VECTRI v1.4.6

Adrian M. Tompkins

Trieste - Italy

May 11, 2017

Contents

1 Background

1.1 Software requirements

The malaria model presented is a grid-point distributed dynamical model and is referred to as the VECToR borne disease community model of ICTP, TRIeste. (VECTRI).

Getting started:

Key requirements to get/run the code

gfortran
netcdf libraries
git

Useful optional code to postprocess and examine files

cdo
R
ncview
ncdump

Installing requisite software

Ubuntu:

```
$ sudo apt-get install libnetcdf-dev
$ sudo apt-get install libnetcdff-dev
$ sudo apt-get install netcdf-bin
$ sudo apt-get install gfortran
$ sudo apt-get install git
$ sudo apt-get install cdo
$ sudo apt-get install ncview
```

MAC: On a MAC, one can use MACPORTS to update software. After you have insstalled macports, you can then do:

e.g. `$ sudo port install gfortran`

and so on...

1.2 To get VECTRI

If you want to use VECTRI in the future, please sign up online and download instructions will be sent to you:

`http://users.ictp.it/~tompkins/vectri/download`

You will get an automated email reply telling you how to download a copy from the gitlab repository. ICTP keeps a register of users who may be occasionally (once or twice per year) contacted with VECTRI news unless you specify *not* to be included.

However, you can simply clone the code directly using the following command. This will retrieve the code and put it in a *subdirectory* of your present location called "vectri". Make sure you run the following command from a suitable location (e.g. `$ cd $HOME` first to put the code under your home directory).

```
$ git clone https://gitlab.com/tompkins/vectri.git
```

In the present version each location (grid-point) is independent, but the structure of the model will allow communication between gridpoints such as vector flight or human population migration to be easily incorporated.

The model broadly follows the approach typical of dynamical models in that it explicitly resolves the growth stages of the egg-larvae-pupa cycle in addition to the gonotrophic and the sporogenic cycles.

2 Running the model

VECTRI code location

Since v1.2.4, the runtime directories are kept separate from all the source and datafiles to keep things cleaner. The model needs to know where these files are. To do this set up an environmental variable VECTRI with the location of the model.

ksh or bash: `$ export VECTRI=directory`

csh: `$ setenv VECTRI directory`

If you have just run the clone command you can set up the variable with the following command:

```
$ export VECTRI=`pwd`/vectri
```

Note the forward sloping single quotes. These execute the `pwd` command and place the results in the quoted location.

TIP: If you want to run VECTRI regularly, you can place this definition command in your .cshrc .kshrc or .bashrc (according to your shell choice) to set this variable automatically on login, however, you will need to make sure you set the whole absolute path! e.g. if you have placed the directory under your HOME:

```
$ export VECTRI=$HOME/vectri.
```

You should not use the `` `pwd` `` as that depends on the directory location from where the command is run.

You can check the variable is set correctly by:

```
$ ls $VECTRI
```

which should give a listing of the code directories:

```
data doc graphics README.txt scripts source
```

running location

where do you want to run? Just make a run directory and go there! e.g.

```
$ mkdir -p $HOME/run
$ cd $HOME/run
```

TIP: In bash hit "tab" to complete path and filenames

TIP: Remember you can use the short-cuts ~or $HOME to refer to your home directory.

Running the model

The model can be run using two different methods.

Method 1 - fixed form input

```
$ $VECTRI/scripts/vectri_driver mode options
```

following by a list of numbers that define the options

Method 2 - free form input

```
$ $VECTRI/scripts/vectri_driver --option1 value --option2 value
```

Method 1 is shorter (less typing) but you need to put the options in the correct order. Method 2 is more flexible but you need to know which combination of options to specify. The driver code looks at the first argument, and if it an integer between 1 to 5, the fixed form method 1 is assumed. Otherwise free-form options are assumed. **You can not mix the two options**.

2.1 Fixed form input

The first number, *mode* after the vectri_driver command, tells the program which mode to run vectri in. At the moment there are three main modes, with two special modes:

mode=

1. constant value temperature and rainfall
2. station data in a netcdf file (defunct - now use mode 3)
3. netcdf file input - can import gridded satellite rainfall and ERA temperatures
4. Special mode to read ASCII station data (fixed format for West Africa)
5. Special mode to read ASCII station data (fixed format for example file)

mode=1: constant input

The model is run using the commands

```
$ $VECTRI/scripts/vectri_driver 1 temperature rainfall population
```

where *temperature* is given in degree C
rainfall in mm day^{-1}
population is the density as people km^{-2}

mode=3: gridded or single location runs, netcdf input format

The model is run using the commands

```
$ $VECTRI/scripts/vectri_driver 3 lat1 nlat dlat lon1 nlat dlat date1
date2 preprocess rainfile tempfile experiment downscale initfile population
run
```

This mode can be used to either run for a region or for a point with netcdf input, or it can also be used to import precipitation observations from CMORPH, FEWS or TRMM (if the datasets are available locally). *Instructions for this mode to follow soon.*

mode=5: single location, ASCII Tmax/Tmin format

```
$ $VECTRI/scripts/vectri_driver 5 stationfile population nday
```

stationfile: the name of the ASCII input file (can include directory structure)
population: the density as people km^{-2}
nday: length of integration in days, set to 0 for the whole file.

