An Implementation of the Rothermel Fire Spread Model in the R Programming Language

Published version:

DOI:10.1007/s10694-014-0405-6

Springer

Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For fax submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/ corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
  Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: http://dx.doi.org/[DOI].
If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: http://www.link.springer.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us if you would like to have these documents returned.

Metadata of the article that will be visualized in OnlineFirst

Please note: Images will appear in color online but will be printed in black and white.

ArticleTitle	An Implementation of the Rothermel Fire Spread Model in the R Programming Language	
Article Sub-Title		
Article CopyRight	Springer Science+Business Media New York (This will be the copyright line in the final PDF)	
Journal Name	Fire Technology	

Corresponding Author	Family Name	**Vacchiano**
	Particle	
	Given Name	**Giorgio**
	Suffix	
	Division	DISAFA
	Organization	Universita degli Studi di Torino
	Address	Via Da Vinci 44, 10095, Grugliasco, TO, Italy
	Email	giorgio.vacchiano@unito.it
Author	Family Name	**Ascoli**
	Particle	
	Given Name	**Davide**
	Suffix	
	Division	DISAFA
	Organization	Universita degli Studi di Torino
	Address	Via Da Vinci 44, 10095, Grugliasco, TO, Italy
	Email	d.ascoli@unito.it

	Received	11 February 2014
Schedule	Revised	
	Accepted	4 April 2014

Abstract	This note describes an implementation of the Rothermel fire spread model in the R programming language. The main function provided, ros(), computes the forward rate of spread at the head of a surface fire according to Rothermel fire behavior model. Additional functions are described to illustrate the potential use and expansions of the package. The function rosunc() carries out uncertainty analysis of fire behavior, that has the ability of generating information-rich, probabilistic predictions, and can be coupled to spatially-explicit fire growth models using an ensemble forecasting technique. The function bestFM() estimates the fit of Standard Fuel Models to observed fire rate of spread, based on absolute bias and root mean square error. Advantages of the R implementation of Rothermel model include: open-source coding, cross-platform availability, high computational efficiency, and linking to other R packages to perform complex analyses on Rothermel fire predictions.

Keywords (separated by '-')	Fire behaviour - Fuel models - Fire spread - Prescribed fire - Wildfire
Footnote Information	

Journal: 10694
Article: 405

Author Query Form

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author

During the process of typesetting your article, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query	Details required	Author's response
1.	Please check and confirm the inserted citation of Figures 1, 2, and 3 are correct. If not correct please suggest an alternate citation.	
2.	Please update Ref. 39. If possible.	

Fire Technology

DOI: 10.1007/s10694-014-0405-6

An Implementation of the Rothermel Fire Spread Model in the R Programming Language

Giorgio Vacchiano and Davide Ascoli, DISAFA, Universita degli Studi di Torino, Via Da Vinci 44, 10095 Grugliasco, TO, Italy*
e-mail: d.ascoli@unito.it

Received: 11 February 2014/**Accepted:** 4 April 2014

Abstract. This note describes an implementation of the `Rothermel` fire spread model in the R programming language. The main function provided, ros(), computes the forward rate of spread at the head of a surface fire according to Rothermel fire behavior model. Additional functions are described to illustrate the potential use and expansions of the package. The function `rosunc()` carries out uncertainty analysis of fire behavior, that has the ability of generating information-rich, probabilistic predictions, and can be coupled to spatially-explicit fire growth models using an ensemble forecasting technique. The function `bestFM()` estimates the fit of Standard Fuel Models to observed fire rate of spread, based on absolute bias and root mean square error. Advantages of the R implementation of Rothermel model include: open-source coding, cross-platform availability, high computational efficiency, and linking to other R packages to perform complex analyses on Rothermel fire predictions.

Keywords: Fire behaviour, Fuel models, Fire spread, Prescribed fire, Wildfire

1. Introduction

Mathematical models of wildland fire behaviour have been of great importance in both fire ecology research and fire management (e.g., [6, 26, 27, 38]). Rothermel model for forward fire rate of spread (hereafter ROS) in surface fuels is one of the most widely used fire models [29].

Rothermel model has been programmed into computer code-based versions [2], and included as a fundamental part of several fire modeling software. Examples of simulators operating at the stand scale are Behave/BehavePlus [4, 5], and the Fire and Fuel Extension to the Forest Vegetation Simulator [28], both programmed in Fortran. Furthermore, Rothermel model has been included in spatially-explicit fire

* Correspondence should be addressed to: Giorgio Vacchiano, E-mail: giorgio.vacchiano@unito.it

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405 ☐ LE ☑ CP ☐ TYPESET ☑ DISK

simulators (e.g., [1, 17, 19, 24, 25]), or as extension to proprietary (e.g. [18]) or open-source Geographical Information Systems (e.g., the r.ros module for GRASS GIS [42]).

