Some thoughts on designing a genomic query language

Limsoon Wong

Short talk for GeCo Workshop, Como, March 2019
Two perspectives on a query language

**Surface syntax**
- Easy to read, understand, & write queries correctly
- Sufficient power to express needed *queries*
- Prevent expensive queries

**Abstract syntax**
- Easy to analyze, manipulate, and optimize
- Easy to cater for extensions
- Sufficient power to express needed *algorithms*

**Compositionality & orthogonality** are key principles for query language design

\[
\{ \{ x, z \} \mid (x, y) \in A, (y', z) \in B, y = y' \} \\
\text{select } (a.x, b.z) \\
\text{from } a \text{ in } A, b \text{ in } B \\
\text{where } a.y = b.y' \\
U \{ U \{ \text{if } a.y = b.y' \} \\
\text{then } \{ (a.x, b.z) \} \text{ else } \{ \} \\
\mid b \in B \} \\
\mid a \in A \}
\]
Compositionality & orthogonality in NRC

Types

\[ s, t ::= \mid \text{bool} \mid b \mid s \times t \mid \{s\} \]

Expressions, constructs are provided for each type orthogonally

\[ x^s : s \quad \frac{e_1 : s \quad e_2 : t}{(e_1, e_2) : s \times t} \quad \frac{e : s \times t}{\pi_1 e : s \quad \pi_2 e : t} \]

\[ \text{true} : \text{bool} \quad \text{false} : \text{bool} \quad \frac{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 : s}{e_1 : \text{bool} \quad e_2 : s \quad e_3 : s} \]

\[ \{\}^s : \{s\} \quad \{e\} : \{s\} \quad \frac{e : s}{e_1 \cup e_2 : \{s\}} \]

\[ \{e_1 \mid x^s \in e_2\} : \{s\} \quad \frac{e : \{s\}}{	ext{empty } e : \text{bool}} \quad \frac{e_1 : s \quad e_2 : s}{e_1 = e_2 : \text{bool}} \]

Translating into comprehension syntax

\[ \{e_1 \mid x^s \in e_2\} = \{y \mid x^s \in e_2\} \]

Translating from comprehension syntax

\[ \{e_1 \mid x^s \in e_2, \Delta\} = \{x^s \in e_2\} \quad \{e_1 \mid C, \Delta\} = \text{if } C \text{ then } \{e_1 \mid \Delta\} \text{ else } \{\} \]

\[ \{e_1\} = \{e_1\} \]

⇒ Treat comprehension as a nice syntactic sugar
Genomic data

Loci
Tracks
Annotations

E.g. you see a row denoted “Base position”; this is the coordinates on a reference genome. The rest of the rows (e.g. “Gene catalog” and “day7_ESTs blat”) are “tracks” bearing different kinds of annotations. Each track corresponds to one kind of experiments, one kind of predictions, etc.; e.g. “day7_ESTs blat” are short stretches of RNAs from a day-7 sample that have been mapped (using a tool called “blat”) to specific positions on the reference genome.
Genomic data types

An **annotation** datatype ![t] and its subtype **landmark** ![t] of type \( t = (\text{#name}: \text{string}, \text{#pval}: \text{real}, \ldots) \) are represented as \((\text{#loc}: \text{Loc}, \text{#anno}: (\text{name}: \text{string}, \text{#pval}: \text{real}, \ldots))\).

A **track** datatype {![t]} and its subtype **landmark track** {![t]} are represented as { ![t] } and { ![t] }

Equipped with some implicit / automatic normalizations / constraints, e.g. sorted by #loc, idempotency and non-overlapping loci on the same track.

Landmarks on the same landmark track are non-overlapping, and all annotations can “see” no more than one landmark on the same landmark track.

