Efficient Statistical Learning from "Big Science" Data

Andrew Moore
Auton Lab

Andrew Connolly
U. Pittsburgh

Jeremy Kubica
Auton Lab

Ting Liu
Auton Lab

The Auton Lab
School of Computer Science
Carnegie Mellon University
www.autonlab.org

The Auton Lab

Faculty:	Andrew Moore (Prof), Jeff Schneider (Research Scientist), Artur Dubrawski (Systems Scientist)
Postdoctoral Fellows:	Brigham Anderson, Alexander Gray, Paul Komarek, Dan Pelleg
Graduate Students:	Brent Bryan, Kaustav Das, Khalid El-Arini, Anna Goldenberg, Jeremy Kubica, Ting Liu, Daniel Neill, Sajid Siddiqi, Purna Sarkar, Ajit Singh, Weng-Keen Wong
Head of Software Development:	Jeanie Komarek
Programmers:	Patrick Choi, Adam Goode, Pat Gunn, Joey Liang, John Ostlund, Robin Sabhnani, Rahul Sankathar
Executive Assistant:	Kristen Schrauder
Head of Sys. Admin:	Jacob Joseph
Undergraduate and Masters Interns:	Kenny Daniel, Sandy Hsu, Dongryeol Lee, Jennifer Lee, Avilay Parekh, Chris Rotella, Jonathan Terleski
Recent Alumni:	Drew Bagnell (RI faculty), Scott Davies (Google), David Cohn (Google), Geoff Gordon (CMU), Paul Hsiung (USC), Marina Meila (U. Washington), Remi Munos (Ecole Polytechnique), Malcolm Strens (Qinetiq)

Current Sponsors

- National Science Foundation (NSF)
- NASA
- Defense Advanced Research Projects Agency (DARPA)
- Department of Homeland Security (DHS)
- Homeland Security Advanced Research Projects Agency (HSARPA)
- United States Department of Agriculture (USDA)
- State of Pennsylvania
- Pfizer Corporation
- Caterpillar Corporation
- British Petroleum
- Psychogenics Corporation
- Transform Pharmaceuticals
- Health Canada

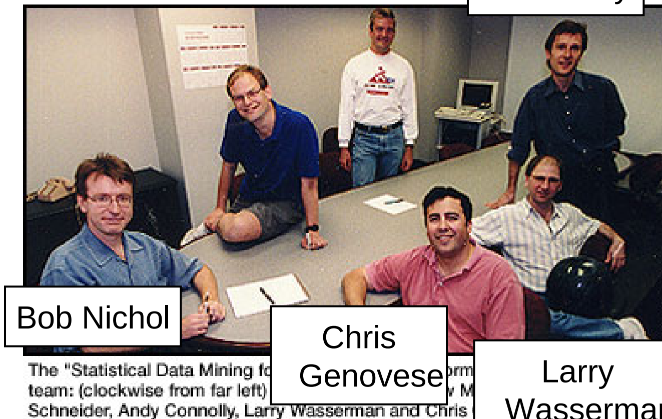
Collaborators...

Ting Liu

Paul Hsiung

Jeff Schneider

Andy Connolly



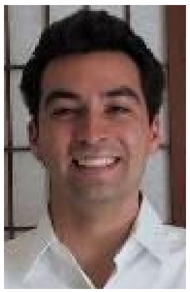
Bob Nichol

Chris Genovese

Larry Wasserman

Istvan Szapudi

Alex Szalay



Alex Gray

Kan Deng

Brigham Anderson

Anna Goldenberg

Greg Cooper

Mike Wagner

Andrew Lawson

Rich Tsui

Daniel Neill

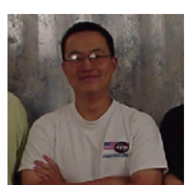
Jeremy Kubica

Scott Davies

Ajit Singh

Paul Komarek

Drew Bagnell



Weng-Keen Wong

Dan Pelleg

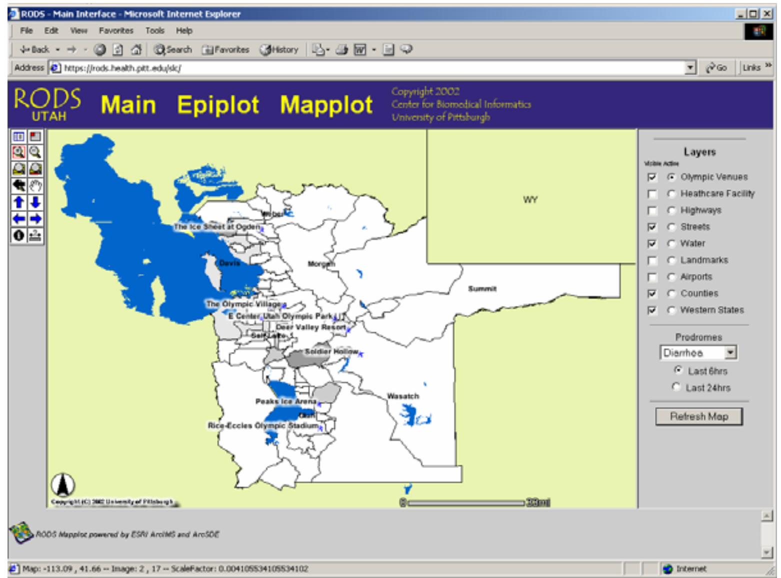
Sajid Siddiqi

Auton/SPR Deployments

1996	1997	1998	1999	2000	2001	2002	2003
M&M Mars Line control Adrenaline (NOX mini-malization)	Kodak (Image sta-bilization) Digital Equipment (pregnan-cy monit-oring)	M&M Mars (manufac-turing) NASA/ NSF (Astrophysics mining) 3M: Textile tension control	Caterpillar (Spare parts) US Army (biotoxin detection) M&M Mars: Scheduling with uncer-tainty 3M (Adhesive design)	DigitalMC (Music tastes) Caterpillar (emissions) SmartMoney (anomalies) Unilever (Brand Management) Phillips Petroleum (work-force optimization) Cellomics (screened anomaly detection)	Biometrics company (health monitor) Boeing (intrusion) Masterfoods (new product development) Cellomics (pro-teomics screen) ABB (Circuit-breaker supply chain) SwissAir (Flight delays) 3M (secret) Washington Public Hospital System (ER delays) Unilever (targeted marketing)	NASA (National Virtual Observatory) NSF (astrostatistics software) DARPA (national disease monitor) Masterfoods (bio-chemistry) Pfizer (High-throughput screen) Caterpillar Inc. (Self-optimizing Engines) Beverage Company (Ingredients/Manufacturing/Marketing/Sales Bayes Net) Transform Pharma (massive autonomous experiment design) Census Bureau (privacy protection) Psychogenics Inc: Effects of psychotropic drugs on rats	NSF (astrostatistics software) Masterfoods (bio-chemistry) State of PA (National Disease Monitor [with Mike Wagner of U. Pitt]) State of PA (Anti Cancer [collaboration with CMU Biology] DARPA (detecting patterns in links) Other Government Departments (identifying dangerous people, potential collaborators, and aliases) Other Government Departments (detecting a class of clusters) Other Pharma Research Co. Life Science specific data mining United States Department of Agriculture: Early warning system for food terrorism NSF: Biosurveillance Algorithms

Our 5 biggest applications in 2004

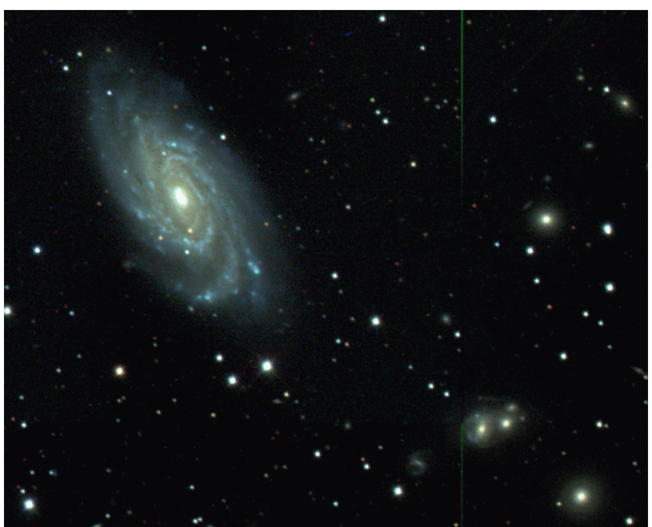
Biomedical Security (with Mike Wagner, University of Pittsburgh)



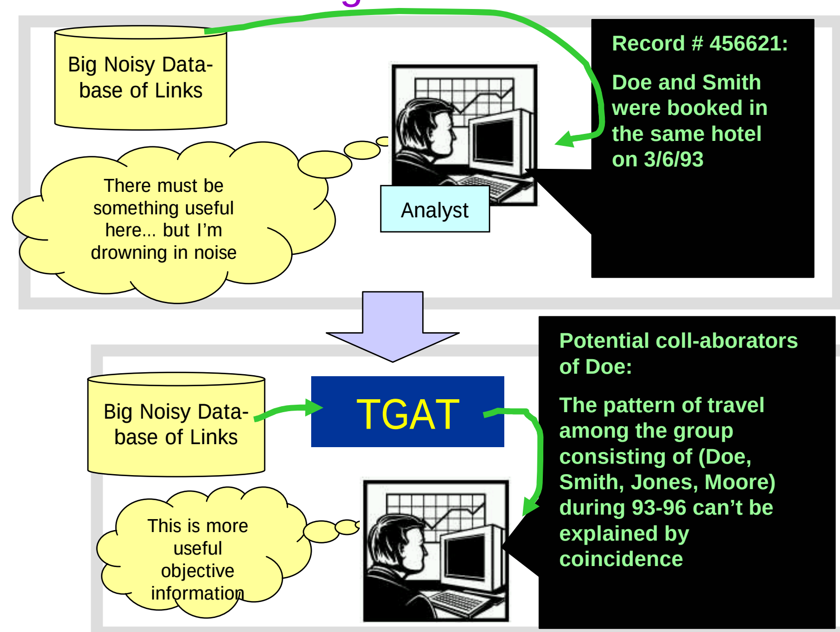
Autonomous self-tweaking engines

Drug Screening

Big Astrophysics Automated Science



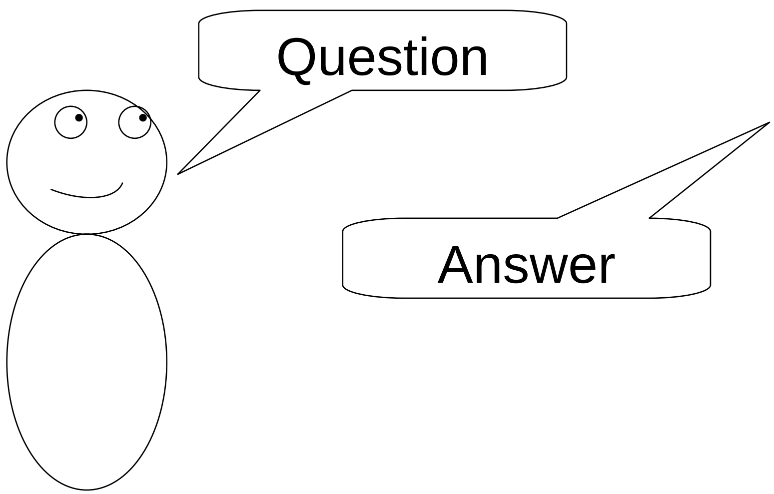
Intelligence Data



Outline

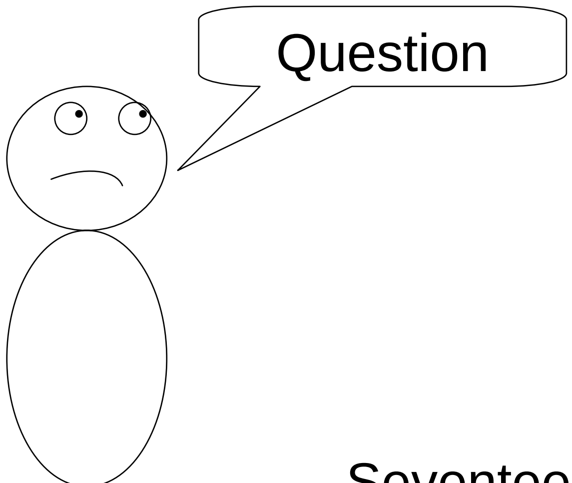
- Cached Sufficient Statistics
- Kd-trees and Ball Trees
- K-nearest neighbor with ball trees
  - Very fast non-parametric classification
  - skewed binary outputs
  - General binary outputs
  - multi-classed outputs
- Very fast kernel-based statistics
  - n-point computations
  - clustering
  - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

Data Analysis: The old days

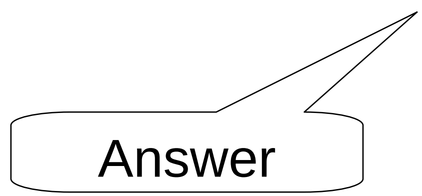


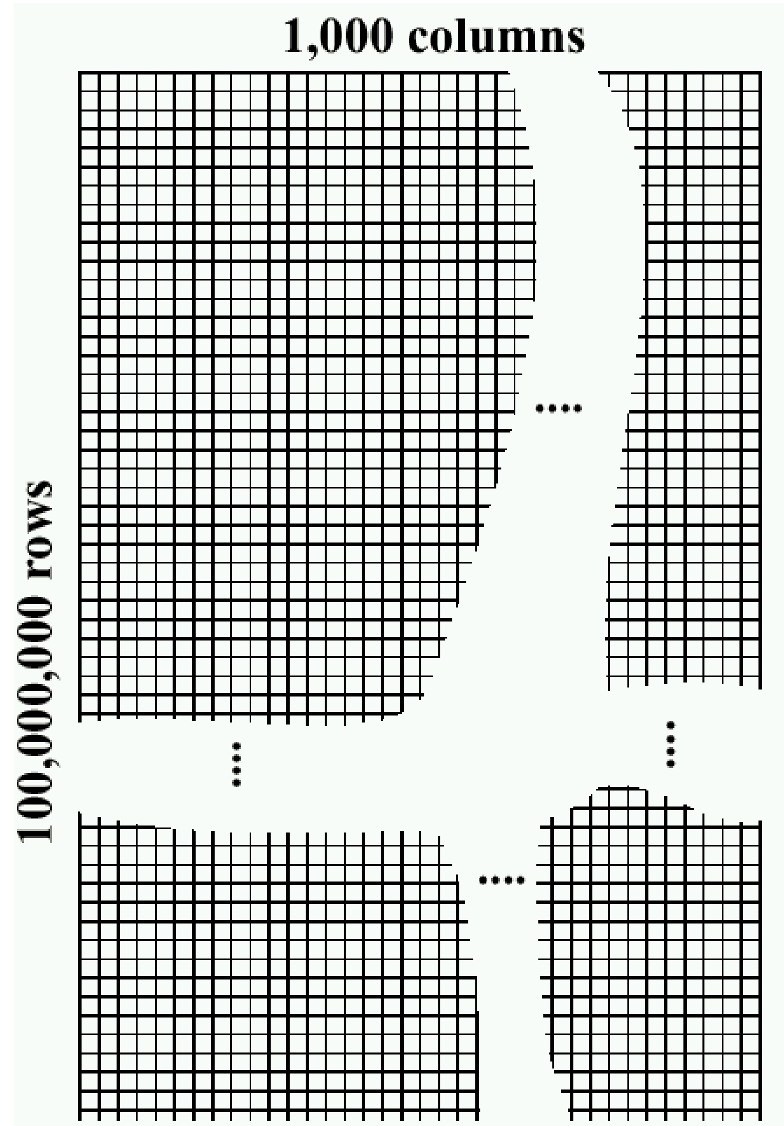
Size	Ellipticity	Color
23	0.96	Red
33	0.55	Red
36		Green
40		
20		
48		

Data Analysis: The new days

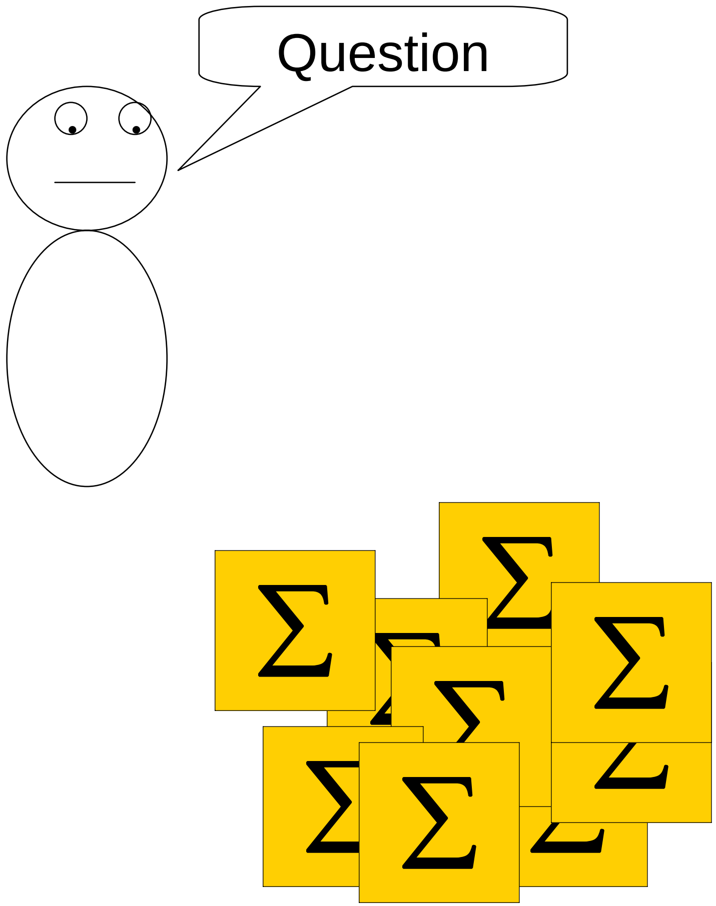


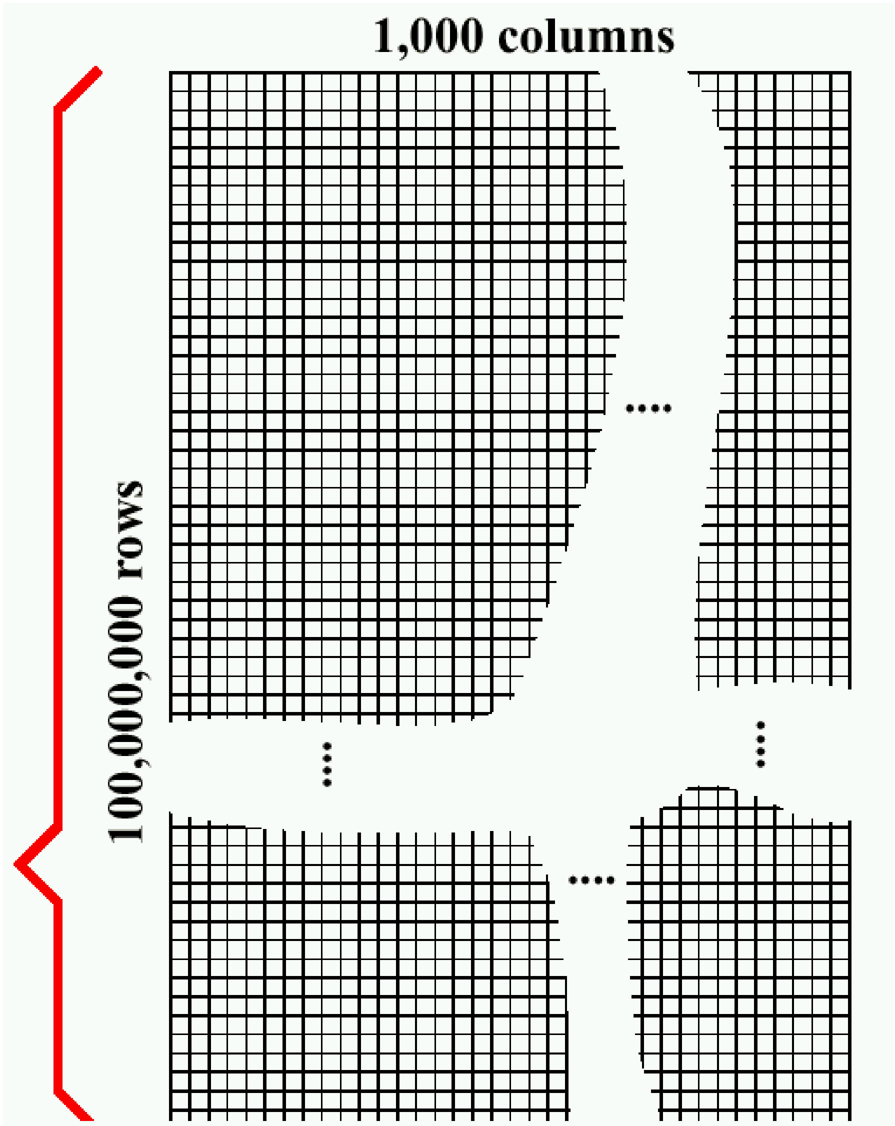
Seventeen
months later…



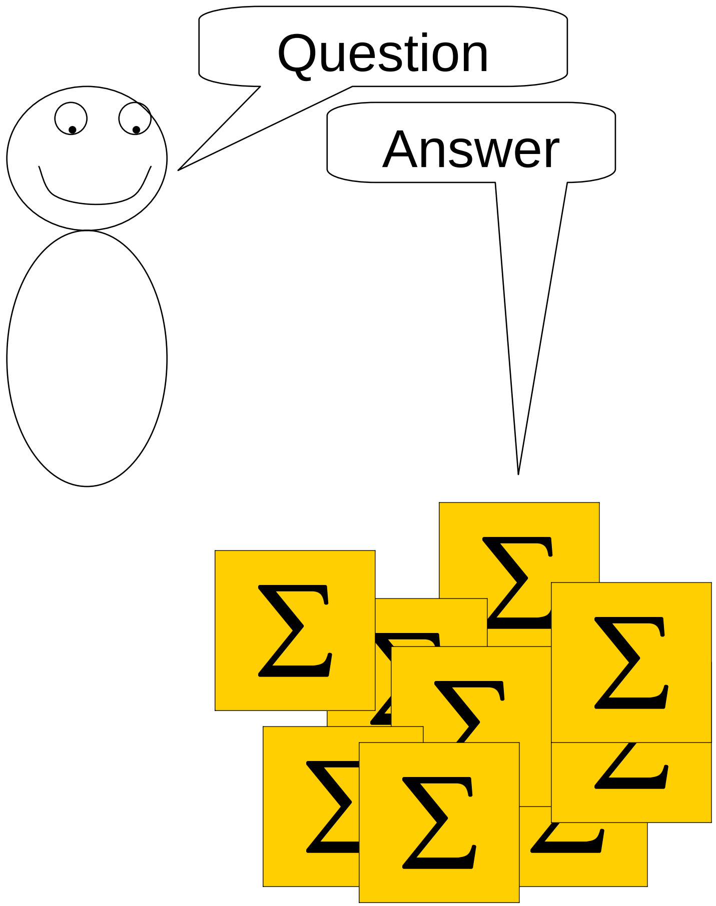


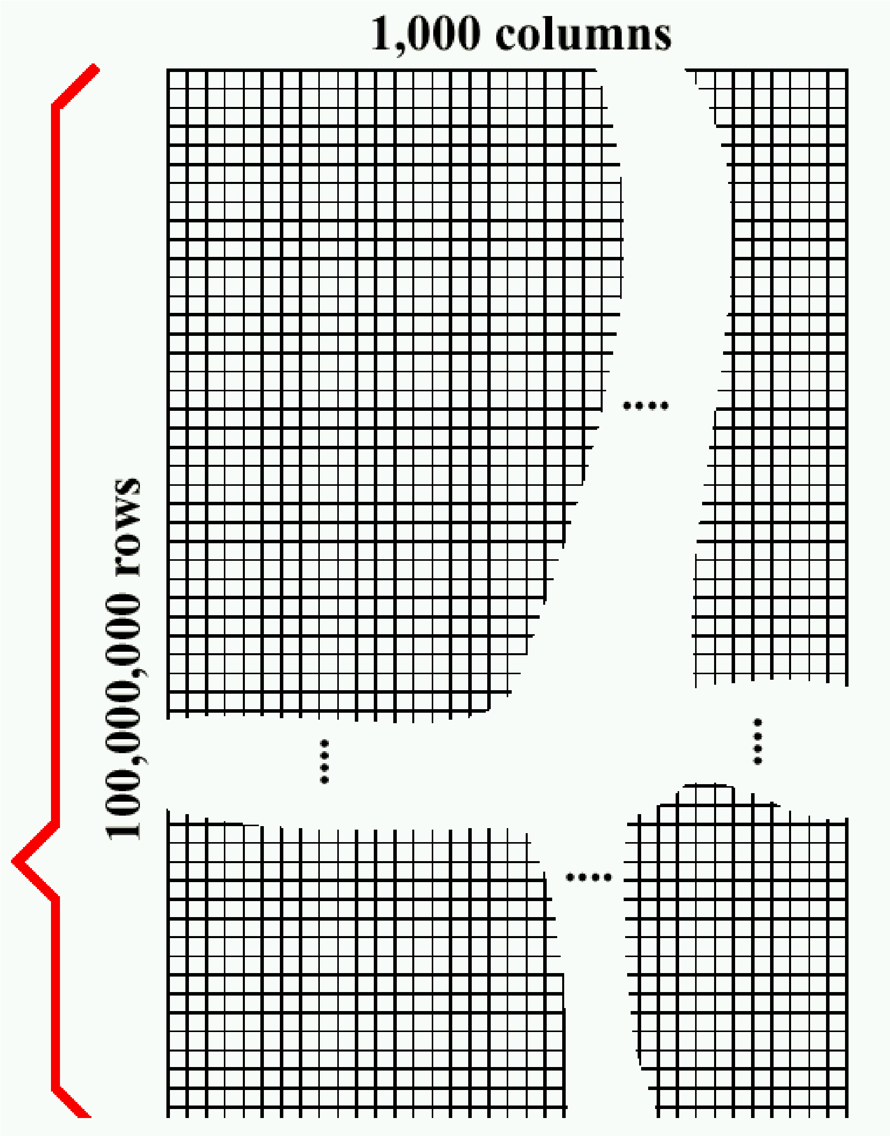
Cached Sufficient Statistics





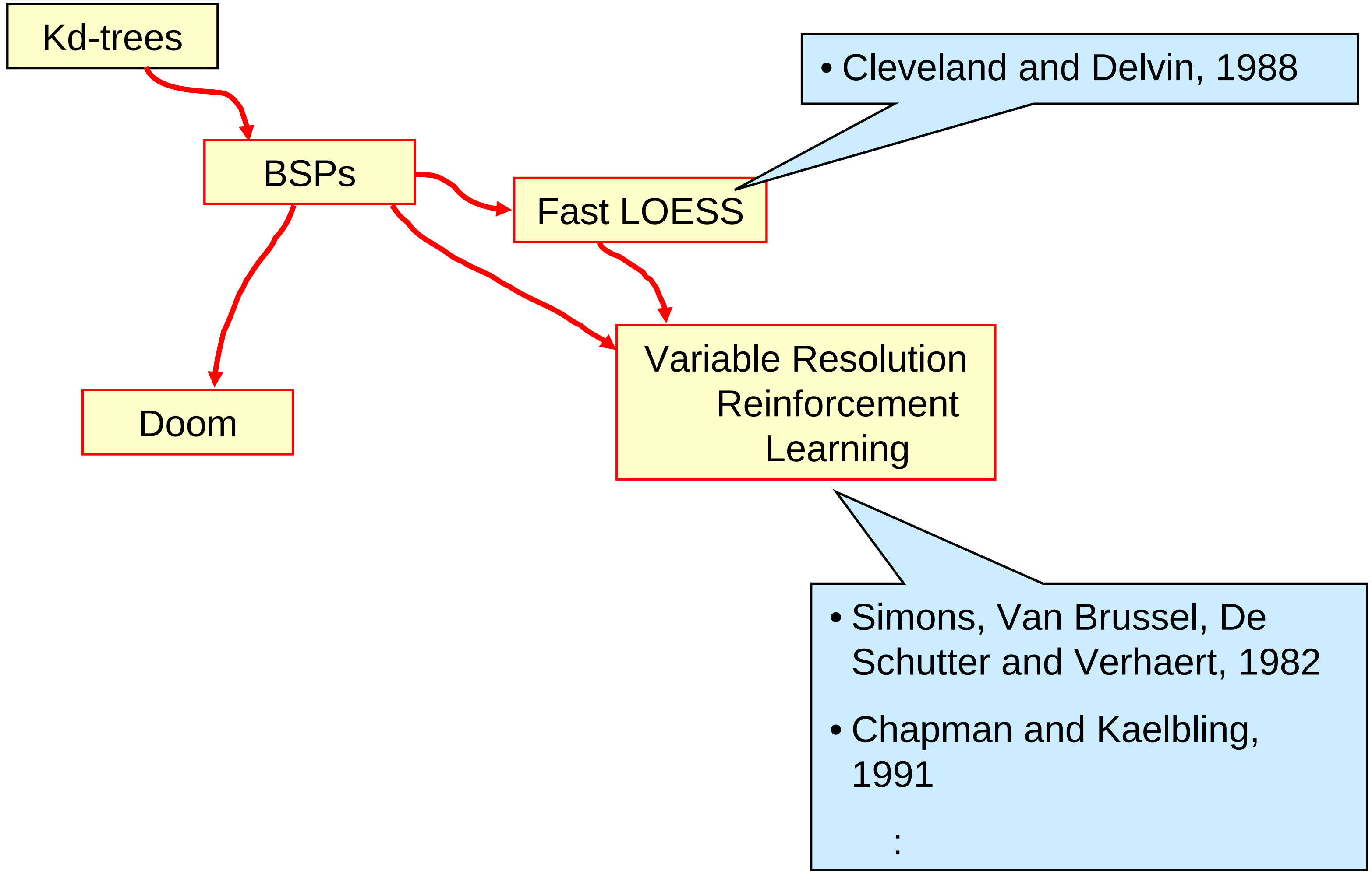
Cached Sufficient Statistics

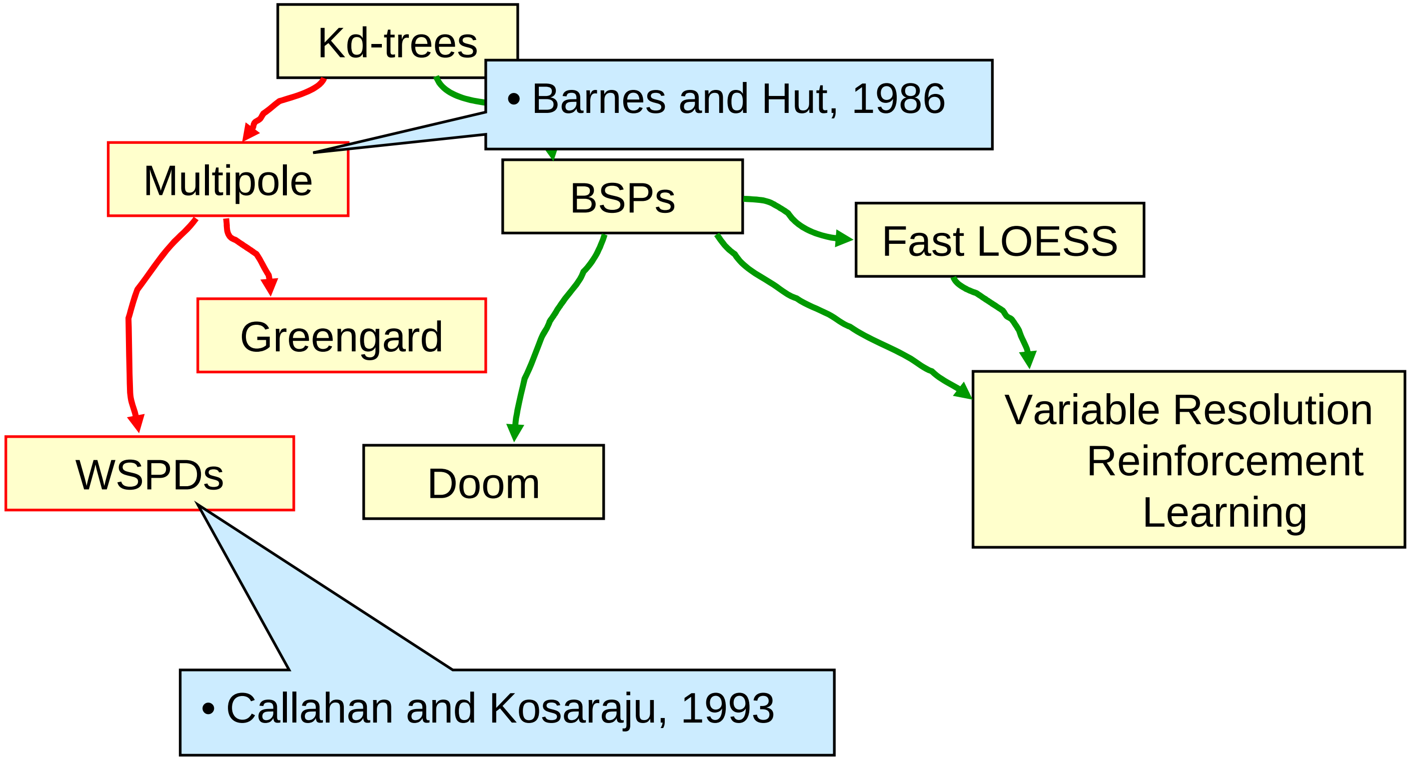


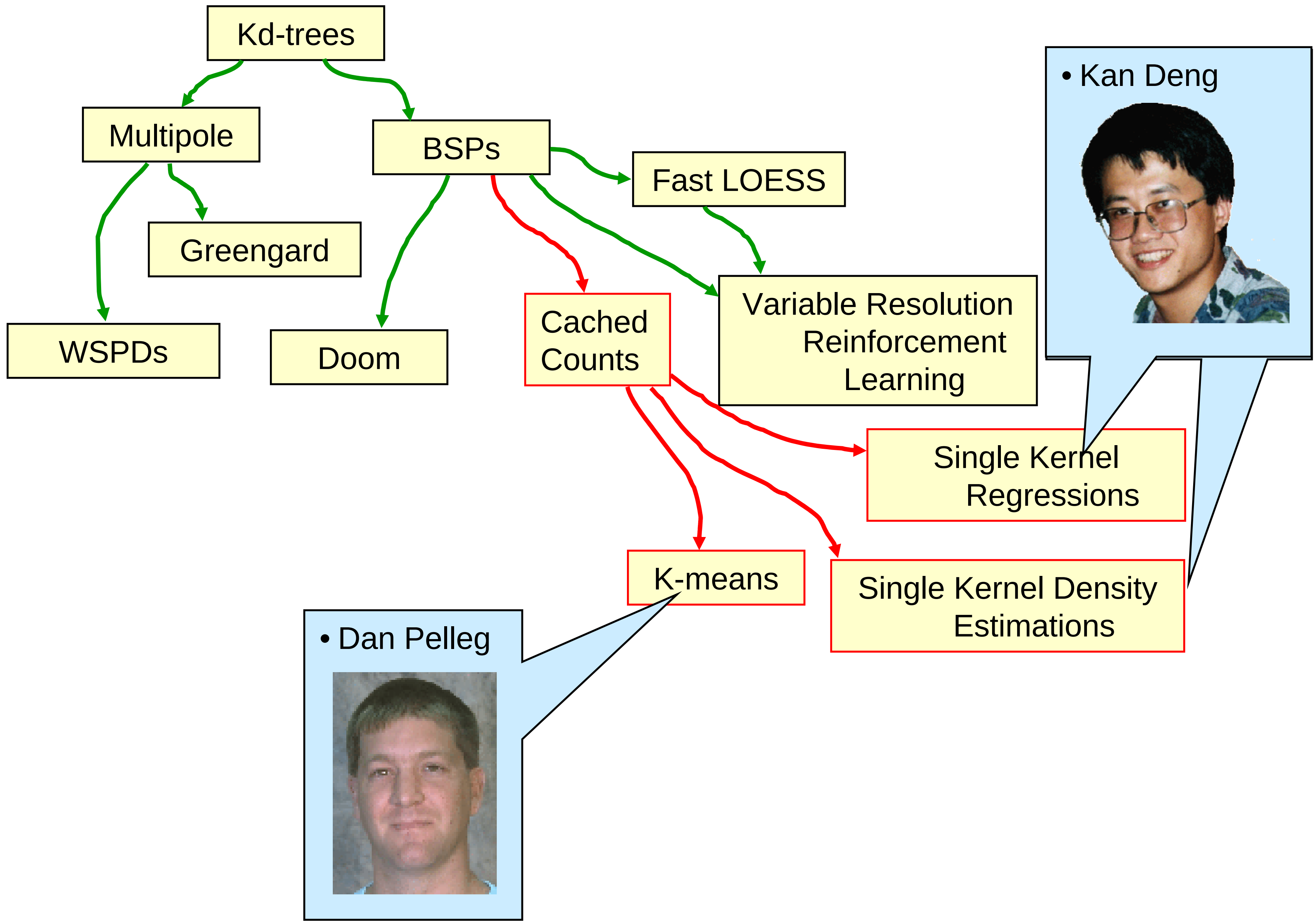


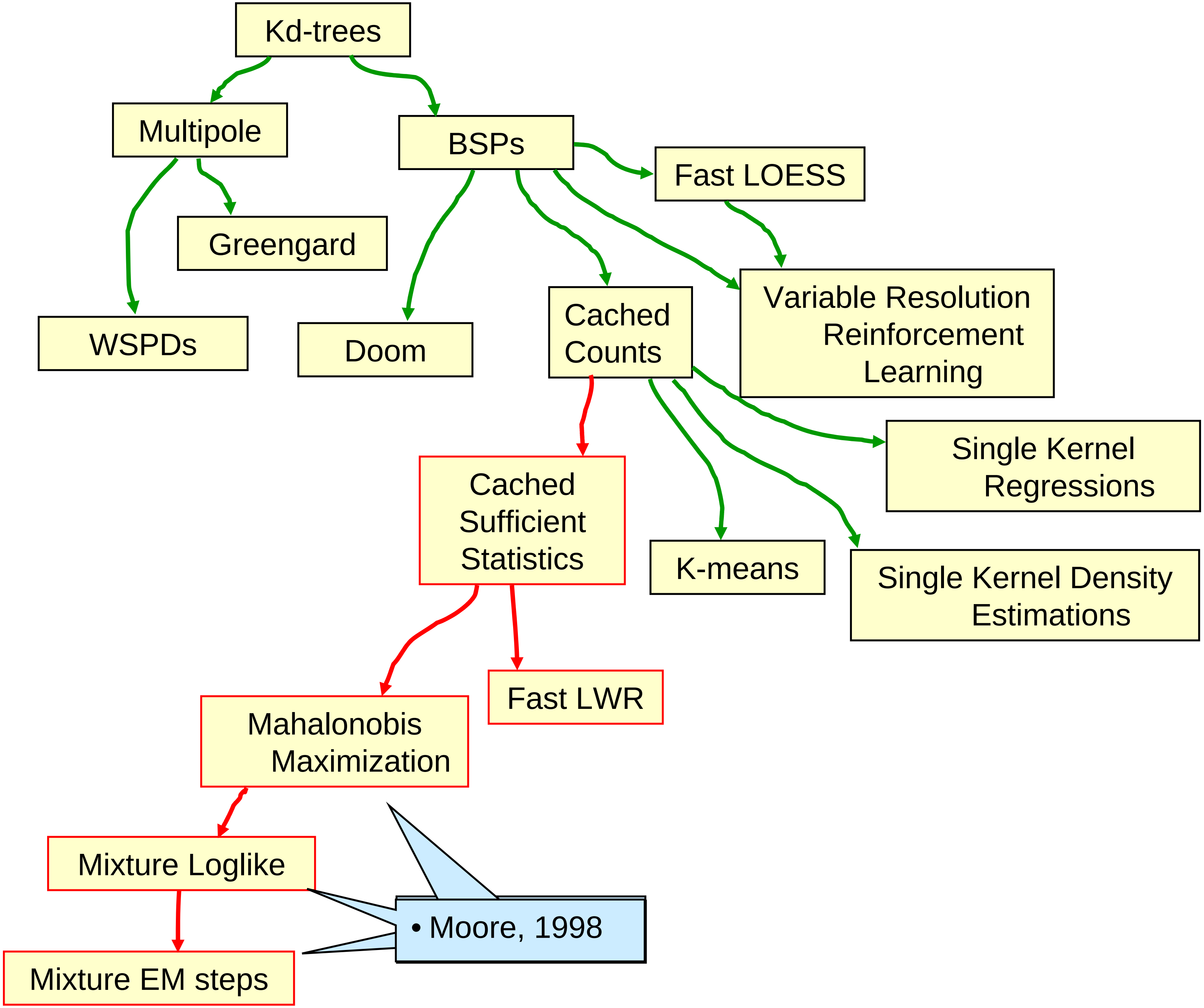
Kd-trees

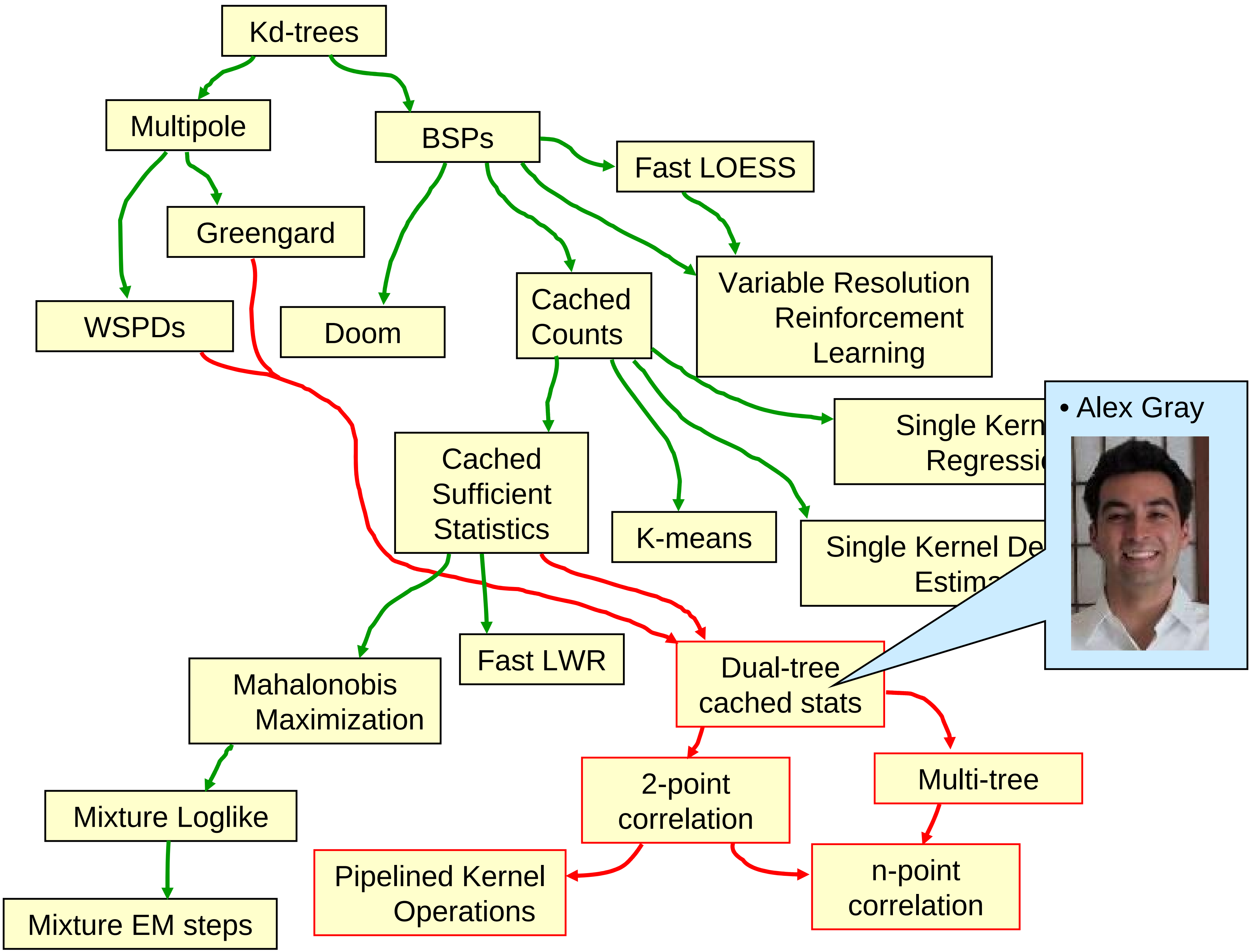
- Friedman, Bentley and Finkel, 1977

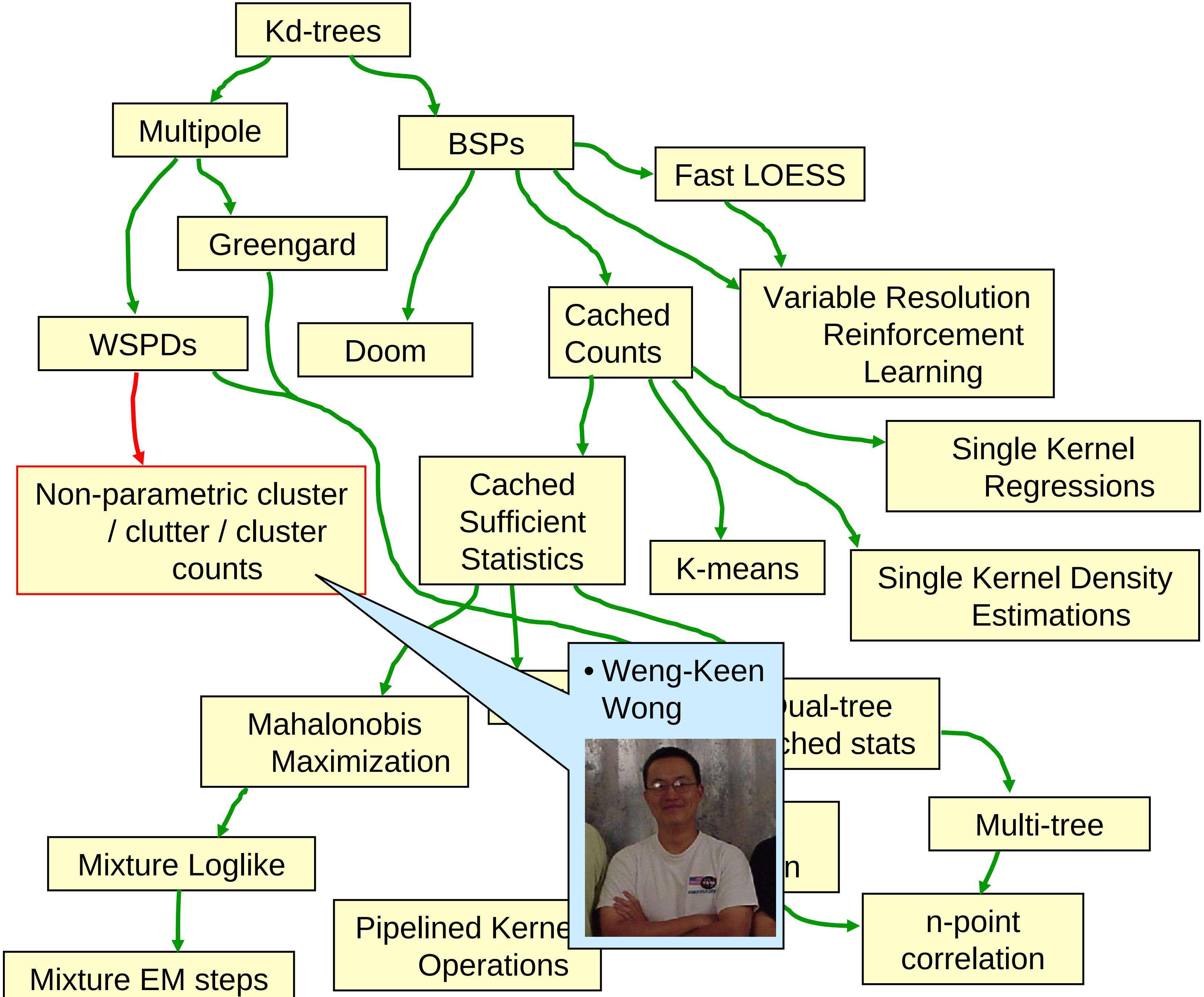

Kd-trees
BSPs
Fast LOESS
Doom
Variable Resolution Reinforcement Learning
• Cleveland and Delvin, 1988
• Simons, Van Brussel, De Schutter and Verhaert, 1982
• Chapman and Kaelbling, 1991
:

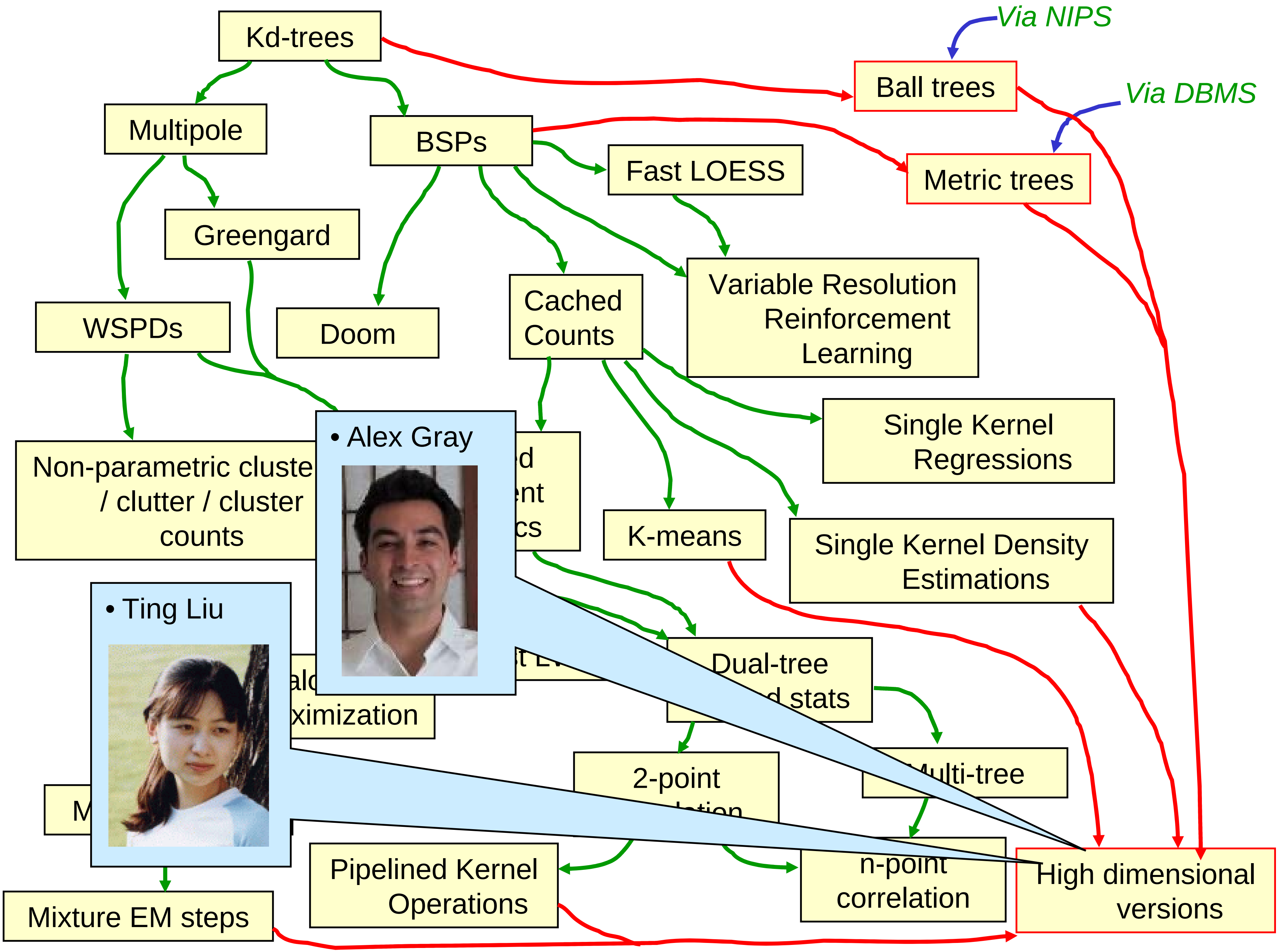

Kd-trees
• Barnes and Hut, 1986
Multipole
BSPs
Fast LOESS
Greengard
Variable Resolution Reinforcement Learning
WSPDs
Doom
• Callahan and Kosaraju, 1993

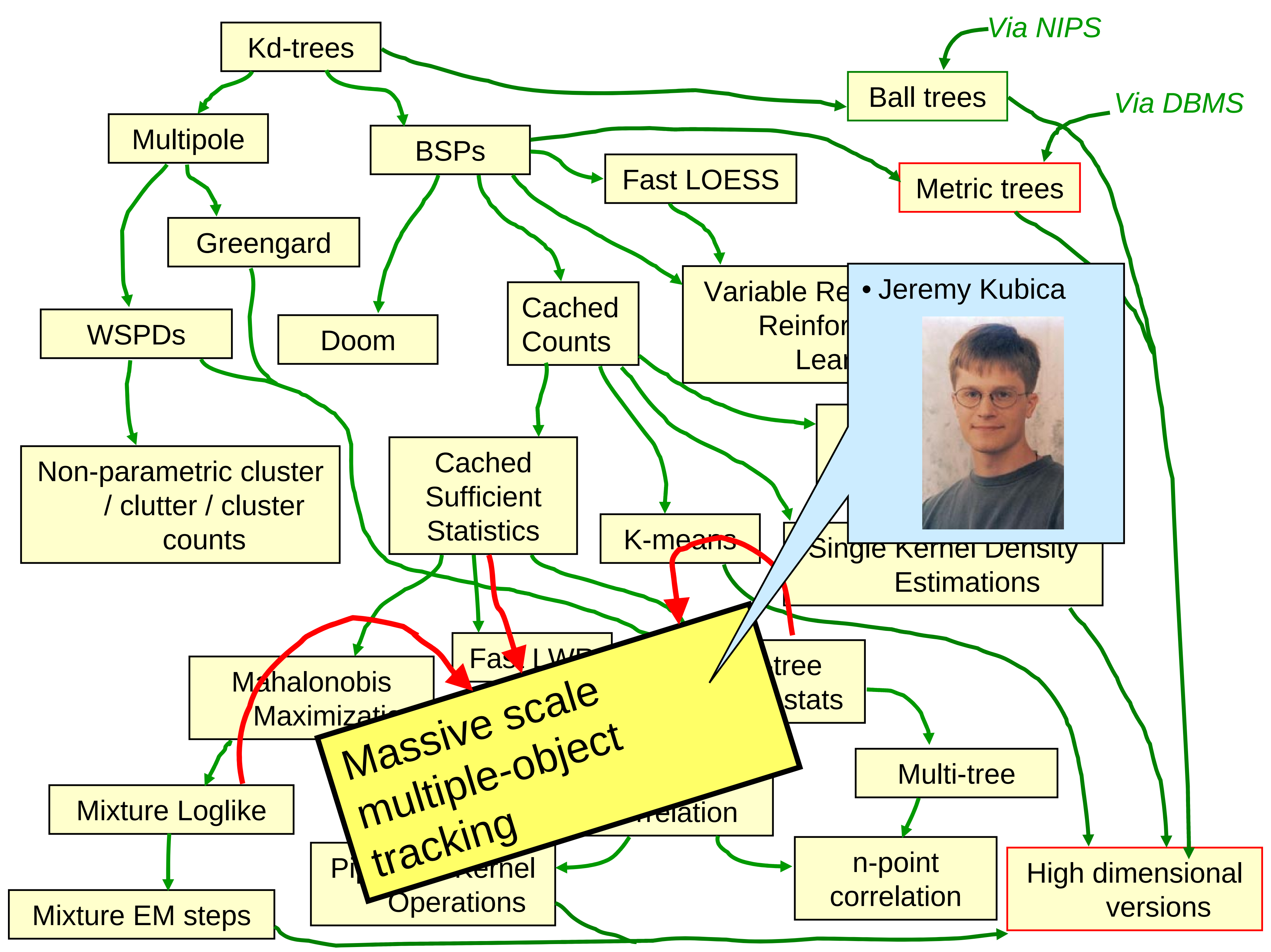

Kd-trees
Multipole
BSPs
Fast LOESS
Greengard
WSPDs
Doom
Cached Counts
Variable Resolution Reinforcement Learning
Single Kernel Regressions
K-means
Single Kernel Density Estimations
• Kan Deng
• Dan Pelleg