However, these packages often operate as a black-box, i.e., are opaque to customization of input parameters (except for those allowed by the Graphical User Interface), model form, and cross-format analysis of model output. We identified a need for scientists and managers to run surface fire simulations based on Rothermel model within a larger, seamless workflow of pre- and post- wildfire modeling analyses, such as input data preparation, iterative model runs, or plotting and statistically manipulating model results (e.g., [7, 10, 16]).

The aim of this Research Note is to present the Rothermel package for the R programming language (R Core Team, 2013). The package currently resides on the CRAN repository (URL: cran.r-project.org/web/packages/rothermel). R is an open-source programming language and statistical analysis framework that is rapidly becoming standard in scientific research. It allows data handling (Appendix 1), statistical analysis, and graphical representations, thanks to a suite of pre-installed statistical methods, and more than 4,000 add-on packages. It functions under all operating systems, including Windows, Linux and OSX. To date, some fire-related packages have been developed for R (e.g., `paleofire` [21], `fume` [34], and `fwi.fbp` [41]), but the Rothermel fire spread model has not been ported yet.

2. The `ros()` Function

2.1. Description

The `ros()` function computes ROS [m min^{-1}] and other output variables from Rothermel model (Table 1). Rothermel model has been subject to several corrections. The model implemented here includes the following changes to the orginal system of equations: an updated weighting factor for reaction intensity by fuel category [20], updated equations for mineral content, damping coefficient, reaction velocity, weighting factor for fuel loadings, and live fuel moisture of extinction [2], and removing the maximum wind factor limit [7].

Inputs required by the fire spread model are specified by the fire behavior fuel model (hereafter: fuel model). Other inputs are related to environmental variables such as slope steepness, midflame wind speed, and the moisture content of each fuel category and size class (Table 1). Rothermel model is static, therefore it assumes constant weather variables for each simulation [29].

The inputs and outputs of `ros()` are in metric units, but the function converts all inputs to imperial units in order to apply the original coefficients of Rothermel model. The function accepts both single values, and `data.frames` with multiple observations. If `modeltype` is set to D, a dynamic

Journal : **10694_FIRE**	Dispatch : **16-4-2014**	Pages : **14**
MS NO : **405**	☐ LE ☑ CP	☐ TYPESET ☑ DISK

Table 1
Input and Output Variables for the `ros()` Function

Input	Units	Description
`modeltype`	–	S(tatic), D(ynamic)
`w`	t ha^{-1}	A vector or data frame of fuel load for fuel classes 1-h, 10-h, 100-h, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent)
`s`	$m^2\ m^{-3}$	A vector or data frame of surface-to-volume ratio for fuel classes 1-h, 10-h, 100-h, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent)
`delta`	cm	A value or vector of fuel bed depth
`mx.dead`	%	A value or vector of dead fuel moisture of extinction
`h`	kJ kg^{-1}	A vector or data frame of heat content for fuel classes 1-h, 10-h, 100-h, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent)
`m`	%	A vector or data frame of percent moisture on a dry weight basis for fuel classes 1-h, 10-h, 100-h, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent)
`u`	km h^{-1}	A value or vector of midflame windspeed
`slope`	%	A value or vector of site slope
Output	Units	Description
m.live	%	Characteristic dead fuel moisture
m.dead	%	Characteristic live fuel moisture
mx.live	%	Live fuel moisture of extinction
cSAV	$m^2\ m^{-3}$	Characteristic (weighted) SA/V
rho	kg m^{-3}	Fuel bulk density
beta	–	Packing ratio
rpr	–	Relative packing ratio
IR dead	kW m^{-2}	Dead fuel reaction intensity
IR live	kW m^{-2}	Live fuel reaction intensity
IR	kW m^{-2}	Reaction intensity
fw	0–100	Wind correction factor
fs	0–1	Slope correction factor
Heat source	kW m^{-2}	Numerator of Rothermel model
Heat sink	kJ m^{-3}	Denominator of Rothermel model
ROS	m min^{-1}	Rate of spread

Journal : 10694_FIRE
MS NO : 405
Dispatch : 16-4-2014
Pages : 14
☐ LE
☑ CP
☐ TYPESET
☑ DISK

fuel model will be invoked, where part of the cured herbaceous fuel is transferred to the 1-h fuel size class, as a function of herb fuel moisture [35]. If characteristic fuel moisture is higher than the fuel moisture of extinction, both for live and dead fuels, the respective reaction intensity is set to zero [5]. The following two examples demonstrate the usage of `ros()`.

