CS273A

Lecture 12: Inferring Evolution: Chains & Nets

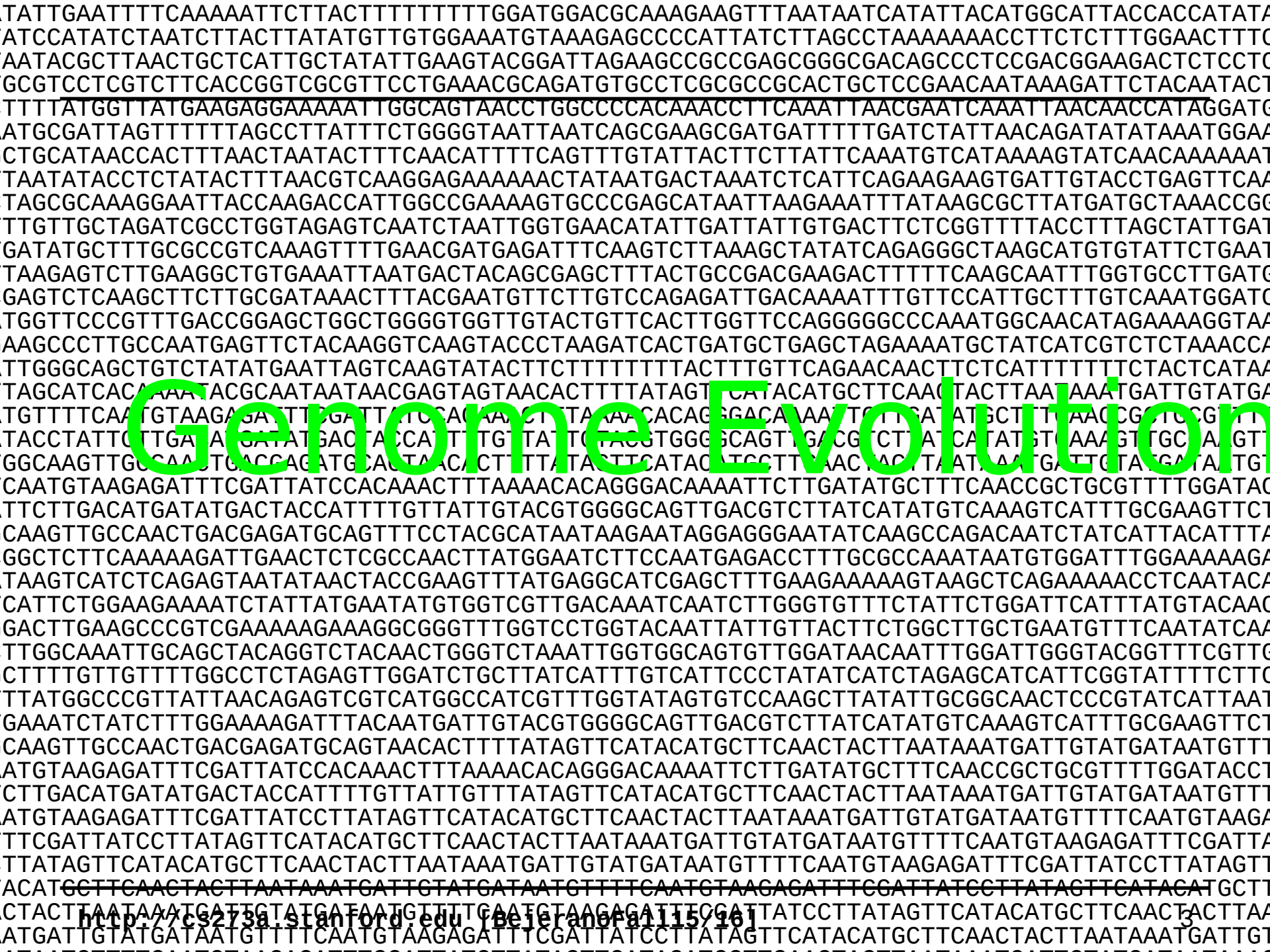
MW 1:30-2:50pm in Clark **S361*** (behind Peet's)

Profs: Serafim Batzoglou & Gill Bejerano

CAs: Karthik Jagadeesh & Johannes Birgmeier

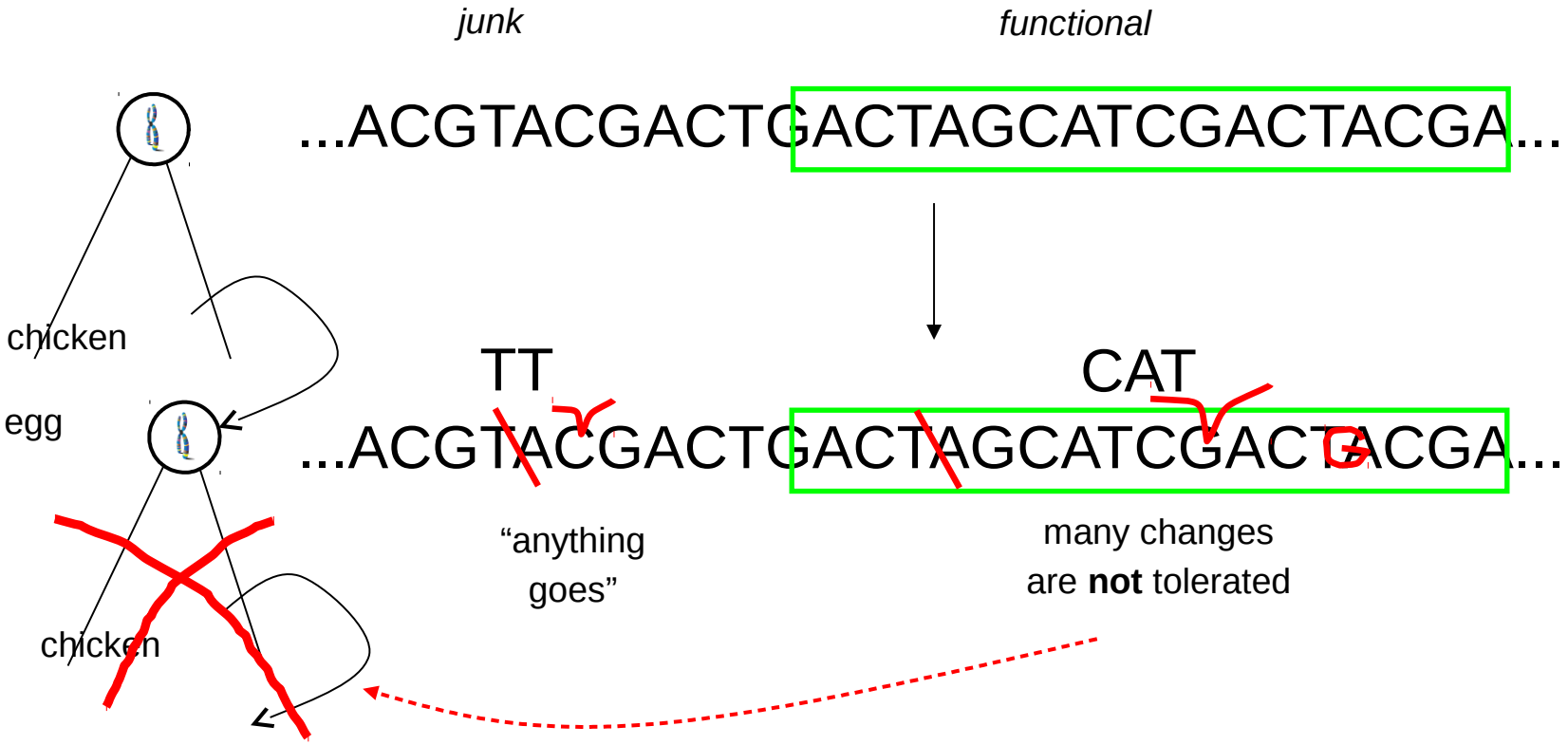
* Mostly: track on website/piazza

Announcements


Genome Evolution

Evolution = Mutation + Selection

Mistakes can happen during DNA replication. Mistakes are oblivious to DNA segment function. But then selection kicks in.



This has bad implications – disease, and good implications – adaptation.

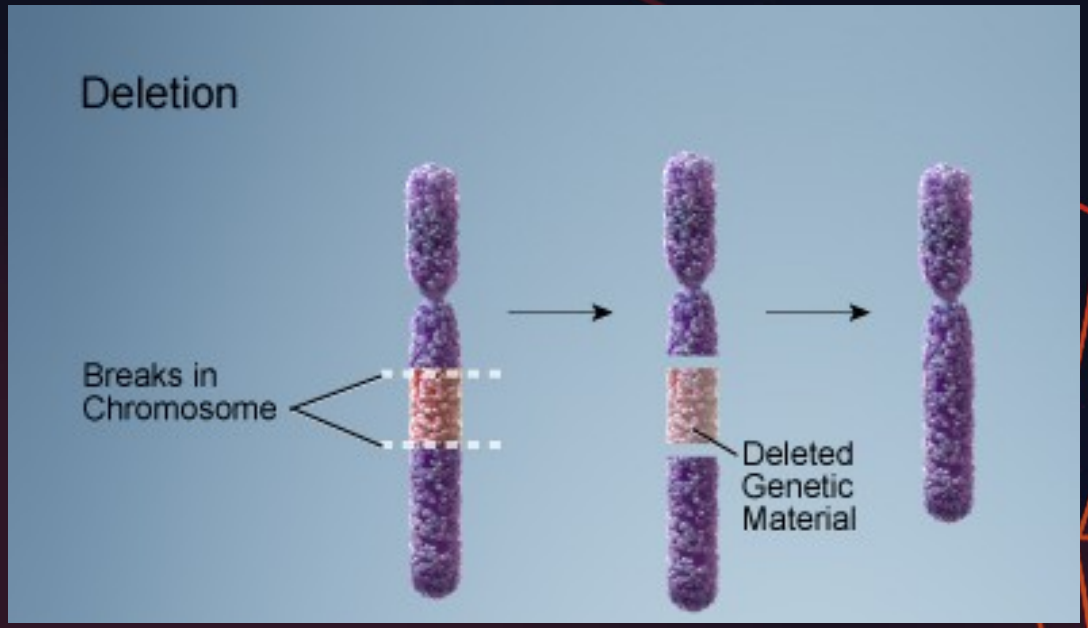
Mutation

Chromosomal (ie big) Mutations

- Five types exist:
  - **Deletion**
  - **Inversion**
  - **Duplication**
  - **Translocation**
  - **Nondisjunction**

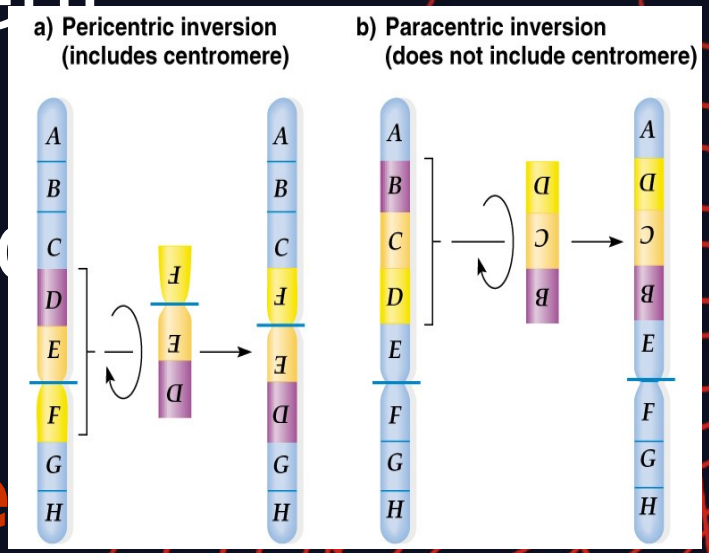
Deletion

- Due to **breakage**
- A **piece** of a chromosome is **lost**



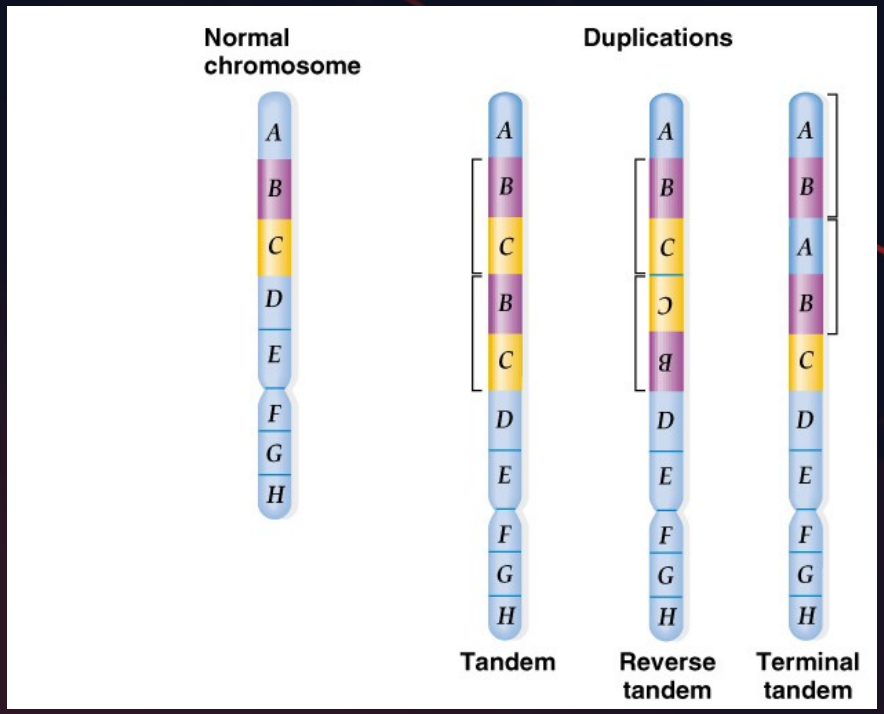
Inversion

- Chromosome segment **breaks off**
- Segment flips aroun **backwards**
- Segment **reattache**
- This reverses **and complements** the sequence.