The ASCII input file is assumed to have a column format of form: *yyyy mm dd* T_{min} T_{max} *rainfall*. VECTRI assumes that the mean temperature is simply the average of the maximum and minimum values.

There is an example file for mode 5 in the data directory. You can view it using

```
$ less $VECTRI/data/highland_location_africa.txt
```

VECTRI automatically looks for the file in the `data` directory. If your data file is stored elsewhere you need to remember to include the *path* in the filename (e.g. `/scratch/myname/mydatadir/myinputc`)

2.2 Free form input

Note, long word options are with a **double** hyphen!

-m mode OR --mode mode

domain opts: --lat1 x --nlat x --dlat x --lon1 x --nlon x --dlon x

date opts: --date1 yyyymmdd --date2 yyyymmdd

process opts: --preproc 1/0 --dscale 0/1 - switches to preprocess/interpolate input files and downscale temperature

climate opts: --temp ncdf_filename or "erai"

climate opts: --prec ncdf_filename or "obsfew", "obstrmm", "obscmorph"

population opts: --pop density or "afripop" "grump"

misc opts: --init none or restart_file --station station_filename

3 Output

vectri.nc

The main output file is called *vectri.nc* which is found in the subdirectory *output* after each successful run. The file is in standard netcdf format and you can use two commands to examine its content `$ ncview` and `$ ncdump`.

Using ncdump

ncdump outputs the entire contents of a file, perhaps a little more than you bargained for, so use the useful option `-h` to examined the file *header*:

```
$ ncdump -h ./output/vectri.nc
```

The command lets us know the dimensions of the output, gives a list of the model output variables, as well as their *metadata*. Lastly, but certainly not least, the dump provides a list of all the model run *global attributes*. Here you find stored all of the run parameters. Many of these can be set by the user, so it is very useful to have a record *inside the file*!

We could dump a individual variable using the `$ ncdump -v variable` command, but it is also possible to graphically interrogate the file using ncview.

Using ncview

Using the command

```
$ ncview ./output/vectri.nc
```

launches the graphical viewer. There is a panel containing a list of variables. Clicking on one of these variables launches a display window (Fig. 1). If you conducted a gridded regional integration then you will get a map of the output, otherwise you will get a line graph for a station integration

The input variables of population, temperature and rainfall are stored in the file.

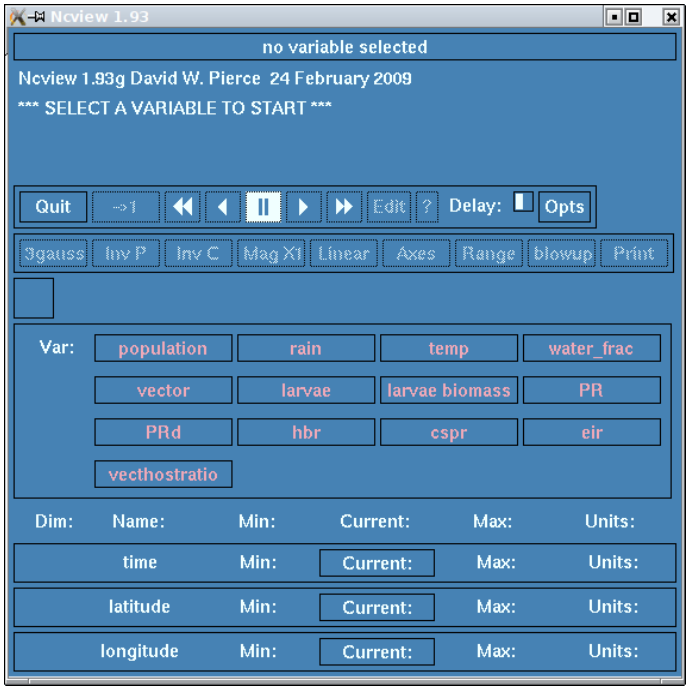

Figure 1: An example ncview window showing vectri output

Table 1: The Vectri output

name	definition	units
population	population density	m^{-2}
rain	rainfall	mm day^{-1}
temp	temperature	deg C
water_frac	fraction of grid box covered by pond breeding sites	
vector	mosquito density	m^{-2}
larvae	larvae density	m^{-2}
larvae_biomass	larvae biomass	mg m^{-2}
PRd	proportion of population with detectable malaria (day 10+)	
hbr	human bite rate	number $person^{-1}$ day^{-1}
cspr	Circum sporozoite protein ratio (fraction eir/hbr)	fraction
eir	entomological inoculation rate (number of infectious bites)	number $person^{-1}$ day^{-1}
cases	number of new cases[1]	fraction
immunity	proportion of immune population	fraction

4 Setting options in VECTRI

It is possible to set options in VECTRI. This is done in the file *vectri.options* which you should save in the sub-directory **input**.

vectri.options
You may open the file vectri.options using the text editor of your choice (e.g. gedit, vi, vim, emacs, xemacs):

```
$ gedit ./output/vectri.options.
```

