Nimbus

An opensource library to implement random forests in genome-wide prediction

Oscar González-Recio Juanjo Bazán Selma Forni

62nd Annual Meeting EAAP 2011
August 29th – September 2nd
Stavanger NORWAY

Nimbus

The Problem

Massive amount of information from high throughput genotyping platforms.

… that consumes the attention of its recipients. We need to allocate that attention efficiently.

Need to extract knowledge from large, noisy, redundant, missing and fuzzy data.

Nimbus

The Problem

Massive amount of information from high troughput genotyping platforms.

Need to extract knowledge from large, noisy, redundant, missing and fuzzy data.

Massive amount of information consumes the attention of its recipients.
We need to allocate that attention efficiently.

Why Random Forest?

Machine Learning technique: we can extract hidden relationships that exist in these huge volumes of data; do not follow a particular parametric design.

Random Forest has desirable statistical properties.

Random Forest scales well computationally.

Random Forest performs extremely well in a variety of possible complex domains (Breiman, 2001; Gonzalez-Recio & Forni, 2011).

Nimbus

The Algorithm

It is an ensemble methods:

- Combines different models (usually simple models).
- They have very good predictive ability because use additivity of model performances.

Based on Classification And Regression Trees (CART).

Uses Randomization and Bagging.

Performs Feature Subset Selection.

Convenient for classification problems.

Fast computation.

Simple interpretation of results for human minds.

Previous work in genome-wide prediction (Gonzalez-Recio and Forni, 2011)

The Algorithm

Brief description

$$y = c_0 + c_1 f_1(\mathbf{y}, \mathbf{X}) + c_2 f_2(\mathbf{y}, \mathbf{X}) + \ldots + c_i f_i(\mathbf{y}, \mathbf{X}) + \ldots + c_M f_M(\mathbf{y}, \mathbf{X}) + \mathbf{e}$$

•Workflow

1. BOOTSTRAP THE DATA

   Take a bootstrapped set of *n* records of the original training set.

   Contains (aprox) 63% of the original data (some records appear more than once and other not at all)

   Around 37% of records are kept out of bag (OOB samples).

The Algorithm

Brief description

$$y = c_0 + c_1 f_1(\mathbf{y}, \mathbf{X}) + c_2 f_2(\mathbf{y}, \mathbf{X}) + \ldots + c_i f_i(\mathbf{y}, \mathbf{X}) + \ldots + c_M f_M(\mathbf{y}, \mathbf{X}) + \mathbf{e}$$

- **Workflow**

2. SELECT A SNP TO SPLIT THE DATA IN THREE NEW BRANCHES
   - Select *mtry* SNPs out of *p* at random.
   - Select the SNP *j* that minimizes a given loss function
     - Regression - Quadratic loss function
     - Classification - Gini coefficient

The Algorithm

Brief description

$$y = c_0 + c_1 f_1(\mathbf{y}, \mathbf{X}) + c_2 f_2(\mathbf{y}, \mathbf{X}) + \ldots + c_i f_i(\mathbf{y}, \mathbf{X}) + \ldots + c_M f_M(\mathbf{y}, \mathbf{X}) + \mathbf{e}$$

- **Workflow**

3. SPLIT THE NODE

Create three new branches according to the genotype of SNP *j* that one individual may or may not have.
i.e. 0,1,2

The Algorithm

Brief description

$$y = c_0 + c_1 f_1(\mathbf{y}, \mathbf{X}) + c_2 f_2(\mathbf{y}, \mathbf{X}) + \ldots + c_i f_i(\mathbf{y}, \mathbf{X}) + \ldots + c_M f_M(\mathbf{y}, \mathbf{X}) + \mathbf{e}$$

•Workflow

4. GROW TREE

Repeat steps 2-4 until a minimum size (e.g. <5) is reached or L does not improve

Estimated phenotype (label) is the average phenotype of individuals in the terminal node (Regression) majority vote of individuals in the terminal node (Classification)

Estimates of yet to be observed records are calculated as:

Pass the genotype *i* through the tree until reach a terminal node. The estimate for individual *i* is the corresponding label to the terminal node reached

