prioritizr

Systematic conservation planning in

Planificación sistematica de la conservación en

prioritizr.net

Jeffrey Hanson & Richard Schuster

jeffrey.hanson@uqconnect.edu.au

richard.schuster@glel.carleton.ca

Acknowledgements/Agradecimientos

- Jeff Hanson
- Matt Strimas-Mackey
- Nina Morrell
- Matt Watts
- Joe Bennett
- Peter Arcese
- Hugh Possingham

LIBER ERO FELLOWSHIP PROGRAM

Carleton UNIVERSITY

UBC THE UNIVERSITY OF BRITISH COLUMBIA

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

The Why?/ El por qué?

- Conservation planning = challenging/ Planificación de la conservación = desafío
- Range of objectives and constraints/ Existe una gama de objetivos y limitaciones.
- Analytical tools tailored to specific tasks/ Herramientas analíticas adaptadas a tareas específicas.

Proposal:

Flexible, powerful framework

+

User friendly (browser) interface

Propuesta:

Marco flexible y potente

+

Interfaz amigable para el usuario (navegador)

Why ? Por qué ?

- It’s free/ Es gratuito
- Reproducibility/ Replicabilidad
- Flexible interface/ Interfaz flexible
- Diverse package ecosystem/ Diversidad de paquetes
- So many resources/Gran cantidad de recursos: Stackoverflow, tutorials, blogs, twitter, slack channels, mailing lists

prioritizr

1. Design your problem/ Diseña el problema

1. Find optimal solutions/ Encuentra soluciones óptimas

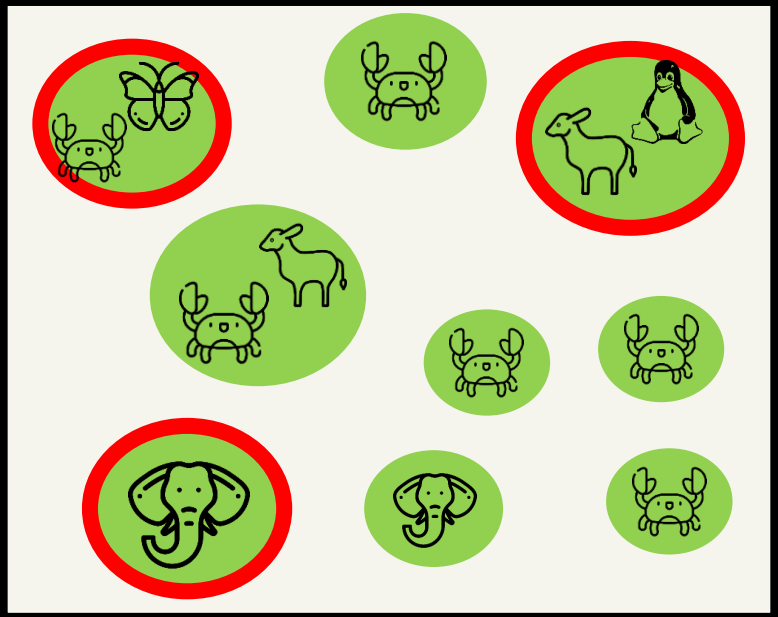
1. Ideally, fast!/ Ideal y rápido

Decision science/ Ciencia de la desición

- Goal: what is our vision for the future?/ Meta: ¿cuál es nuestra visión para el futuro?
- Objective: what quantity are we maximizing/minimizing to help achieve the goal?/Objetivo: ¿qué cantidad estamos maximizando / minimizando para ayudar a lograr el objetivo?
- Constraints: what things must our solution do to help achieve the goal?/ Restricciones: ¿qué cosas debe hacer nuestra solución para ayudar a lograr el objetivo?
- Decisions: what actions could we do to maximize/minimize the objective?/ Decisiones

Case-study: Reserve design/ Caso de studio: Diseño de una reserva

- Goal: conserve biodiversity
- Objective: min. # of islands
- Constraints: sufficient habitat for each species
- Decisions: which places should be protected?



- Meta: conservar la biodiversidad
- Objetivo: min. # de islas
- Restricciones: hábitat suficiente para cada especie
- Decisiones: ¿qué lugares deben protegerse?

prioritzr features/ Caracteristicas de prioritzr

https://prioritizr.net/articles/prioritizr.html#package-overview

Objectives
Targets
Constraints
Penalties
Decisions
Solvers
Solution portfolios
Irreplaceability

Single zone
Multi zone

Objetivos
Metas
Restricciones
Sanciones
Decisiones
Solucionadores
Portafolios de soluciones
Irreemplazabilidad

Zona única
Multizona

Extensive documentation with lots of tutorials and examples: prioritizr.net

Amplia documentación con variedad de tutoriales y ejemplos: prioritizr.net

Objective functions/ Funciones objetivo

https://prioritizr.net/reference/objectives.html

The following objectives can be added to a conservation planning `problem`:

`add_min_set_objecti...`	Minimize the cost of the solution whilst ensuring that all targets are met. This objective is similar to that used in *Marxan*.
`add_max_cover_objec...`	Represent at least one instance of as many features as possible within a given budget.
`add_max_features_ob...`	Fulfill as many targets as possible while ensuring that the cost of the solution does not exceed a budget.
`add_min_shortfall_o...`	Minimize the shortfall for as many targets as possible while ensuring that the cost of the solution does not exceed a budget.
`add_max_phylo_div_o...`	Maximize the phylogenetic diversity of the features represented in the solution subject to a budget.
`add_max_phylo_end_o...`	Maximize the phylogenetic endemism of the features represented in the solution subject to a budget.
`add_max_utility_obj...`	Secure as much of the features as possible without exceeding a budget.