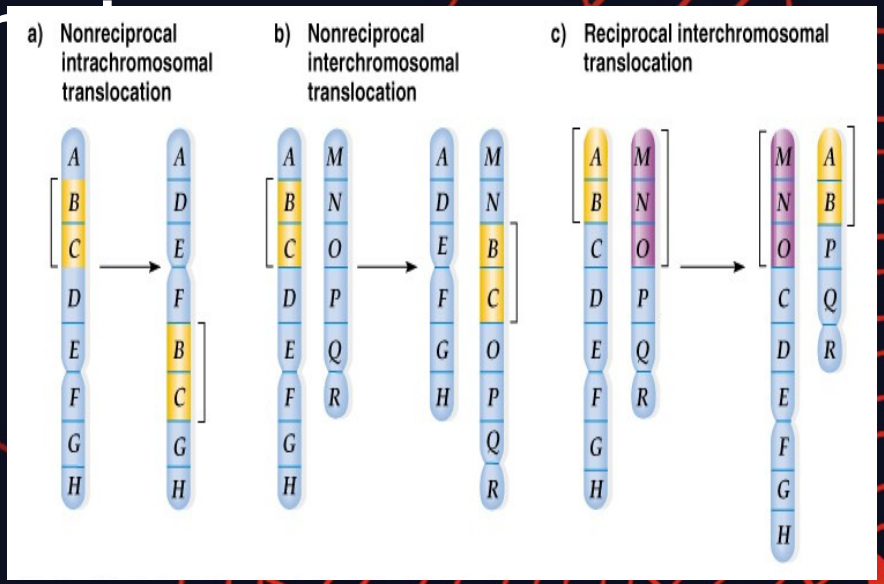
Duplication

- Occurs when a genomic **region is repeated**



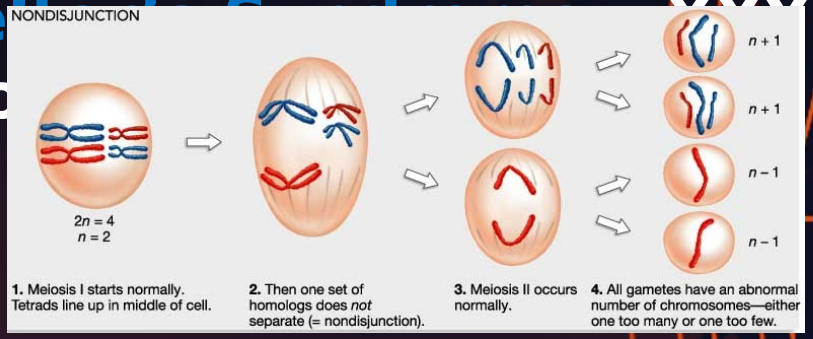
Translocation

- Involves **two chromosomes** th
aren’t homologou
- **Part** of one chromosome is **transferred to another** chromosomes



Nondisjunction

- **Failure** of chromosomes **to separate** during meiosis
- Causes gamete to have **too many or too few chromosomes**
- **Disorders:**
  - **Down Syndrome – three 21st chromosomes**
  - **Turner Syndrome – single X chromosome**
  - **Klinefe… – XXY chromo…**



Genomic (ie small) Mutations

- Six types exist:
  - **Substitution (eg G□T)**
  - **Deletion**
  - **Insertion**
  - **Inversion**
  - **Duplication**
  - **Translocation**

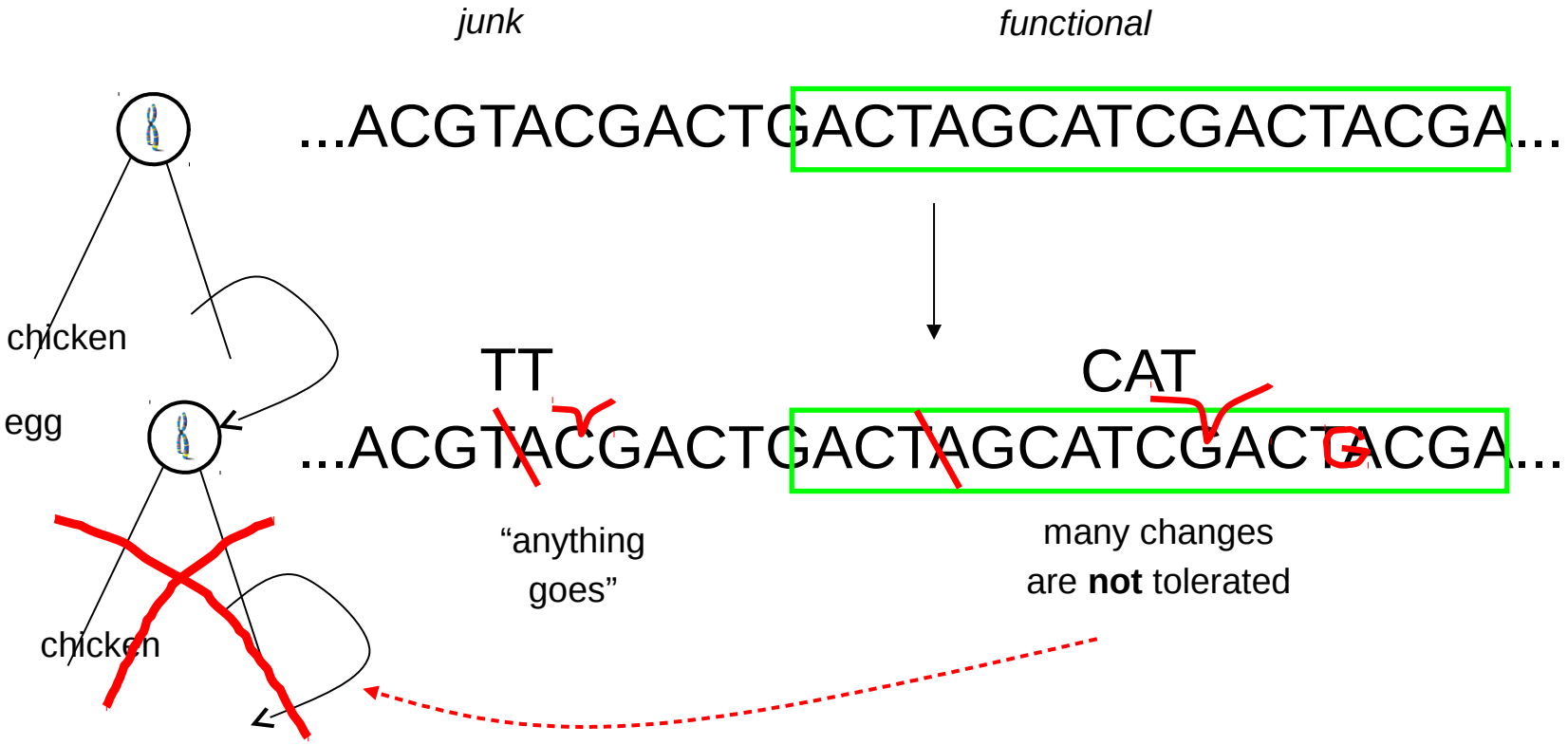
Selection

Mutation is oblivious of functional context.

But then selection kicks in.

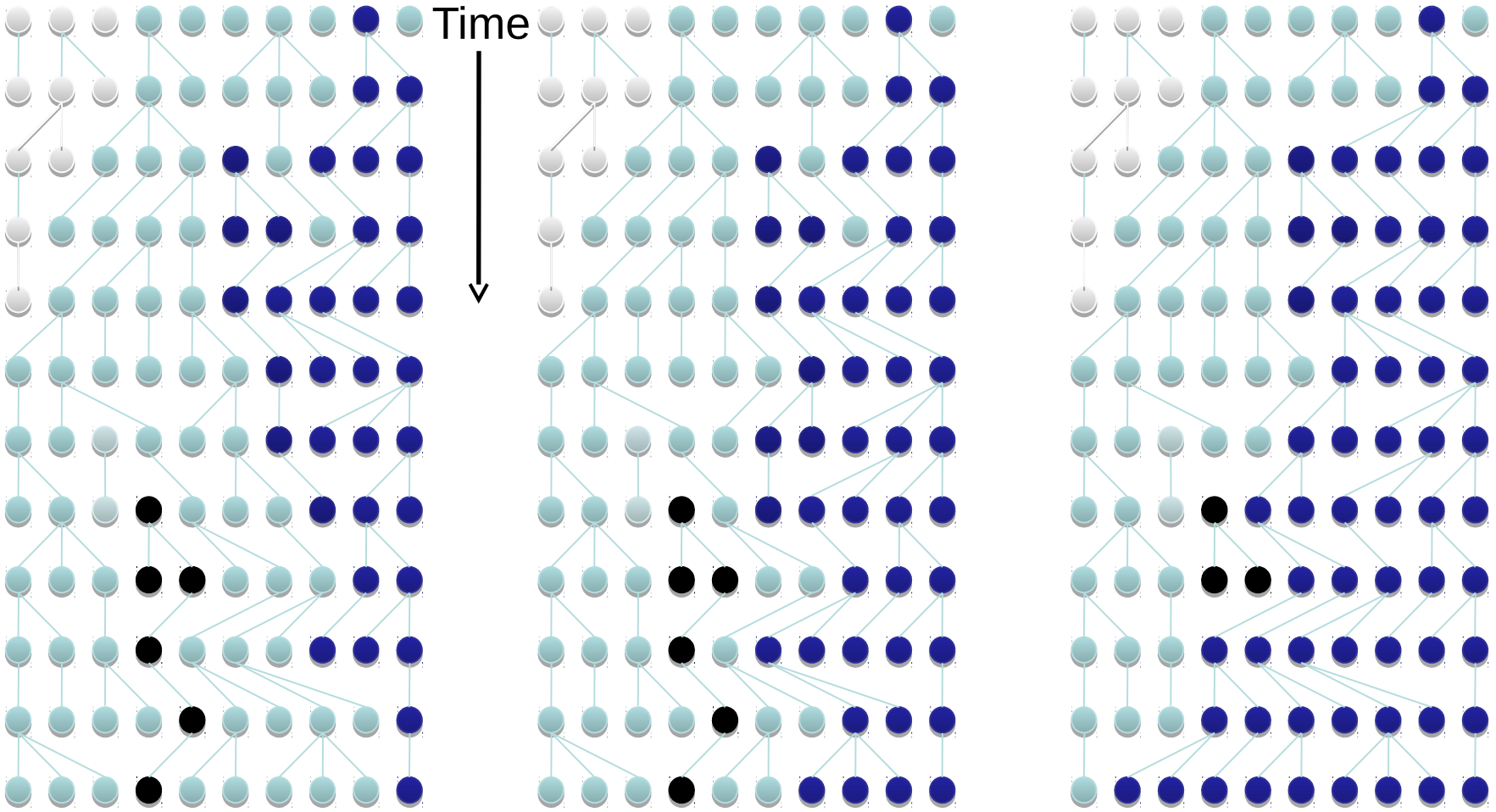
Evolution = Mutation + Selection

Mistakes can happen during DNA replication. Mistakes are oblivious to DNA segment function. But then selection kicks in.



This has bad implications – disease, and good implications – adaptation.

Single Locus View

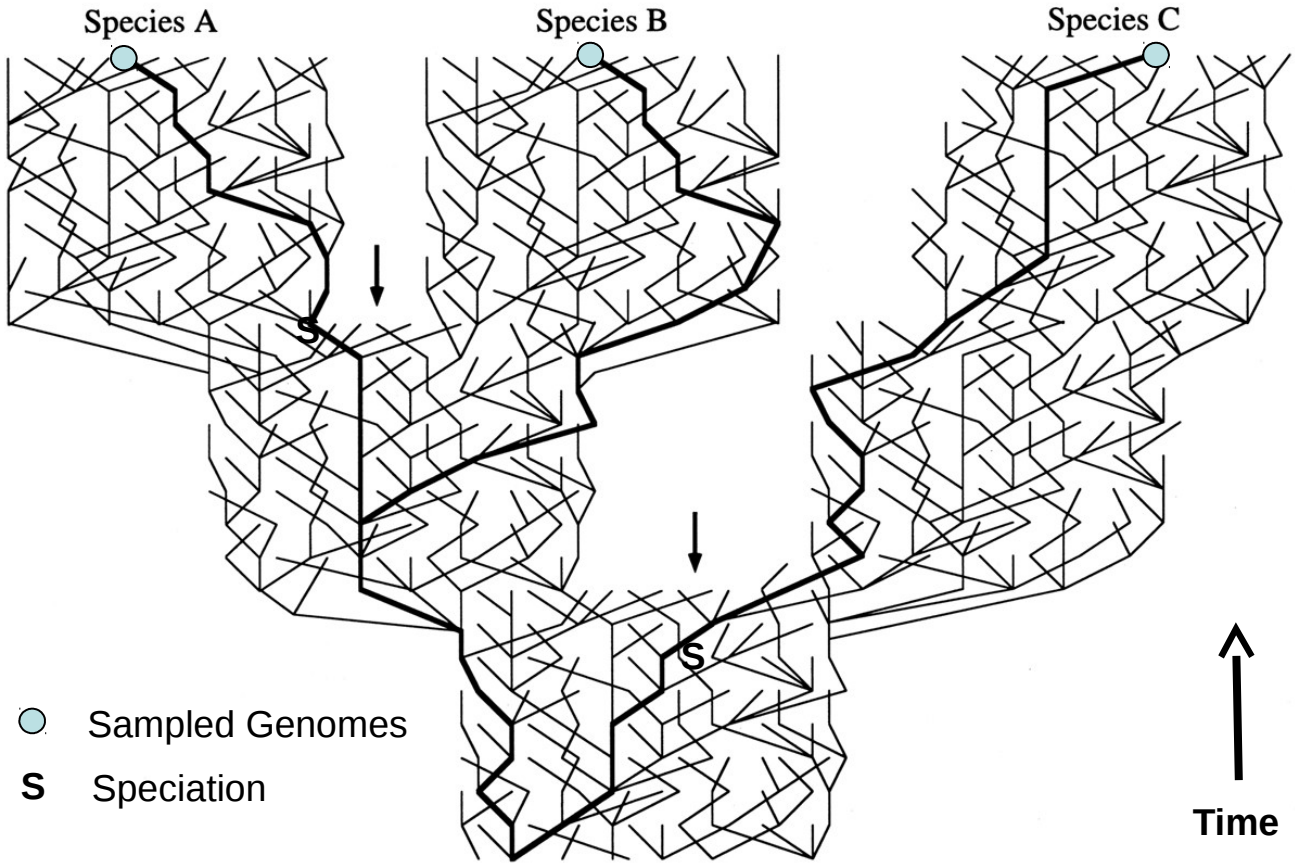


Negative Selection

Neutral Drift

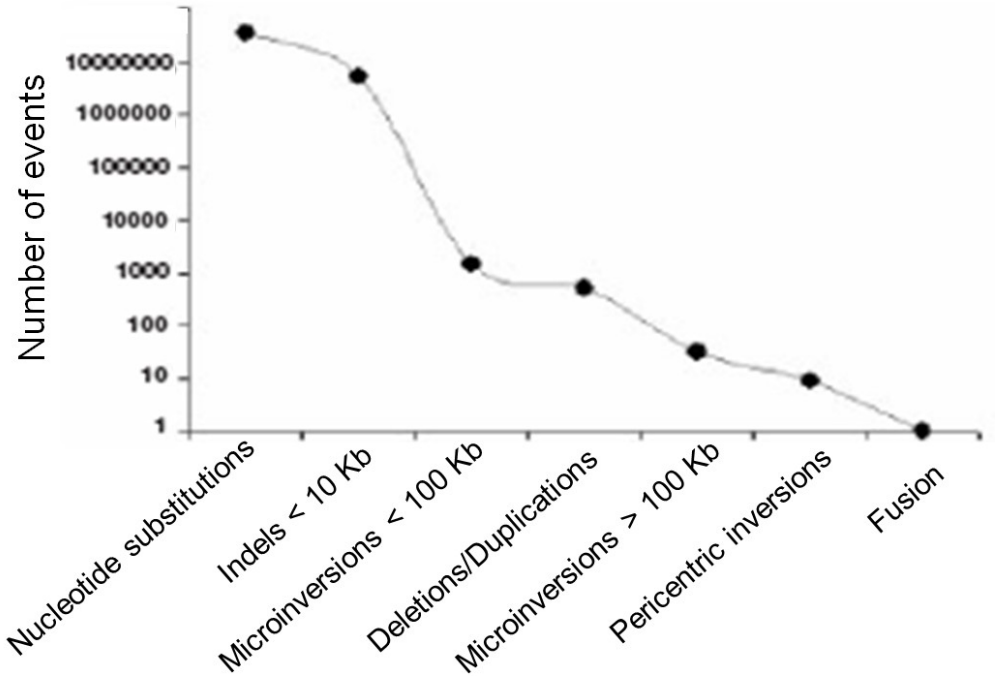
Positive Selection

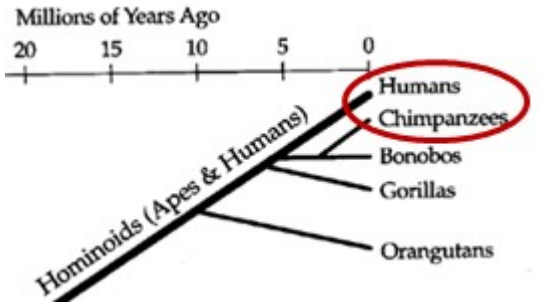
The Species Tree



When we compare one individual from two species, most, but not all mutations we see are fixed differences between the two species.