Kd-trees
Multipole
BSPs
Fast LOESS
Greengard
WSPDs
Doom
Cached Counts
Variable Resolution Reinforcement Learning
Single Kernel Regressions
Cached Sufficient Statistics
K-means
Single Kernel Density Estimations
Mahalonobis Maximization
Fast LWR
Mixture Loglike
Mixture EM steps
• Moore, 1998


Kd-trees
Multipole
BSPs
Fast LOESS
Greengard
Variable Resolution Reinforcement Learning
WSPDs
Doom
Cached Counts
Single Kern
Regressi
Cached Sufficient Statistics
K-means
Single Kernel De
Estima
• Alex Gray
Fast LWR
Dual-tree cached stats
Mahalonobis Maximization
2-point correlation
Multi-tree
Mixture Loglike
Pipelined Kernel Operations
n-point correlation
Mixture EM steps


Kd-trees
Multipole
BSPs
Fast LOESS
Greengard
Variable Resolution Reinforcement Learning
Cached Counts
WSPDs
Doom
Single Kernel Regressions
Non-parametric cluster / clutter / cluster counts
Cached Sufficient Statistics
K-means
Single Kernel Density Estimations
• Weng-Keen Wong
ual-tree
ched stats
Mahalonobis Maximization
Multi-tree
Mixture Loglike
n
n-point correlation
Pipelined Kerne
Operations
Mixture EM steps

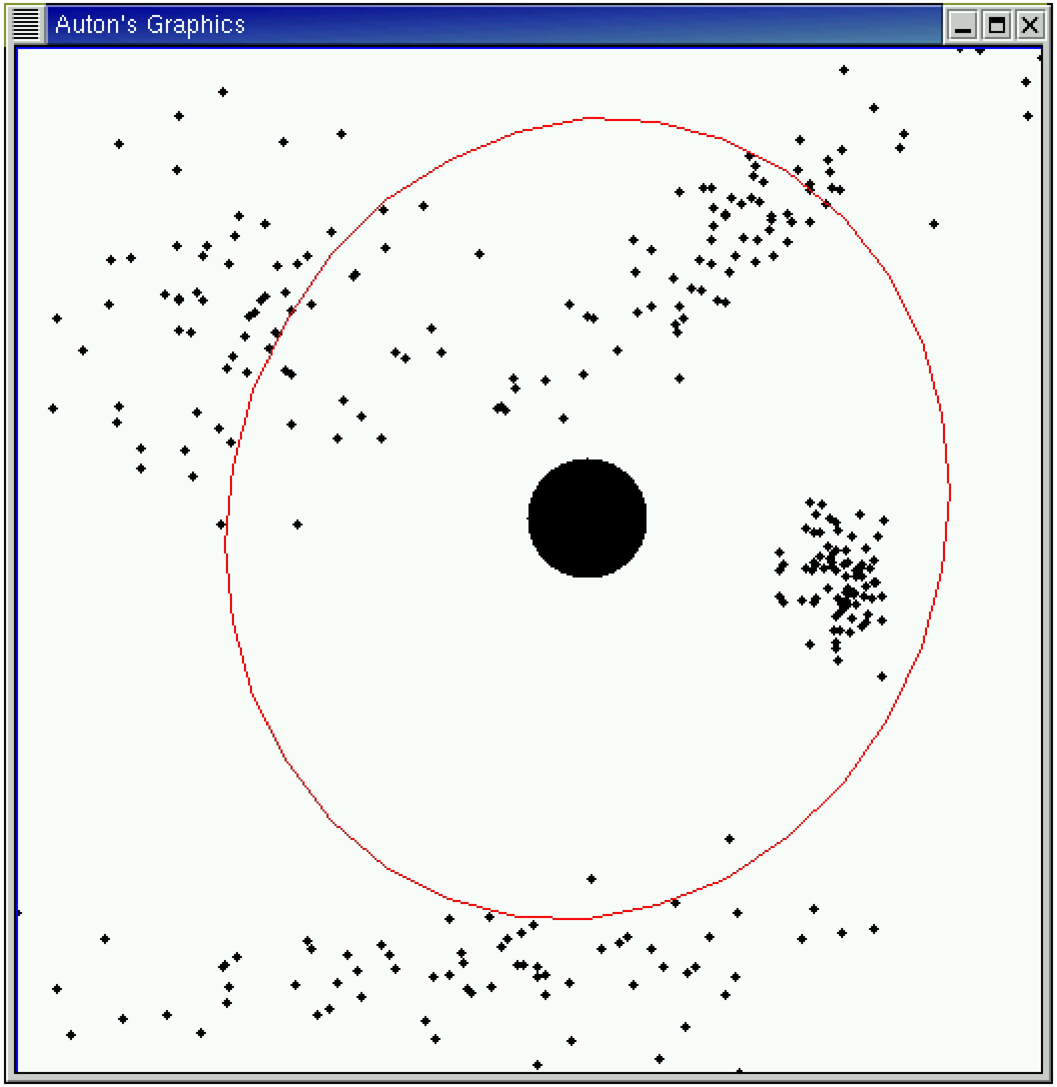

Via NIPS
Kd-trees
Ball trees
Via DBMS
Multipole
BSPs
Fast LOESS
Metric trees
Greengard
Variable Resolution Reinforcement Learning
WSPDs
Doom
Cached Counts
Single Kernel Regressions
Non-parametric cluste
/ clutter / cluster
counts
• Alex Gray
K-means
Single Kernel Density Estimations
• Ting Liu
Dual-tree
d stats
ximization
2-point
ulti-tree
n-point correlation
Pipelined Kernel Operations
High dimensional versions
Mixture EM steps


Via NIPS
Via DBMS
Kd-trees
Ball trees
Multipole
BSPs
Fast LOESS
Metric trees
Greengard
Variable
WSPDs
Doom
Cached Counts
• Jeremy Kubica
Non-parametric cluster / clutter / cluster counts
Cached Sufficient Statistics
K-means
Estimations
Mahalonobis
Multi-tree
Mixture Loglike
Mixture EM steps
Operations
n-point correlation
High dimensional versions
Massive scale multiple-object tracking

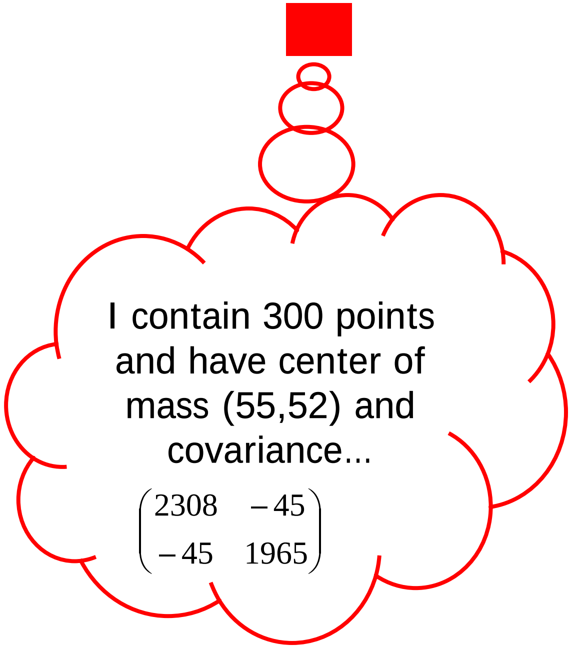
Outline

- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
    - Very fast non-parametric classification
    - skewed binary outputs
    - General binary outputs
    - multi-classed outputs
- Very fast kernel-based statistics
    - n-point computations
    - clustering
    - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

Structure of kdtrees



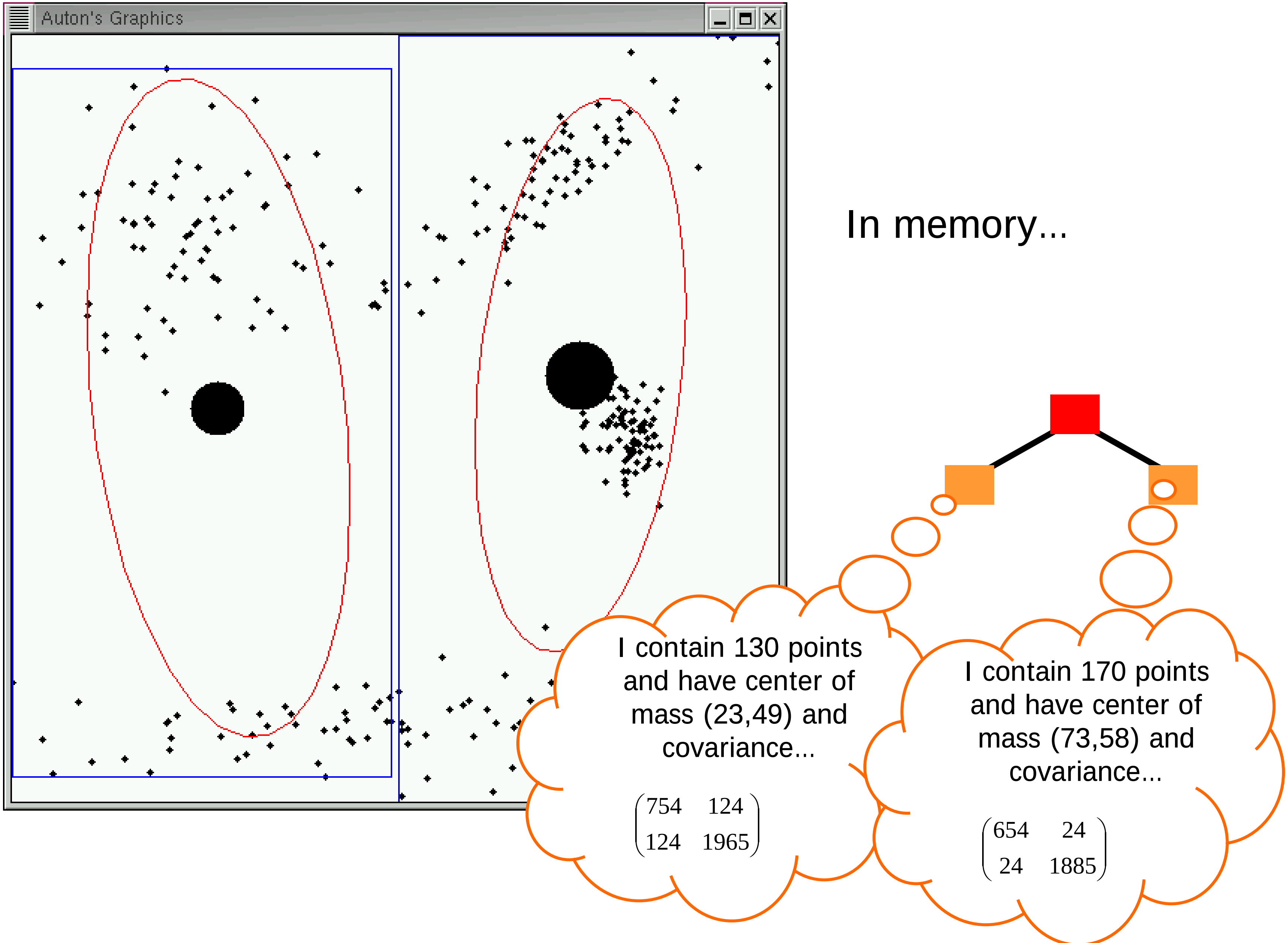
In memory...



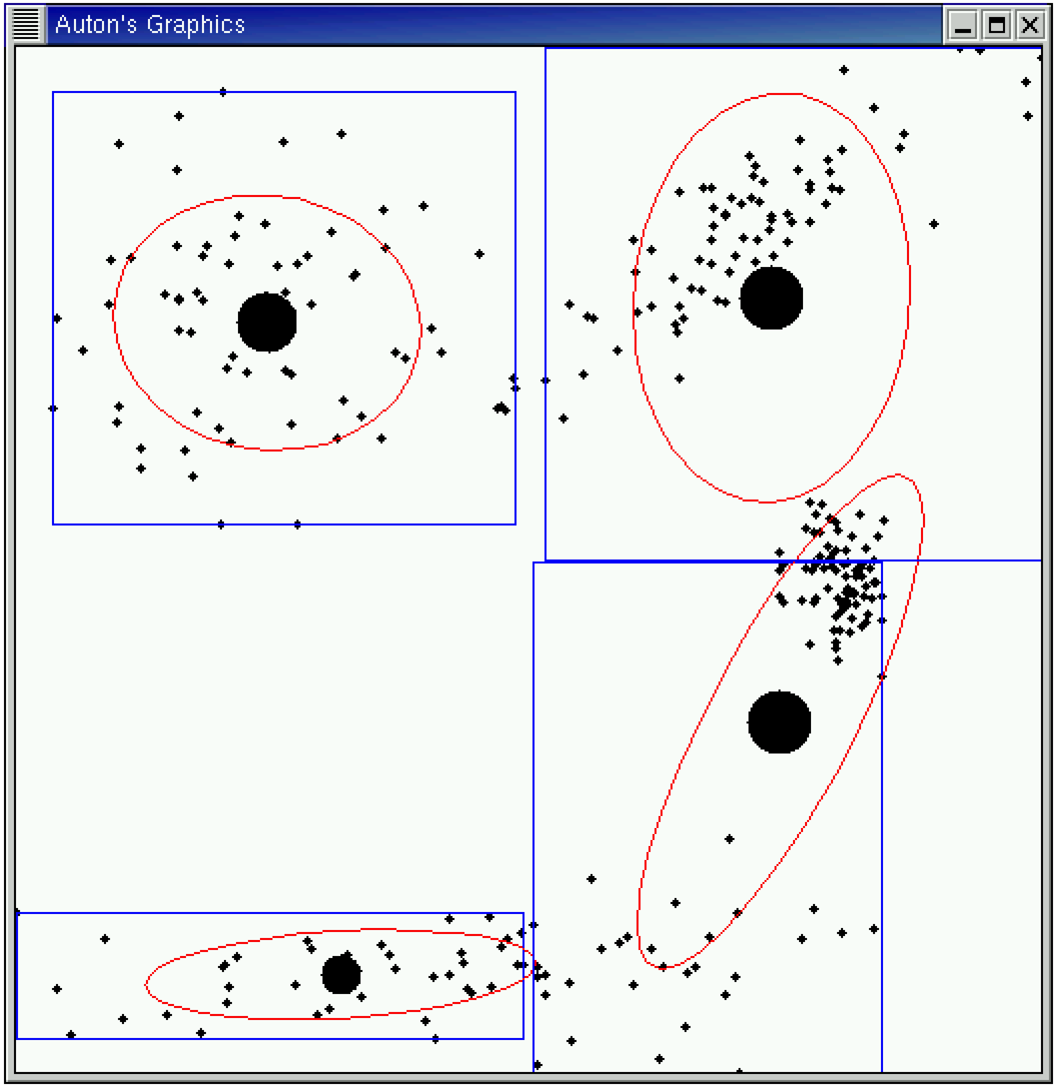
$$\begin{pmatrix} 2308 & -45 \\ -45 & 1965 \end{pmatrix}$$

Structure of of kdtrees

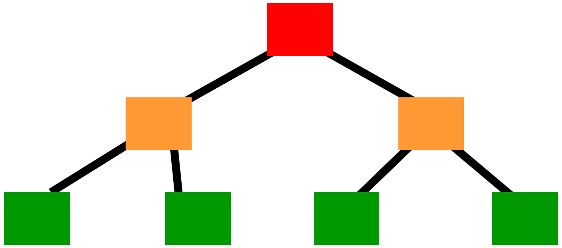
In memory...



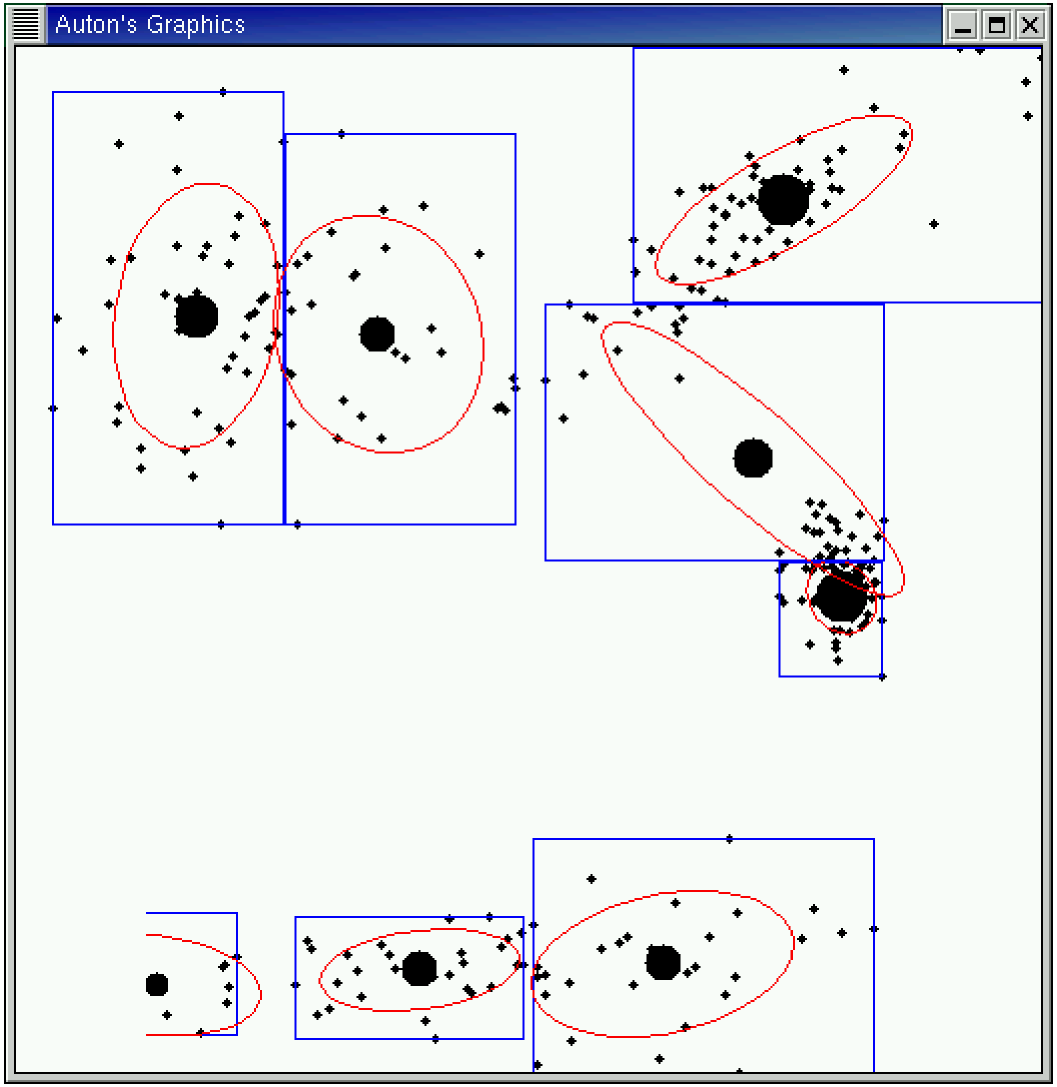
Structure of of kdtrees



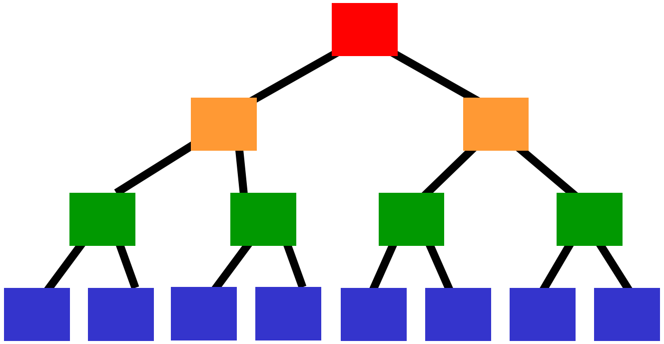
In memory...



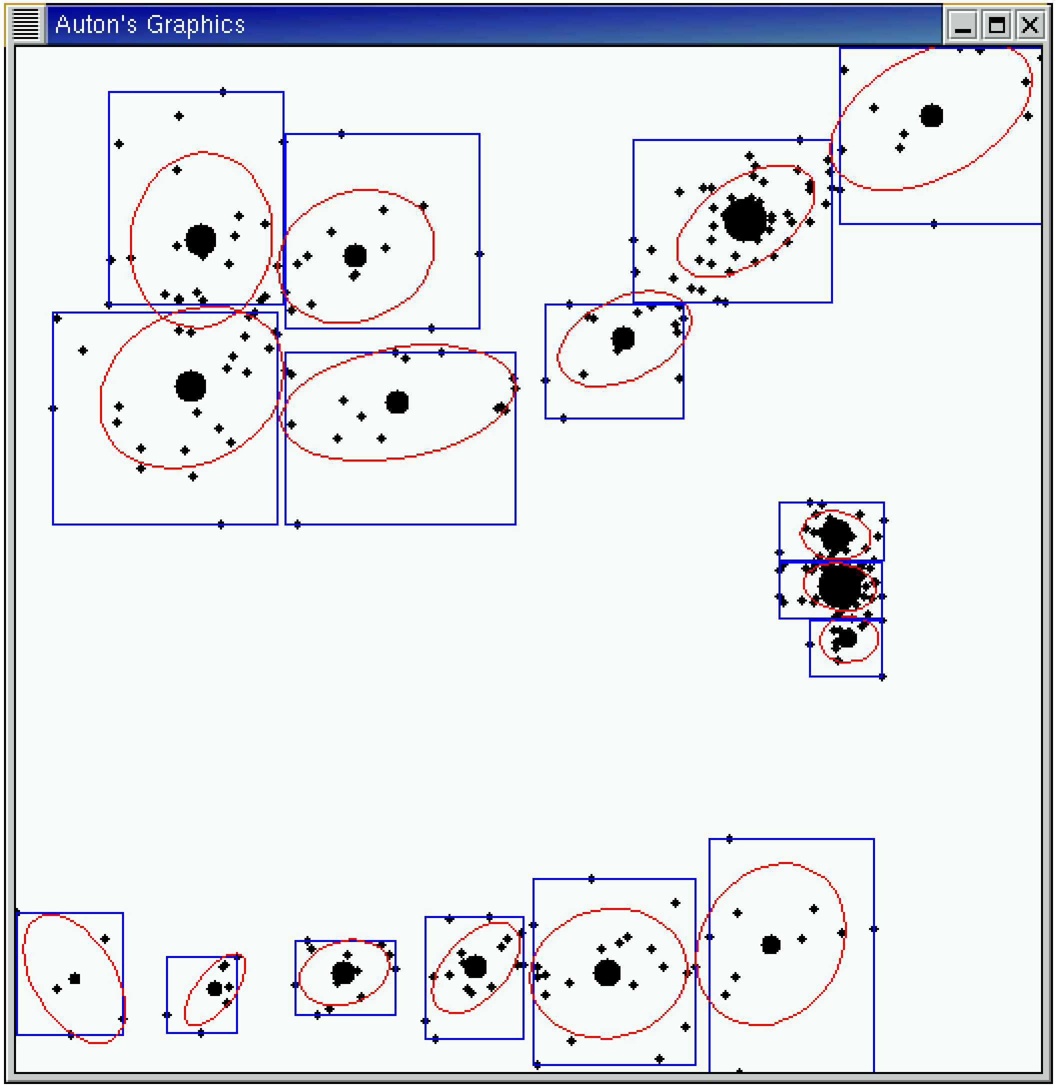
Structure of of kdtrees



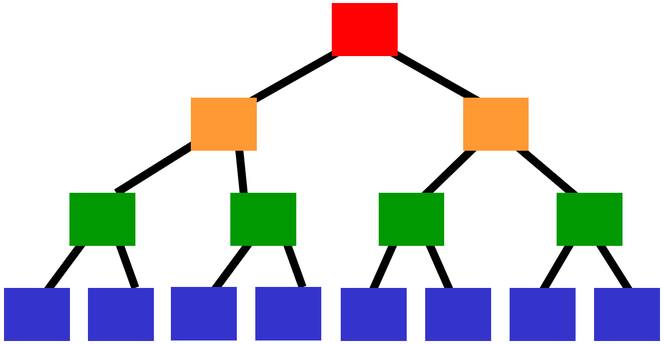
In memory...



Structure of of kdtrees

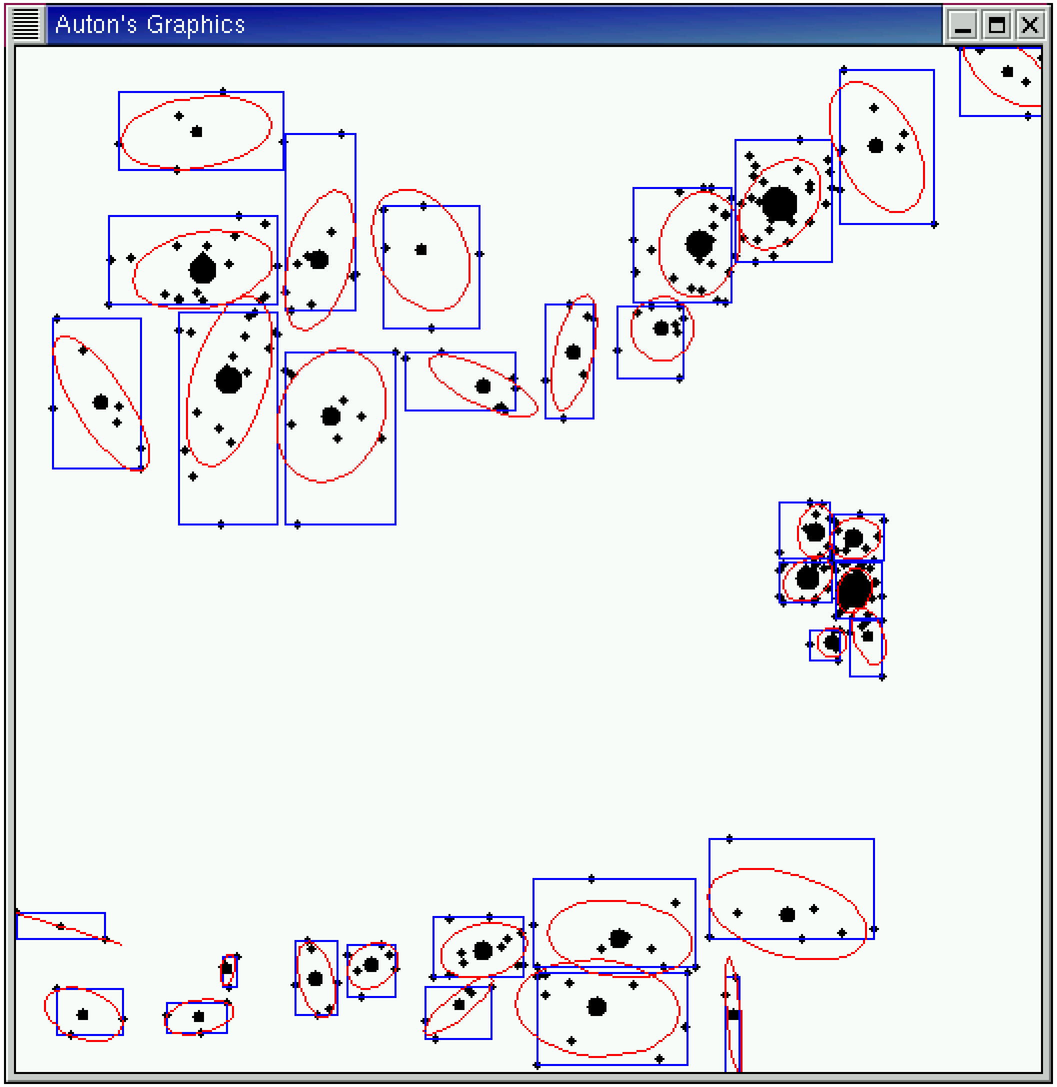


In memory...

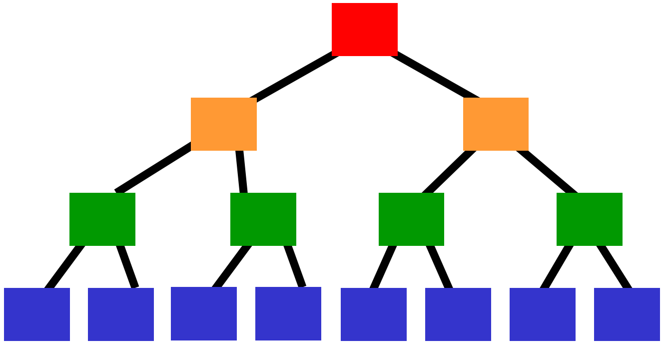


<more levels here>

Structure of of kdtrees

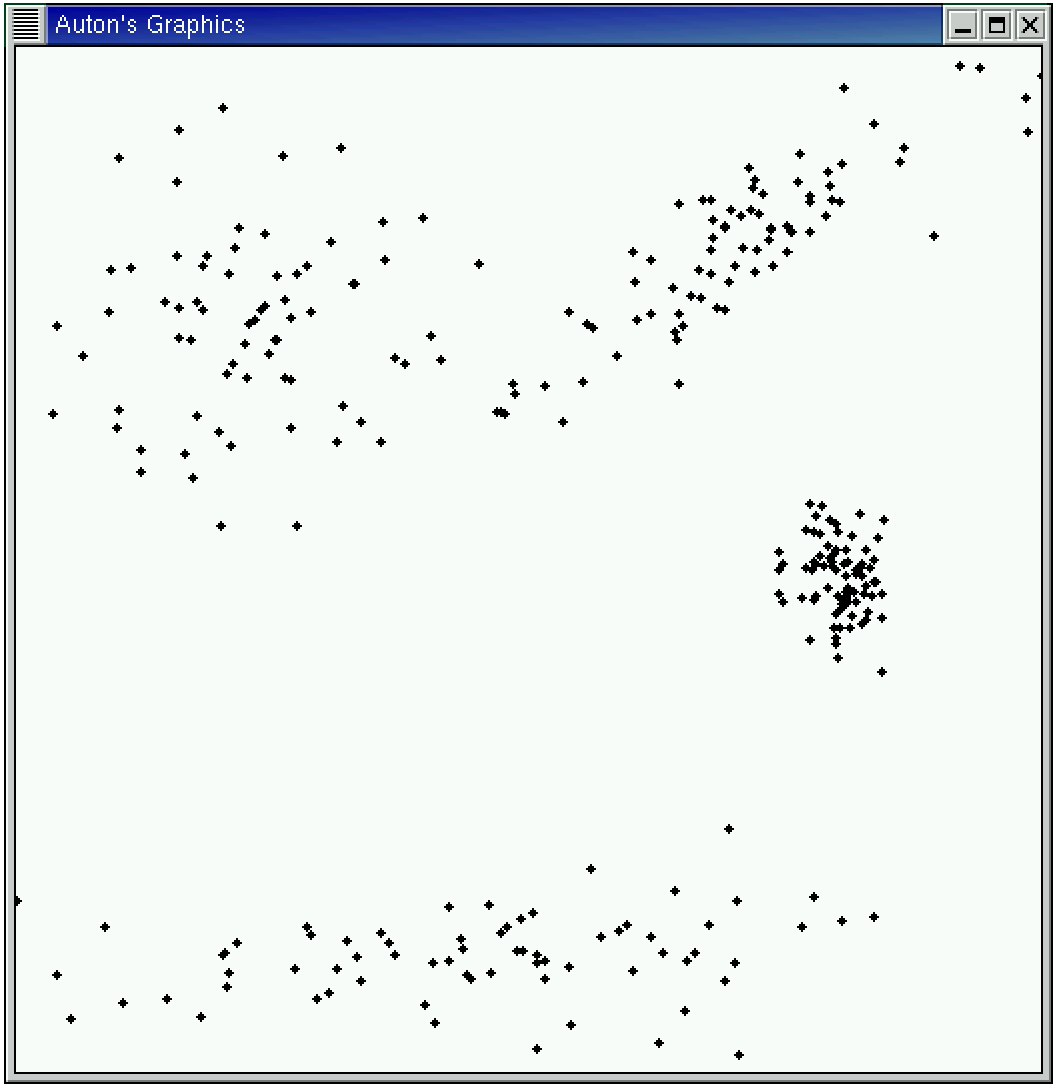


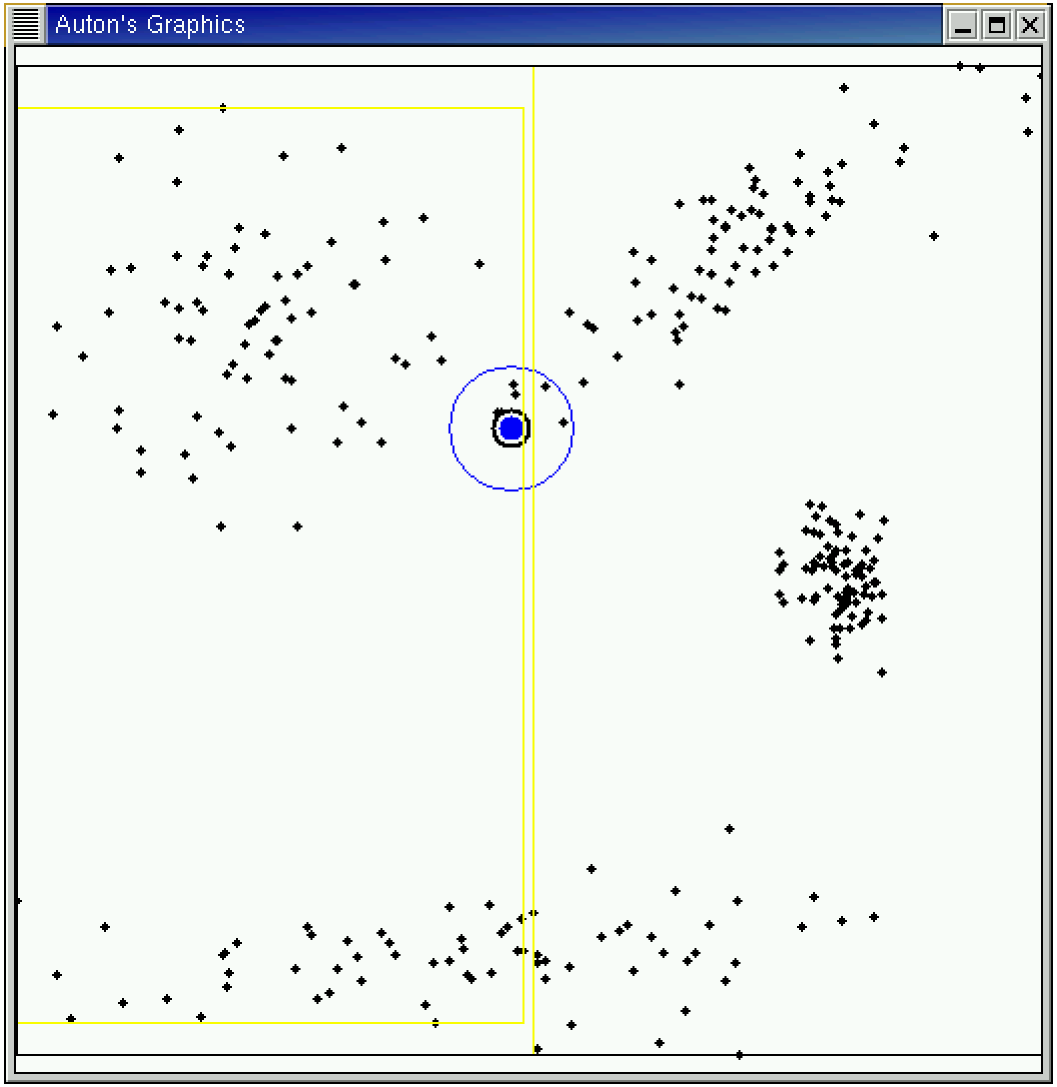
In memory...