Penalties/ Sanciones

https://prioritizr.net/reference/penalties.html

- Boundary penalties = symmetric/
  Penalizaciones de límite = simétricas

```
add_boundary_penalties(x, penalty, edge_factor = rep(0.5,
  number_of_zones(x)), zones = diag(number_of_zones(x)), data = NULL)
```

- Connectivity penalties = asymmetric allowed/
  Penalizaciones de conectividad = asimétrico permitido

Constraints/ Restricciones

https://prioritizr.net/reference/constraints.html

`add_locked_in_const...`	Add constraints to ensure that certain planning units are selected in the solution.
`add_locked_out_cons...`	Add constraints to ensure that certain planning units are not selected in the solution.
`add_neighbor_constr...`	Add constraints to ensure that all selected planning units have at least a certain number of neighbors.
`add_contiguity_cons...`	Add constraints to a ensure that all selected planning units are spatially connected to each other and form a single contiguous unit.
`add_feature_contigu...`	Add constraints to #' ensure that each feature is represented in a contiguous unit of dispersible habitat. These constraints are a more advanced version of those implemented in the `add_contiguity_constraints` function, because they ensure that each feature is represented in a contiguous unit and not that the entire solution should form a contiguous unit.
`add_mandatory_alloc...`	Add constraints to ensure that every planning unit is allocated to a management zone in the solution. **This function can only be used with problems that contain multiple zones.**

Portfolios/ Portafolios

https://prioritizr.net/reference/portfolios.html

`add_default_portfol...`	Generate a single solution.
`add_extra_portfolio`	Generate a portfolio of solutions by storing feasible solutions found during the optimization process. This method is useful for quickly obtaining multiple solutions, but does not provide any guarantees on the number of solutions, or the quality of solutions. Note that it requires the *Gurobi* solver.
`add_top_portfolio`	Generate a portfolio of solutions by finding a pre-specified number of solutions that are closest to optimality (i.e the top solutions). This is useful for examining differences among near-optimal solutions. It can also be used to generate multiple solutions and, in turn, to calculate selection frequencies for small problems. Note that it requires the *Gurobi* solver.
`add_gap_portfolio`	Generate a portfolio of solutions by finding a certain number of solutions that are all within a pre-specified optimality gap. This method is useful for generating multiple solutions that can be used to calculate selection frequencies for moderate and large-sized problems (similar to *Marxan*). Note that it requires the *Gurobi* solver.
`add_cuts_portfolio`	Generate a portfolio of distinct solutions within a pre-specified optimality gap using Bender's cuts. This is recommended as a replacement for `add_top_portfolio` when the *Gurobi* software is not available.
`add_shuffle_portfol...`	Generate a portfolio of solutions by randomly reordering the data prior to attempting to solve the problem. This is recommended as a replacement for `add_gap_portfolio` when the *Gurobi* software is not available.

Irreplaceability/ Irreemplazabilidad

https://prioritizr.net/reference/irreplaceability.html

The following methods are available for calculating irreplaceability scores:

`replacement_cost`	Calculate irreplaceability scores using the replacement cost method. This method is generally recommended for calculating irreplaceability scores.
`rarity_weighted_ric...`	Calculate irreplaceability scores using rarity weighted richness. This method is only recommended for particularly large-scale conservation planning problems where the replacement cost method would take too long to produce scores in a feasible period of time.

-

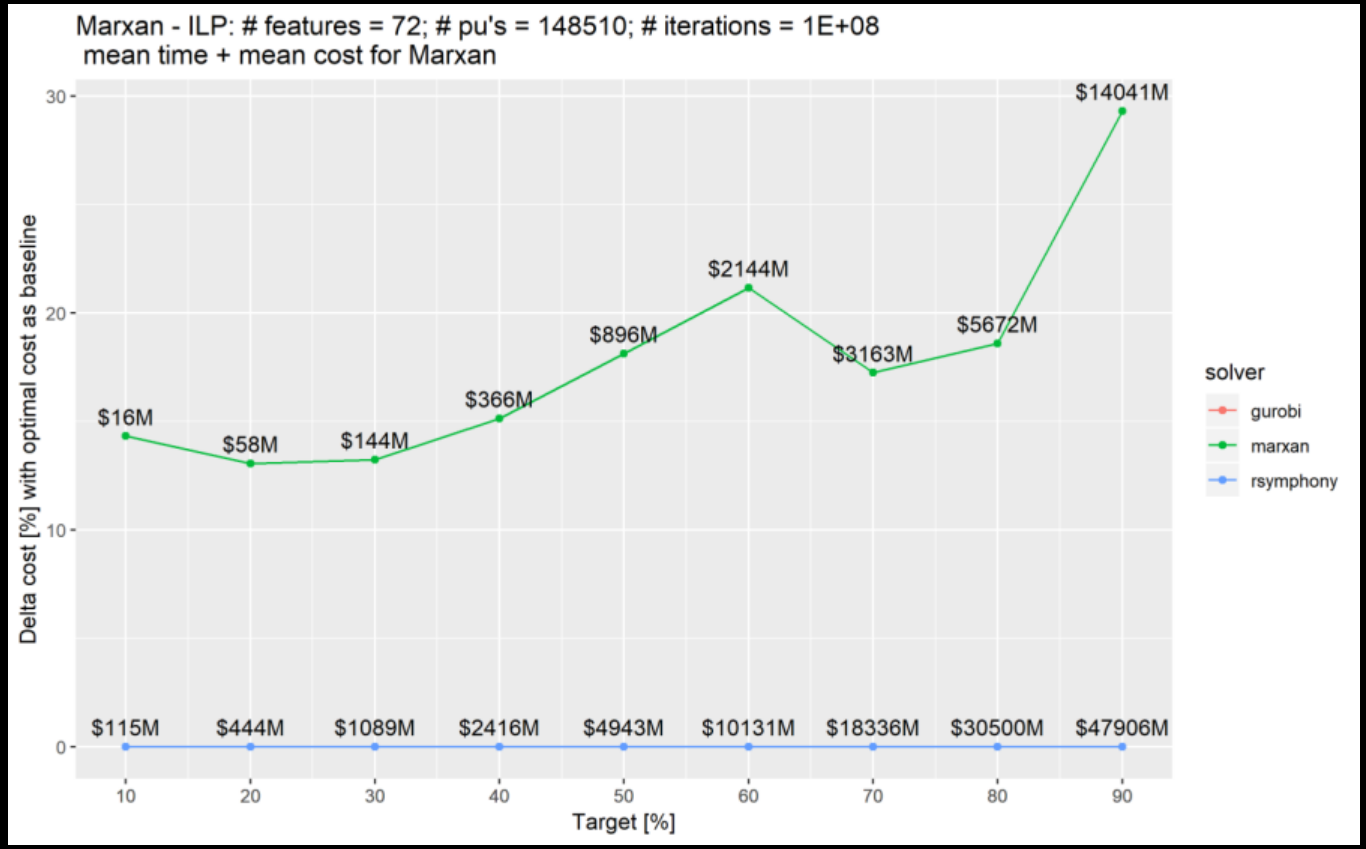
Solvers/ Solucionadores

https://prioritizr.net/reference/solvers.html

The following solvers can be used to find solutions for a conservation planning `problem`:

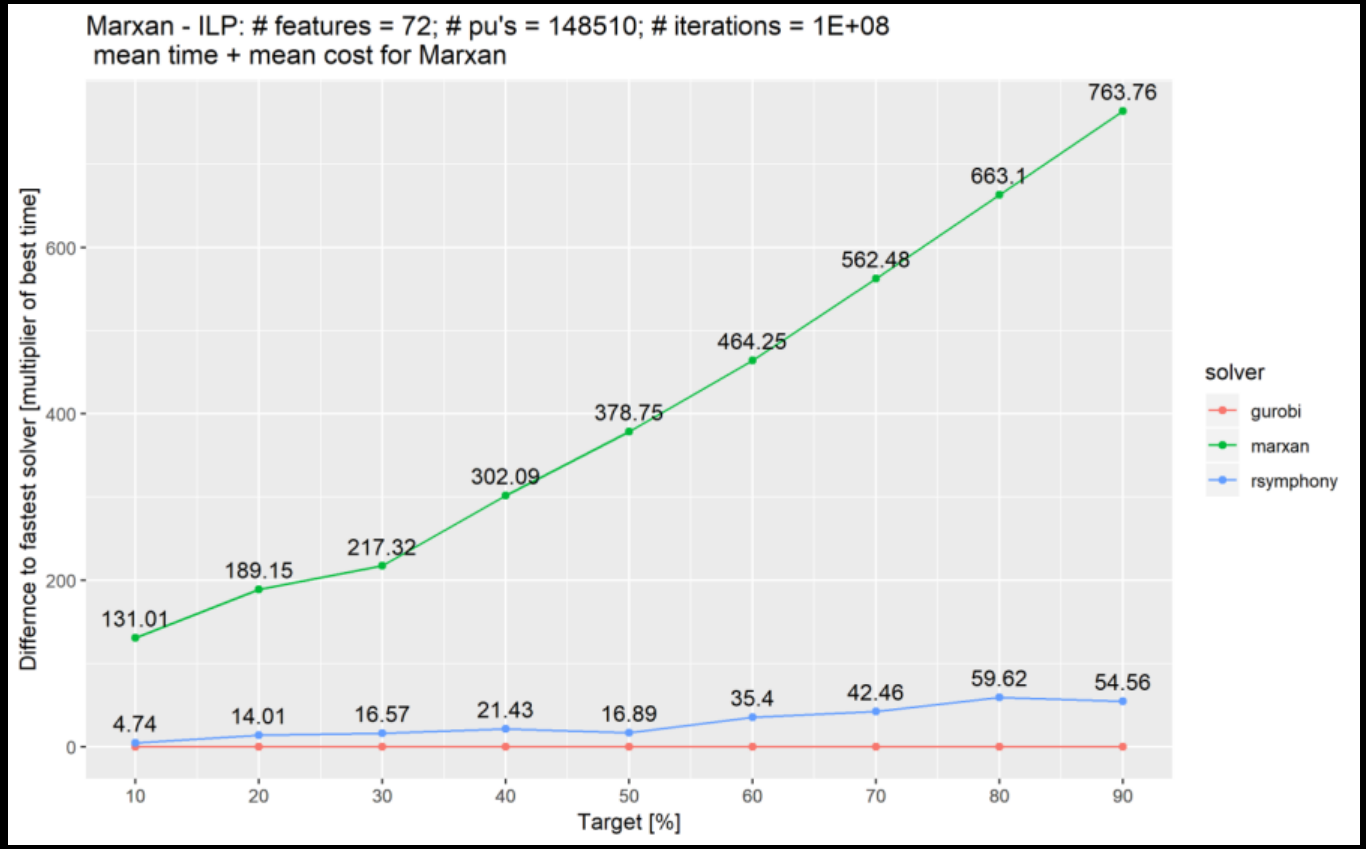
`add_default_solver`	This solver uses the best software currently installed on the system.
`add_gurobi_solver`	*Gurobi* is a state-of-the-art commercial optimization software with an R package interface. It is by far the fastest of the solvers available in this package, however, it is also the only solver that is not freely available. That said, licenses are available to academics at no cost. The gurobi package is distributed with the *Gurobi* software suite. This solver uses the gurobi package to solve problems.
`add_rsymphony_solver`	*SYMPHONY* is an open-source integer programming solver that is part of the Computational Infrastructure for Operations Research (COIN-OR) project, an initiative to promote development of open-source tools for operations research (a field that includes linear programming). The Rsymphony package provides an interface to COIN-OR and is available on CRAN. This solver uses the Rsymphony package to solve problems.
`add_lpsymphony_solv...`	The lpsymphony package provides a different interface to the COIN-OR software suite. Unlike the Rsymhpony package, the lpsymphony package is distributed through Bioconductor. The lpsymphony package may be easier to install on Windows or Max OSX systems than the Rsymphony package.

Exact algorithms guarantee optimality/ Algoritmos exactos garantizan la optimización



Schuster et al. (2020)

Exact algorithms are fast/ Los algoritmos son rápidos



Schuster et al. (2020)

Human-readable code/ Código entendible

Mental model/ Modelo mental

```
problem <- data +
           objective +
           constraints +
           penalties +
           decision type +
           solver

solution <- solve(problem)
```

Human-readable code/ Código entendible

Mental model/ Modelo mental

```
problem <- data +
           objective +
           constraints +
           penalties +
           decision type +
           solver


solution <- solve(problem)
```