Example: Human-Chimp Genomic Differences





Human and chimp genomes are 96% identical.

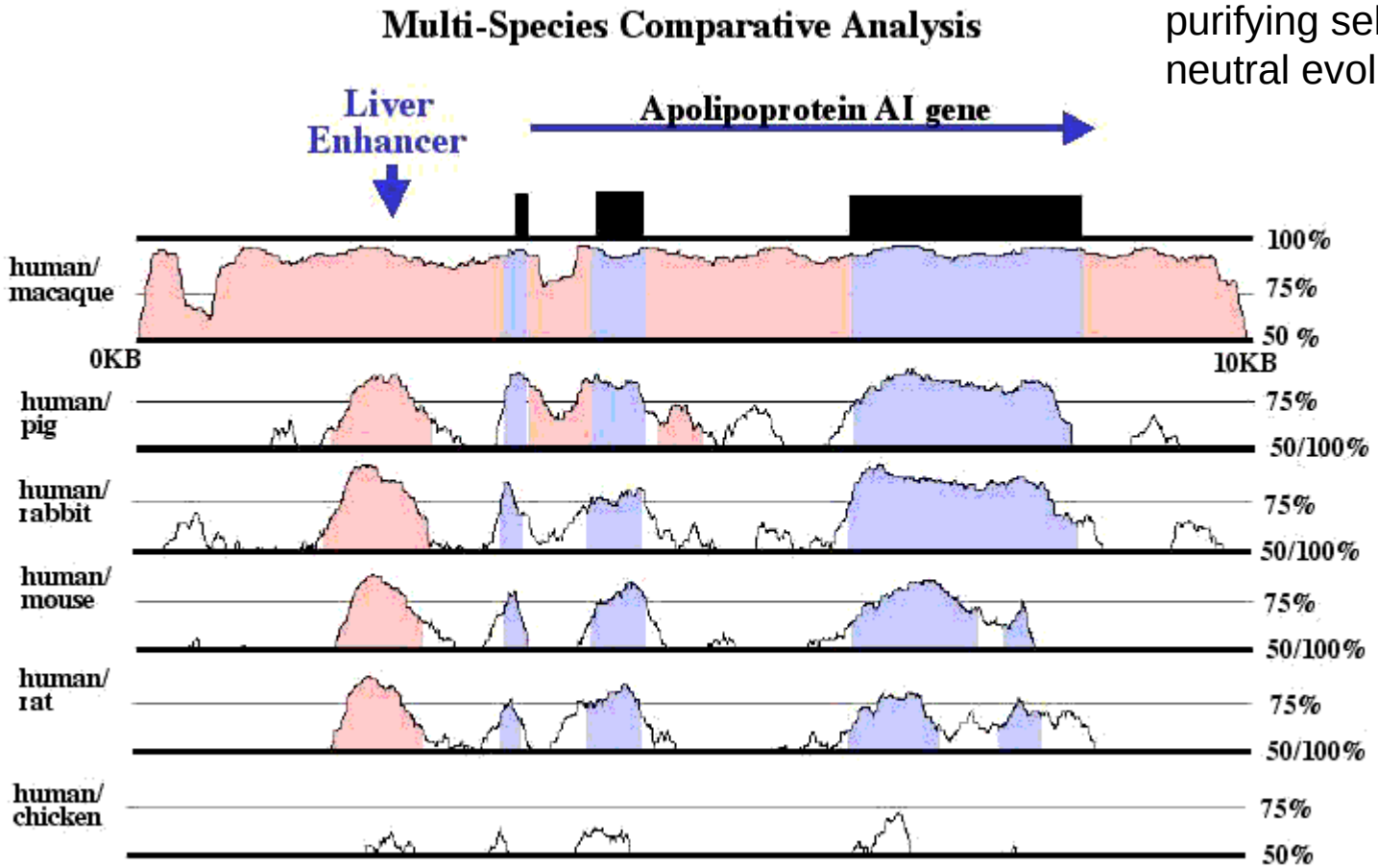
Mutations kill functional elements.

Mutations give rise to new functional elements
(by duplicating existing ones, or creating new ones)

Selection whittles this constant flow of genomic innovations.

Conservation implies function

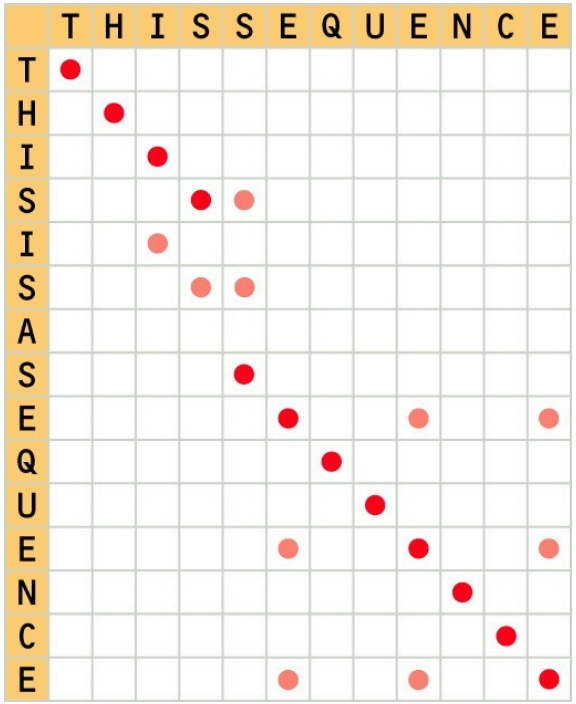
purifying selection vs. neutral evolution

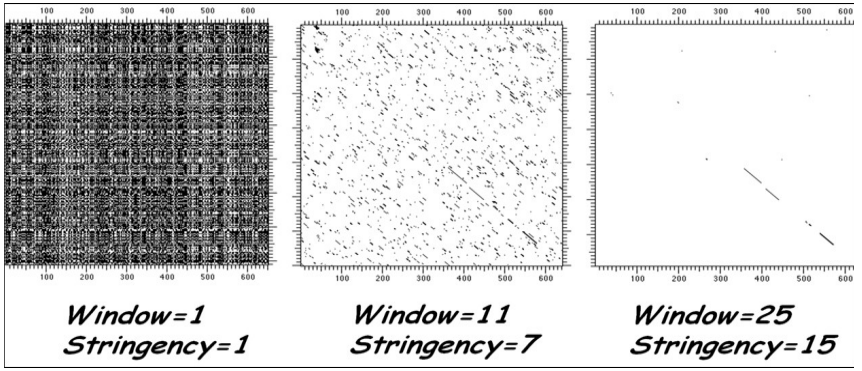


Note: Lack of sequence conservation does NOT imply lack of function.
NOR does it rule out function conservation.

Dotplots

- Dotplots are a simple way of seeing alignments
  - We really like to see good visual demonstrations, not just tables of numbers
- It's a grid: put one sequence along the top and the other down the side, and put a dot wherever they match.
- You see the alignment as a diagonal
- Note that DNA dotplots are messier because the alphabet has only 4 letters
  - Smoothing by windows helps:

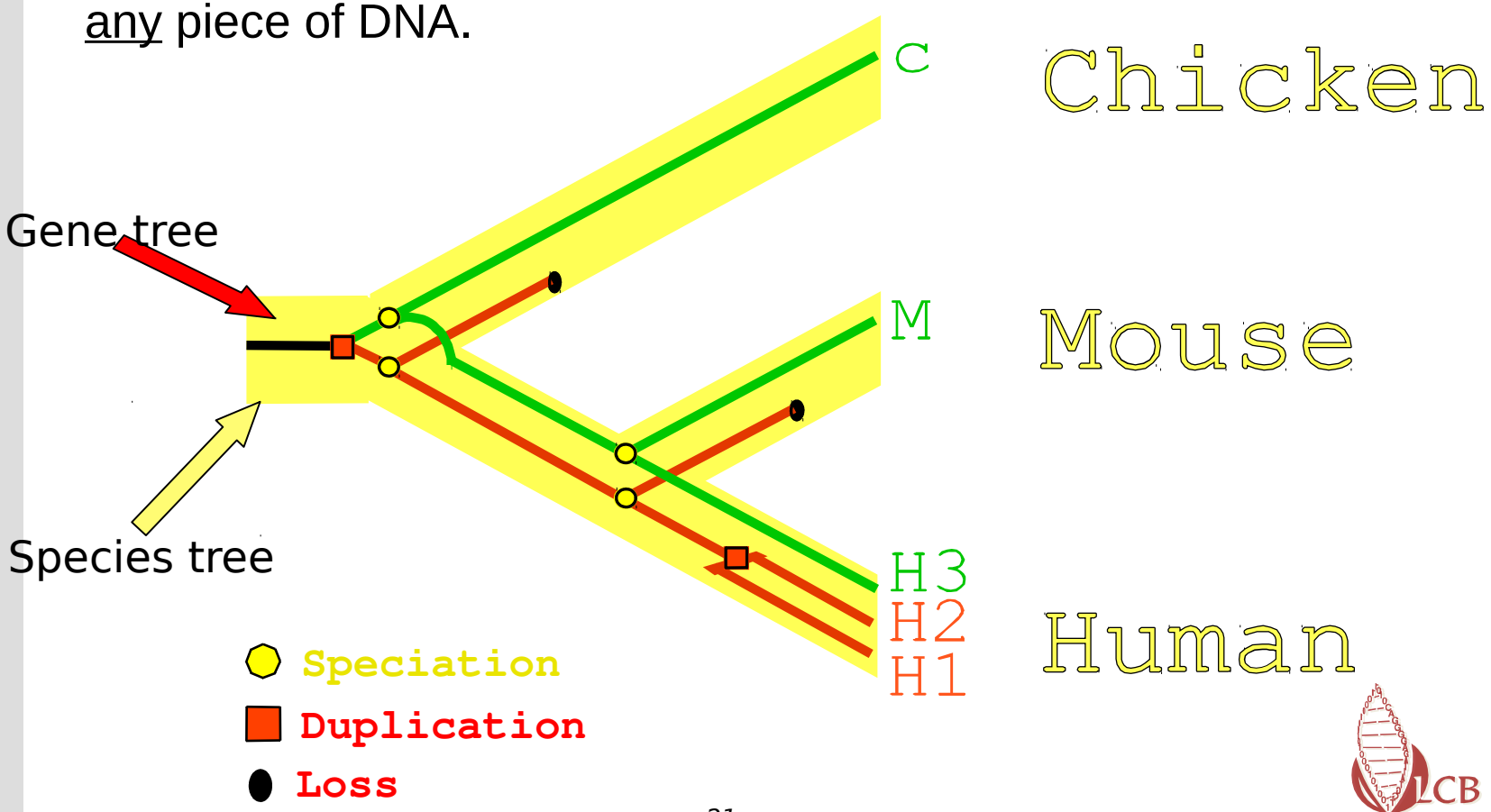




Inferring Genomic Histories
From Alignments of Genomes

A Gene tree evolves with respect to a Species tree

By “Gene” we mean
<u>any</u> piece of DNA.

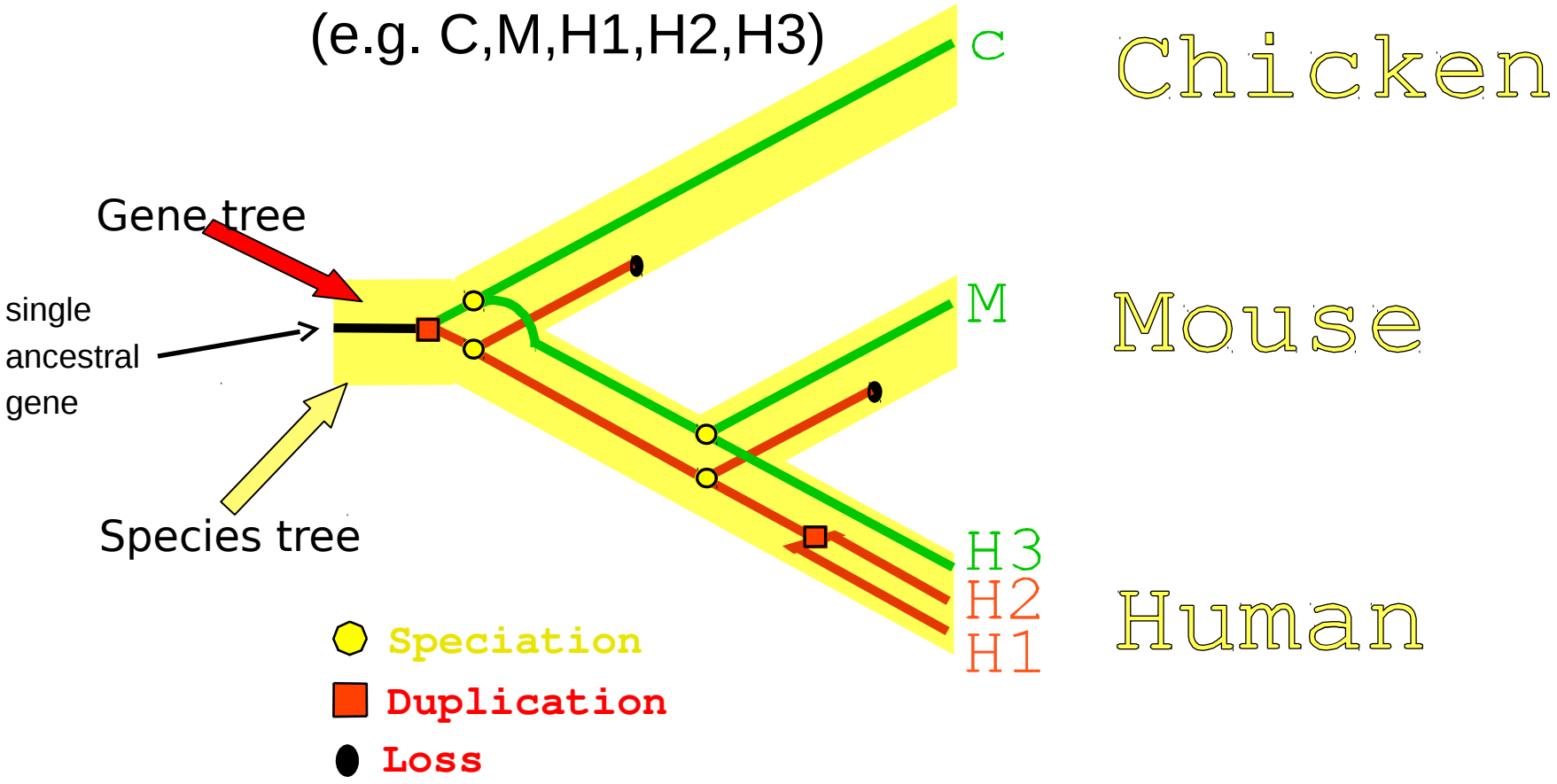


Terminology

Orthologs : Genes related via speciation (e.g. C,M,H3)

Paralogs: Genes related through duplication (e.g. H1,H2,H3)

Homologs: Genes that share a common origin (e.g. C,M,H1,H2,H3)