The Algorithm

Brief description

$$y = c_0 + c_1 f_1(\mathbf{y}, \mathbf{X}) + c_2 f_2(\mathbf{y}, \mathbf{X}) + \ldots + c_i f_i(\mathbf{y}, \mathbf{X}) + \ldots + c_M f_M(\mathbf{y}, \mathbf{X}) + \mathbf{e}$$

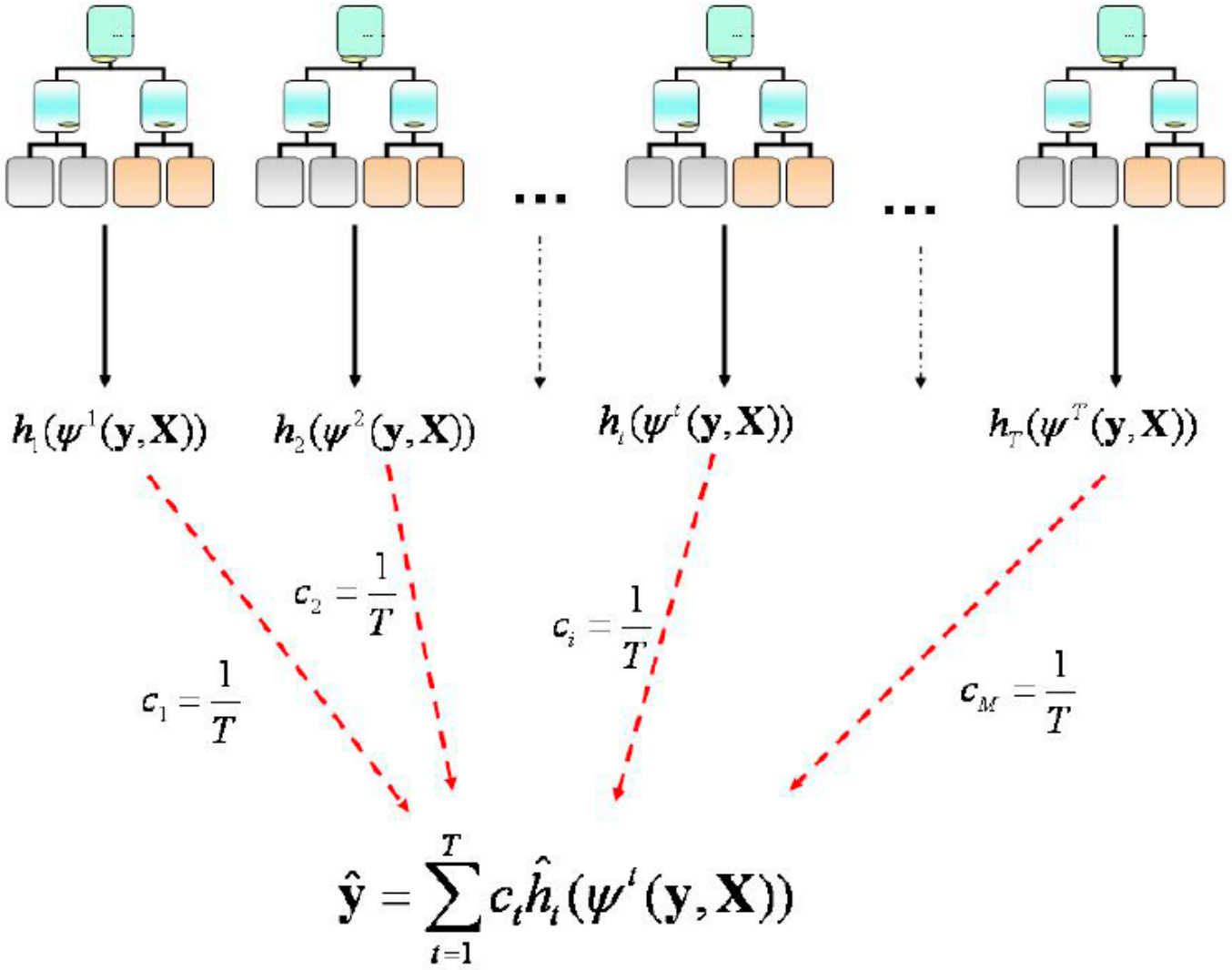
- **Workflow**

5. GROW FOREST

Repeat steps 1-5 until a large number of times. Average estimates across trees to make final predictions