Code/ Código

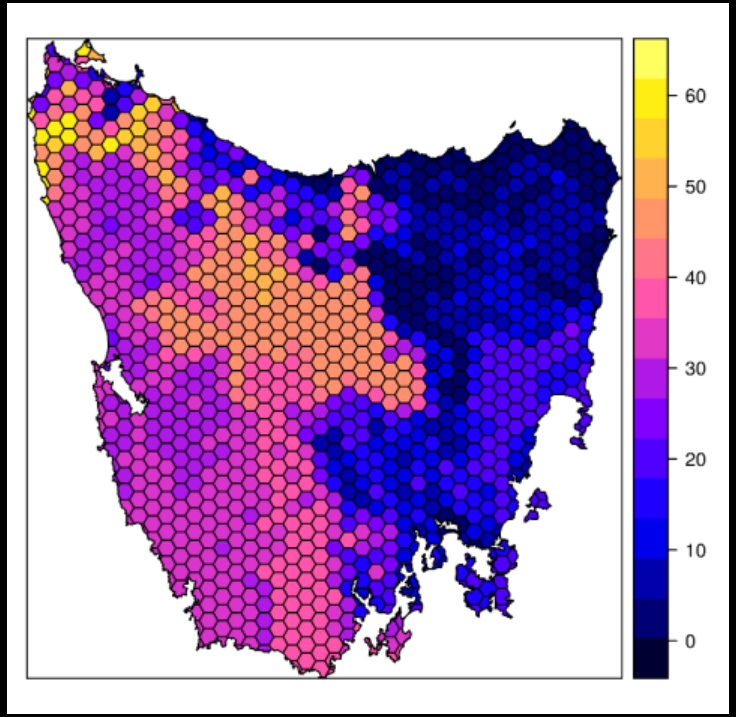
```
p <- problem(areas, feats) %>%
  add_min_set_objective() %>%
  add_relative_targets(0.1) %>%
  add_boundary_penalties(5) %>%
  add_binary_decisions() %>%
  add_rsymphony_solver()


solution <- solve(p)
```

Design your problem/ Diseño del problema

Study area/ Área de estudio: Tasmania, Australia

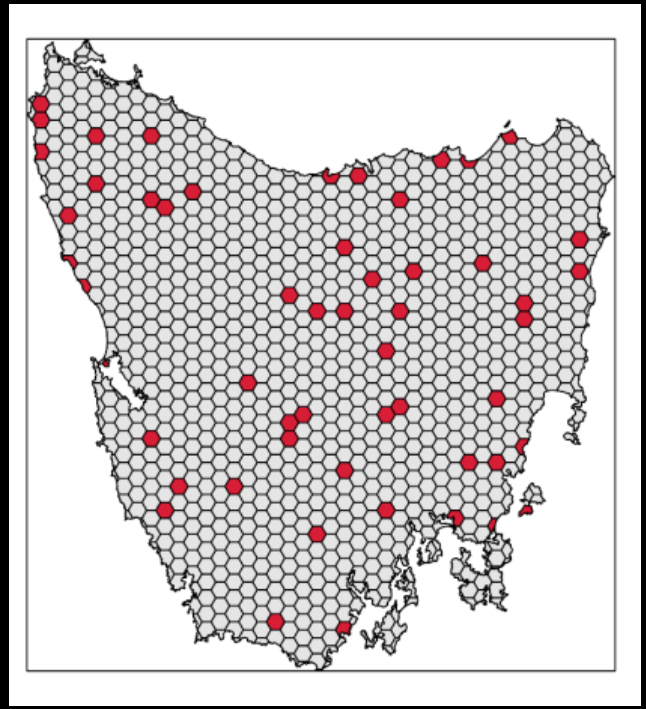
Planning units/ Unidades de planeación: 1130 hexagons

Features/ Características: 63 vegetation types/ Tipos de vegetación



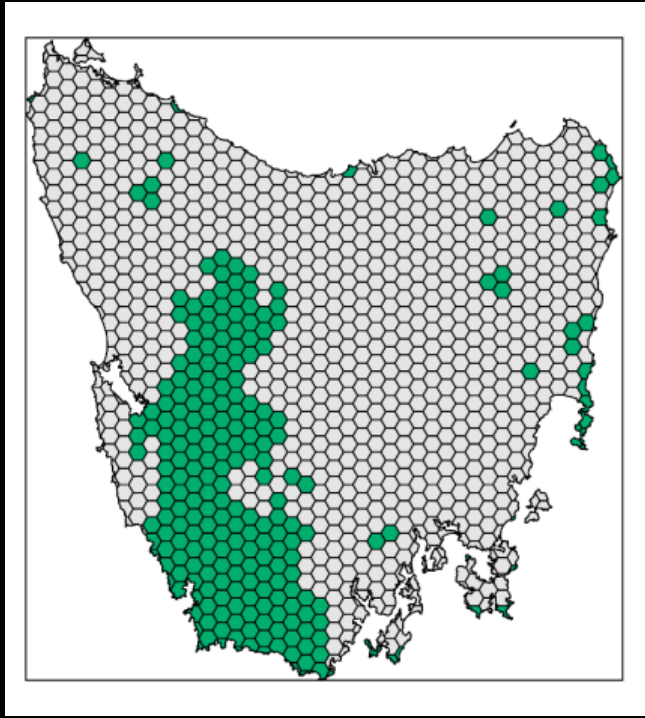
Design your problem/ Diseño del problema

```
problem(tas_pu, tas_features,
        "cost") %>%
add_min_set_objective() %>%
add_relative_targets(0.1) %>%
add_binary_decisions() %>%
add_gurobi_solver(gap = 0) %>%
solve()
```



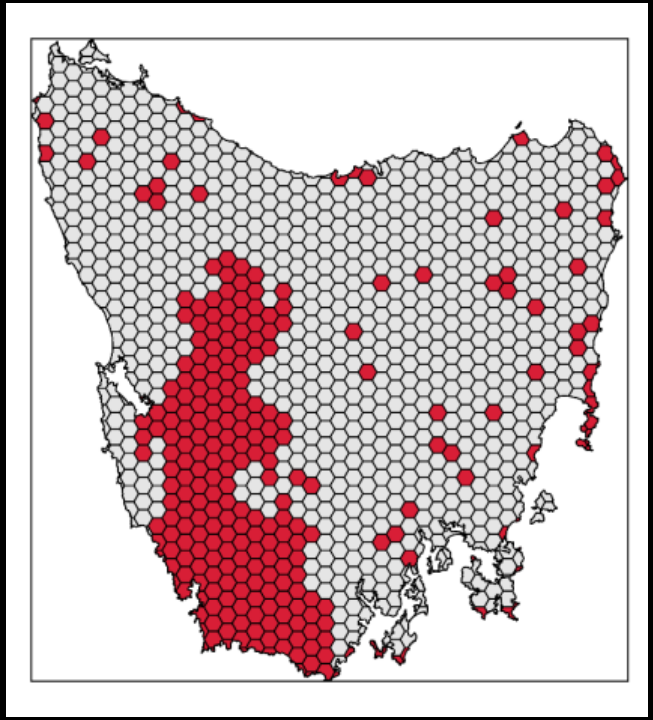
Design your problem/ Diseño del problema

```
problem(tas_pu, tas_features,
        "cost") %>%
add_min_set_objective() %>%
add_relative_targets(0.1) %>%
add_locked_in_constraints("in") %>%
add_binary_decisions() %>%
add_gurobi_solver(gap = 0) %>%
solve()
```



