TPMplt: R Toolkit for Dynamic Materials Modeling

Chen ZHANG [1,2]

[1]Tohoku University, [2]Institute for Material Research @ Sendai, Japan

Git URL: CubicZebra/TPMplt.git

2018.11.29

1. Introduction
   - Depandencies & Installations
   - VBT Data Frame
   - Workflow Overview
2. Practical Application
   - Hot Working Properties for Alloys
3. Under Developement
   - Insufficiencies & Solutions
   - Modification for Next Version

Depandencies

1. R framework:
   1.1 For Windows & OS X: Install via official installer
   1.2 For linux: Install r-base as root
2. An IDE for R: RStudio, Visual Studio (Windows only), ...
3. X11 framework:
   3.1 For Windows & OS X: XQuartz via offical installer
   3.2 For linux: `sudo apt-get install xorg openbox libx11-dev libglu1-mesa-dev libfreetype6-dev`

Installation

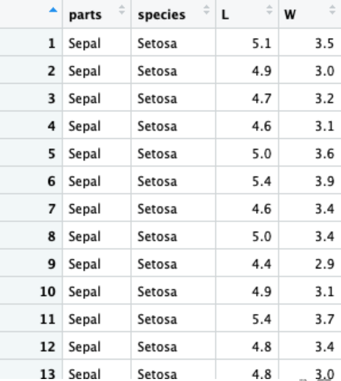
1. Version: 0.1.0
2. License: GPLv3
3. Installation:
   3.1 From CRAN (stable): `install.packages("TPMplt")`
   3.2 From Git Repo (developement):

```
if(!"devtools" %in% installed.packages())
    install.packages("devtools")
devtools::install_github("CubicZebra/TPMplt")
```

Conventional Data Frame

- Including *discrete* and *continous* variables
- *Discrete* ones copied and aligned in *individual columns*
- Take `iris3` as example →

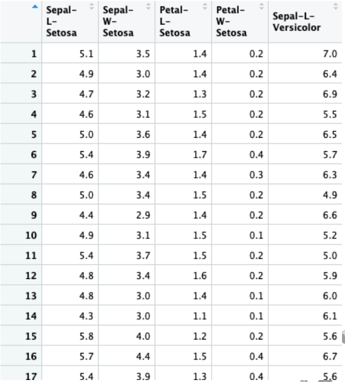
  (*note: `iris3` is a basic dataset for testing)



	parts	species	L	W
1	Sepal	Setosa	5.1	3.5
2	Sepal	Setosa	4.9	3.0
3	Sepal	Setosa	4.7	3.2
4	Sepal	Setosa	4.6	3.1
5	Sepal	Setosa	5.0	3.6
6	Sepal	Setosa	5.4	3.9
7	Sepal	Setosa	4.6	3.4
8	Sepal	Setosa	5.0	3.4
9	Sepal	Setosa	4.4	2.9
10	Sepal	Setosa	4.9	3.1
11	Sepal	Setosa	5.4	3.7
12	Sepal	Setosa	4.8	3.4
13	Sepal	Setosa	4.8	3.0

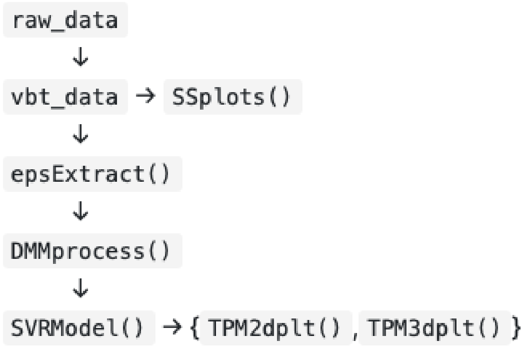
VBT Data Frame

- Including *discrete* and *continous* variables
- *Discrete* ones contained in *column names* with specific structure
  (*intuitively corresponding most experimental data)
- Take `iris3` as example →
- For details: VBTree package