Landmarks can be used for organizing storage & distribution of annotations.
Conceptual organization of annotation & landmark tracks

- Landmark track A
- Landmark track B
- Annotation track C
- Annotation track D
Some operations for the loci type

\begin{align*}
\text{p before q} & \quad \text{p overlap q} \quad \text{p near q} \\
\text{p, q : Loc} & \quad \text{p can-see r} \\
\text{p : Loc, r : !!t} &
\end{align*}

satisfying “p can-see r whenever (p overlap or near q) & q = r.loc”,

plus maybe other convexity constraints to be thought up

Precise set of operations on loci (e.g. p is-nearer q_1 than q_2) is not important here

But a well-designed set of operations should constraint users from “bad”
“expensive” queries, while providing sufficient expressive power for commonly
needed queries
Operations on annotations and tracks

Let's call this language NRC_{genome} in this talk.
Common genomic queries in $\text{NRC}_{\text{genome}}$

“Extract from a track $R$, the annotations in a given region (e.g. 21q22.3) of the genome”

$\{ ! \ x \mid x \in R, \ x.\text{loc} \text{ overlap} \ 21q22.3 \}$

“Extract from the HMMPFAM prediction track, those RBP predictions with $pval < 1E-6$”

$\{ ! \ x \mid x \in \text{HMMPFAM, } x.\text{anno.name} = \text{“RBP”, } x.\text{anno.pval} < 1E-6 \}$

These queries operate on a single track
They can be executed efficiently, viz. $O(n)$, in $\text{NRC}_{\text{genome}}$
“Extract from the TP53 chip-seq track, those TP53 binding sites with pval < 1E-6 and are in promoters of genes”

{ ! x | y ∈ GENES,  x ∈ TP53,
  x.loc before y.loc,
  x.loc near y.loc,
  x.anno.pval < 1E-6 } 

This query operates on two tracks
Its “natural” complexity is $O(\vert GENES\vert \times \vert TP53\vert)$ in NRC_{genome}
“Extract from the TP53 and the HDAC1 chip-seq tracks, those TP53 and HDAC1 binding sites that are closest to each other in the promoters of the same genes”

\[
\{! (\text{loc: } g.\text{loc}, \text{anno: } (\text{name: } g.\text{anno.name}, \text{pval: } 0, \text{tp53: } u, \text{hdac1: } v)) \mid g \in \text{GENES,} \\
(u, v) \in \text{closest } \{ (x, y) \mid x \in \text{TP53, } x.\text{loc near } g.\text{loc, } x.\text{loc before } g.\text{loc,} \\
y \in \text{HDAC1, } y.\text{loc near } g.\text{loc, } y.\text{loc before } g.\text{loc} \} \}
\]

This query has complexity $O(|\text{GENES}| \times |\text{TP53}| \times |\text{HDAC1}|)$ in NRC$_{\text{genome}}$ Does this need to be cubic?
“Extract from the TP53 and the HDAC1 chip-seq tracks, those TP53 and HDAC1 binding sites that are in the promoters of the same genes”

\[
\{! u | u \in \{! (\#loc: g.loc, \#anno: (\#name: g.anno.name, \#pval: 0, \\
\quad \#tp53: \{ x | x \in TP53, x.loc \text{ near } g.loc, x.loc \text{ before } g.loc \}, \\
\quad \#hdac1: \{ y | y \in HDAC1, y.loc \text{ near } g.loc, y.loc \text{ before } g.loc \} \}) \\
| g \in GENES \}, \\
\text{u.anno.tp53} \neq \{! \}, \text{u.anno.hdac1} \neq \{! \}\}
\]

This query has complexity \(O(|GENES| \times (|TP53| + |HDAC1|))\) in NRC\textsubscript{genome}

Does this need to be quadratic?
What is needed? An idea

\[ e : \{! t \}, \ e_1 : \{!! t_1 \}, \ e_2 : \{! t_2 \}, \ldots, \ e_k : \{! t_k \} \]
\[ \cup \{! e \mid (x_1, X_2, \ldots, X_k) \in \in (e_1, e_2, \ldots, e_k) \} : \{! t \} \]