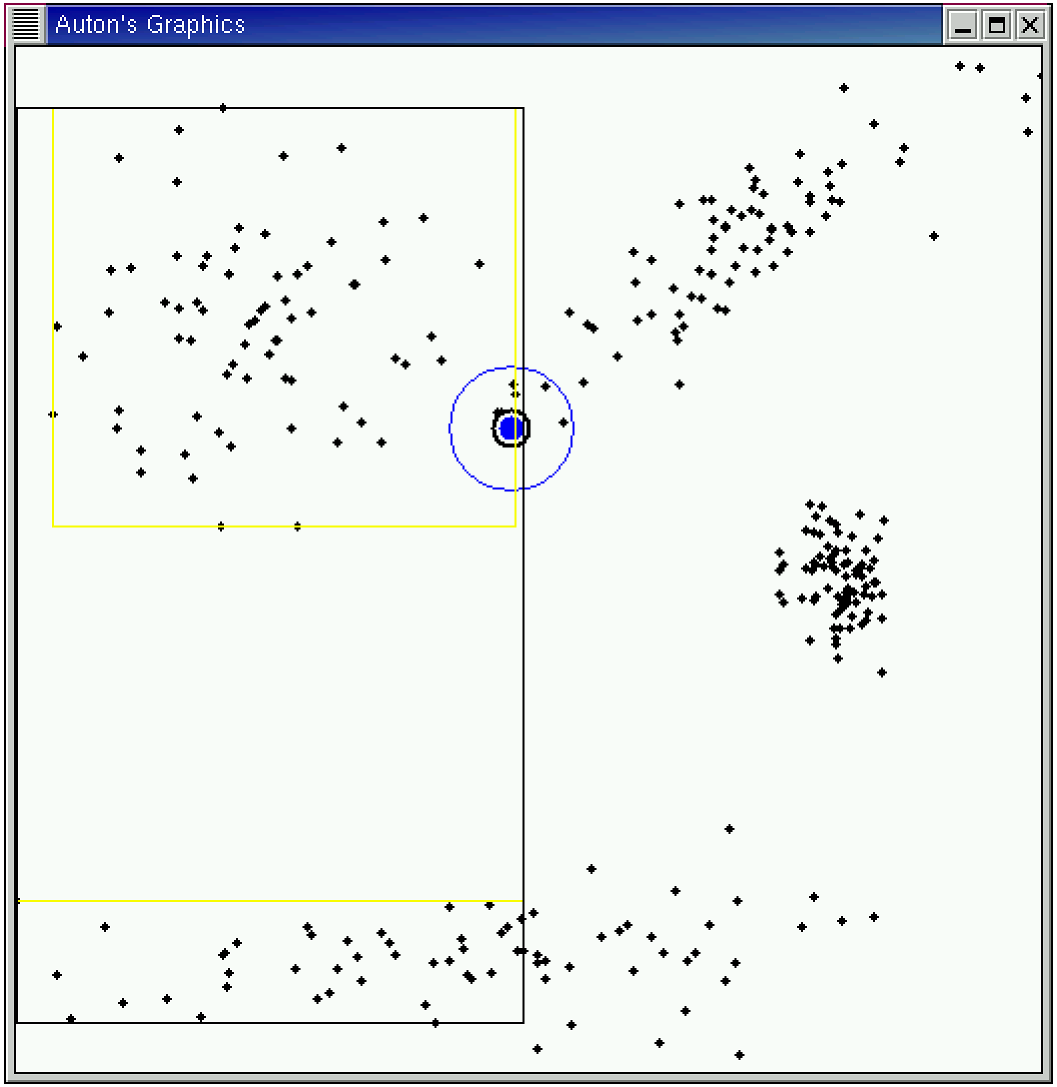


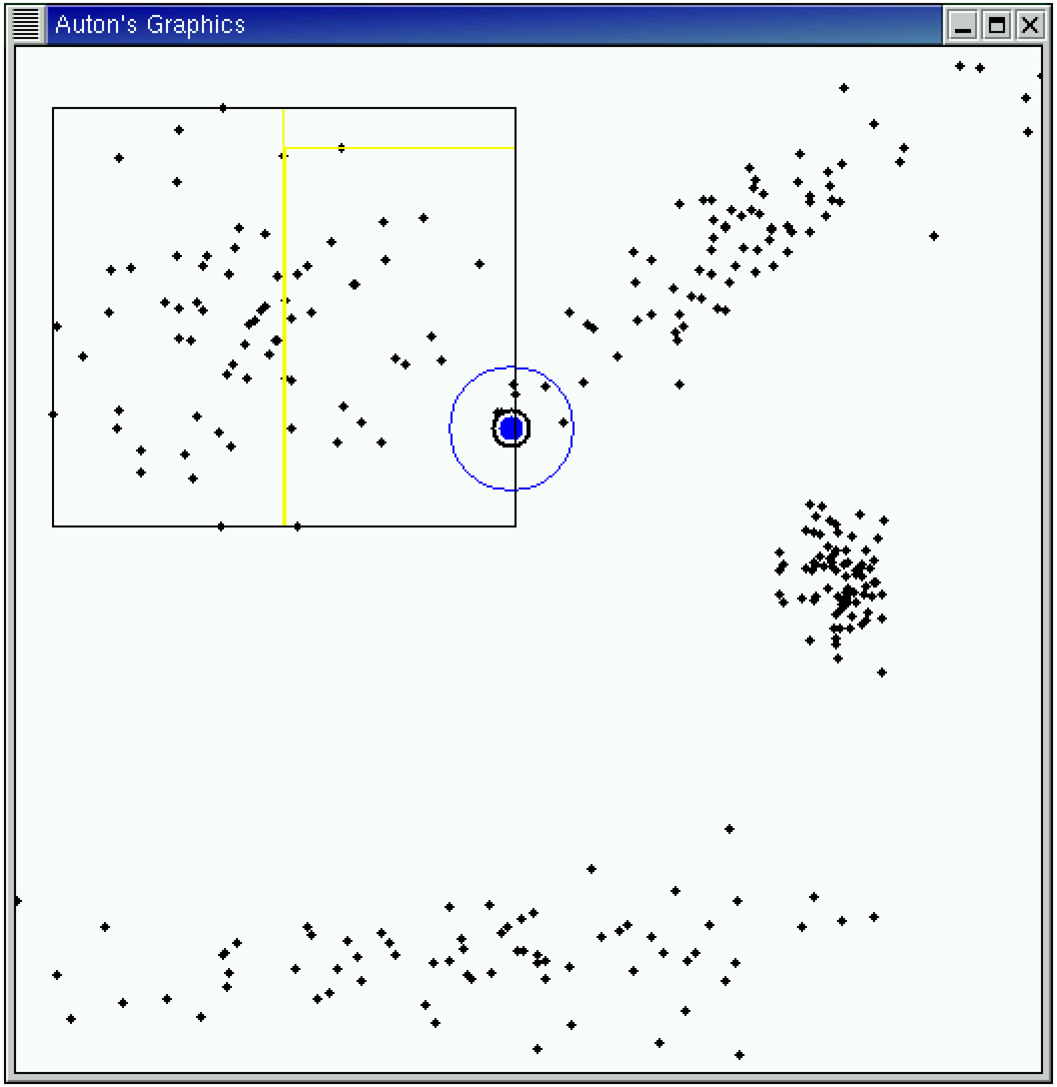
<more levels here>

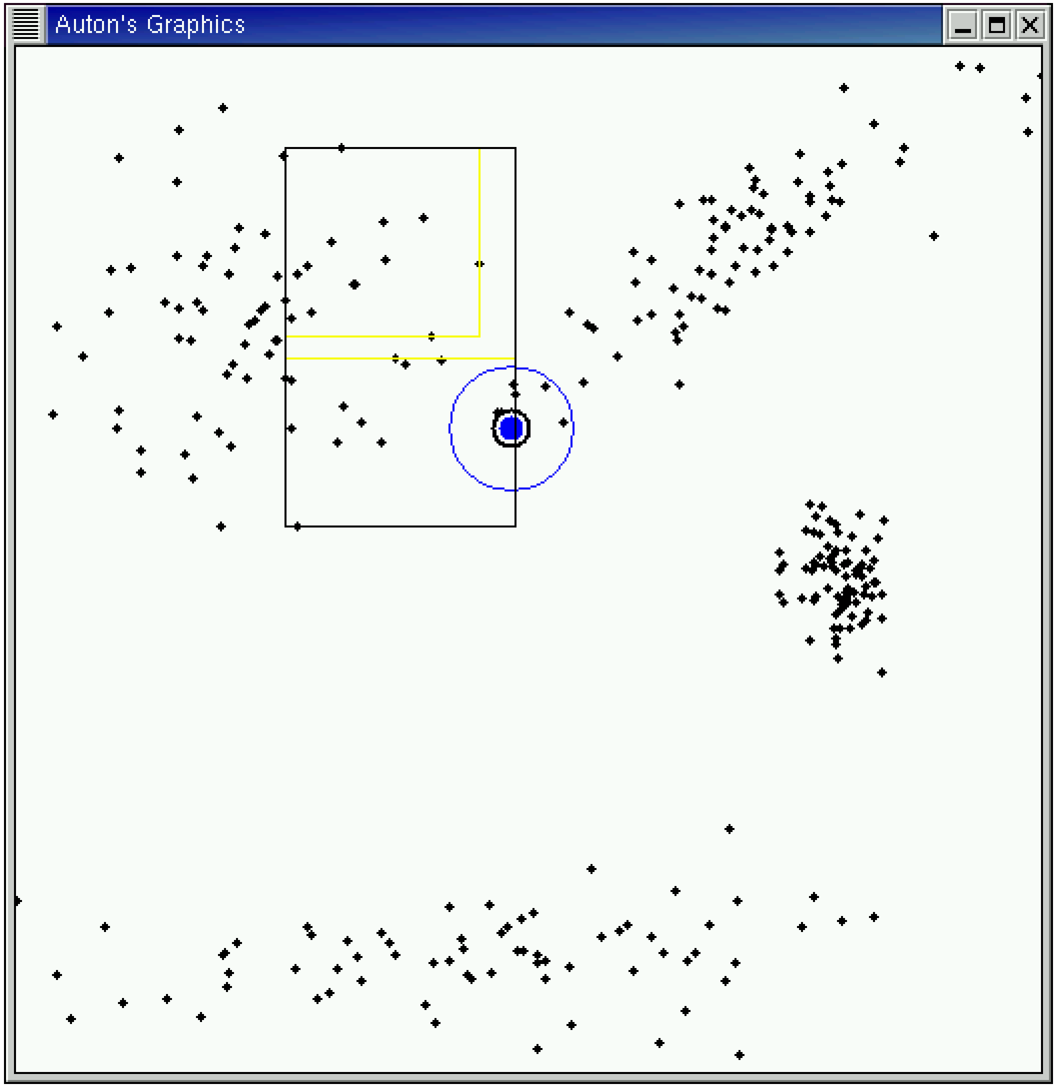
Range Search

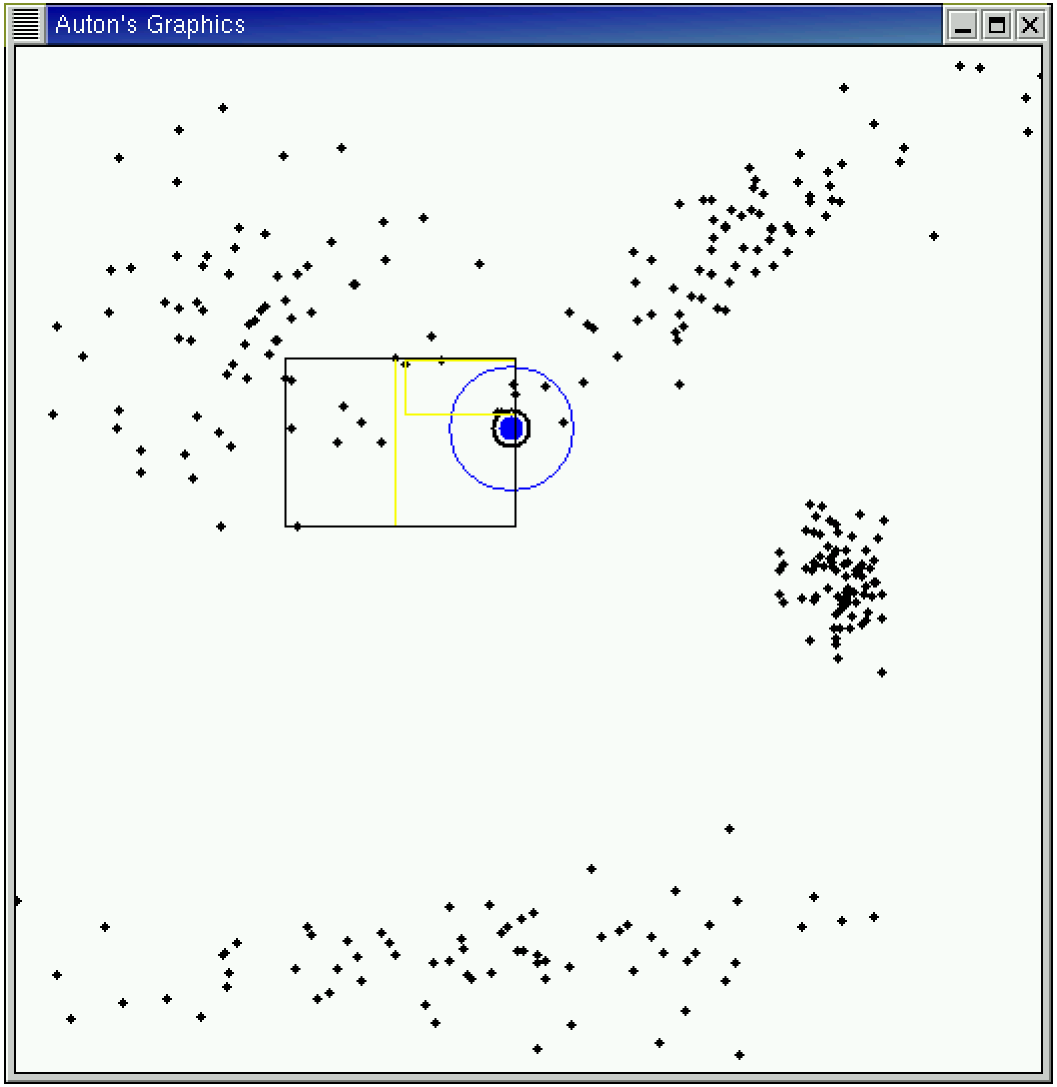


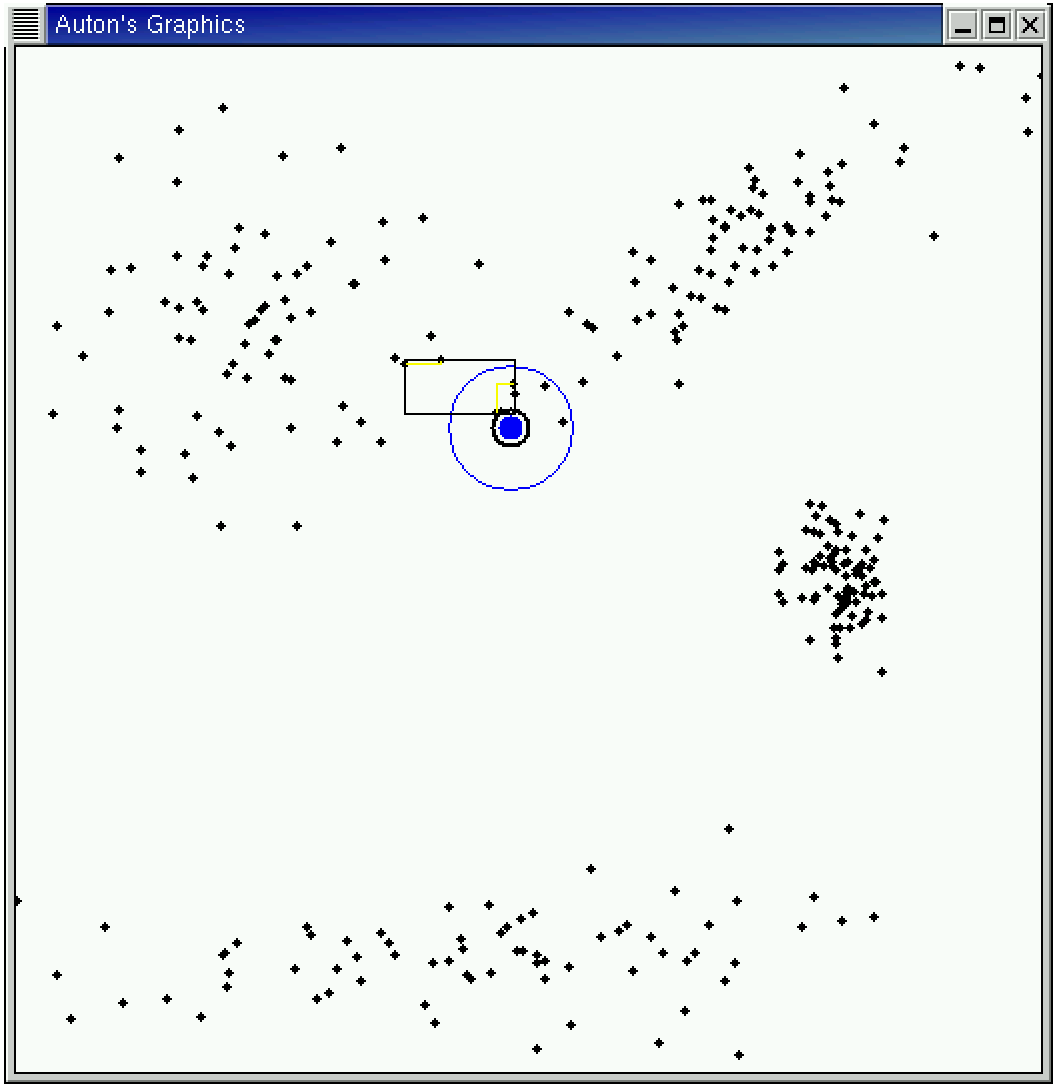

Auton's Graphics

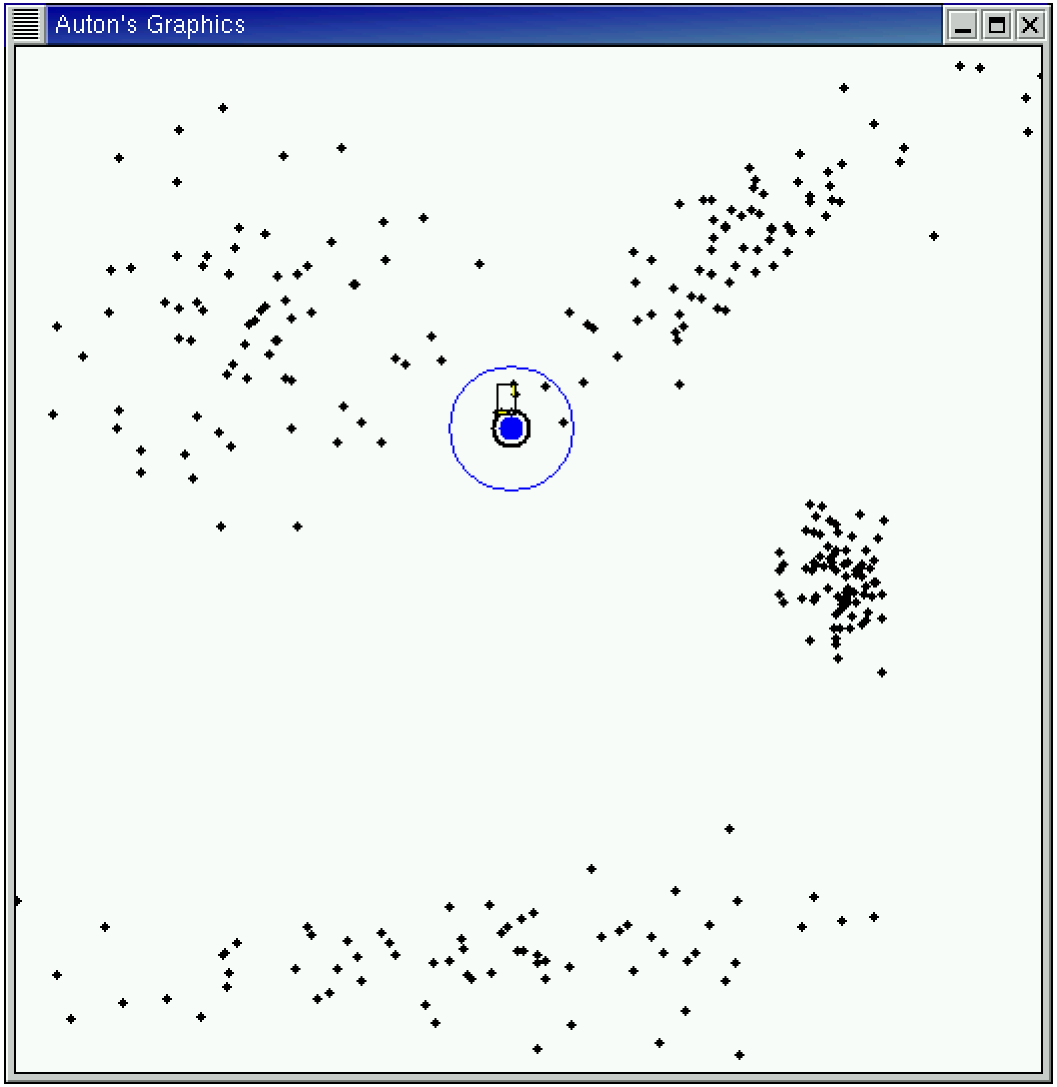

Auton's Graphics

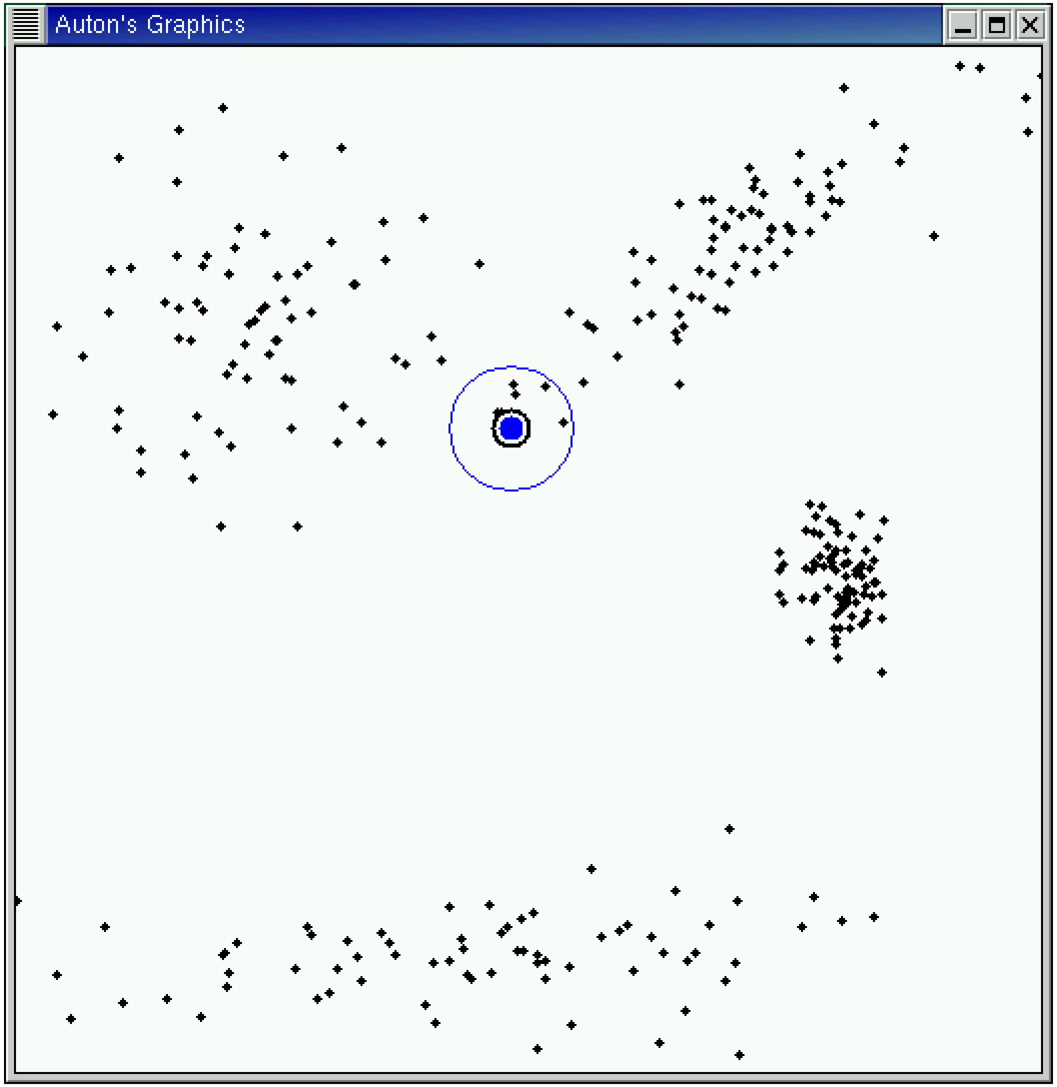

Auton's Graphics

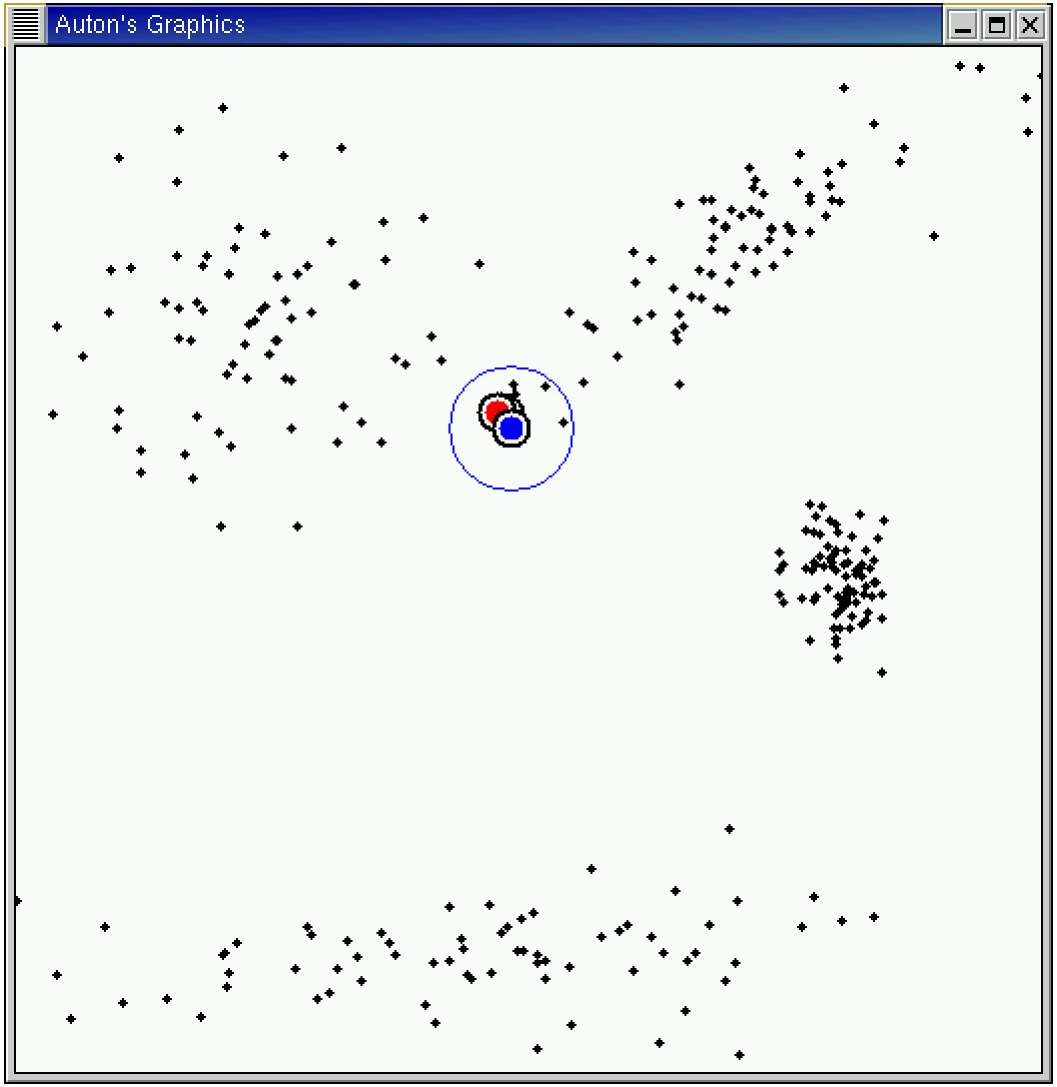

Auton's Graphics

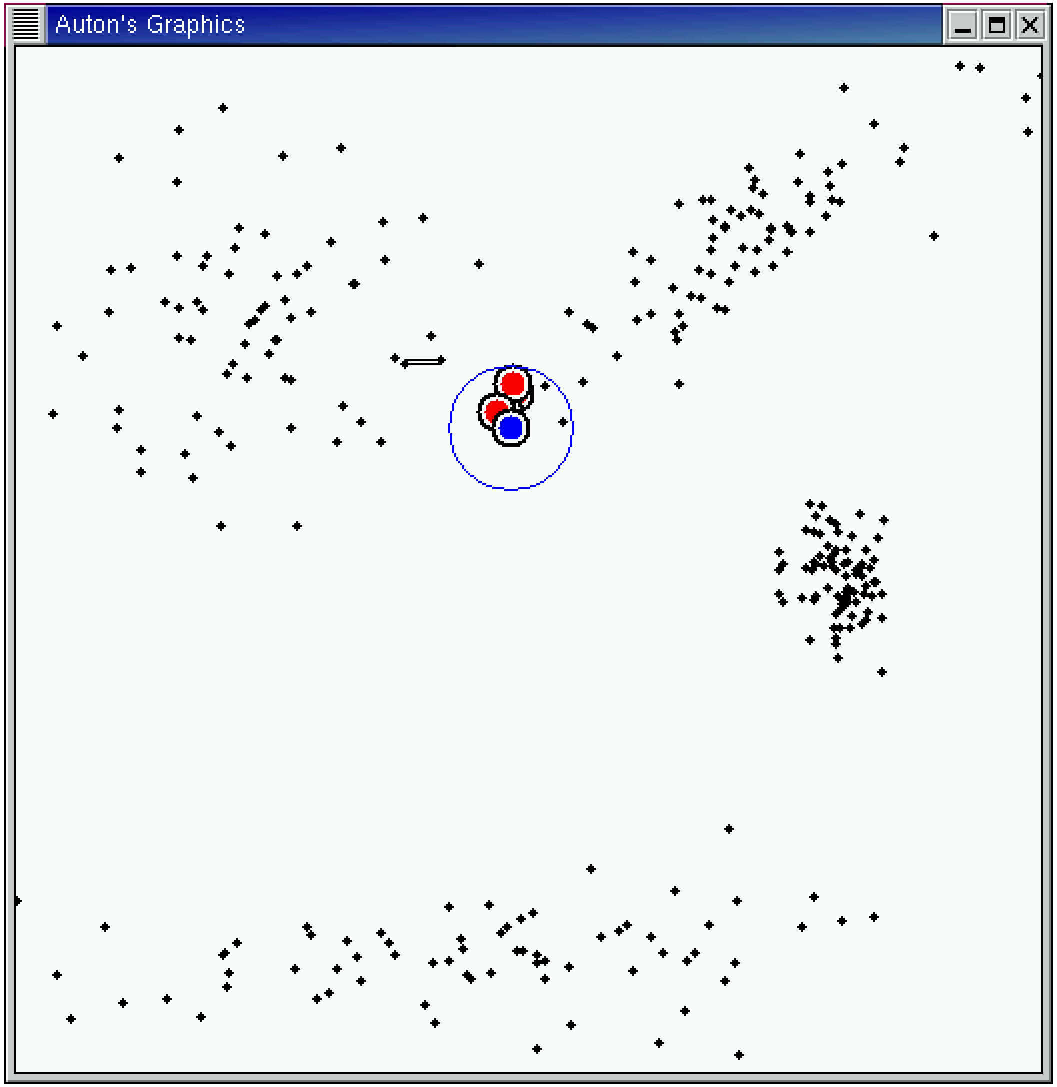

Auton's Graphics

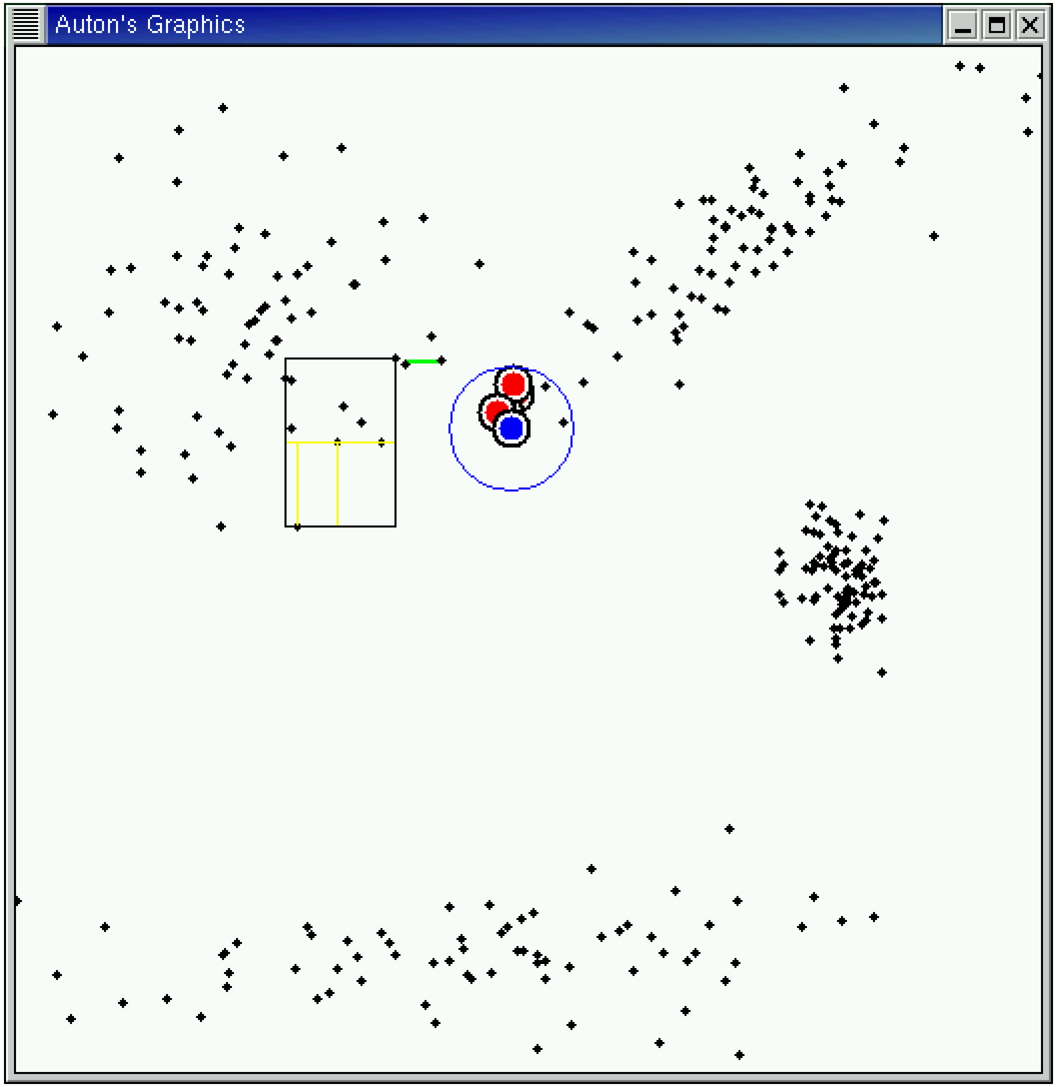

Auton's Graphics


Auton's Graphics

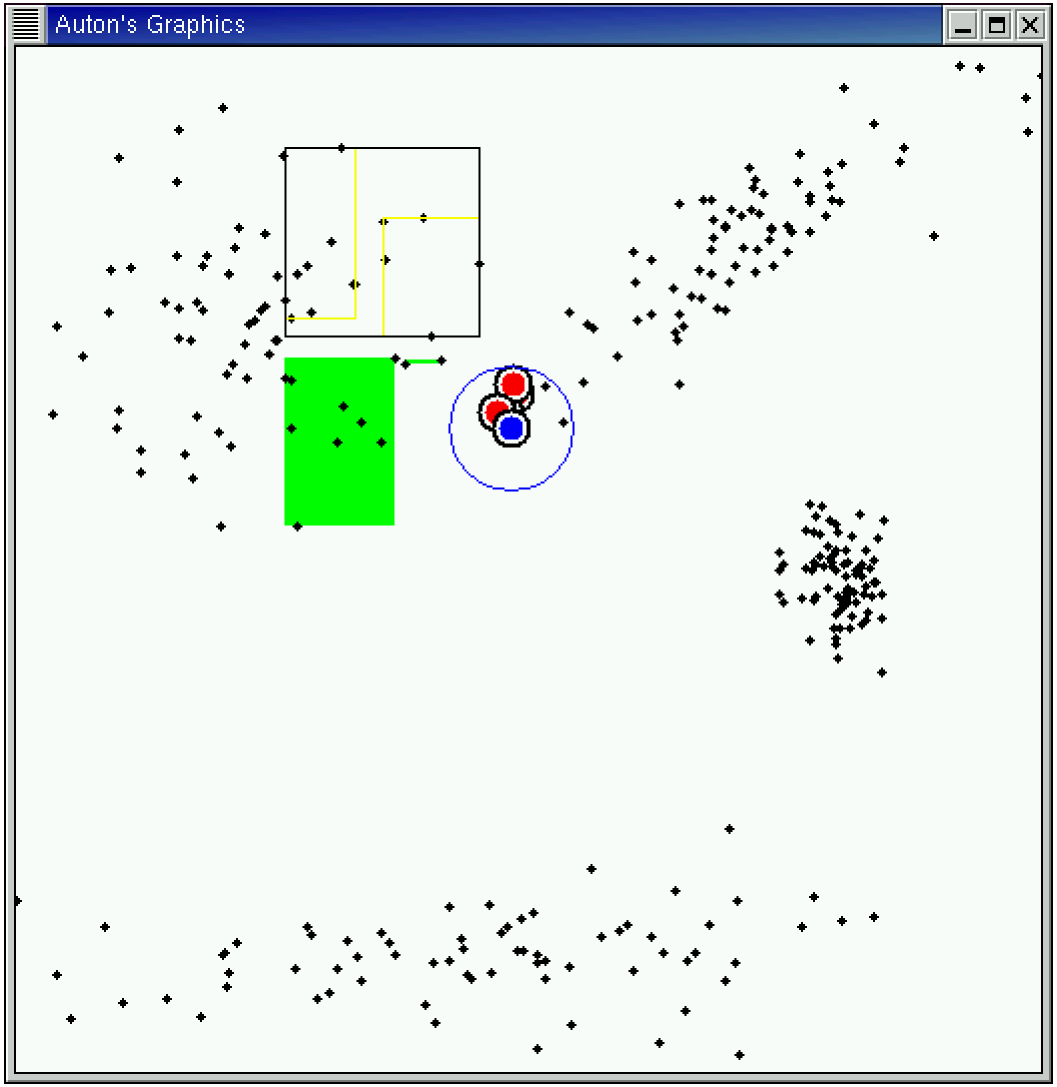

Auton's Graphics

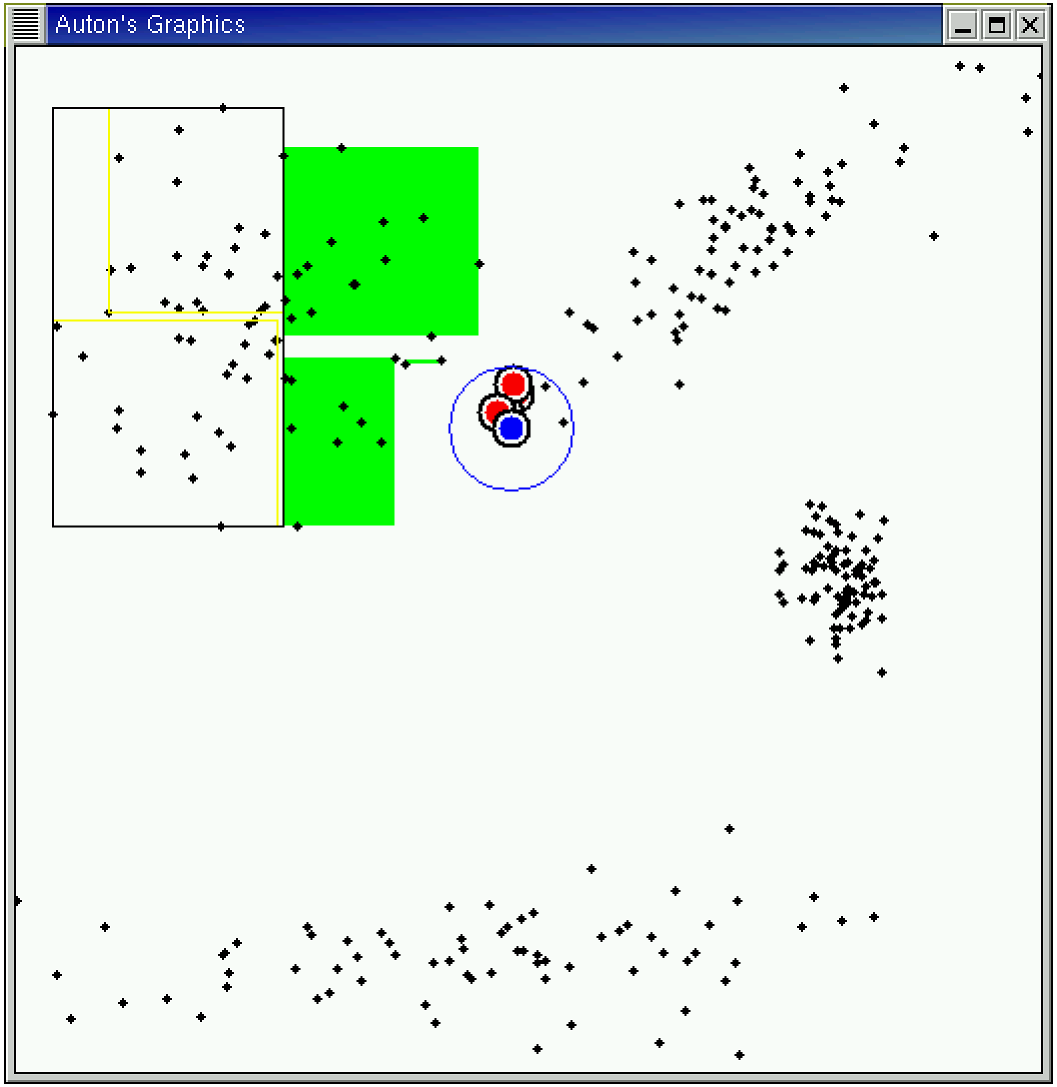

Auton's Graphics

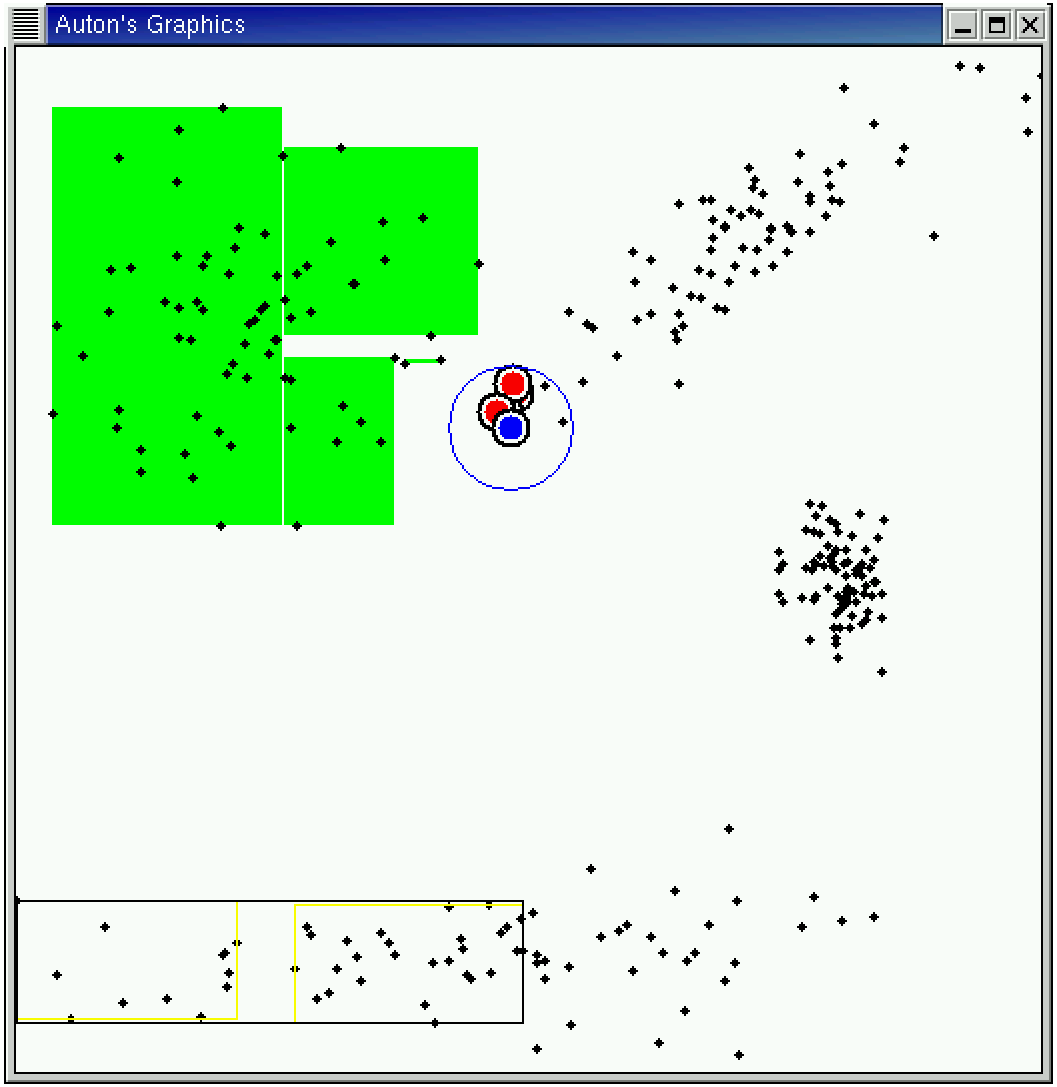

Auton's Graphics

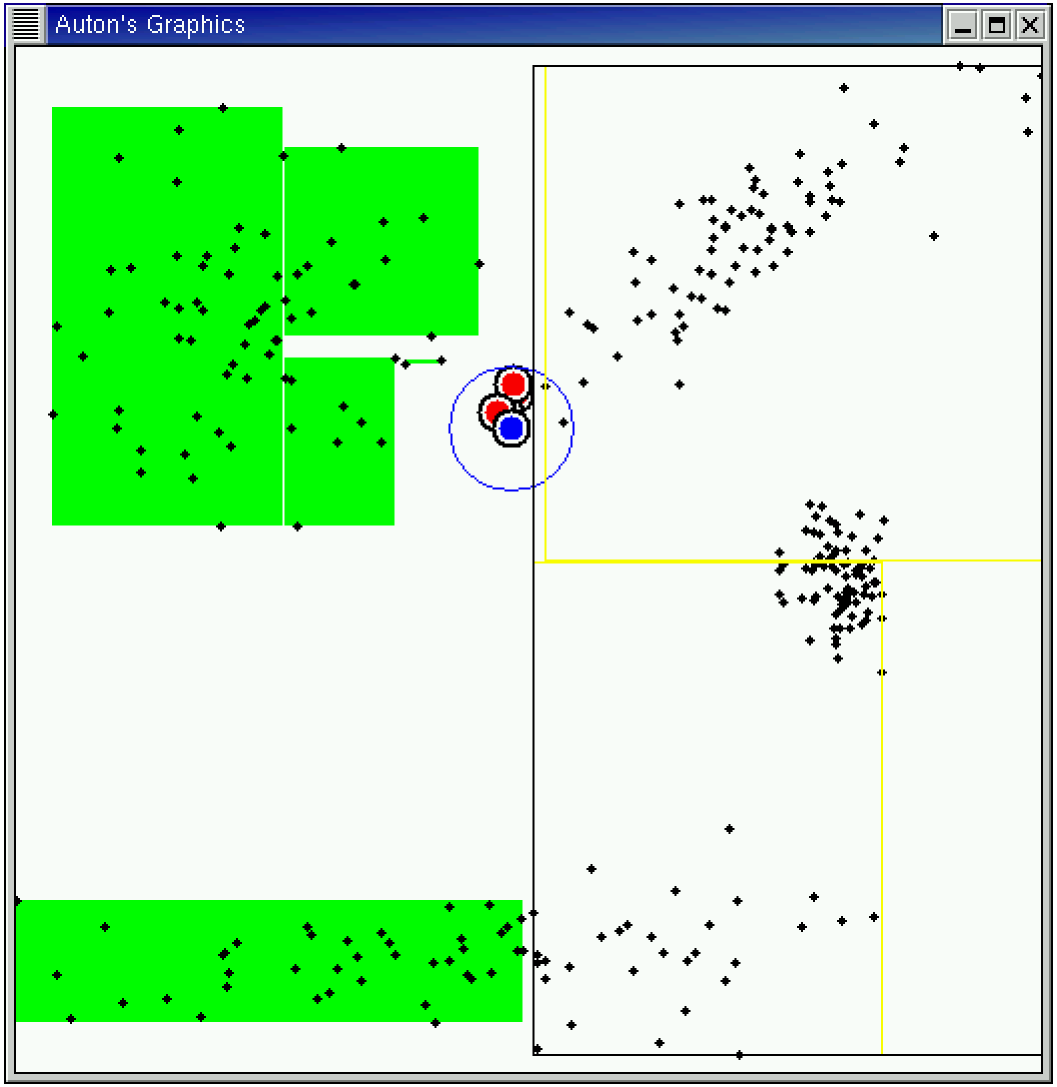


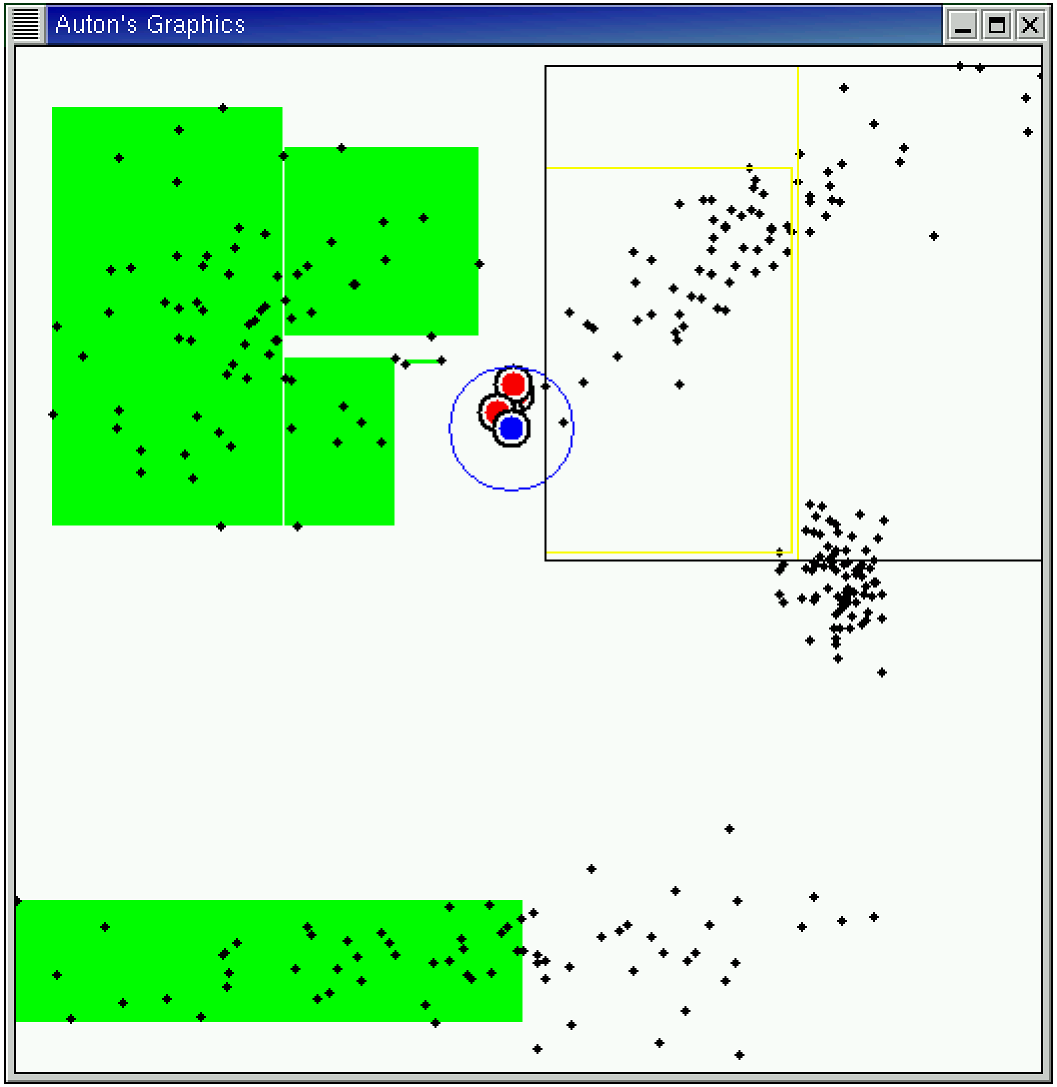
Auton's Graphics

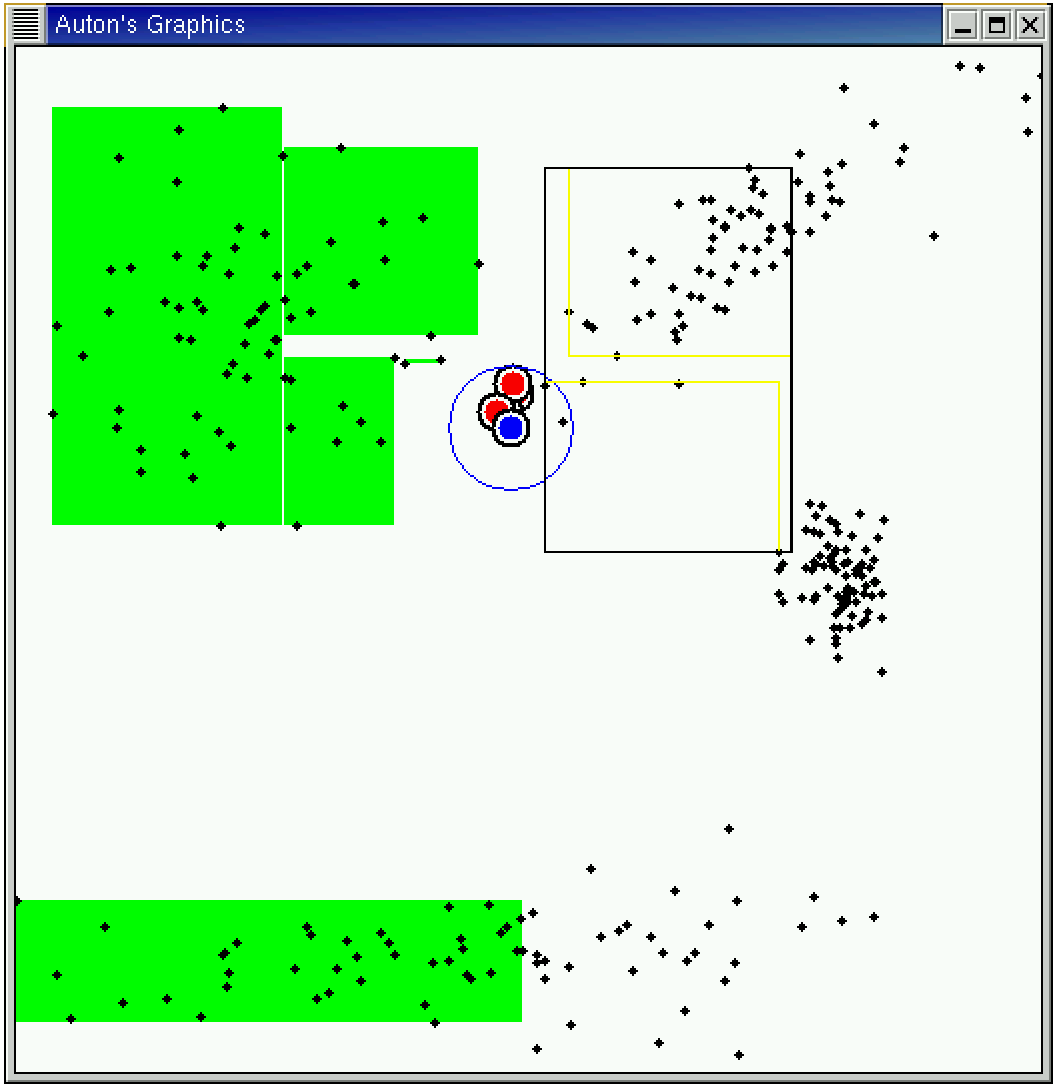

Auton's Graphics

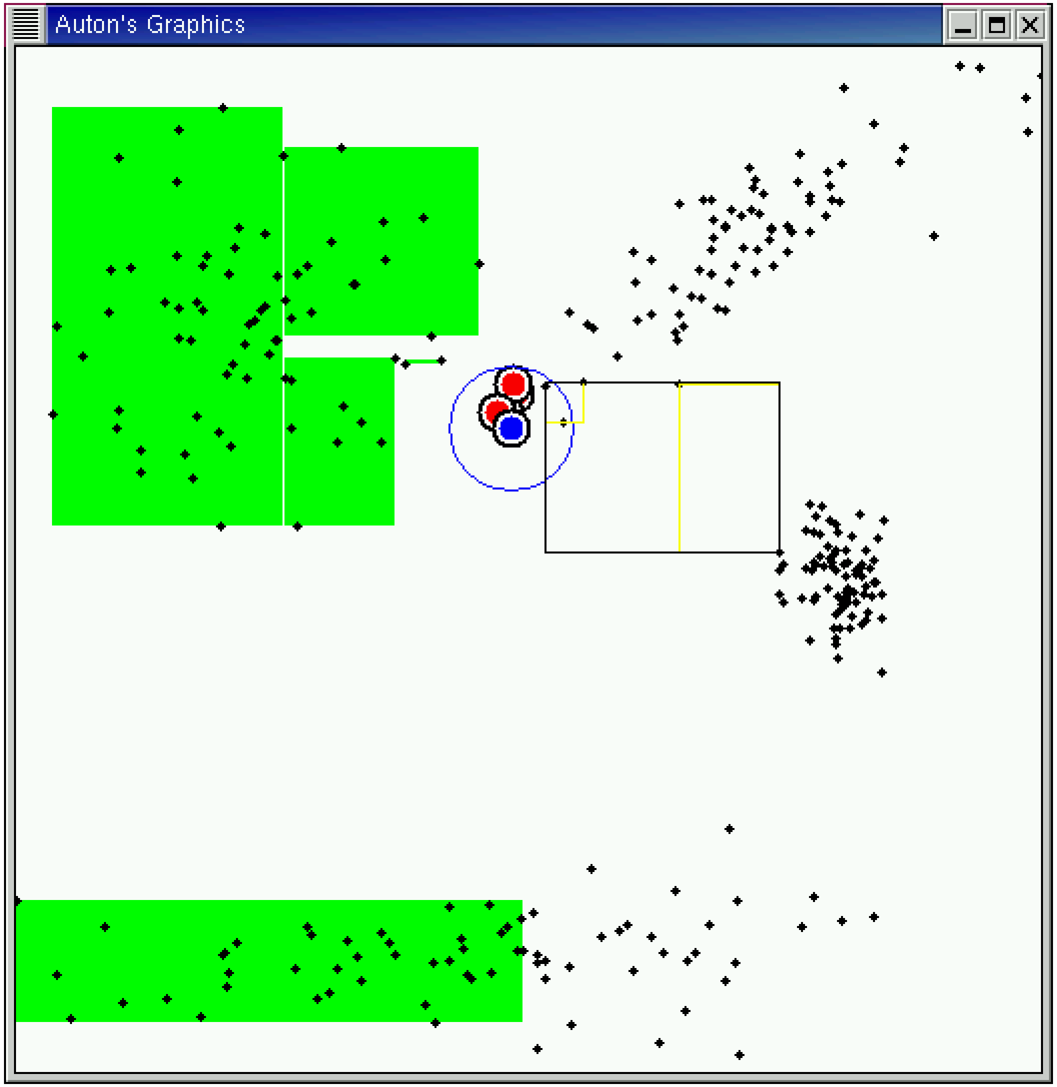

Auton's Graphics

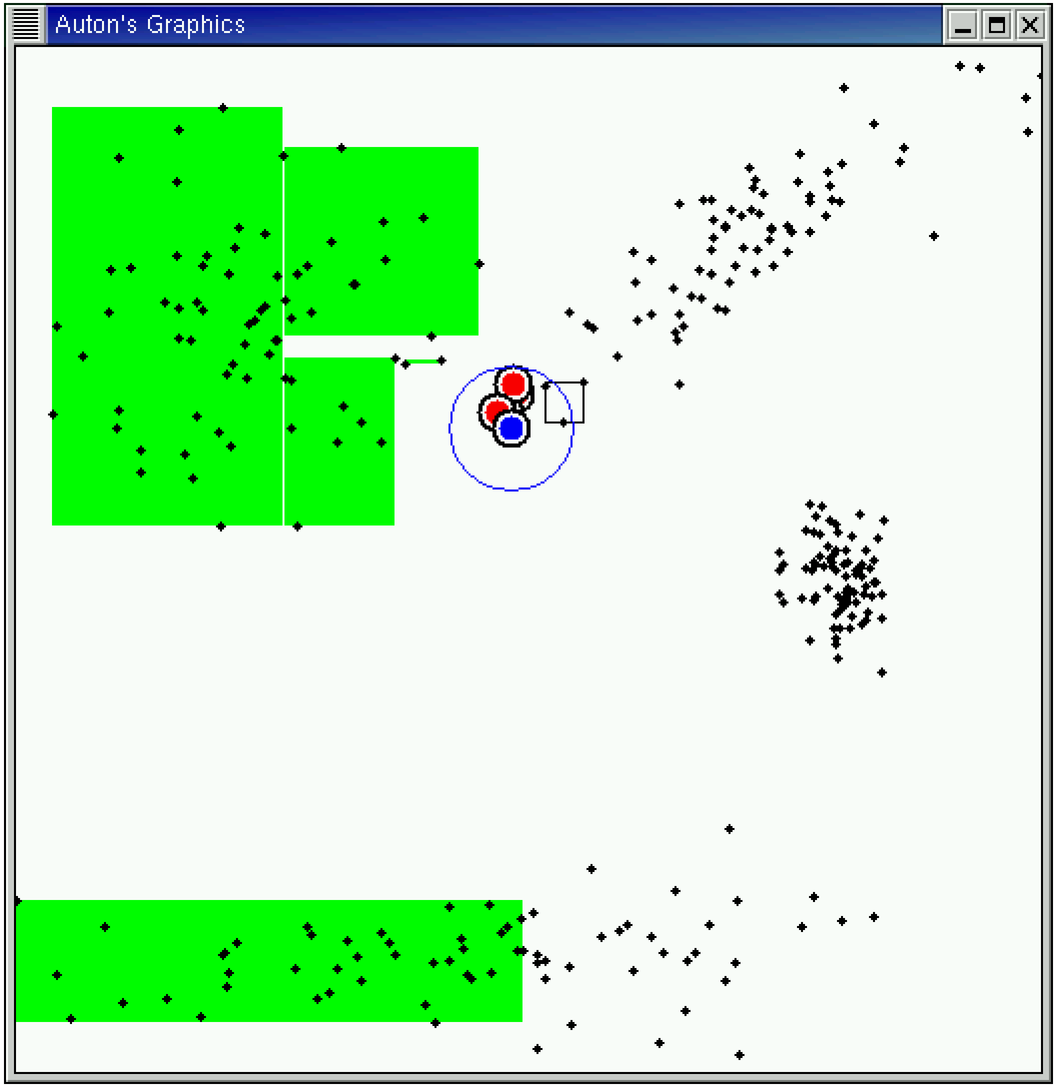

Auton's Graphics

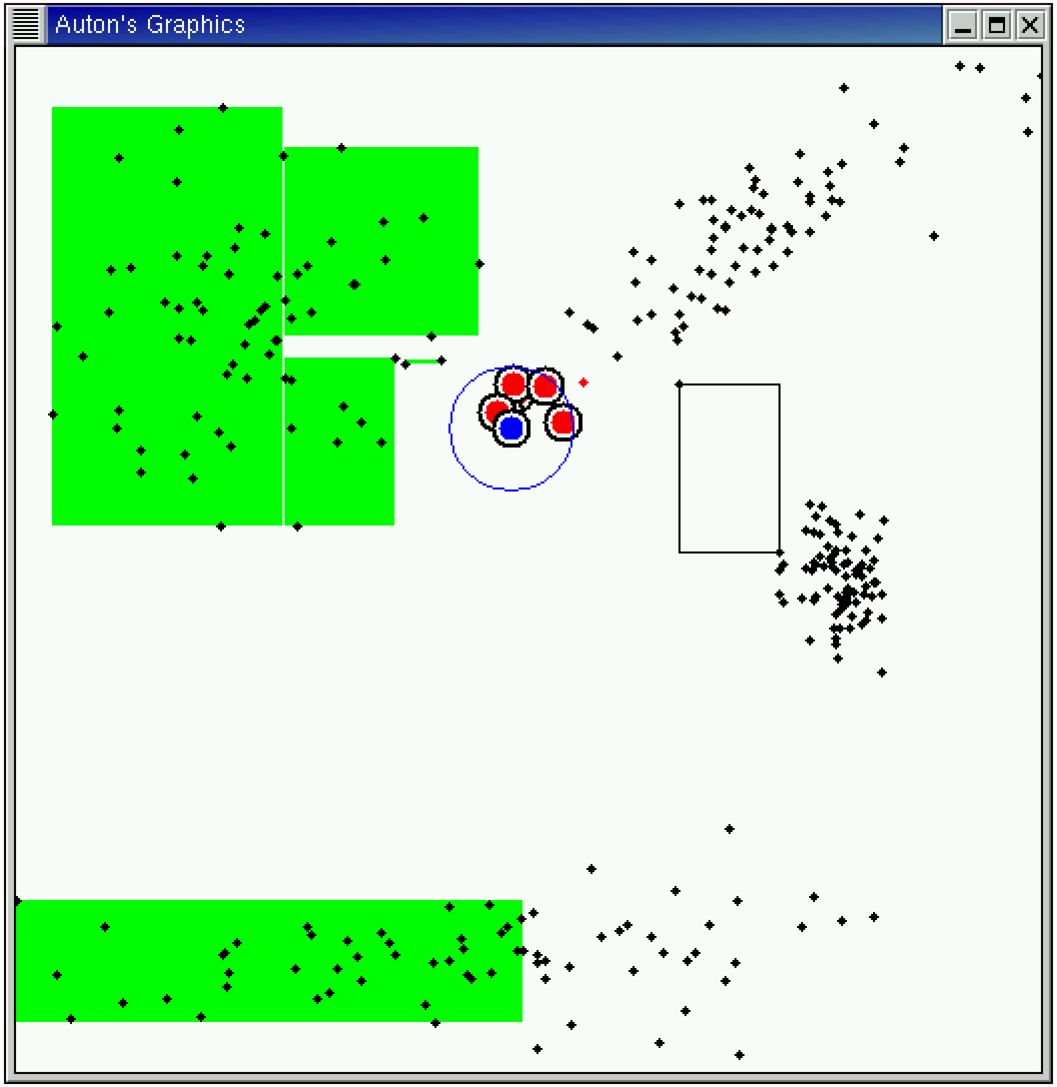

Auton's Graphics

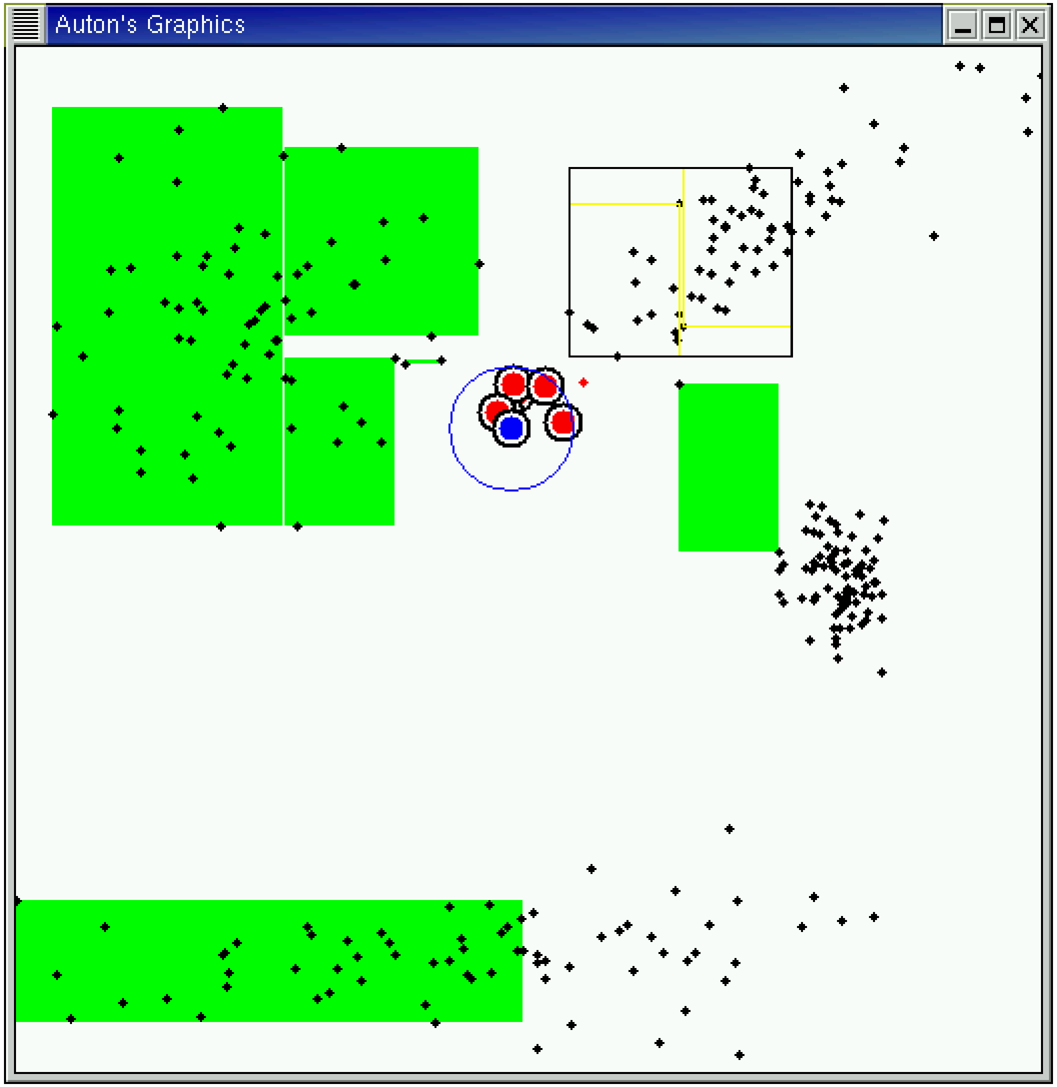

Auton's Graphics

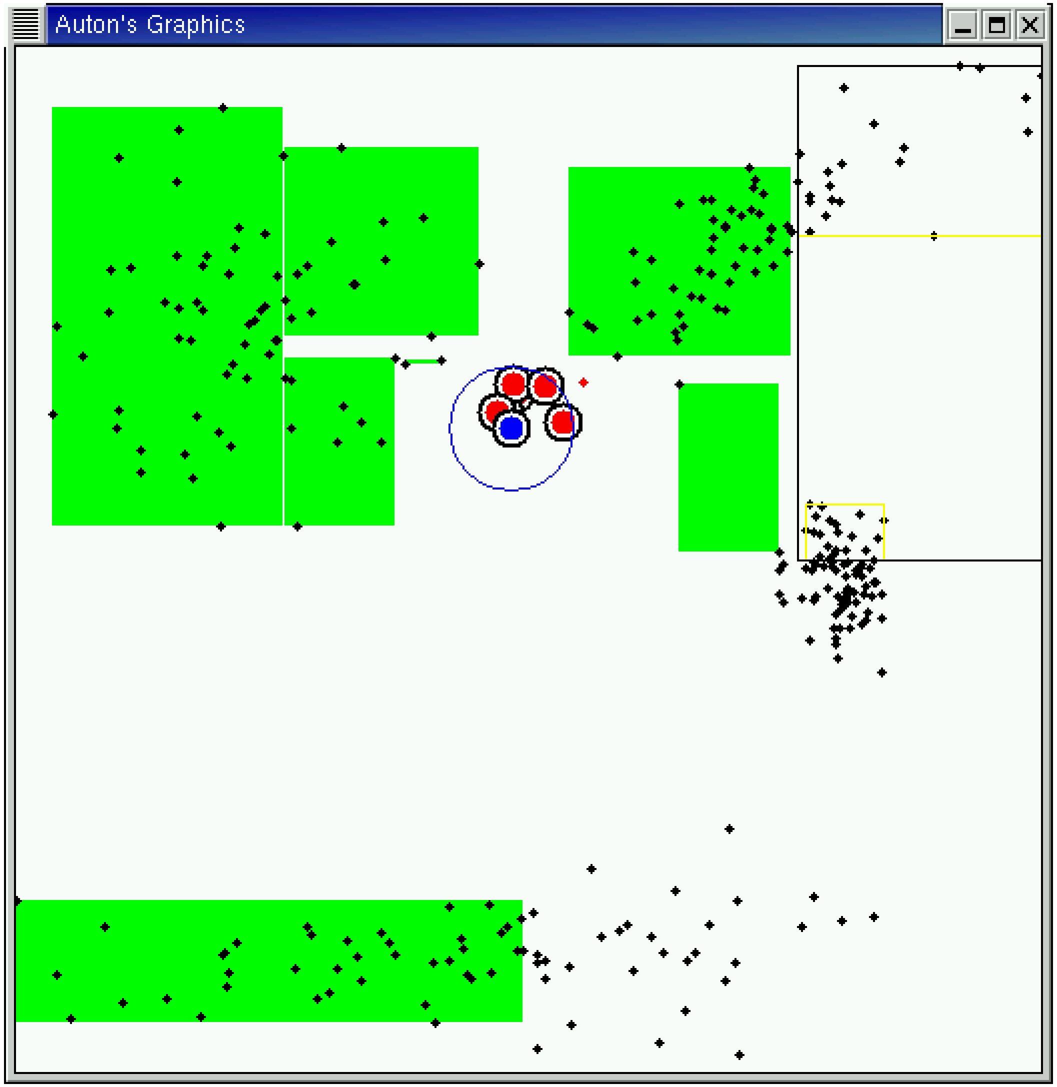

Auton's Graphics

