Statistical Modeling of Loss Distributions Using actuar

by Vincent Goulet and Mathieu Pigeon

Introduction

actuar (Dutang et al., 2008) is a package providing additional Actuarial Science functionality to R. Although various packages on CRAN provide functions useful to actuaries, **actuar** aims to serve as a central location for more specifically actuarial functions and data sets. The current feature set of the package can be split in four main categories: loss distributions modeling, risk theory (including ruin theory), simulation of compound hierarchical models and credibility theory.

This paper reviews the loss distributions modeling features of the package — those most likely to interest R News readers and to have links with other fields of statistical practice.

Actuaries need to model claim amounts distributions for ratemaking, loss reserving and other risk evaluation purposes. Typically, claim amounts data are nonnegative and skewed to the right, often heavily. The probability laws used in modeling must match these characteristics. Furthermore, depending on the line of business, data can be truncated from below, censored from above or both.

The main **actuar** features to aid in loss modeling are the following:

1. Introduction of 18 additional probability laws and functions to get raw moments, limited moments and the moment generating function.

2. Fairly extensive support of grouped data.

3. Calculation of the empirical raw and limited moments.

4. Minimum distance estimation using three different measures.

5. Treatment of coverage modifications (deductibles, limits, inflation, coinsurance).

Probability laws

R already includes functions to compute the probability density function (pdf), the cumulative distribution function (cdf) and the quantile function of a fair number of probability laws, as well as functions to generate variates from these laws. For some root `*foo*`, the functions are named `d*foo*`, `p*foo*`, `q*foo*` and `r*foo*`, respectively.

The **actuar** package provides `d`, `p`, `q` and `r` functions for all the probability laws useful for loss severity modeling found in Appendix A of Klugman et al. (2004) and not already present in base R, excluding the inverse Gaussian and log-*t* but including the loggamma distribution (Hogg and Klugman, 1984). We tried to make these functions as similar as possible to those in the **stats** package, with respect to the interface, the names of the arguments and the handling of limit cases.

Table 1 lists the supported distributions as named in Klugman et al. (2004) along with the root names of the R functions. The name or the parametrization of some distributions may differ in other fields; check with the `lossdist` package vignette for the pdf and cdf of each distribution.

In addition to the `d`, `p`, `q` and `r` functions, the package provides `m`, `lev` and `mgf` functions to compute, respectively, theoretical raw moments

$$m_k = \mathbb{E}[X^k], \tag{1}$$

theoretical limited moments

$$\mathbb{E}[(X \wedge x)^k] = \mathbb{E}[(\min X, x)^k] \tag{2}$$

and the moment generating function

$$M_X(t) = \mathbb{E}[e^{tX}], \tag{3}$$

when it exists. Every probability law of Table 1 is supported, plus the following ones: beta, exponential, chi-square, gamma, lognormal, normal (except `lev`), uniform and Weibull of base R, and the inverse Gaussian distribution of package **SuppDists** (Wheeler, 2006). The `m` and `lev` functions are especially useful with estimation methods based on the matching of raw or limited moments; see below for their empirical counterparts. The `mgf` functions are introduced in the package mostly for calculation of the adjustment coefficient in ruin theory; see the `"risk"` package vignette.

In addition to the 17 distributions of Table 1, the package provides support for phase-type distributions (Neuts, 1981). These are not so much included in the package for statistical inference, but rather for ruin probability calculations. A phase-type distribution is defined as the distribution of the time until absorption of a continuous time, finite state Markov process with m transient states and one absorbing state. Let

$$Q = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{t} \\ \boldsymbol{0} & 0 \end{bmatrix} \tag{4}$$

be the transition rates matrix (or intensity matrix) of such a process and let $(\boldsymbol{\pi}, \pi_{m+1})$ be the initial probability vector. Here, $\boldsymbol{T}$ is an $m \times m$ non-singular matrix with $t_{ii} < 0$ for $i = 1, \dots, m$ and $t_{ij} \geq 0$ for $i \neq j$,

Table 1: Probability laws supported by **actuar** classified by family and root names of the R functions.

