Proposal: `Method#source_location` or `Method#owner` for refined methods

I'm working on a gem with Ruby refinements, and some methods are invented in new Ruby versions like 2.5 or 3.0, and I'd like to define in this gem should I refine a method or not.

So… current behavior of Method#source_location returns nil for core and/or refined methods. I think, refinements should be caught by this method. Also I didn't know about Method#owner, but it can be a case for refinements too.

---

#1 - 03/10/2021 01:31 PM - osyo (manga osyo)

hi.

In Ruby 2.7 and later, #method supports Refinements.

```ruby
# test.rb
module Cover
  refine Range do
    def cover?(value)
      return super unless value.is_a?(Range)
      super(value.first) && super(value.last)
    end
  end
end
using Cover

pp (1..10).method(:cover?).source_location
# Ruby 2.6 => nil
# Ruby 2.7 => ["/test.rb", 4]

pp (1..10).method(:cover?).owner
# Ruby 2.6 => Range
# Ruby 2.7 => #<refinement:Range@Cover>
```

see: https://bugs.ruby-lang.org/issues/15373

---

#2 - 03/12/2021 06:18 PM - AlexWayfer (Alexander Popov)

osyo (manga osyo) wrote in #note-1:

```
hi.

In Ruby 2.7 and later, #method supports Refinements.

```ruby
# test.rb
module Cover
  refine Range do
    def cover?(value)
      return super unless value.is_a?(Range)
      super(value.first) && super(value.last)
    end
  end
end
using Cover

pp (1..10).method(:cover?).source_location
# Ruby 2.6 => nil
# Ruby 2.7 => ["/test.rb", 4]
```
Can't confirm:

```ruby
# frozen_string_literal: true

puts RUBY_VERSION
p (1..10).method(:cover?).source_location
```

```ruby
~/Projects/ruby/test/source_location
> ruby test.rb
3.0.0
nil
~/Projects/ruby/test/source_location
> rbenv local 2.7.2
~/Projects/ruby/test/source_location
> ruby test.rb
2.7.2
nil
```

I'd like to attach a link to something like repl.it, but there is only Ruby 2.5.

#3 - 03/13/2021 06:42 AM - xtkoba (Tee KOBAYASHI)
The example code in #note-2 does not seem to refine Range#cover?. This method is defined natively in range.c, and so the behavior is expected (Method#source_location should return nil).

#4 - 03/15/2021 10:37 PM - AlexWayfer (Alexander Popov)
xtkoba (Tee KOBAYASHI) wrote in #note-3:

> The example code in #note-2 does not seem to refine Range#cover?. This method is defined natively in range.c, and so the behavior is expected (Method#source_location should return nil).

Sorry, you're correct, I didn't understand the code from #note-1 correctly.

(I don't see an ability to close this issue by myself)

#5 - 03/15/2021 11:15 PM - jeremyevans0 (Jeremy Evans)
- Status changed from Open to Closed