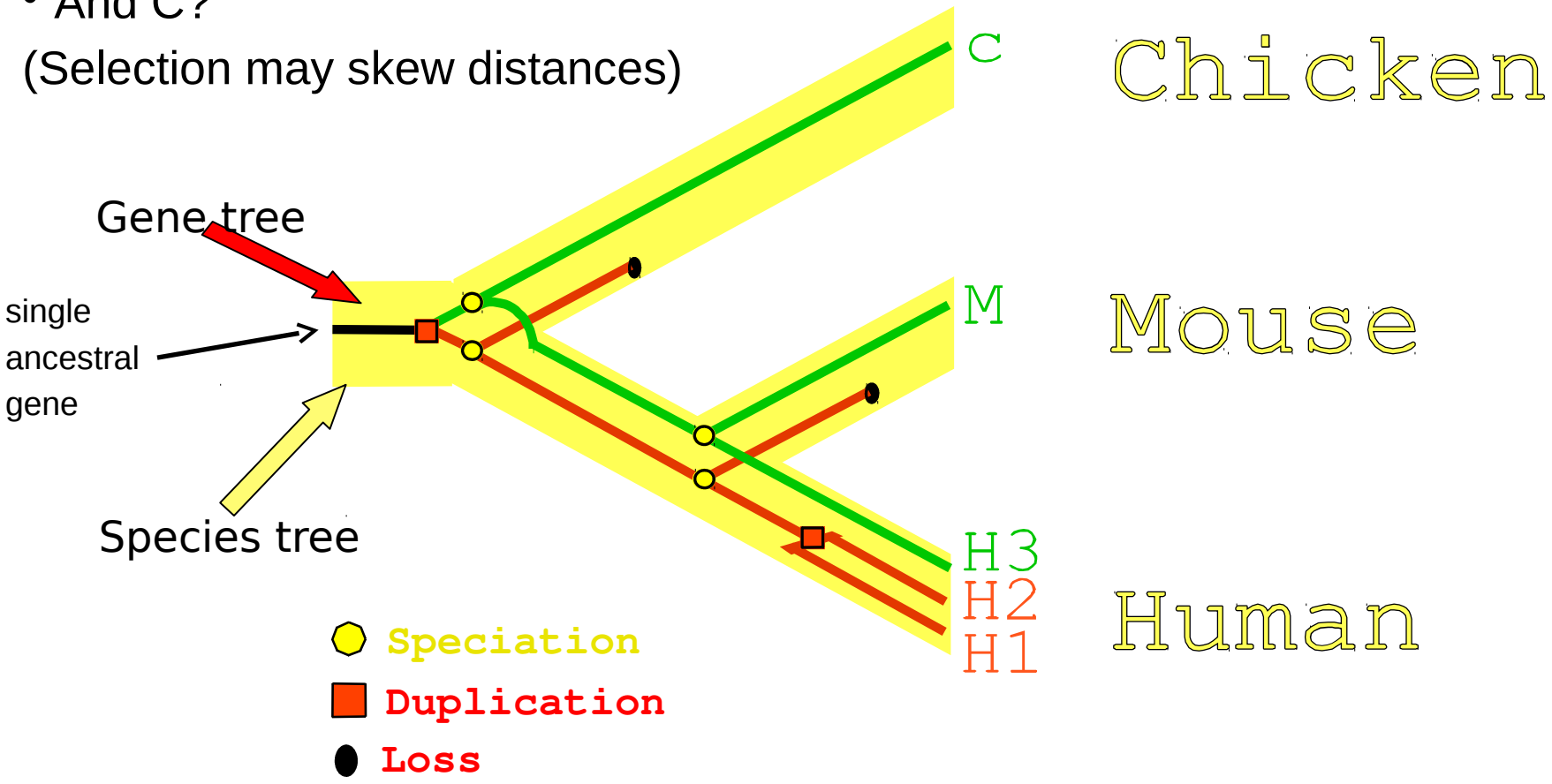


Typical Molecular Distances

If they were evolving at a constant rate:

- To which is H1 closer in sequence, H2 or H3?
- To which H is M closest?
- And C?

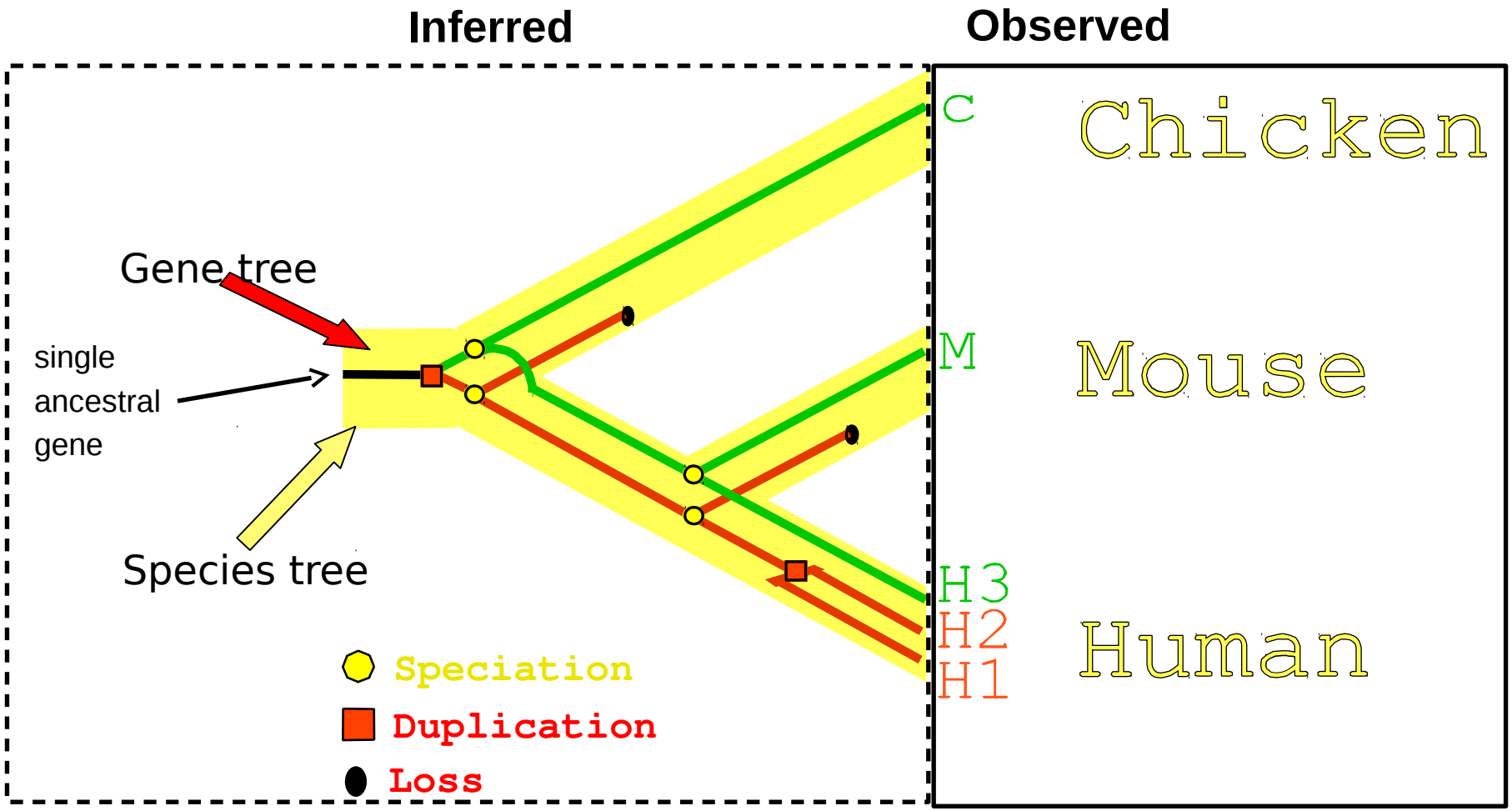
(Selection may skew distances)



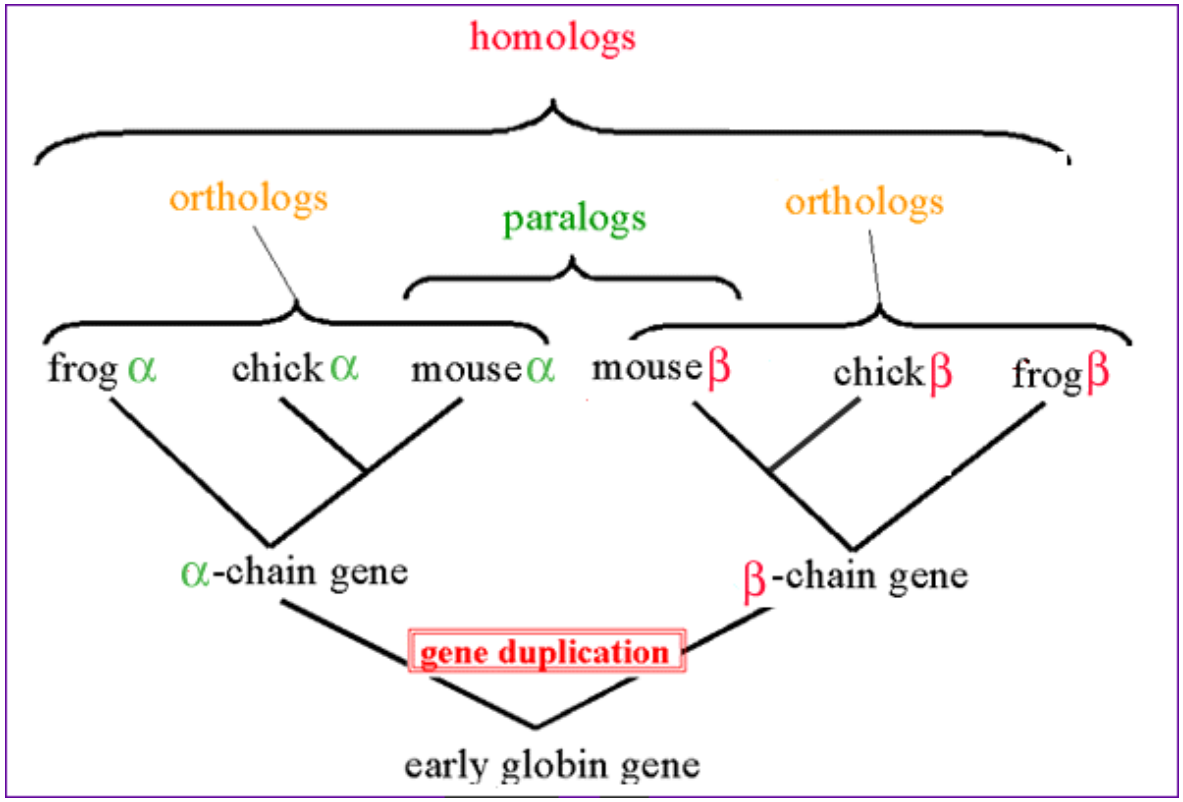
Gene trees and even species trees are figments of our (scientific) imagination

Species trees and gene trees can be wrong.

All we really have are extant observations, and fossils.

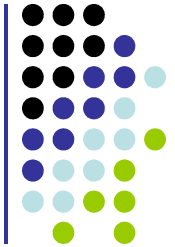


Gene Families



- What?
  - Compare whole genomes
    - Compare two genomes
      - Within (intra) species
      - Between (inter) species
    - Compare genome to itself
  - Search for an element in a genome
- Why?
  - To learn about genome evolution (and phenotype evolution!)
  - Homologous functional regions often have similar functions
  - Modification of functional regions can reveal
    - Neutral and functional regions
    - Disease susceptibility
    - Adaptation
  - And more..
- How?

Sequence Alignment



AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC

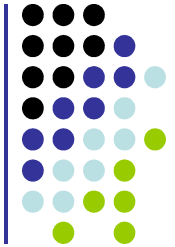
-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTTGCCCGAC

Definition

Given two strings $x = x_1x_2...x_M$, $y = y_1y_2...y_N$,

an alignment is an assignment of gaps to positions 0,..., N in x, and 0,..., N in y, so as to line up each letter in one sequence with either a letter, or a gap in the other sequence

Scoring Function



- Sequence edits:

  AGGCCTC

  - Mutations **AGGACTC**
  - Insertions **AGGGCCTC**
  - Deletions **AGG . CTC**

Alternative definition:

minimal edit distance

"Given two strings x, y, find minimum # of edits (insertions, deletions, mutations) to transform one string to the other"

Scoring Function:

Match: +m
Mismatch: -s
Gap: -d

Cost of edit operations needs to be biologically inspired (eg DEL length).

Solve via Dynamic Programming

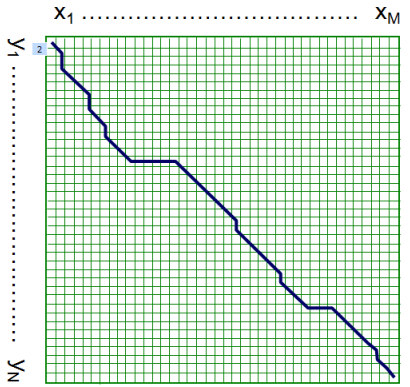
Score F = (# matches) × m - (# mismatches) × s – (#gaps) ×

Are two sequences homologous?

AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC

-**AGG**C**TATCAC**CT**GACC**T**C**CA**GG**C**CGA**--**TGCCC**---
T**AG**-**CTATCAC**--**GACC**G**C**--**GG**T**CGA**TT**TGCCC**GAC

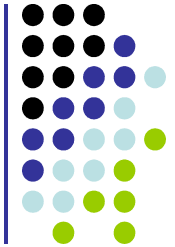
DP matrix:



Given an (optimal) alignment between two genome regions,
you can ask what is the probability that they are (not) related by homology?

Note that (when known) the answer is a function of the molecular distance between the two (eg, between two species)

Sequence Alignment



AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC

-AGG**CTATCAC**C**T****GACC**T**C**C**A****GG**C**CGA****--TGCCC****---**
T**AG-CTATCAC--GACC**G**C--GG**T**CGA**TT**TGCCC**GAC

Similarity is often measured using “%id”, or percent identity

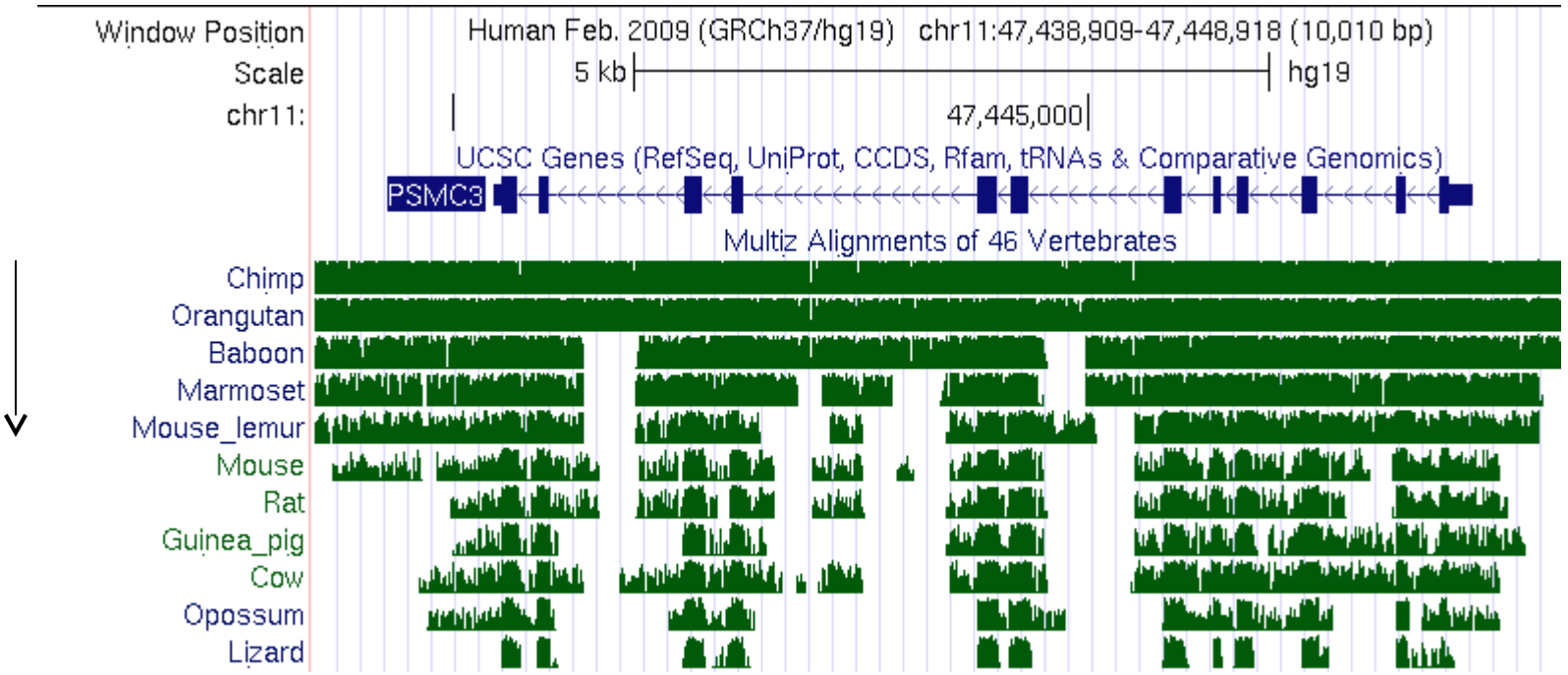
%id = number of matching bases / number of alignment columns

