CORE FUNCTIONALITY THAT'S MISSING

In update, $pull can't take a selector like {$gt: 3} (but it can take {x: 3}, or {x: {$gt: 3}} -- basically, selectors that match documents can be used, but selectors that are intended to match non-document values won't work.)

Sort does not correctly implement the Mongo behavior where only array entries that match the query's selector are used as sort keys. Specifically, when sorting a document with sort specifier {a:1}, the document {a: [5, 10]} will usually have sort key 5 (the minimum is used for ascending sorts and the maximum for descending sorts). But if the query's selector is {a: {$gt: 7}}, then Mongo will actually use 10 as the sort key because 5 does not match.

Unsupported selectors:

- $elemMatch inside $all
- geoqueries other than $near ($nearSphere, $geoIntersects, $geoWithin)

In MongoDB, a query with $near that matches a document which has multiple points matching the key (eg, `c.find({$near: 'a'})` with the document `{a: [[1,1], [2, 2]]}` can return the document multiple times (perhaps not even consecutively). In minimongo, queries never return a document multiple times.

ON TYPES

We don't implement these Mongo types completely: timestamp (but date works), symbol, javascript code (with or without scope), minkey/maxkey, regexp (stored in the database), fixed-precision integers.

If your Javascript implementation enumerates the keys of objects in a consistent order, then we implement object equality and object comparison in the same way that Mongo does it (defined relative to the key order in the objects.) If your JS implementation doesn't keep the keys of objects in order (or you choose to consider its behavior as undefined), then object equality and comparison is undefined in your mongo queries.

In update, we don't support $bit (because $bit only works on the integer type, and we don't support the integer type yet.)

API

find() doesn't support the min and max parameters.

findAndModify isn't supported.

The aggregate functions distinct(), and group() aren't supported. Map/reduce isn't supported.

update() should have a clear stance on atomicity (both in terms of multiple ops on a single document, and on multi-document update mode.) It just hasn't been looked at/thought about yet.

In general, the API needs tests, especially update. (On the other hand, the underlying selector and mutator code is quite well tested.)

OTHER STUFF

We ignore the 'x' and 's' flags on regular expressions.

"Natural order" isn't very well defined.

We don't support capped collections.

Little performance optimization has been done. In particular, there are no indexes.