Family	Distribution	Root (alias)
Transformed beta	Transformed beta	`trbeta` (`pearson6`)
	Burr	`burr`
	Loglogistic	`llogis`
	Paralogistic	`paralogis`
	Generalized Pareto	`genpareto`
	Pareto	`pareto` (`pareto2`)
	Inverse Burr	`invburr`
	Inverse Pareto	`invpareto`
	Inverse paralogistic	`invparalogis`
Transformed gamma	Transformed gamma	`trgamma`
	Inverse transformed gamma	`invtrgamma`
	Inverse gamma	`invgamma`
	Inverse Weibull	`invweibull` (`lgompertz`)
	Inverse exponential	`invexp`
Other	Loggamma	`lgamma`
	Single parameter Pareto	`pareto1`
	Generalized beta	`genbeta`

$\boldsymbol{t} = -\boldsymbol{T}\boldsymbol{e}$ and $\boldsymbol{e}$ is a column vector with all components equal to 1. Then the cdf of the time until absorption random variable with parameters $\boldsymbol{\pi}$ and $\boldsymbol{T}$ is

$$F(x) = \begin{cases} 1 - \boldsymbol{\pi} e^{\boldsymbol{T}x} \boldsymbol{e}, & x > 0 \\ \pi_{m+1}, & x = 0, \end{cases} \tag{5}$$

where

$$e^{\boldsymbol{M}} = \sum_{n=0}^{\infty} \frac{\boldsymbol{M}^n}{n!} \tag{6}$$

is the *matrix exponential* of matrix $\boldsymbol{M}$.

The exponential, the Erlang (gamma with integer shape parameter) and discrete mixtures thereof are common special cases of phase-type distributions. The package provides `d`, `p`, `r`, `m` and `mgf` functions for phase-type distributions. The root is `phtype` and parameters $\boldsymbol{\pi}$ and $\boldsymbol{T}$ are named `prob` and `rates`, respectively.

The core of all the functions presented in this section is written in C for speed. The matrix exponential C routine is based on `expm()` from the package **Matrix** (Bates and Maechler, 2007).

Grouped data

What is commonly referred to in Actuarial Science as grouped data is data represented in an interval-frequency manner. In insurance applications, a grouped data set will typically report that there were n_j claims in the interval $(c_{j-1}, c_j]$, $j = 1, \dots, r$ (with the possibility that $c_r = \infty$). This representation is much more compact than an individual data set (where the value of each claim is known), but it also carries far less information. Now that storage space in computers has almost become a non-issue, grouped data has somewhat fallen out of fashion.

Still, grouped data remains useful in some fields of actuarial practice and for parameter estimation. For these reasons, **actuar** provides facilities to store, manipulate and summarize grouped data. A standard storage method is needed since there are many ways to represent grouped data in the computer: using a list or a matrix, aligning the n_js with the c_{j-1}s or with the c_js, omitting c_0 or not, etc. Moreover, with appropriate extraction, replacement and summary functions, manipulation of grouped data becomes similar to that of individual data.

First, function `grouped.data` creates a grouped data object similar to — and inheriting from — a data frame. The input of the function is a vector of group boundaries $c_0, c_1, \dots, c_r$ and one or more vectors of group frequencies $n_1, \dots, n_r$. Note that there should be one group boundary more than group frequencies. Furthermore, the function assumes that the intervals are contiguous. For example, the following data is entered and represented in R as

Group	Frequency (Line 1)	Frequency (Line 2)
$(0, 25]$	30	26
$(25, 50]$	31	33
$(50, 100]$	57	31
$(100, 150]$	42	19
$(150, 250]$	65	16
$(250, 500]$	84	11

```
> x <- grouped.data(Group = c(0, 25,
+     50, 100, 150, 250, 500), Line.1 = c(30,
+     31, 57, 42, 65, 84), Line.2 = c(26,
+     33, 31, 19, 16, 11))
```

Object `x` is stored internally as a list with class

```
> class(x)

[1] "grouped.data" "data.frame"
```

With a suitable `print` method, these objects can be displayed in an unambiguous manner:

```
> x

       Group Line.1 Line.2
1   (0,  25]     30     26
2  (25,  50]     31     33
3  (50, 100]     57     31
4 (100, 150]     42     19
5 (150, 250]     65     16
6 (250, 500]     84     11
```

Second, the package supports the most common extraction and replacement methods for `"grouped.data"` objects using the usual `[` and `[<-` operators; see `?Extract.grouped.data` for details.