Semantics

\[ \cup \{! e \mid (x_1, X_2, \ldots, X_k) \in \{ (x_1, \{ x_2 \mid x_2 \in e_2, x_2.loc \text{ can-see } x_1 \}, \ldots, \{ x_k \mid x_k \in e_k, x_k.loc \text{ can-see } x_1 \}) \mid x_1 \in e_1 \} \} \]

The part in bold is executed for each landmark, considering only annotations which can see that landmark (assuming these are stored with that landmark)
Common genomic queries revisited

“Extract from the TP53 chip-seq track those TP53 binding sites with pval < 1E-6 and are in promoters of genes”

\[
\{! x \mid y \in GENES, \ x \in TP53, \\
\quad x.\text{loc} \text{ before} y.\text{loc}, \\
\quad x.\text{loc} \text{ near} y.\text{loc}, \\
\quad x.\text{anno}.\text{pval} < 1E-6 \}
\]

GENES is a landmark track

\[
\{! x \mid (y, X) \in (GENES, TP53), \ x \in X, \\
\quad x.\text{loc} \text{ before} y.\text{loc}, \\
\quad x.\text{loc} \text{ near} y.\text{loc}, \\
\quad x.\text{anno}.\text{pval} < 1E-6 \}
\]

Complexity is maybe \( O(|GENES| * 1\% \text{ of } |TP53|) \)
Common genomic queries revisited

“Extract from the TP53 and the HDAC1 chip-seq tracks, those TP53 and HDAC1 binding sites that are closest to each other in the promoters of the same genes”

\[
\{! (\text{loc: } g.\text{loc}, \text{anno: (name: } g.\text{anno.name, } \text{pval: } 0, \text{tp53: } u, \text{hdac1: } v)) \\
| g \in \text{GENES,} \\
(\text{u, v}) \in \text{closest }\{ (x, y) \mid x \in \text{TP53, } x.\text{loc near } g.\text{loc, } x.\text{loc before } g.\text{loc,} \\
y \in \text{HDAC1, } y.\text{loc near } g.\text{loc, } y.\text{loc before } g.\text{loc }\} \}
\]

\[
\{! (\text{loc: } g.\text{loc, } \text{anno: (name: } g.\text{anno.name, } \text{pval: } 0, \text{tp53: } u, \text{hdac1: } v)) \\
| (g, U, V) \in \in (\text{GENES, TP53, HDAC1),} \\
(\text{u, v}) \in \text{closest }\{(x,y) \mid x \in U, x.\text{loc near } g.\text{loc, } x.\text{loc before } g.\text{loc,} \\
y \in V, y.\text{loc near } g.\text{loc, } y.\text{loc before } g.\text{loc}\} \}
\]

Complexity is maybe \(O(|\text{GENES}| \times (1\% \ of \ |\text{TP53}| \times 1\% \ of \ |\text{HDAC1}|)))\)
“Extract from the TP53 and the HDAC1 chip-seq tracks, those TP53 and HDAC1 binding sites that are in the promoters of the same genes”

\[
\{ ! (\text{loc}: g.\text{loc}, \text{anno}: (\text{name}: g.\text{anno}.\text{name}, \text{pval}: 0, \\
\quad \text{tp53}: \{ x | x \in \text{TP53}, x.\text{loc} \text{near} g.\text{loc}, x.\text{loc} \text{before} g.\text{loc} \}, \\
\quad \text{hdac1}: \{ y | y \in \text{HDAC1}, y.\text{loc} \text{near} g.\text{loc}, y.\text{loc} \text{before} g.\text{loc} \}) \} \\
| g \in \text{GENES} \}
\]

\[
\{ ! (\text{loc}: g.\text{loc}, \text{anno}: (\text{name}: g.\text{anno}.\text{name}, \text{pval}: 0, \\
\quad \text{tp53}: \{ x | x \in U, x.\text{loc} \text{near} g.\text{loc}, x.\text{loc} \text{before} g.\text{loc} \}, \\
\quad \text{hdac1}: \{ y | y \in V, y.\text{loc} \text{near} g.\text{loc}, y.\text{loc} \text{before} g.\text{loc} \}) \} \\
| (g, U, V) \in (\text{GENES, TP53, HDAC1}) \}
\]

