GTrack 1.0:

Unified data format providing customizable representation and high-speed analysis performance within Galaxy

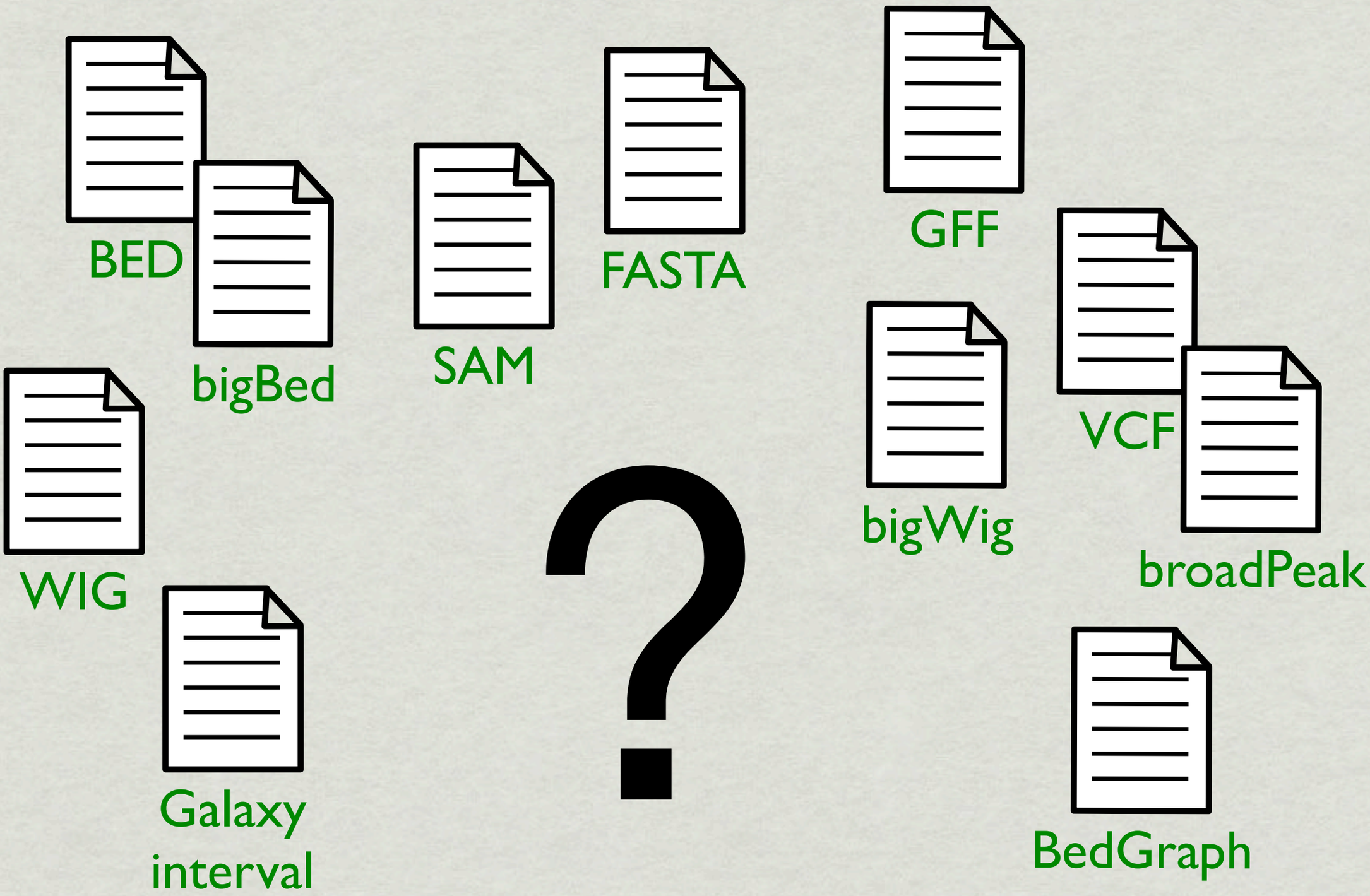
Sveinung Gundersen, PhD student
The Norwegian Radium Hospital, OUH

September 3rd, 1967, Stockholm, Sweden

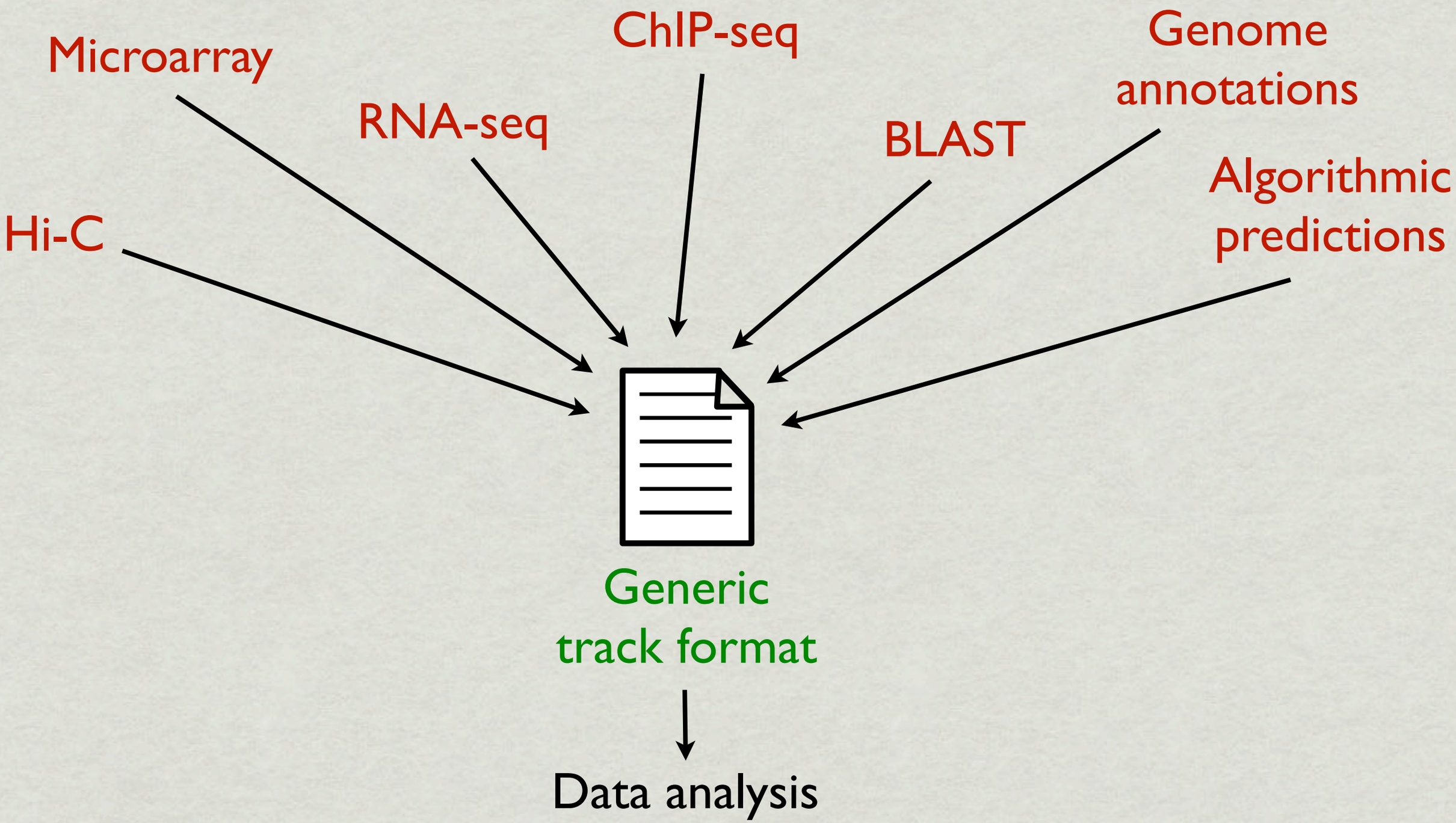
from http://retronaut.com

The day Sweden changed from driving on the left to driving on the right

File formats for genomic track data



Why couldn't our world just be like this?



Why all the formats?

1. Different columns
2. Underlying structural differences
3. Format-specific characteristics
4. Tool-specific format requirements
5. Performance issues

Why all the formats?