The package defines methods of a few existing summary functions for grouped data objects. Computing the mean

$$\sum_{j=1}^{r} \left(\frac{c_{j-1} + c_j}{2} \right) n_j \bigg/ \sum_{j=1}^{r} n_j \qquad (7)$$

is made simple with a method for the `mean` function:

```
> mean(x)

Line.1 Line.2
 179.8   99.9
```

Higher empirical moments can be computed with `emm`; see below.

A method for function `hist` draws a histogram for already grouped data. Only the first frequencies column is considered (see Figure 1 for the resulting graph):

```
> hist(x[, -3])
```

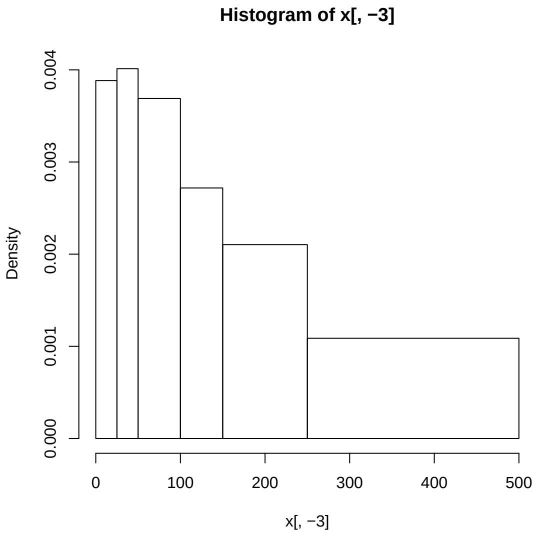

Figure 1: Histogram of a grouped data object

R has a function `ecdf` to compute the empirical cdf of an individual data set,

$$F_n(x) = \frac{1}{n} \sum_{j=1}^{n} I\{x_j \leq x\},$$

where $I\{\mathcal{A}\} = 1$ if $\mathcal{A}$ is true and $I\{\mathcal{A}\} = 0$ otherwise. The function returns a `"function"` object to compute the value of $F_n(x)$ in any x. The approximation of the empirical cdf for grouped data is called an ogive (Klugman et al., 1998; Hogg and Klugman, 1984). It is obtained by joining the known values of $F_n(x)$ at group boundaries with straight line segments:

$$\tilde{F}_n(x) = \begin{cases} 0, \; x \leq c_0 \\ \dfrac{(c_j - x)F_n(c_{j-1}) + (x - c_{j-1})F_n(c_j)}{c_j - c_{j-1}}, \\ \quad c_{j-1} < x \leq c_j \\ 1, \; x > c_r. \end{cases} \qquad (8)$$

The package includes a function `ogive` that otherwise behaves exactly like `ecdf`. In particular, methods for functions `knots` and `plot` allow, respectively, to obtain the knots $c_0, c_1, \dots, c_r$ of the ogive and a graph.

Calculation of empirical moments

In the sequel, we frequently use two data sets provided by the package: the individual dental claims (`dental`) and grouped dental claims (`gdental`) of Klugman et al. (2004).

The package provides two functions useful for estimation based on moments. First, function `emm` computes the kth empirical moment of a sample, whether in individual or grouped data form:

```
> emm(dental, order = 1:3)

[1] 3.355e+02 2.931e+05 3.729e+08

> emm(gdental, order = 1:3)

[1] 3.533e+02 3.577e+05 6.586e+08
```

Second, in the same spirit as `ecdf` and `ogive`, function `elev` returns a function to compute the empirical limited expected value — or first limited moment — of a sample for any limit. Again, there are methods for individual and grouped data (see Figure 2 for the graphs):

```
> lev <- elev(dental)
> lev(knots(lev))

 [1]  16.0  37.6  42.4  85.1 105.5 164.5
 [7] 187.7 197.9 241.1 335.5

> plot(lev, type = "o", pch = 19)
> lev <- elev(gdental)
> lev(knots(lev))
```