Complexity is maybe \( O(|\text{GENES}| \times (1\% \text{ of } |\text{TP53}| + 1\% \text{ of } |\text{HDAC1}|)) \)

Is it necessary to process \( U \) and \( V \) twice?
A better idea?

\[ e : \{! t \}, e_1 : \{!! t_1 \}, e_2 : \{! t_2 \}, \ldots, e_k : \{! t_k \}, \gamma_1 : \text{bool}, \ldots, \gamma_k : \text{bool} \]

\[ \bigcup\{! e \mid x_1 \in\in e_1 \text{ st } \gamma_1, \ X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2, \ldots, X_k \subseteq e_k \ni x_k \text{ st } \gamma_k \} : \{! t \} \]

\[ \text{FV}(\gamma_j) \setminus \{x_1, x_j\} \subseteq \text{FV}(\bigcup\{! e \mid x_1 \in\in e_1, X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2, \ldots, X_k \subseteq e_k \ni x_k \text{ st } \gamma_k \}), \ \text{and } \text{FV}(e) \cap \{x_2, \ldots, x_k\} = \{\} \]

Semantics

\[ \bigcup\{! e \mid (x_1, X_2, \ldots, X_k) \in \{ (x_1, \{ x_2 \mid x_2 \in e_2, x_2.\text{loc can-see } x_1, \gamma_2 \}, \ldots,
    \{ x_k \mid x_k \in e_k, x_k.\text{loc can-see } x_1, \gamma_k \} )
    \mid x_1 \in e_1, \gamma_1 \} \} \]

The part in bold is executed for each landmark, considering only annotations which can see that landmark (assuming these are stored with that landmark)
“Extract from the TP53 and the HDAC1 chip-seq tracks, those TP53 and HDAC1 binding sites that are in the promoters of the same genes”

\[ \{! \,(\#loc: g.\text{loc}, \#anno: (\#name: g.\text{anno}.name, \#pval: 0, \\
\quad \#tp53: \{ x \mid x \in \text{TP53}, x.\text{loc} \text{ near} g.\text{loc}, x.\text{loc} \text{ before} g.\text{loc} \}, \\
\quad \#hdac1: \{ y \mid y \in \text{HDAC1}, y.\text{loc} \text{ near} g.\text{loc}, y.\text{loc} \text{ before} g.\text{loc} \} ) ) \\
\mid g \in \text{GENES} \} \]

\[
\{! \,(\#loc: g.\text{loc}, \#anno: (\#name: g.\text{anno}.name, \#pval: 0, \#tp53: U, \#hdac1: V)) \\
\mid g \in \in \text{GENES}, \\
U \subseteq \text{TP53} \ni u \text{ st } u.\text{loc} \text{ near} g.\text{loc} \& u.\text{loc} \text{ before} g.\text{loc} , \\
V \subseteq \text{HDAC1} \ni v \text{ st } v.\text{loc} \text{ near} g.\text{loc}, v.\text{loc} \text{ before} g.\text{loc} \} \\
\]

Complexity is maybe \( O(|\text{GENES}| \times (1\% \text{ of } |\text{TP53}| + 1\% \text{ of } |\text{HDAC1}|)) \)
Common genomic queries revisited again

“Extract from the TP53 and the HDAC1 chip-seq tracks, those TP53 and HDAC1 binding sites that are closest to each other in the promoters of the same genes”

\[
{! (\text{loc: } g.\text{loc}, \text{anno: } (\#\text{name: } g.\text{anno.name}, \#\text{pval: } 0, \#\text{tp53: } u, \#\text{hdac1: } v))}
\mid g \in \text{GENES},
\quad (u, v) \in \text{closest} \{(x, y) \mid x \in \text{TP53}, x.\text{loc near } g.\text{loc}, x.\text{loc before } g.\text{loc},
\quad y \in \text{HDAC1}, y.\text{loc near } g.\text{loc}, y.\text{loc before } g.\text{loc} \}}
\]