2.2. Example 1

This example computes Rothermel equations by using a single fuel model, moisture scenario, and unique slope and wind values.

```
> library(Rothermel)
> modeltype <- "D"
> w <-c (2, 1, 0.5, 3, 8)
> s <- c (5600, 358, 98, 6200, 8000)
> delta <- 50
> mx.dead <- 30
> h <- c (18622, 18622, 18622, 19500, 20000)
> m <- c (7, 8, 9, 40, 60)
> u <- 5
> slope <- 10
> ros (modeltype, w, s, delta, mx.dead, h, m, u, slope)
```

The result is a `list` of the following values:
[1] Characteristic dead fuel moisture [%] 7.02
[2] Characteristic live fuel moisture [%] 59.37
[3] Live fuel moisture of extinction [%] 128.40
[4] Characteristic SA/V [m^2 m^{-3}] 7325.13
[5] Bulk density [kg m^{-3}] 2.90
[6] Packing ratio [dimensionless] 0.01
[7] Relative packing ratio [dimensionless] 0.93
[8] Dead fuel Reaction intensity [kW m^{-2}] 553.34
[9] Live fuel Reaction intensity [kW m^{-2}] 933.21
[10] Reaction intensity [kW m^{-2}] 1486.55
[11] Wind factor [0-100] 6.75
[12] Slope factor [0-1] 0.25
[13] Heat source [kW m^{-2}] 501.85
[14] Heat sink [kJ m^{-3}] 4682.05
[15] ROS [m min^{-1}] 6.43

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405
☐ LE ☑ CP ☐ TYPESET ☑ DISK

The result is a `list` of the following values:

[1] Characteristic dead fuel moisture [%] 7.02
[2] Characteristic live fuel moisture [%] 59.37
[3] Live fuel moisture of extinction [%] 128.40
[4] Characteristic SA/V [$m^2\ m^{-3}$] 7325.13
[5] Bulk density [kg m^{-3}] 2.90
[6] Packing ratio [dimensionless] 0.01
[7] Relative packing ratio [dimensionless] 0.93
[8] Dead fuel Reaction intensity [kW m^{-2}] 553.34
[9] Live fuel Reaction intensity [kW m^{-2}] 933.21
[10] Reaction intensity [kW m^{-2}] 1486.55
[11] Wind factor [0–100] 6.75
[12] Slope factor [0–1] 0.25
[13] Heat source [kW m^{-2}] 501.85
[14] Heat sink [kJ m^{-3}] 4682.05
[15] ROS [m min^{-1}] 6.43

2.3. *Example 2*

Here we illustrate how to compute ROS using data from fire field experiments, and validate Rothermel predictions against observed rate of spread. This example uses the dataset `firexp` of the `Rothermel` R package. The dataset includes ROS measured using a microplot scale approach [36] during field fire experiments in heathland fuels (mixed grass-shrub). The experiments were carried out on flat terrain under variable fire weather [8, 39]. For each observed ROS, environmental and fuel parameters were measured before and during the fire. Some ranges in the dataset are: ROS 0.9–26.3 m min^{-1}; wind speed 0.4–7.9 km h^{-1}; 1-h fuel moisture 10–27%. We predict ROS using data from three Standard Fuel Models ([35]) and environmental variables measured in the field, and validate it against observed values.