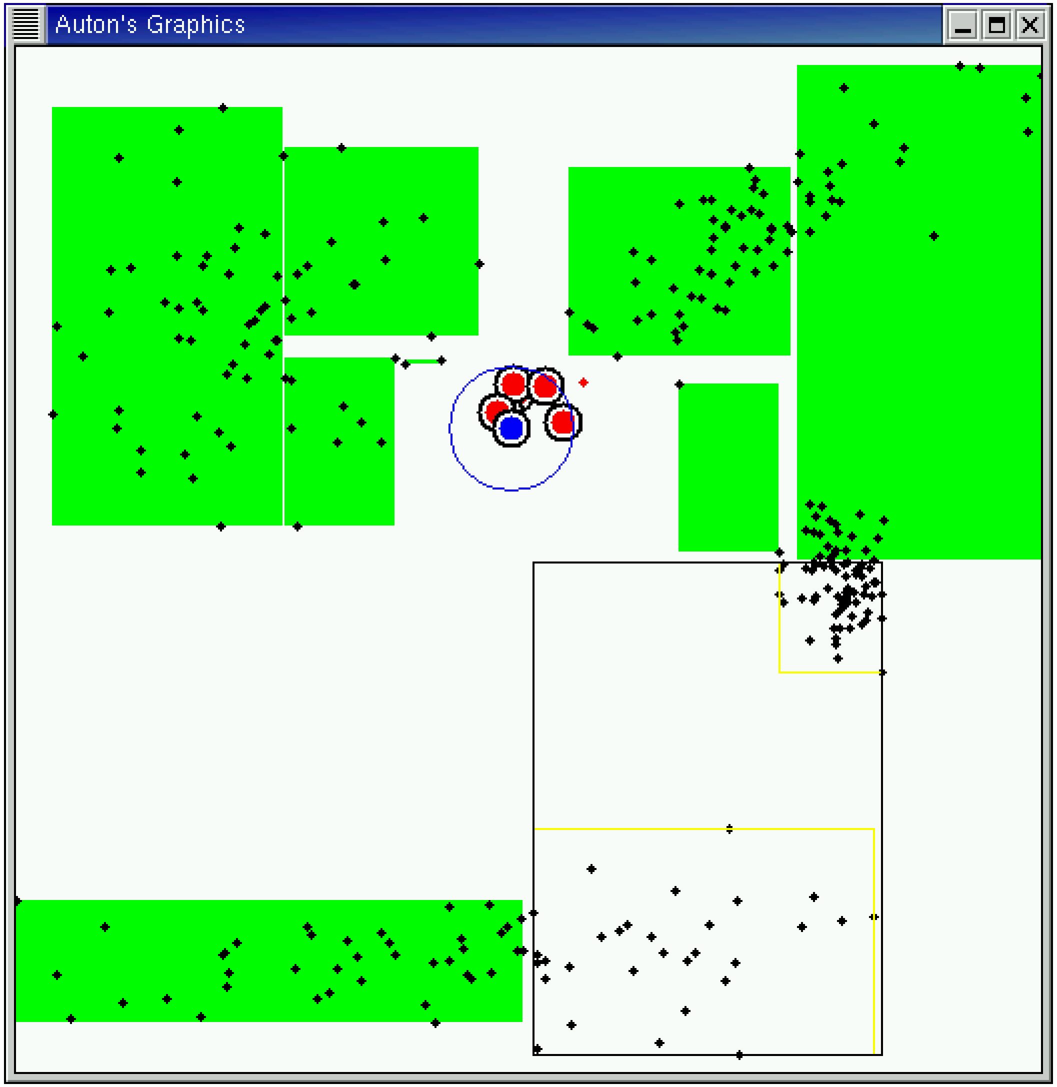

Auton's Graphics

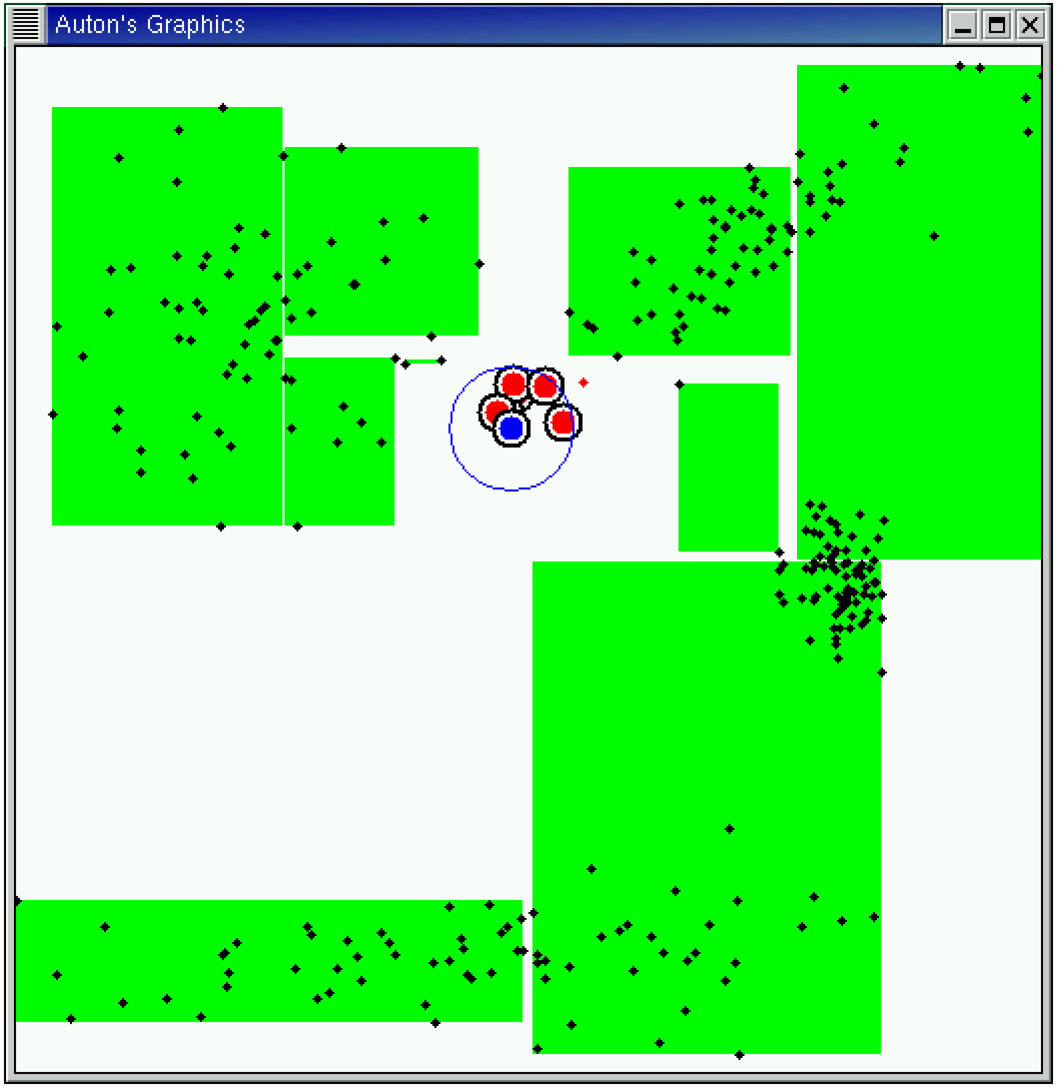

Auton's Graphics


Auton's Graphics


Auton's Graphics

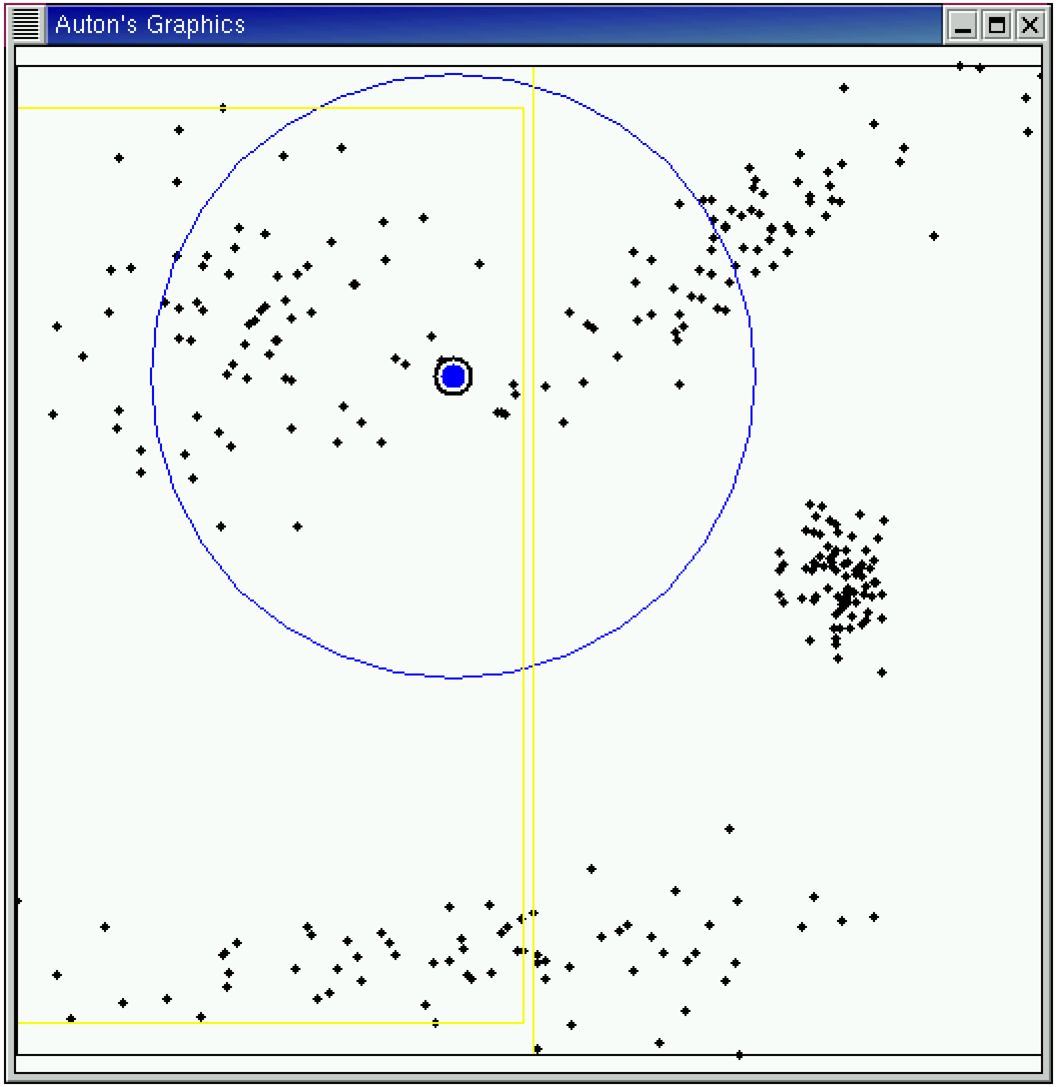

Auton's Graphics


Auton's Graphics

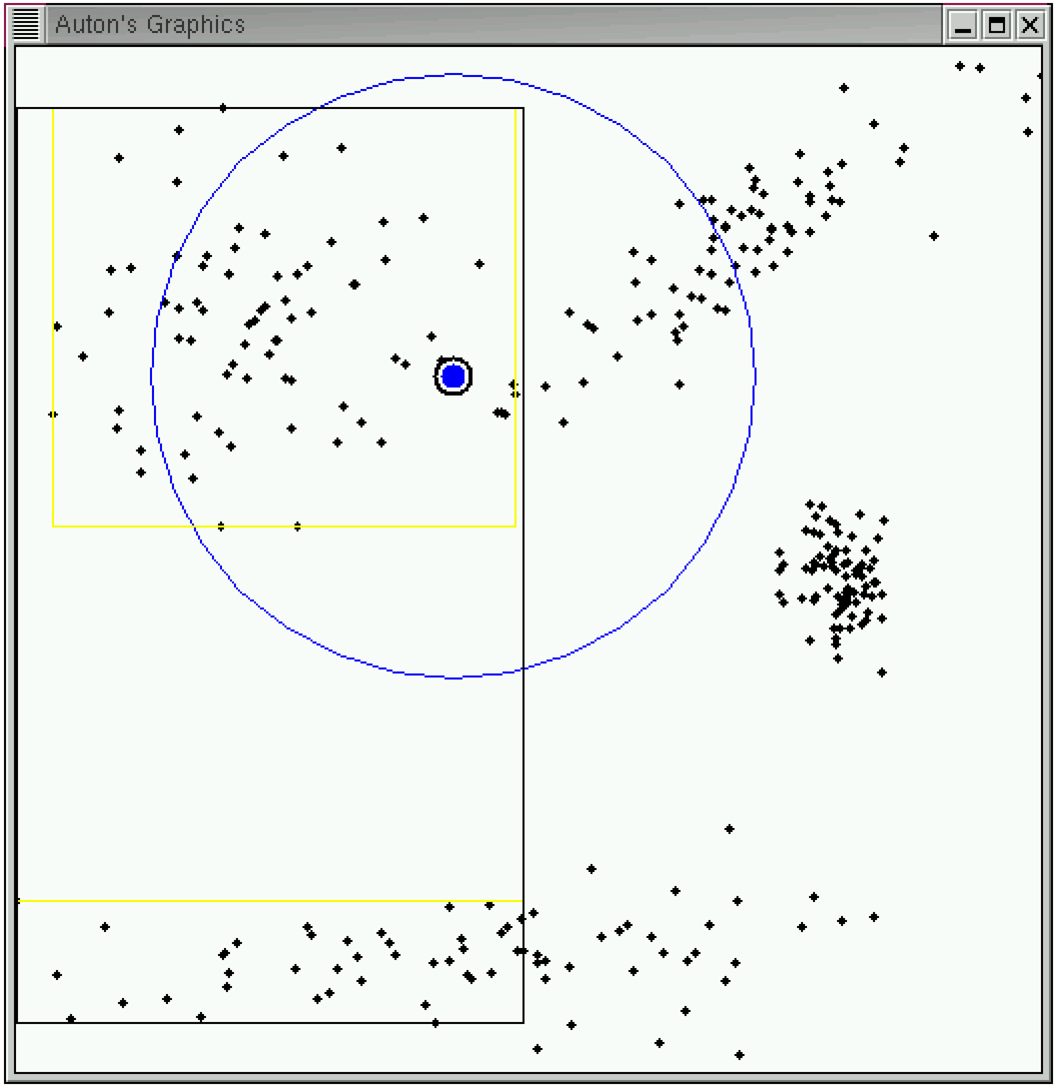


Auton's Graphics

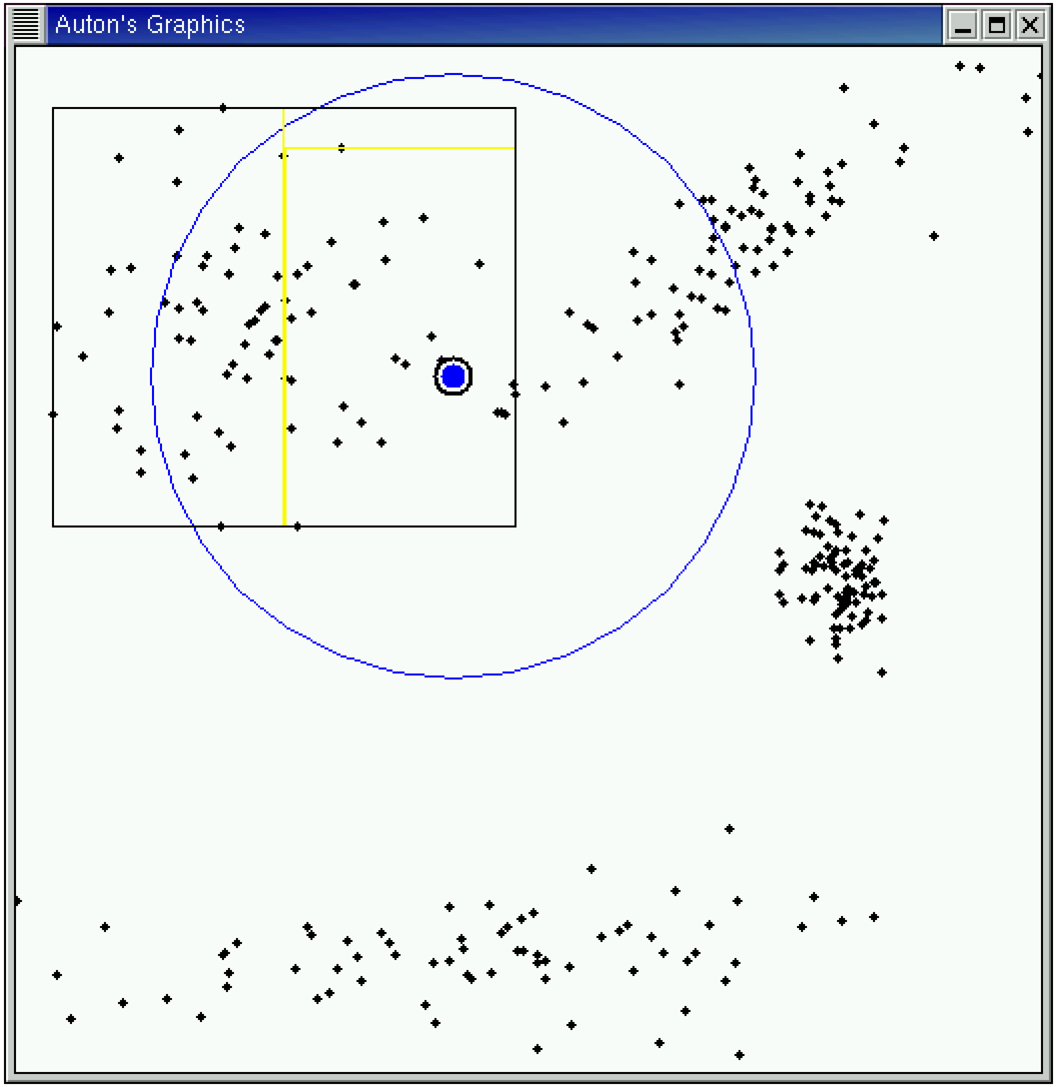
Range Count



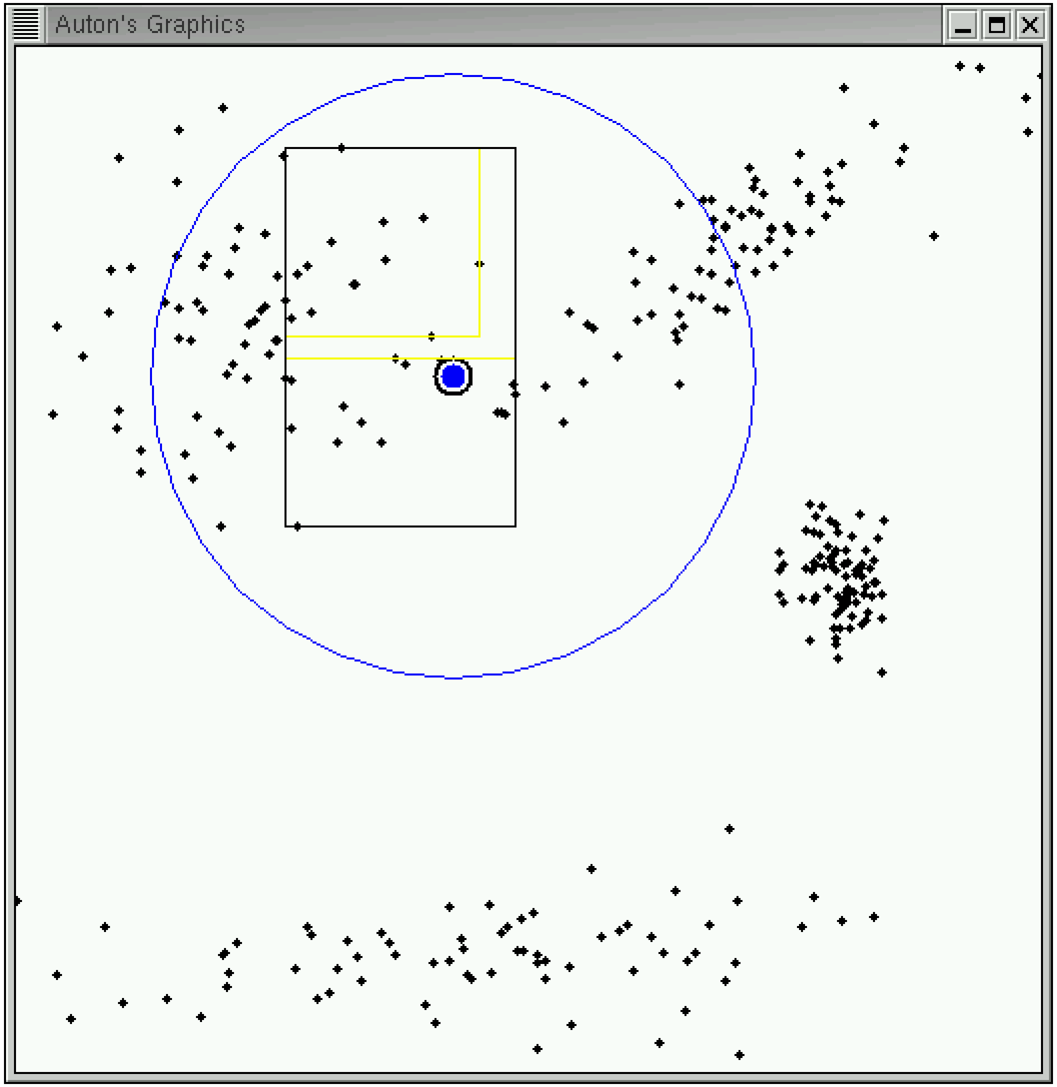
Range Count



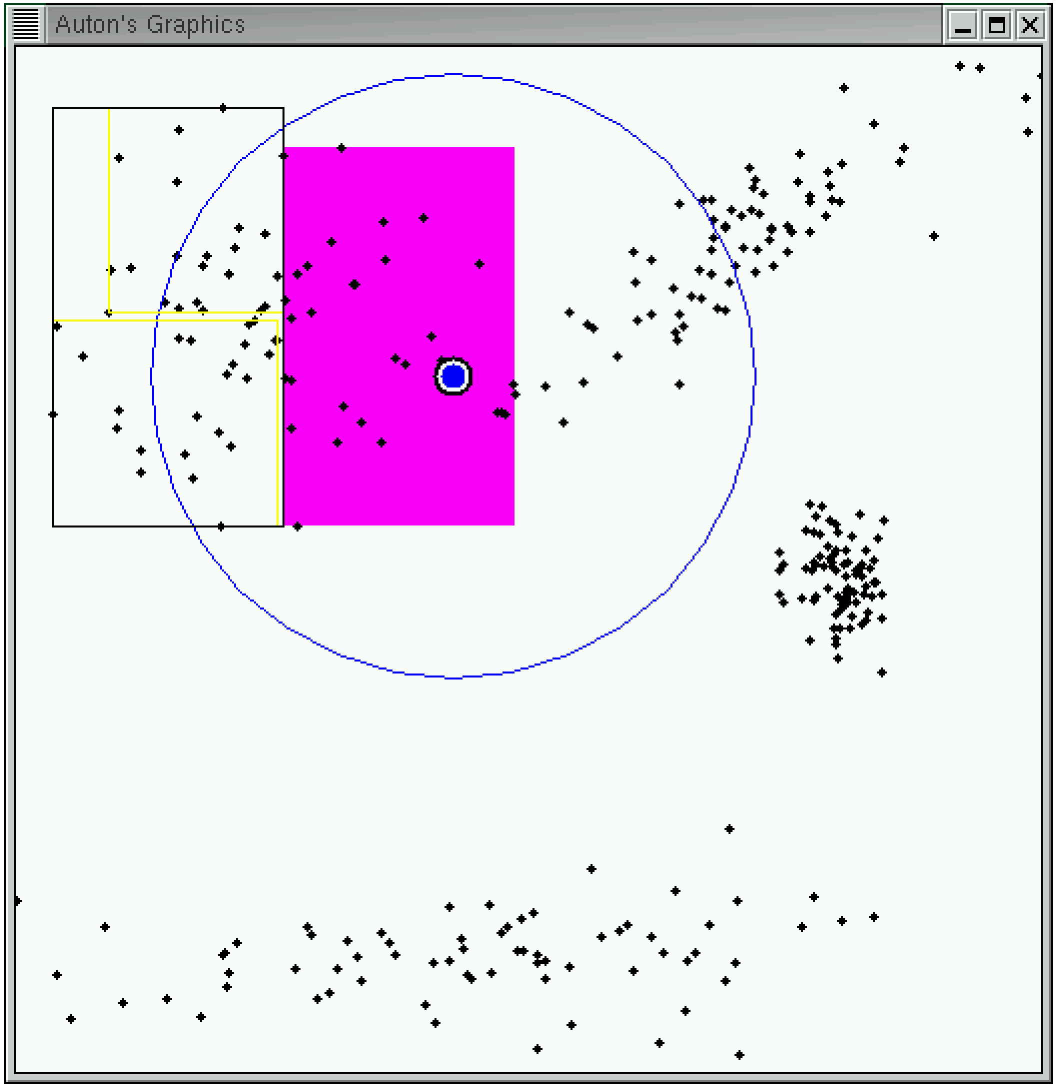
Range Count



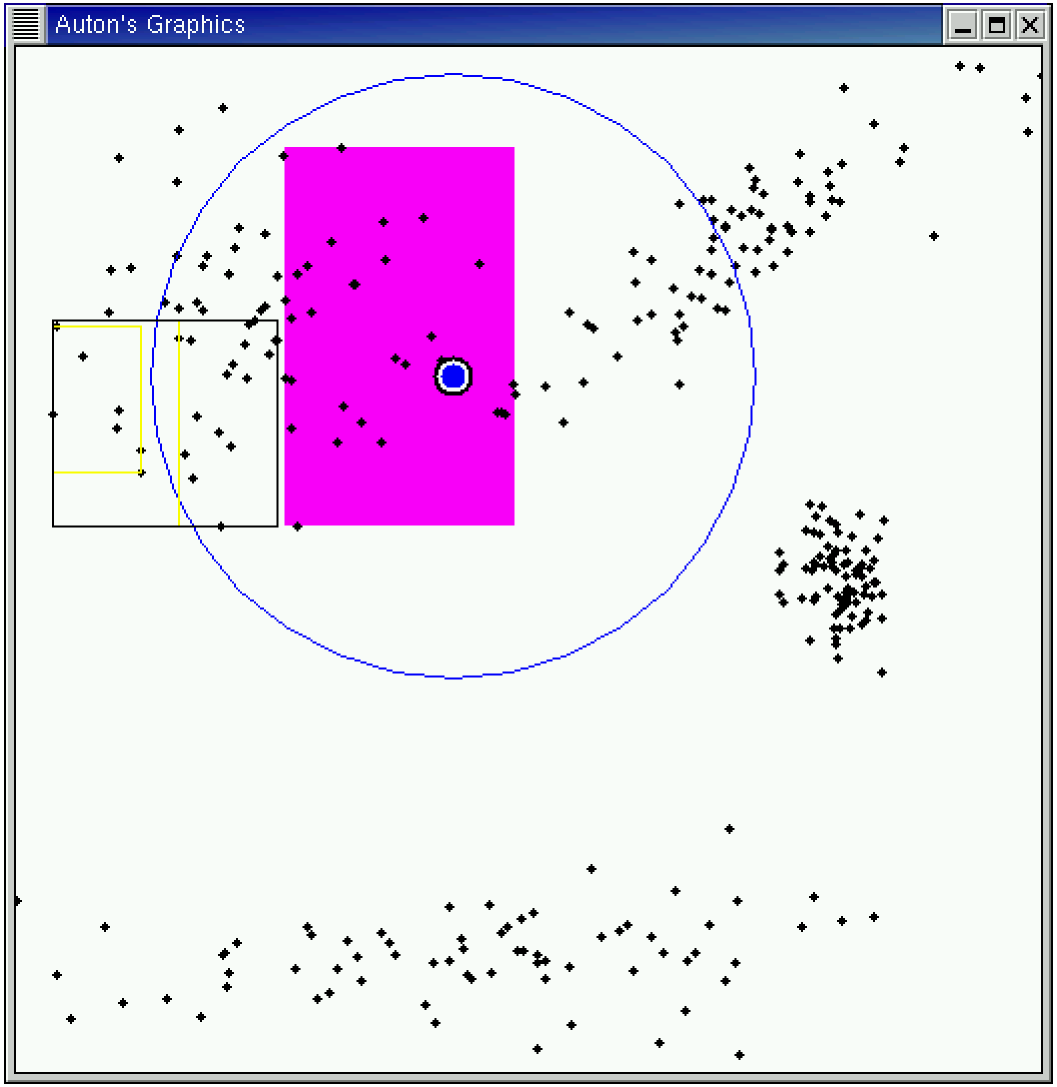
Range Count



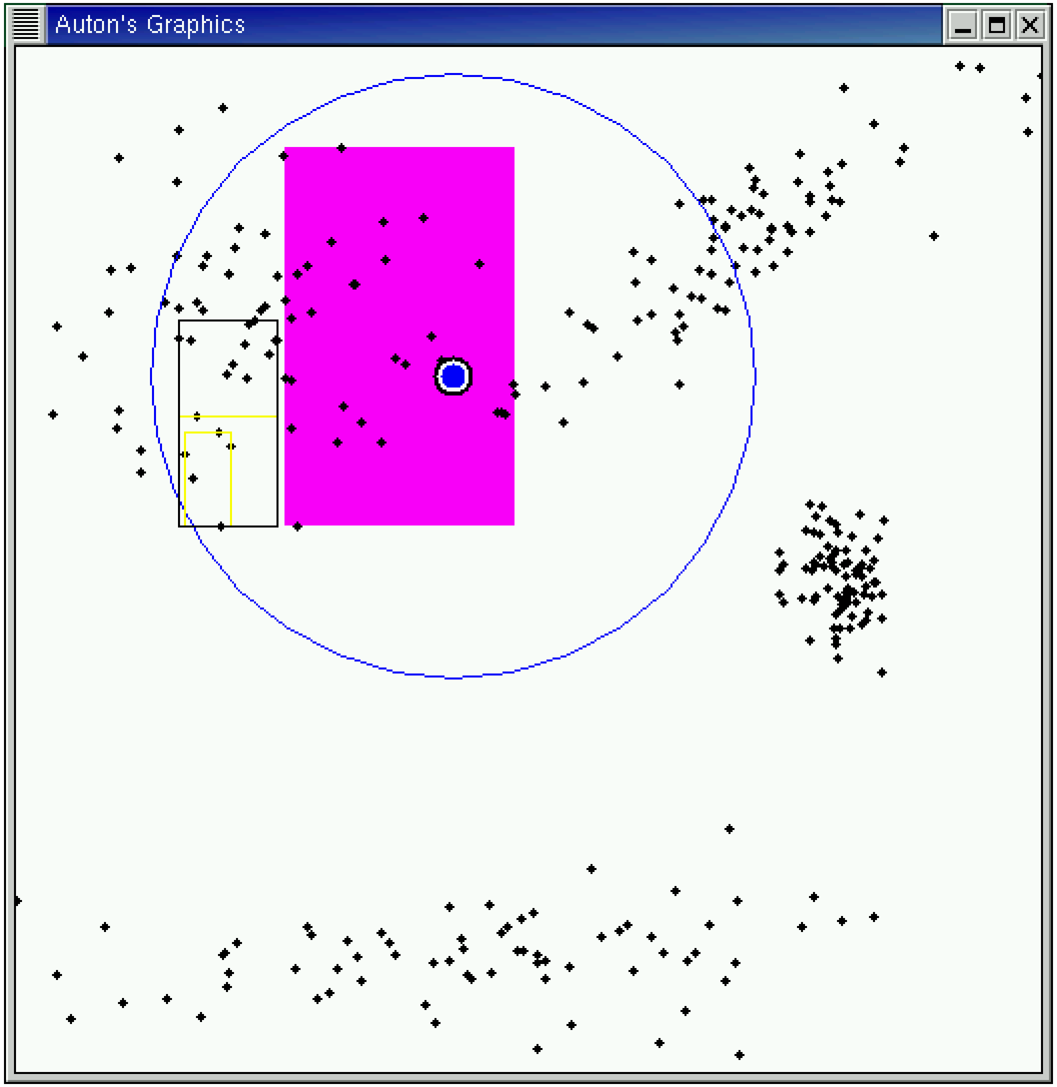
Range Count



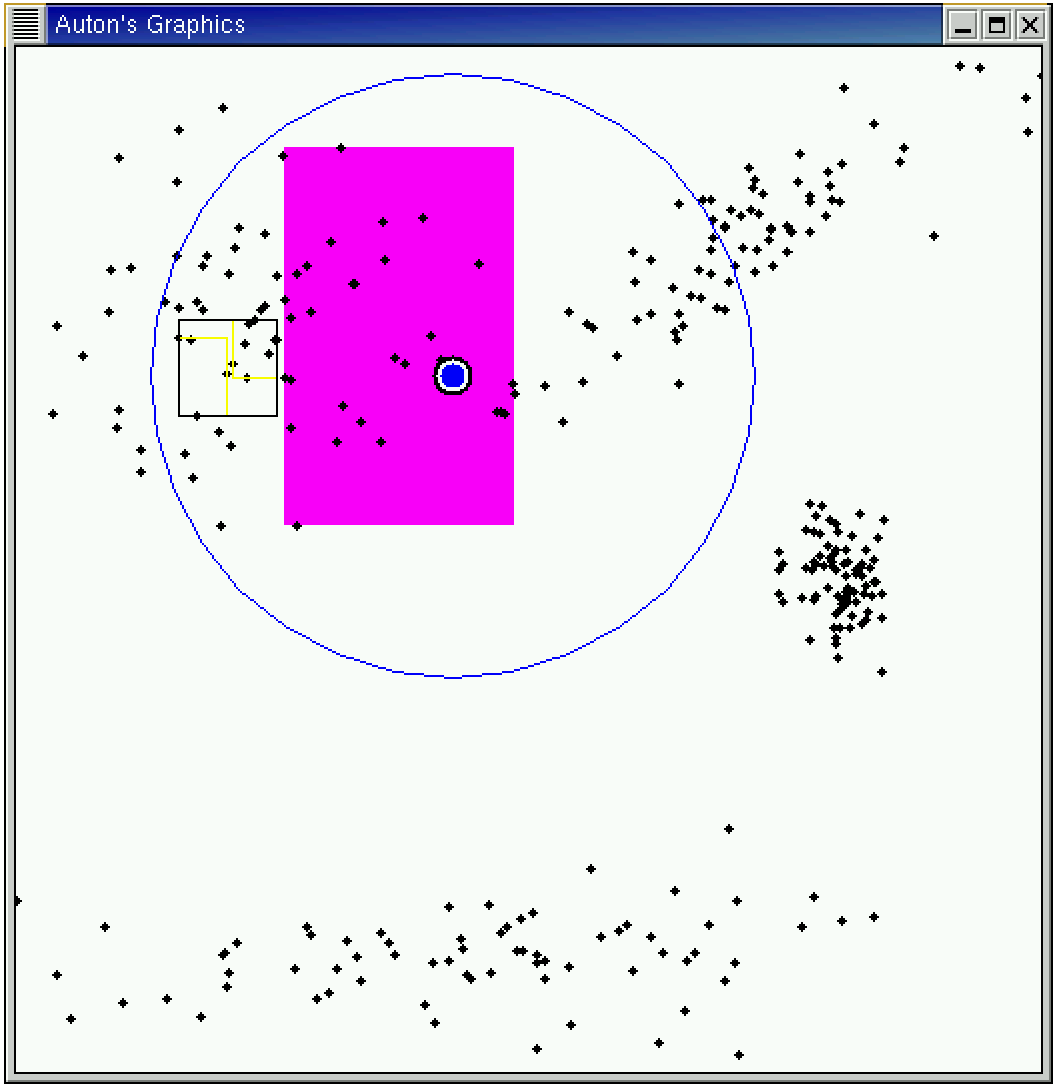
Range Count



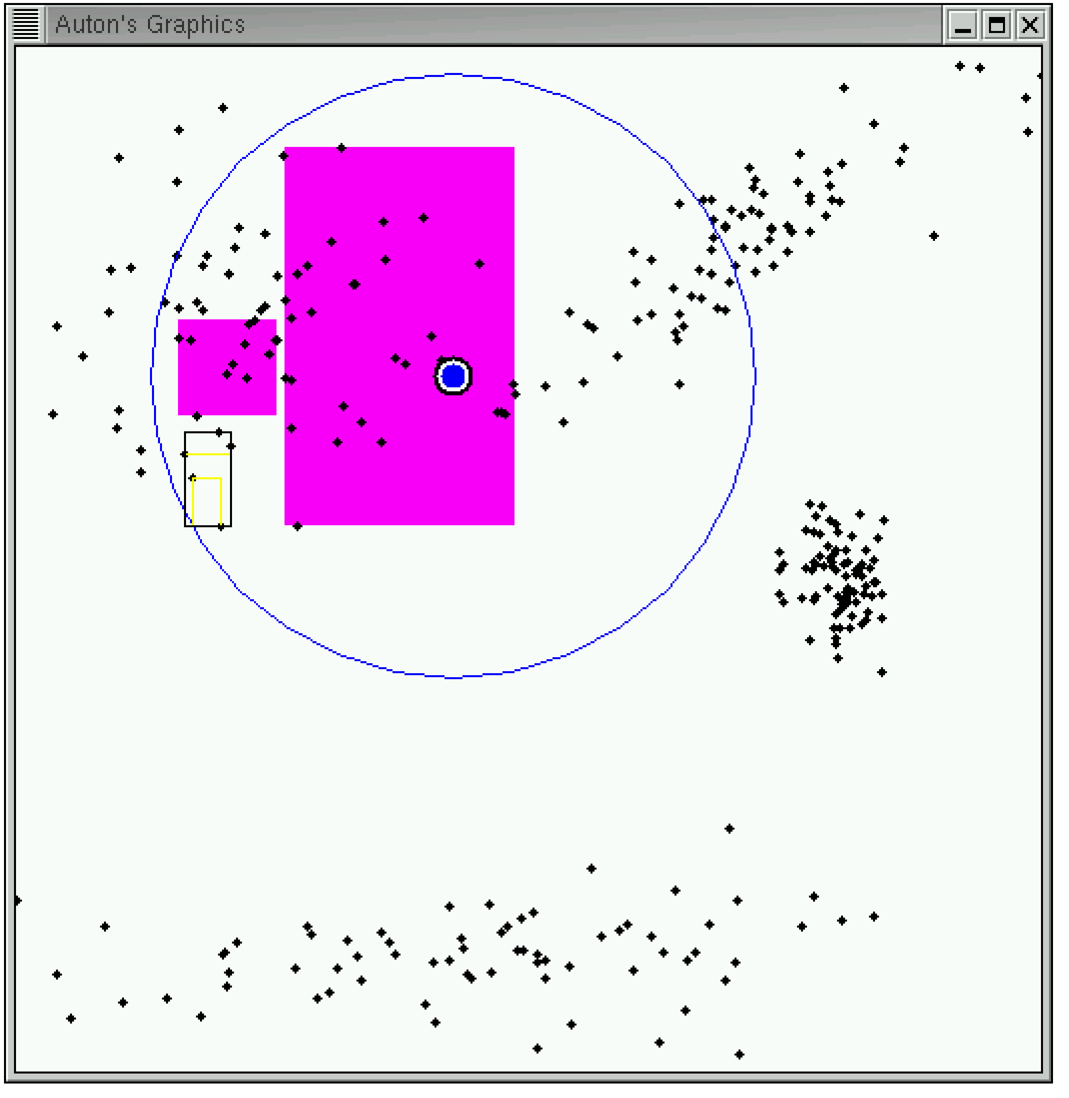
Range Count



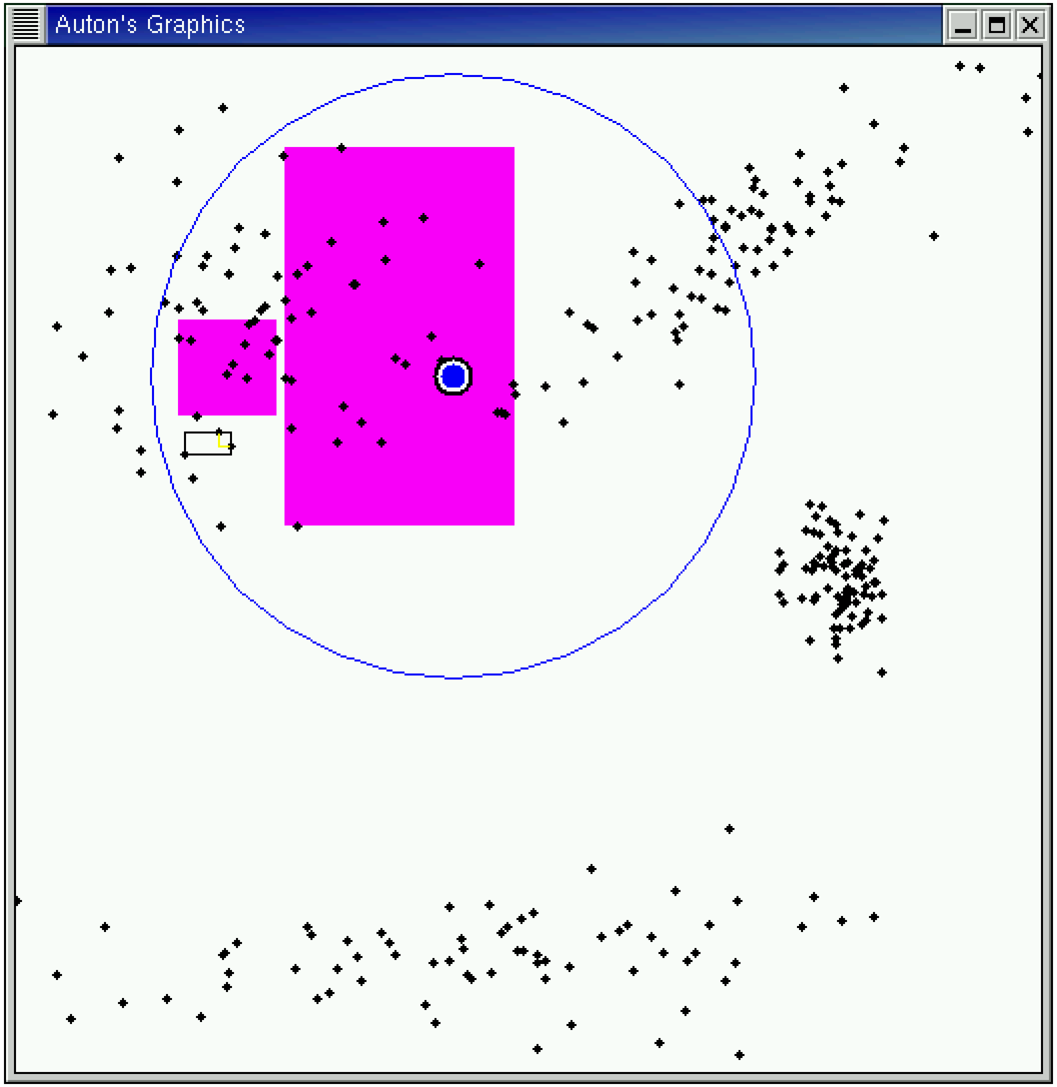
Range Count



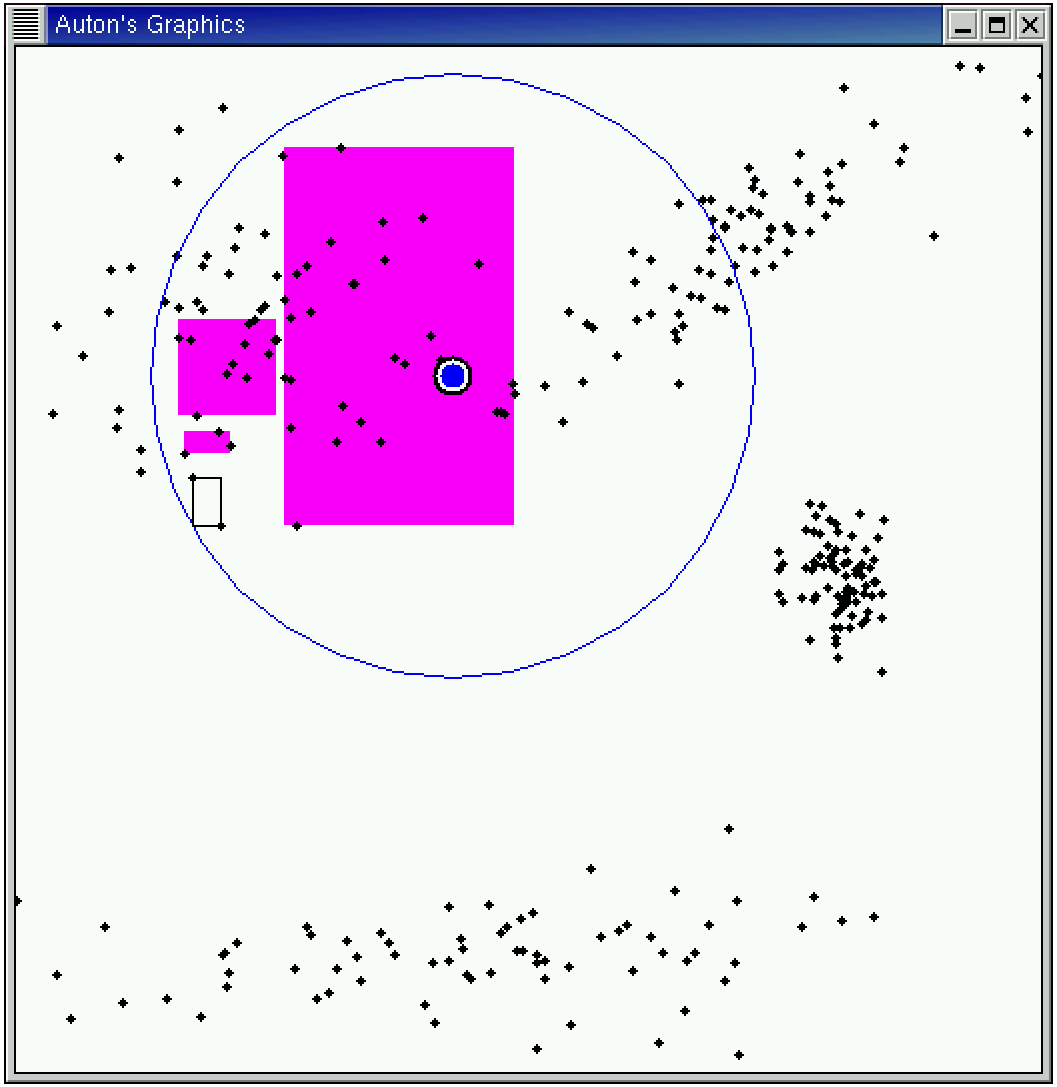
Range Count



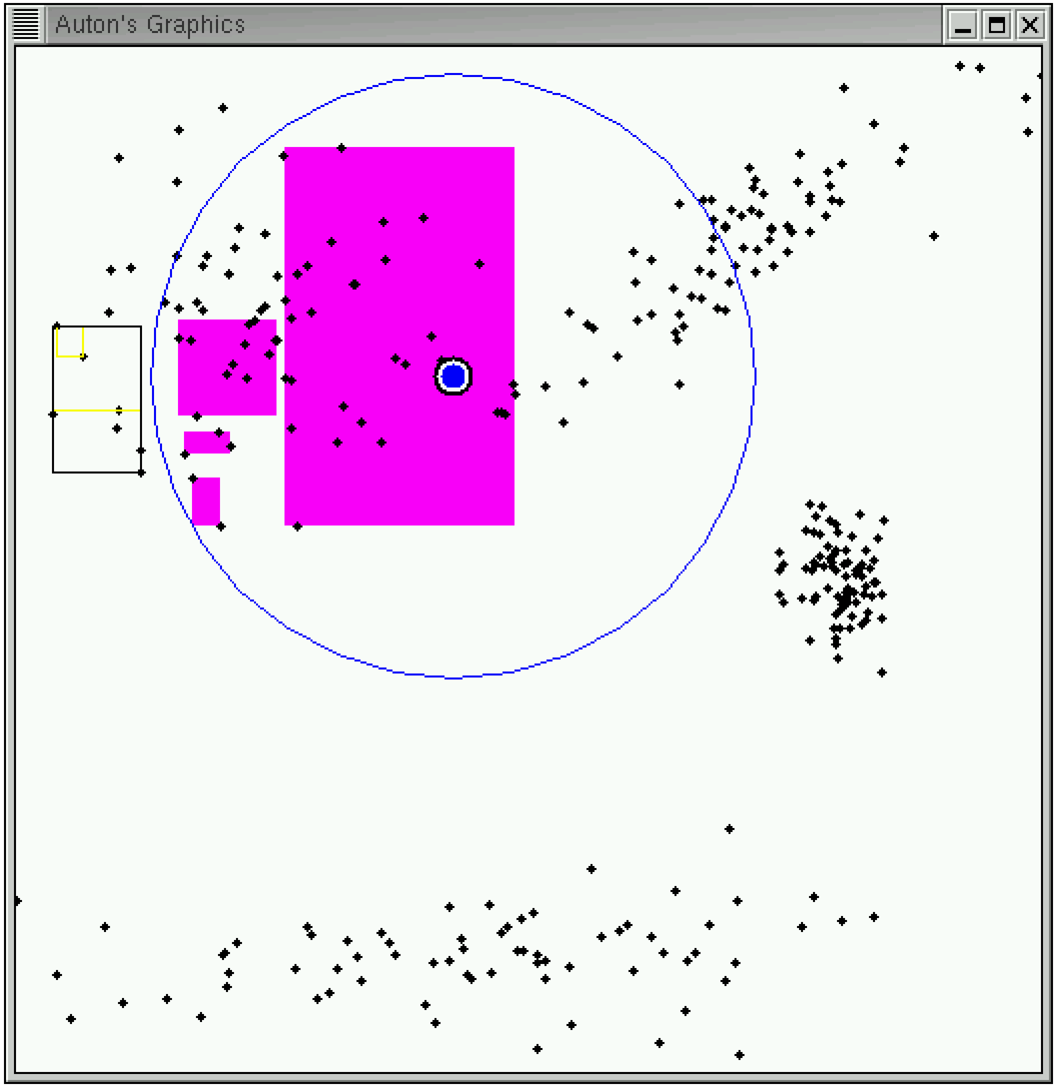
Range Count



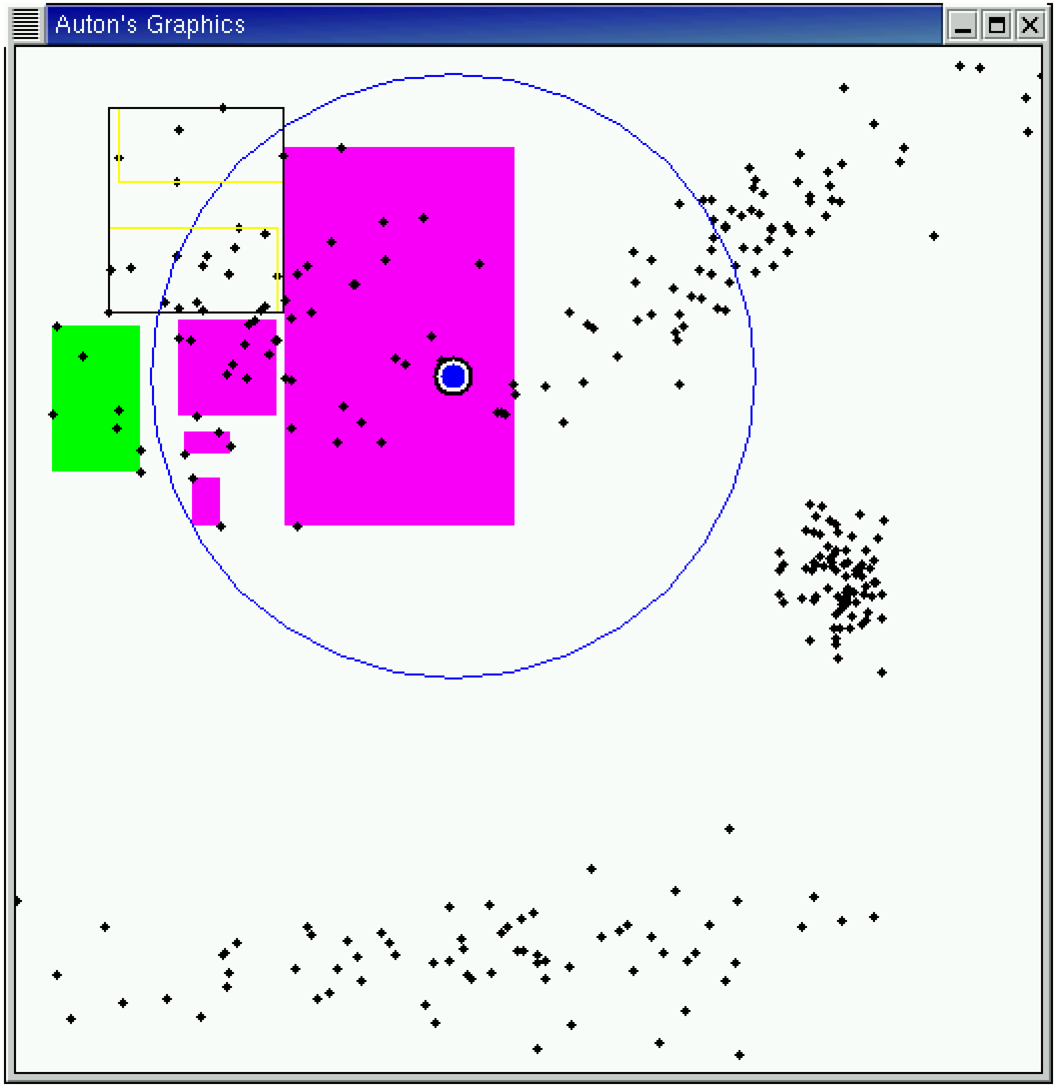
Range Count



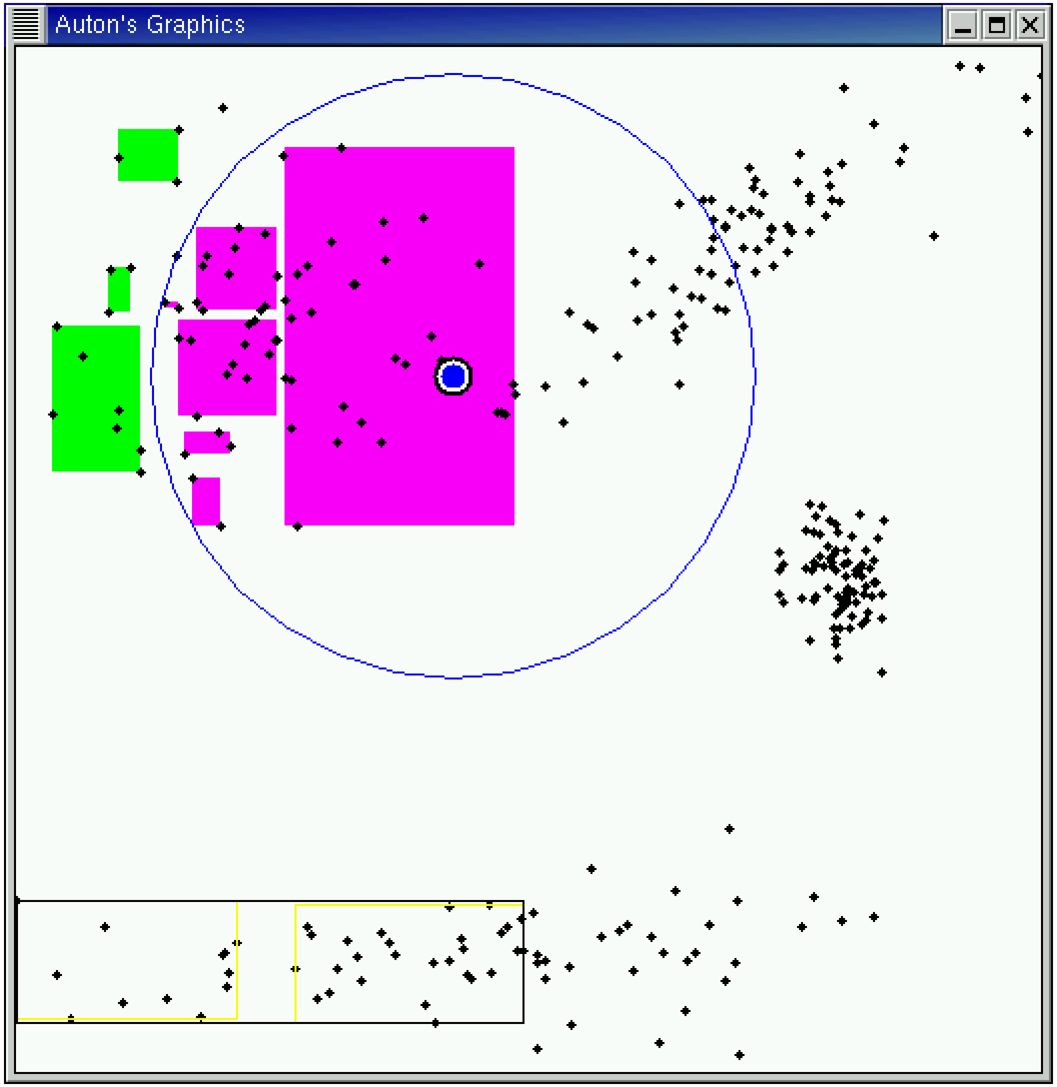
Range Count



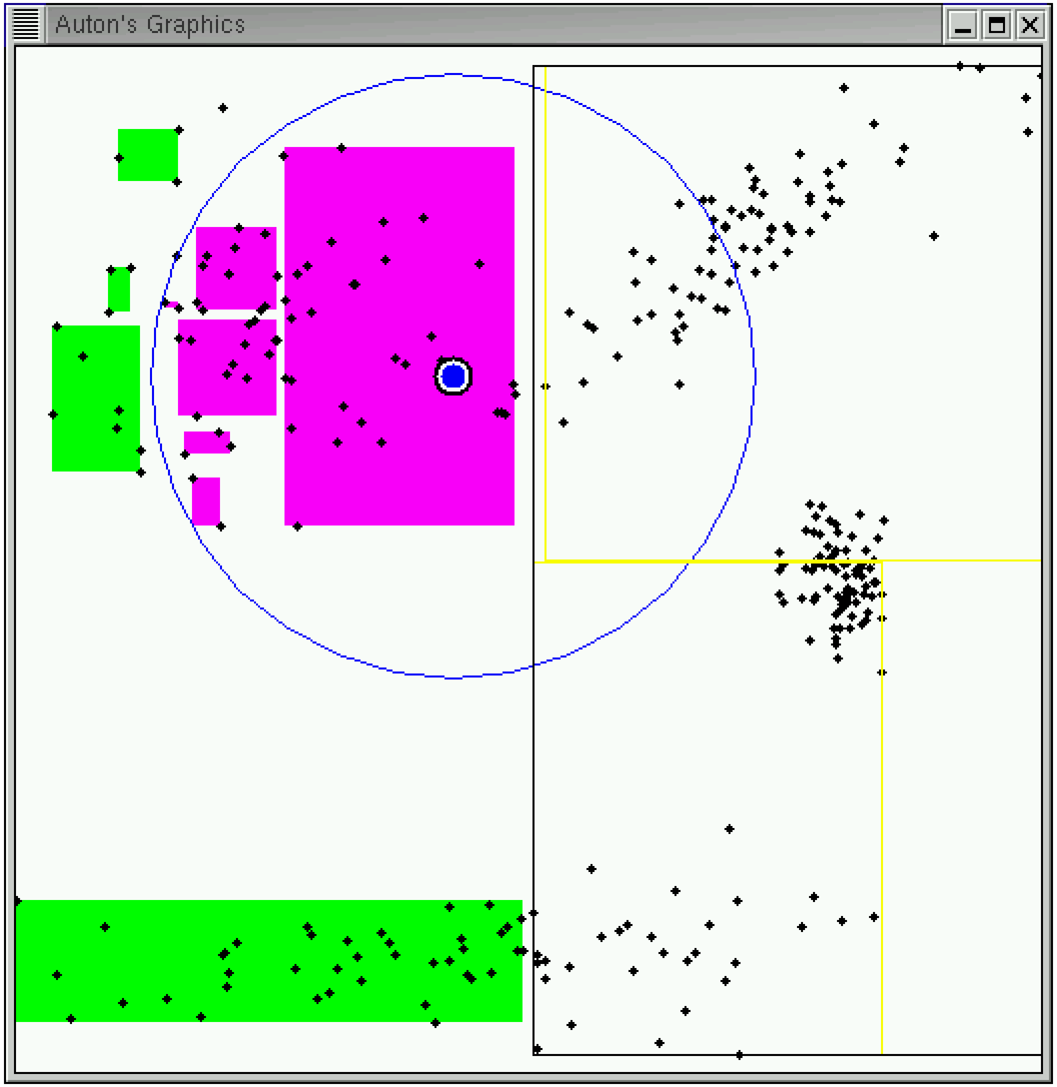
Range Count



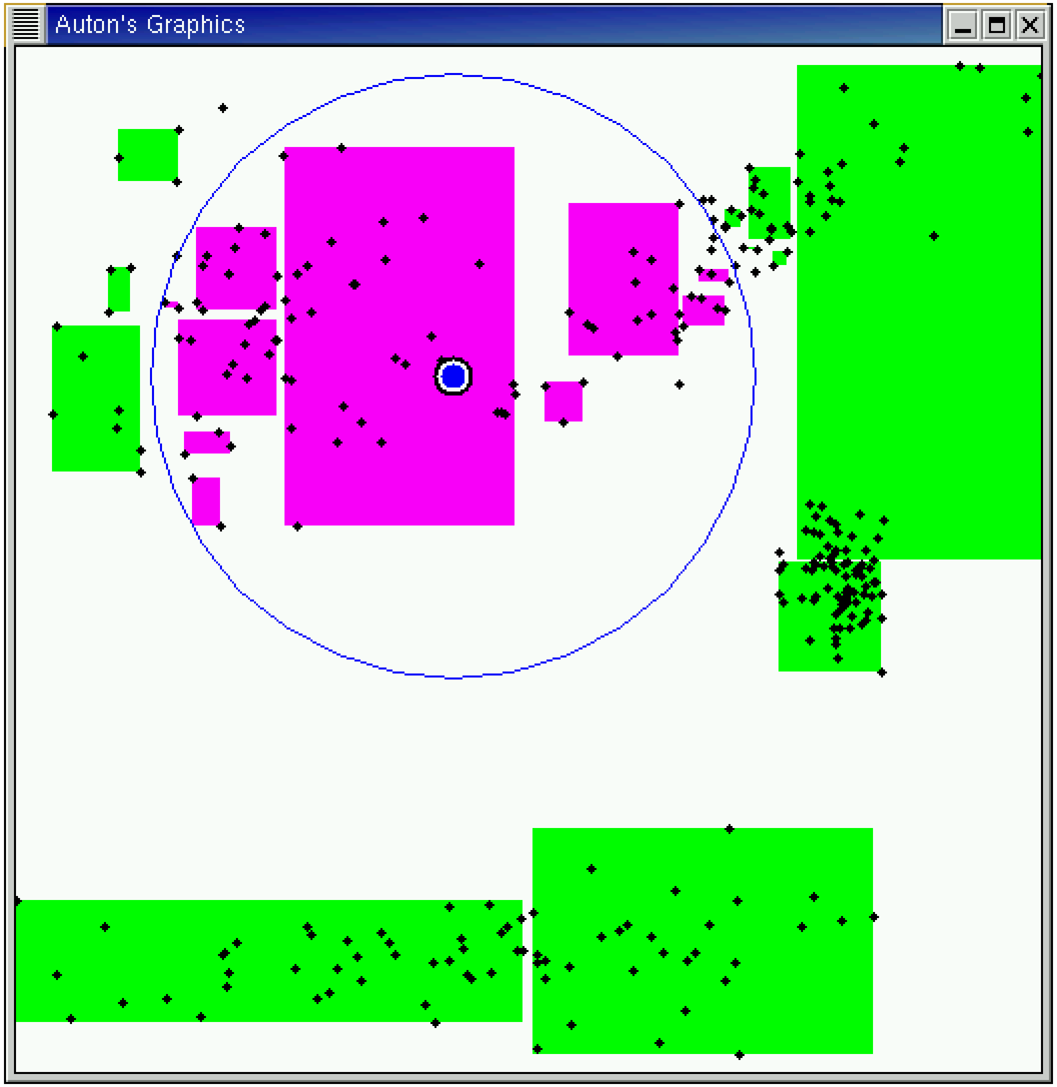
Range Count

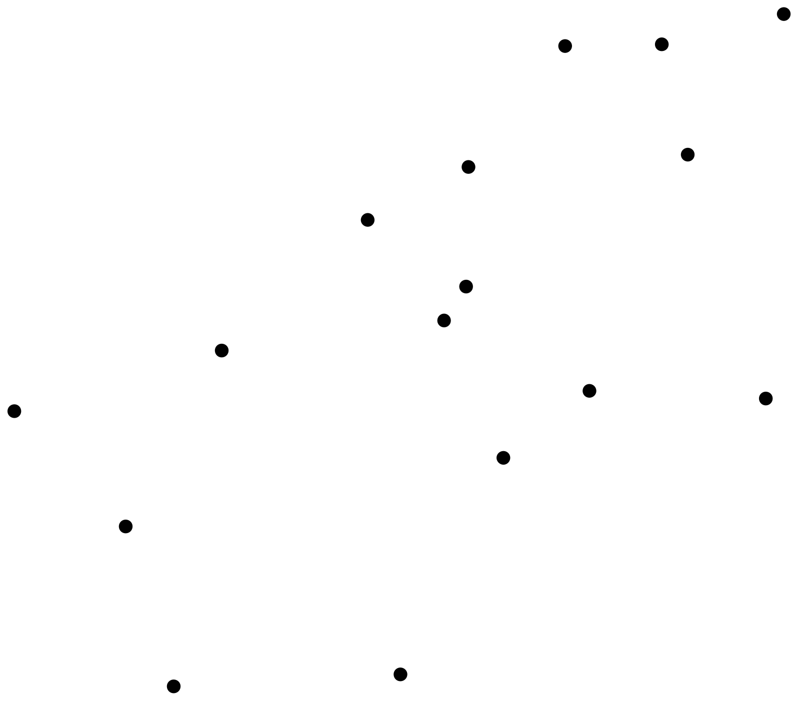


Range Count

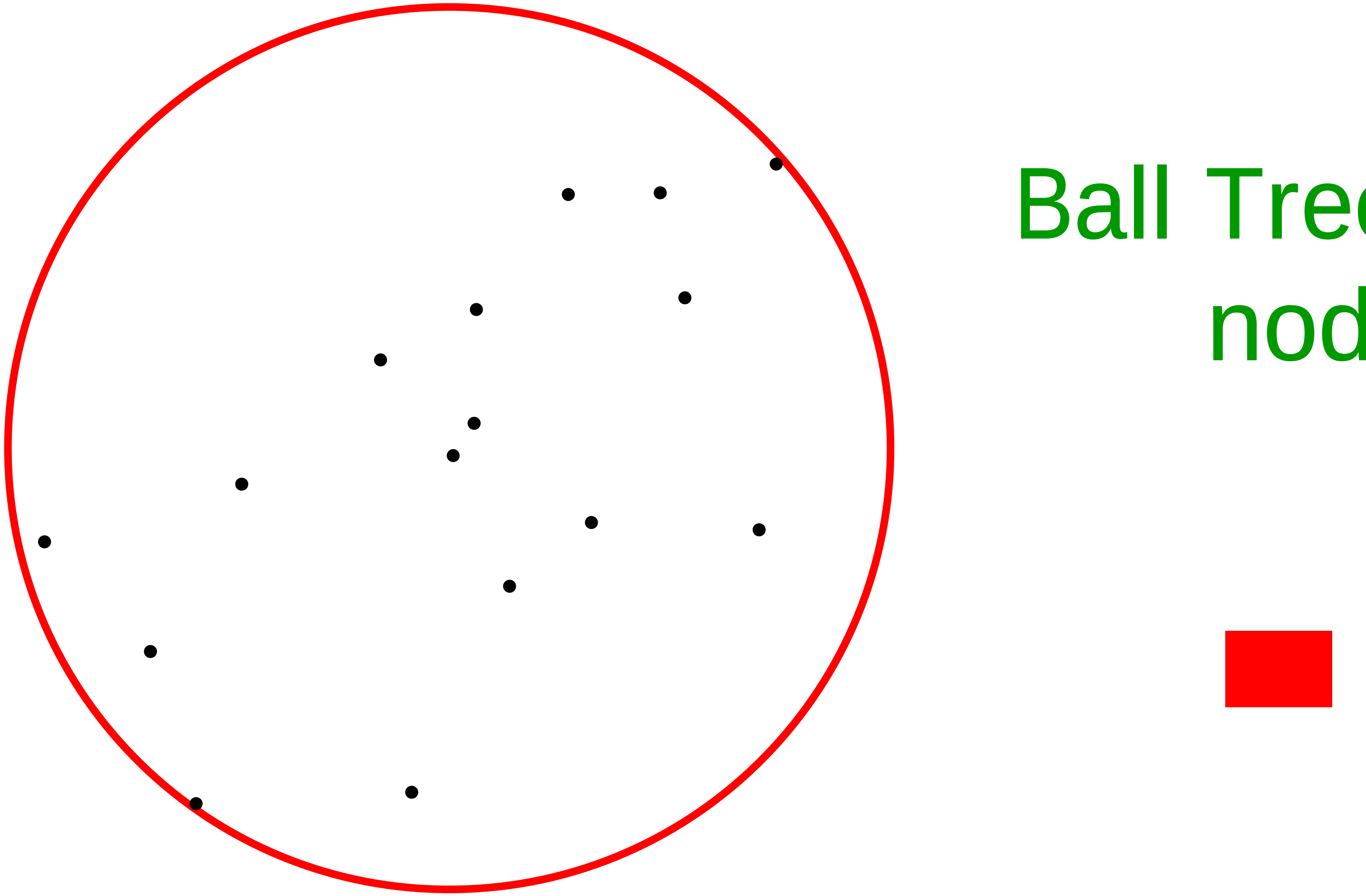


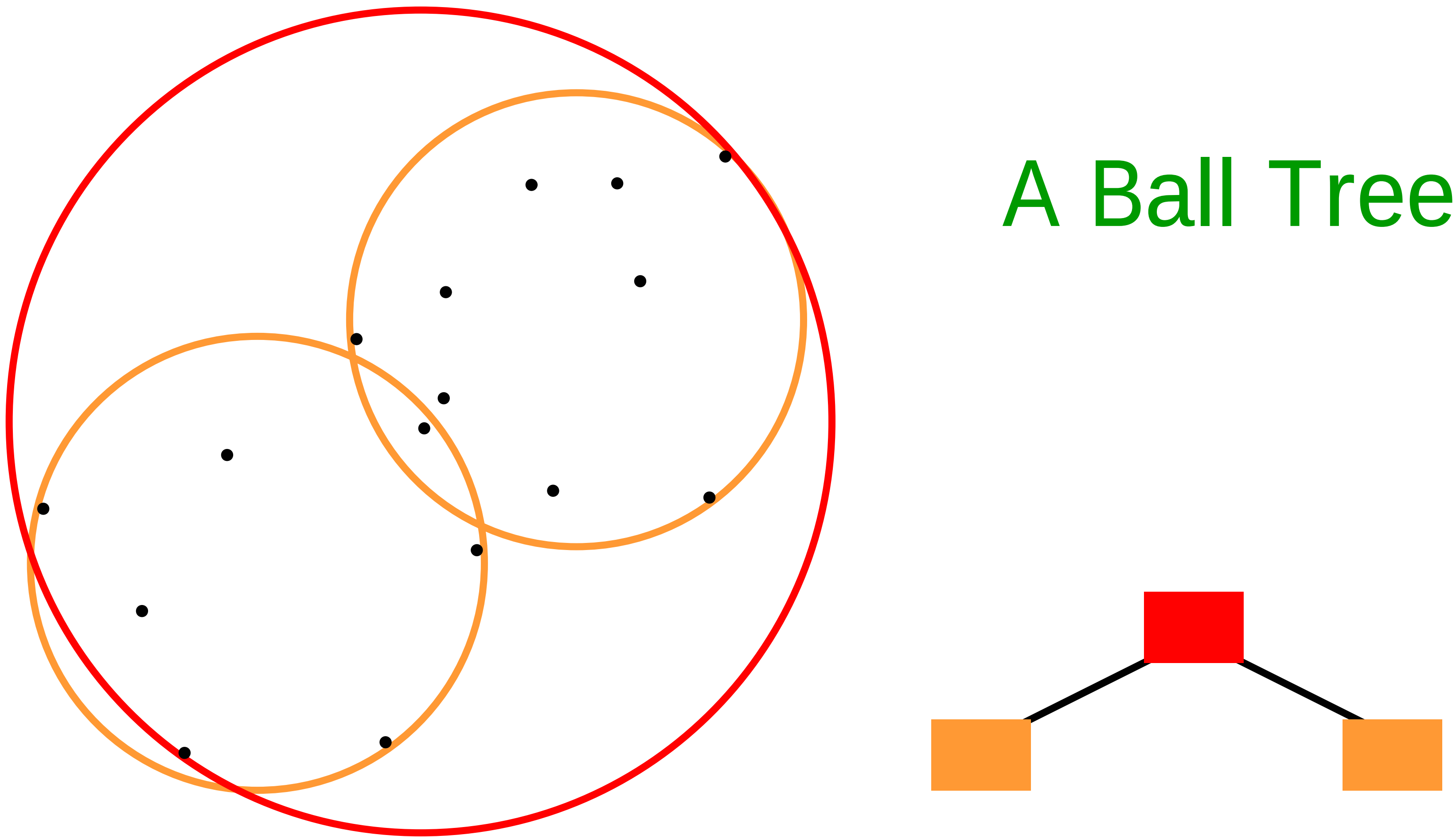
Range Count