1. Different columns
2. Underlying structural differences
3. Format-specific characteristics
4. Tool-specific format requirements
5. Performance issues

Different columns

Different columns

- Different set of columns in the different formats

Different columns

- Different set of columns in the different formats
- Solution: handle flexible columns in one format

Different columns

- Different set of columns in the different formats
- Solution: handle flexible columns in one format
- Galaxy interval format handles flexible columns:

#CHROM	START	END	STRAND	NAME	COMMENT
chr1	10	100	+	exon	myExon
chrX	1000	10050	-	gene	myGene

Different columns

- Different set of columns in the different formats
- Solution: handle flexible columns in one format
- Galaxy interval format handles flexible columns:

#CHROM	START	END	STRAND	NAME	COMMENT
chr1	10	100	+	exon	myExon
chrX	1000	10050	-	gene	myGene

- However:
  - columns are specified as Galaxy metadata (*i.e.* not readily exportable to other tools)
  - only supports interval data

Different columns

- bigBed also handles flexible columns
- However:
  - columns are specified in server-side AutoSQL files
  - only supports interval data

Why all the formats?

1. Different columns
2. Underlying structural differences
3. Format-specific characteristics
4. Tool-specific format requirements
5. Performance issues

Underlying data type differences

Underlying data type differences

- Example: one value per base pair

Underlying data type differences

- Example: one value per base pair
- BedGraph format:

```
chr1     0     1      1.0
chr1     1     2     -1.3
chr1     2     3      2.4
...
```

Underlying data type differences

- Example: one value per base pair
- BedGraph format:

```
chr1     0     1      1.0
chr1     1     2     -1.3
chr1     2     3      2.4
...
```

- WIG

```
fixedStep chrom=chr1
1.0
-1.3
2.4
...
```

Underlying data type differences

- Example: one value per base pair
- BedGraph format:

Size (Human genome)

```
chr1    0    1     1.0
chr1    1    2    -1.3
chr1    2    3     2.4
...
```

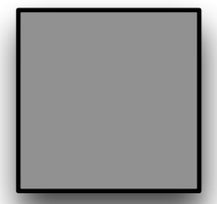
100 GB

- WIG

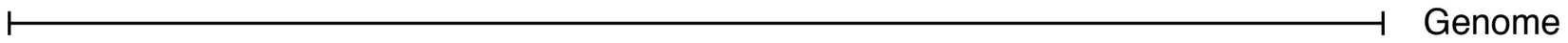
```
fixedStep chrom=chr1
1.0
-1.3
2.4
...
```

20 GB

No information

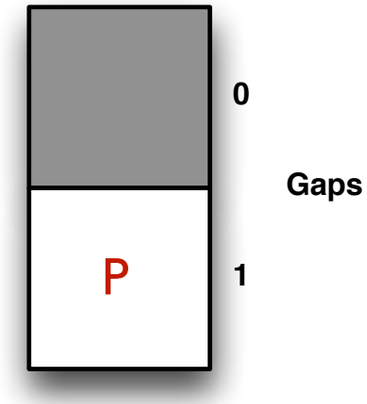


Track geometry

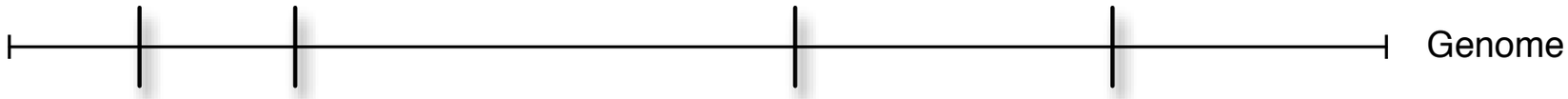


Track type #1:

Points

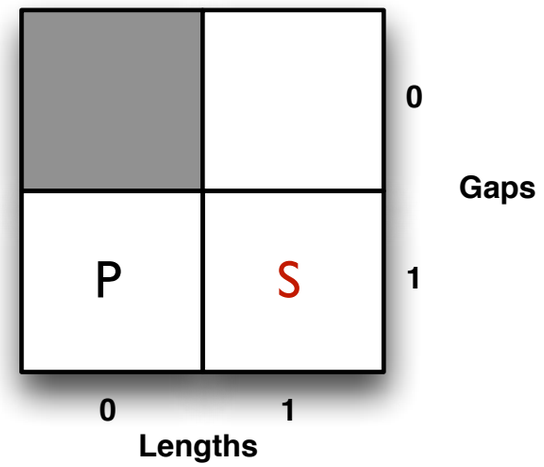


Track geometry

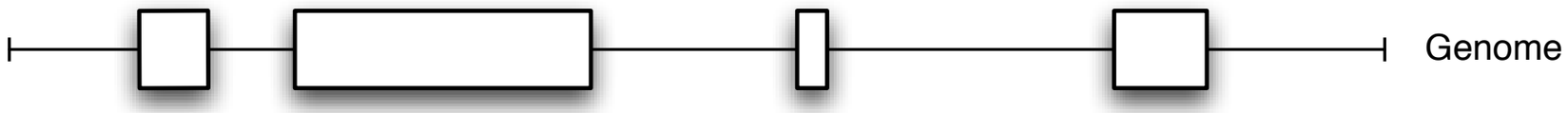


Track type #2:

Segments

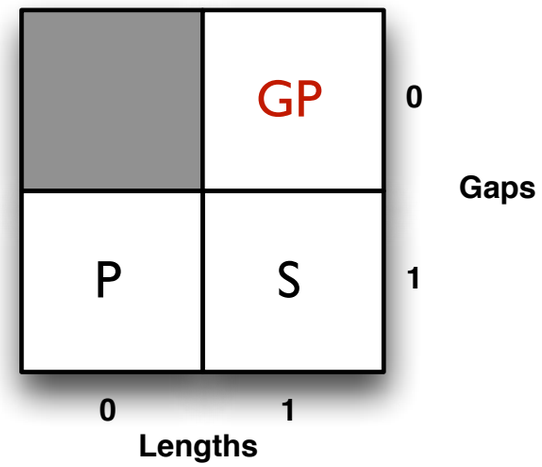


Track geometry

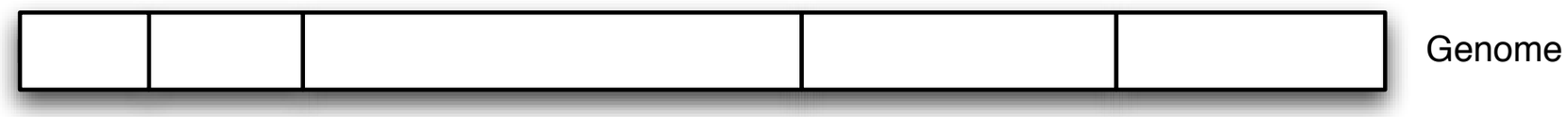


Track type #3:

Genome partition

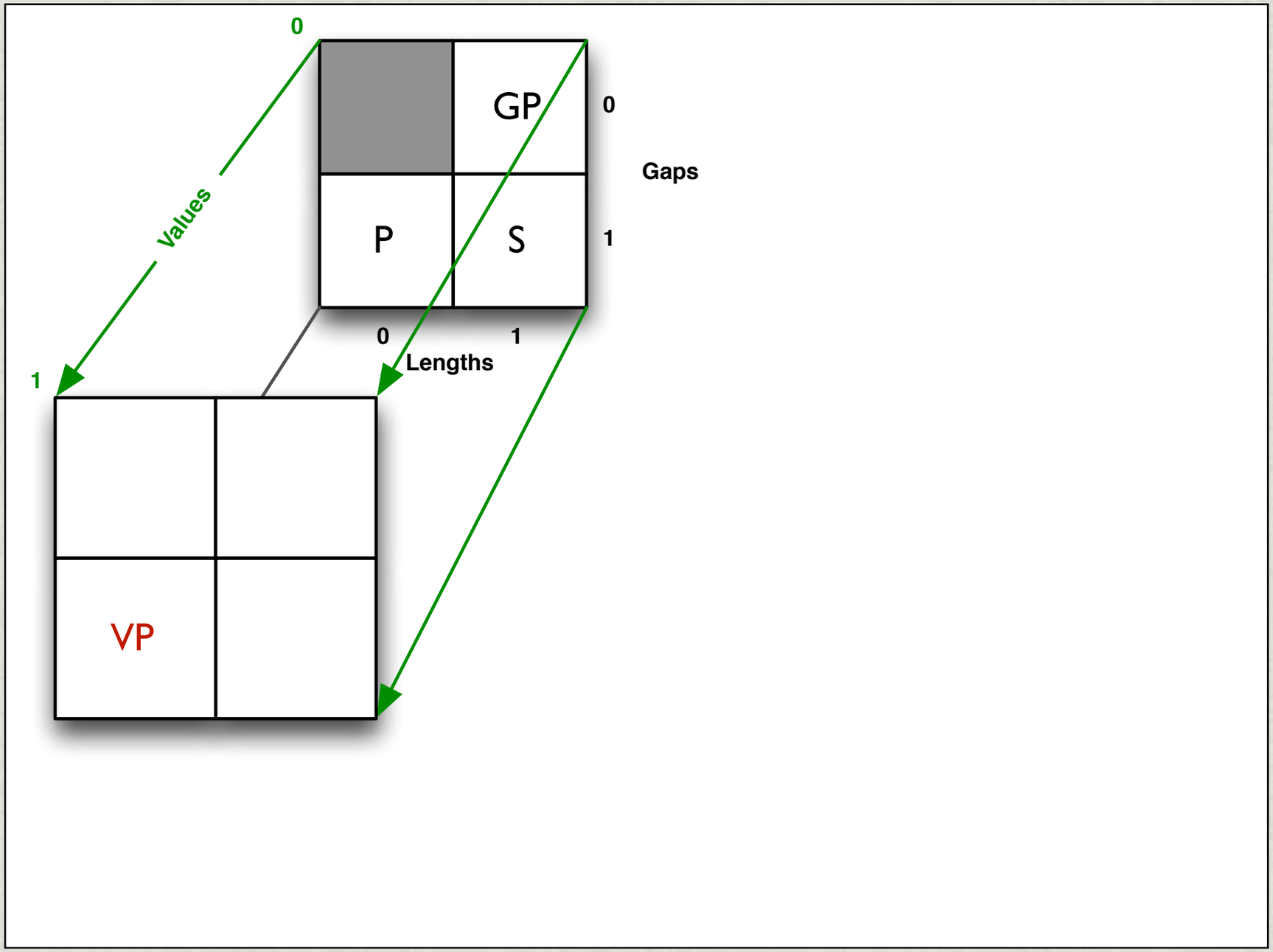


Track geometry

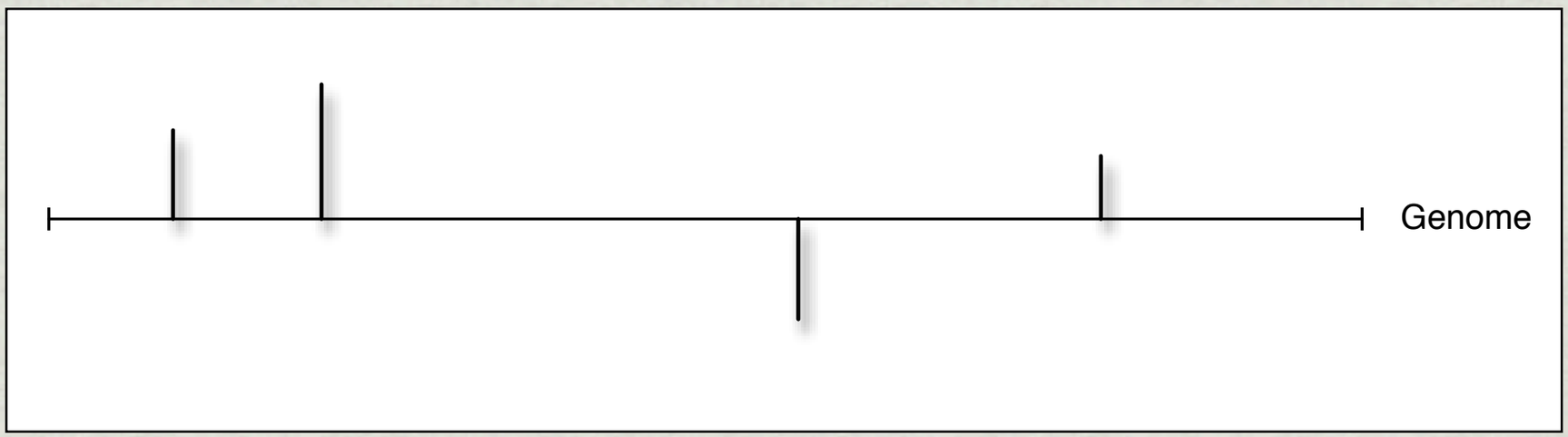


Track type #4:

Valued points

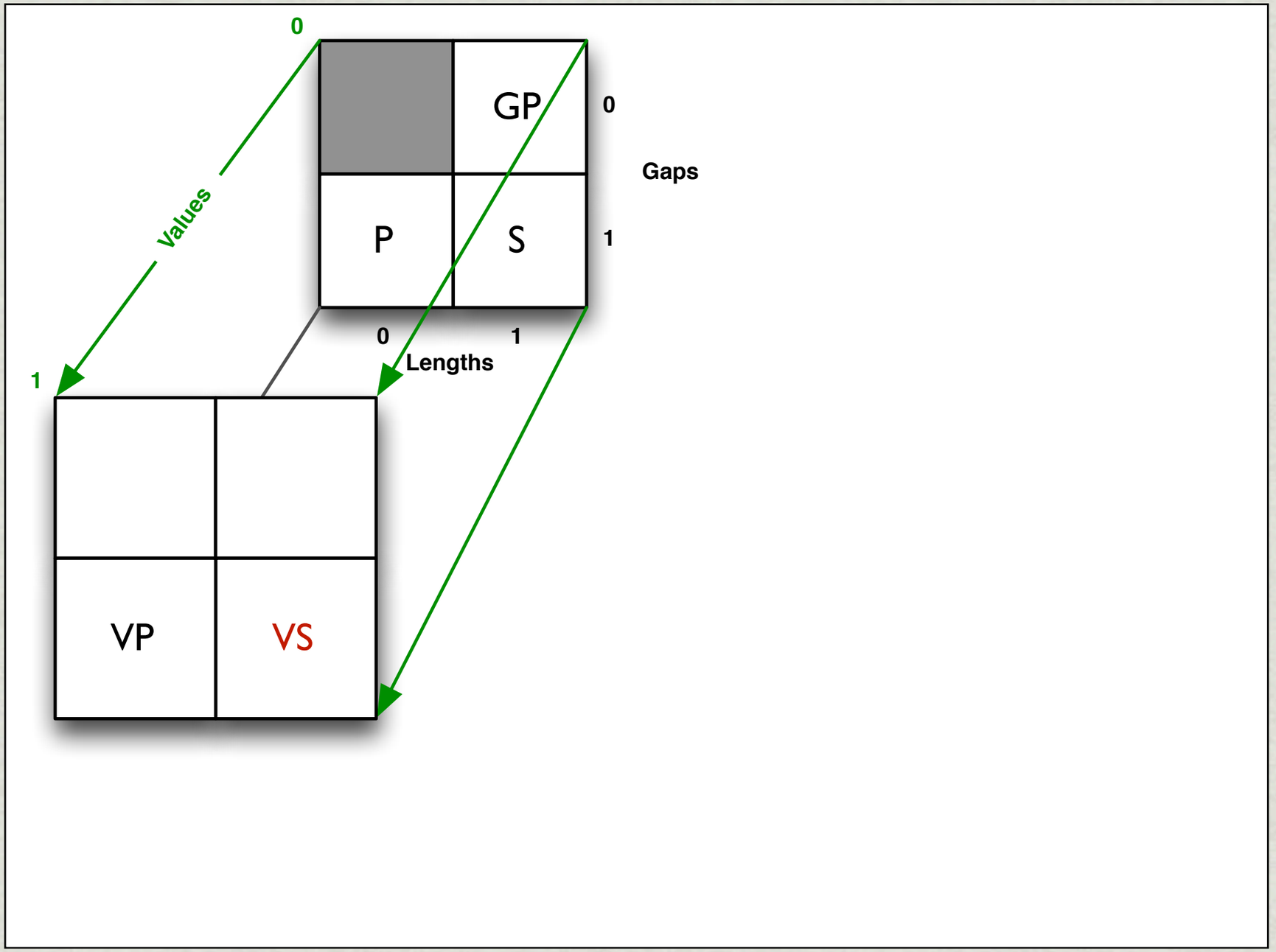


Track geometry

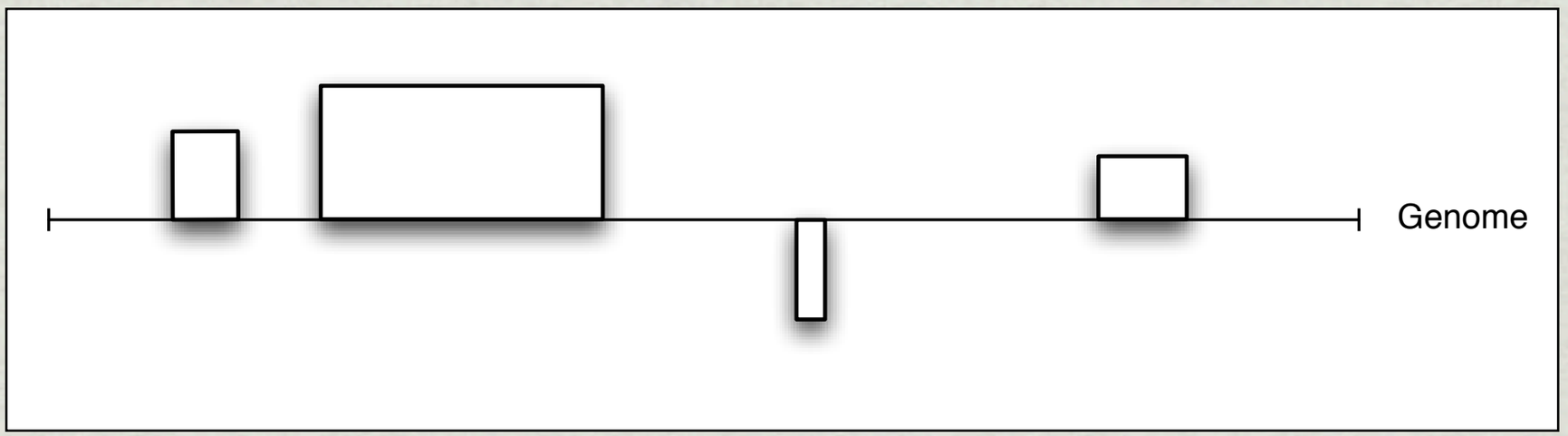


Track type #5:

Valued segments

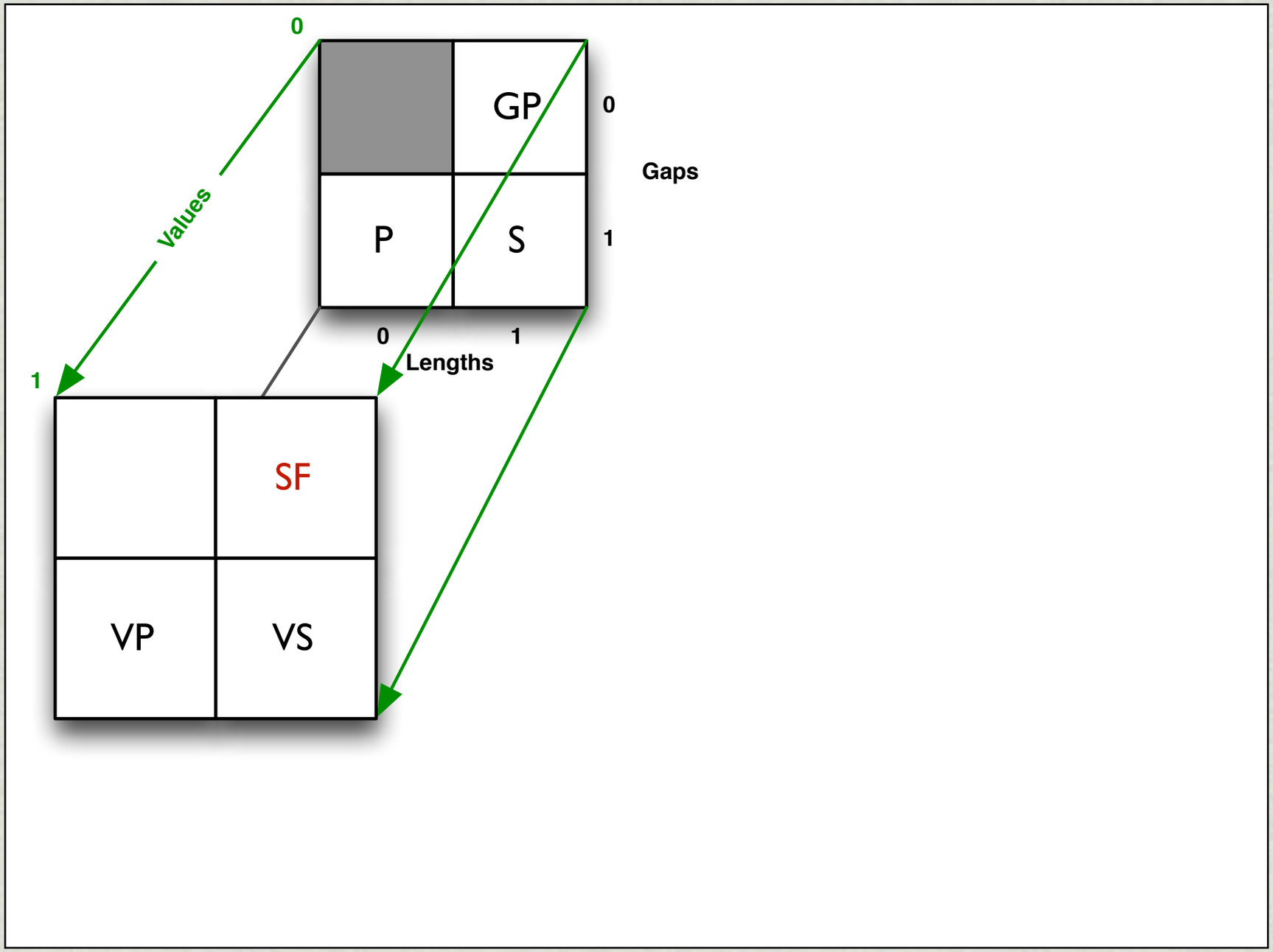


Track geometry

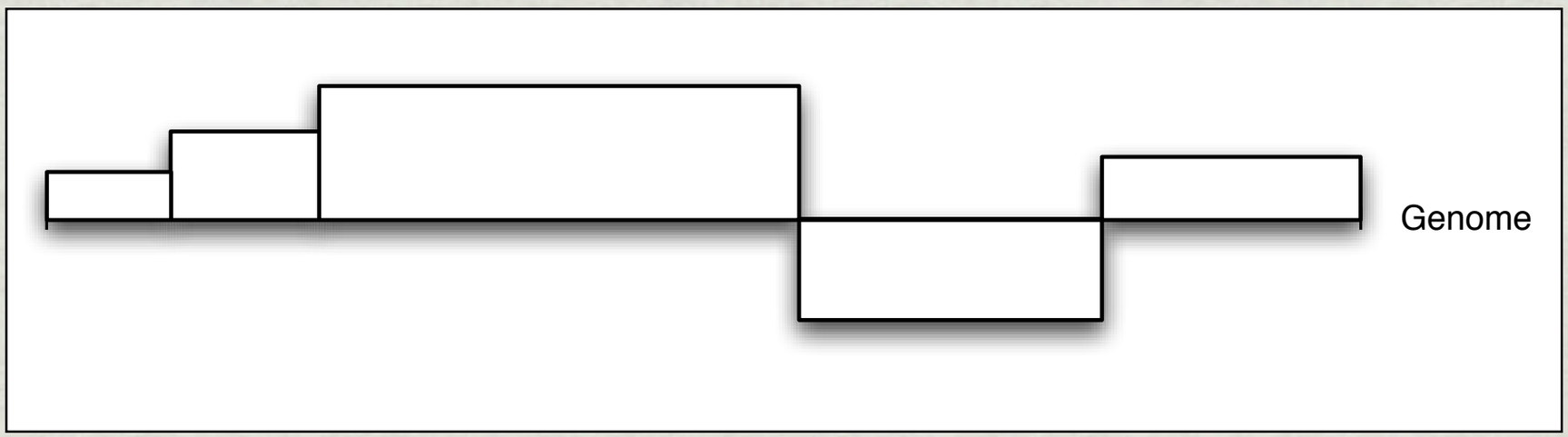


Track type #6:

Step function

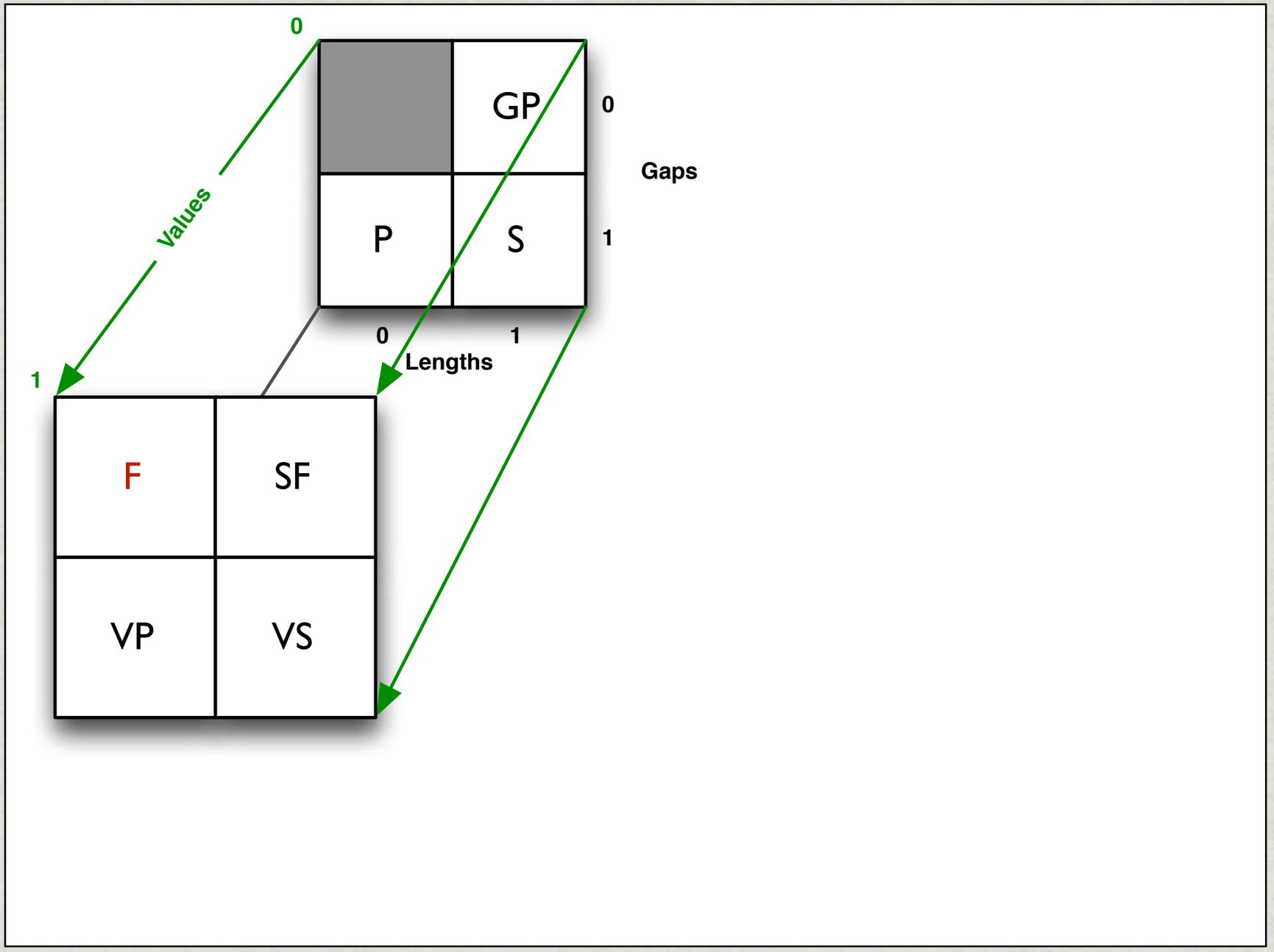


Track geometry

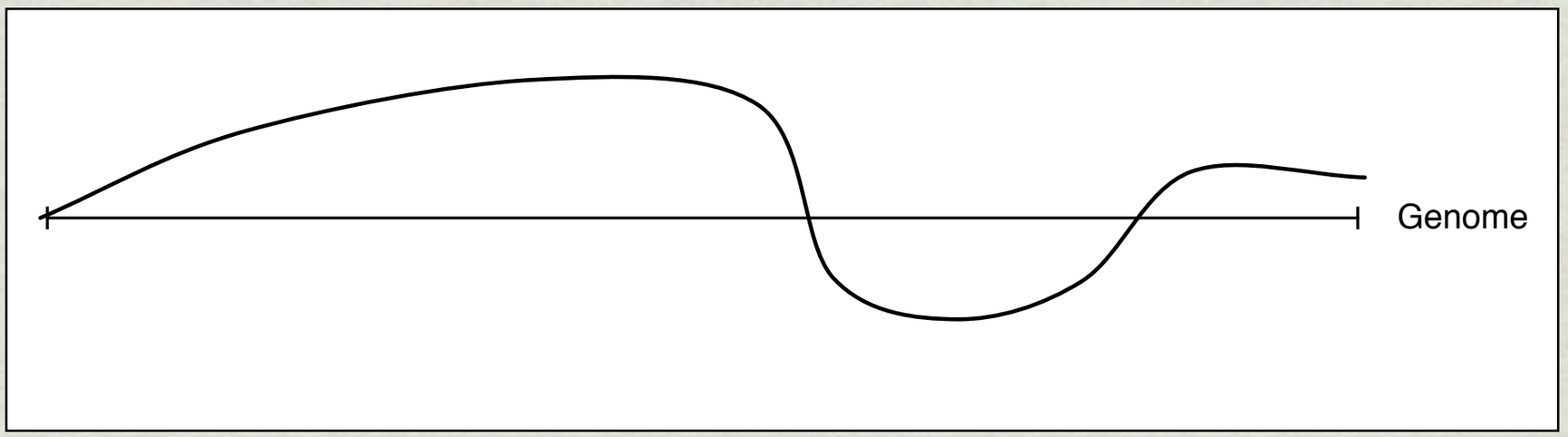


Track type #7:

Function

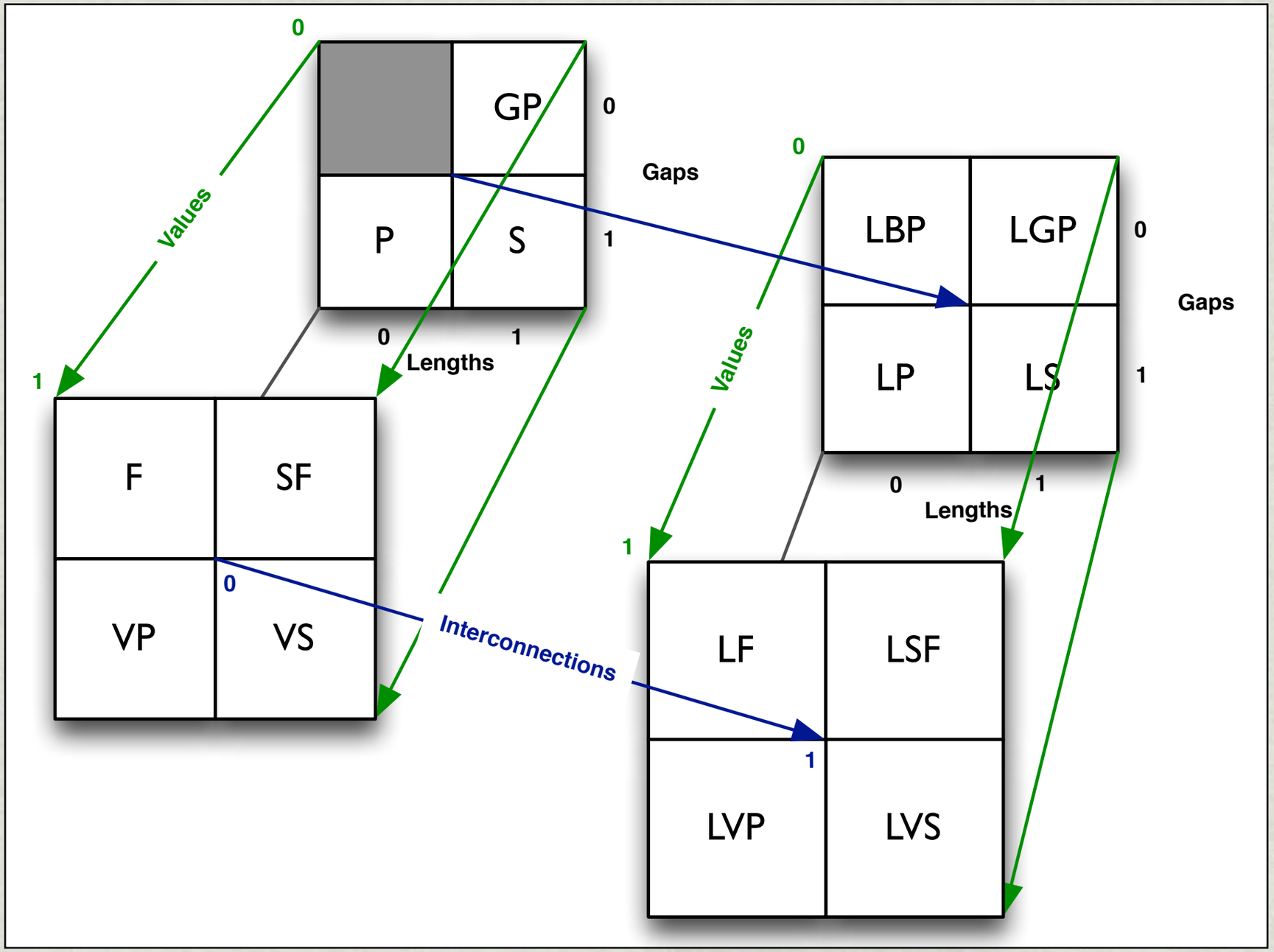


Track geometry

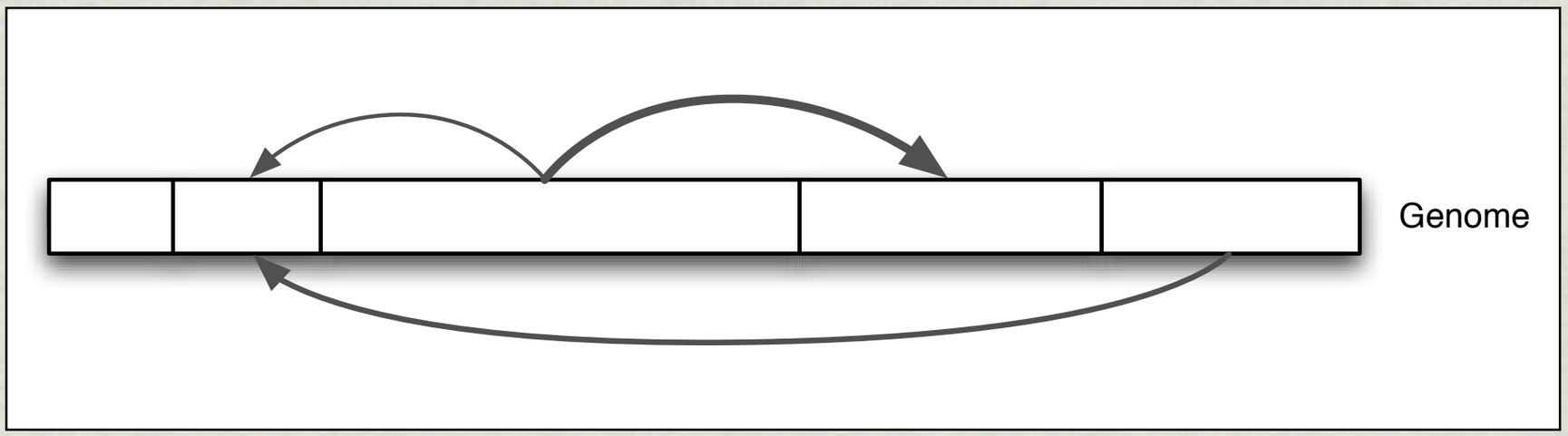


Track type #8-15:

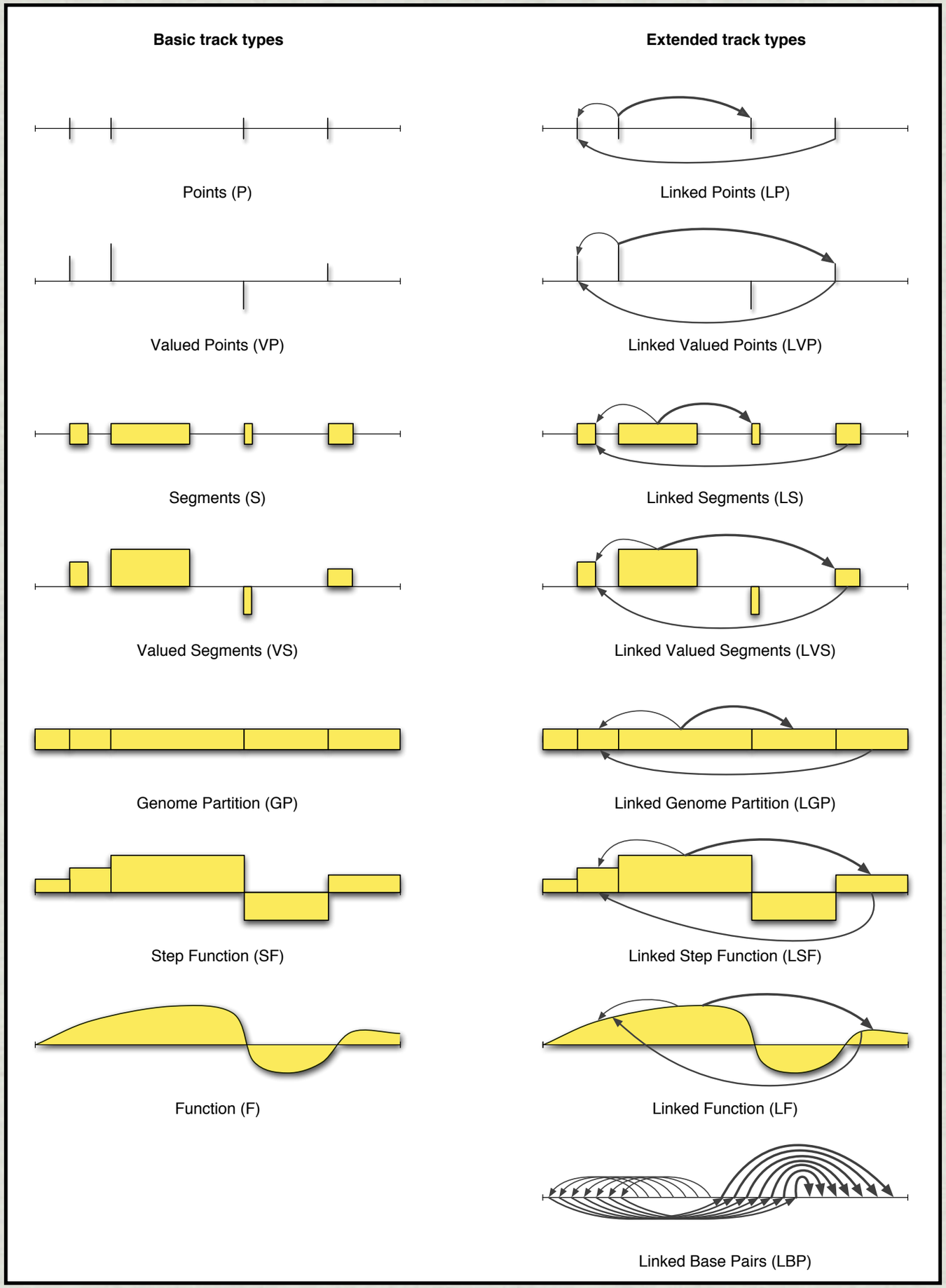
Linked track types



Track geometry



All 15 track types



Track types relate to analysis questions

P	P	Different frequencies?
P	P	Located nearby?
P	S	Located inside?
P	S	Located nonuniformly inside?
P	S	Located nearby?
S	S	Similar segments?
S	S	Overlap?
S	S	Located nearby?
F	F	Correlated?
P	F	Higher values at locations?
S	F	Higher values inside?
P	VS	Located in segments with high values?
S	VP	Higher values inside segments?
VP	VP	Nearby values similar?
P	VS (c/c)	Located in case segments
VS (c/c)	S	Preferential overlap?
VP (cat)	VS (cat)	Category pairs differentially co-located?
LGP	P	Colocalized in 3D?