The file is empty by default, but you can enter a list of parameter values to control vectri, which must be separated by a comma. You may put each parameter on a new line for clarity if you wish:

```
parameter1=value,
parameter2=value,
```

The parameters you may specify are given in the follow tables, which associate parameters according to themes:

Table 2: run control parameters

parameter	definition	default	units
nyearspinup	length of spinup period discarded from output	0	years

Table 3: scheme choice parameters

parameter	definition	default	units
nlarv_scheme	larvae water temperature scheme (1,2,3) 1=ermert:2011a,2=jepsen47,3=bayoh lindsey 2003	1	
nsurvival_scheme	mosquito survival temperature scheme (1,2,3) 1=MartinsI,2=MartinsII,3=Bayoh	2	

Table 4: larvae parameters (version 1.2.6)

parameter	definition	default	units
neggmn	eggs laid per female vector	120	
nlayingmax	maximum number of vectors allow to lay (-ve no limit)	-99	
rlarv_tmax	maximum temperature for larvae survival	35	deg C
rlarv_eggtime	time for egg hatching	1.0	days
rlarv_pupaetime	time for pupae stage	1.0	days
rlarv_flushmin	minimal daily survival L1 larvae after intense rainfall	0.4	
rlarv_flushtau	exponential decay of flushing with rainrate	50	mm day^{-1}
rmasslarv_stage4	mass of L4 larvae	0.45	mg
biocapacity	larvae biomass carry capacity of pools	300	mg m^{-2}
rlarvsurv	base survival rate due to preditation	0.825	

Table 5: larvae parameter (version 1.3.5 onwards)

parameter	definition	default	units
neggmn	eggs laid per female vector	80	
nlayingmax	maximum number of vectors allow to lay (-ve no limit)	-99	
rlarv_tmin	minimum temperature for larvae survival	12.16	deg C
rlarv_tmax	maximum temperature for larvae survival	38	deg C
rlarv_eggtime	time for egg hatching	1.0	days
rlarv_pupaetime	time for pupae stage	1.0	days
rlarv_flushmin	minimal daily survival L1 larvae after intense rainfall	0.4	
rlarv_flushtau	exponential decay of flushing with rainrate	20	mm day^{-1}
rmasslarv_stage4	mass of L4 larvae	0.45	mg
biocapacity	larvae biomass carry capacity of pools	300	mg m^{-2}
rlarvsurv	base survival rate due to preditation and non-climatic factors	0.987	

Table 6: hydrology (version 1.2.6, a new beta scheme is available in 1.3.0)

parameter	paper	definition	default	units
ipud_vers		puddle parametrization version (126/130)	126	
rwaterfrac_perm	w_0	permanent breeding site fraction	10^{-6}	
rwaterfrac_max	w_{max}	maximum coverage by breeding sites	0.2	
rwaterfrac_rate	K_w	pond geometry factor	10^{-3}	mm^{-1}
rwaterfrac_evap126	$E + I$	infiltration and evaporation rate loss of ponds	250	mm day^{-1}
rwater_tempoffset	T_{wat}	temperature offset of pond to air	0	deg C

Table 7: interventions (not yet operative)

parameter	definition	default	units
rnobednetuse	proportional of host NOT using bednet	1	
rbednettreat (IN-OPERATIVE)	proportion of bednets insecticide treated	0.0	

Table 8: biting parameters

parameter	definition	default	units
rzoophilic_tau	e-folding population density for zoophilicity rate	30×10^{-6}	m^{-2}
rzoophilic_min	minimum anthropophilic bite rate at low population densities	0.1	

Table 9: immunity and transmission parameters

parameter	definition	default	units
rhostclear	clearance rate for non-immune adults	50	days
rpthost2vect_I	probability of infected host transmission to vector	0.2	
rpthost2vect_R	probability of host with immunity transmission to vector (blocking)	0.04	
rptvect2host	probability of infected transmission from infected vector to host	0.3	
rimmune_gain_eira	Annual EIR required for 95% of population to gain immunity.	100 [0=no immunity]	[bites $year^{-1}$]
rbitehighrisk	ratio of risk of biting between infectious groups and susceptibles (crude behaviour)	5	
rimmune_loss_tau	e-folding timescale for immunity loss	365	days

Table 10: gonotrophic and sporogenic

parameter	definition	default	units
rtgono	threshold temperature for egg development in vector	7.7	deg C
dgono	degree days for egg development in vector	37.1	days
rtsporo	threshold temperature for parasite development	16.0	deg C
dsporo	degree days for for parasite development	111	days

Table 11: population

parameter	definition	default	units
rpop_growth	population growth rate	1.0	ratio of population increase per year
rpop_death_rate	fraction of population renewed each year (births=deaths)	0.02	
rpopdensity_min	minimum population density considered	10^{-6}	m^{-2}
rmigration	simple migration fraction imported cases	0.0	$year^{-1}$

Table 12: toy climate

parameter	definition	default	units
rtemperature_offset	temperature offset	0.0	deg C
rtemperature_trend	temperature trend	0.0	deg C yr^{-1}
rrainfall_factor	ratio to scale rainfall by	1.0	