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405 ☐ LE ☑ CP ☐ TYPESET ☑ DISK

```
> library (Rothermel); data (firexp); data (SFM_metric)
> # Observed variables
> m <- firexp [, 18:22]
> u <- firexp [, "u"]
> slope <- firexp [, "slope"]
> obs <- firexp[,"ros"]
> # Predict ROS using Standard Fuel Models GR5, GS3 and SH7
> a = list ( )
> models = which (rownames (SFM_metric) == "GR5" |
      rownames (SFM_metric) == "GS3" |
      rownames (SFM_metric) == "SH7")

> for (i in 1 : length (models) ) {
     modeltype <- SFM_metric [models [i], 1]
     w <- SFM_metric [models [i], 2:6]
     s <- SFM_metric [models [i], 7:11]
     delta <- SFM_metric [models [i], "Fuel_Bed_Depth"]
     mx.dead <- SFM_metric [models [i], "Mx_dead"]
     h <- SFM_metric [models [i], 14:18]
     a [i] <- ros (modeltype, w, s, delta, mx.dead, h,
        m, u, slope)[15]}

> # Plot
> plot (obs, a [[1]], xlab = "Observed rate of spread (m/min)",
         ylab = "Predicted rate of spread (m/min)",  col = "red",
         pch =19, xlim = c (0, 30), cex.lab = 1.1)
> points (obs, a [[2]], pch = 19, col = "green2")
> points (obs, a [[3]], pch = 19, col = "blue2")
> abline (coef = c(0, 1))
> abline (coef = c(0, 0.7),lty = 2); text (13.6, 19.2, "-30%")
> abline (coef = c(0, 1.3),lty = 2); text (28.7, 19.2, "+30%")
> legend (0, 19.2, c("GR5", "GS3", "SH7"), pch = 19,
        col = c("red", "green2", "blue2"), title = "Fuel model")

> # Inset Residual plot
> par (fig = c (.57, .98, .07, .55), new = T)
> plot (obs, a[[1]] - obs, xlab= "", ylab= "", col = "red",
       main= "Residuals", font.main = 1, pch=19, cex=.7)
> points (obs, a [[2]] - obs, pch = 19, cex =.7, col = "green2")
> points (obs, a [[3]] - obs, pch = 19, cex =.7, col = "blue2")
> abline (h = 0)
> par (fig = c (0, 1, 0, 1))
```


Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
LE TYPESET
MS NO : 405 CP DISK

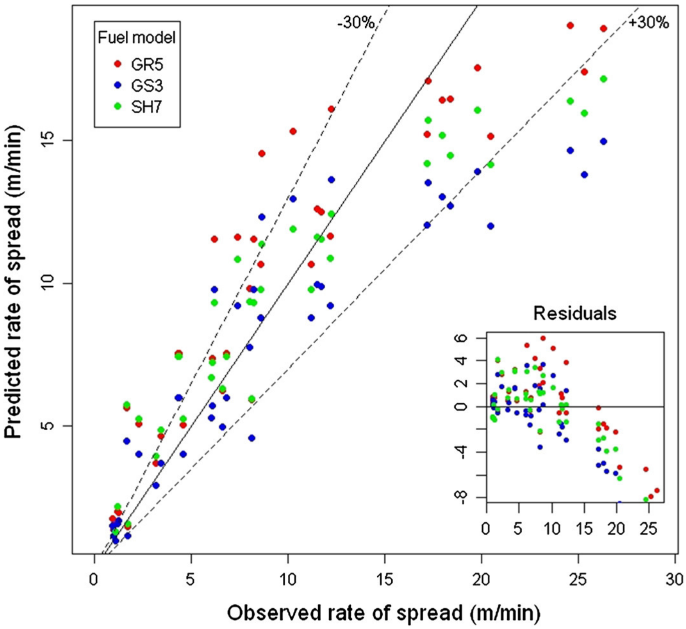

Figure 1. Observed vs. Predicted ROS for the `firexp` dataset using Standard Fuel Models GR5, GS3 and SH7.

3. Potential Expansion of the Package: Example of Functions

The `ros()` function can be implemented in more complex analyses of fire behavior and effects. We illustrate below two cases of the potential development of new functions based on `ros()`. The first case is a function for uncertainty analysis of rate of spread, that implements methods already explored by the literature [9, 14, 23, 37]. The second example is a newly developed function to evaluate the fit of preset fire behavior fuel models to observed ROS (Figure 1).

3.1. The `rosunc()` *Function*

Several authors have stressed the importance of introducing stochasticity in fire behavior prediction [9, 14, 23, 37]. The advantage of stochastic fire models is to obtain error bounds and probability-based outcomes for the main fire behavior parameters. Although Rothermel model is essentially deterministic, a probability density function of ROS or other model outputs can be obtained by perturbing one or more input variables (usually environmental ones). The probability associated to each output value is represented by the relative frequency of such output among all model realizations. Manually perturbing model inputs is a tedious task. The `rosunc()` function of the `Rothermel` package automatically perturbs inputs by randomly sampling from gaussian distributions, where the mean is the observed value and the standard deviation is specified by the user (in the form of coefficient of variation, 0–1). The output is a vector of ROS. The function accepts the same arguments as in `ros()`, plus the desired coefficients of variations for wind speed, fuel moisture, slope, fuel load, and fuel bed depth, and the number of simulations desired to produce a Monte-Carlo based probability density function for ROS [14, 23]. Consequently, the function runs on one fuel set at a time (i.e., no `data.frames` allowed as input).