	Sepal-L-Setosa	Sepal-W-Setosa	Petal-L-Setosa	Petal-W-Setosa	Sepal-L-Versicolor
1	5.1	3.5	1.4	0.2	7.0
2	4.9	3.0	1.4	0.2	6.4
3	4.7	3.2	1.3	0.2	6.9
4	4.6	3.1	1.5	0.2	5.5
5	5.0	3.6	1.4	0.2	6.5
6	5.4	3.9	1.7	0.4	5.7
7	4.6	3.4	1.4	0.3	6.3
8	5.0	3.4	1.5	0.2	4.9
9	4.4	2.9	1.4	0.2	6.6
10	4.9	3.1	1.5	0.1	5.2
11	5.4	3.7	1.5	0.2	5.0
12	4.8	3.4	1.6	0.2	5.9
13	4.8	3.0	1.4	0.1	6.0
14	4.3	3.0	1.1	0.1	6.1
15	5.8	4.0	1.2	0.2	5.6
16	5.7	4.4	1.5	0.4	6.7
17	5.4	3.9	1.3	0.4	5.6

Workflow

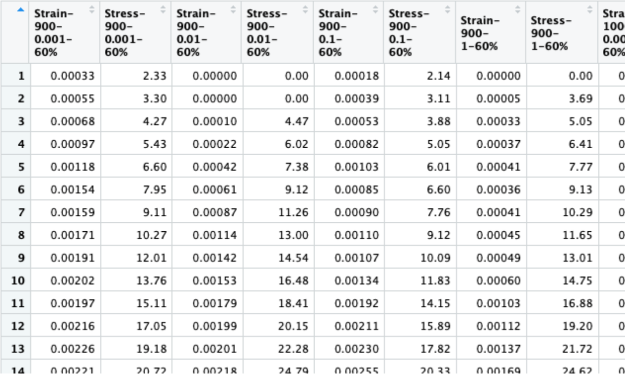


```
raw_data
   ↓
vbt_data → SSplots()
   ↓
epsExtract()
   ↓
DMMprocess()
   ↓
SVRModel() → { TPM2dplt() , TPM3dplt() }
```

Fe13CrWCuC

- Summary VBT data frame

  (32 columns with 4 temperatures * 4 strain rates * {Strain, Stress})

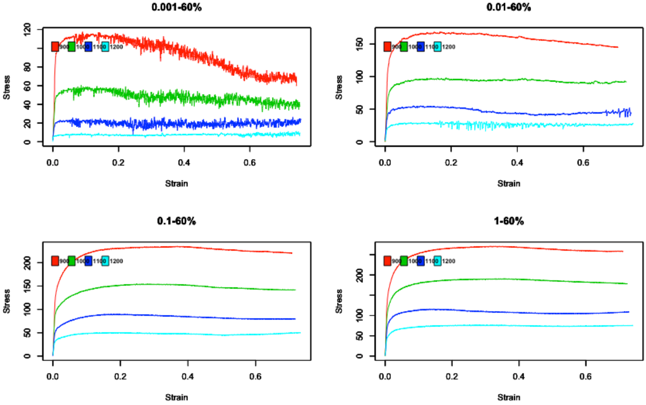


	Strain-900-0.001-60%	Stress-900-0.001-60%	Strain-900-0.01-60%	Stress-900-0.01-60%	Strain-900-0.1-60%	Stress-900-0.1-60%	Strain-900-1-60%	Stress-900-1-60%	Stra 100 0.00 60%
1	0.00033	2.33	0.00000	0.00	0.00018	2.14	0.00000	0.00	0
2	0.00055	3.30	0.00000	0.00	0.00039	3.11	0.00005	3.69	0
3	0.00068	4.27	0.00010	4.47	0.00053	3.88	0.00033	5.05	0
4	0.00097	5.43	0.00022	6.02	0.00082	5.05	0.00037	6.41	0
5	0.00118	6.60	0.00042	7.38	0.00103	6.01	0.00041	7.77	0
6	0.00154	7.95	0.00061	9.12	0.00085	6.60	0.00036	9.13	0
7	0.00159	9.11	0.00087	11.26	0.00090	7.76	0.00041	10.29	0
8	0.00171	10.27	0.00114	13.00	0.00110	9.12	0.00045	11.65	0
9	0.00191	12.01	0.00142	14.54	0.00107	10.09	0.00049	13.01	0
10	0.00202	13.76	0.00153	16.48	0.00134	11.83	0.00060	14.75	0
11	0.00197	15.11	0.00179	18.41	0.00192	14.15	0.00103	16.88	0
12	0.00216	17.05	0.00199	20.15	0.00211	15.89	0.00112	19.20	0
13	0.00226	19.18	0.00201	22.28	0.00230	17.82	0.00137	21.72	0
14	0.00221	20.72	0.00218	24.79	0.00255	20.33	0.00169	24.62	0

