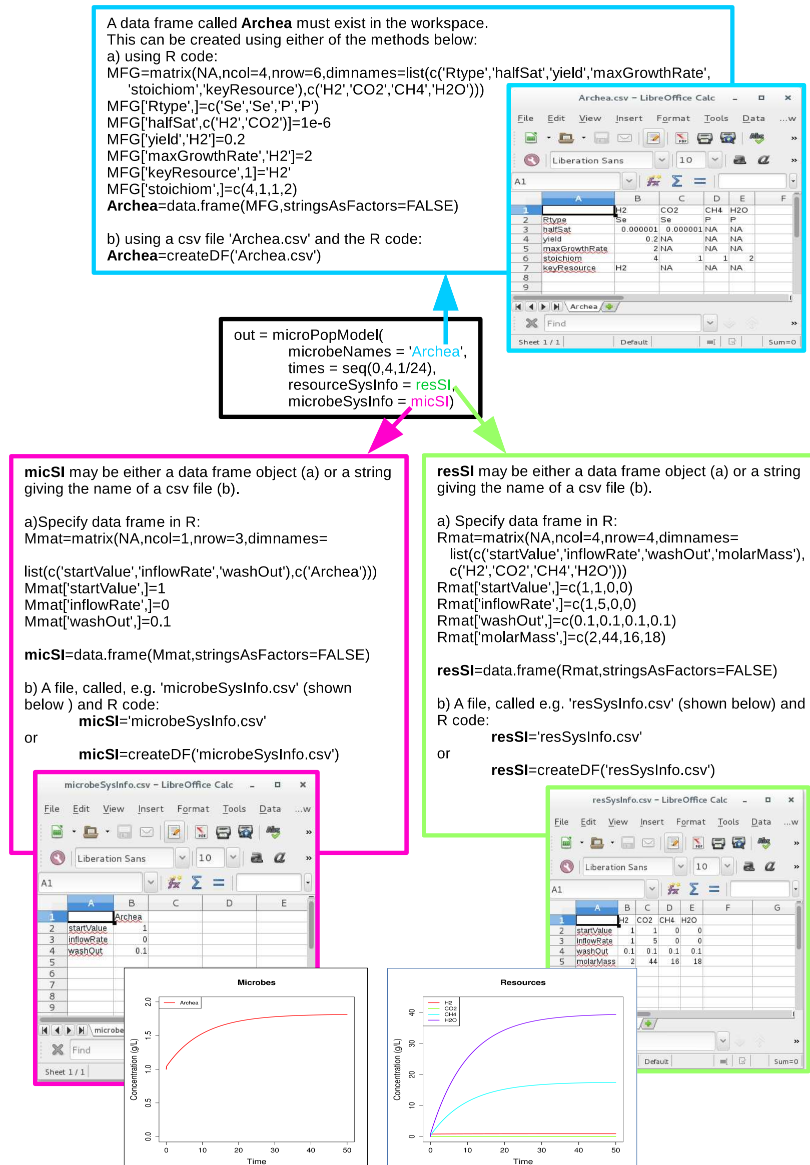


Figure 9: Schematic showing how to call **microPopModel** (black box) for the example shown in **help(microPopModel)**. See section 3 for more details. The csv (comma separated values) files are shown in Libre office but can be created in Microsoft Excel or any text editor. The automatically generated plots are shown at the bottom. Note that here the group data frame (‘`Archea`’) is defined but this is not necessary if the user wishes to use any of the group data frames already included in the package (Table 1). Also note that this example uses the default rate functions. For information on how to change these please look at the Example Applications in Section 2 and the code in Scripts (supp. info.).