Track types supported by different formats

Format	Ref.	Data	Repr.	P	S	VP	VS	GP	SF	F	L	Strand	#Cols	Value type
GFF3/GTF	[2]	General	Tab.	√(1)	√	√(1)	√				(2)	√	9	Float (3)
BED/bigBed	[4]	General	Tab./ Bin.	√(1)	√	√(1)	√				(2)	√	3-12	Int(0-1000) /string (4)
BED15	[4]	Microarray	Tab.			√(1)	√				(2)	√	15	List of floats (5)
bedGraph	[4]	General	Tab.			√(1)	√						4	Float
WIG/bigWig (fixedStep)	[8]	General	Tab./ Bin.			√	√		√	√			1	Float
WIG/bigWig (variableStep)	[8]	General	Tab./ Bin.			√	√						2	Float
CNT	[36]	Copy number	Tab.			√							4	Float
Personal Genome SNP	[4]	Variation	Tab.			√(1)	√						7	String (6)
VCF	[37]	Variation	Tab.			√	√						≥ 8	String (6) (3)
GVF	[6]	General/ Variation	Tab.	√(1)	√	√(1)	√				(2)	√	9	Float (3)
PSL	[4]	Alignment	Tab.		√		√					√	21	Int (7)
SAM/BAM	[38]	Alignment	Tab./ Bin.		√		√					√	11	Int /string (8)
BioHDF	[39]	Alignment	Bin.		√		√					√	11	Int /string (8)
MAF	[4]	Multiple Alignment	Tab.		√		√				(9)	√	2-7	Float /string (8)
FASTA	[40]	Sequence	Text							√			N/A	Char
DAS XML	[12]	General	XML	√(1)	√	√(1)	√				(2)	√	N/A	Float
BioXSD 1.0	[16]	General	XML	√(10)	√(10)	√(10)	√(10)				√(11)	√	N/A	Float (12)
USeq	[19]	General	Bin.	√	√	√	√					√	N/A	Int/float/string
Genomedata	[41]	General	Bin.			√	√		√	√			N/A	Int/float/char

Why all the formats?

1. Different columns
2. Underlying structural differences
3. Format-specific characteristics
4. Tool-specific format requirements
5. Performance issues

Format-specific characteristics

- GFF format:
  - 1-based coordinates
  - Circular genomes
  - Format-specific headers

Why all the formats?

1. Different columns
2. Underlying structural differences
3. Format-specific characteristics
4. Tool-specific format requirements
5. Performance issues

Tool-specific format requirements

- Tools may have specific format requirements

Tool-specific format requirements

- Tools may have specific format requirements
- Galaxy tool (BedTools):
  - intersect multiple sorted BED files

Tool-specific format requirements

- Tools may have specific format requirements
- Galaxy tool (BedTools):
  - intersect multiple sorted BED files
- Segtools toolkit:
  - variant of BED where column 4 is interpreted as a category, rather than a name

Why all the formats?

1. Different columns
2. Underlying structural differences
3. Format-specific characteristics
4. Tool-specific format requirements
5. Performance issues

Performance issues

- Binary (indexed) versions of tabular formats are commonly created to improve performance:
  - SAM => BAM
  - BED => bigBed
  - WIG => bigWig
  - Any interval => Tabix

Introducing GTrack v1.0...

GTrack example (simple)

```
chr1        2396586     2827369
chr1        92014277    93306499
chr1        100983315   101455310
chr1        116832232   116909542
chr1        190733439   190814781
chr1        199128354   199336605
...
```

GTrack example (intermediate)

```
##gtrack version: 1.0
##track type: valued segments
##value type: binary
##sorted elements: true
##1-indexed: true
##end inclusive: true
##my custom header: Yes!
###seqid  start       end         value
chr1      2396587     2827369     1
chr1      92014278    93306499    1
chr1      100983316   101455310   0
chr1      116832233   116909542   0
chr1      190733440   190814781   1
chr1      199128355   199336605   0
...
```

GTrack example (intermediate)

```
##gtrack version: 1.0
##track type: valued segments
##value type: binary
##sorted elements: true
##1-indexed: true
##end inclusive: true
##my custom header: Yes!
###seqid   start        end          value
chr1       2396587      2827369      1
chr1       92014278     93306499     1
chr1       100983316    101455310    0
chr1       116832233    116909542    0
chr1       190733440    190814781    1
chr1       199128355    199336605    0
...
```

Flexible columns

GTrack example (intermediate)

```
##gtrack version: 1.0
##track type: valued segments
##value type: binary
##sorted elements: true
##1-indexed: true
##end inclusive: true
##my custom header: Yes!
###seqid  start       end         value
chr1      2396587     2827369     1
chr1      92014278    93306499    1
chr1      100983316   101455310   0
chr1      116832233   116909542   0
chr1      190733440   190814781   1
chr1      199128355   199336605   0
...
```

Supports all 15 track types

GTrack example (intermediate)

```
##gtrack version: 1.0
##track type: valued segments
##value type: binary
##sorted elements: true
##1-indexed: true
##end inclusive: true
##my custom header: Yes!
###seqid   start        end          value
chr1       2396587      2827369      1
chr1       92014278     93306499     1
chr1       100983316    101455310    0
chr1       116832233    116909542    0
chr1       190733440    190814781    1
chr1       199128355    199336605    0
...
```

Format-specific characteristics, custom headers

GTrack example (intermediate)

```
##gtrack version: 1.0
##track type: valued segments
##value type: binary
##sorted elements: true
##1-indexed: true
##end inclusive: true
##my custom header: Yes!
###seqid  start       end         value
chr1      2396587     2827369     1
chr1      92014278    93306499    1
chr1      100983316   101455310   0
chr1      116832233   116909542   0
chr1      190733440   190814781   1
chr1      199128355   199336605   0
...
```

Tool-specific format requirements

GTrack example (intermediate)

Specification of GTrack subtypes via URL
with automatic validation

```
##subtype url: http://my.edu/frmt.gtrack
chr1        2396587     2827369     1
chr1        92014278    93306499    1
chr1        100983316   101455310   0
chr1        116832233   116909542   0
chr1        190733440   190814781   1
chr1        199128355   199336605   0
...
```

GTrack example (advanced)

```
#Example track
##gtrack version: 1.0
##track type: linked genome partition
##edge weights: true
###end    id     edges
####seqid=chr1;start=1000;end=2000
1015       A       C=1.3,D=0.1
1060       B       A=1.0
1154       C       A=1.3
1267       D       .

...
```

GTrack example (advanced)

```
#Example track
##gtrack version: 1.0
##track type: linked genome partition
##edge weights: true
###end      id        edges
####seqid=chr1;start=1000;end=2000
1015        A         C=1.3,D=0.1
1060        B         A=1.0
1154        C         A=1.3
1267        D         .

...
```

Specification of edges with weights

GTrack example (advanced)

```
#Example track
##gtrack version: 1.0
##track type: linked genome partition
##edge weights: true
###end     id      edges
####seqid=chr1;start=1000;end=2000
1015        A       C=1.3,D=0.1
1060        B       A=1.0
1154        C       A=1.3
1267        D       .

...
```

Specification of bounding regions (to exclude missing data / assembly gaps)

GTrack performance

- GTrackCore library is in development, parsing GTrack files into a binary format for fast analysis

GTrack performance

- GTrackCore library is in development, parsing GTrack files into a binary format for fast analysis
- Based on the backbone of the Genomic HyperBrowser

GTrack performance

- GTrackCore library is in development, parsing GTrack files into a binary format for fast analysis
- Based on the backbone of the Genomic HyperBrowser
- Blazingly fast, vector-based calculations using numpy!

GTrack performance

- GTrackCore library is in development, parsing GTrack files into a binary format for fast analysis
- Based on the backbone of the Genomic HyperBrowser
- Blazingly fast, vector-based calculations using numpy!
- Converts to/from many common formats

GTrack performance

- GTrackCore library is in development, parsing GTrack files into a binary format for fast analysis
- Based on the backbone of the Genomic HyperBrowser
- Blazingly fast, vector-based calculations using numpy!
- Converts to/from many common formats
- Planned to move to PyTables:
  - HDF5-based, compressed storage
  - advanced indexing and querying
  - solid performance
  - several tracks in one file

Galaxy relevance of GTrack

- 15 track types (including 3D type data)!