Fe13CrWCuC

- Stress-Strain plots grouped by temperature:

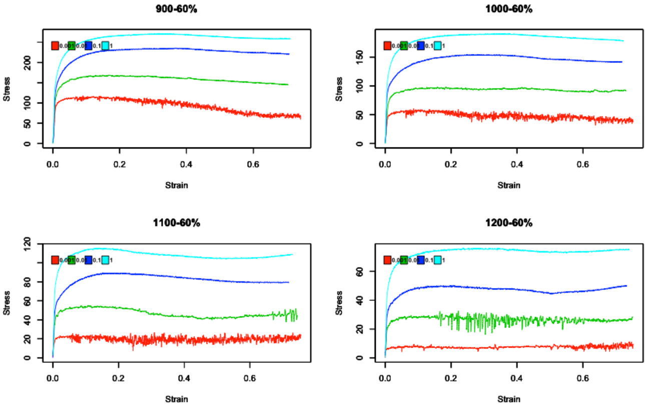
```
R> SSplot(vbt_data, 2, mfrow=c(2,2))
```



Fe13CrWCuC

- Stress-Strain plots grouped by strain rate:

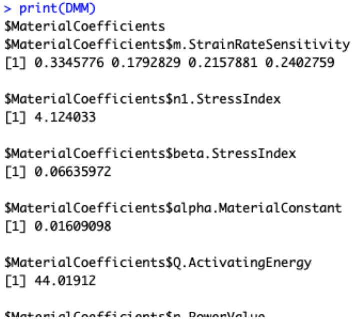
```
R> SSplot(vbt_data, 3, mfrow=c(2,2))
```



Fe13CrWCuC

- Build dynamic materials model (at 0.7 strain):

```
R> SRT <- epsExtract(vbt_data, 0.7, 2, 3)
R> DMM <- DMMprocess(SRT)

R> print(DMM)
```

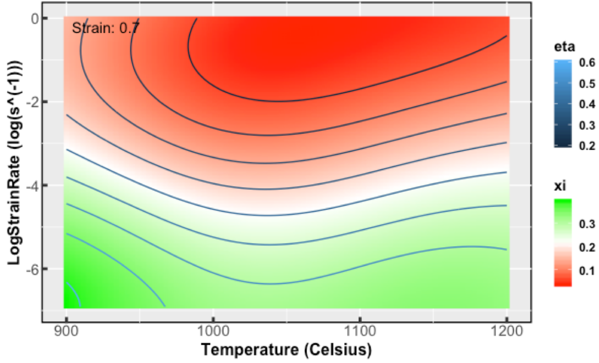


```
> print(DMM)
$MaterialCoefficients
$MaterialCoefficients$m.StrainRateSensitivity
[1] 0.3345776 0.1792829 0.2157881 0.2402759

$MaterialCoefficients$n1.StressIndex
[1] 4.124033

$MaterialCoefficients$beta.StressIndex
[1] 0.06635972

$MaterialCoefficients$alpha.MaterialConstant
[1] 0.01609098

$MaterialCoefficients$Q.ActivatingEnergy
[1] 44.01912

$MaterialCoefficients$n.PowerValue
```