3.2. Example 3

Here, one observation (row) is selected from the `firexp` dataset. Input values are selected similarly to `ros()`, and a coefficient of variation of 0.3 is specified to generate a gaussian distribution of fuel moisture values. The probability distribution function of ROS is generated by 1000 Monte Carlo simulations and graphically compared with the observed value. This example's output may differ from actual results due to the stochastic simulation of moisture values.

```
> data ("firexp"); varnames <- names (firexp)
> firexp <- as.numeric (firexp [5, ]); names (firexp) <- varnames

> pred <- rosunc (modeltype = "D",
        w = firexp [1:5],
        s = firexp [6:10],
        delta = firexp ["Fuel_Bed_Depth"],
        mx.dead = firexp ["Mx_dead"],
        h = firexp [13:17],
        m = firexp [18:22],
        u = firexp ["u"],
        slope = firexp ["slope"],
        sdm = 0.3,
        nsim = 1000)

> summary (pred)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6.11   11.06   12.19   13.34   14.56   28.98
```

3.3. The `bestFM()` *Function*

A set of Standard Fuel Models (SFM) was developed to parameterize fuel properties of different fuel complexes [3, 35]. In the process of testing the predictions of Rothermel model vs. observed ROS in a given vegetation, one of the first steps is to verify whether any of the SFM yields a satisfactory prediction [22, 30, 35]. This is a crucial step before undertaking the calibration of a custom fuel model [11].

The function `bestFM()` estimates the fit of the 53 SFM to a vector of observed ROS, based on absolute bias (predicted - observed ROS), and root mean square error (RMSE). Arguments of the function include environmental variables, which are not a part of SFM, and the observed value or vector of ROS. The function calls a dataset of SFM that has been embedded in the `Rothermel` package (dataset `SFM_metric`), simulates ROS using SFM data and environ-

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405 ☐ LE ☑ CP ☐ TYPESET ☑ DISK

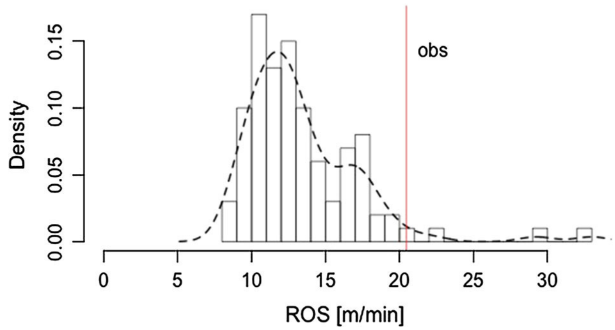

Figure 2. Probability density function of ROS and the observed value.

mental variables, and outputs a `data.frame` of RMSE and/or absolute bias. Simulations can also be run under predefined fuel moisture scenarios [35] by calling the dataset `scenarios` (Figure 2).

3.4. Example 4

This example loads a vector of observed ROS and environmental parameters from the `firexp` dataset, and compares them with `ros()` predictions from a dataset of 53 Standard Fuel Models. A sorted barplot of increasing RMSE is produced to illustrate the output of the function. The sign of prediction bias is indicated by the bar color (Figure 3).

```
> data ("firexp")
> a <- bestFM (obs = firexp [,"ros"],
               m = firexp [, 18:22],
               u = firexp [, "u"],
               slope = firexp [, "slope"])
```


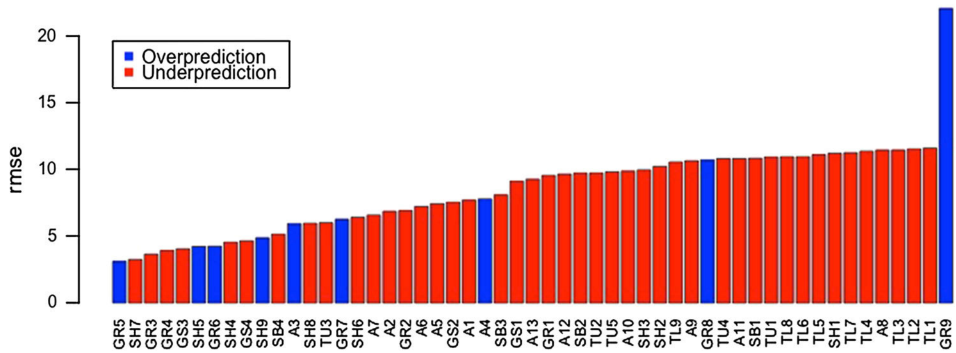

Figure 3. RMSE of 53 SFM against a dataset of observed ROS in heathland mixed grass-shrub fuels.