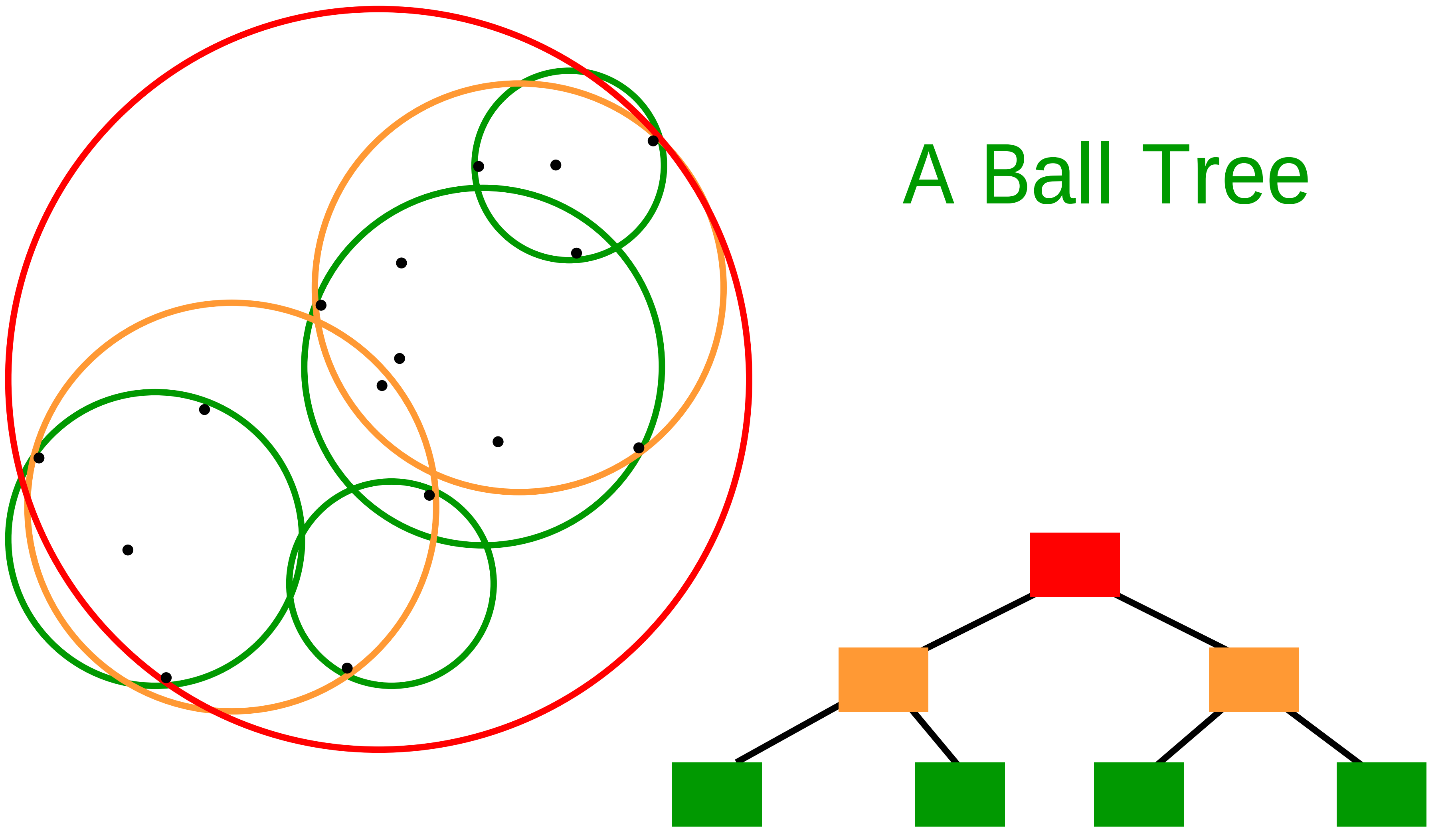


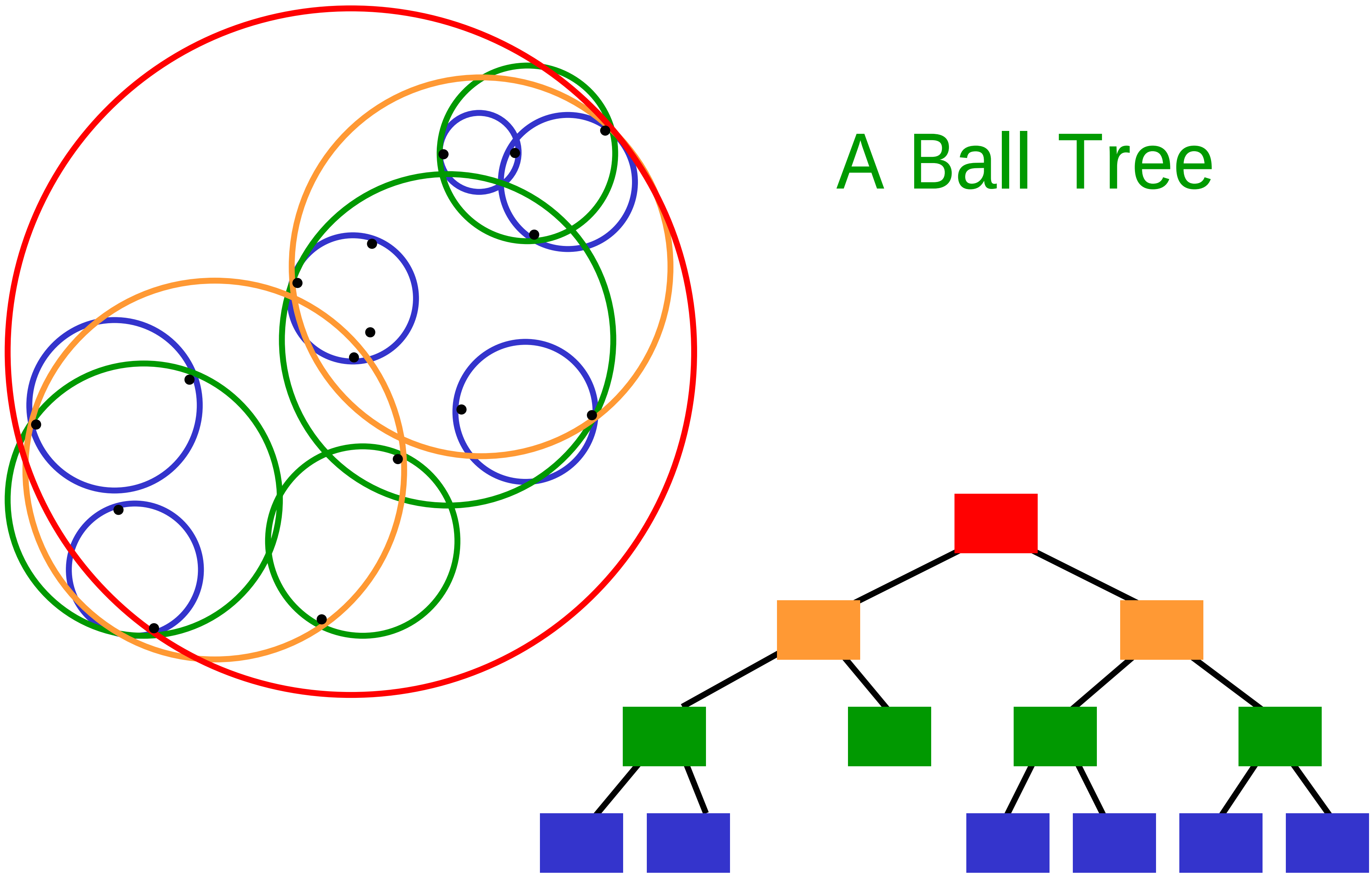


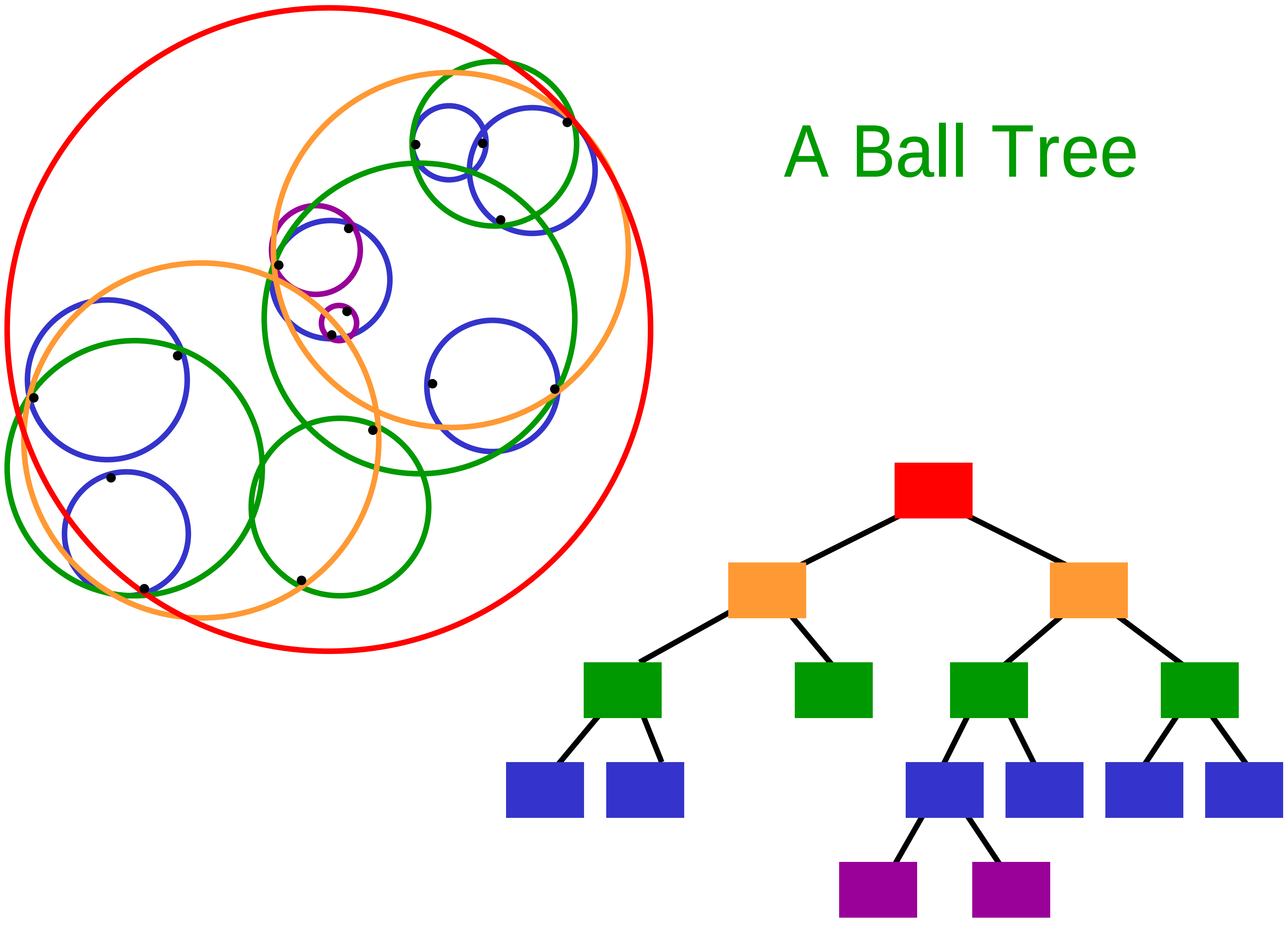
A Set of Points in a metric space

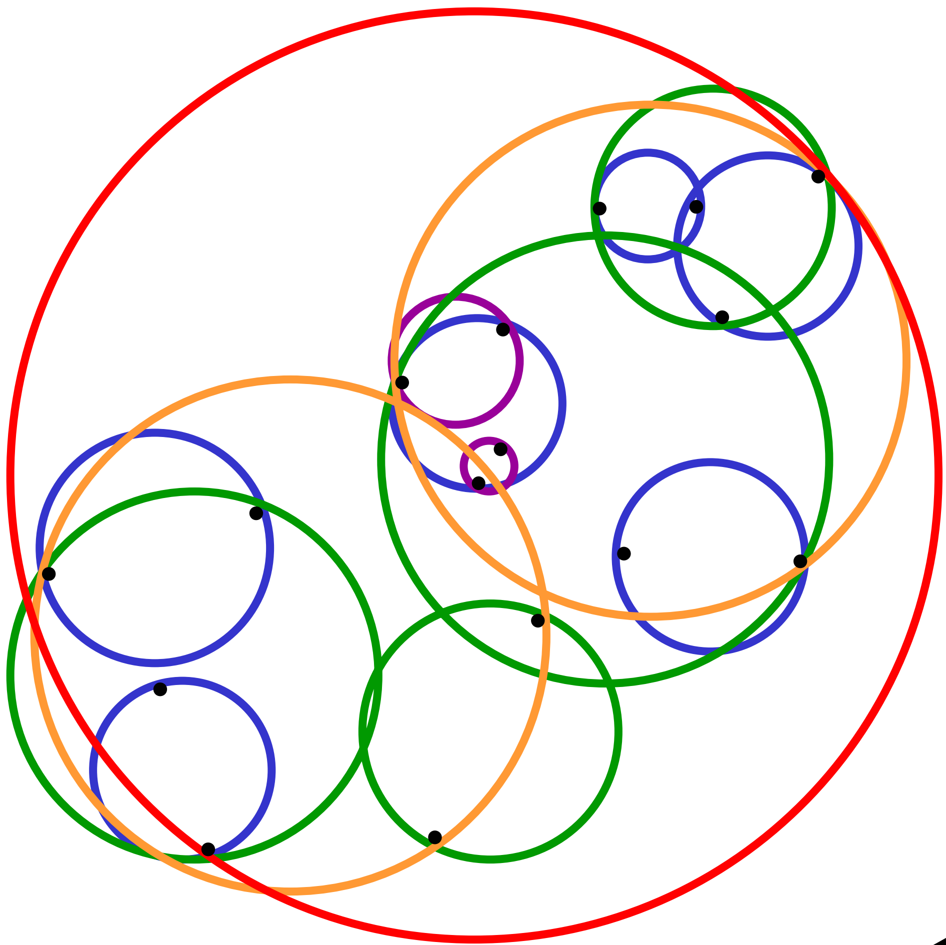

Ball Tree root
node


A Ball Tree

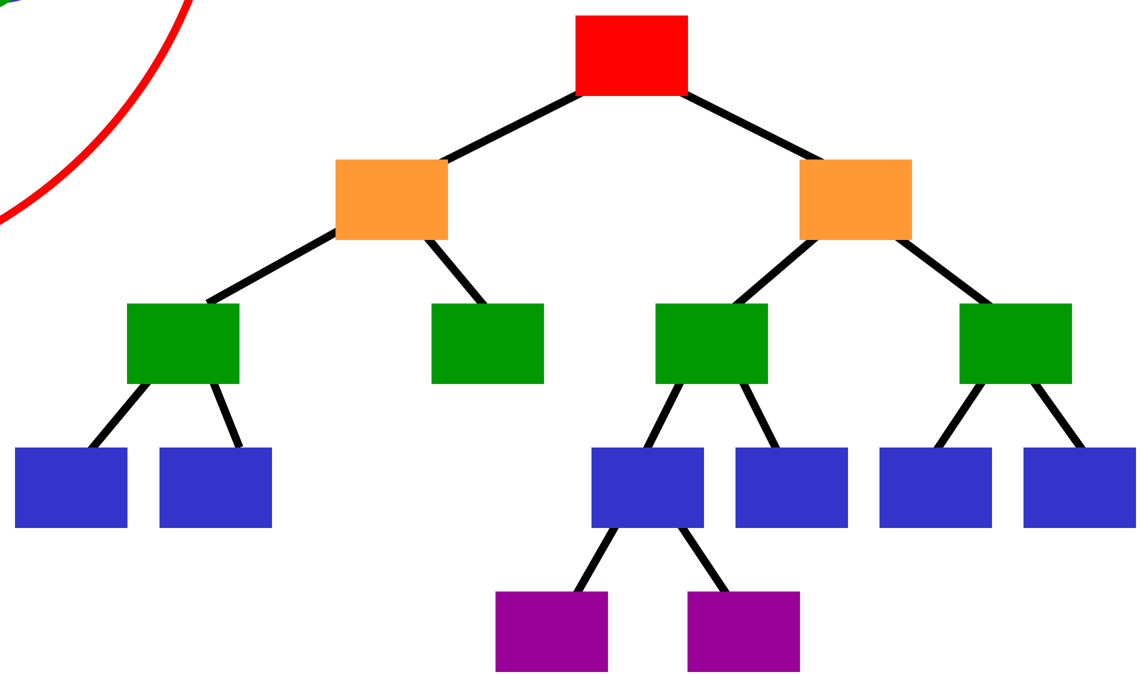

A Ball Tree


A Ball Tree


A Ball Tree



A Ball Tree



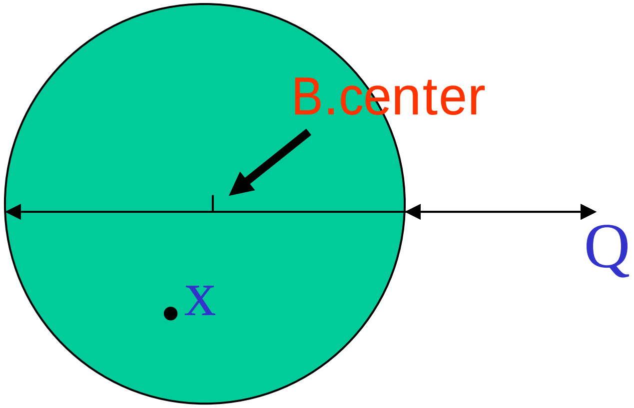
- J. Uhlmann, 1991
- S. Omohundro, NIPS 1991

Ball-trees: properties

Let *Q* be any query point and let *x* be a point inside ball *B*

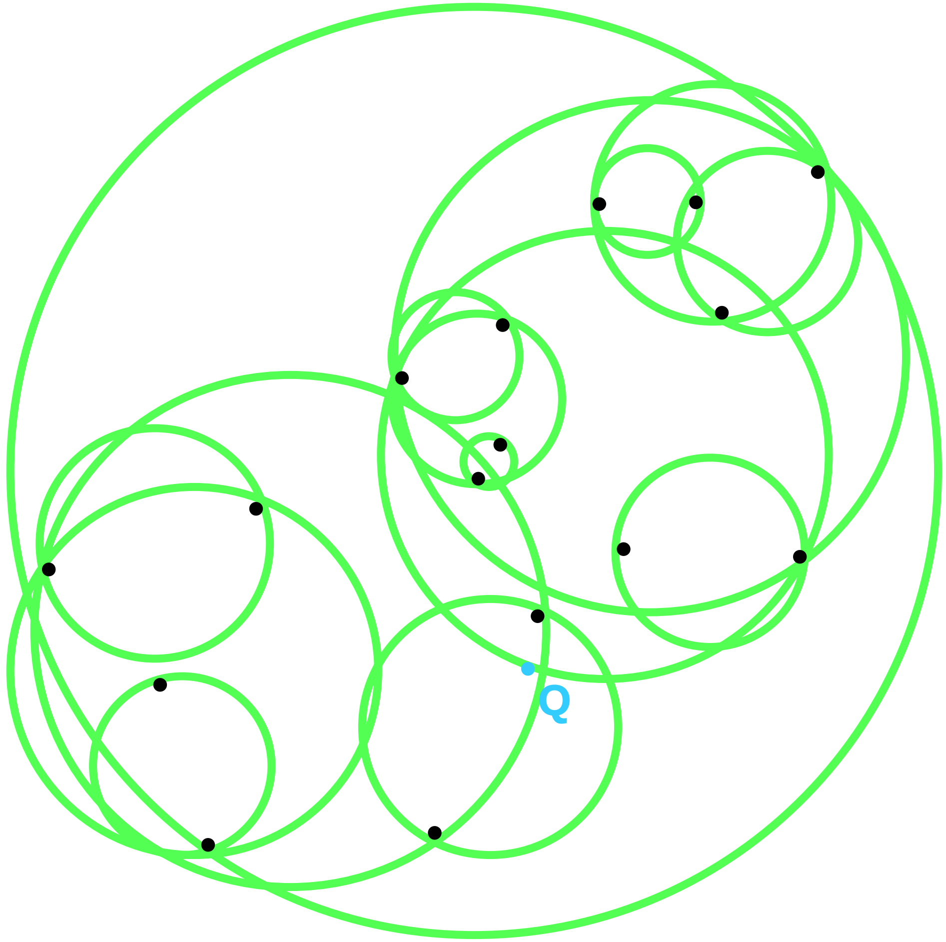
$|x - \boldsymbol{Q}| \geq |\boldsymbol{Q} - B.center| - B.radius$

$|x - \boldsymbol{Q}| \leq |\boldsymbol{Q} - B.center| + B.radius$



Outline

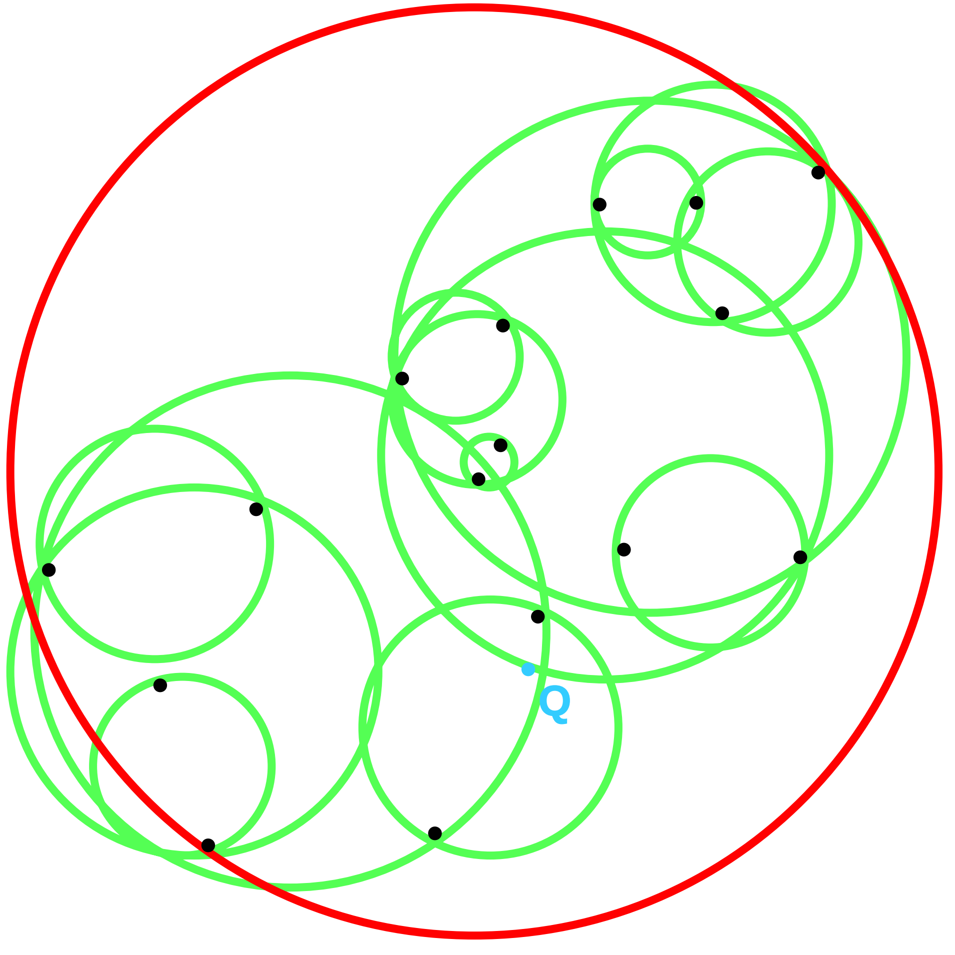
- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- ► K-nearest neighbor with ball trees
  - Very fast non-parametric classification
  - skewed binary outputs
  - General binary outputs
  - multi-classed outputs
- Very fast kernel-based statistics
  - n-point computations
  - clustering
  - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

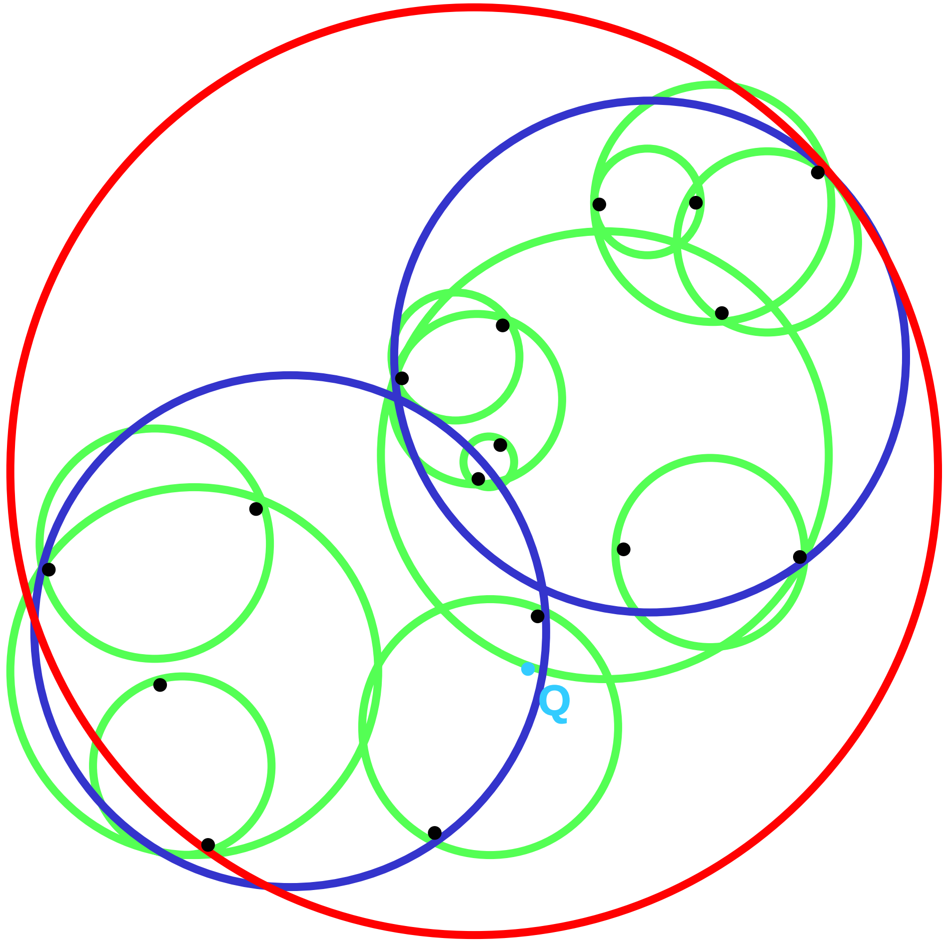


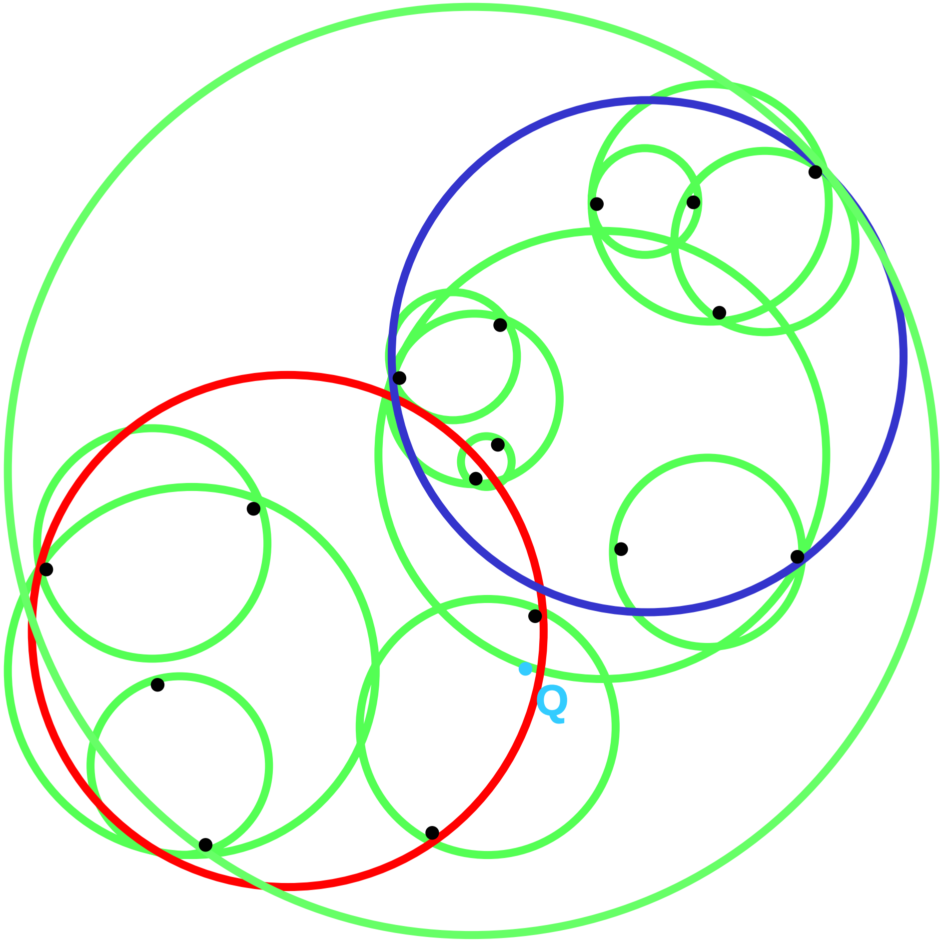
Goal: Find out the 2-nearest neighbors of Q.