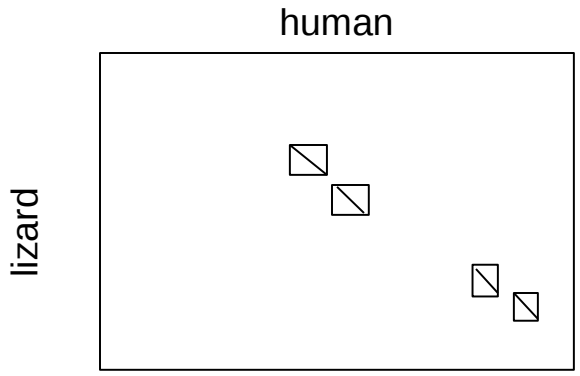
Where

Every alignment column is a match / mismatch / indel base

Where indel = insertion or deletion (requires an outgroup to resolve)

Note the pattern of sequence conservation / divergence



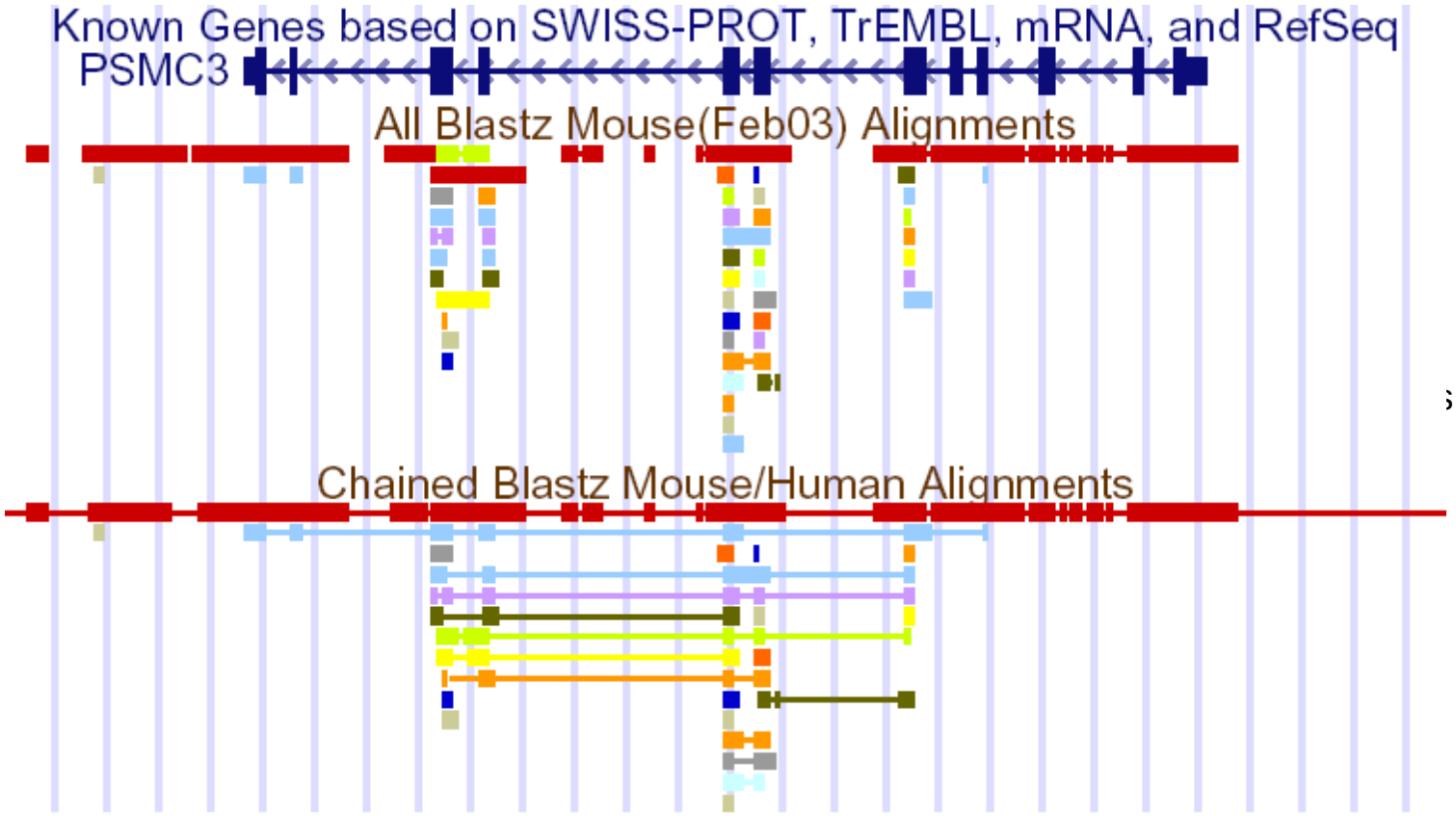


Objective: find local alignment blocks, that are likely homologous (share common origin)

O(mn) examine the full matrix using DP

O(m+n) heuristics based on seeding + extension trades sensitivity for speed

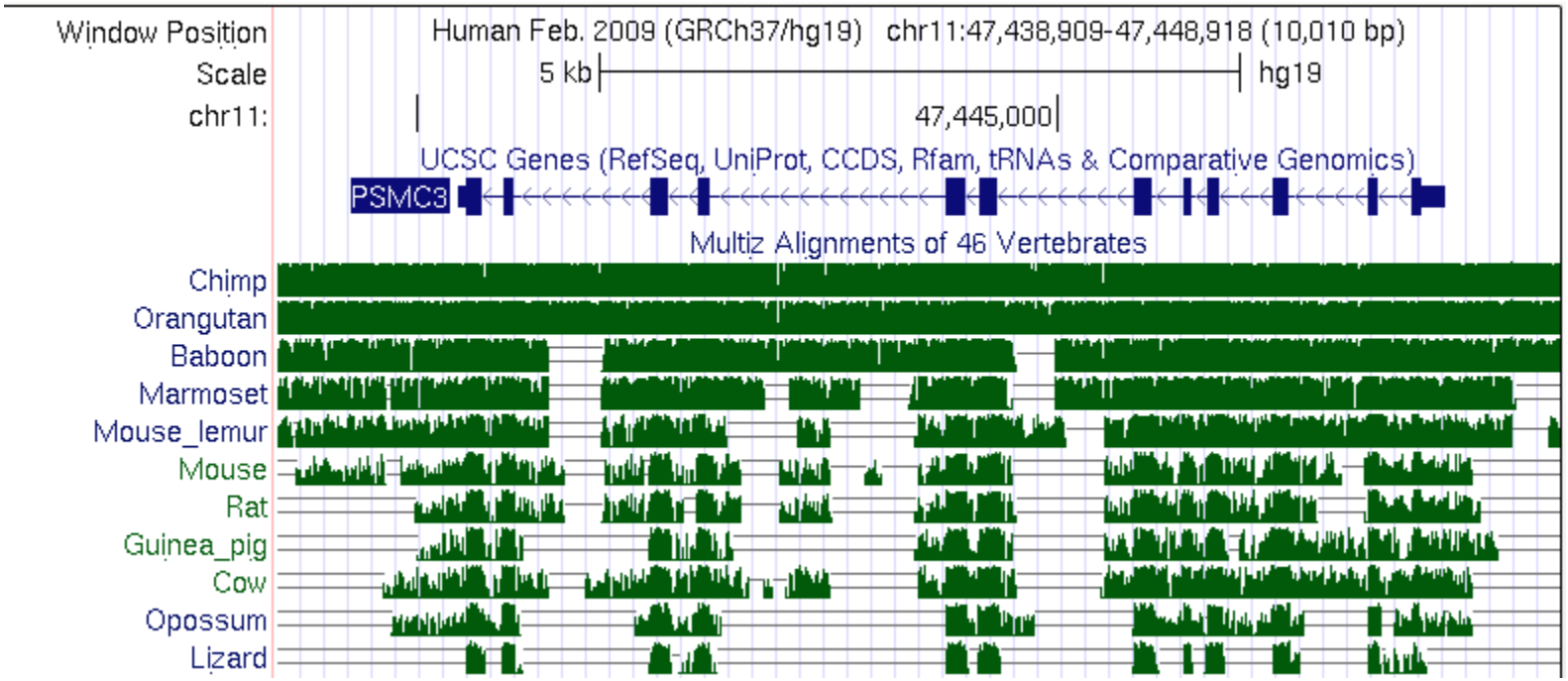
“Raw” (B)lastz track (no longer displayed)

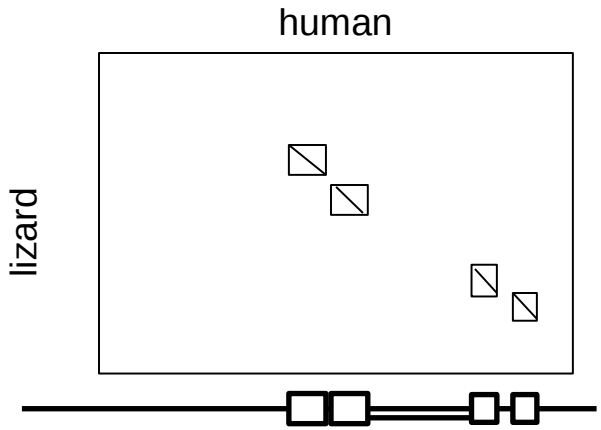


s regions

Protease Regulatory Subunit 3

Chaining co-linear alignment blocks



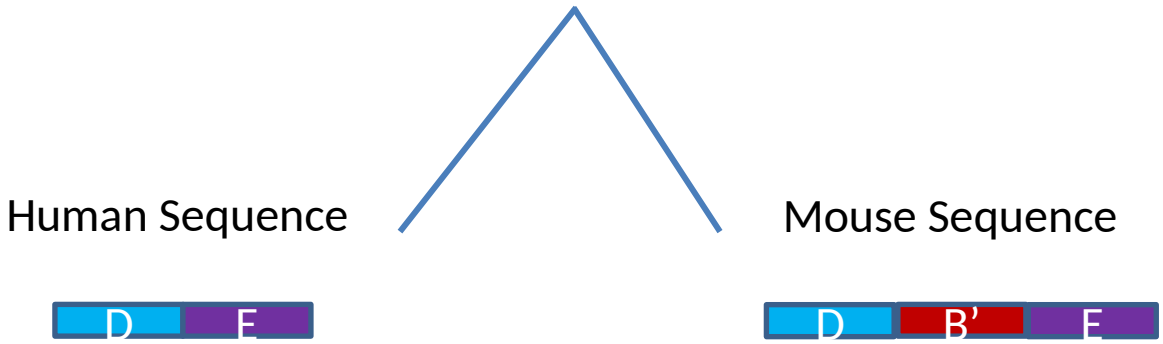


Objective: find local alignment blocks, that are likely homologous (share common origin)

Chaining strings together co-linear blocks in the target genome to which we are comparing.

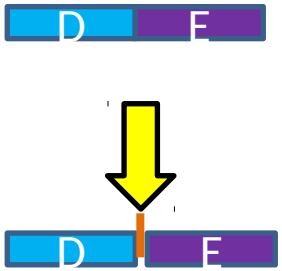
Double lines when there is unalignable sequence in the other species. Single lines when there isn't.

Gap Types: Single vs Double sided



In Human Browser

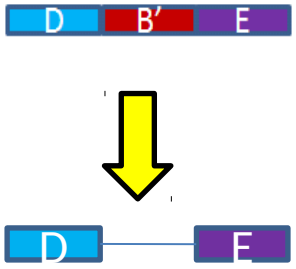
Human sequence



Mouse homology

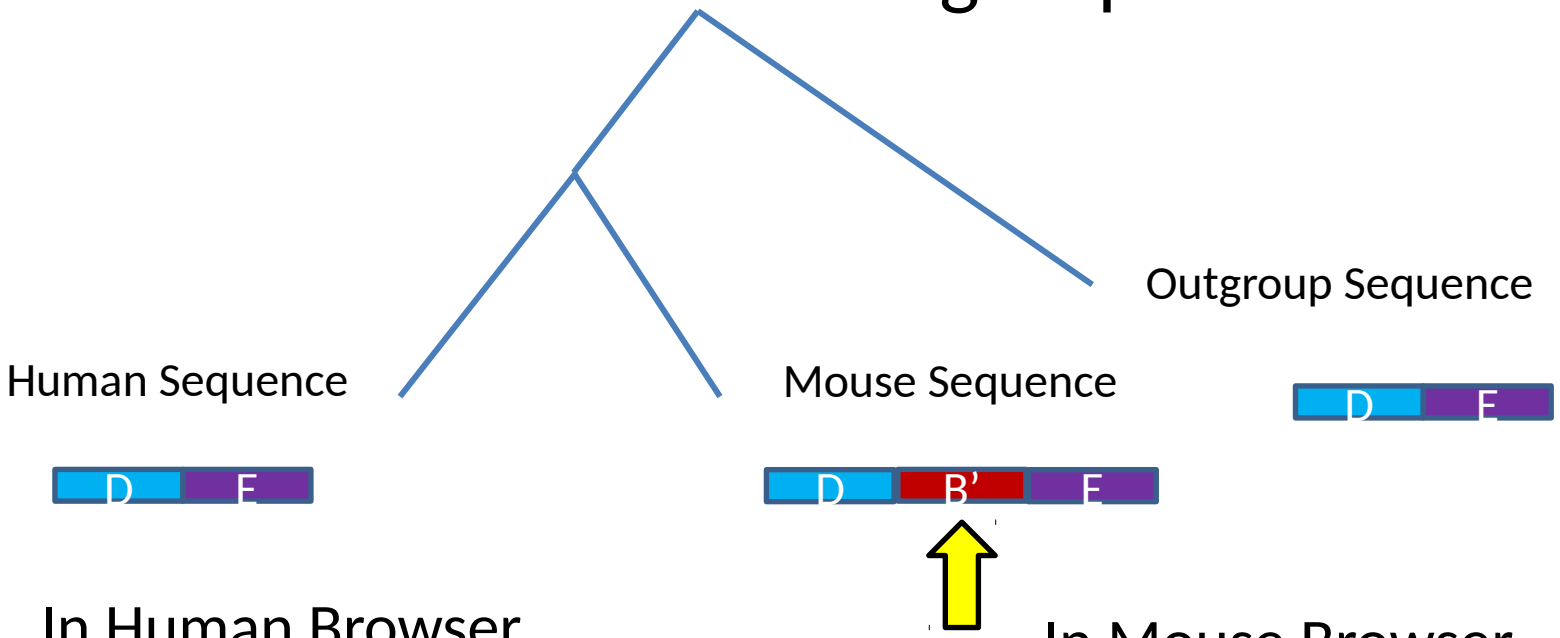
In Mouse Browser

Mouse sequence



Human homology

Did Mouse insert or Human delete?
The Need for an Outgroup



In Human Browser

In Mouse Browser

Human sequence

Mouse sequence

Mouse homology

Human homology

Conservation Track Documentation

Gap Annotation

The *Display chains between alignments* configuration option enables display of gaps between alignment blocks in the pairwise alignments in a manner similar to the Chain track display. The following conventions are used:

- **Single line:** no bases in the aligned species. Possibly due to a lineage-specific insertion between the aligned blocks in the human genome or a lineage-specific deletion between the aligned blocks in the aligning species.
- **Double line:** aligning species has one or more unalignable bases in the gap region. Possibly due to excessive evolutionary distance between species or independent indels in the region between the aligned blocks in both species.
- **Pale yellow coloring:** aligning species has Ns in the gap region. Reflects uncertainty in the relationship between the DNA of both species, due to lack of sequence in relevant portions of the aligning species.

