GBrowse 3.0

Ian Holmes, Mitch Skinner *et al*
Berkeley

Live demo: *(muahahahahahahaaaaaaa!!!)*

http://genome.biowiki.org/

GBrowse3.0 = Web2.0

- **The AJAX experience: fluid, playful**
  - No more page-based views! *(less functionality, for now)*
  - Smooth panning & zooming
  - Drag-and-drop track selection, ordering
- **Community annotation**
  - Upload, revise tracks *("genome wiki")*
  - Comments, ratings, messaging, etc. *(social net.app.)*
- **Philosophy:** within the parameters laid down by application requirements, **it should be fun**

Architectural principles

- **Javascript client downloads track from server**
  - builds Nested Containment List
- **JS client does layout dynamically**
  - or, density-histogram at min.zoom
- **Features represented as DIVs/images**
  - enables “live zoom” by resizing parent div
- **Tracks can also be pre-rendered images**
  - e.g. quantitative tracks
  - just-in-time server-side rendering also possible
- **“Lazy” wherever possible (e.g. panning)**

Nested Containment Lists

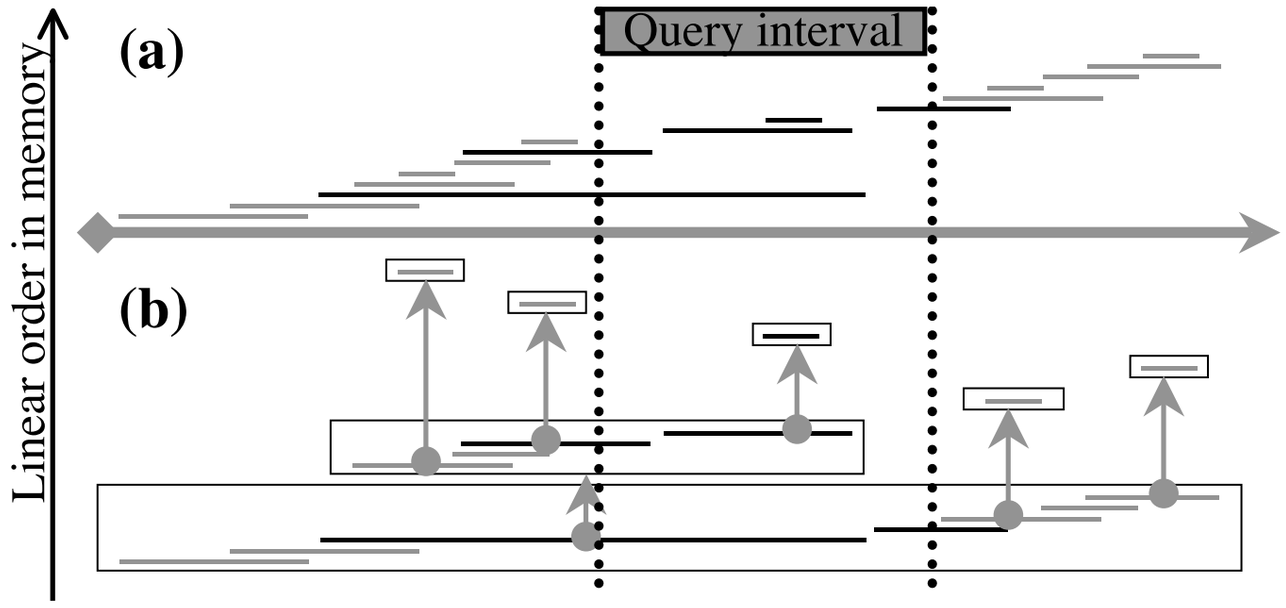

Fig. 1. Storage and querying for interval overlap. In (a), we demonstrate that using conventional sorting of interval database by start, end coordinates the interval overlap query cannot halt at first non-qualifying interval (see text), but has to continue until the end/start of the database, scanning on average half of the stored intervals. In NCList (b), however, the result set intervals (in black) are located back to back in each of the sublists (each sublist is shown in a separate box) and sublists potentially containing more results are linked to each other by containment links. Therefore, the scanning problem is eliminated in this case, resulting in lower query complexity.

Within each sublist, there are no containments
Sorting by startpoint leaves endpoints sorted too

Query: ***O(n+logN)***
N=database
n=number of results

5-500x faster than
R-trees, MySQL B-trees
(indexing/binning),
Ensembl, UCSC...

Construction: ***O(N)***

Alekseyenko & Lee
Bioinformatics, 2007

Limitations of Javascript

- Ultimately, fewer insurmountables than you'd think
- Ian: "the browser will never handle that many DIVs"
  - Mitch: test-based refutation
    (evolved into current code base)
- Limited graphics; can't plot lines/shapes
  (DIVs, overview trapezoids are nice hacks...)
- There may be others, but not so many...

modENCODE

- **Wide-exposure demo needed**
  ...to sufficiently sample feature requests
- **Exemplary** in several ways:
  - Dense tracks, quantitative tracks
  - Community annotation; "gatekeeping"
- **We** (the browser developers)
  are also users (of modENCODE)

modENCODE goals

- **Completed** (live at genome.biowiki.org)
  - Panning, zooming; location overview
  - Drag & drop track selection; persistence via cookies
- **Ongoing** (needed for "minimal" modENCODE functionality)
  - Click-thru to gene page (trivial)
  - Subfeature aggregation, i.e. exon structure (nontrivial)
  - Quantitative/WIG tracks (not too bad)
  - Search by gene name (pending)
- **Upcoming** (beyond minimal modENCODE functionality)
  - Basically, everything else (more UI, wiki, etc.)

Approx. schedule?

- **2008:** *A Lightweight AJAX Genome Browser*
- **2009:** *An AJAX Genome Wiki*

Path to Genome Wiki

- **Again, aim to identify minimum functionality**
  - probably also for modENCODE
- **Need for user accounts (login, register, etc.)**
  - ...so that uploaded tracks can be managed
  - ...also, to distribute the "metadata-collection burden"
  - first steps: Catalyst framework (...DAS?)
- **Examples of things that are NOT minimal:**
  - drag-to-edit features (creep towards curation tool)
  - editing of gene pages (belongs in MediaWiki)

Acknowledgements

- Code: **Mitch Skinner**
- Prototype: **Andrew Uzilov** (+Ian Holmes)
- Idea: **Chris Mungall**
- Steering: **Lincoln Stein**
- Kibitzing: **Ian Holmes**

- Funding: **NHGRI**