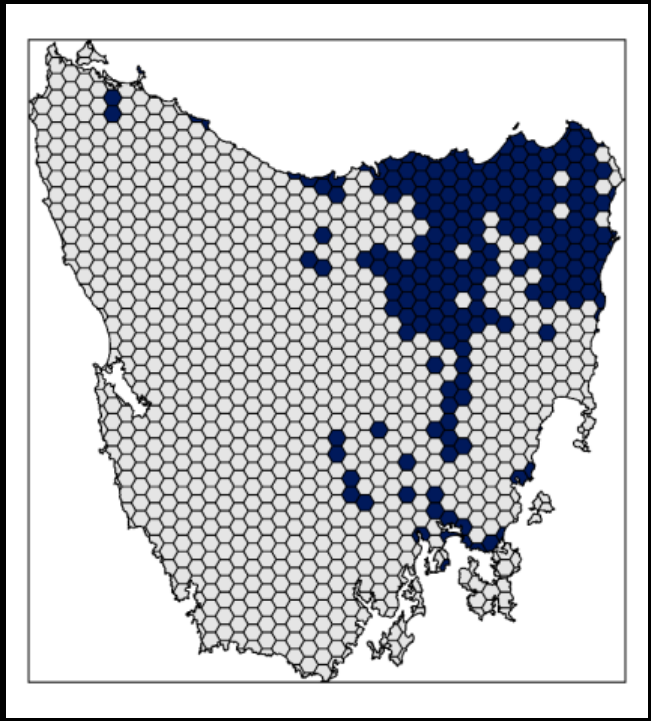
Design your problem/ Diseño del problema

```
problem(tas_pu, tas_features,
        "cost") %>%
add_min_set_objective() %>%
add_relative_targets(0.1) %>%
add_locked_in_constraints("in") %>%
add_binary_decisions() %>%
add_gurobi_solver(gap = 0) %>%
solve()
```



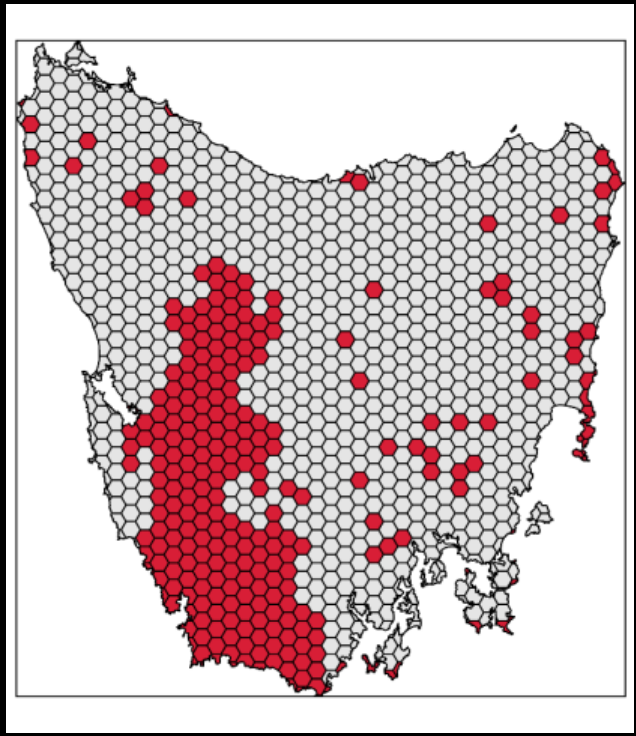
Design your problem/ Diseño del problema

```
problem(tas_pu, tas_features,
        "cost") %>%
add_min_set_objective() %>%
add_relative_targets(0.1) %>%
add_locked_in_constraints("in") %>%
add_locked_out_constraints("out") %>%
add_binary_decisions() %>%
add_gurobi_solver(gap = 0) %>%
solve()
```



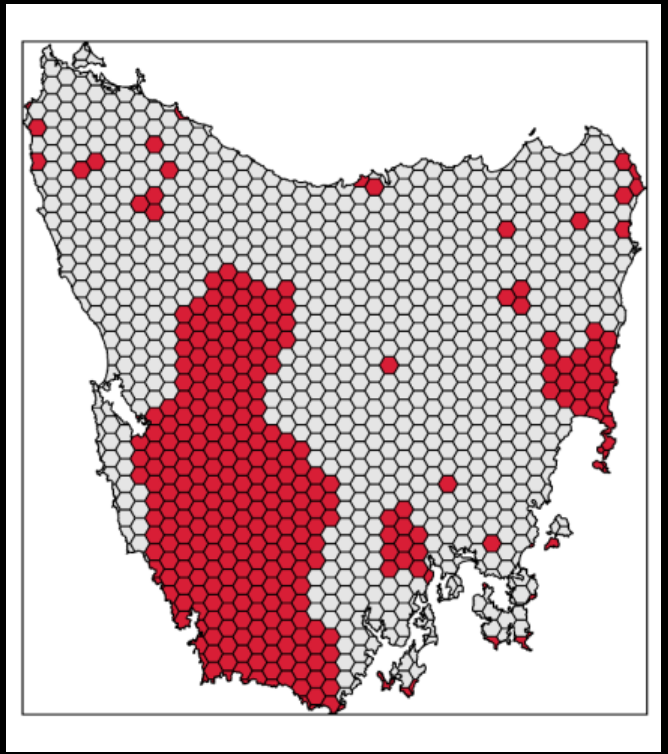
Design your problem/ Diseño del problema

```
problem(tas_pu, tas_features,
        "cost") %>%
add_min_set_objective() %>%
add_relative_targets(0.1) %>%
add_locked_in_constraints("in") %>%
add_locked_out_constraints("out") %>%
add_binary_decisions() %>%
add_gurobi_solver(gap = 0) %>%
solve()
```



Design your problem/ Diseño del problema

```
problem(tas_pu, tas_features,
        "cost") %>%
add_min_set_objective() %>%
add_relative_targets(0.1) %>%
add_locked_in_constraints("in") %>%
add_locked_out_constraints("out") %>%
add_boundary_penalties(0.01, 0.5) %>%
add_binary_decisions() %>%
add_gurobi_solver(gap = 0) %>%
solve()
```