Genomic Breaks

Discontinuities in the genomic context (chromosome, scaffold or region) of the aligned DNA in the aligning species are shown as follows:

- **Vertical blue bar:** represents a discontinuity that persists indefinitely on either side, *e.g.* a large region of DNA on either side of the bar comes from a different chromosome in the aligned species due to a large scale rearrangement.
- **Green square brackets:** enclose shorter alignments consisting of DNA from one genomic context in the aligned species nested inside a larger chain of alignments from a different genomic context. The alignment within the brackets may represent a short misalignment, a lineage-specific insertion of a transposon in the human genome that aligns to a paralogous copy somewhere else in the aligned species, or other similar occurrence.

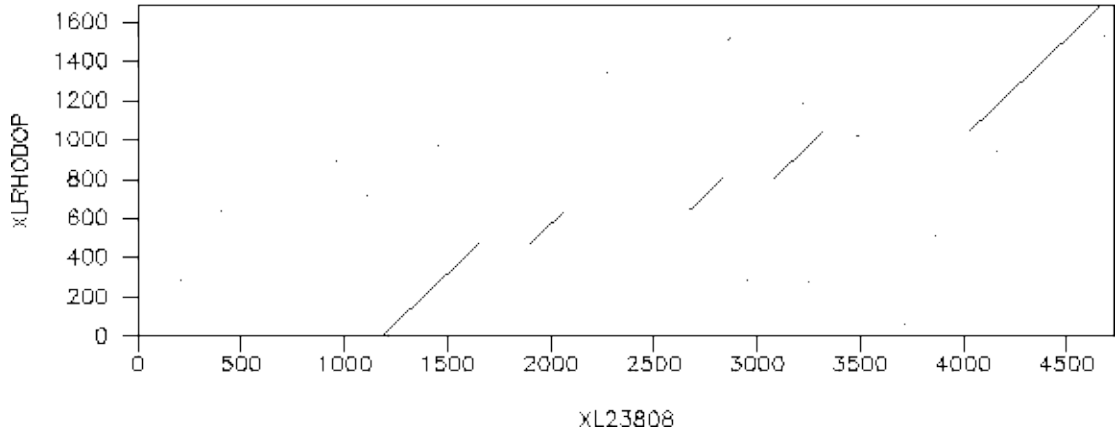
Chaining Alignments

Chaining highlights homologous regions between genomes, bridging the gulf between syntenic blocks and base-by-base alignments.

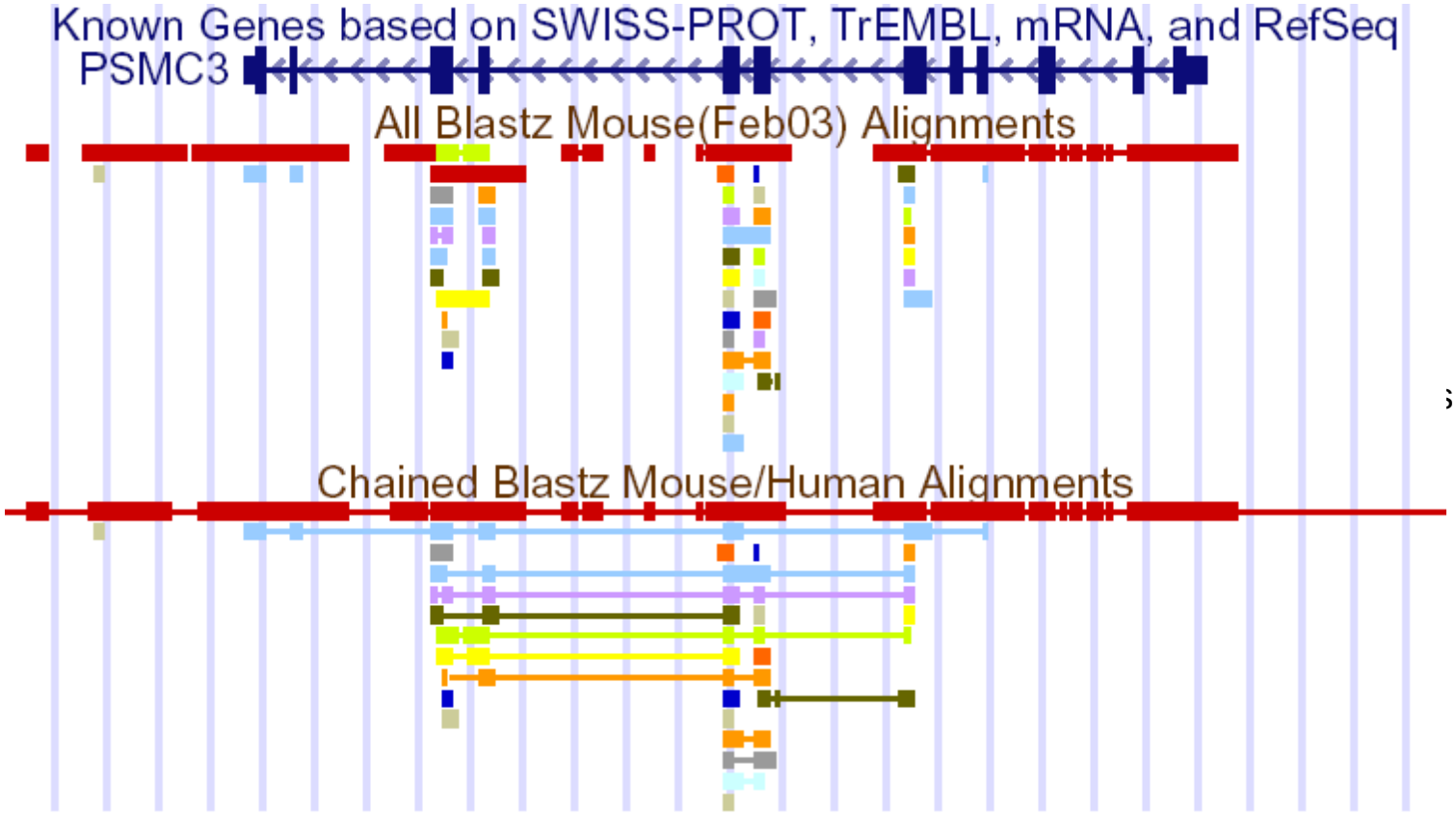
Local alignments tend to break at transposon insertions, inversions, duplications, etc.

Global alignments tend to force non-homologous bases to align.

Chaining is a rigorous way of joining together local alignments into larger structures.



“Raw” (B)lastz track (no longer displayed)



s regions

Protease Regulatory Subunit 3

Chains & Nets: How they're built

- 1: Blastz one genome to another
  - **Local** alignment algorithm
  - Finds short blocks of similarity

```
Hg18:   AAAAAACCCCCAAAAA
Mm8:    AAAAAAGGGGG
```