Nimbus

The Algorithm



Nimbus library

Nimbus library

Written in Ruby

Open source programming language

Syntax focused on simplicity

Natural to read and easy to write

www.ruby-lang.org

Nimbus

Nimbus library

Prerequisites:

Ruby and Rubygems (default library manager) installed in the system

How to install:

> gem install nimbus

Nimbus

Nimbus library

How to run:

> nimbus

Configuration (parameter file):

Via *config.yml* file

Nimbus library

config.yml file:

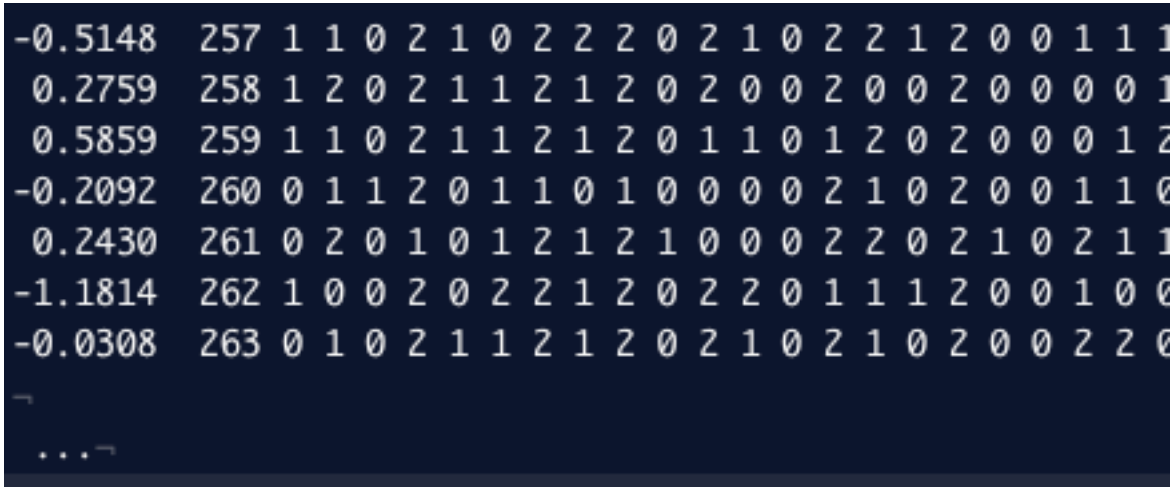
```
#Input files
input:
  training: regression_training.data
  testing:  regression_testing.data
  forest:   regression_random_forest.yml

#Forest parameters
forest:
  forest_size: 3 #how many trees
  SNP_sample_size_mtry: 60 #mtry
  SNP_total_count: 200
  node_min_size: 5
```

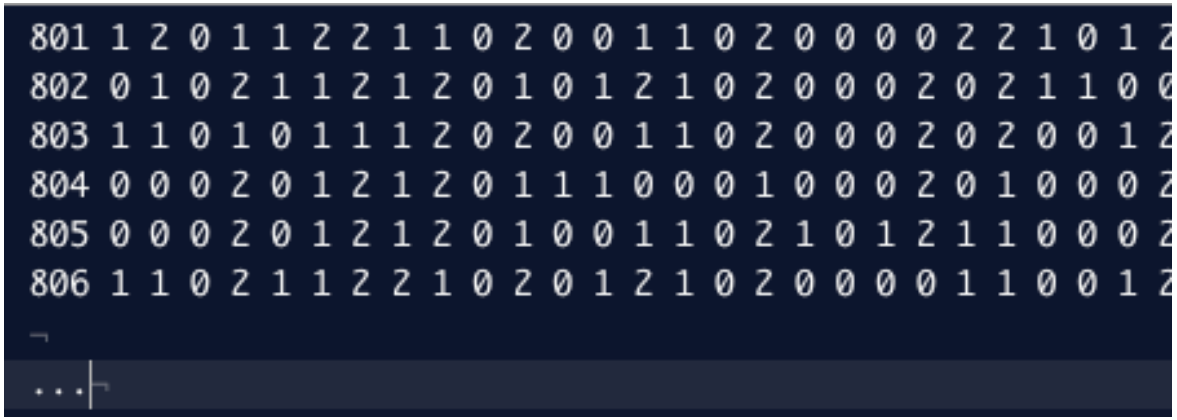
Nimbus

Nimbus library

Input files:



training



testing

Nimbus

Nimbus library

Use cases:

Training sample file

- Nimbus creates a reutilizable forest
- Estimate genomic merit for each individual
- Generalization error are computed for every tree in the forest
- Nimbus calculates SNP importances

Previously saved forest

- Use this forest to predict genomic value of individuals in the testing set

Testing sample file

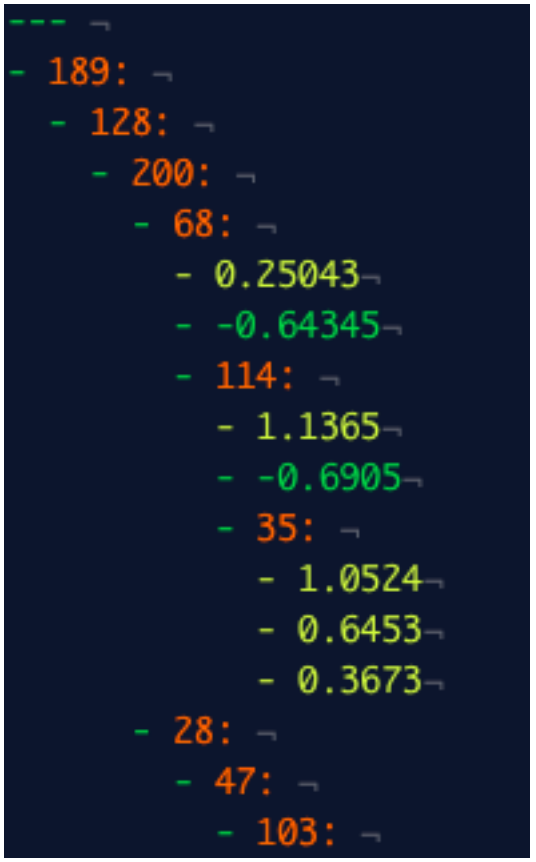
- Provide genomic predictions for each individual
- Using new or new forest, specified via *config.yml*