Design your problem/ Diseño del problema

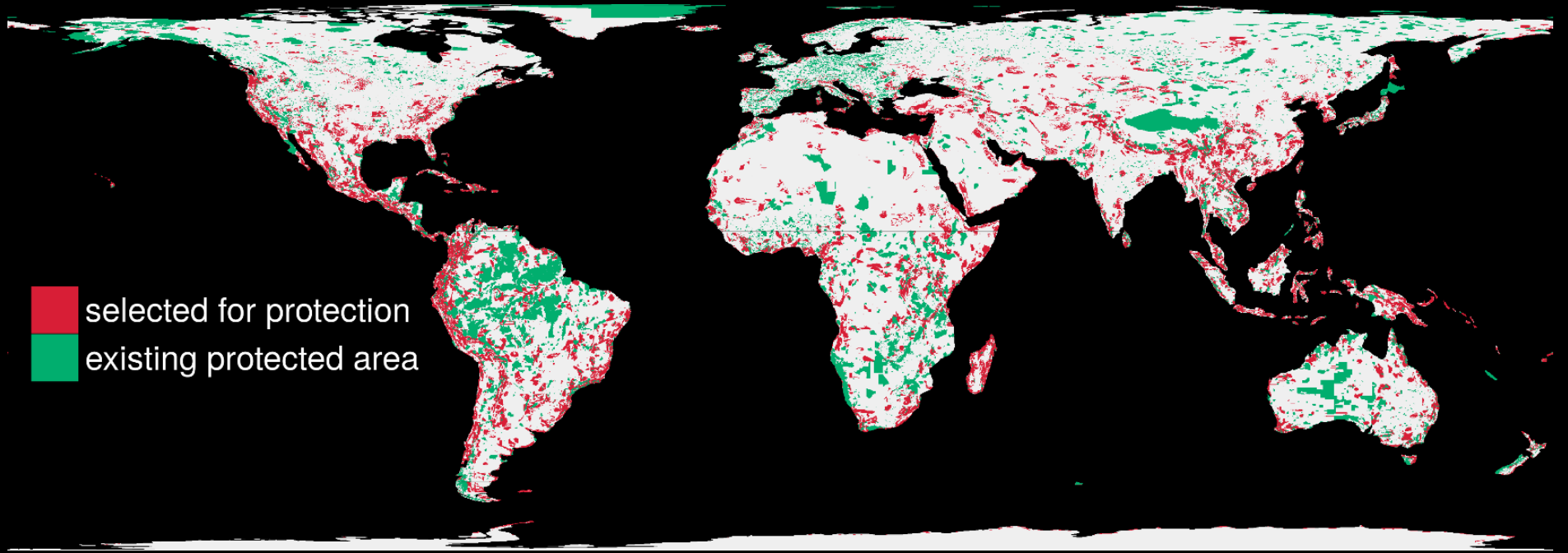
```
problem(pu, fts, "cost") %>%
add_min_set_objective() %>%
add_relative_targets(...) %>%
add_binary_decisions() %>%
solve()
```

Single zone Problem/ Problema de zona única

Multi-zone Problem/ Problema multizona

```
problem(pu, zones(fts_z1, fts_z2, ...),
                  c("cost_1", "cost_2", ...)) %>%
add_min_set_objective() %>%
add_relative_targets(...) %>%
add_binary_decisions() %>%
solve()
```

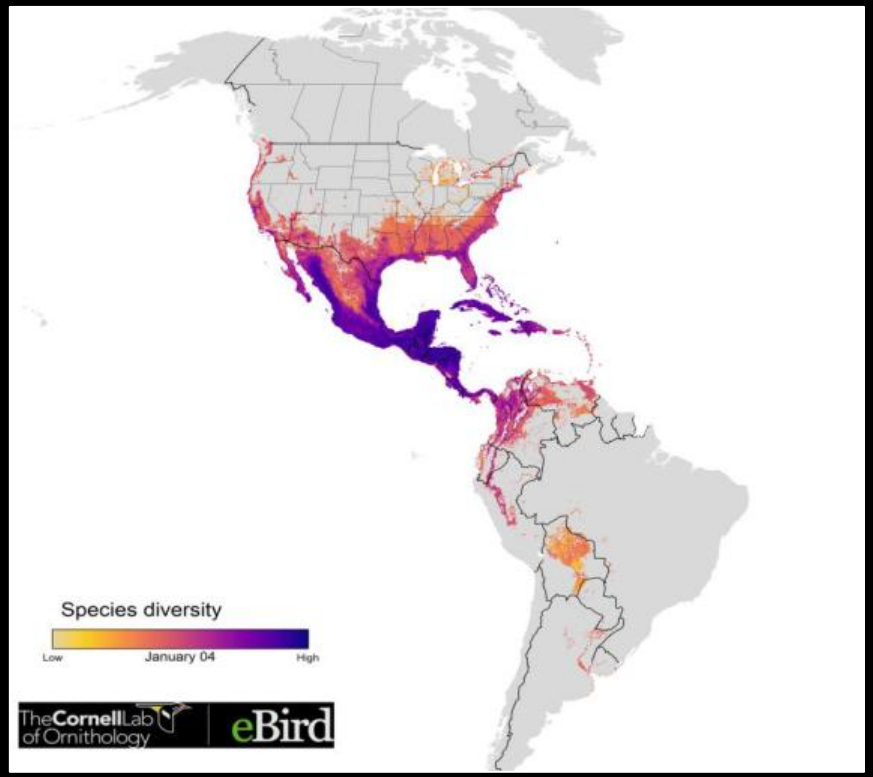
Solve it fast!/ ¡Resuélvelo rápido!