\[
{! (\text{loc: } g.\text{loc}, \text{anno: } (\#\text{name: } g.\text{anno.name}, \#\text{pval: } 0, \#\text{tp53: } x, \#\text{hdac1: } y))}
\mid g \in \in \text{GENES}, U \subseteq \text{TP53} \ni u \text{ st } u.\text{loc near } g.\text{loc} \& u.\text{loc before } g.\text{loc},
\quad V \subseteq \text{HDAC1} \ni v \text{ st } v.\text{loc near } g.\text{loc} \& v.\text{loc before } g.\text{loc} ,
\quad (x, y) \in \text{closest} \{(u,v) \mid u \in U, v \in V}\}
\]

Complexity is maybe \(O(\mid\text{GENES}\mid \ast (1\% \text{ of } \mid\text{TP53}\mid \ast 1\% \text{ of } \mid\text{HDAC1}\mid))\)
And this idea? It is really a syntactic sugar

\[
e : \{! t \}, e_1 : \{!! t_1 \}, e_2 : \{! t_2 \}, \ldots, e_k : \{! t_k \}, \gamma_1 : \text{bool}, \ldots, \gamma_k : \text{bool}
\]

\[
\bigcup\{! e \mid x_1 \in e_1 \text{ st } \gamma_1, x_2 \in e_2 \text{ st } \gamma_2, \ldots, x_k \in e_k \text{ st } \gamma_k \} : \{! t \}
\]

\[
FV(\gamma) \setminus \{x_1, x_j\} \subseteq FV(\bigcup\{! e \mid x_1 \in e_1, x_2 \in e_2 \text{ st } \gamma_2, \ldots, x_k \in e_k \text{ st } \gamma_k \})
\]

Semantics

\[
\bigcup\{! e \mid (x_1, X_2, \ldots, X_k) \in \{ (x_1, \{ x_2 \mid x_2 \in e_2, x_2 \text{.loc can-see } x_1, \gamma_2 \}), \ldots,
\]

\[
\{ x_k \mid x_k \in e_k, x_k \text{.loc can-see } x_1, \gamma_k \}) \mid x_1 \in e_1, \gamma_1 \},
\]

\[
x_2 \in X_2, \ldots, x_k \in X_k
\]

The part in bold is executed for each landmark, considering only annotations which can see that landmark (assuming these are stored with that landmark)
Common genomic queries revisited again

“Extract from the TP53 chip-seq track those TP53 binding sites with pval < 1E-6 and are in promoters of genes”

{! x | y ∈ GENES, x ∈ TP53,
  x.loc before y.loc,
  x.loc near y.loc,
  x.anno.pval < 1E-6 }

GENES is a landmark track

{! x | y ∈ GENES,
  x ∈ TP53 st x.loc before y.loc &
  x.loc near y.loc & x.anno.pval < 1E-6 }

Complexity is maybe O(|GENES| * 1% of |TP53|)
Implementing “synchronized” processing of multiple lists / tracks

\[
l\text{zip} \colon (t_1 \to \text{bool}) \times (t_1 \times t_2 \to \text{bool}) \times (t_1 \times t_2 \to \text{bool}) \times (t_2 \times t' \to t') \times (t_1 \times t' \to t') \times (t' \to \{t\}) \times t' \times t' \\
\rightarrow \{t_1\} \times \{t_2\} \rightarrow \{t\}
\]

\[
l\text{zip} (sx, sy, ay, h, g, f, a, e) \colon (\{\}, Y) = f \ a \\
l\text{zip} (sx, sy, ay, h, g, f, a, e) \colon (X, \{\}) = f \ a \\
l\text{zip} (sx, sy, ay, h, g, f, a, e) \colon (x::X, y::Y) = \\
\quad \text{if } sx(x) \\
\quad \quad \text{then if } sy(x, y) \\
\quad \quad \quad \text{then if } ay(x,y) \\
\quad \quad \quad \quad \text{then } l\text{zip} (sx, sy, ay, h, g, f, h(y, g(x, a)), e) \colon (x::X, Y) \\
\quad \quad \quad \text{else } l\text{zip} (sx, sy, ay, h, g, f, g(x, a), e) \colon (x::X, Y) \\
\quad \quad \text{else } f \ (g(x, a)) \cup l\text{zip} (sx, sy, ay, h, g, f, e, e) \colon (X, y::Y) \\
\quad \text{else } f \ a \cup l\text{zip} (sx, sy, ay, h, g, f, e, e) \colon (X, y::Y)
\]