Nimbus

Outputs

Random Forest file

In standard YAML format, that allows storing data structures

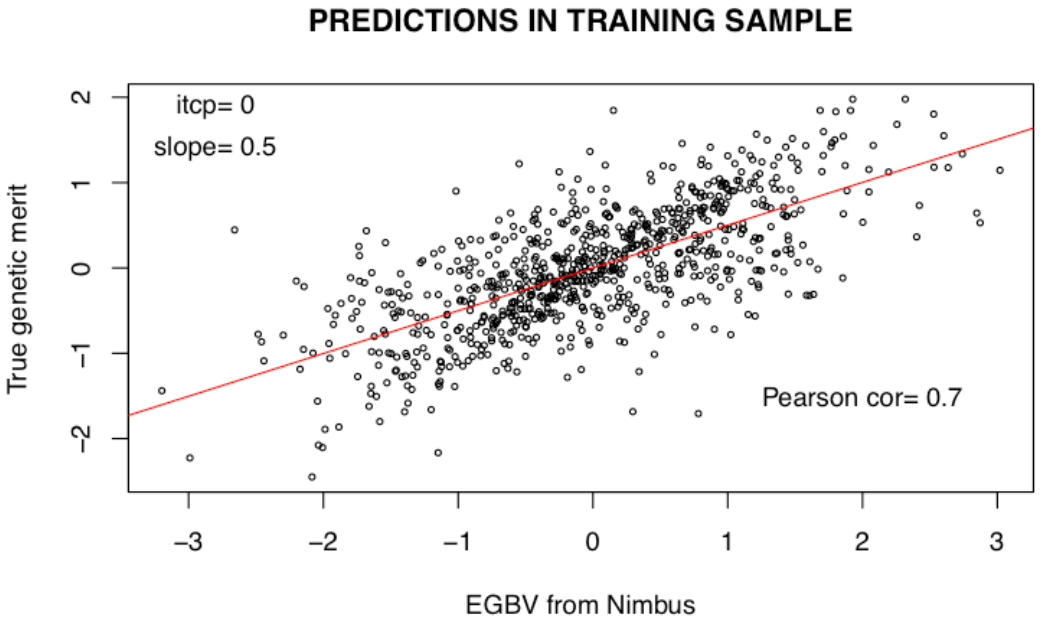


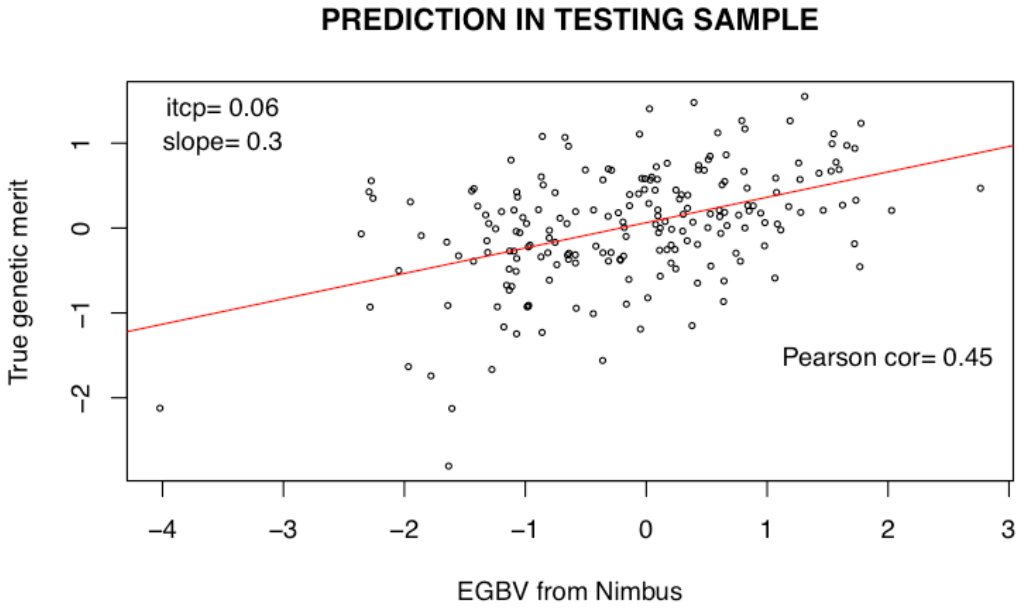
May be used as input file
to predict GBV in a testing set

Nimbus

Outputs

Estimated genomic breeding values (training and testing)

