It can be important to get the only element for which a block returns true, and to assert that this is the only element as such. For example, this can be a very helpful sanity check when one is working with data that's outside of their control and is not perfectly understood. They may have a guess as to how to get a specific element matching some criteria, but if they used Enumerable#detect might be hiding the fact that they have written an incorrect block and that there's in fact more than one element that matches it. It could also be a parameter on Typically, I'd do it like this:

```ruby
matches = array.select { |elem| some_method(elem) }
raise if matches.size != 0
match = matches.first
```

Here, it would be shortened to:

```ruby
match = array.detect_only { |elem| some_method(elem) }
```

It could also be a parameter on Enumerable#detect instead of a separate method.

---

**Related issues:**

- Related to Ruby master - Feature #13683: Add strict Enumerable#single
Do you mean raise if matches.size != 1 instead of ... != 0?

The idea of take_only is already proposed in #13683.

#6 - 09/02/2021 05:44 PM - meisel (Michael Eisel)
Yes, I meant ... != 1, thanks. And I think we can close this issue, and I'll add my thoughts to the issue that this dupes.