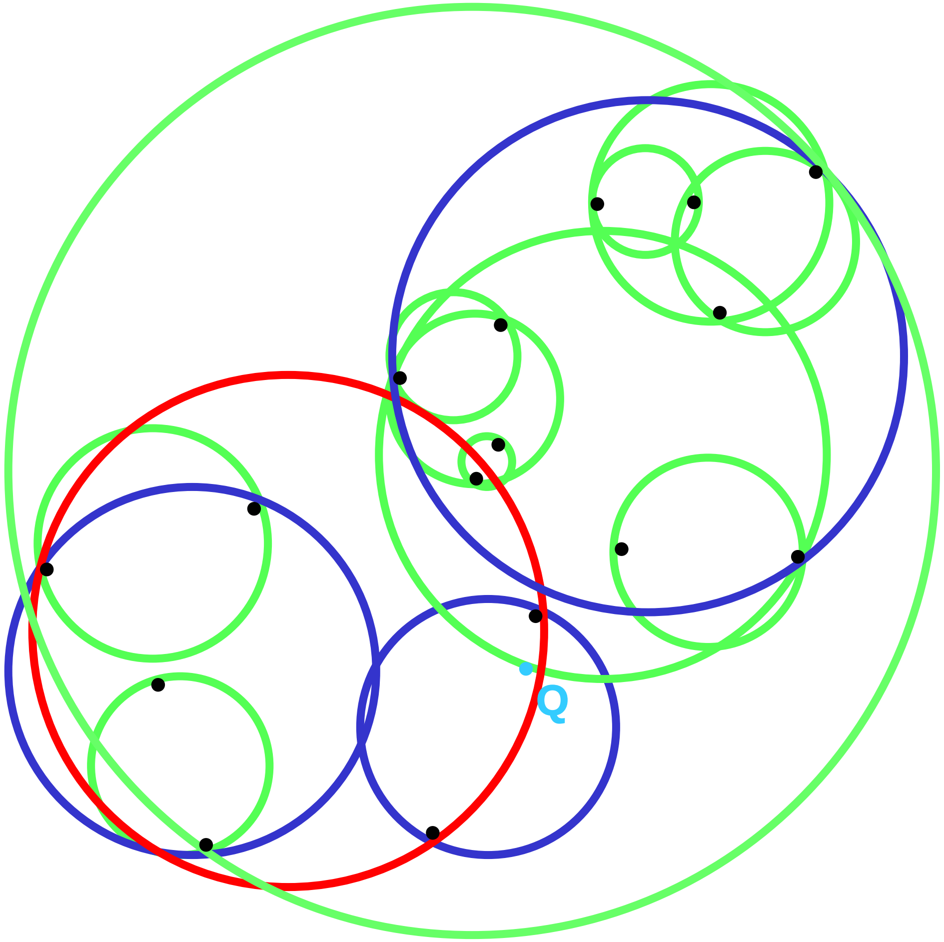
•J. Uhlmann, 1991

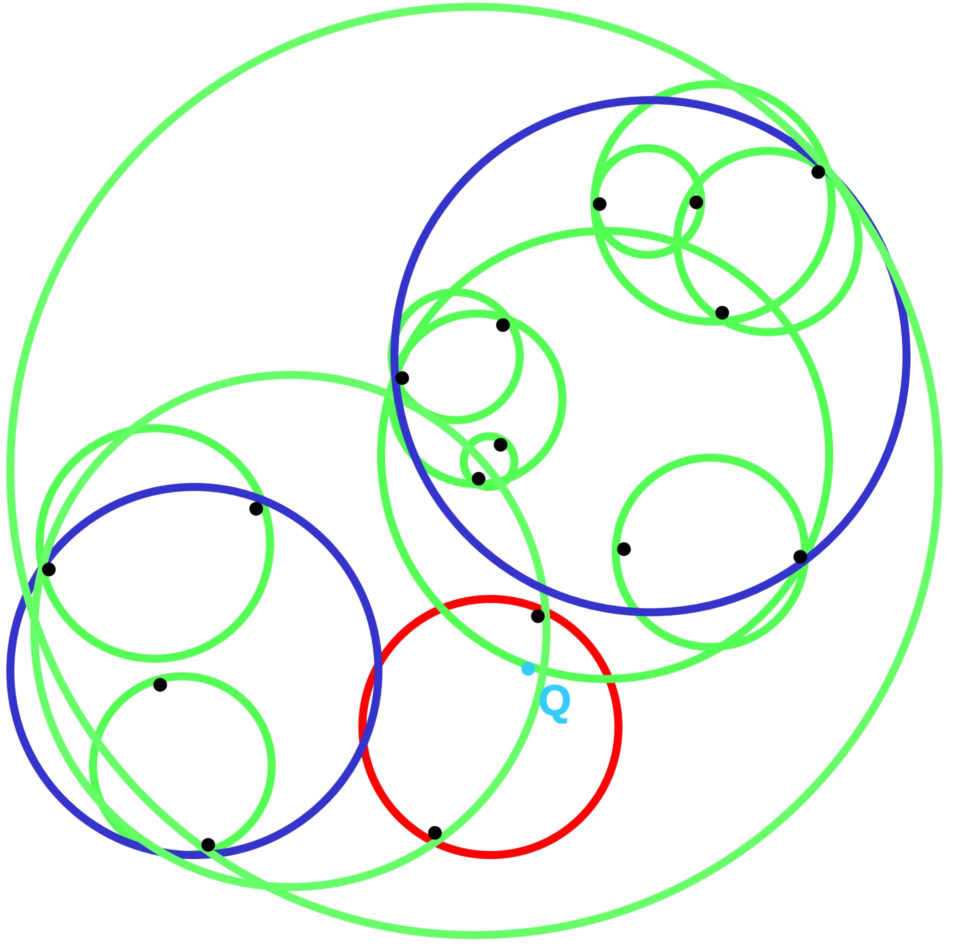
•S. Omohundro, NIPS 1991

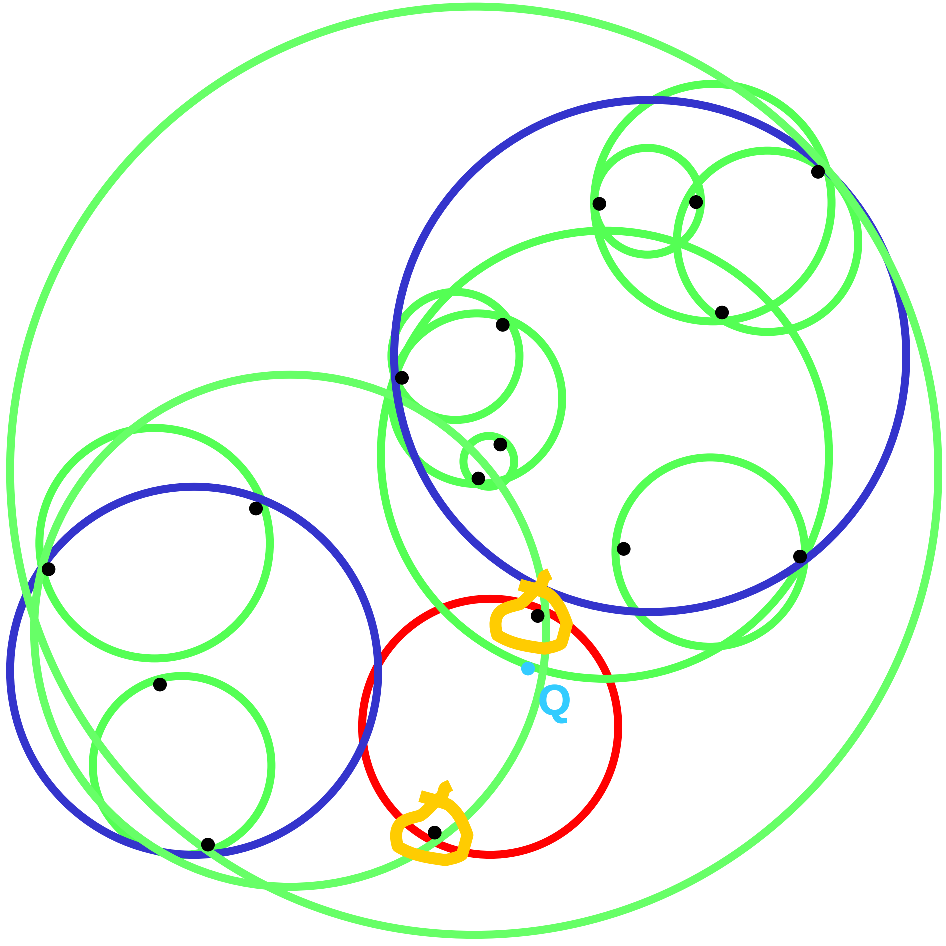

Q


Q

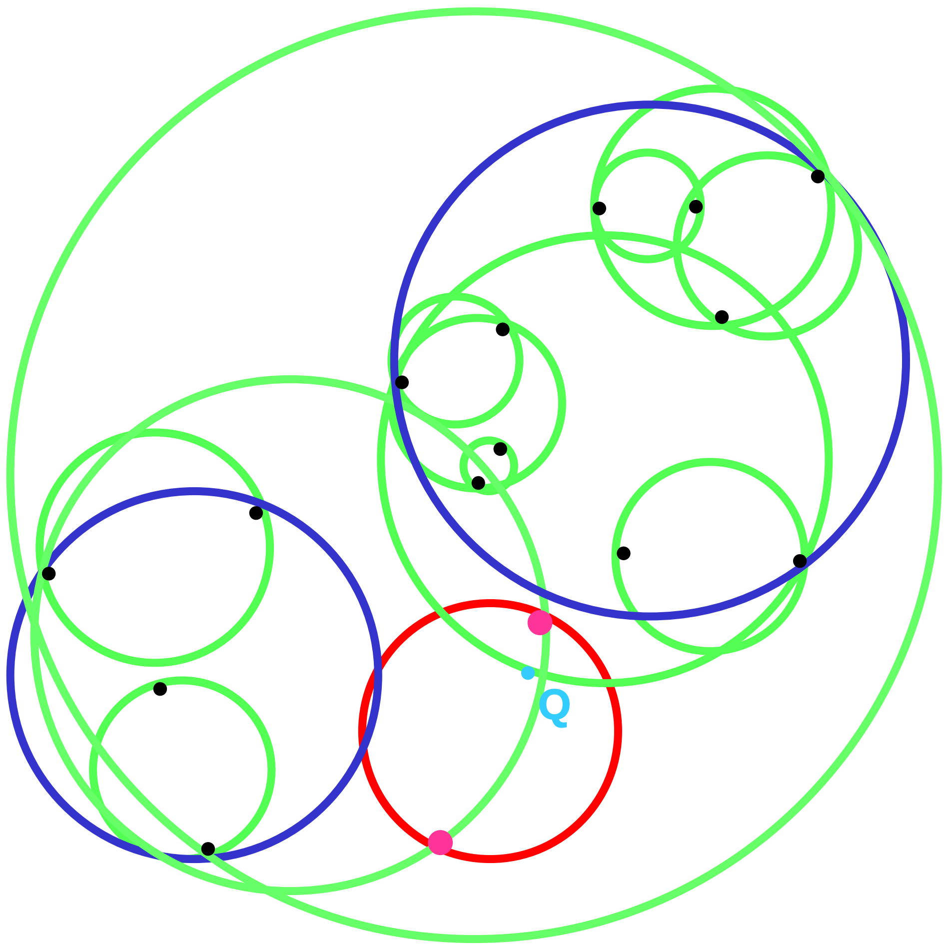

Q


Q


Q

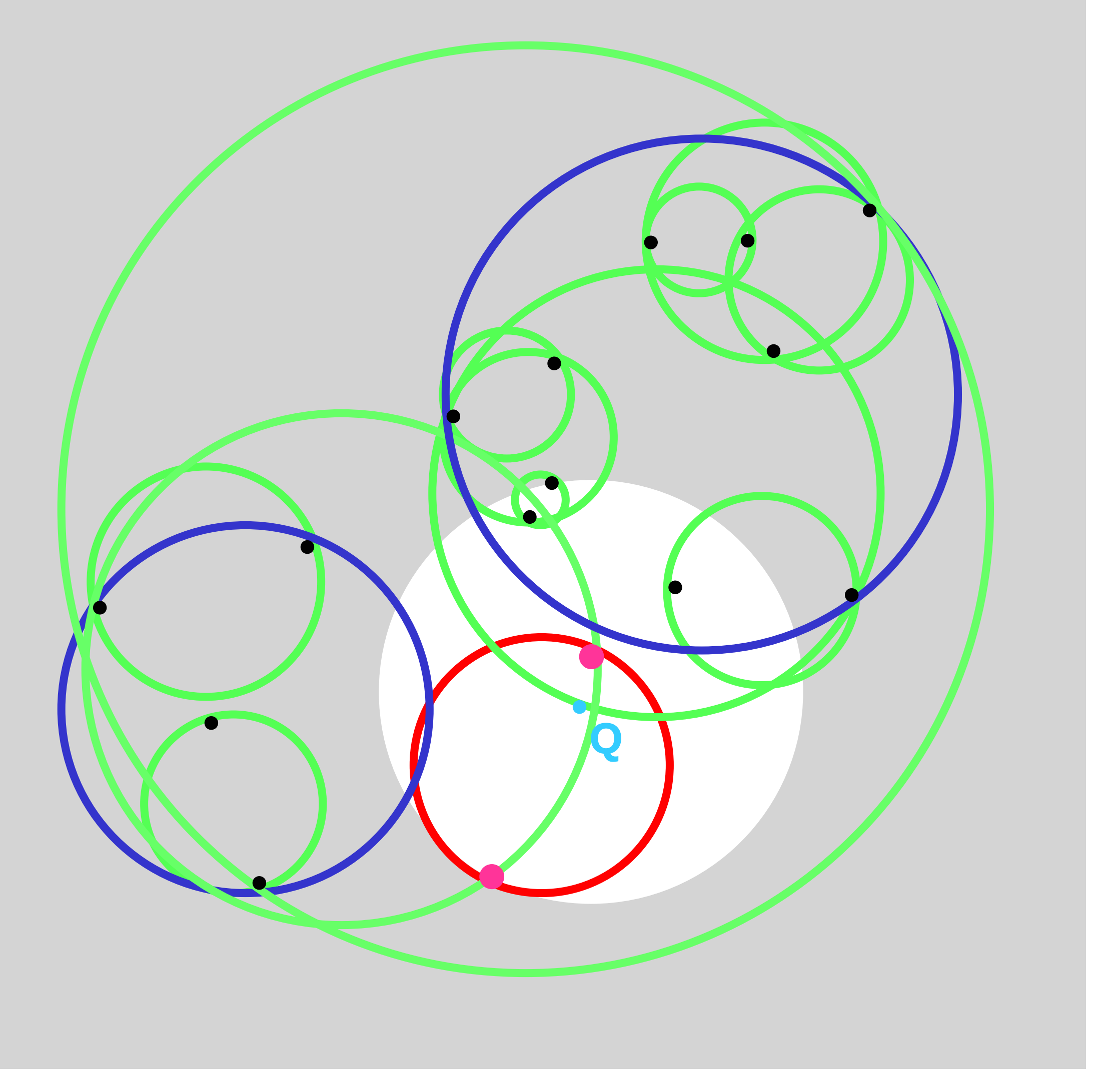


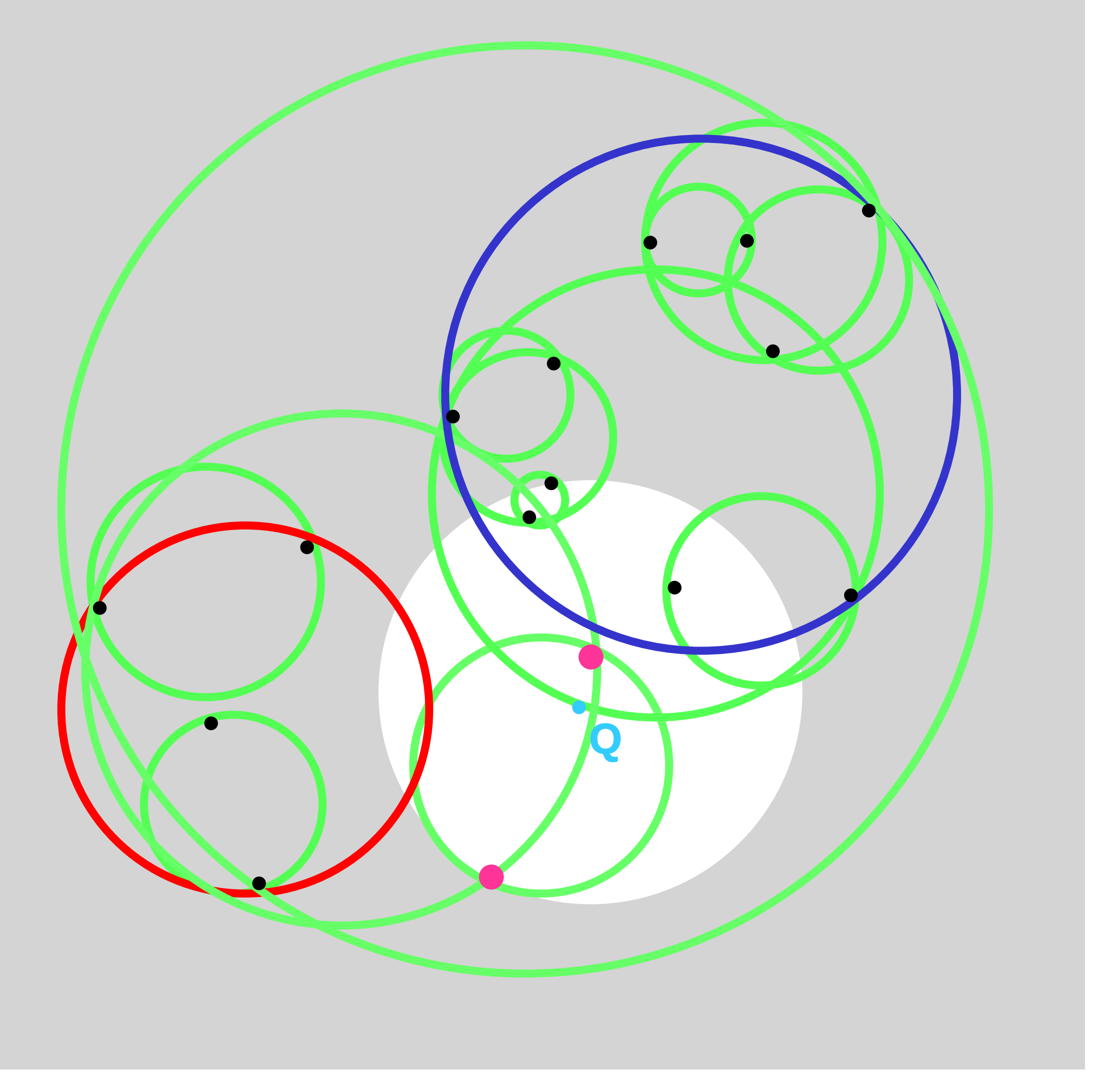
We've hit a leaf node, so we explicitly look at the points in the node

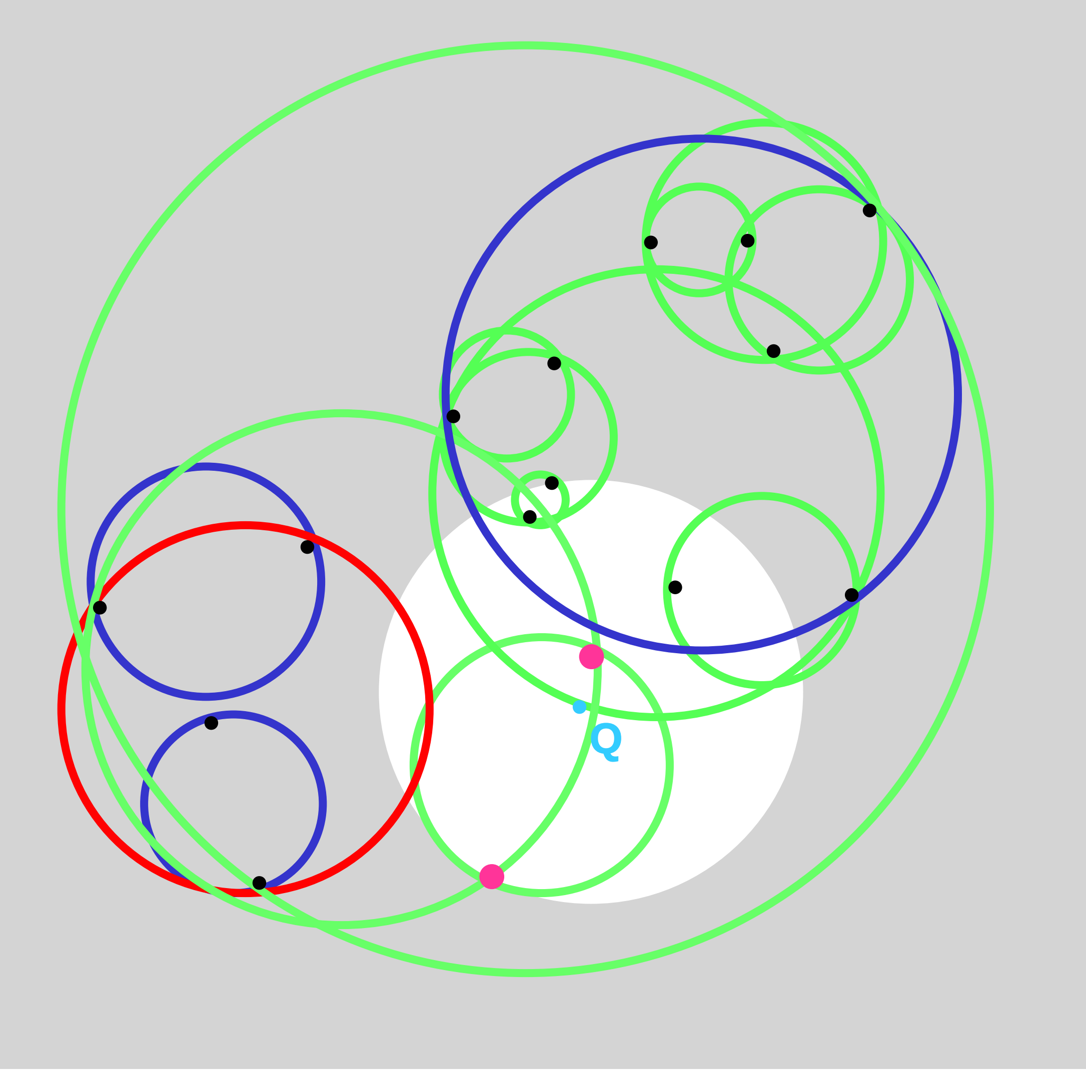


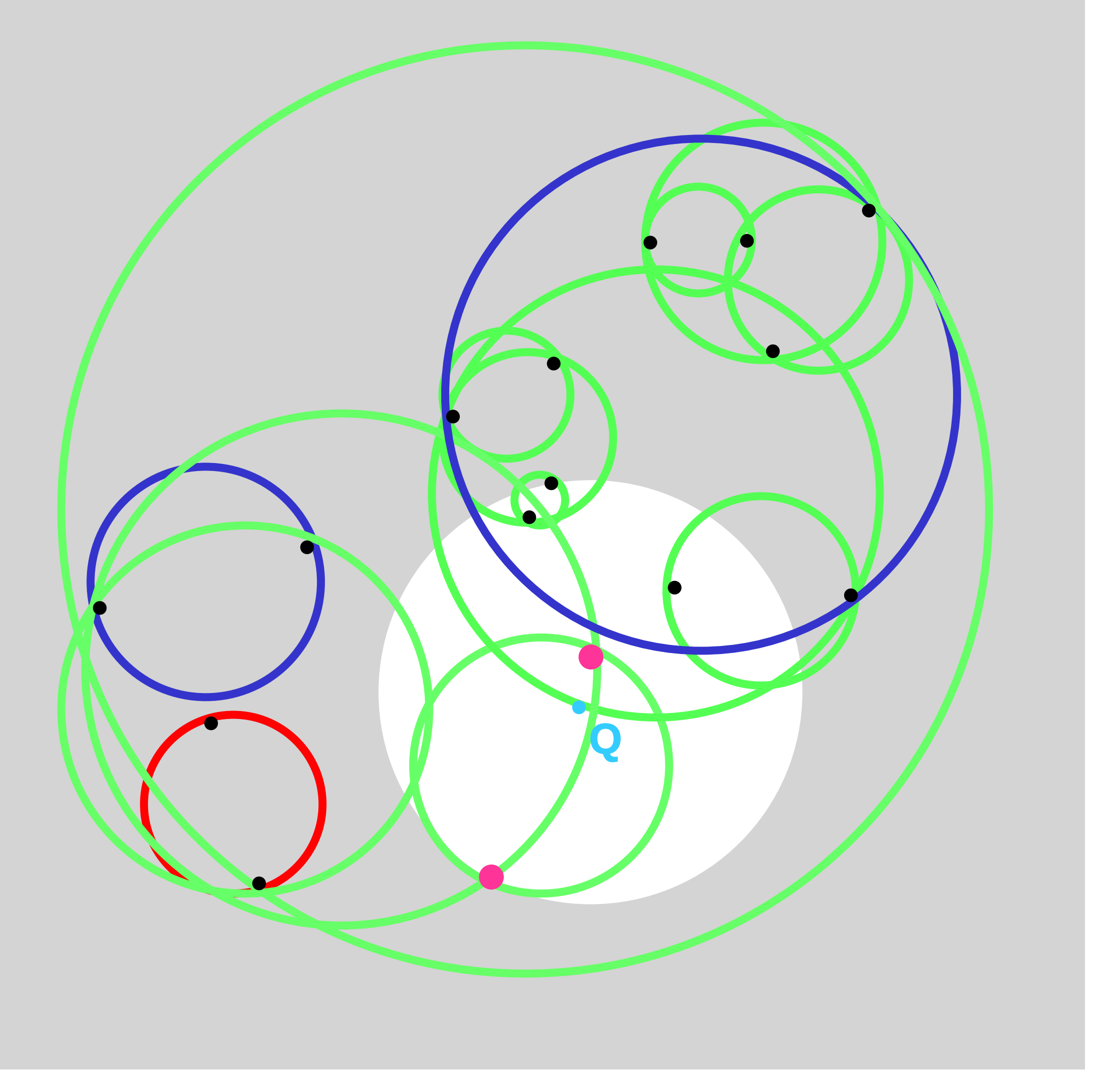
Two nearest neighbors found so far are in pink

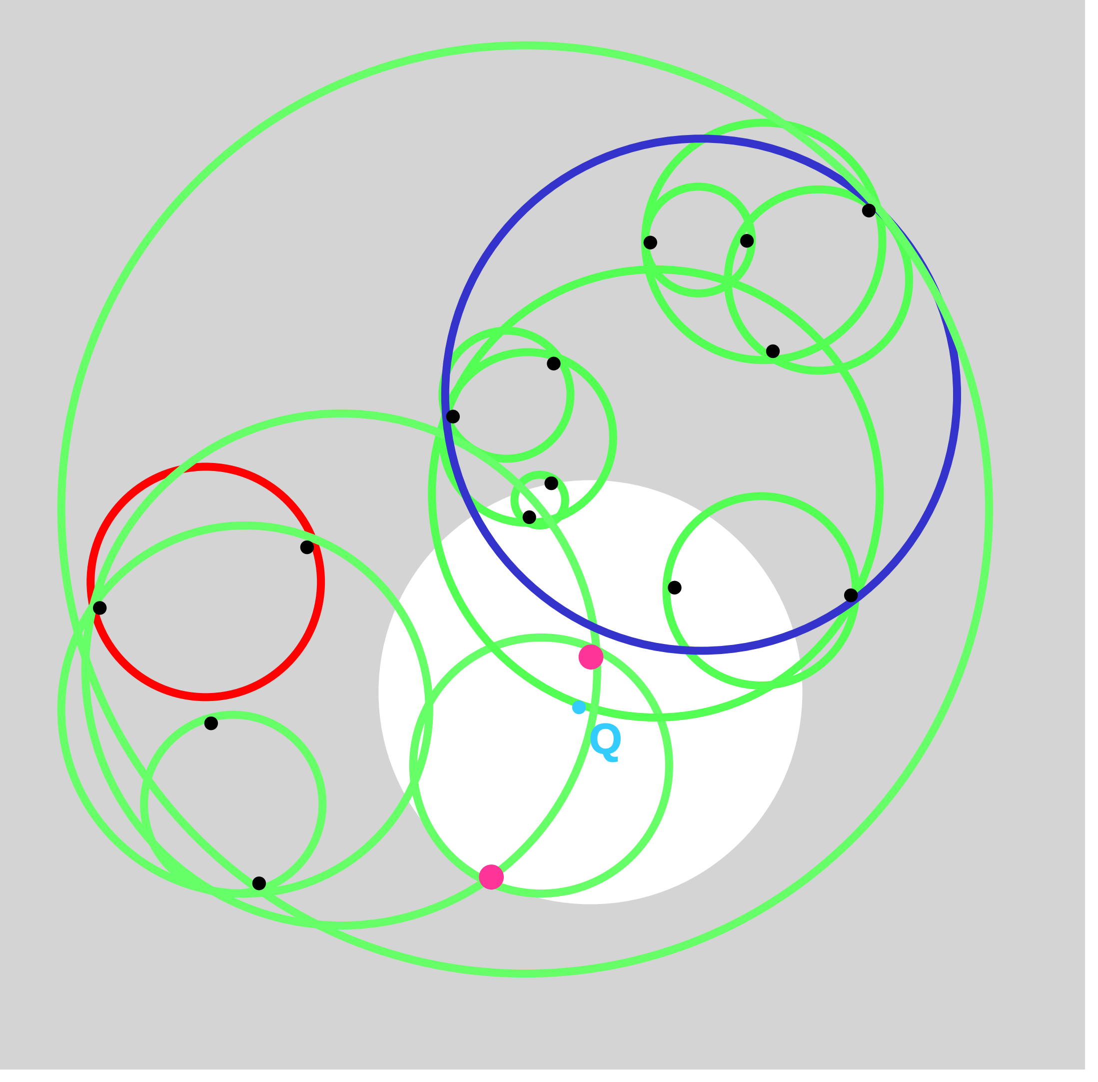
(remember we have yet to search the blue balls)