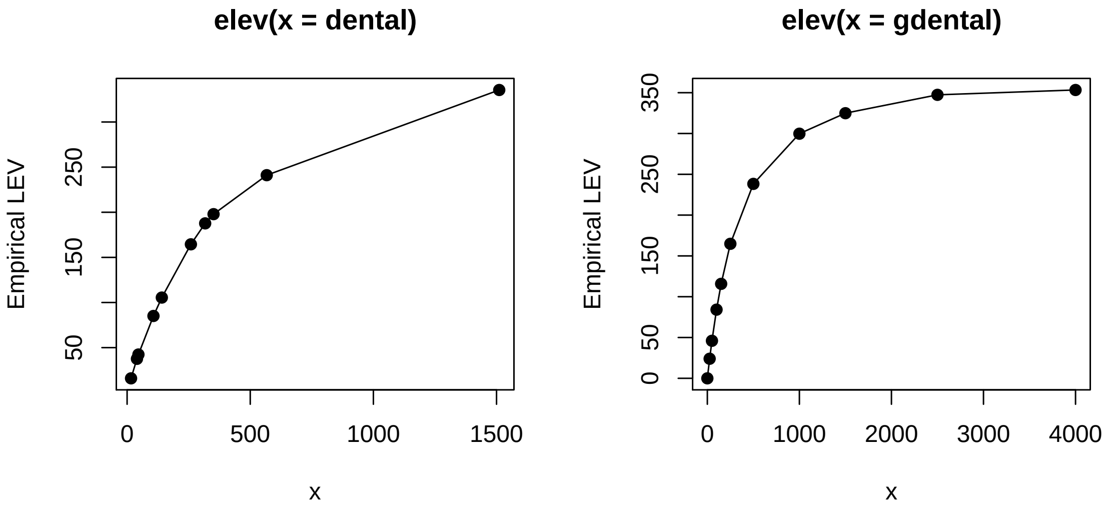

Figure 2: Empirical limited expected value function of an individual data object (left) and a grouped data object (right)

```
 [1]   0.00  24.01  46.00  84.16 115.77
 [6] 164.85 238.26 299.77 324.90 347.39
[11] 353.34

> plot(lev, type = "o", pch = 19)
```

Minimum distance estimation

Two methods are widely used by actuaries to fit models to data: maximum likelihood and minimum distance. The first technique applied to individual data is well covered by function `fitdistr` of the **MASS** package (Venables and Ripley, 2002).

The second technique minimizes a chosen distance function between theoretical and empirical distributions. The **actuar** package provides function `mde`, very similar in usage and inner working to `fitdistr`, to fit models according to any of the following three distance minimization methods.

1. The Cramér-von Mises method (`CvM`) minimizes the squared difference between the theoretical cdf and the empirical cdf or ogive at their knots:

$$d(\theta) = \sum_{j=1}^{n} w_j [F(x_j; \theta) - F_n(x_j; \theta)]^2 \qquad (9)$$

for individual data and

$$d(\theta) = \sum_{j=1}^{r} w_j [F(c_j; \theta) - \tilde{F}_n(c_j; \theta)]^2 \qquad (10)$$

for grouped data. Here, $F(x)$ is the theoretical cdf of a parametric family, $F_n(x)$ is the empirical cdf, $\tilde{F}_n(x)$ is the ogive and $w_1 \geq 0, w_2 \geq 0, \dots$ are arbitrary weights (defaulting to 1).

2. The modified chi-square method (`chi-square`) applies to grouped data only and minimizes the squared difference between the expected and observed frequency within each group:

$$d(\theta) = \sum_{j=1}^{r} w_j [n(F(c_j; \theta) - F(c_{j-1}; \theta)) - n_j]^2, \qquad (11)$$

where $n = \sum_{j=1}^{r} n_j$. By default, $w_j = n_j^{-1}$. The method is called “modified” because the default denominators are the observed rather than the expected frequencies.

3. The layer average severity method (`LAS`) applies to grouped data only and minimizes the squared difference between the theoretical and empirical limited expected value within each group:

$$d(\theta) = \sum_{j=1}^{r} w_j [\text{LAS}(c_{j-1}, c_j; \theta) - \tilde{\text{LAS}}_n(c_{j-1}, c_j; \theta)]^2, \qquad (12)$$

where $\text{LAS}(x, y) = \mathbb{E}[X \wedge y] - \mathbb{E}[X \wedge x]$, $\tilde{\text{LAS}}_n(x, y) = \tilde{\mathbb{E}}_n[X \wedge y] - \tilde{\mathbb{E}}_n[X \wedge x]$, and $\tilde{\mathbb{E}}_n[X \wedge x]$ is the empirical limited expected value for grouped data.