1.5 million planning units + 22,644 species (34 billion cells)/ 1.5 millones de unidades de planeación + 22,644 species (34 billones de celdas):

76 minutes/ 76 minutos

Optimizing the conservation of migratory species over their full annual cycle/ Optimización de la conservación de las especies migratorias durante su ciclo anual completo



117 species/ especies
73 million km^2
1.7 million unique locations/ ubicaciones únicas
14 million checklists/ listas de verificación

≤ 30,420 features/ caraterísticas
1.05 million planning units/ unidades de planeación

Analysis powered by:

Schuster et al. (2019) Nature Communications

Marxan

- ✓ Completely free/ Totalmente gratuito
- ✓ Lots of resources/ Muchos recursos
- Only minimum set/ Solo conjunto mínimo
- Solves a specific problem/ Soluciona un problema específico
- 100k planning units limit/ Límite de 100 mil unidades de planeación
- SPF calibration/ Calibración SPF
- Solution quality unknown/ Se desconoce la calidad de la solución
- Need to run 100+ times/ Se necesita ejecutar más de 100 veces
- Point-and-click/ Apuntar y hacer clic

prioritizr

- Free but you really want Gurobi/ Gratuito
- Documented (few papers yet)/ Documentado (pocos artículos)
- ✓ Many objective functions (7)/ Variedad de funciones objetivo
- ✓ Solves lots of different problems/ Soluciona gran variedad de problemas
- ✓ >5M planning units/ Más de 5 millones de unidades de planeación
- ✓ No SPF calibration/ Sin calibración SPF
- ✓ Solution quality guaranteed/ Se garantiza la calidad de la solucuión
- ✓ Run once/ Se corre una sola vez
- Minor programming skills/ Se necesita conocimientos mínimos de programación

The vision/ La visión

Flexible + powerful framework

+

User friendly (browser) interface

=

A novice stakeholder can devise a high-quality, data-driven spatial plan in just one hour.

prioritizr.net

Marco flexible y potente

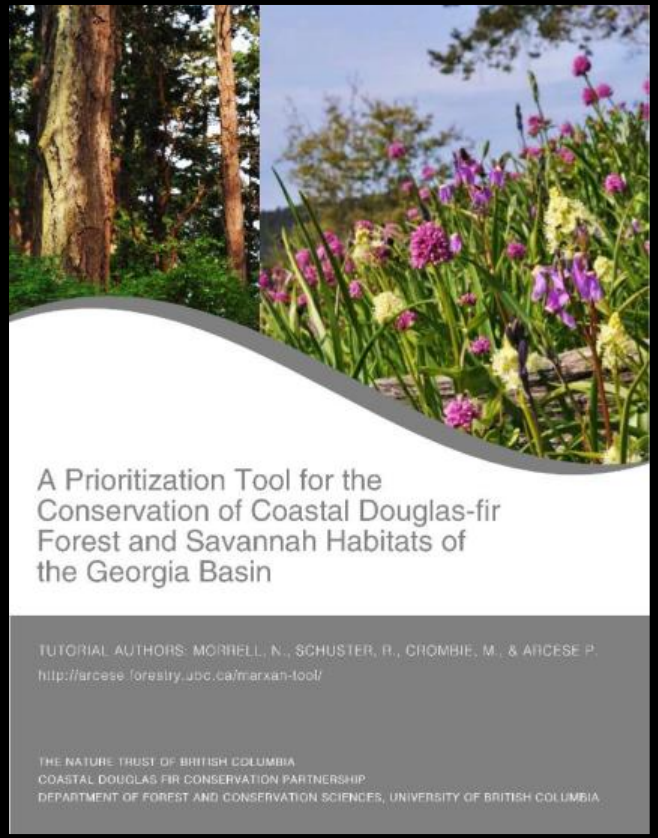
+

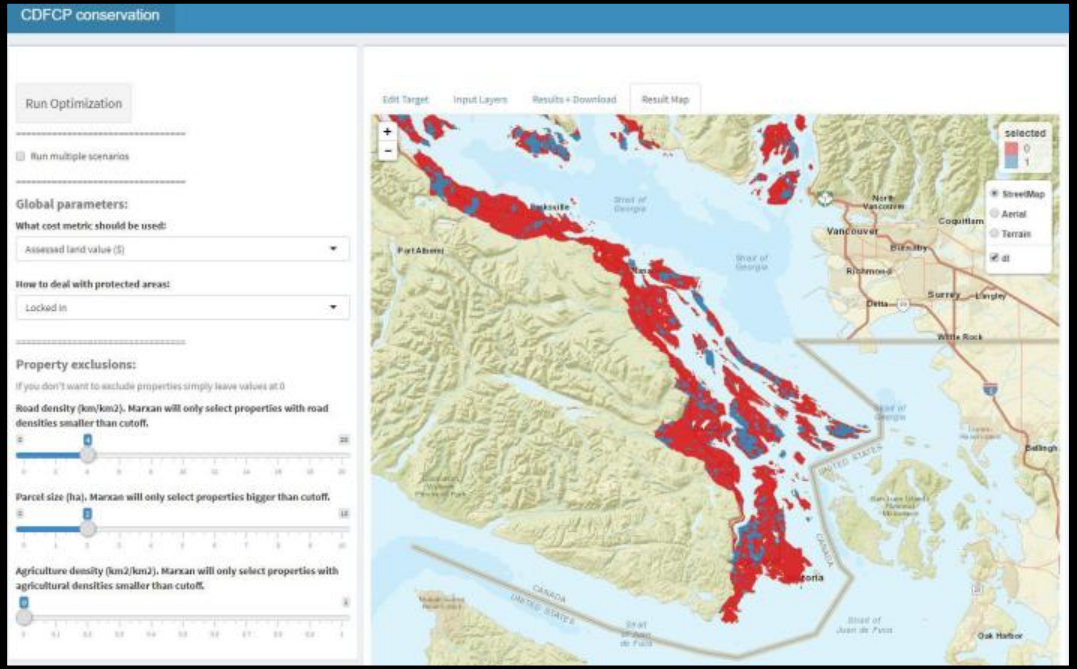
Interfaz amigable para el usuario (navegador)