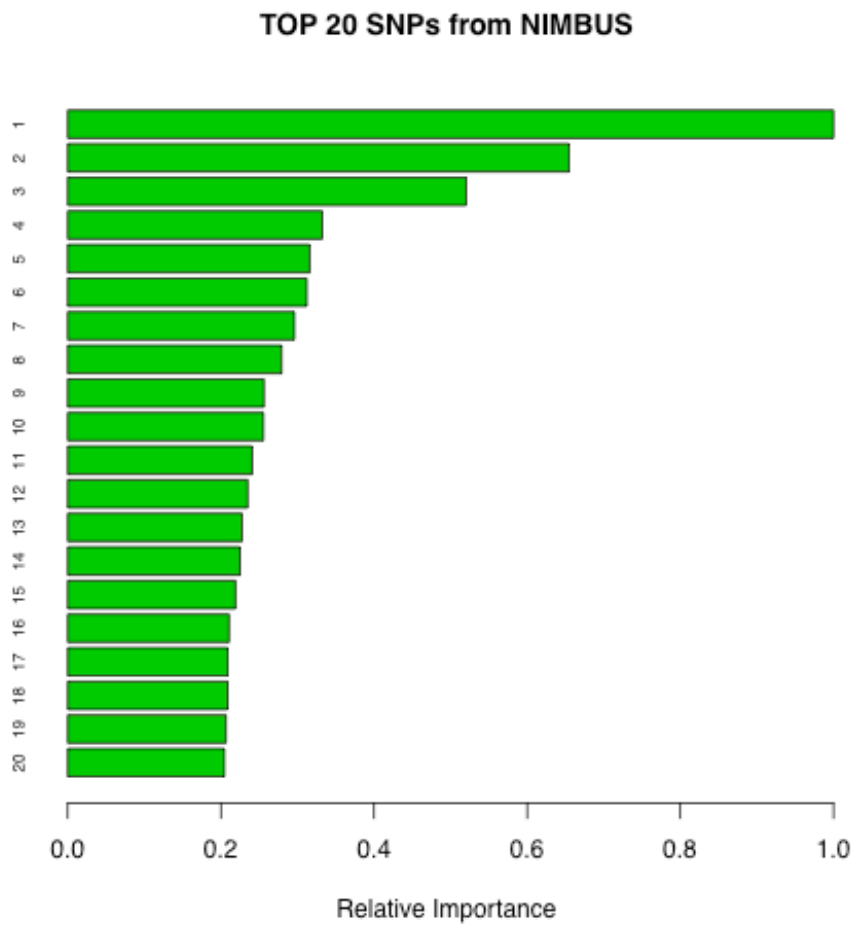


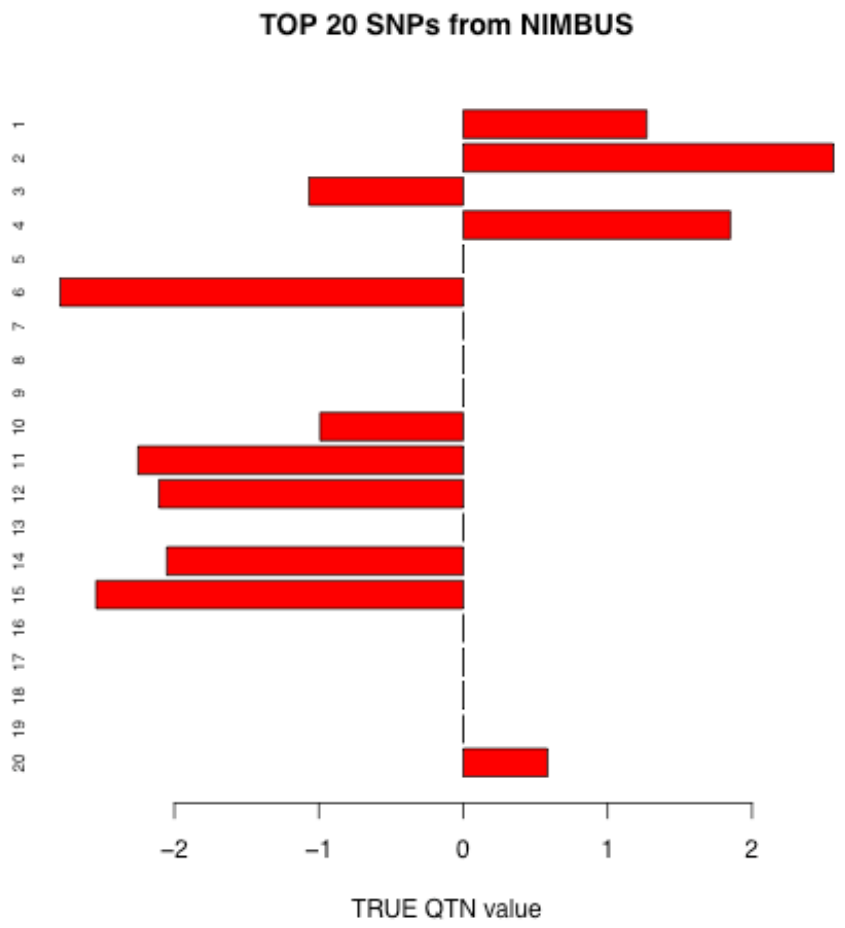


Nimbus

Outputs

SNP importances





Nimbus

More info:

Nimbus website:

www.nimbusgem.org

Source code:

www.github.com/xuanxu/nimbus

Report bugs/request features:

www.github.com/xuanxu/nimbus/issues

Nimbus

Thank you!

Nimbus

Questions?