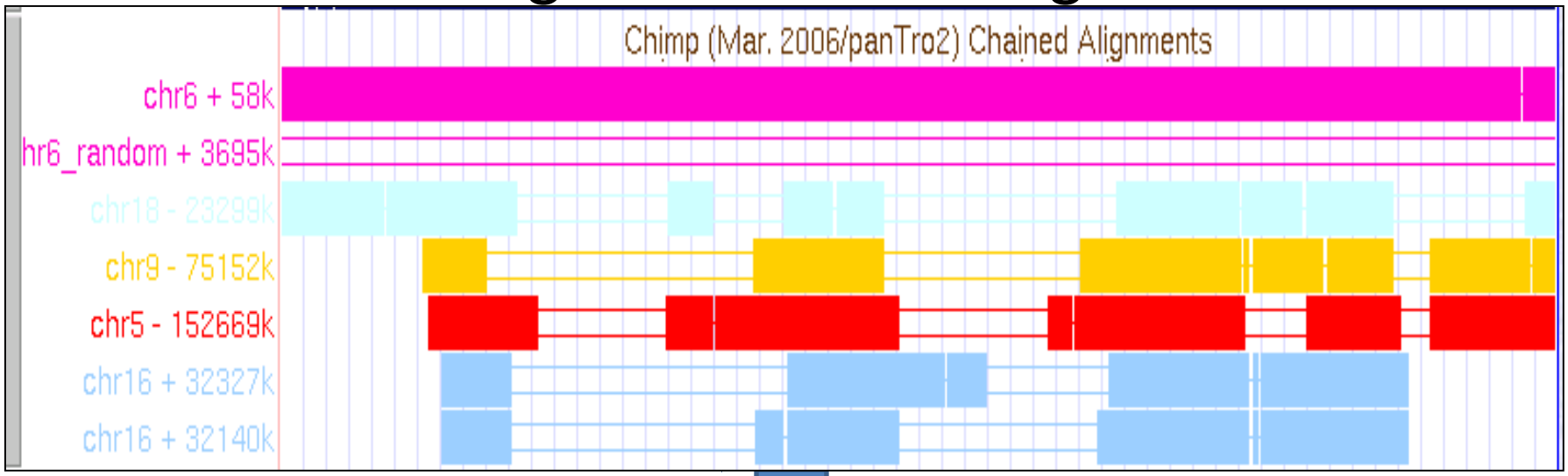
Hg18.1-6 + AAAAAA
Mm8.1-6 + AAAAAA

Hg18.7-11 + CCCCC
Mm8.1-5 - CCCCC

Hg18.12-16 + AAAAA
Mm8.1-5 + AAAAA

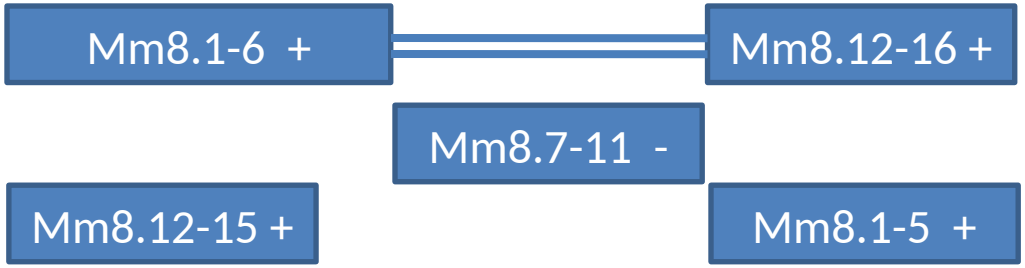
Chains & Nets: How they're built

- 2: "Chain" alignment blocks together

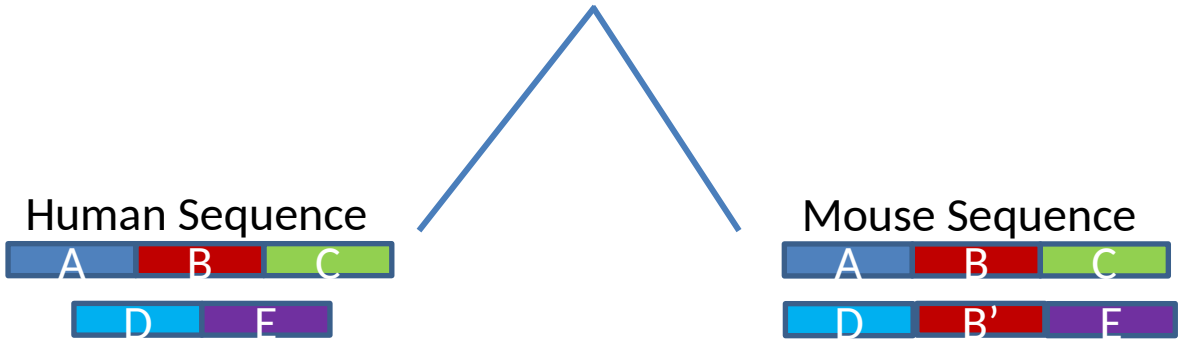


Hg18: AAAAAACCCCCAAAAA

Mm8 chains



Another Chain Example



In Human Browser

Implicit Human sequence

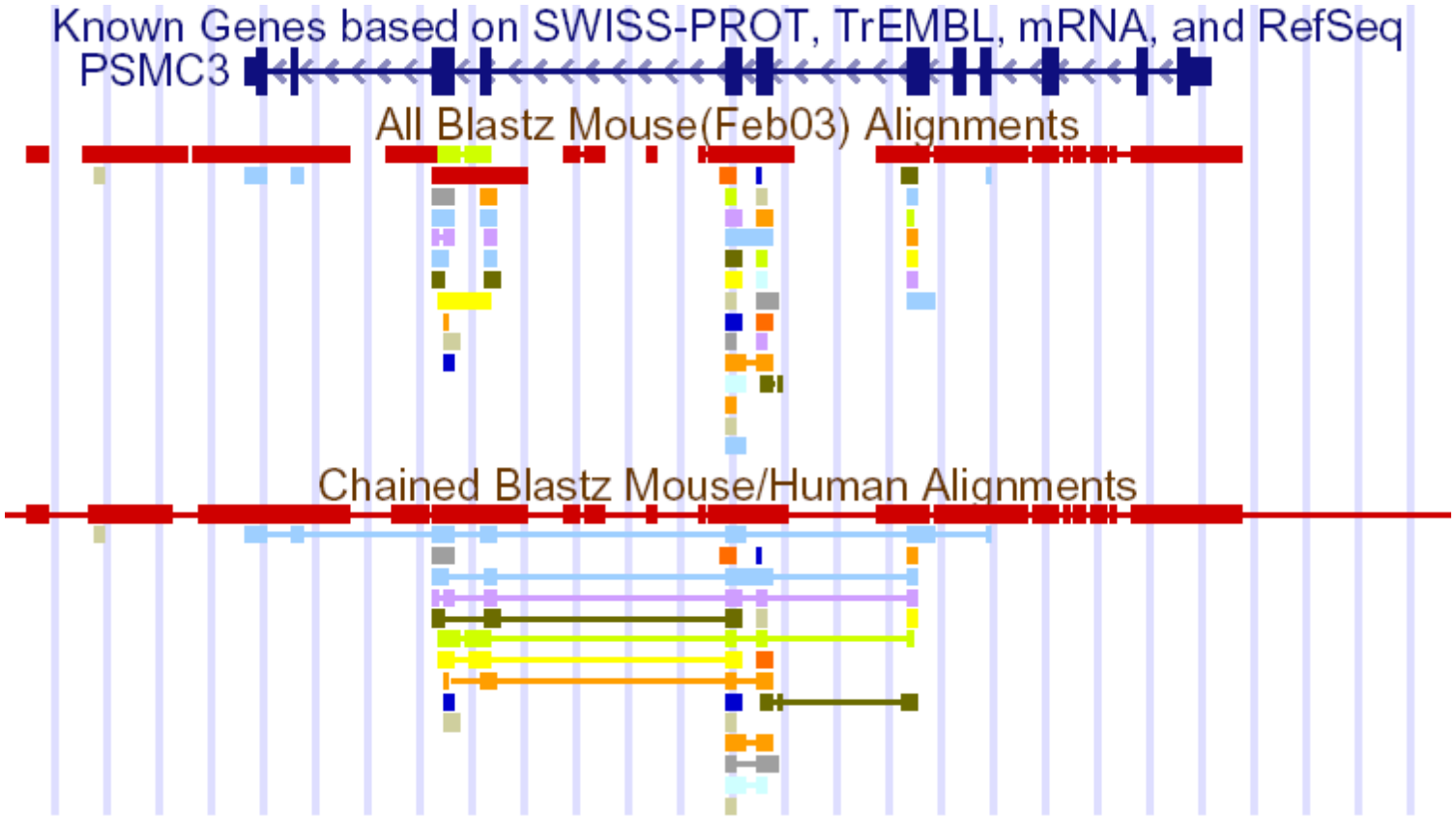
Mouse chains

In Mouse Browser

Implicit Mouse sequence

Human chains

Chains join together related local alignments



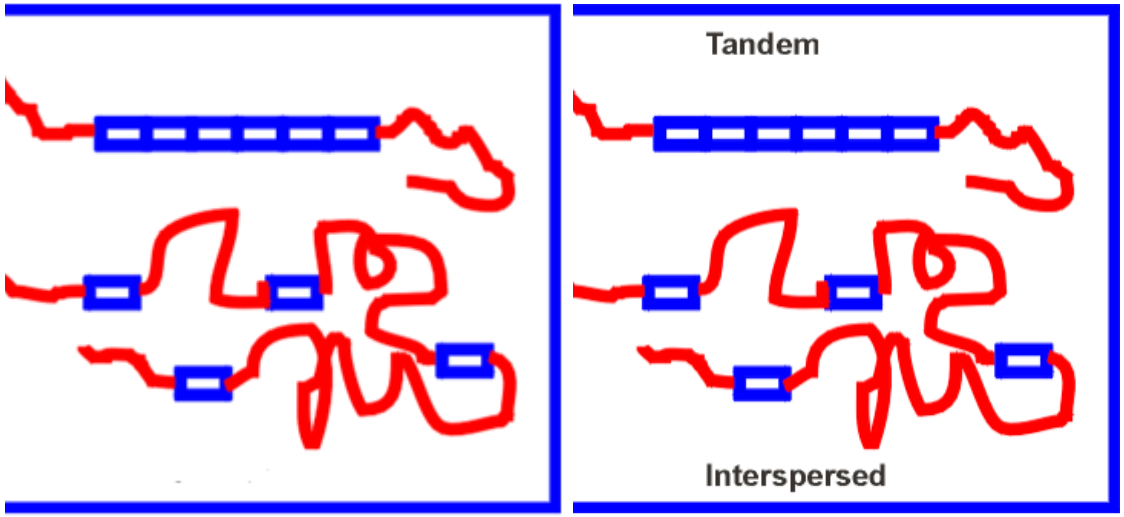
Protease Regulatory Subunit 3

Interspersed vs. Simple Repeats

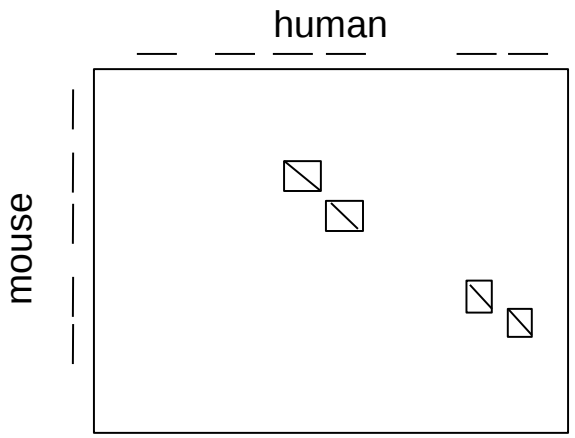
From an evolutionary point of view transposons and simple repeats are very different.

Different instances of the same transposon share common ancestry (but not necessarily a direct common progenitor).

Different instances of the same simple repeat most often do not.



Note: repeats are a nuisance



If, for example, human and mouse have each 10,000 copies of the same repeat:

We will obtain and need to output 10^8 alignments of all these copies to each other.

Note that for the sake of this comparison interspersed repeats and simple repeats are equal nuisances.

However, note that simple repeats, but not interspersed repeats, violate the assumption that similar sequences are homologous.

Solution:

1 Discover all repetitive sequences in each genome.

2 Mask them when doing genome to genome comparison.

3 Chain your alignments.

4 Add back to the alignments only repeat matches that lie within pre-computed chains.

This re-introduces back into the chains (mostly) orthologous copies. (which is valuable!)

Chains

- a chain is a sequence of gapless aligned blocks, where there must be no overlaps of blocks' target or query coords within the chain.
- Within a chain, target and query coords are monotonically non-decreasing. (i.e. always increasing or flat)
- double-sided gaps are a new capability (blastz can't do that) that allow extremely long chains to be constructed.
- not just orthologs, but paralogs too, can result in good chains. but that's useful!
- chains should be symmetrical -- e.g. swap human-mouse -> mouse-human chains, and you should get approx. the same chains as if you chain swapped mouse-human blastz alignments.
- chained blastz alignments are not single-coverage in either target or query unless some subsequent filtering (like netting) is done.
- chain tracks can contain massive pileups when a piece of the target aligns well to many places in the query. Common causes of this include insufficient masking of repeats and high-copy-number genes (or paralogs).

[Angie Hinrichs, UCSC wiki]

Before and After Chaining

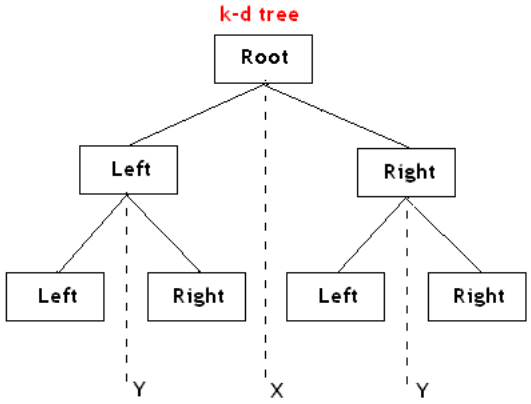
	Blastz	AxtChain	Net Top
Longest	63 k	115 M	115 M
Average	608	23 k	147 k
Count	8.6 M	147 k	20 k

Chaining Algorithm

Input - blocks of gapless alignments from (b)lastz
Dynamic program based on the recurrence relationship:

$$score(B_i) = \max_{j<i}(score(B_j) + match(B_i) - gap(B_i, B_j))$$

Uses Miller's KD-tree algorithm to minimize which parts of dynamic programming graph to traverse. Timing is O(N logN), where N is number of blocks (which is in hundreds of thousands)



See [Kent et al, 2003]
"Evolution's cauldron: Duplication, deletion, and rearrangement in the mouse and human genomes"

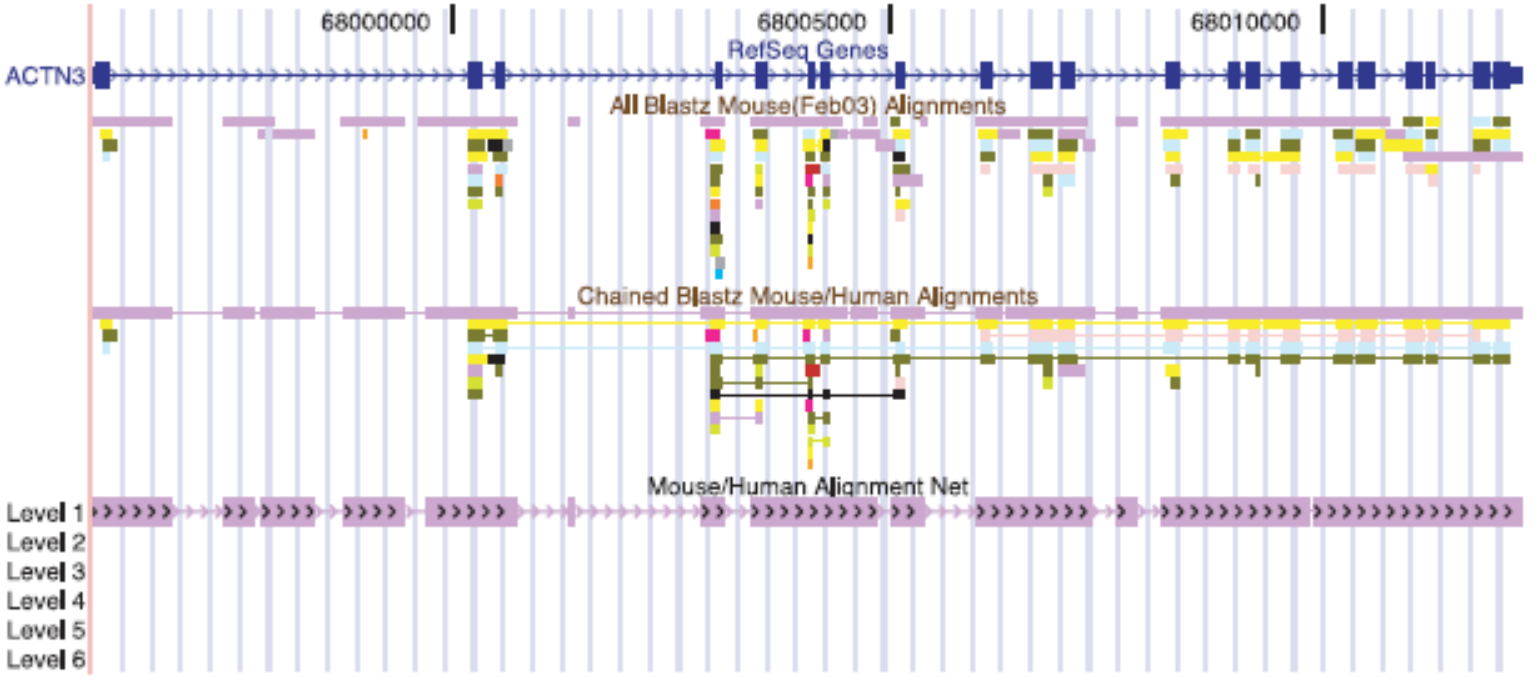
Netting Alignments

Commonly multiple mouse alignments can be found for a particular human region, eg including for most coding regions.

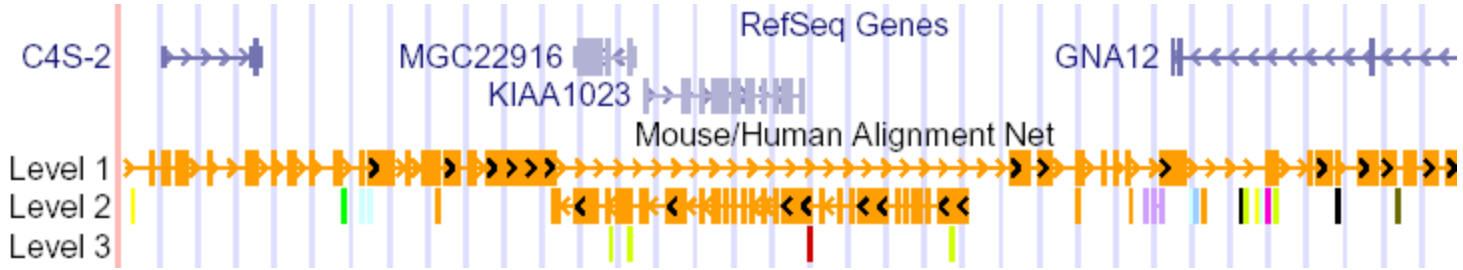
Net finds best match mouse match for each human region.

Highest scoring chains are used first.

Lower scoring chains fill in gaps within chains inducing a natural hierarchy.