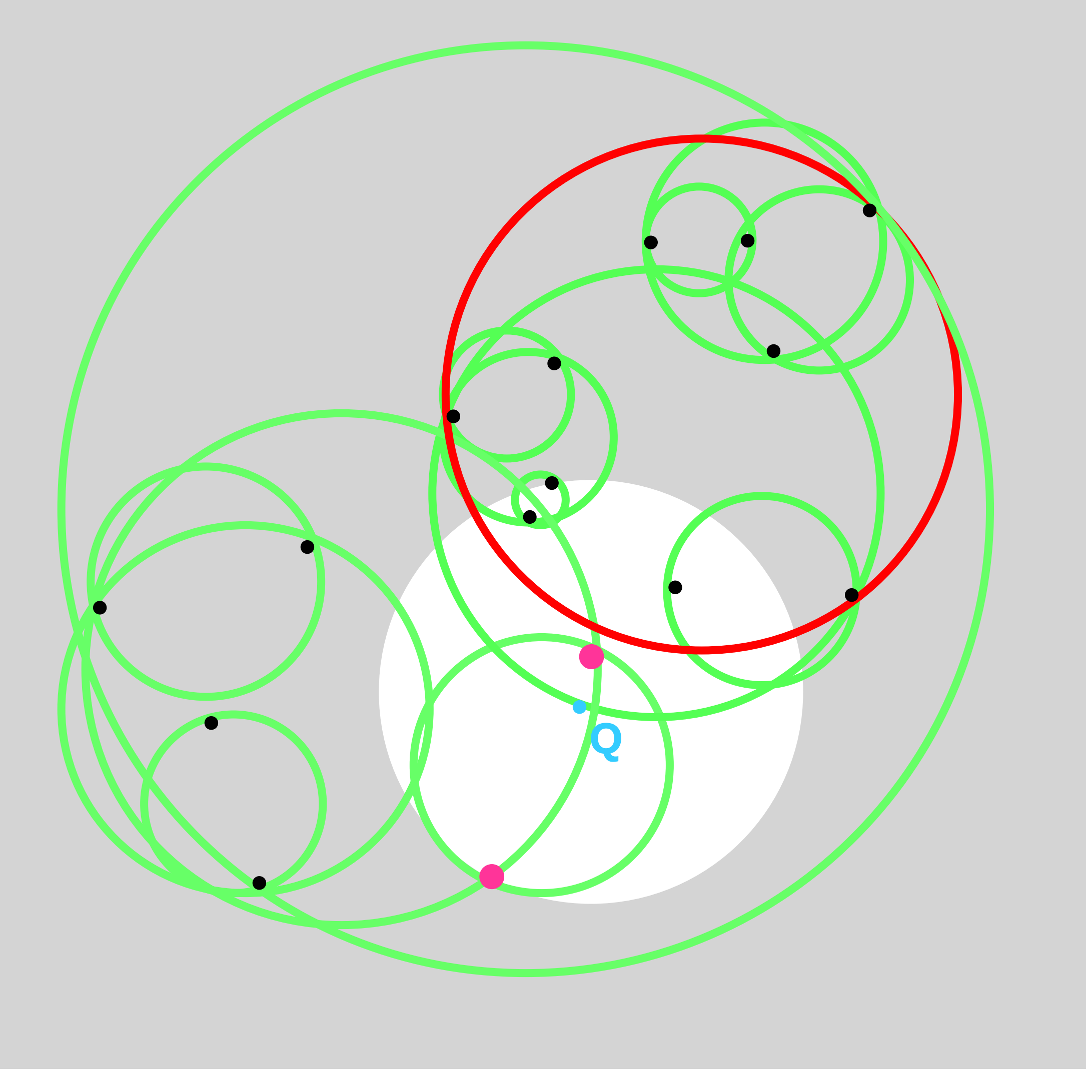

Q

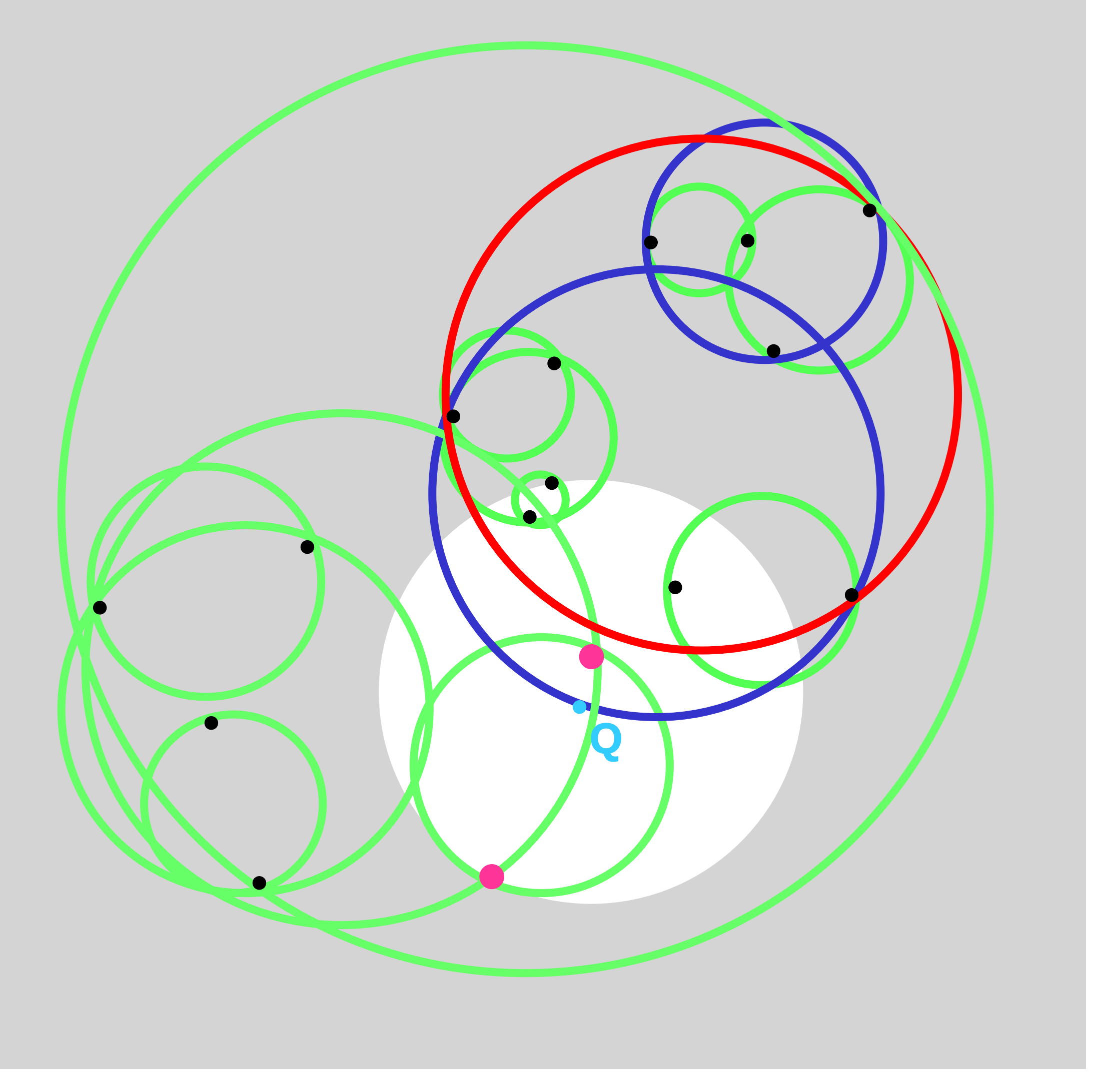

Q

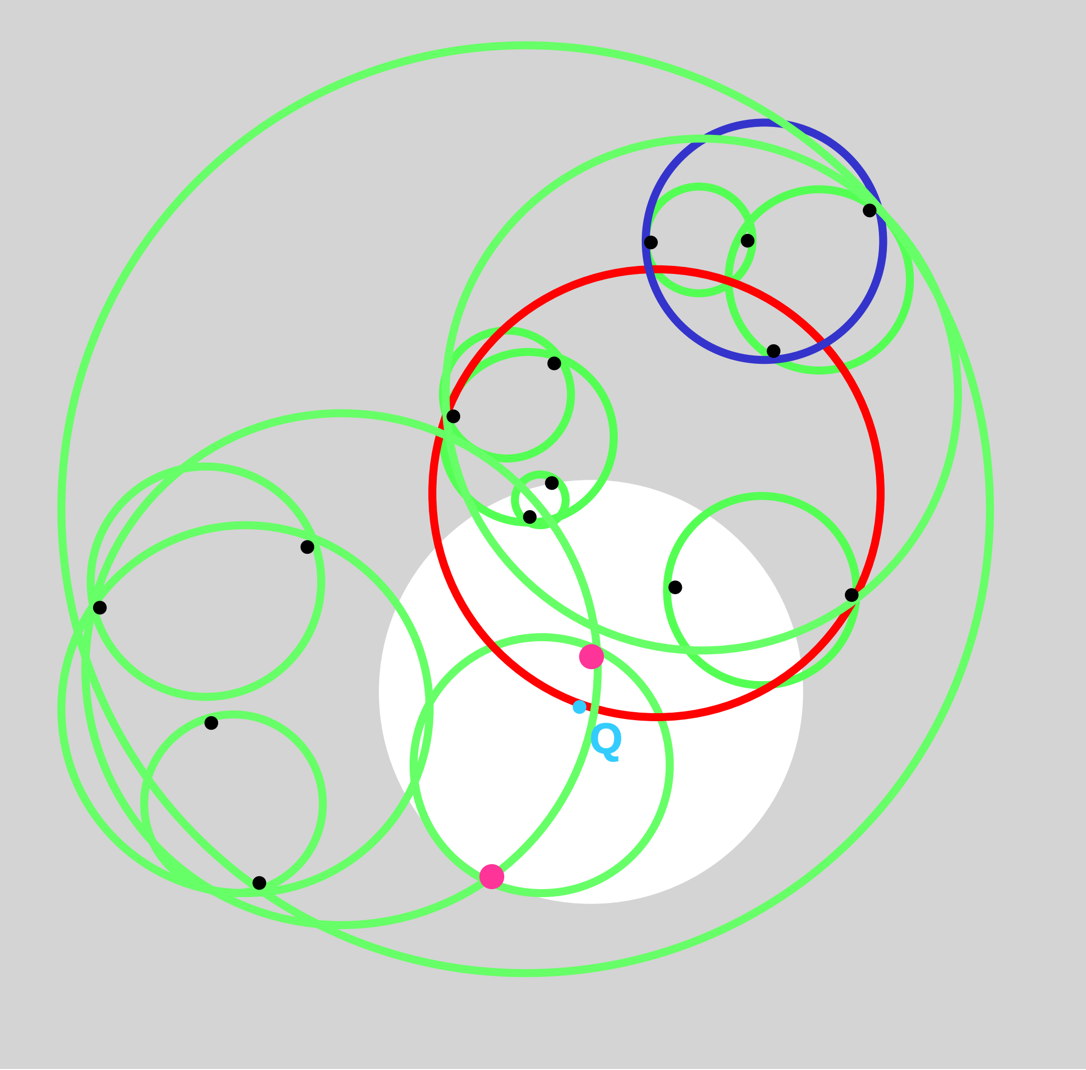

Q

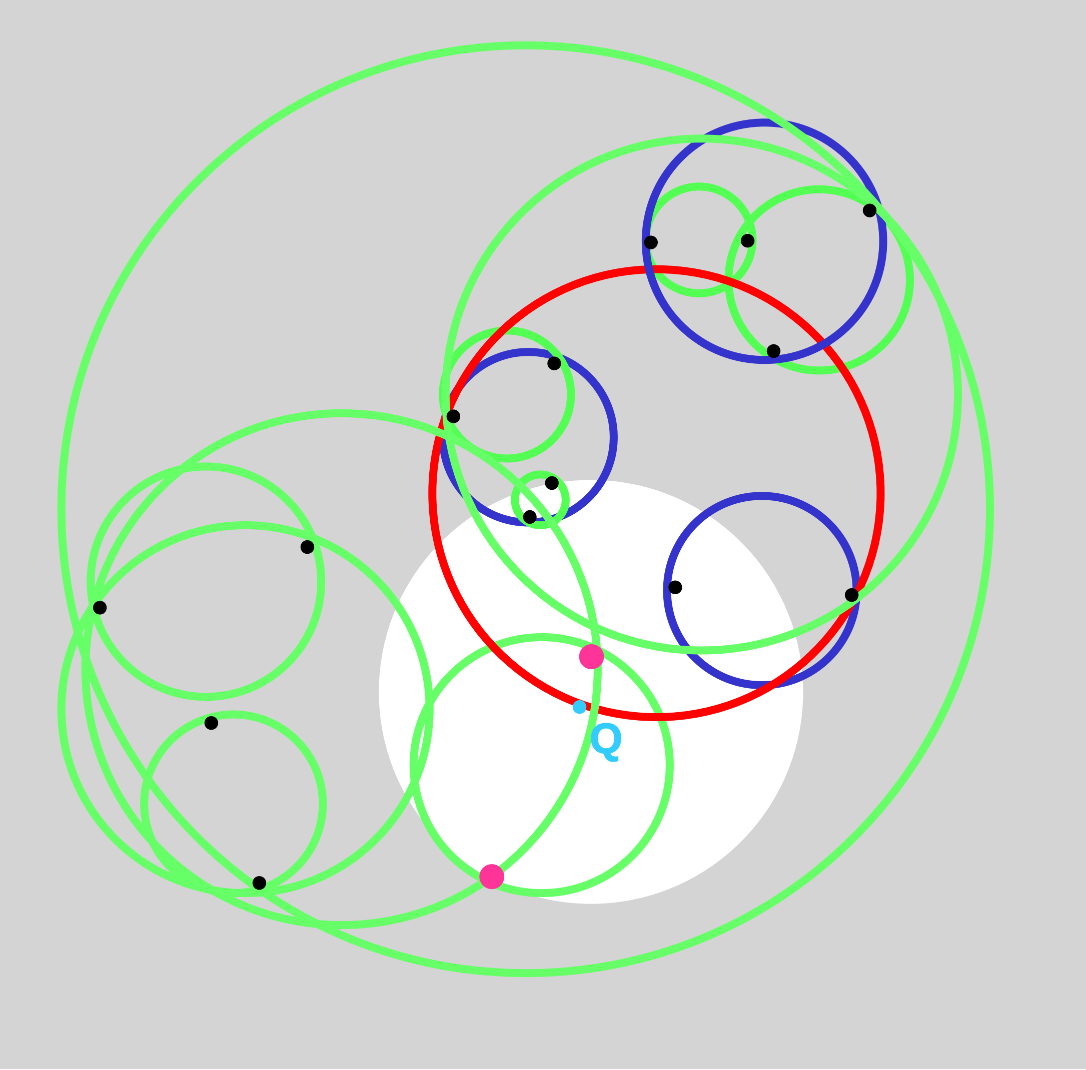

Q

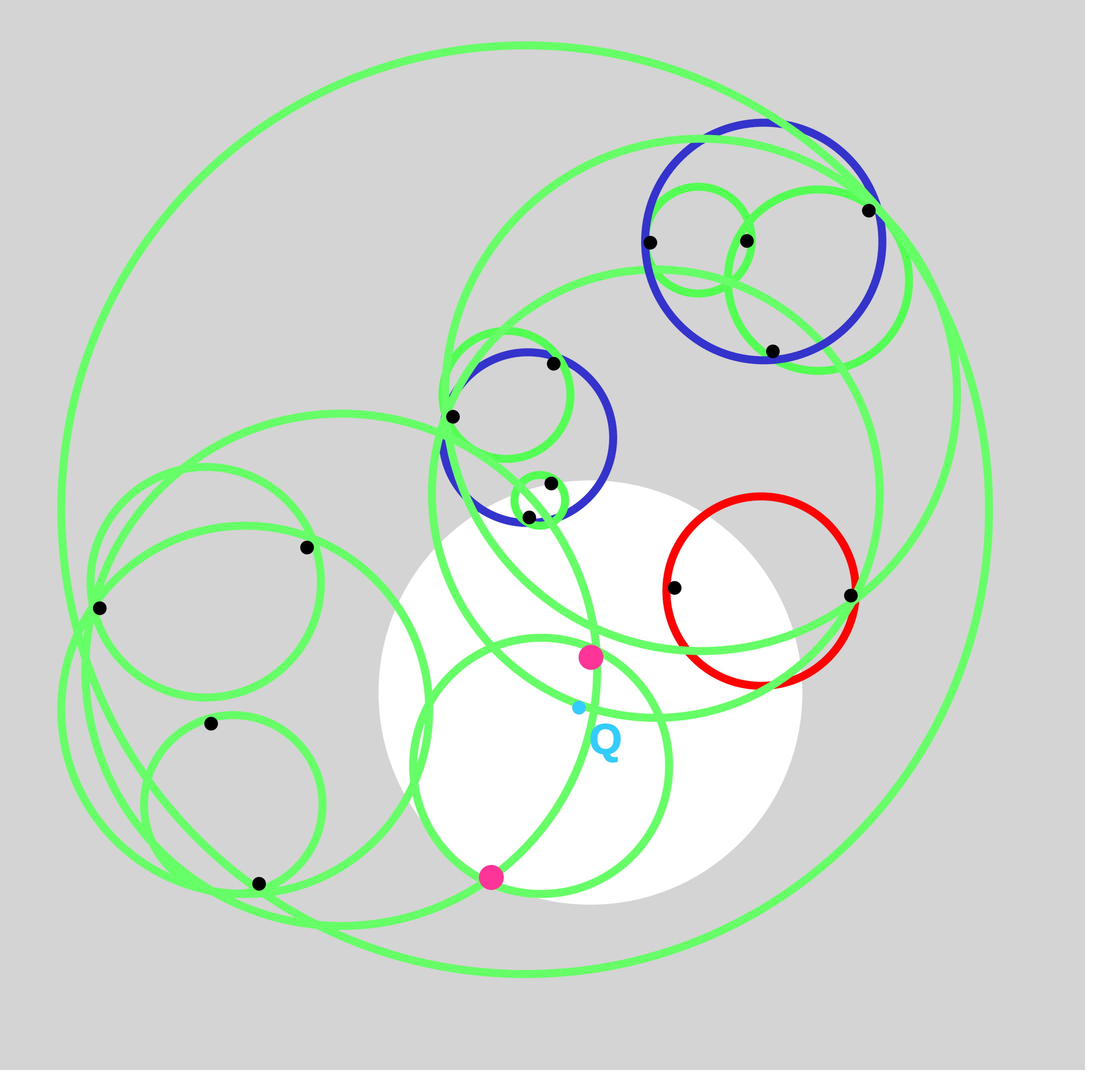

Q

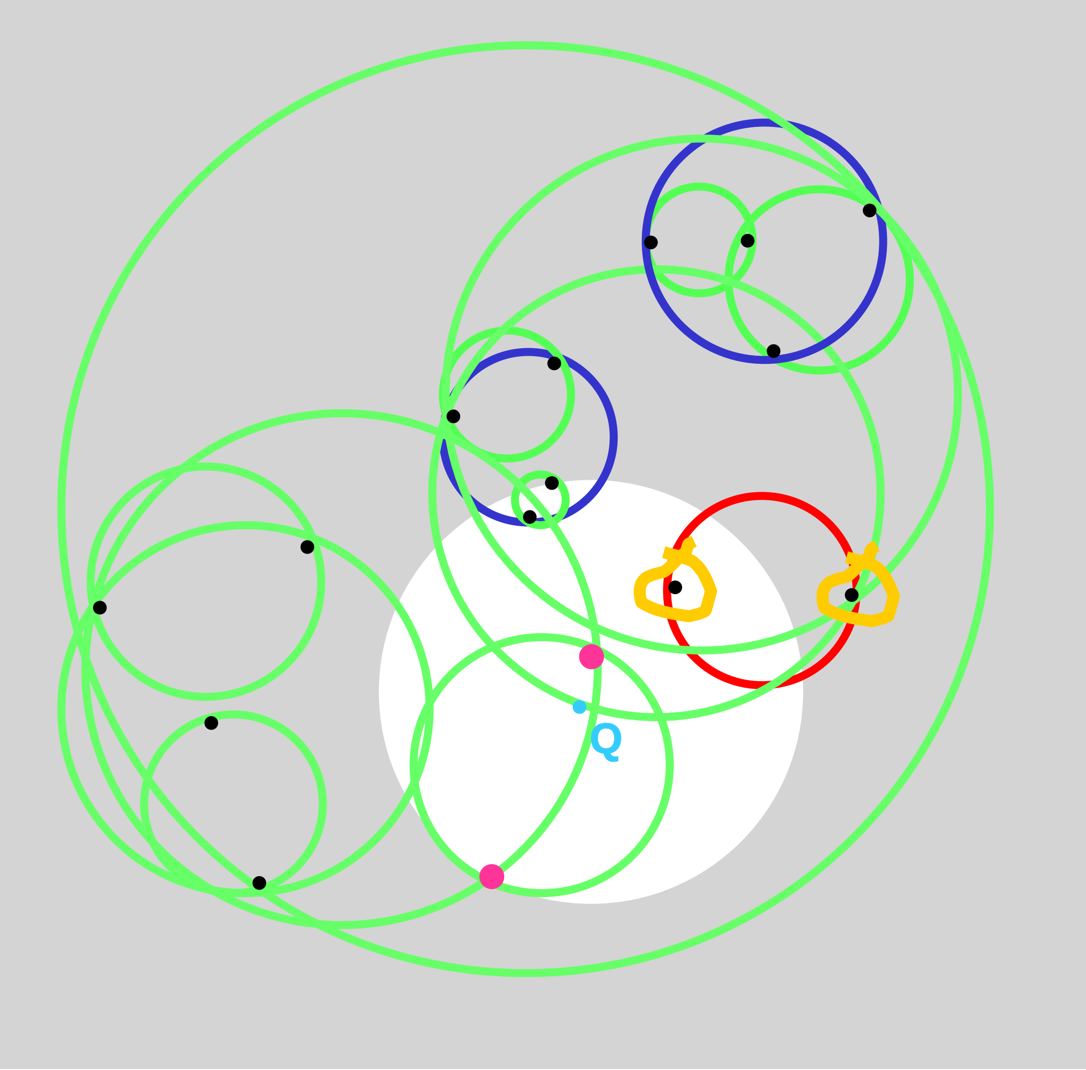

Q

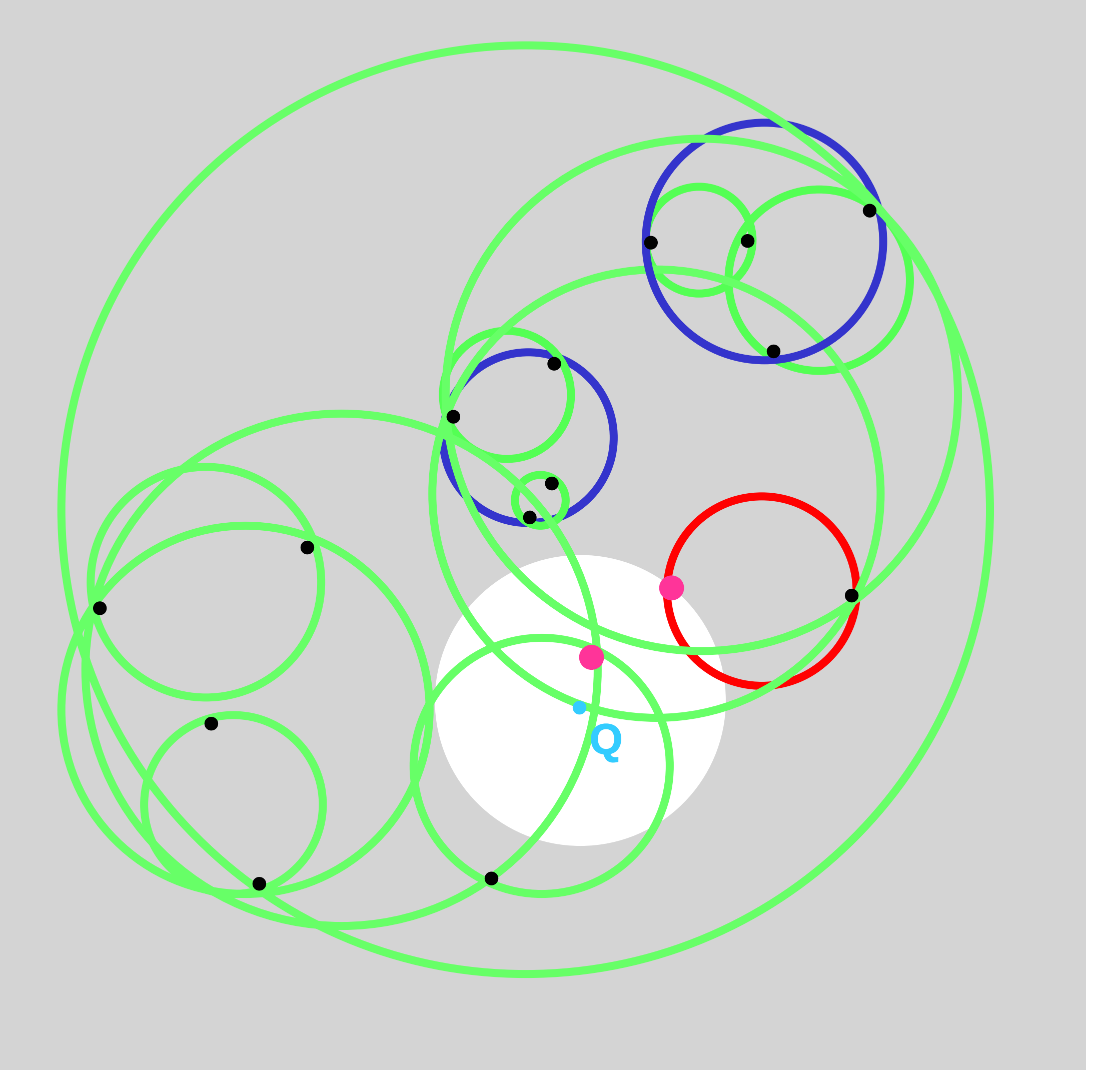

Q

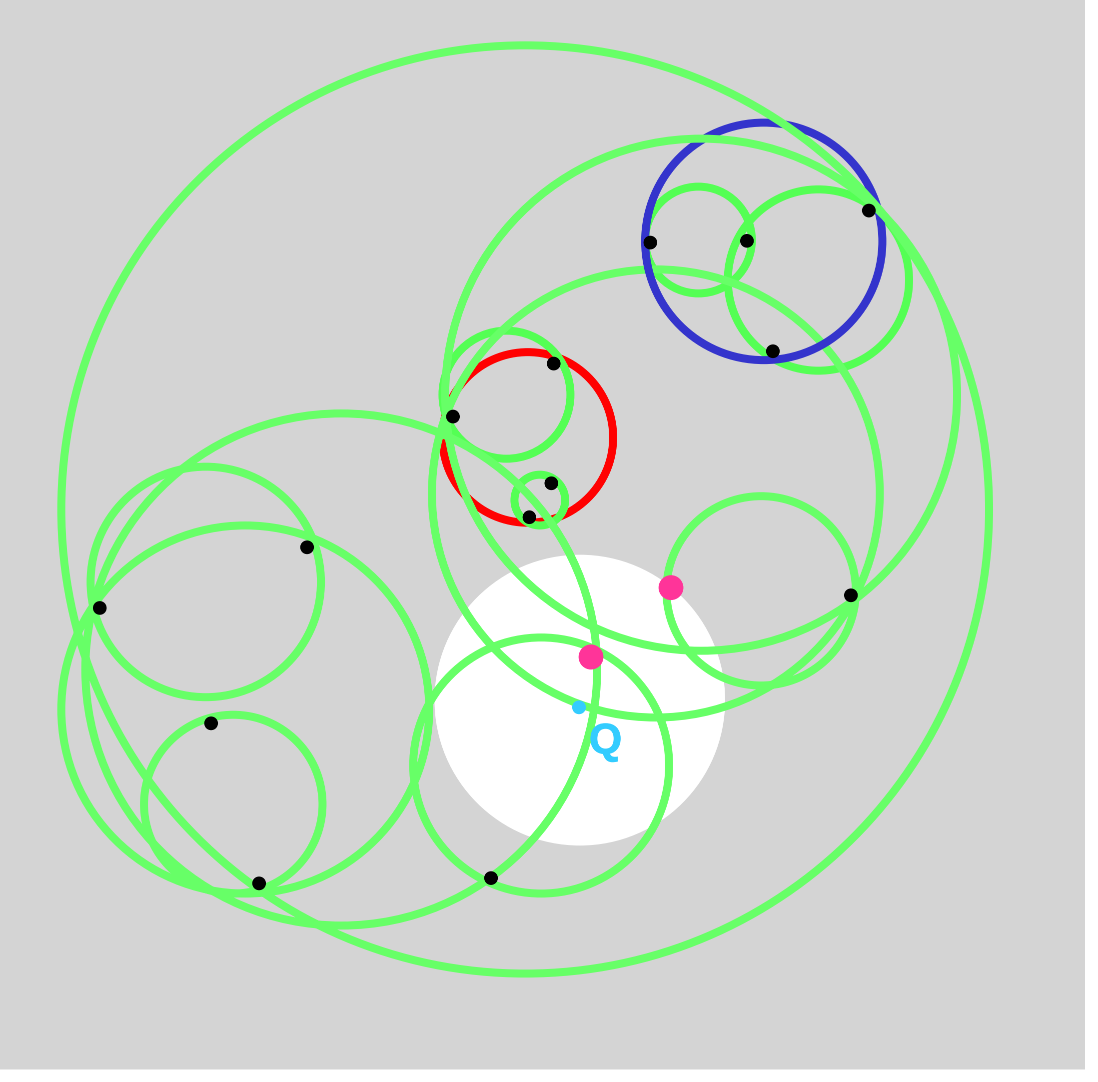

Q

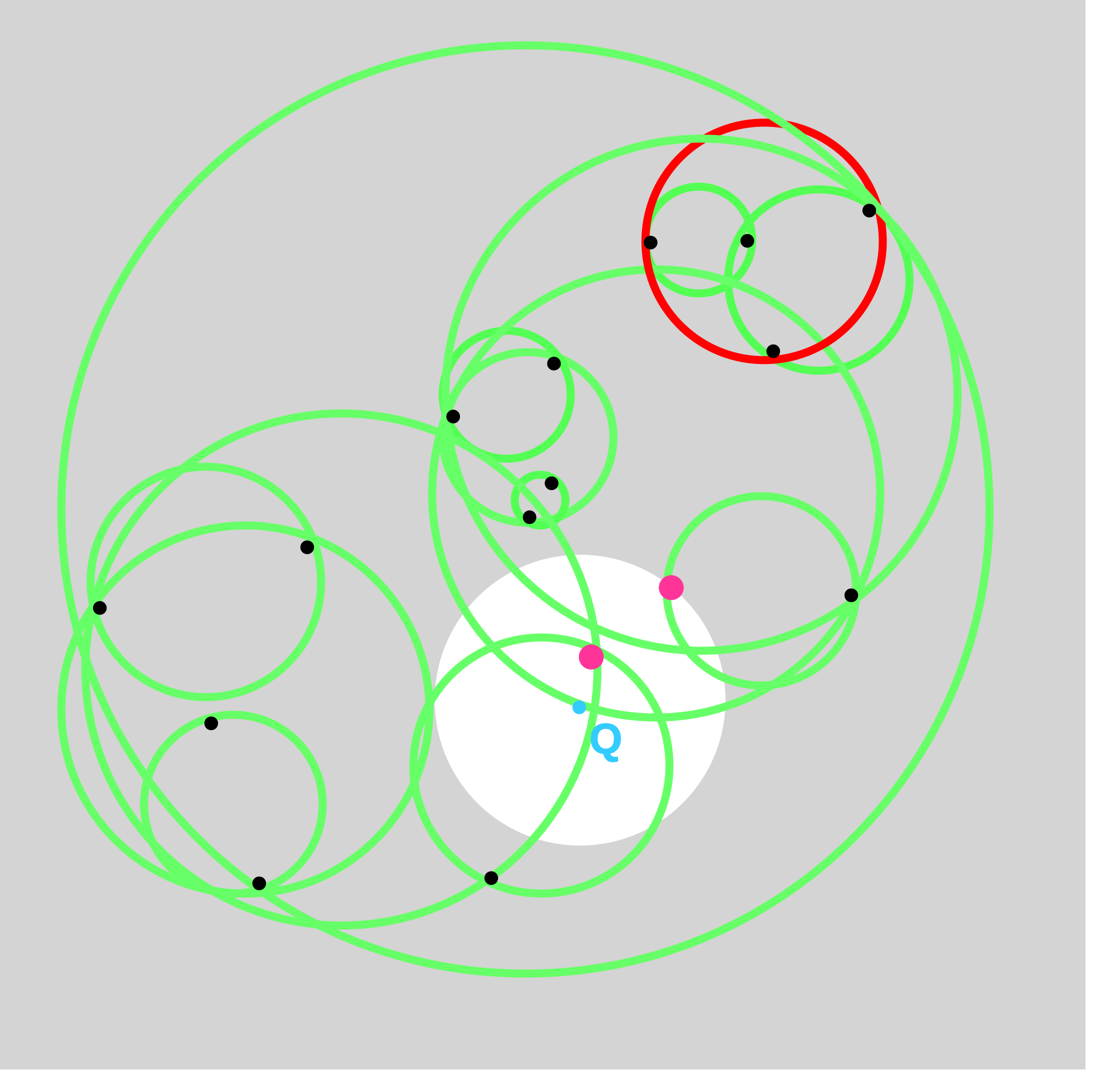

Q

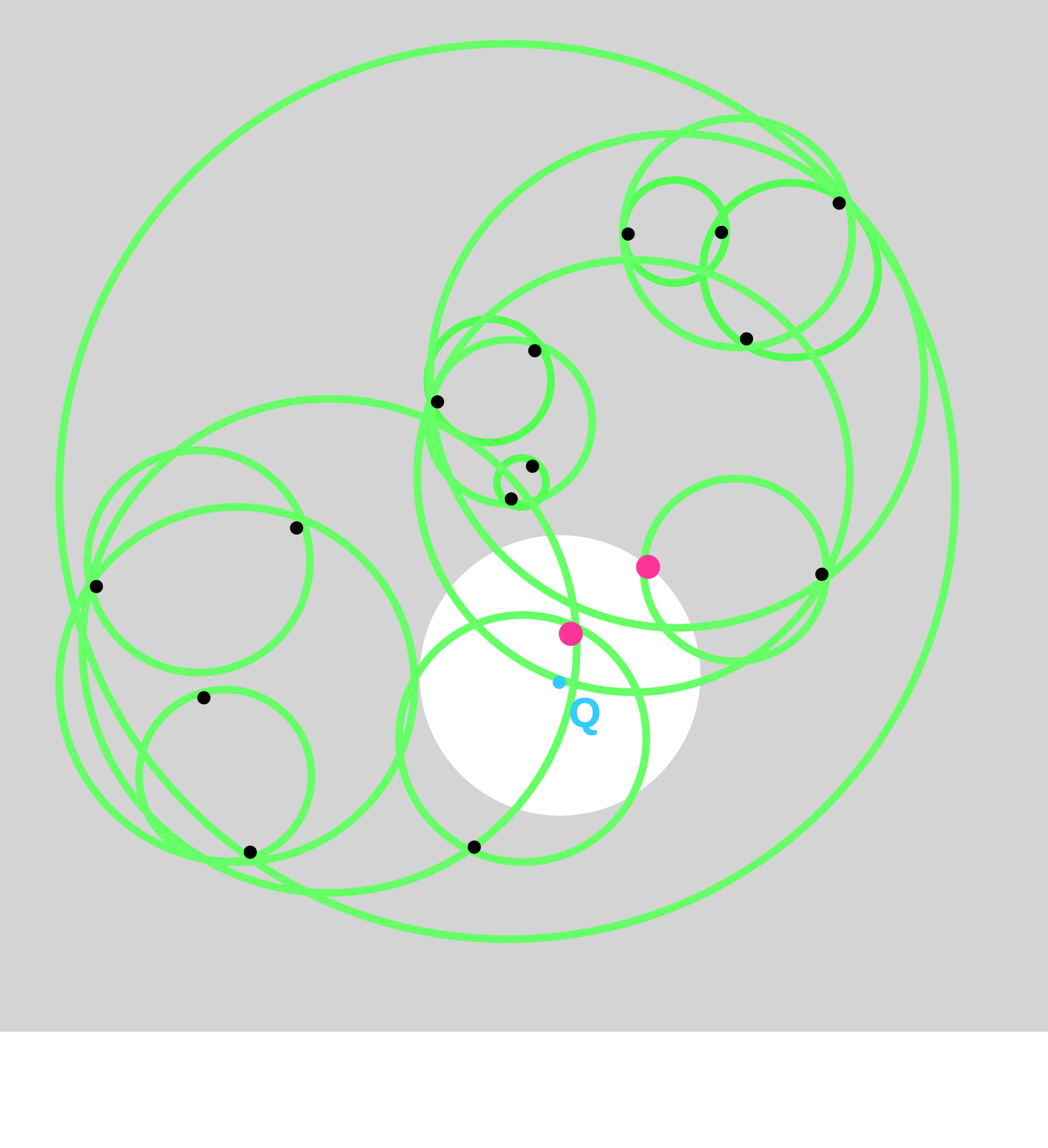

Q


Q


Q


Q


Q


Q

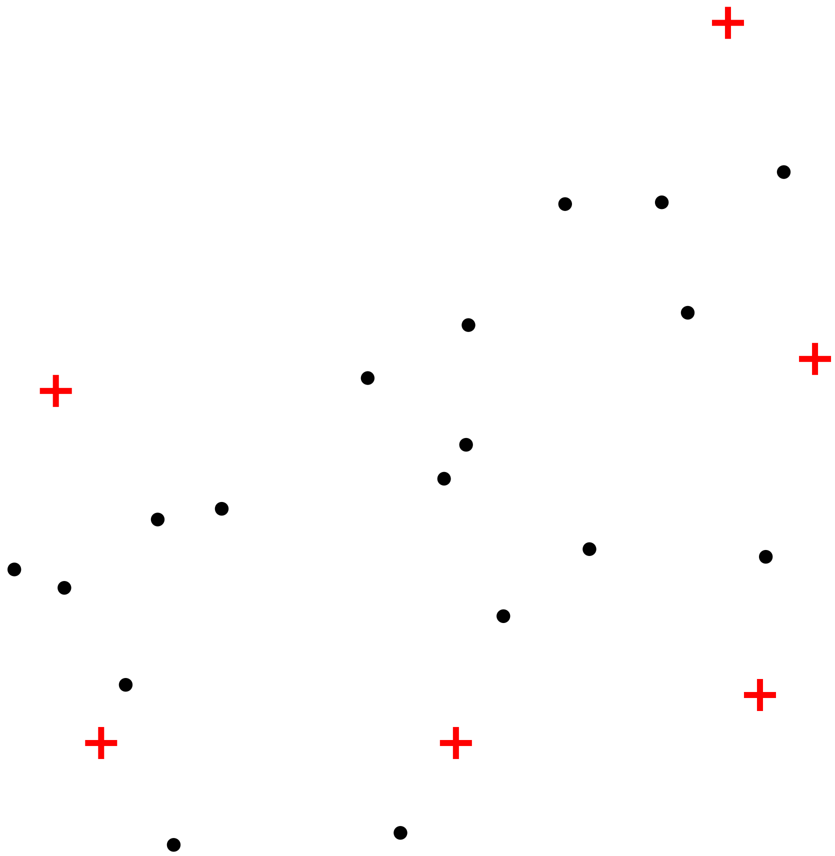
Outline

- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
    - ► Very fast non-parametric classification
    - skewed binary outputs
    - General binary outputs
    - multi-classed outputs
- Very fast kernel-based statistics
    - n-point computations
    - clustering
    - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

KNS2

- Assume binary output
- Assume positive class is much less frequent than negative class
- Assume we want more than a “positive/negative” prediction: we want to know exactly how many of the K-NN are from the +ve class

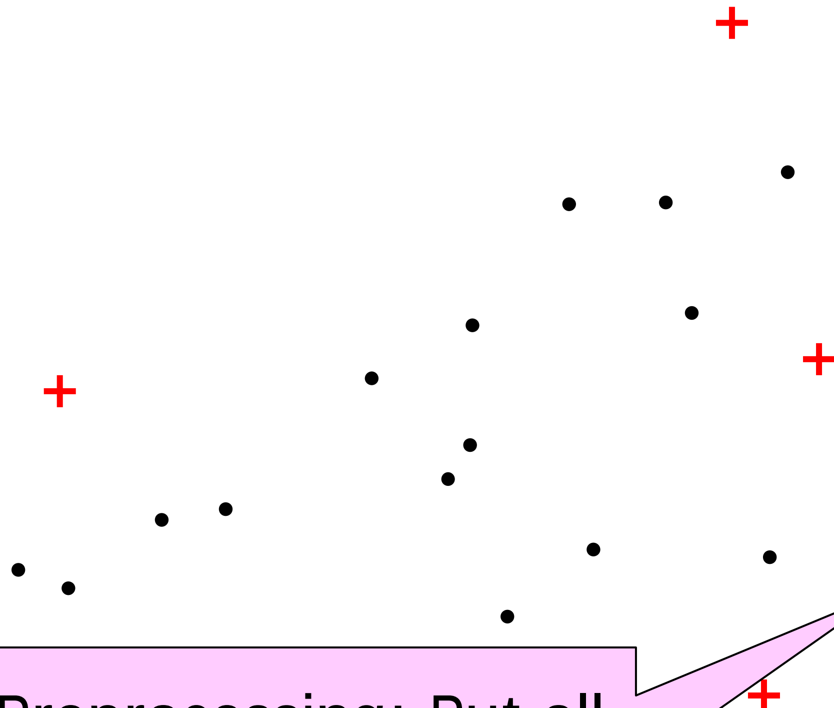
KNS2 does this without finding the K-NN



Assume we have a set of data points.

Some are +ve points (denoted +)

The large majority are –ve points (denoted ●)



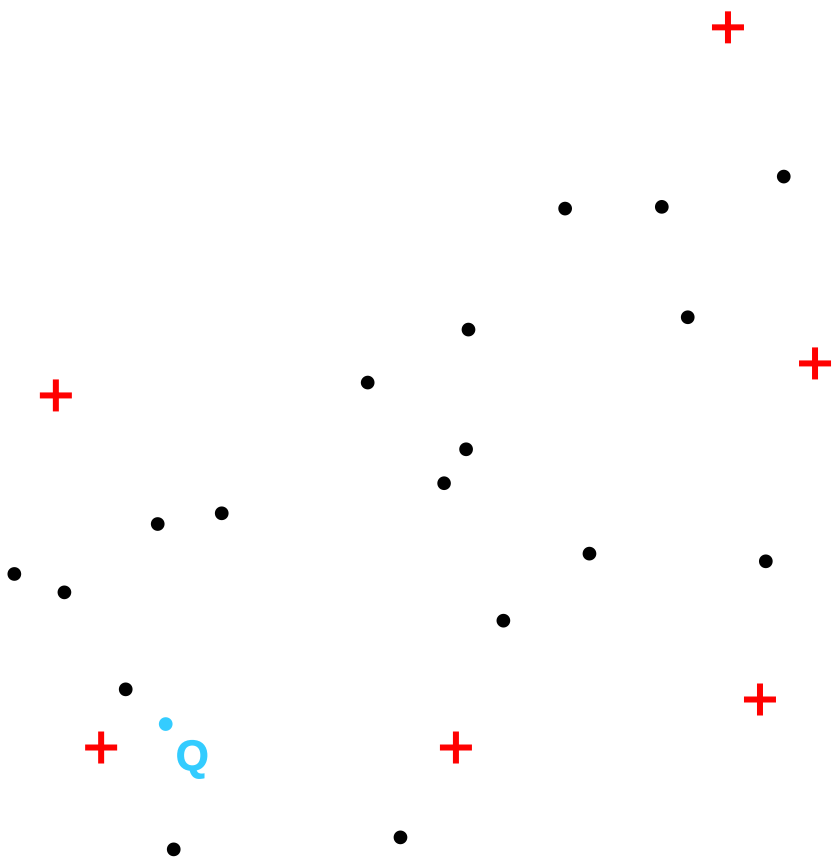
Assume we have a set of data points.

Some are +ve points (denoted **+**)

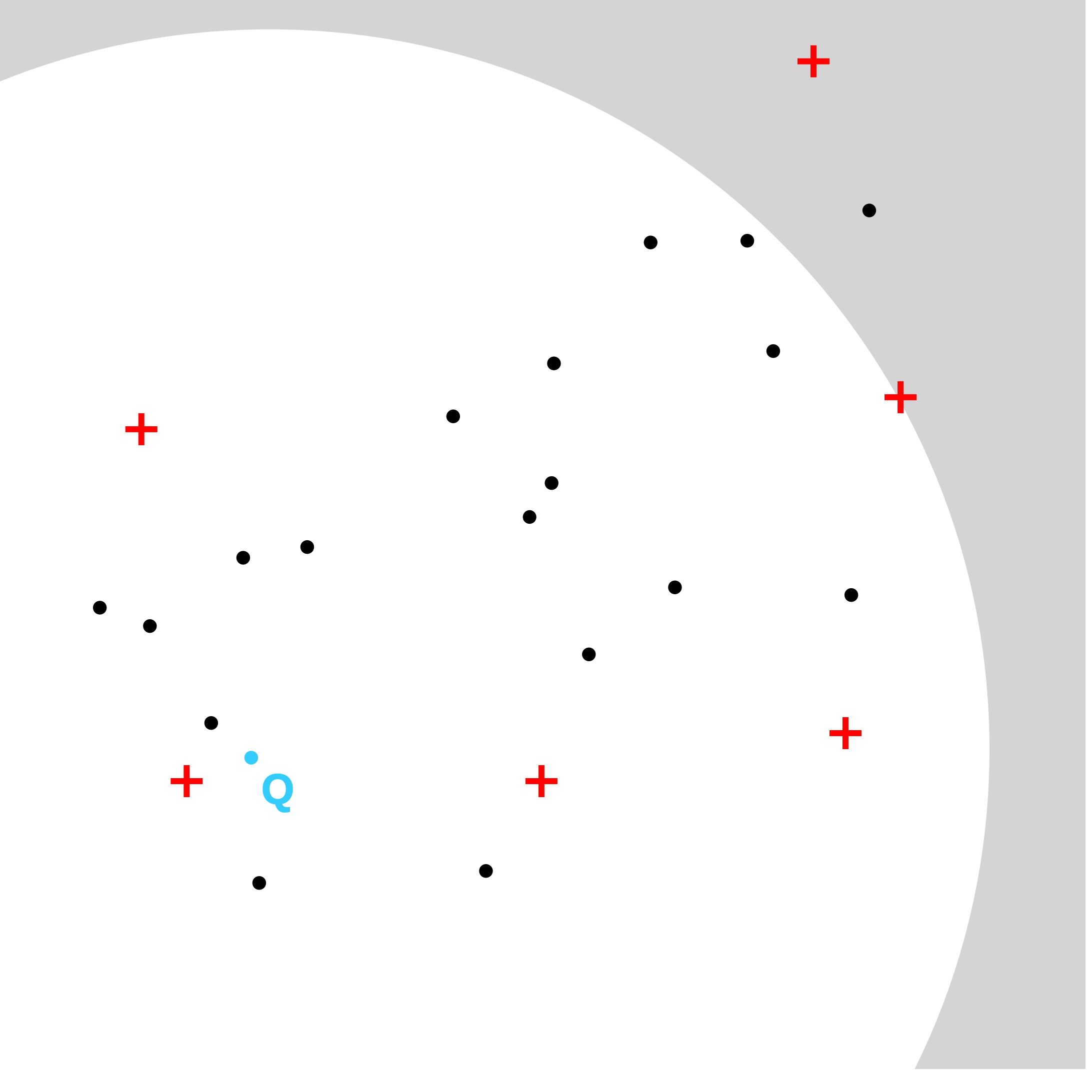
The vast majority are –ve points (denoted ●)

Preprocessing: Put all your +ve points in a small ball tree

Preprocessing: Put all your -ve points in a separate large ball tree

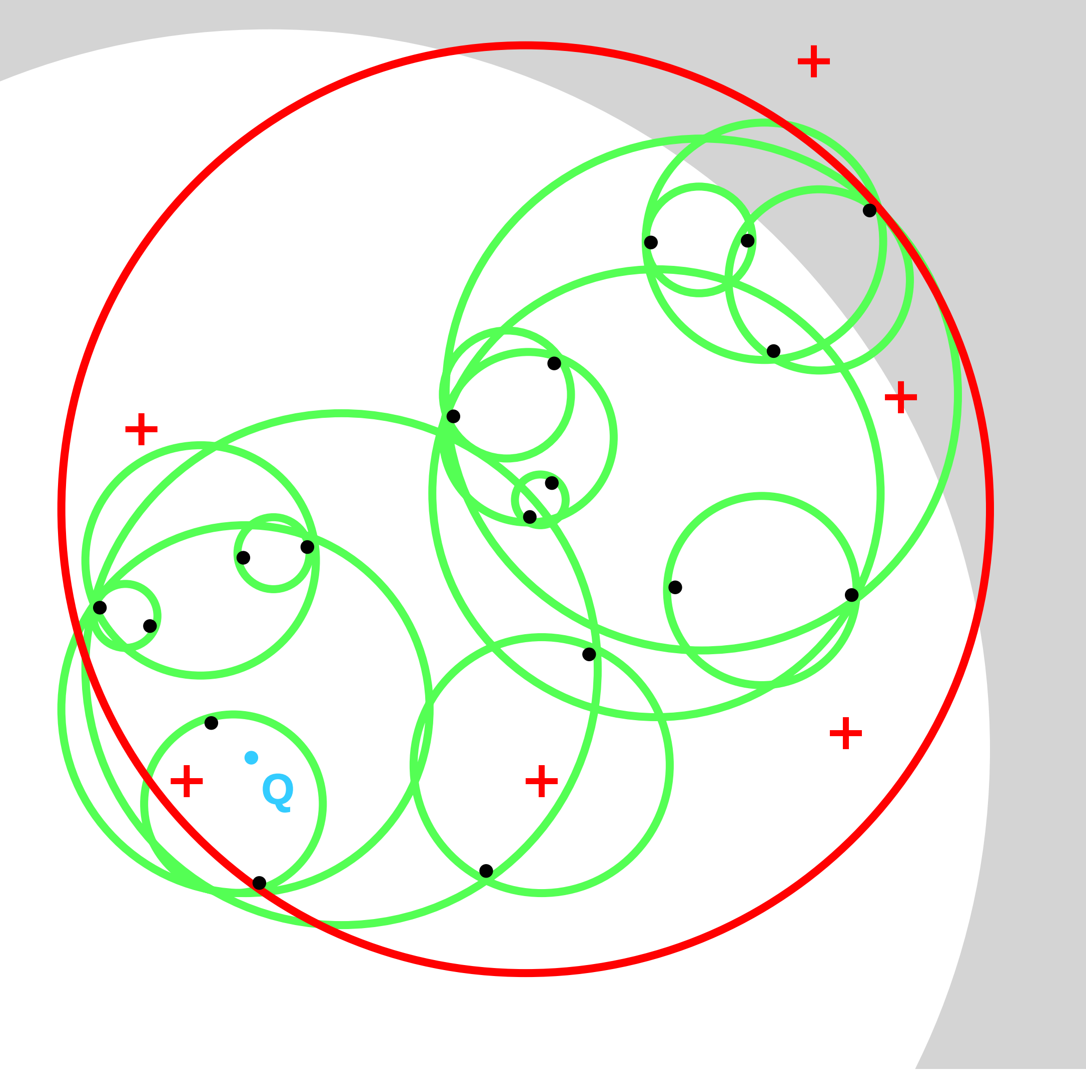


Goal: Find out how many of the 5-nearest neighbors of Q are positive.

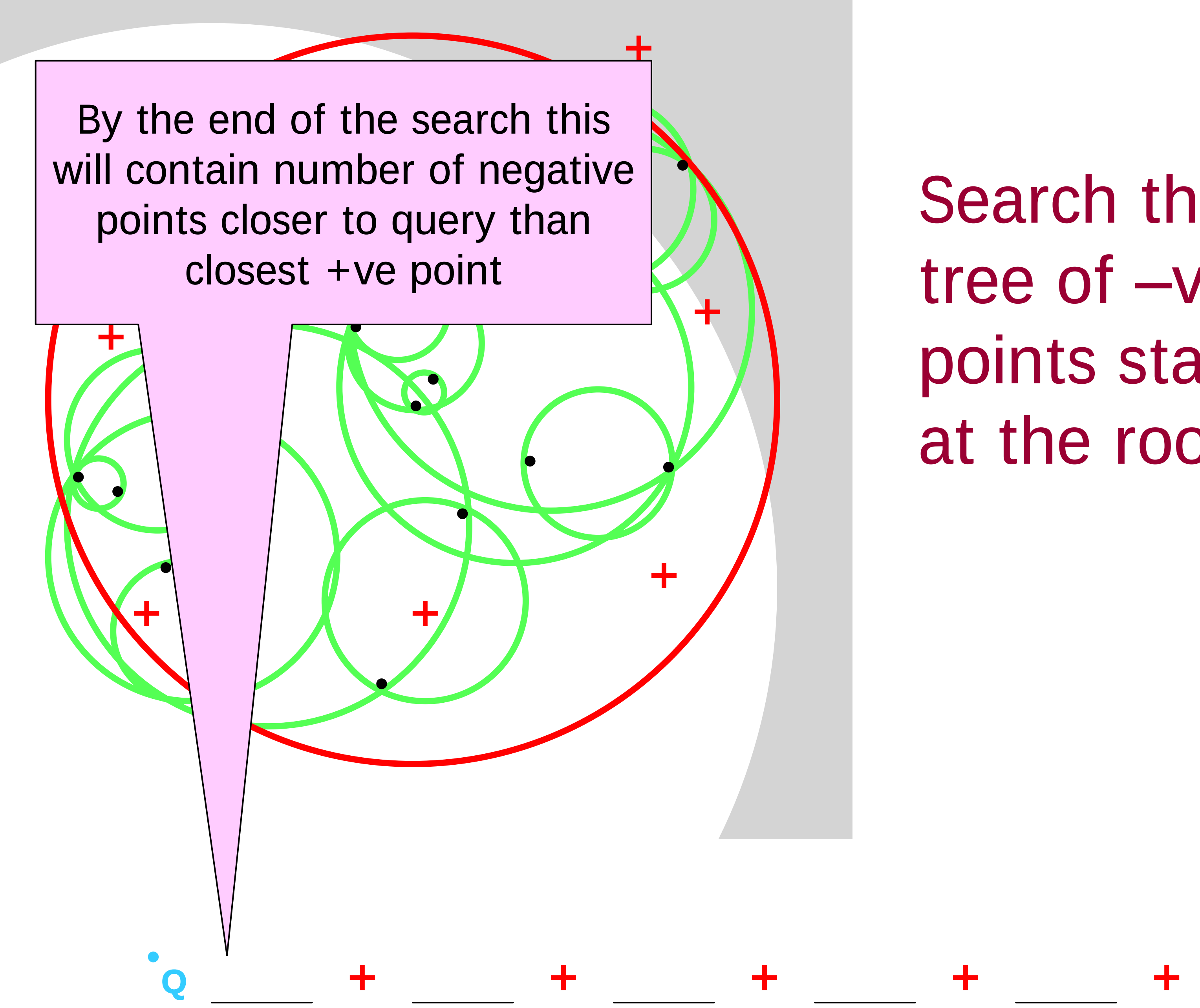


Step One: Find the five nearest +ve points using KNS1.

We're assuming there are far fewer +ves than –ves so this is not the dominant cost.

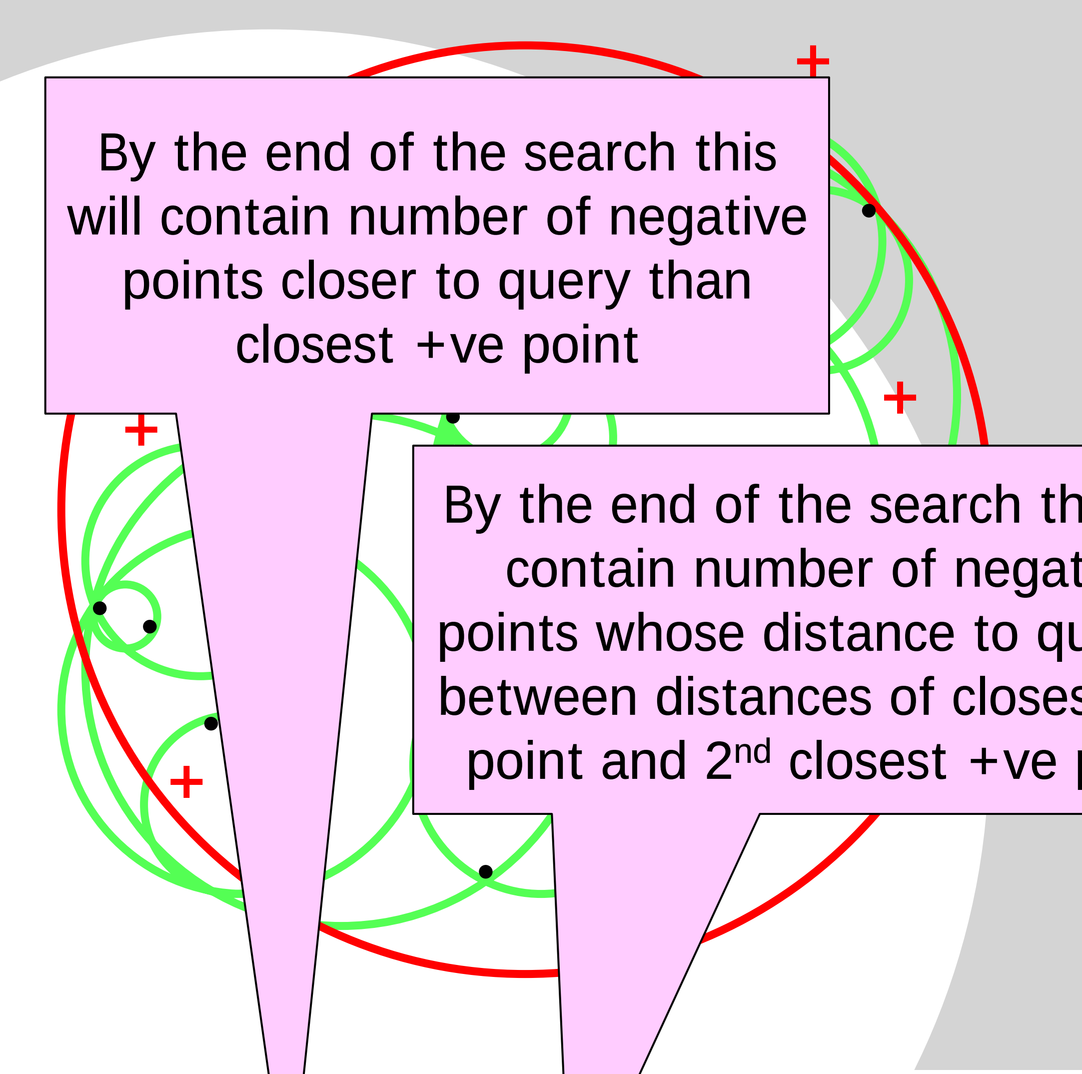


Step 2: Search the ball-tree of –ve points starting at the root.



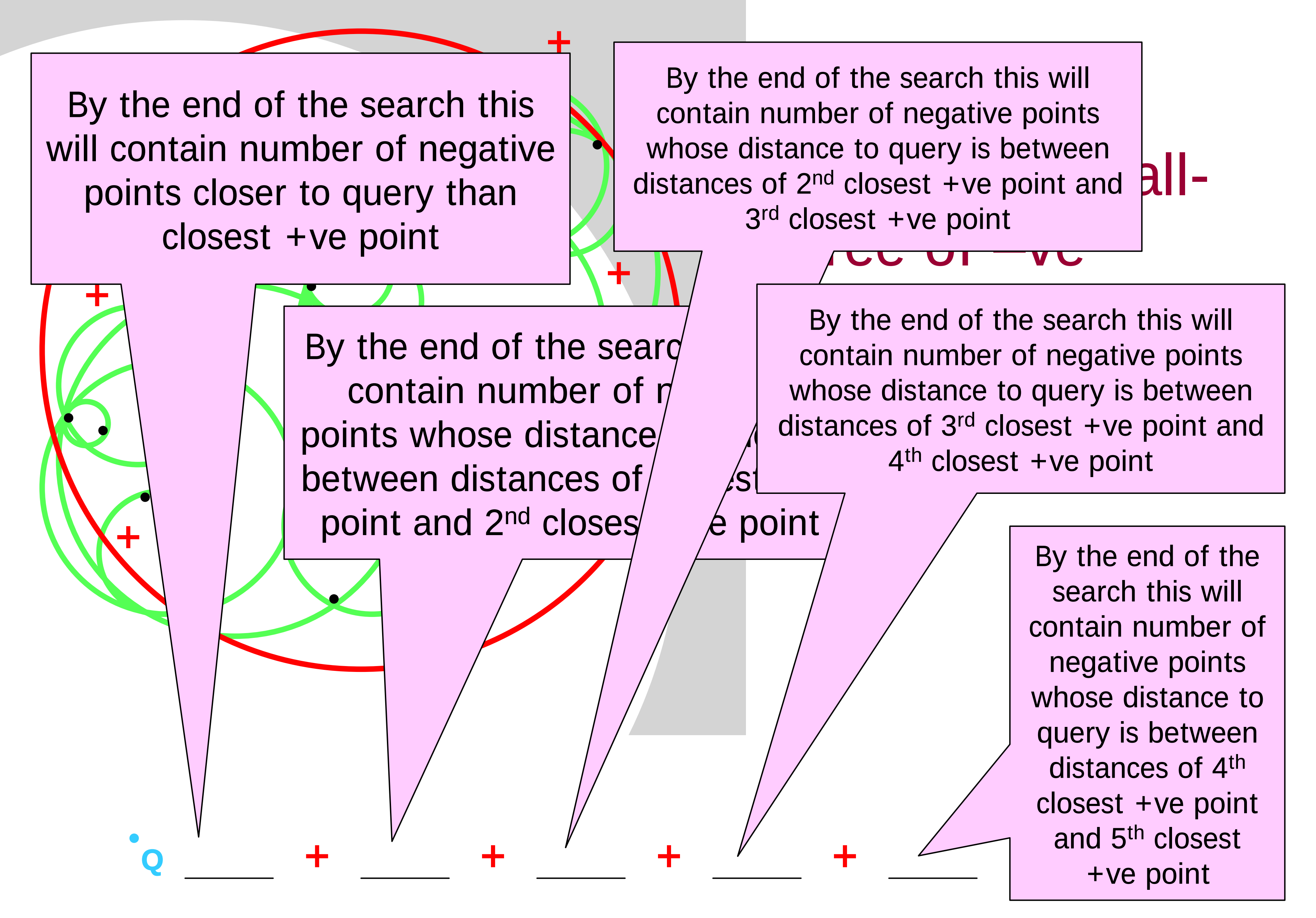
Search the ball-tree of –ve points starting at the root.

Search the ball-tree of –ve points starting the root.

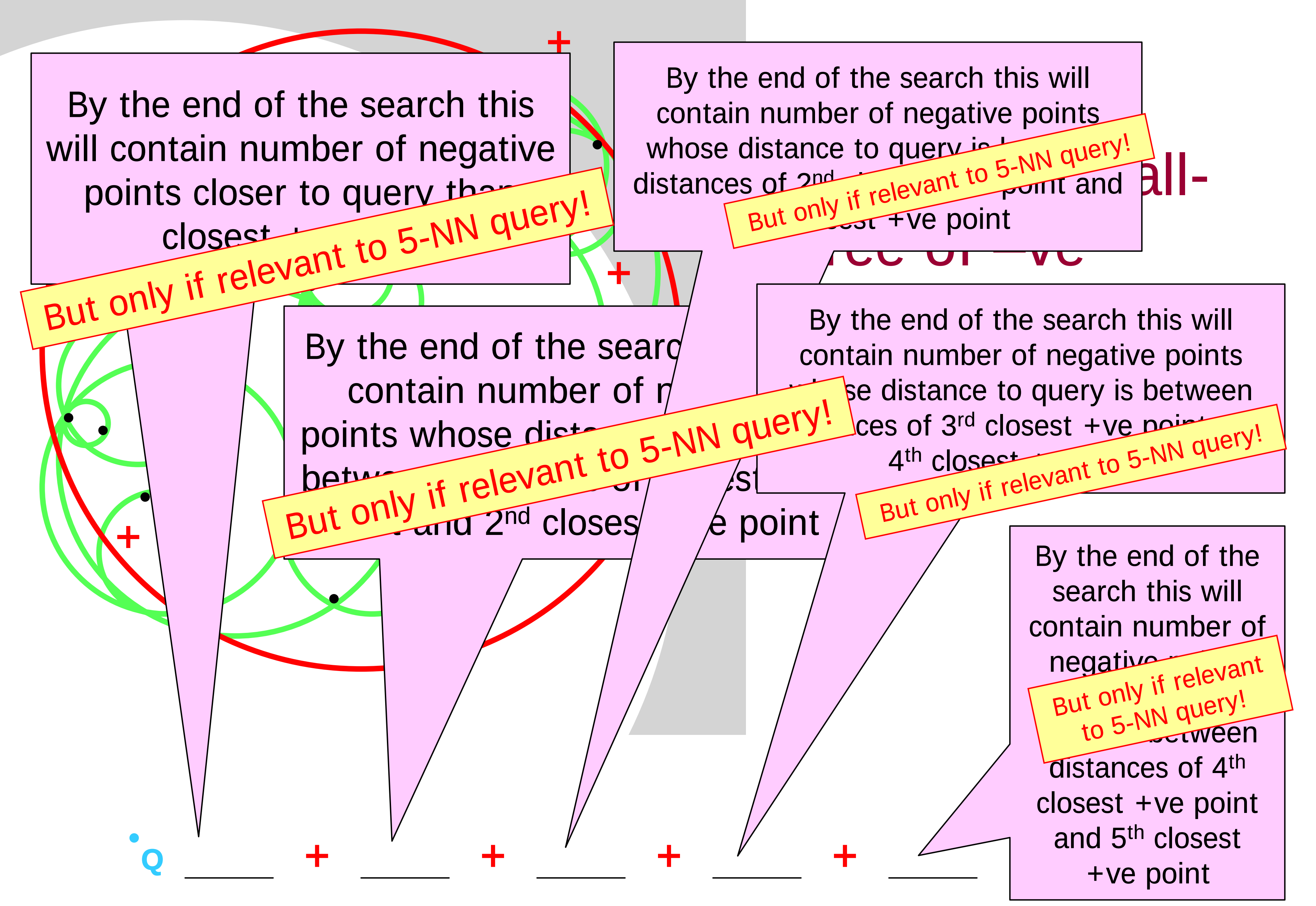


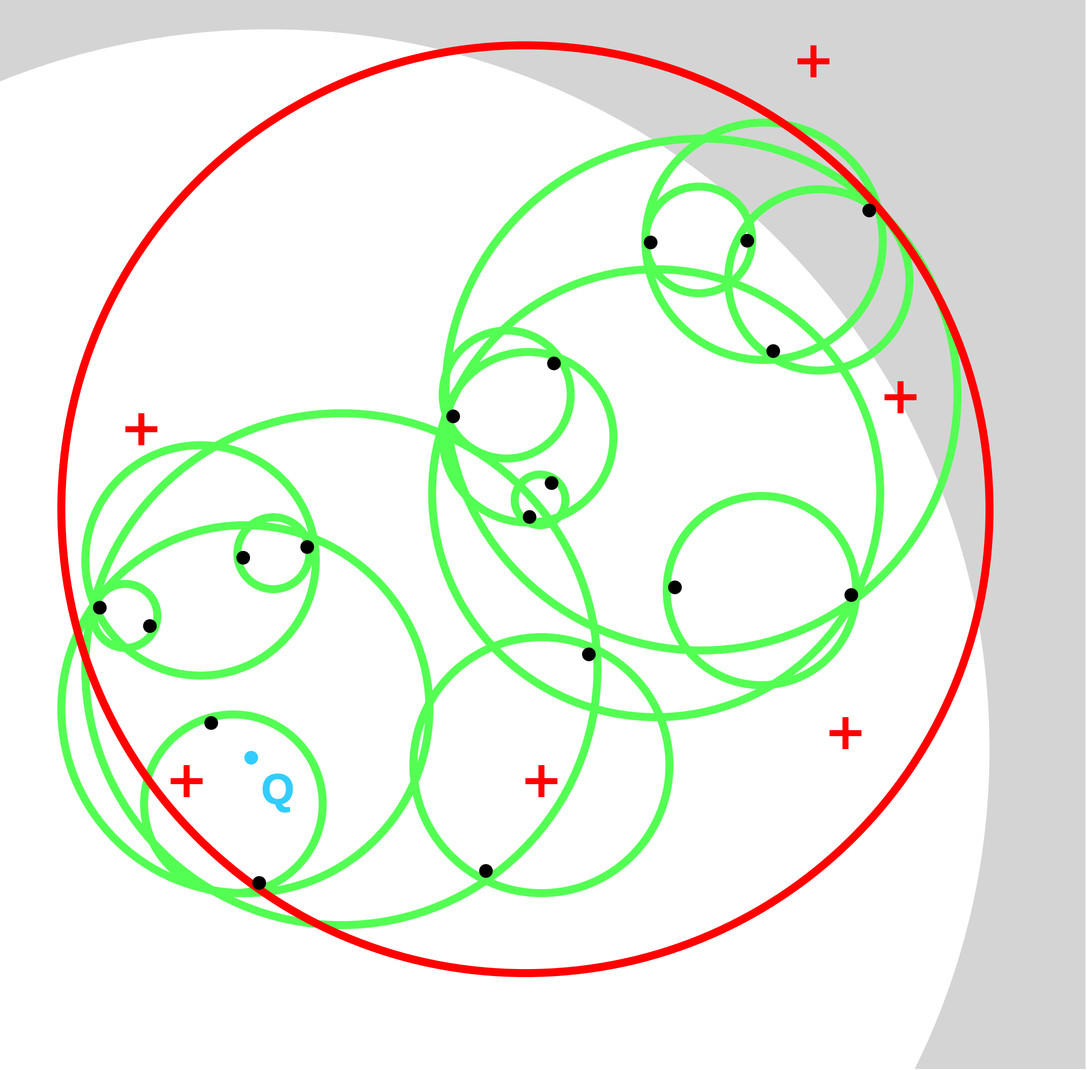
By the end of the search this will contain number of negative points closer to query than closest +ve point

By the end of the search this will contain number of negative points whose distance to query is between distances of closest +ve point and 2nd closest +ve point

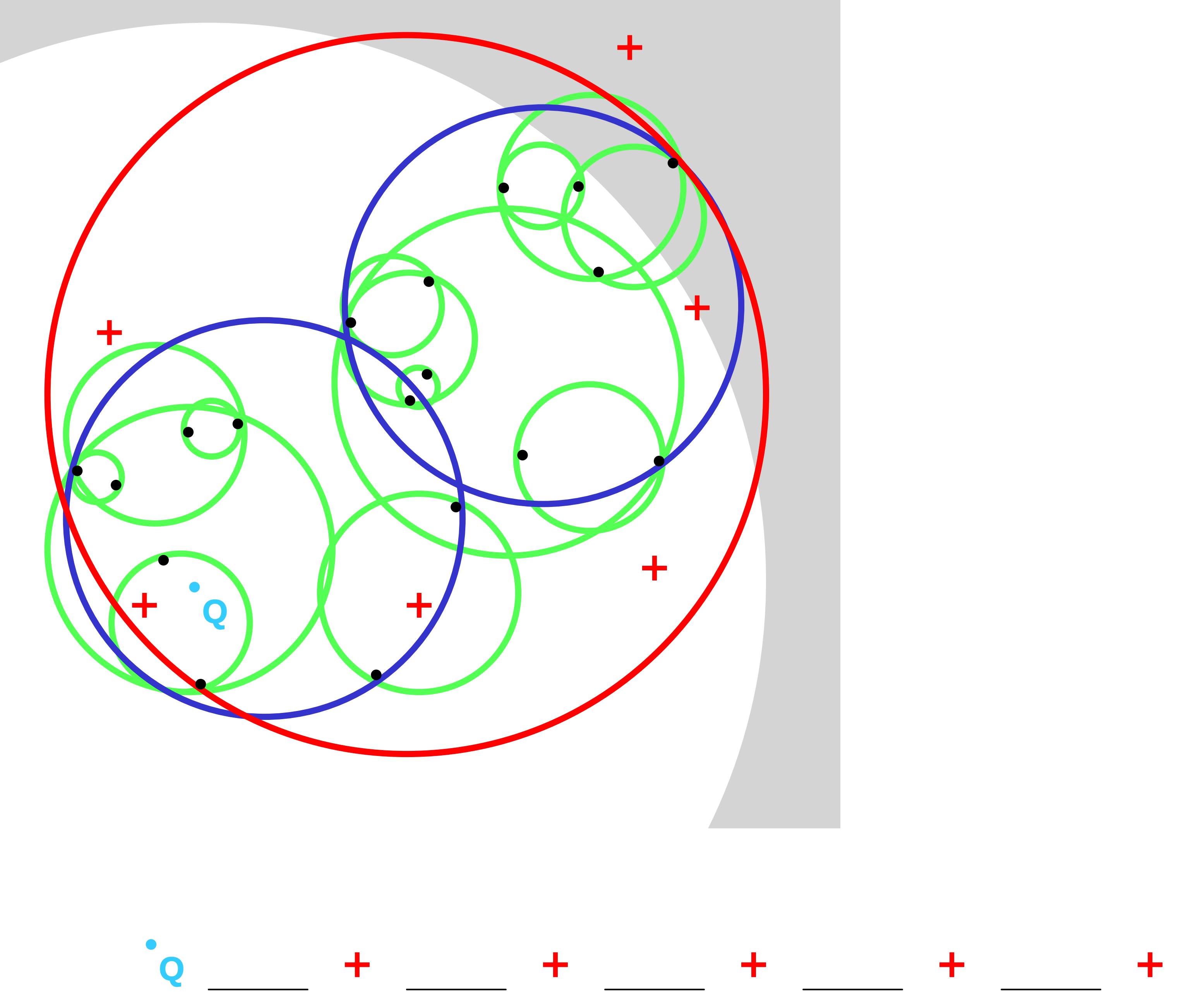


By the end of the search this will contain number of negative points closer to query than closest +ve point
By the end of the search this will contain number of negative points whose distance to query is between distances of 2nd closest +ve point and 3rd closest +ve point
By the end of the search this will contain number of negative points whose distance to query is between distances of 3rd closest +ve point and 4th closest +ve point
By the end of the search this will contain number of negative points whose distance to query is between distances of 4th closest +ve point and 5th closest +ve point
Q

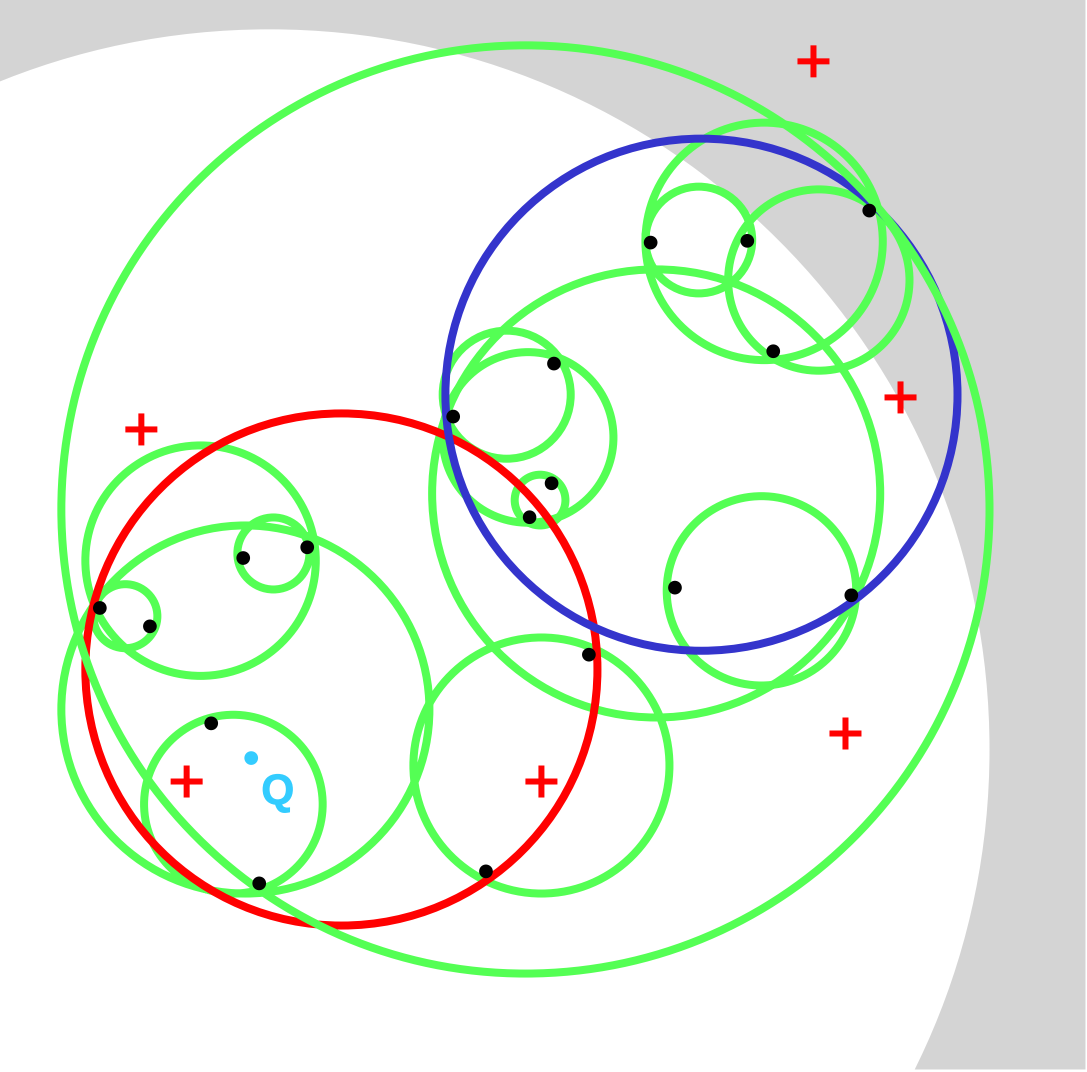

all-
ree of –ve
By the end of the search this will contain number of negative points closer to query tha
closest
But only if relevant to 5-NN query!
By the end of the search this will contain number of negative points whose distance to query is
distances of 2nd
point and
closest +ve point
But only if relevant to 5-NN query!
By the end of the searc
contain number of n
points whose dist
betw
t and 2nd closes
e point
But only if relevant to 5-NN query!
By the end of the search this will contain number of negative points
se distance to query is between
ces of 3rd closest +ve poin
4th closest
But only if relevant to 5-NN query!
By the end of the search this will contain number of
negative
between
distances of 4th closest +ve point and 5th closest +ve point
But only if relevant to 5-NN query!
Q
+
+
+
+
+
+
+



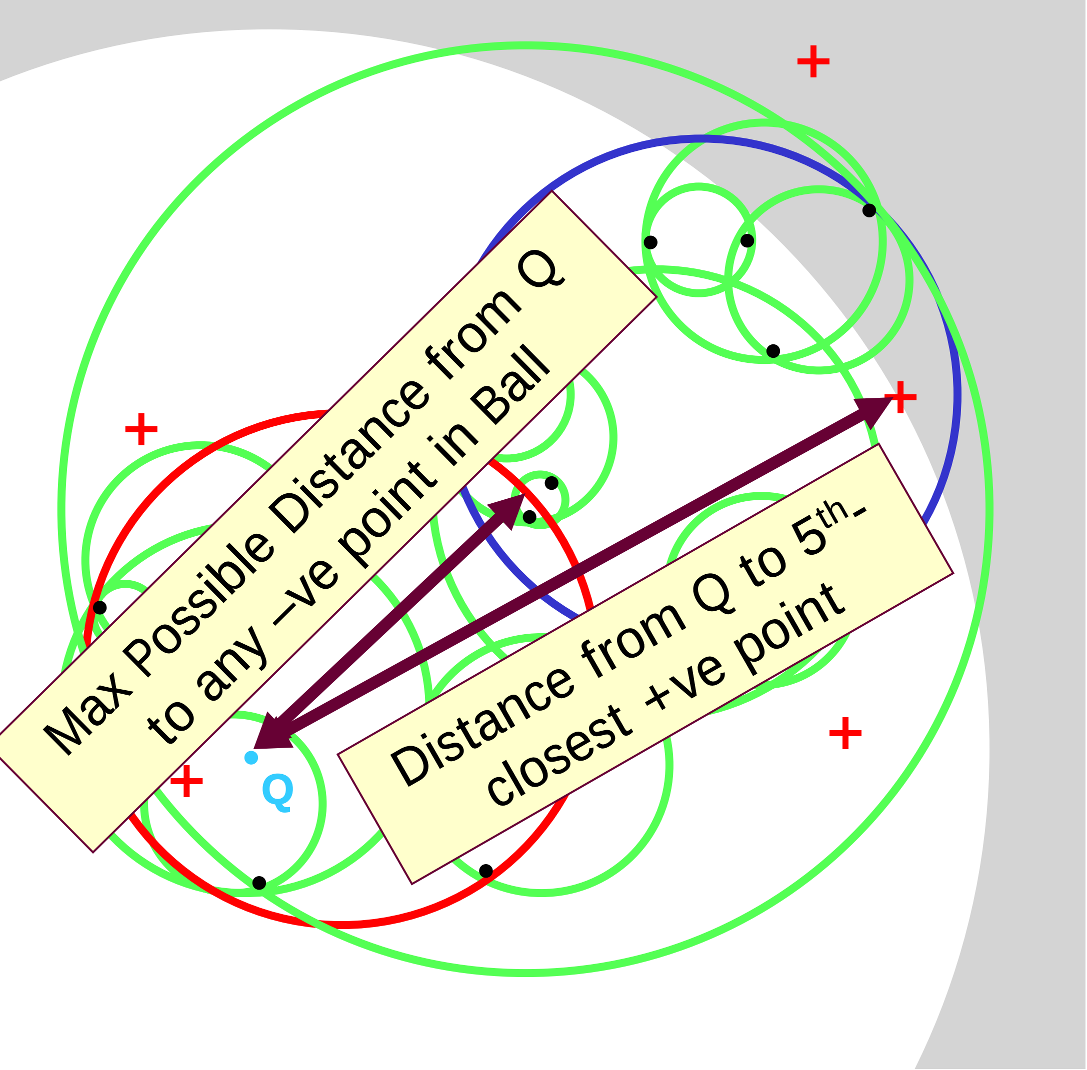
Search the ball-tree of –ve points starting at the root.