At every step, either x or y gets shifted. So complexity is \(O(|X| + |Y| \times \alpha)\), where \(\alpha\) is complexity of sx, sy, etc.
Implementing $\cup\{! e \mid x_1 \in\in e_1 \text{ st } \gamma_1, X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2, \ldots, X_k \subseteq e_k \ni x_k \text{ st } \gamma_k \}$

$\cup\{! e \mid x_1 \in\in e_1 \text{ st } \gamma_1, X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2 \} :=$

$\text{lzip}(sx, sy, ay, h, g, f, ({!! },{! })), ({!! },{! })) (e_1, e_2)$ where

$sx(x_1) := \gamma_1,$

$ay(x_1, x_2) := x_2.\text{loc can-see } x_1 \& \gamma_2,$

$sy(x_1, x_2) := x_2.\text{loc before } x_1.\text{loc or ay}(x_1, x_2),$

$h(x_2, (X_1, X_2)) := (X_1, X_2 \cup\{! x_2 \}),$

$g(x_1, (X_1, X_2)) := (X_1 \cup\{!! x_1 \}, X_2),$

$f(X_1, X_2) := \cup\{! e \mid x_1 \in X_1\};$

Synchronized scan
A nice property of \( \cup \{! e \mid x_1 \in\in e_1 \text{ st } \gamma_1, \ X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2, \ldots, X_k \subseteq e_k \ni x_k \text{ st } \gamma_k \} \)

\[
\cup \{! e \mid x_1 \in\in e_1 \text{ st } \gamma_1, \ X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2 \} := \\
lzip (\ldots) (e_1, e_2) \text{ where } \ldots
\]

is a homomorphism on \( e_1 \). Thus

\[
\cup \{! e \mid x_1 \in\in e_1 \text{ st } \gamma_1, \ X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2 \} = \\
\cup \{! e \mid x_1 \in\in \{!! o_1, \ldots, o_k\} \text{ st } \gamma_1, \ X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2 \} \cup \ldots \cup \\
\cup \{! e \mid x_1 \in\in \{!! o_k\} \text{ st } \gamma_1, \ X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2 \}
\]

When annotations on track \( e_2 \) are “clustered” (i.e. stored with) the specific landmarks on track \( e_1 \), these annotations “can see”, each \( \cup \{! e \mid x_1 \in\in \{!! o\} \text{ st } \gamma_1, \ X_2 \subseteq e_2 \ni x_2 \text{ st } \gamma_2 \} \) can be run in parallel on each cluster.
Some optimization rules

And $\phi$ is a "positive" condition on loci in both rules.
And …

∪{! ∪{! if $\varphi$ then e else {!} | $x_2 \in e_2$} | $x_1 \in e_1$}

↓

↓ $e_1$: !t$_1$ & $e_2$: !t$_2$ &

↓ $x_1 \not\in$ FV($e_2$) & FV($\varphi$) $\cap \{x_1,x_2\} = \{x_1,x_2\}$ &

↓ $\varphi$ is a positive condition on loci of $x_1,x_2$

↓

∪{! ∪{! if $\varphi$ then e else {!} | $x_1 \in e_1$ st true, $x_2 \in e_2$ st true }

So a user does not need to worry about when to use $\cup\{! e | x_1 \in e_1, \ldots\}$
In fact, ...

It is not necessary for a user to use {!, {!!}, etc.

These can be inferred by a simple type system

And transformed into synchronized/parallel scans by an optimizer