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405 ☐ LE ☐ TYPESET
☑ CP ☑ DISK

4. Discussion and Practical Implications

The main function of the `Rothermel` package computes ROS from user-defined (or standard) fuel and environmental parameters. The `ros()` function computes parameters of the Rothermel model with its most common modifications [2, 7, 20]; however, the code is open to host additional formulations, such as those by the Fuel Characteristic Classification System (FCCS) [33], or alternative fire spread models [15].

The `ros()` function is functionally similar to the US Forest Service software BehavePlus [5], and ROS predictions for aligned head fires are equivalent between the two softwares. Compared to BehavePlus, R provides an open-source platform that runs on multiple operating systems (Windows, OSX, Linux). However, `ros()` lacks the additional functionality of the latter, i.e., supplementary fire behavior and spread models, together with the user friendly interface that made BehavePlus so popular among fire managers. The `ros()` function is not intended as a decision support system for fire management alternative to BehavePlus. Rather, it is a new tool for fire scientists who need to carry out complex analyses using the Rothermel model. To this regard, its objective is similar to the Firelib C function library [10], that was written to give fire simulation modellers a common programming interface to use in building fire growth applications models.

However, compared to existing software, the R implementation of Rothermel model allows to perform many simulations at the same time (Example 2), plot and export the results, and nest the computation of ROS (and of all intermediate outputs of Rothermel model) within more complex analyses, such as `if()` statements or `for()` loops, or sensitivity analysis of model output [32]. Additionally, the R framework can generate web-based user interfaces (package `shiny` [31]), and complex plots such as fire characteristic charts [11].

Much potential is associated to the newly programmed function `rosunc()` that carries out uncertainty analysis of ROS. This method has recently been praised for its ability to generate more information-rich, probabilistic predictions, as compared to traditional deterministic models [23]. Furthermore, by dynamically linking to spatially-explicit fire growth models and forest dynamics simulators at the stand or landscape scale [13], the `rosunc()` function enables modellers to generate probabilistic predictions of fire growth and ensemble forecasts resulting from variable weather or fuel inputs [19].

Finally, the function `bestFM()` is intended as an exploratory analysis of observed ROS in a fuel complex. RMSE from Standard Fuel Models can show which group of models (i.e., GR, GS, SH, TU, TL, SB) have a similar fit to the data. In Example 4, observed ROS in mixed grass-shrub heath fuels from `firexp` showed increasing RMSE starting from GR, SH, GS up to TL models, excluding GR9. Within the first 10 best fuel models, the GR group performed slightly better than SH and GS. Our interpretation is that the herbaceous component in heath fuels is driving the rate of successive ignitions. Consequently, when building a custom fuel model [12] for dry heaths, particular attention should be focused on setting the parameters of the herbaceous fuel category.

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405 ☐ LE ☑ CP ☐ TYPESET ☑ DISK

The Rothermel package is one of the first tools to support fire science in the R programming language. A wealth of packages exists for other research fields in ecology and environmental science, such as climate modelling, biodiversity, natural hazard modelling, or genetics. Similarly, R has the potential to become a privileged platform to carry out data analysis and modelling in fire science. In fact, the R architecture is much suitable to develop tools such as decision support systems and cross-scale hierarchical models, i.e., systems of interacting simulators that take advantage of different modelling approaches (e.g., spatially-explicit fire spread, coupled physical fire models, stochastic weather generation, treatment of remotely sensed imagery...), and may effectively interact with local or remote data repositories.

We believe that the present package nicely fits in what a recent overview of the most up-to-date fire simulator pointed out [5]: 'Care must be taken to avoid black box modelling and to avoid use of default values. (...) A rebuild of the code from the bottom up [is desired] to facilitate integration of fire behaviour, fire effects and fire danger rating systems, as well as point and spatial systems'. Additional contributions to the package are welcome, and will implement complementary functions to enrich the range of fire modeling tools able to exploit the potential of the Rothermel model within the R statistical framework.

Acknowledgments

We would like to thank the CRAN staff for useful support and testing of the package.

Appendix 1: A Primer on the R Language

A complete introduction to the R language goes beyond the scope of this paper. We will briefly illustrate the meaning of some key terms in order for the reader to understand the examples and data structures referenced in this paper. For an introduction to the R language, tutorials and working examples, refer e.g. to 'An introduction to R' [40], from which this section is borrowed, and to the documentation available on the CRAN website (URL: http://cran.r-project.org).