=

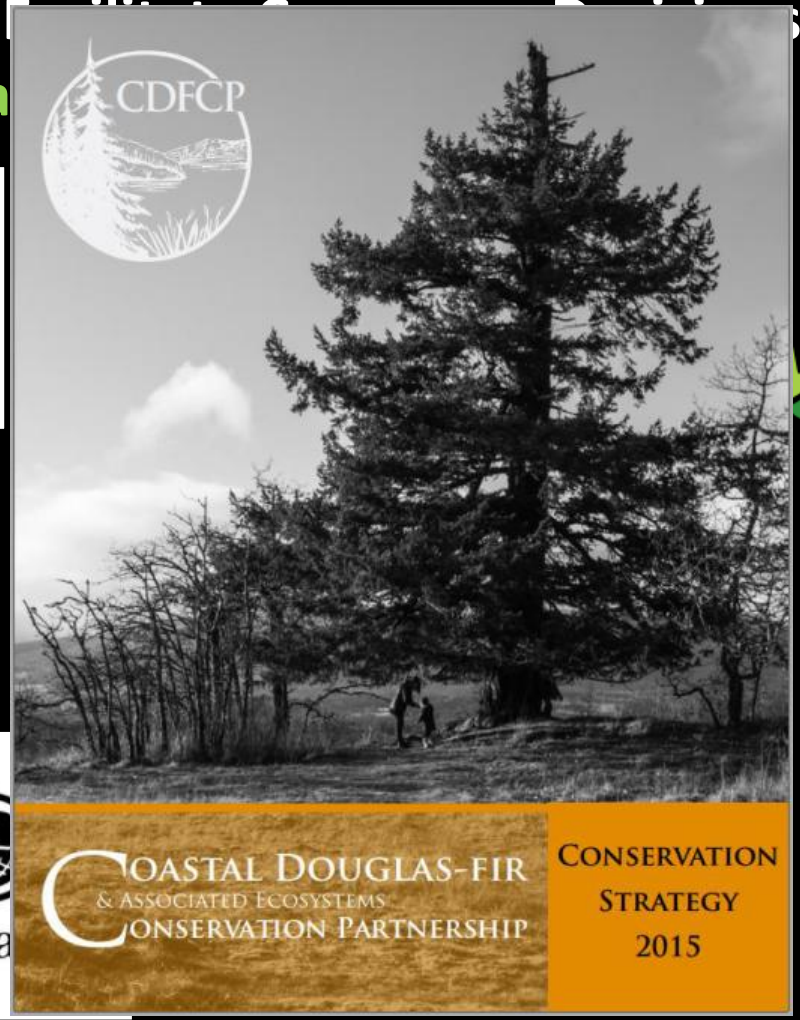
Una persona sin experiencia puede diseñar un plan espacial de alta calidad basado en datos en solo una hora.

Making prioritizr user friendly/ Haciendo que priorizr sea fácil de usar:





on Protection/ Fa sobre protección



BRITISH COLUMBIA
Ministry of Environment

BRITISH COLUMBIA
Ministry of Forests, Lands and Natural Resource Operations

NATURE CONSERVANCY CANADA

HAT
Creating Conservation Legacies

C·V·R·D

Climate Adaptive Planning for British Columbia/ Planificación de adaptación climática para Columbia Británica

- Recently started 3-year project/ Recientemente inició un proyecto de 3 años
- Leads/ Cabecillas: Oscar Venter and Peter Arcese
- Partners/ Socios:

Pacific Institute For Climate Solutions
Knowledge. Insight. Action

NATURE TRUST BRITISH COLUMBIA

BRITISH COLUMBIA Ministry of Environment

BRITISH COLUMBIA Ministry of Forests, Lands, Natural Resource Operations and Rural Development

CPAWS CANADIAN PARKS AND WILDERNESS SOCIETY NORTHERN ALBERTA

BC PARKS FOUNDATION

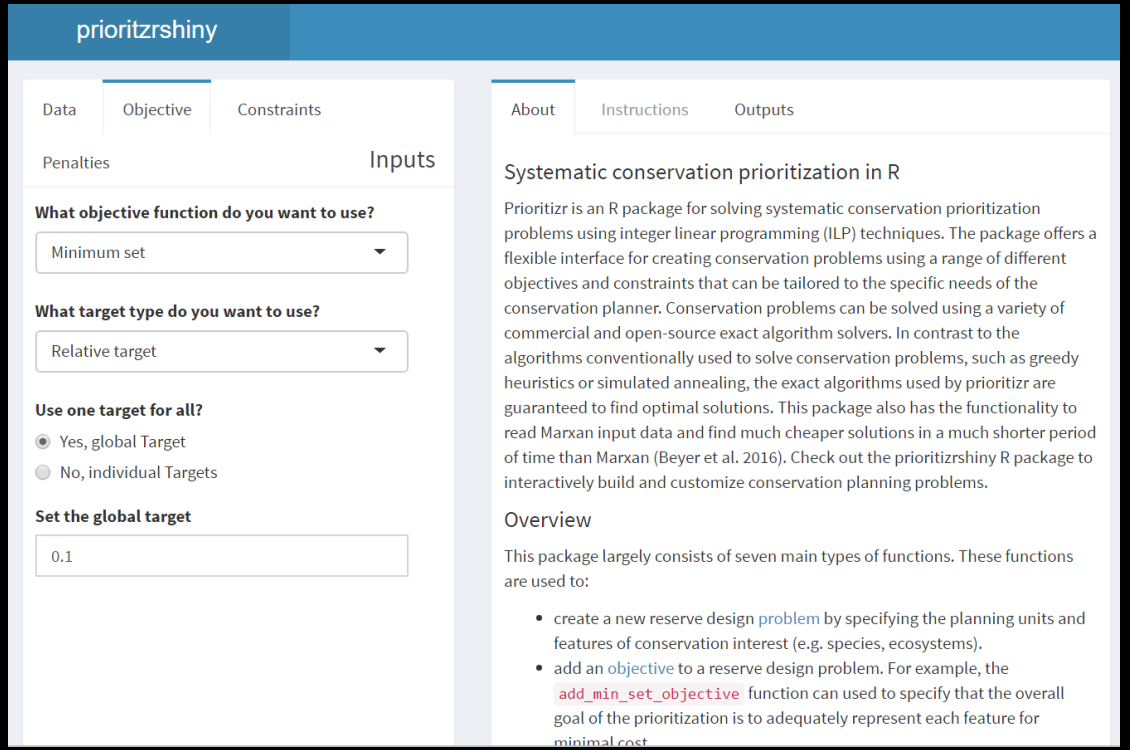
prioritizrshiny

(in development/ En desarollo)

Code/ Código

```
p <- problem(areas, feats) %>%
  add_min_set_objective() %>%
  add_relative_targets(0.1) %>%
  add_boundary_penalties(5) %>%
  add_binary_decisions() %>%
  add_rsymphony_solver()

solution <- solve(p)
```

User interface/ Interfaz de usuario



Take home message/ Mensaje deseado

Flexible + powerful framework

+

User friendly (browser) interface

=

A novice stakeholder can devise a high-quality, data-driven spatial plan in just one hour.

prioritizr.net

Marco flexible y potente

+

Interfaz amigable para el usuario (navegador)

=

Una persona sin experiencia puede diseñar un plan espacial de alta calidad basado en datos en solo una hora.

jeffrey.hanson@uqconnect.edu.au richard.schuster@glel.carleton.ca