Galaxy relevance of GTrack

- 15 track types (including 3D type data)!
- Provides easy way to handle tool-specific format requirements

Galaxy relevance of GTrack

- 15 track types (including 3D type data)!
- Provides easy way to handle tool-specific format requirements
- Everything specified in the file itself, interoperable with other systems

Galaxy relevance of GTrack

- 15 track types (including 3D type data)!
- Provides easy way to handle tool-specific format requirements
- Everything specified in the file itself, interoperable with other systems
- GTrack fully integrated with the HyperBrowser, along with a set of GTrack-specific tools

Galaxy relevance of GTrackCore

- HDF5-based binary file as history element

Galaxy relevance of GTrackCore

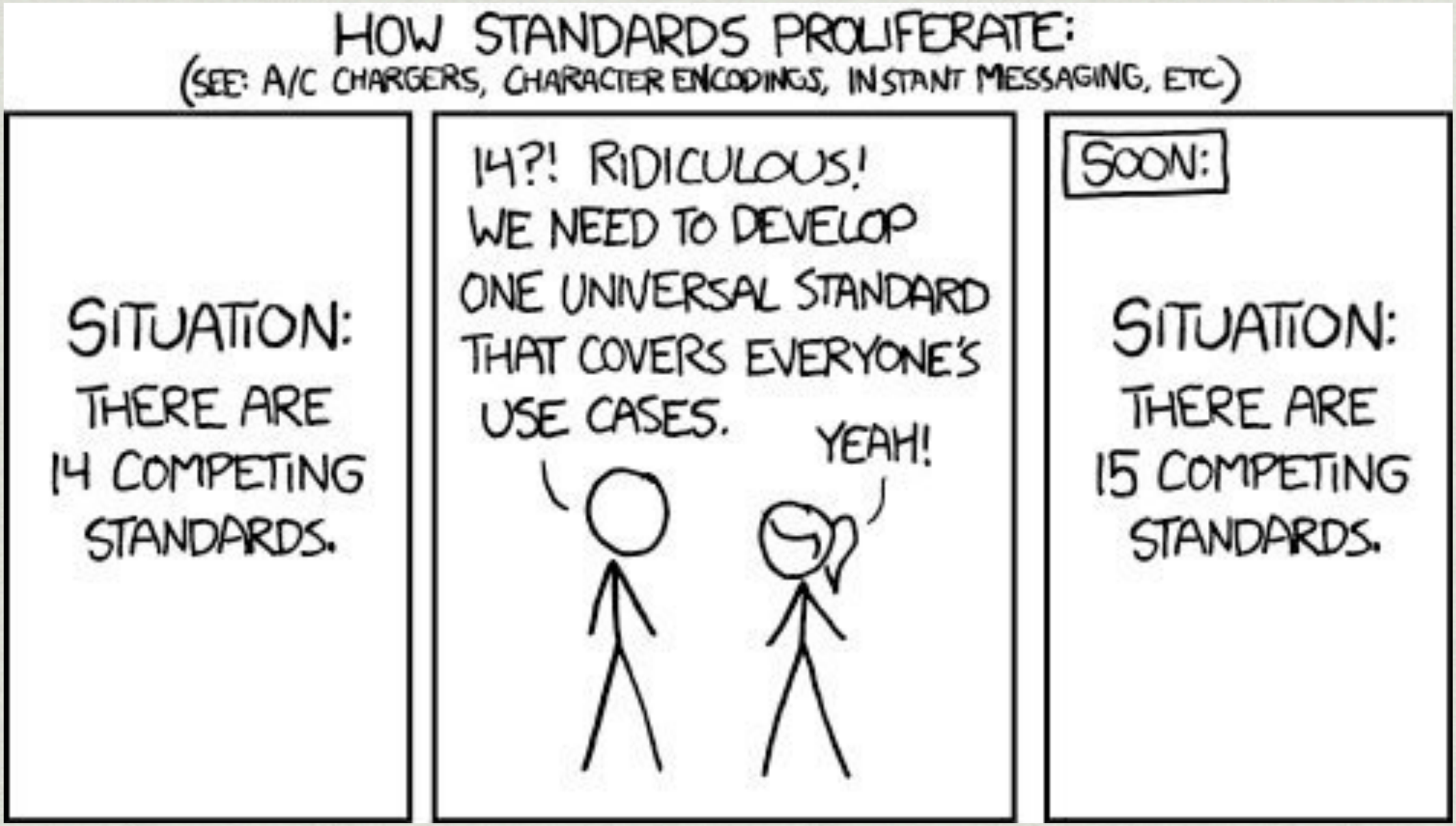
- HDF5-based binary file as history element
- Fast performance (after indexing)

Galaxy relevance of GTrackCore

- HDF5-based binary file as history element
- Fast performance (after indexing)
- Library of operations, analysis, statistics to ease tool development

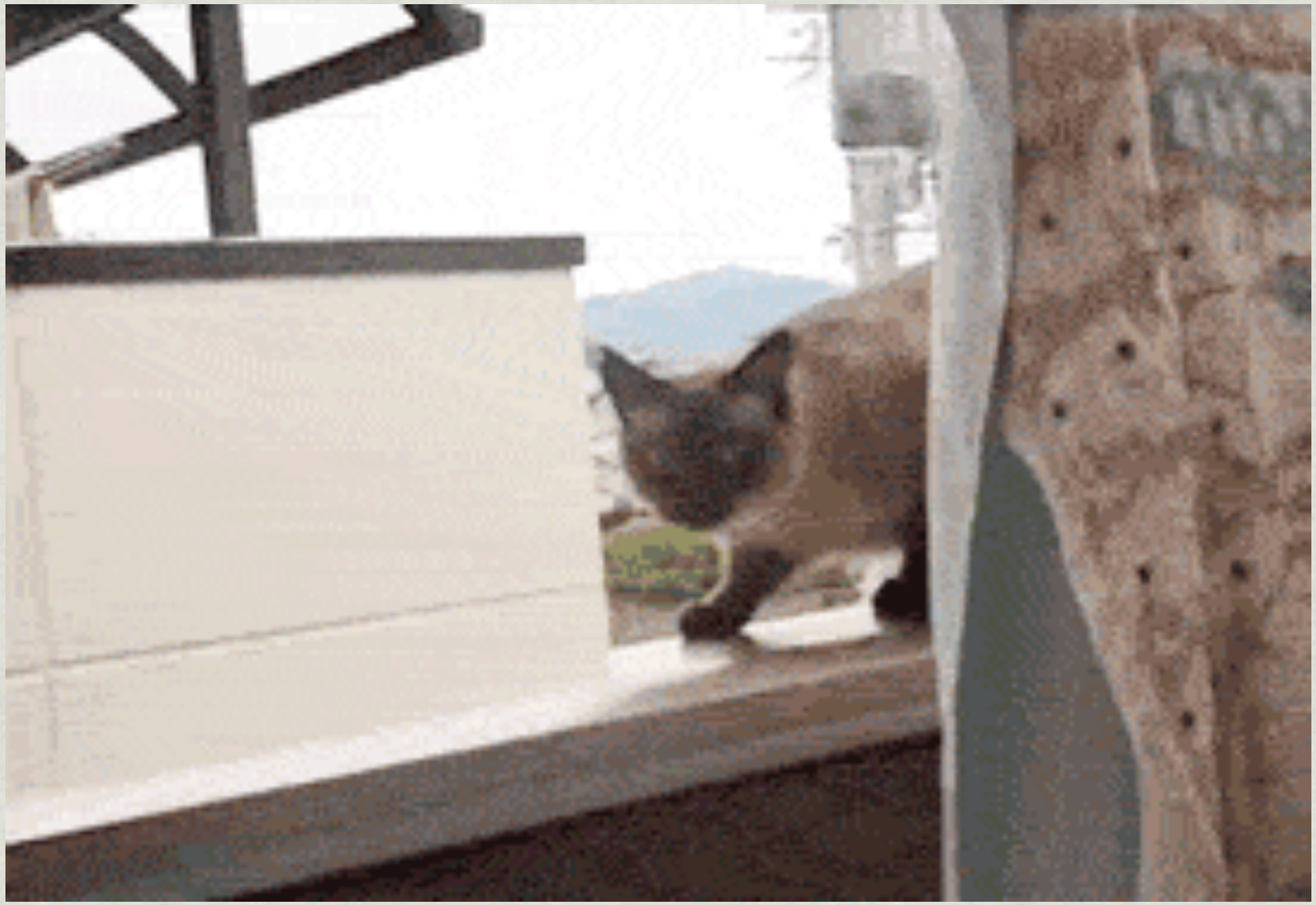
Interested?

We need more developers!



from http://xkcd.com

Nothing ventured,
nothing gained...



Acknowledgements

Sveinung Gundersen

Matúš Kalaš

Osman Abul

Marcin Cieślik

Arnoldo Frigessi

Eivind Hovig

Geir Kjetil Sandve

More info

Gundersen S, Kalas M, Abul O, Frigessi A, Hovig E, Sandve GK: *Identifying elemental genomic track types and representing them uniformly.* BMC Bioinformatics 2011, 12:494.

Specification available at:
www.gtrack.no

GTrackCore source code:
https://github.com/i000/gtrackcore