The user operates R via commands entered at the prompt '>'. Elementary commands consist of either expressions or assignments. Expressions are evaluated, printed (unless specifically made invisible), and the value is lost. An assignment evaluates an expression and passes the value to an object stored in a 'workspace' for future retrieval. The assignment operator is '<–'. R commands are case sensitive; comments can be put almost anywhere, starting with a hashmark ('#').

R operates on named data structures. The simplest such structure is the vector, which is a one-dimensional entity consisting of an ordered collection of numeric or string elements. To set up a vector named `x`, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command `x < - c(10.4, 5.6, 3.1, 6.4, 21.7)`. An R data frame is a two-dimensional entity consisting of rows (i.e., observational units) and columns (i.e., observed variables). Vectors of the same length, for example `x` and `y`, can be concatenated to form columns in a data frame named `df` using the R command `df <- cbind(x, y)`. An R list is an object consisting of an ordered collection of other objects, be them vectors, data frames, or other R data structures. List elements are numbered and may be referred to by the subsetting operator `[[ ]]`.

Finally, functions are R objects that evaluate the result of an expression using user-defined arguments. A call to the function usually takes the form `function.name (argument1, argument2)`. The Rothermel package for R operates mainly by some newly programmed functions.

References

1. Ager AA, Vaillant NM, Finney MA (2011) Integrating fire behavior models and geospatial analysis for wildland fire risk assessment and fuel management planning. J Combust 2011(1):1–19
2. Albini FA (1976) Computer-based models of wildland fire behavior: a user's manual. Tech. rep, USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden UT
3. Anderson HE (1982) Aids to determining fuel models for estimating fire behavior. Tech. Rep. Gen. Tech. Rep. INT-122, USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden UT
4. Andrews PL (1986) BEHAVE fire behavior prediction and fuel modeling system BURN subsystem Part 1—Google Search. Tech. Rep. GTR-INT-194, USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden UT
5. Andrews PL (2013) Current status and future needs of the BehavePlus Fire Modeling System. Int J Wildl Fire. doi:10.1071/WF12167
6. Andrews PL, Queen LP (2001) Fire modeling and information system technology. Int J Wildl Fire 10(4):343–352
7. Andrews PL, Cruz MG, Rothermel RC (2013) Examination of the wind speed limit function in the Rothermel surface fire spread model. Int J Wildl Fire 22(7):959–969
8. Ascoli D, Bovio G (2013) Prescribed burning in italy: issues, advances and challenges. iForest 6:79–89
9. Bachmann A, Allgöwer B (2002) Uncertainty propagation in wildland fire behaviour modelling. Int J Geogr Inf Sci 16(2):115–127
10. Bevins CD (1996) Firelib user manual and technical reference. Tech. rep, Systems for Environmental Management, Missoula MT
11. Burgan RE (1987) Concepts and interpreted examples in advanced fuel modeling. USDA Forest Service, Intermountain Research Station, Ogden UT
12. Burgan RE, Rothermel RC (1984) BEHAVE: fire behavior prediction and fuel modeling system - FUEL subsystem. Tech. Rep. PMS 439–1, USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden UT
13. Crookston N (2012) rFVS: Running FVS from R. URL: http://code.google.com/p/open-fvs/wiki/rFVS
14. Cruz MG (2010) Monte Carlo-based ensemble method for prediction of grassland fire spread. Int J Wildl Fire 19(4):521–530
15. Cruz MG, Alexander ME (2013) Uncertainty associated with model predictions of surface and crown fire rates of spread. Environ Model Softw 47:16–28

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405 ☐ LE ☑ CP ☐ TYPESET ☑ DISK