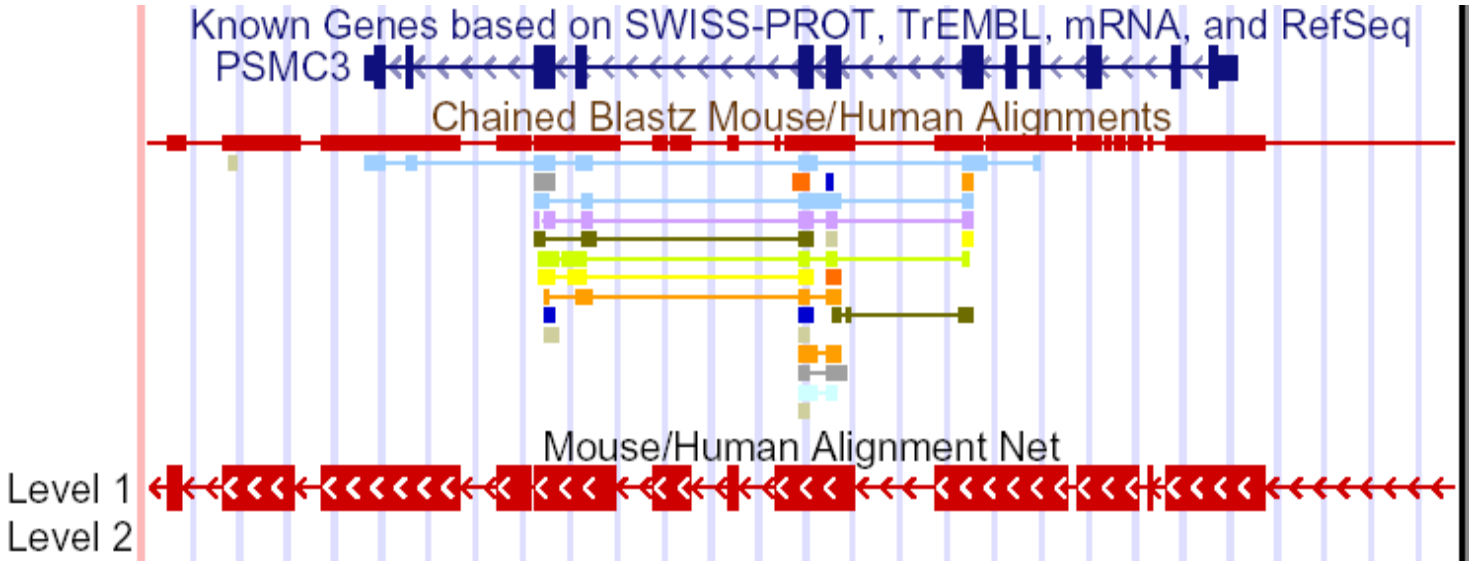


Net highlights rearrangements



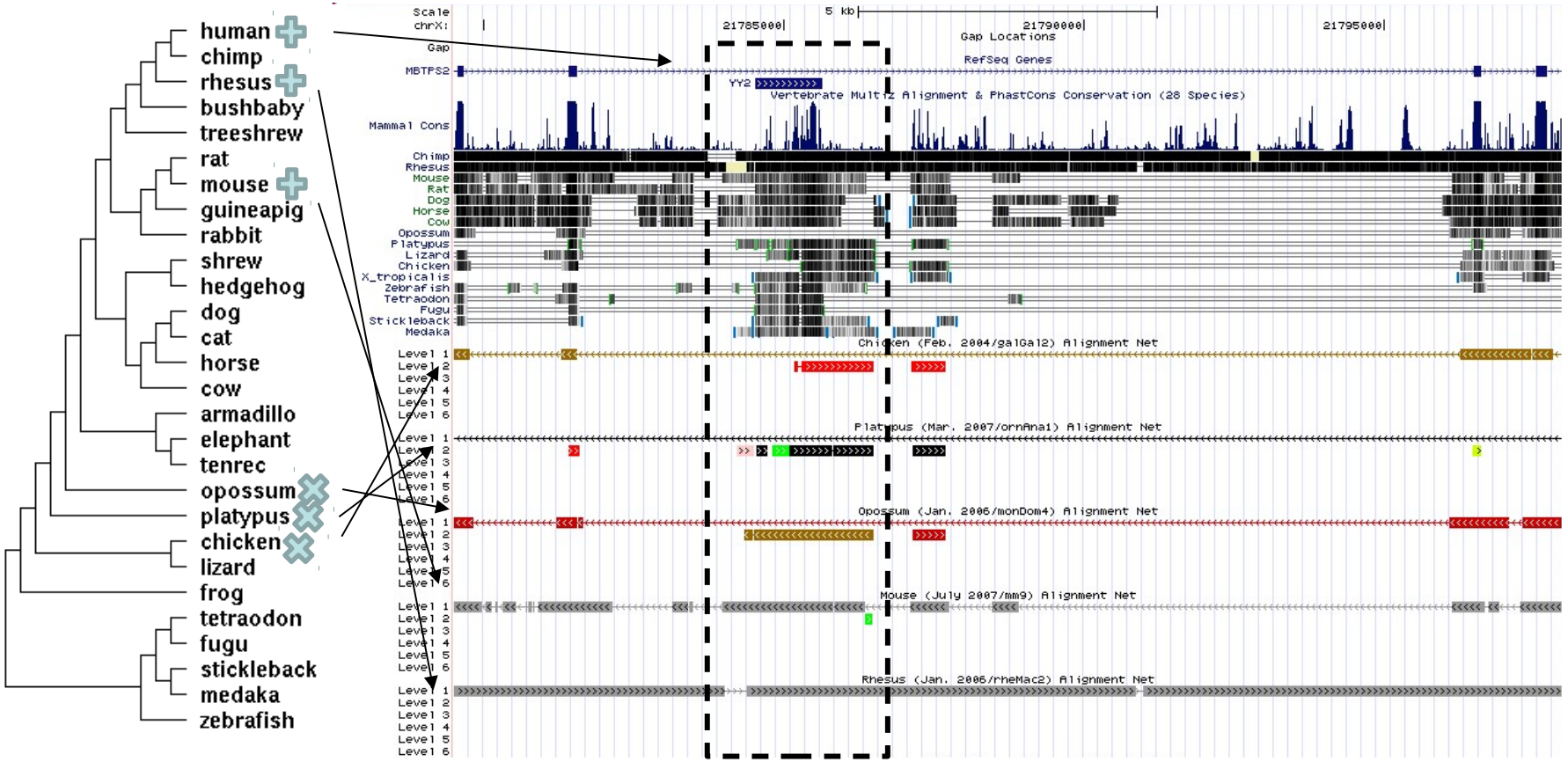
A large gap in the top level of the net is filled by an inversion containing two genes. Numerous smaller gaps are filled in by local duplications and processed pseudo-genes.

Nets attempt to computationally capture orthologs



(they also hide everything else)

Nets/chains can reveal retrogenes (and when they jumped in!)



Nets

- a net is a hierarchical collection of chains, with the highest-scoring non-overlapping chains on top, and their gaps filled in where possible by lower-scoring chains, for several levels.
- a net is single-coverage for target but not for query.
- because it's single-coverage in the target, it's no longer symmetrical.
- the netter has two outputs, one of which we usually ignore: the target-centric net in query coordinates. The reciprocal best process uses that output: the query-referenced (but target-centric / target single-cov) net is turned back into component chains, and then those are netted to get single coverage in the query too; the two outputs of that netting are reciprocal-best in query and target coords. Reciprocal-best nets are symmetrical again.
- nets do a good job of filtering out massive pileups by collapsing them down to (usually) a single level.
- **GB: for human inspection <u>always</u> prefer looking at the chains!**

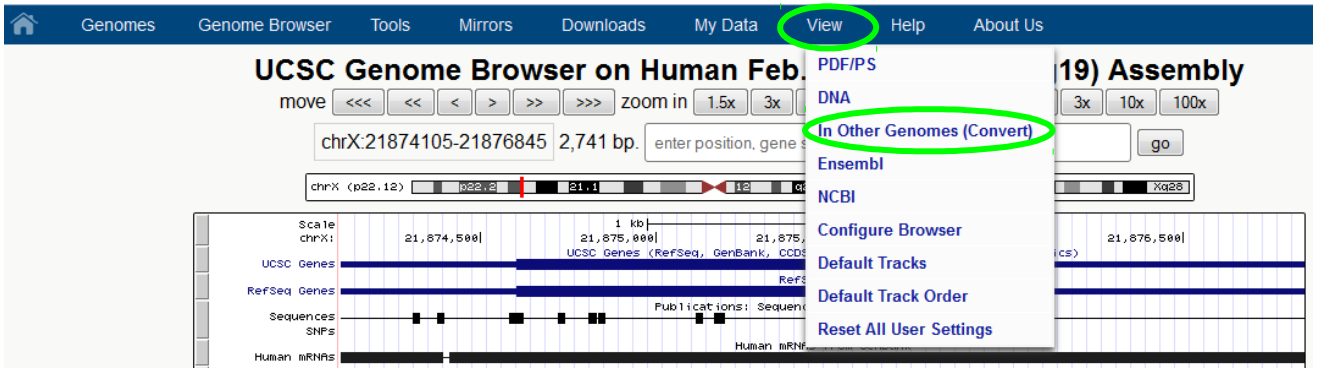
[Angie Hinrichs, UCSC wiki]

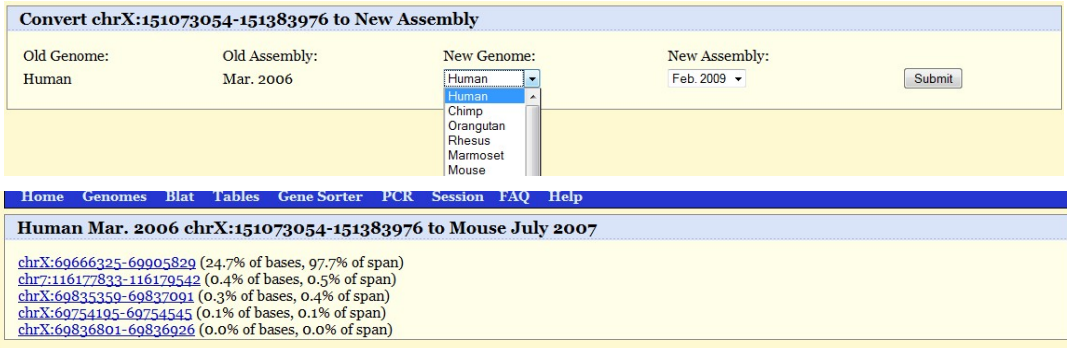
Before and After Netting

	Blastz	AxtChain	Net Top
Longest	63 k	115 M	115 M
Average	608	23 k	147 k
Count	8.6 M	147 k	20 k

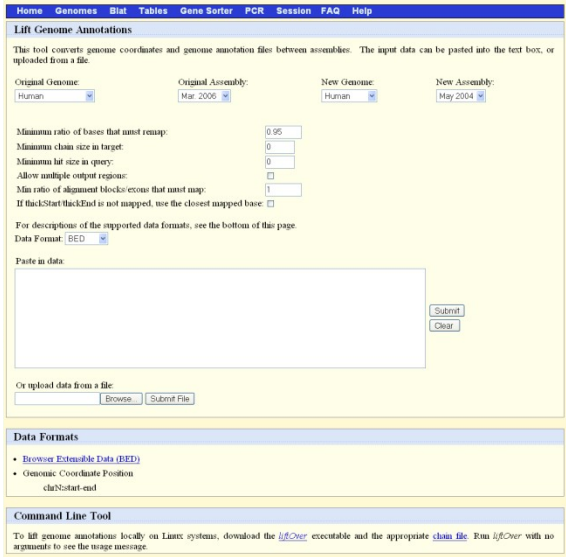
Convert / LiftOver

"LiftOver chains" are actually chains extracted from nets, or chains filtered by the netting process.



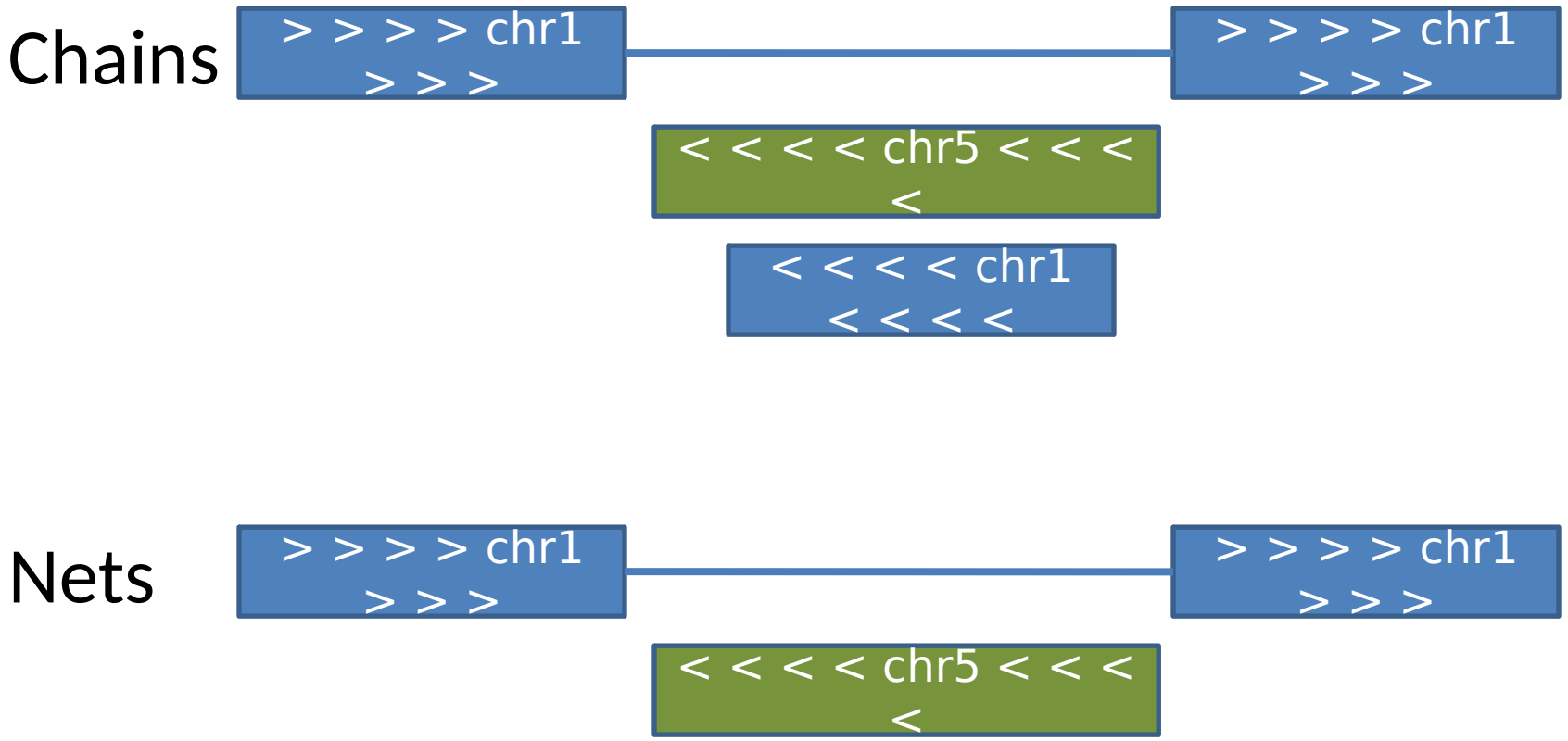


LiftOver – batch utility

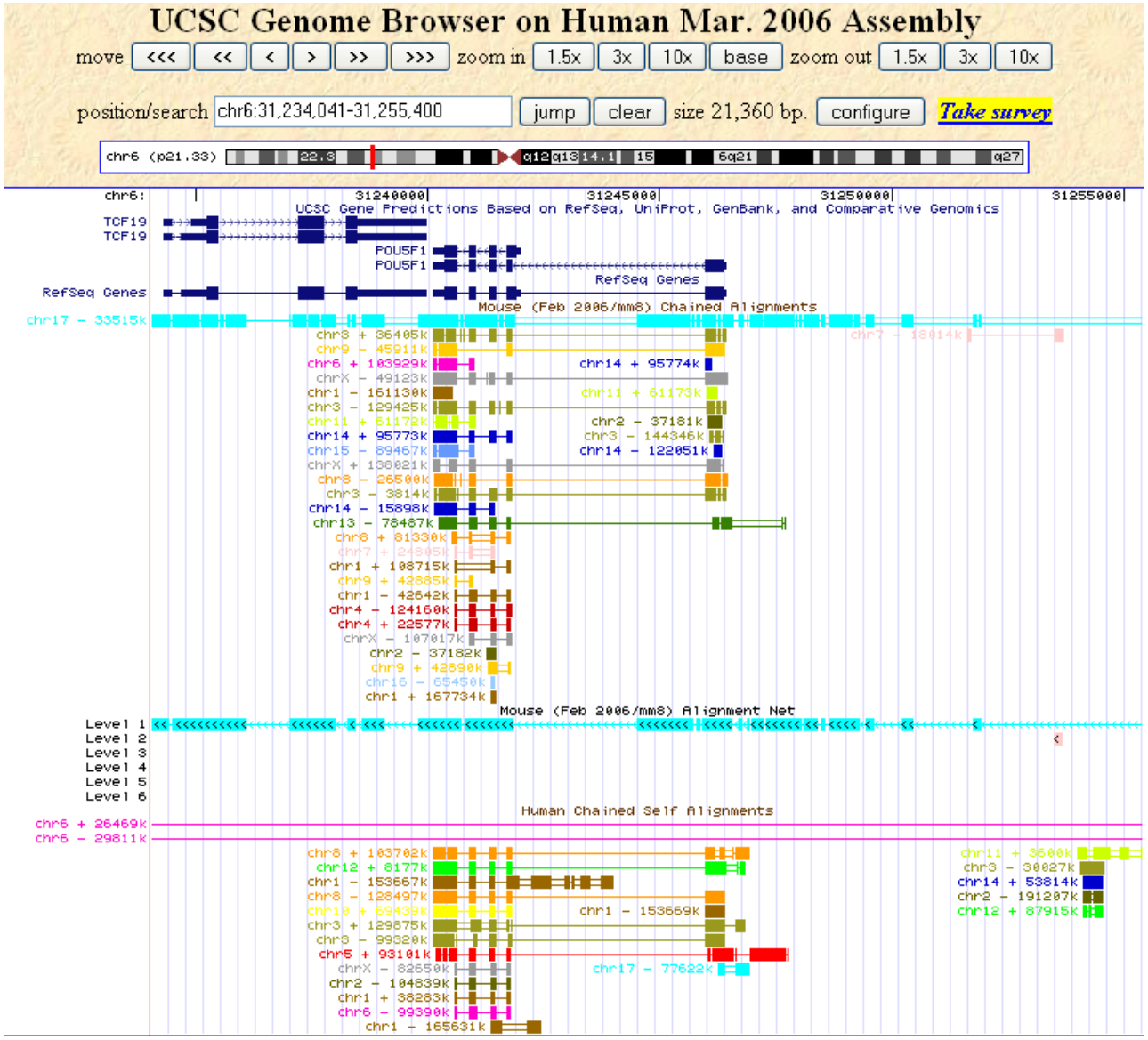


Drawbacks

- Inversions not handled optimally



Self Chain reveals paralogs



(self net is meaningless)