The arguments of `mde` are a data set, a function to compute $F(x)$ or $\mathbb{E}[X \wedge x]$, starting values for the optimization procedure and the name of the method to use. The empirical functions are computed with `ecdf`, `ogive` or `elev`.

The expressions below fit an exponential distribution to the grouped dental data set, as per Example 2.21 of Klugman et al. (1998):

```
> mde(gdental, pexp,
+     start = list(rate = 1/200),
+     measure = "CvM")

   rate
 0.003551

 distance
  0.002842
```

```
> mde(gdental, pexp,
+     start = list(rate = 1/200),
+     measure = "chi-square")

   rate
  0.00364

 distance
     13.54
```

```
> mde(gdental, levexp,
+     start = list(rate = 1/200),
+     measure = "LAS")

   rate
  0.002966

 distance
     694.5
```

It should be noted that optimization is not always that simple to achieve. For example, consider the problem of fitting a Pareto distribution to the same data set using the Cramér–von Mises method:

```
> mde(gdental, ppareto,
+     start = list(shape = 3,
+     scale = 600), measure = "CvM")

Error in mde(gdental, ppareto,
             start = list(shape = 3,
               scale = 600),
             measure = "CvM") :
  optimization failed
```

Working in the log of the parameters often solves the problem since the optimization routine can then flawlessly work with negative parameter values:

```
> f <- function(x, lshape, lscale) ppareto(x,
+     exp(lshape), exp(lscale))
> (p <- mde(gdental, f, list(lshape = log(3),
+     lscale = log(600)), measure = "CvM"))

 lshape   lscale
   1.581    7.128

  distance
  0.0007905
```

The actual estimators of the parameters are obtained with

```
> exp(p$estimate)

  lshape   lscale
   4.861 1246.485
```

This procedure may introduce additional bias in the estimators, though.

Coverage modifications

Let X be the random variable of the actual claim amount for an insurance policy, Y^L be the random variable of the amount paid per loss and Y^P be the random variable of the amount paid per payment. The terminology for the last two random variables refers to whether or not the insurer knows that a loss occurred. Now, the random variables X, Y^L and Y^P will differ if any of the following coverage modifications are present for the policy: an ordinary or a franchise deductible, a limit, coinsurance or inflation adjustment (see Klugman et al., 2004, Chapter 5 for precise definitions of these terms). Table 2 summarizes the definitions of Y^L and Y^P.

The effect of an ordinary deductible is known as truncation from below, and that of a policy limit as censoring from above. Censored data is very common in survival analysis; see the package **survival** (Lumley, 2008) for an extensive treatment in R. Yet, **actuar** provides a different approach.

Suppose one wants to use censored data $Y_1, \dots, Y_n$ from the random variable Y to fit a model on the unobservable random variable X. This requires expressing the pdf or cdf of Y in terms of the pdf or cdf of X. Function `coverage` of **actuar** does just that: given a pdf or cdf and any combination of the coverage modifications mentioned above, `coverage` returns a function object to compute the pdf or cdf of the modified random variable. The function can then be used in modeling or plotting like any other `d`*`foo`* or `p`*`foo`* function.

For example, let Y represent the amount paid (per payment) by an insurer for a policy with an ordinary deductible d and a limit $u - d$ (or maximum covered loss of u). Then the definition of Y is

$$Y = \begin{cases} X - d, & d \leq X \leq u \\ u - d, & X \geq u \end{cases} \tag{13}$$

and its pdf is

$$f_Y(y) = \begin{cases} 0, & y = 0 \\ \dfrac{f_X(y+d)}{1 - F_X(d)}, & 0 < y < u - d \\ \dfrac{1 - F_X(u)}{1 - F_X(d)}, & y = u - d \\ 0, & y > u - d. \end{cases} \tag{14}$$

Assume X has a gamma distribution. Then an R function to compute the pdf (14) in any y for a deductible $d = 1$ and a limit $u = 10$ is obtained with `coverage` as follows:

```
> f <- coverage(pdf = dgamma, cdf = pgamma,
+     deductible = 1, limit = 10)
> f(0, shape = 5, rate = 1)
```

Table 2: Coverage modifications for per-loss variable (Y^L) and per-payment variable (Y^P) as defined in Klugman et al. (2004).