16. Cruz MG, Fernandes PM (2008) Development of fuel models for fire behaviour prediction in maritime pine (Pinus pinaster Ait.) stands. Int J Wildl Fire 17(2):194–204
17. Ferragut L, Asensio ML, Monedero S, Ramrez J (2008a) Scientific advances in fire modelling and its integration in a forest fire decision system. In: De las Heras J, Brebbia CA, Viegas D, Leone V (eds) First international conference on modelling, monitoring and management of forest fires (FIVA 2008), WIT Press, pp 31–38
18. Ferragut L, Monedero S, Asensio M, Ramrez J (2008) Scientific advances in fire modelling and its integration in a forest fire decision system. WIT Trans Ecol Environ 119:31–38
19. Finney MA, Grenfell IC, McHugh CW, Seli RC, Trethewey D, Stratton RD, Brittain S (2011) A method for ensemble wildland fire simulation. Environ Model Assess 16(2):153–167
20. Frandsen WH (1973) Using the effective heating number as a weighting factor in Rothermel's fire spread model. Tech. rep, USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden UT
21. Global Paleofire Working Group (2013) Paleofire: an R package to analyse sedimentary charcoal records from the Global Charcoal Database to reconstruct past biomass burning. URL: http://cran.r-project.org/web/packages/paleofire/index.html
22. Grabner KW, Dwyer JP, Cutter BE (2001) Fuel model selection for behave in midwestern oak savannas. North J Appl For 18(3):74–80
23. Jimenez E, Hussaini MY, Goodrick S (2008) Quantifying parametric uncertainty in the Rothermel model. Int J Wildl Fire 17(5):638–649
24. Loepfe L, Martinez-Vilalta J, Piñol J (2011) An integrative model of human-influenced fire regimes and landscape dynamics. Environ Model Softw 26(8):1028–1040
25. Lopes A, Cruz M, Viegas D (2002) i firestation an integrated software system for the numerical simulation of fire spread on complex topography. Environ Model Softw 17(3):269–285
26. McKenzie D, Peterson DL, Alvarado E (1997) Extrapolation problems in modeling fire effects at large spatial scales: a review. Int J Wildl Fire 6(4):165–176
27. Morvan D (2011) Physical phenomena and length scales governing the behaviour of wildfires: a case for physical modelling. Fire Technol 47:437–460
28. Reinhardt ED, Crookston NLL (2003) The fire and fuels extension to the forest vegetation simulator. Tech. Rep. RMRS-GTR-116, USDA Forest Service, Rocky Mountain Research Station, Fort Collins, CO
29. Rothermel RC (1972) A mathematical model for predicting fire spread in wildland fuels. Tech. Rep. INT-GTR-115, USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT
30. Rothermel RC, Rinehart GC (1983) Field procedures for verification and adjustment of fire behavior predictions. USDA, Forest Service, Intermountain Forest and Range Experiment Station
31. RStudio, Inc (2013) Easy web applications in R. URL: http://www.rstudio.com/shiny/
32. Salvador R, Pinol J, Tarantola S, Pla E (2001) Global sensitivity analysis and scale effects of a fire propagation model used over Mediterranean shrublands. Ecol Model 136(2):175–189
33. Sandberg DV, Riccardi CL, Schaaf MD (2007) Reformulation of rothermel's wildland fire behaviour model for heterogeneous fuelbeds this article is one of a selection of papers published in the special forum on the fuel characteristic classification system. Can J For Res 37(12):2438–2455
34. Santander Meteorology Group (2012) FUME package. URL: http://cran.r-project.org/web/packages/fume/index.html

Journal : 10694_FIRE Dispatch : 16-4-2014 Pages : 14
MS NO : 405 ☐ LE ☑ CP ☐ TYPESET ☑ DISK

35. Scott JH, Burgan RE (2005) Standard fire behavior fuel models: a comprehensive set for use with Rothermel's surface fire spread model. Tech. Rep. RMRS-GTR-153, USDA Forest Service, Rocky Mountain Research Station, Fort Collins, CO
36. Simard A, Deacon A, Adams K (1982) Nondirectional sampling of wildland fire spread. Fire Technol 18:221–228
37. Sneeuwjagt RJ, Frandsen WH (1977) Behavior of experimental grass fires vs. predictions based on Rothermel's fire model. Can J For Res 7(2):357–367
38. Sullivan AL (2009) Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models. Int J Wildl Fire 18(4):369–386
39. Vacchiano G, Motta R, Bovio G, Ascoli D (2013) Calibrating and testing the forest vegetation simulator to simulate tree encroachment and control measures for heathland restoration in southern europe. For Sci 1
40. Venables W, Smith D, R Core Team (2013) An introduction to R—Version 3.0.2. URL: http://cran.r-project.org/doc/manuals/R-intro.pdf
41. Wang X, Cantin A, Parisien MA, Wotton M, Anderson K, Flannigan M (2013) Fire weather index system and fire behaviour prediction system calculations. URL: http://cran.r-project.org/web/packages/fwi.fbp/index.html
42. Xu J (1994) Simulating the spread of wildfires using a geographic information system and remote sensing. PhD thesis, Rutgers University, New Brunswick, NJ

Journal : **10694_FIRE**	Dispatch : **16-4-2014**	Pages : **14**
	☐ LE	☐ TYPESET
MS NO : **405**	☑ CP	☑ DISK