Fe13CrWCuC

- Regression and 2d visulization (radial basis kernel (`rbf`) in SVM):

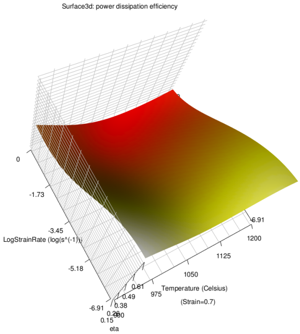
```
R> SVR <- SVRModel(DMM)

R> TPM2dplt(SVR)
```

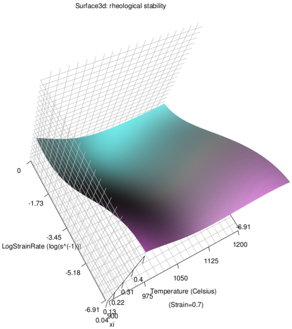


Fe13CrWCuC

- 3d visulization:

```
R> TPM3dplt(SVR)
```





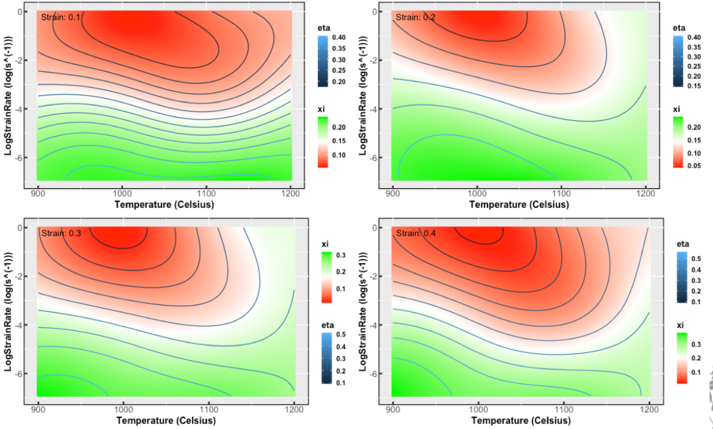
Fe16CrWCuC

- Similar as workflow of 13Cr
- Sequential plots are available using loop script:

```
R> seqeps <- seq(0.1, 0.8, 0.1)
R> for (i in 1:8) {
    + SRT <- epsExtract(vbt_data, seqeps[i], 2, 3)
    + DMM <- DMMprocess(SRT)
    + PLTbd <- SVRModel(DMM)
    + plt <- TPM2dplt(PLTbd)
    + print(plt)
    + Sys.sleep(5)
    + }
R> #Sys.sleep() controls holding time for each plot
```

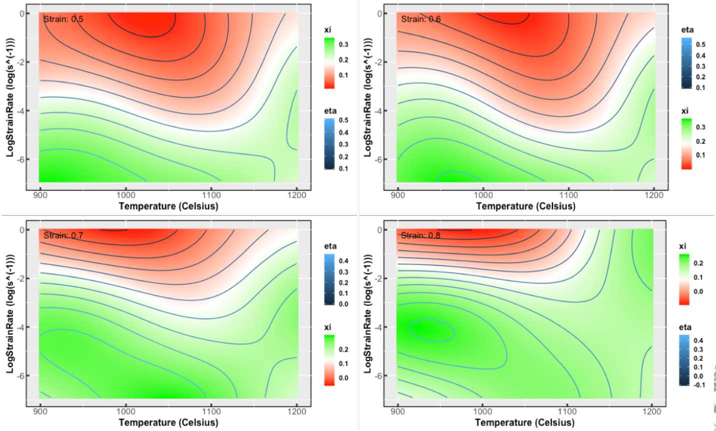
Fe16CrWCuC

- Sequential 2d plots:



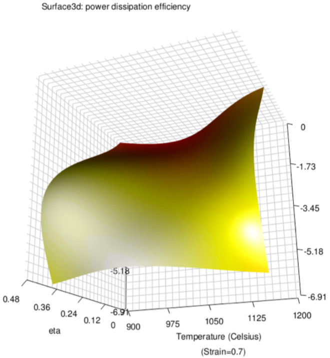
Fe16CrWCuC

- Sequential 2d plots:

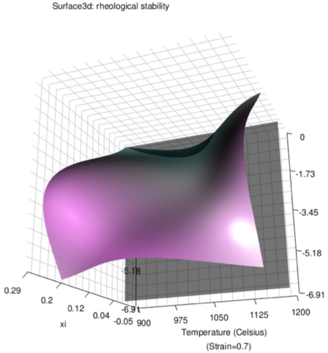


Fe16CrWCuC

- 3d plot using 0.7 strain (unstable plan added for plot of ξ):



Surface3d: power dissipation efficiency



Surface3d: rheological stability

Insufficiencies

1. $\because m \in (0,1] \therefore \eta = 2m/(1+m) \in (0,1]$
   However, some predicted η values are still out of range

2. Result of material constant A looks strange, since it's too small in magnitude

3. Values for contours generated by `TPM2dplt()` are invisible

Corresponding Solutions

1. Rescaling the η using $\tilde{\eta} = (\eta - \eta_{min})/(\eta_{max} - \eta_{min})$
2. Modify related calculations in model-buiding function
3. Updata 2d visualization function with the `directlabels` package

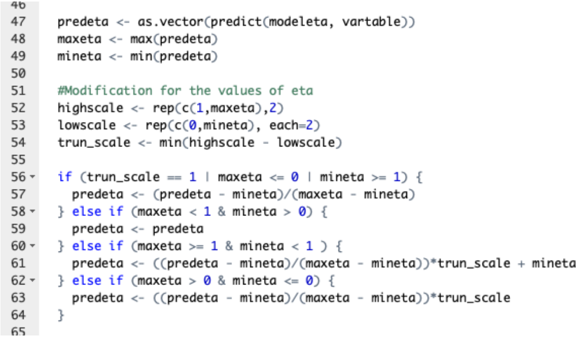
Functions to be Updated

- Functions required modification includes:

  1. `DMMprocess()`
  2. `SVRModel()` (fixed in Dev Version)
  3. `TPM2dplt()` (fixed in Dev Version)

Preview for Next Version

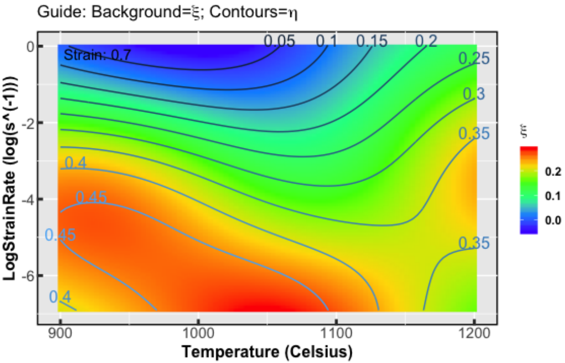
- Rescaling algorithm for the predicted η values in `SVRModel()`:



```r
predeta <- as.vector(predict(modeleta, vartable))
maxeta <- max(predeta)
mineta <- min(predeta)

#Modification for the values of eta
highscale <- rep(c(1,maxeta),2)
lowscale <- rep(c(0,mineta), each=2)
trun_scale <- min(highscale - lowscale)

if (trun_scale == 1 | maxeta <= 0 | mineta >= 1) {
  predeta <- (predeta - mineta)/(maxeta - mineta)
} else if (maxeta < 1 & mineta > 0) {
  predeta <- predeta
} else if (maxeta >= 1 & mineta < 1 ) {
  predeta <- ((predeta - mineta)/(maxeta - mineta))*trun_scale + mineta
} else if (maxeta > 0 & mineta <= 0) {
  predeta <- ((predeta - mineta)/(maxeta - mineta))*trun_scale
}
```

Preview for Next Version

- New `TPM2dplt()` using a) the rainbow color control; b) labels for η contours.



Thanks for your attention.