Coverage modification	Per-loss variable (Y^L)	Per-payment variable (Y^P)
Ordinary deductible (d)	$\begin{cases} 0, & X \le d \\ X - d, & X > d \end{cases}$	$\begin{cases} X - d, & X > d \end{cases}$
Franchise deductible (d)	$\begin{cases} 0, & X \le d \\ X, & X > d \end{cases}$	$\begin{cases} X, & X > d \end{cases}$
Limit (u)	$\begin{cases} X, & X \le u \\ u, & X > u \end{cases}$	$\begin{cases} X, & X \le u \\ u, & X > u \end{cases}$
Coinsurance (α)	αX	αX
Inflation (r)	$(1 + r)X$	$(1 + r)X$

```
[1] 0

> f(5, shape = 5, rate = 1)

[1] 0.1343

> f(9, shape = 5, rate = 1)

[1] 0.02936

> f(12, shape = 5, rate = 1)

[1] 0
```

The function `f` is built specifically for the coverage modifications submitted and contains as little useless code as possible.

Let object `y` contain a sample of claims amounts from policies with the above deductible and limit. Then one can fit a gamma distribution by maximum likelihood to the claim severity process as follows:

```
> library(MASS)
> fitdistr(y, f, start = list(shape = 2,
+     rate = 0.5))

   shape      rate
  4.1204   0.8230
 (0.7054) (0.1465)
```

The package vignette `"coverage"` contains more detailed pdf and cdf formulas under various combinations of coverage modifications.

Conclusion

This paper reviewed the main loss modeling features of **actuar**, namely many new probability laws; new utility functions to access the raw moments, the limited moments and the moment generating function of these laws and some of base R; functions to create and manipulate grouped data sets; functions to ease calculation of empirical moments, in particular for grouped data; a function to fit models by distance minimization; a function to help work with censored data or data subject to coinsurance or inflation adjustments.

We hope some of the tools presented here may be of interest outside the field they were developed for, perhaps provided some adjustments in terminology and nomenclature.

Finally, we note that the **distr*XXX*** family of packages (Ruckdeschel et al., 2006) provides a general, object oriented approach to some of the features of **actuar**, most notably the calculation of moments for many distributions (although not necessarily those presented here), minimum distance estimation and censoring.

Acknowledgments

This research benefited from financial support from the Natural Sciences and Engineering Research Council of Canada and from the *Chaire d'actuariat* (Actuarial Science Chair) of Université Laval. We also thank one anonymous referee and the R News editors for corrections and improvements to the paper.

Bibliography

D. Bates and M. Maechler. **Matrix**: *A matrix package for R*, 2007. R package version 0.999375-3.

C. Dutang, V. Goulet, and M. Pigeon. **actuar**: An R package for actuarial science. *Journal of Statistical Software*, 25(7), 2008. URL `http://www.actuar-project.org`.

R. V. Hogg and S. A. Klugman. *Loss Distributions*. Wiley, New York, 1984. ISBN 0-4718792-9-0.

S. A. Klugman, H. H. Panjer, and G. Willmot. *Loss Models: From Data to Decisions*. Wiley, New York, 1998. ISBN 0-4712388-4-8.

S. A. Klugman, H. H. Panjer, and G. Willmot. *Loss Models: From Data to Decisions*. Wiley, New York, 2 edition, 2004. ISBN 0-4712157-7-5.

T. Lumley. **survival***: Survival analysis, including penalised likelihood*, 2008. R package version 2.34. S original by Terry Therneau.

M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Dover Publications, 1981. ISBN 978-0-4866834-2-3.

P. Ruckdeschel, M. Kohl, T. Stabla, and F. Camphausen. S4 classes for distributions. *R News*, 6 (2):2–6, May 2006. URL http://distr.r-forge.r-project.org.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, 4 edition, 2002. ISBN 0-3879545-7-0.

B. Wheeler. **SuppDists***: Supplementary distributions*, 2006. URL http://www.bobwheeler.com/stat. R package version 1.1-0.

Vincent Goulet and Mathieu Pigeon
Université Laval, Canada
vincent.goulet@act.ulaval.ca