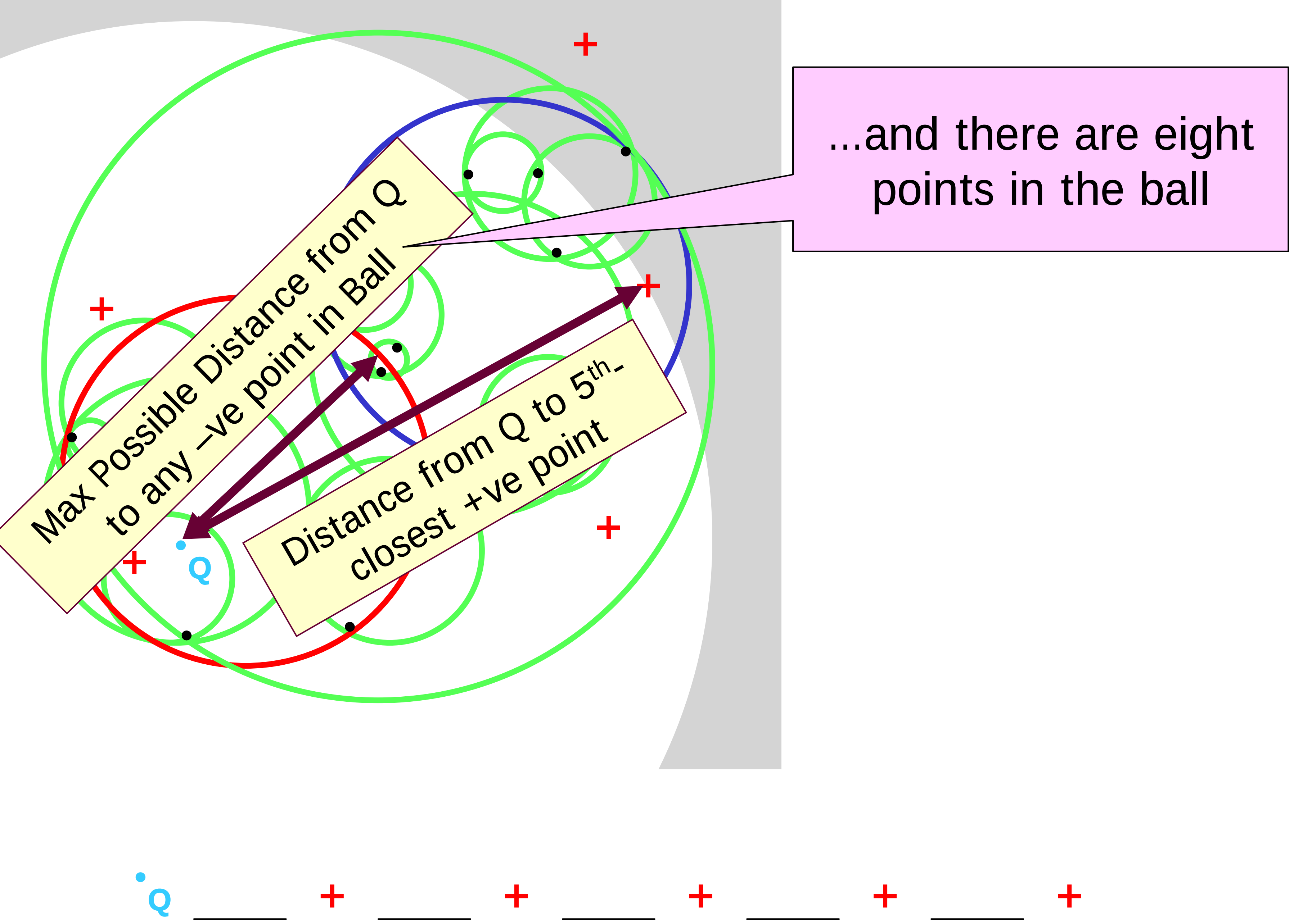
Q
Q
+ + + + +



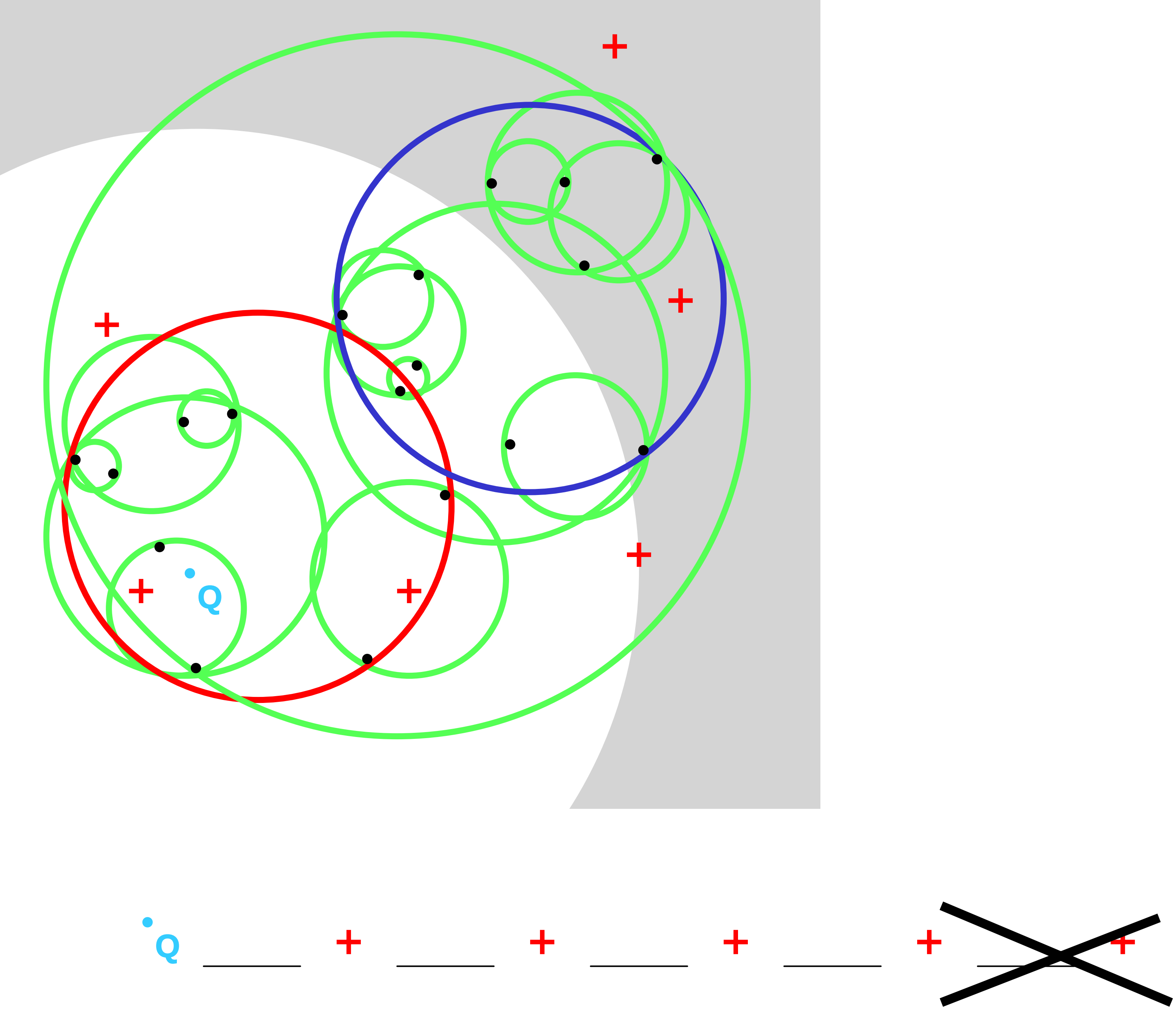
Q ______ + ______ + ______ + ______ + ______ +


Max Possible Distance from Q
to any –ve point in Ball
Distance from Q to 5th-
closest +ve point
Q

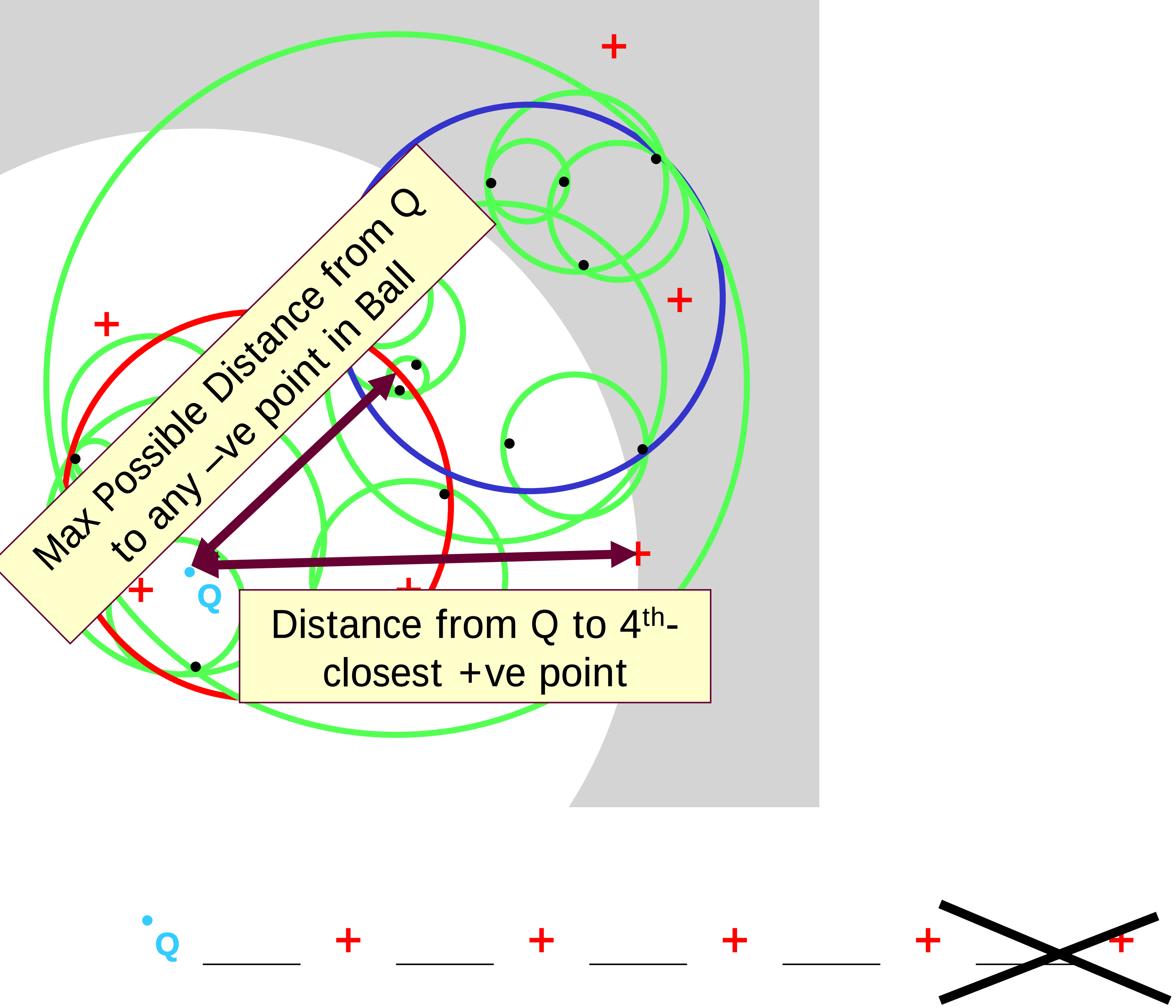

Q

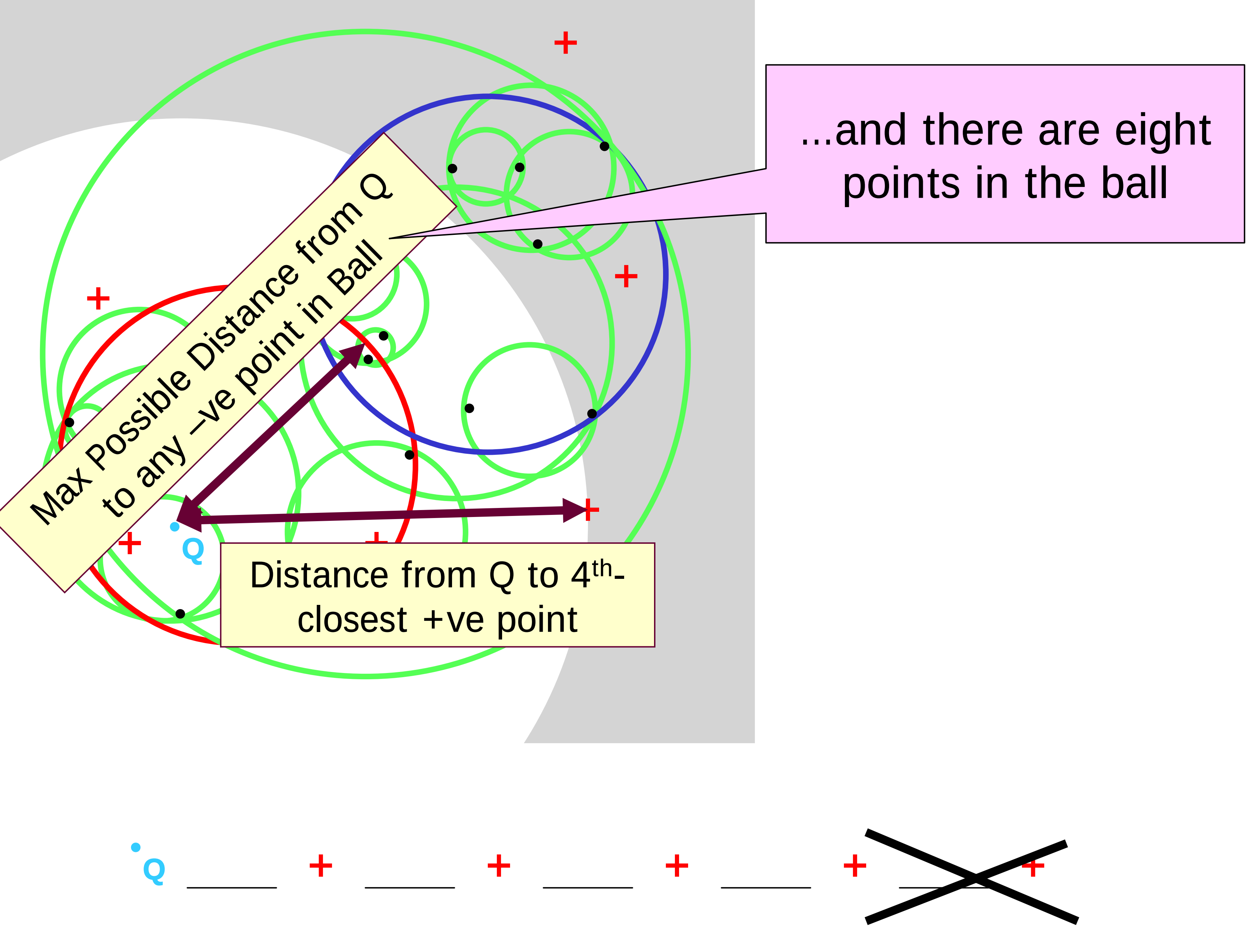


...and there are eight points in the ball
Max Possible Distance from Q to any –ve point in Ball
Distance from Q to 5th-closest +ve point
Q

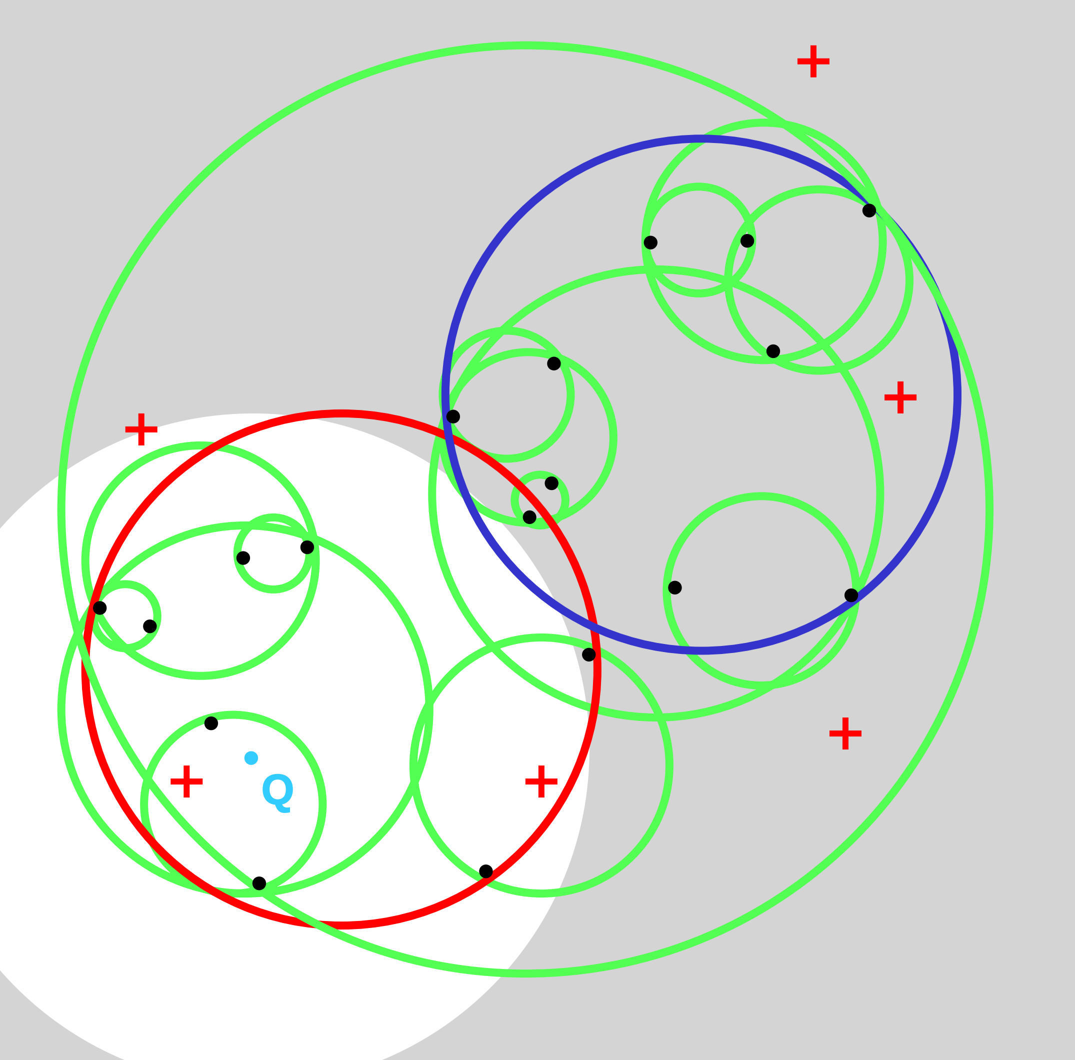


Q
Q

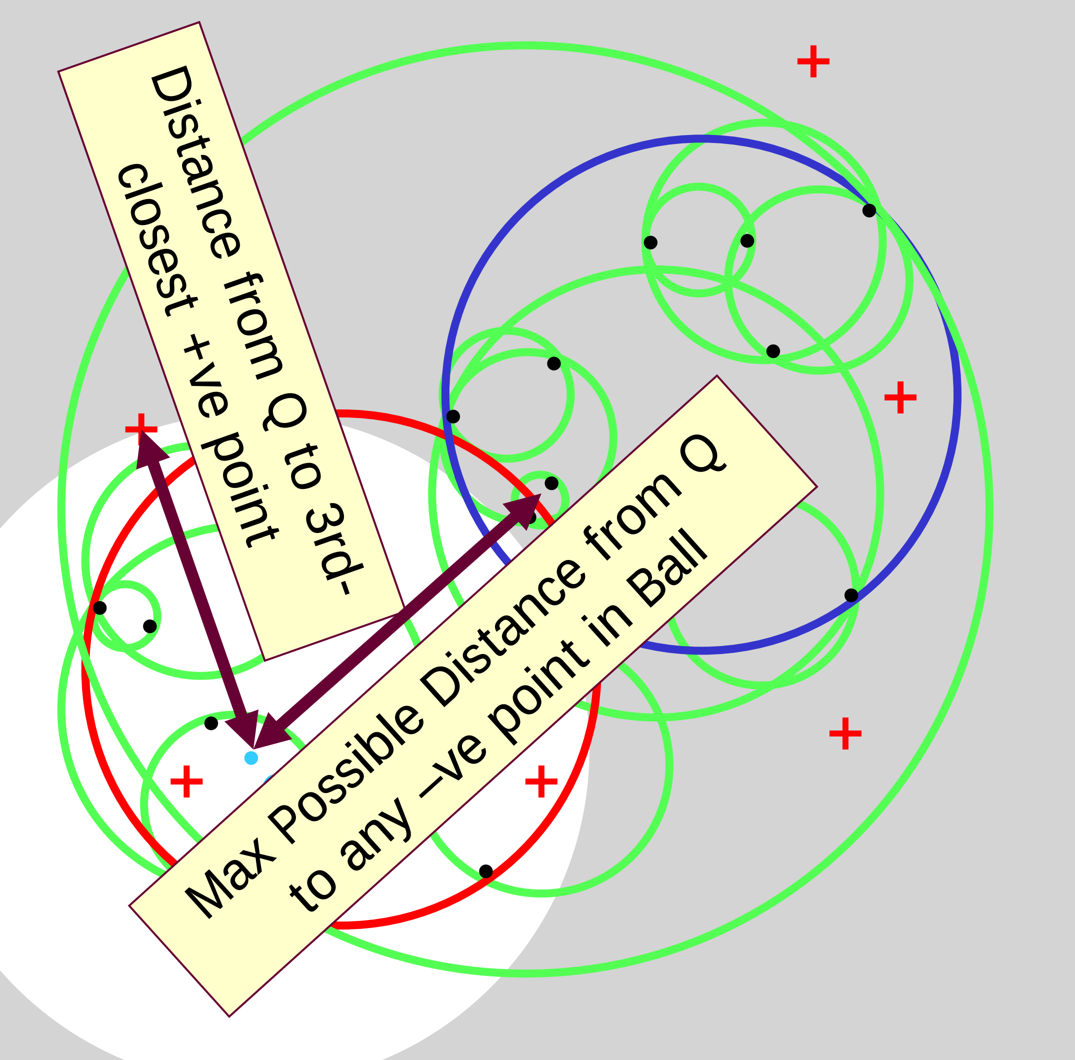

Max Possible Distance from Q
to any –ve point in Ball
Q
Distance from Q to 4th-closest +ve point
Q



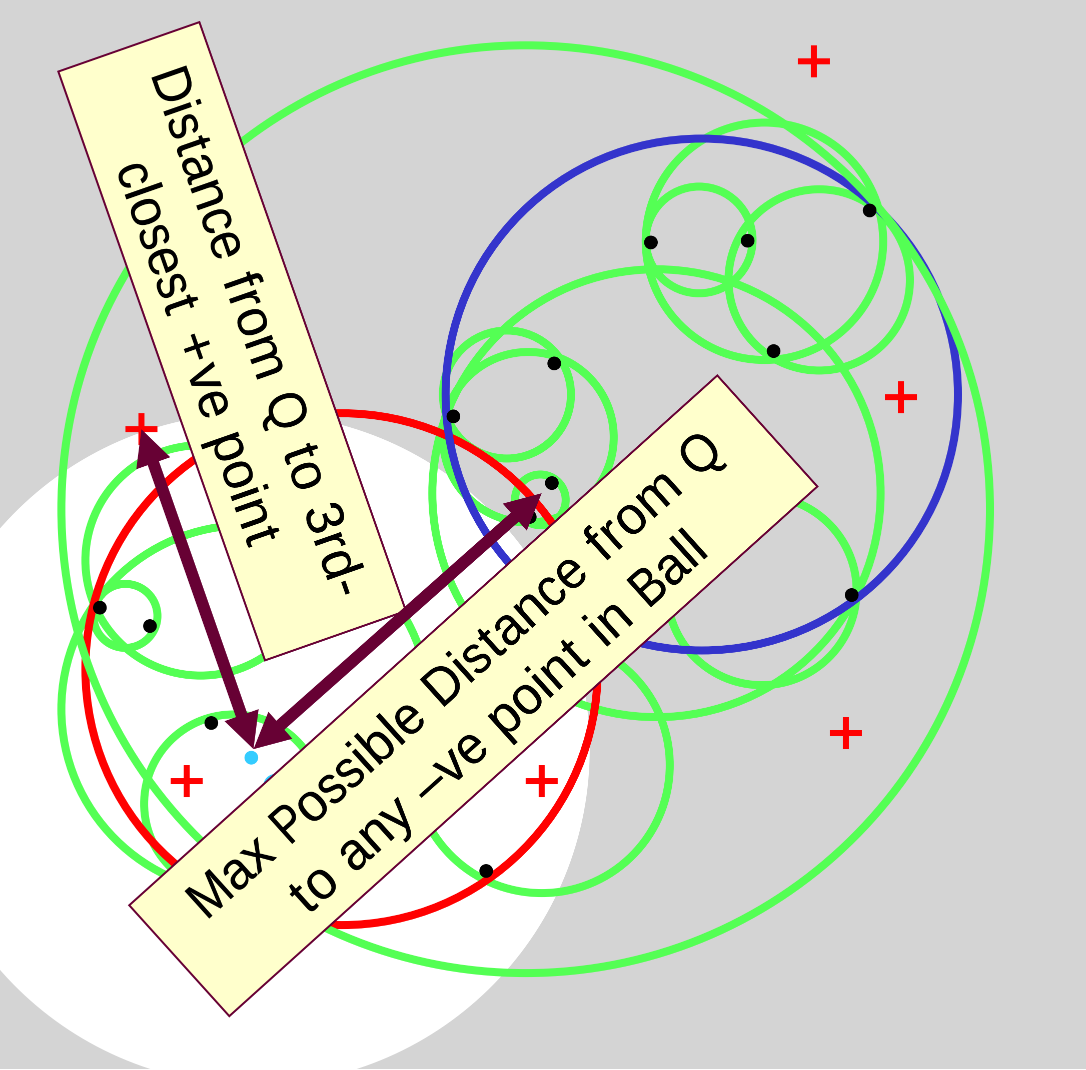
...and there are eight points in the ball
Max Possible Distance from Q to any –ve point in Ball
Distance from Q to 4th-closest +ve point
Q
+
Q
+ + + + +


Q


Q

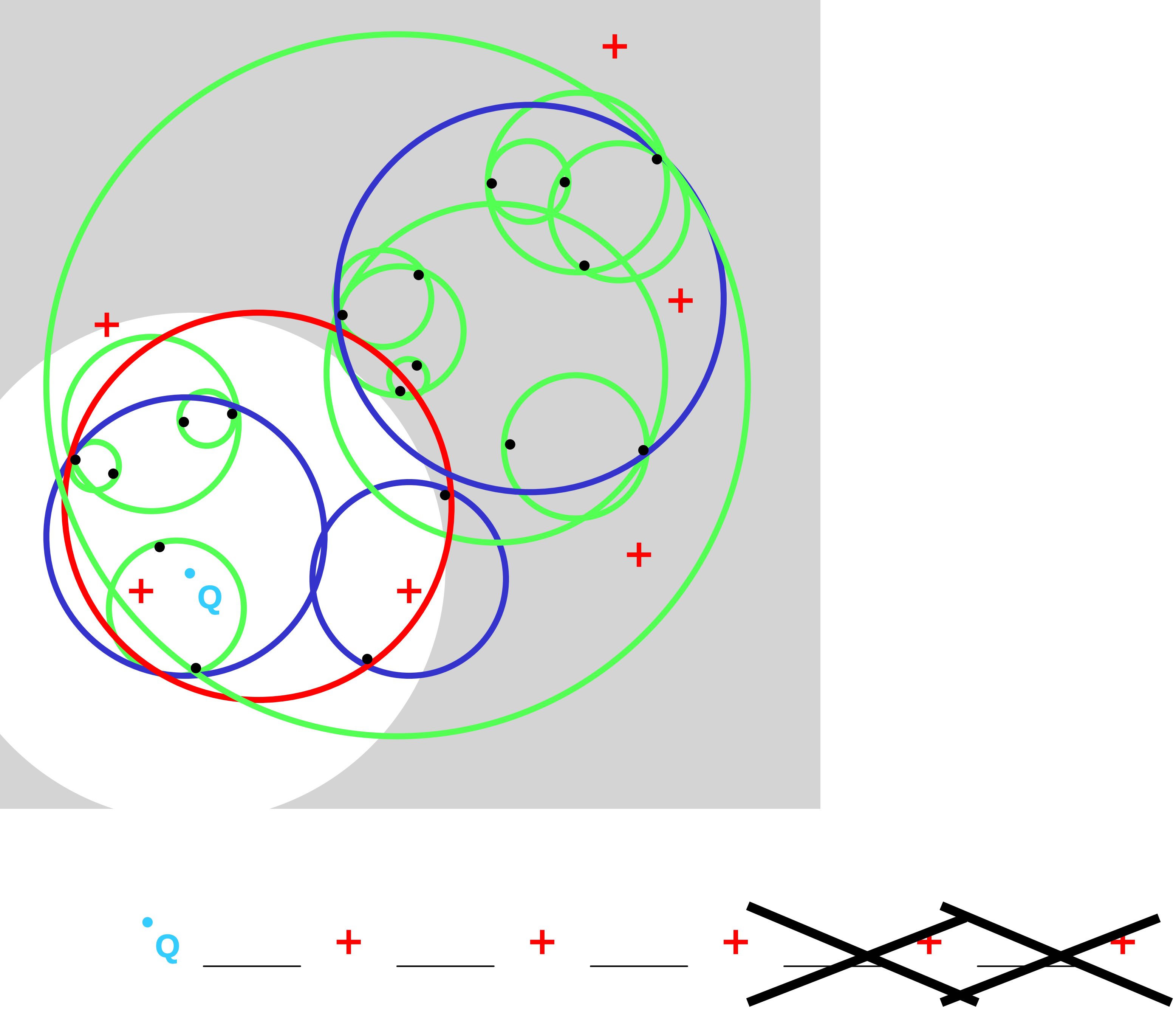

Distance from Q to 3rd-closest +ve point
Max Possible Distance from Q to any –ve point in Ball

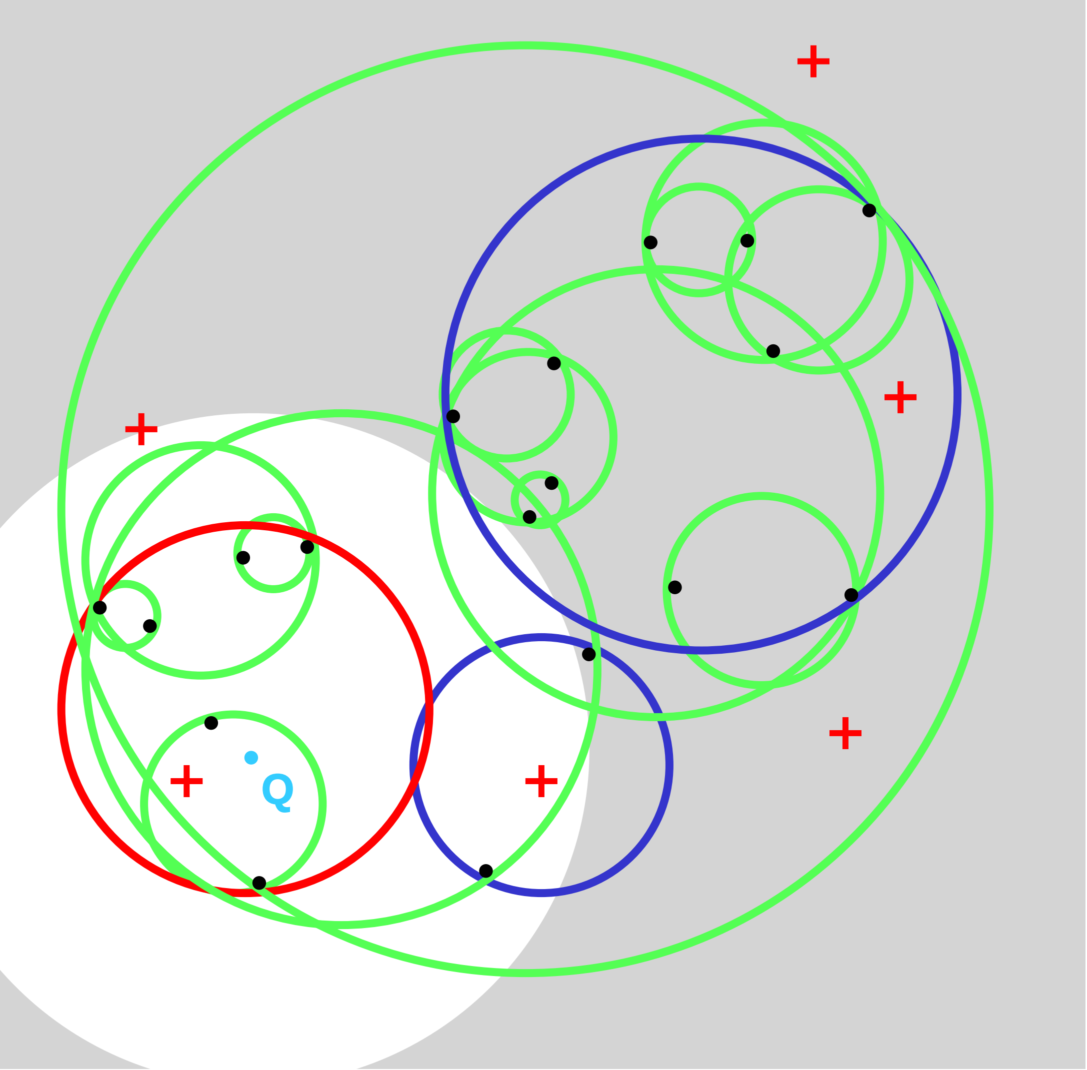

Q


Distance from Q to 3rd-closest +ve point
Max Possible Distance from Q to any –ve point in Ball

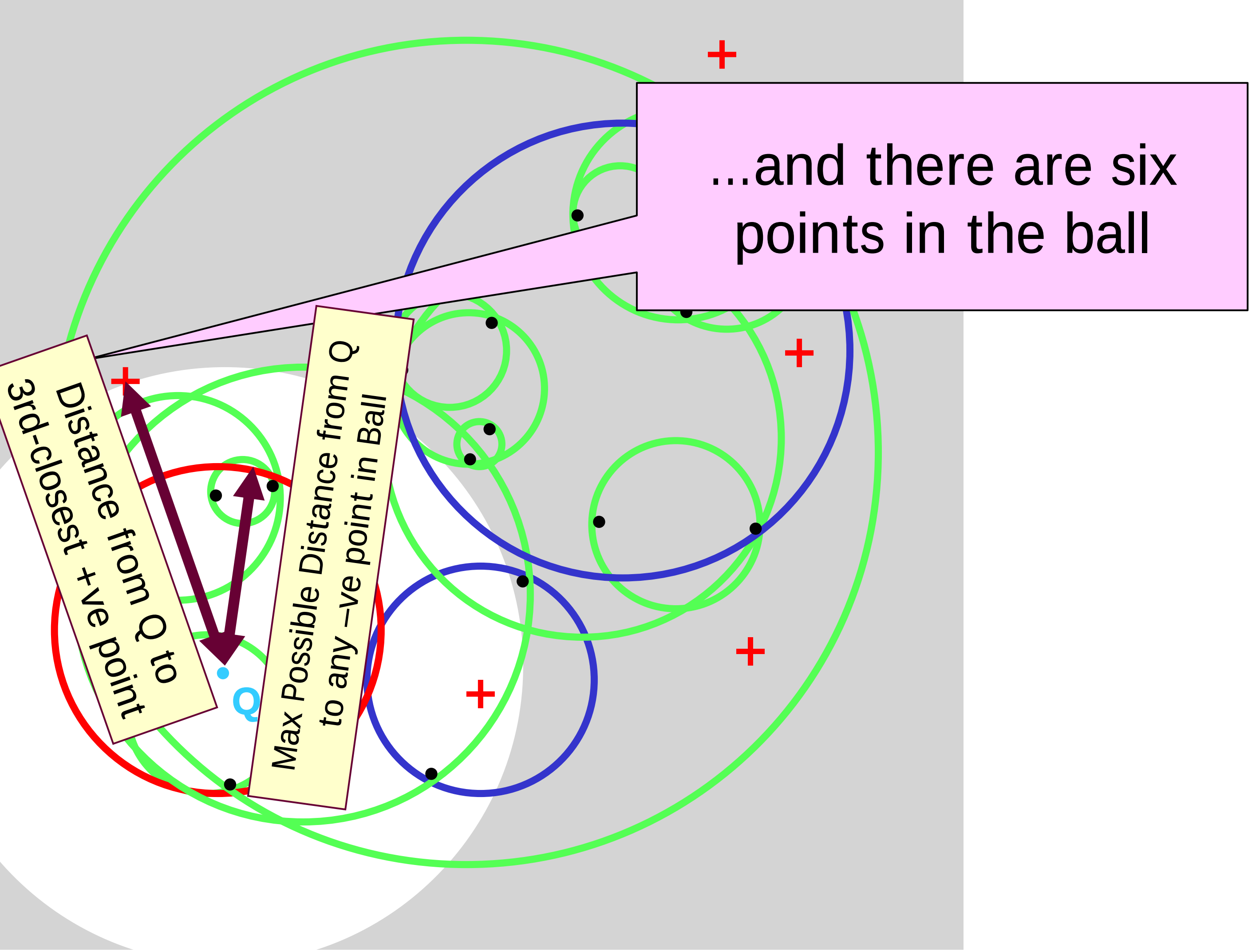
No prune!


Q

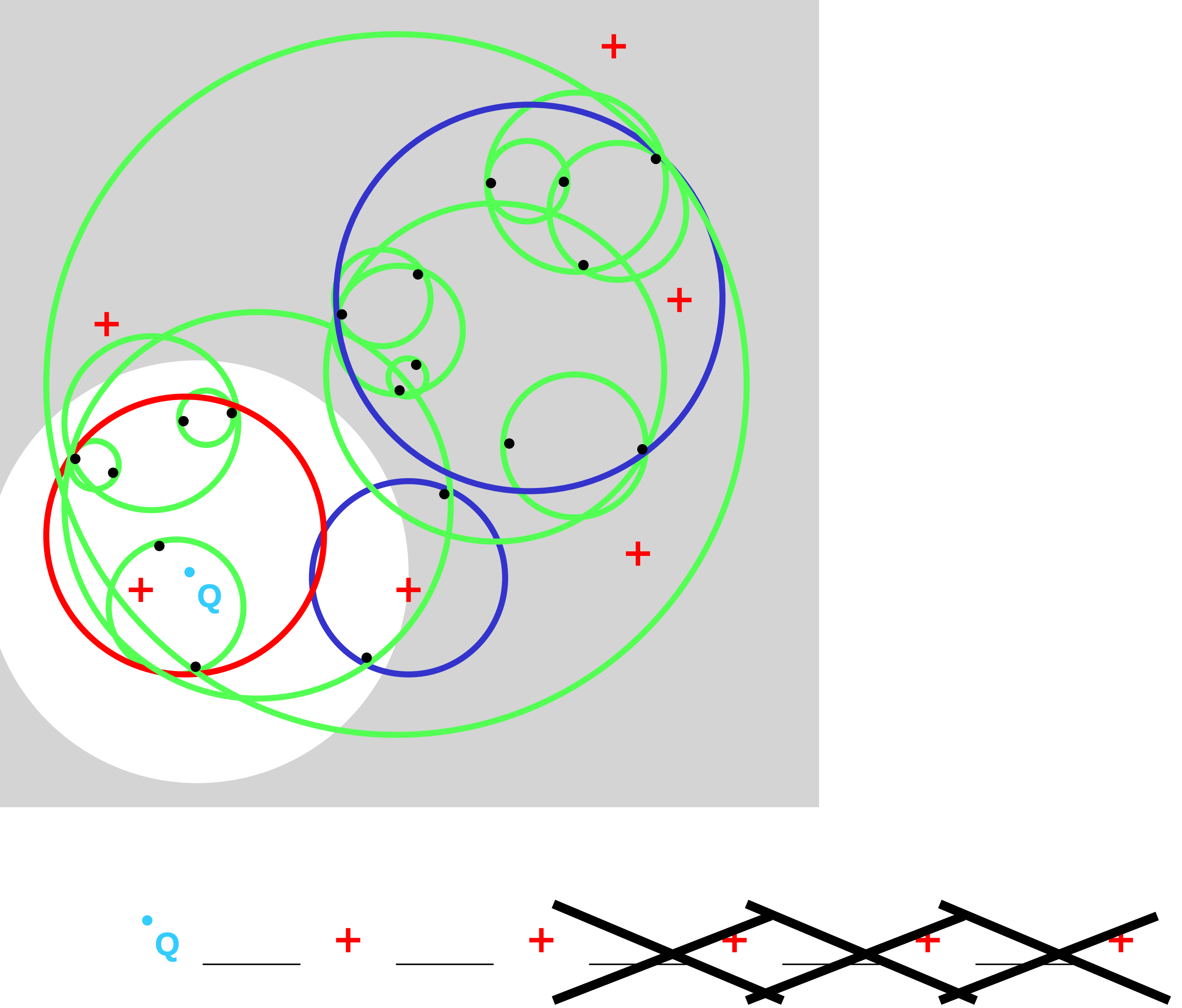

Q
Q

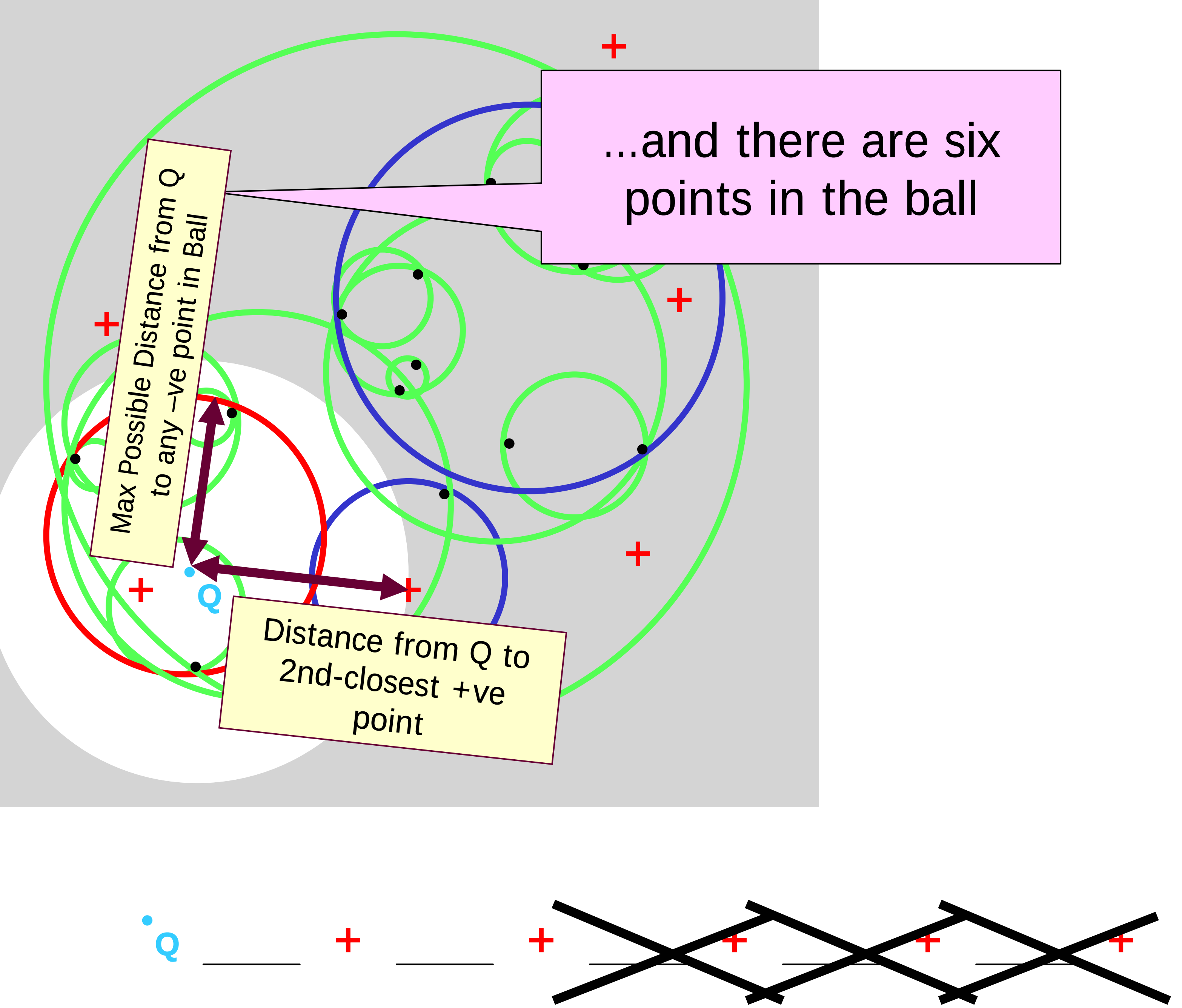

Q


Q

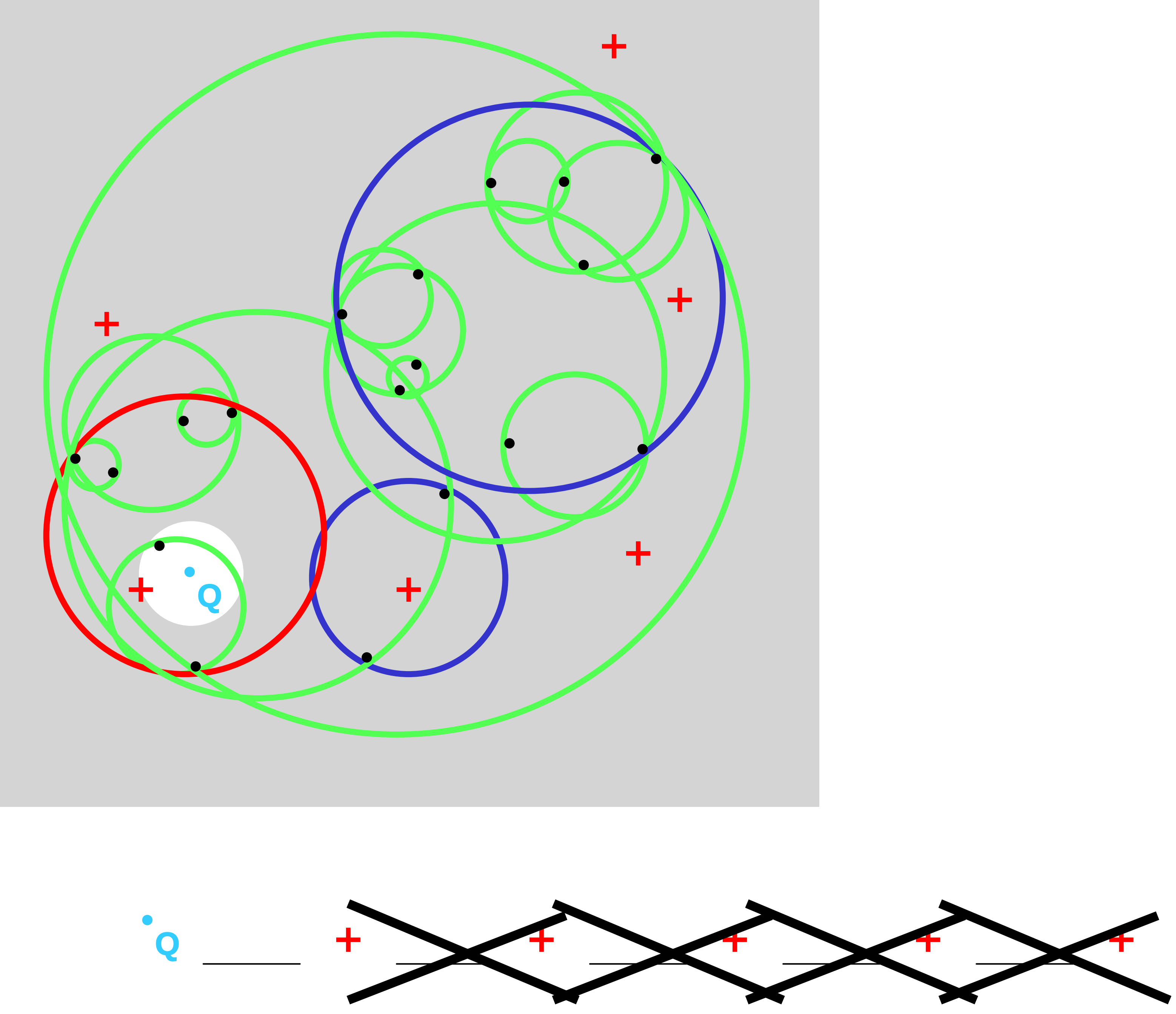

...and there are six points in the ball
Distance from Q to 3rd-closest +ve point
Max Possible Distance from Q to any –ve point in Ball
Q
+


Q
+

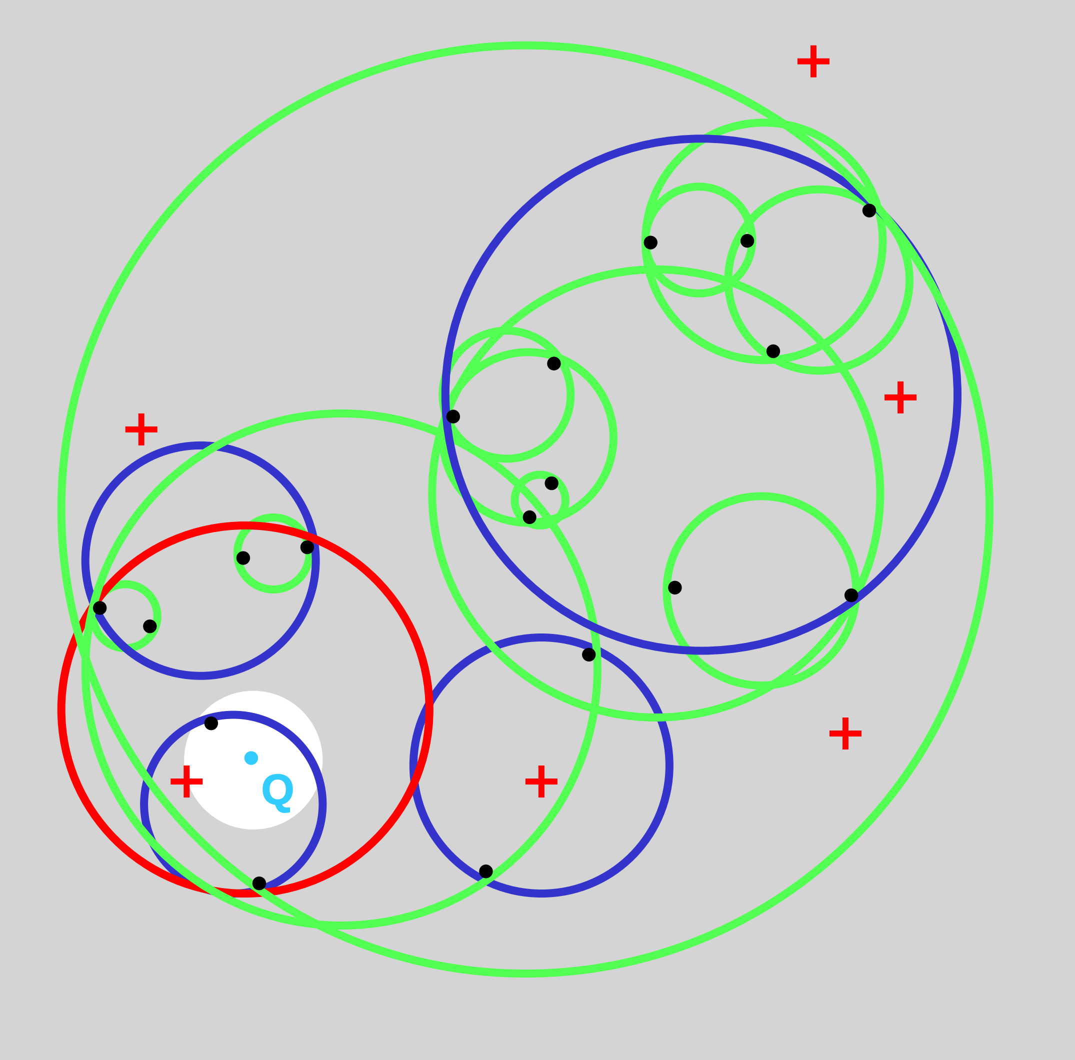

Q
Q

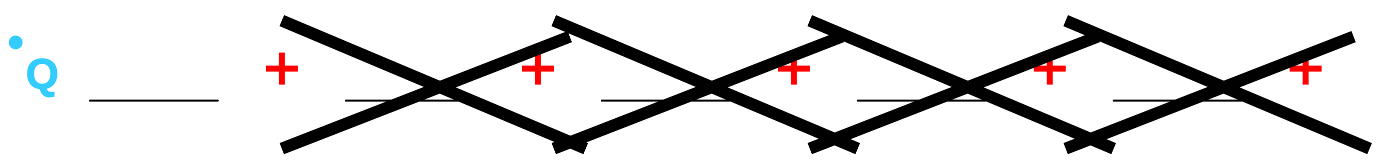


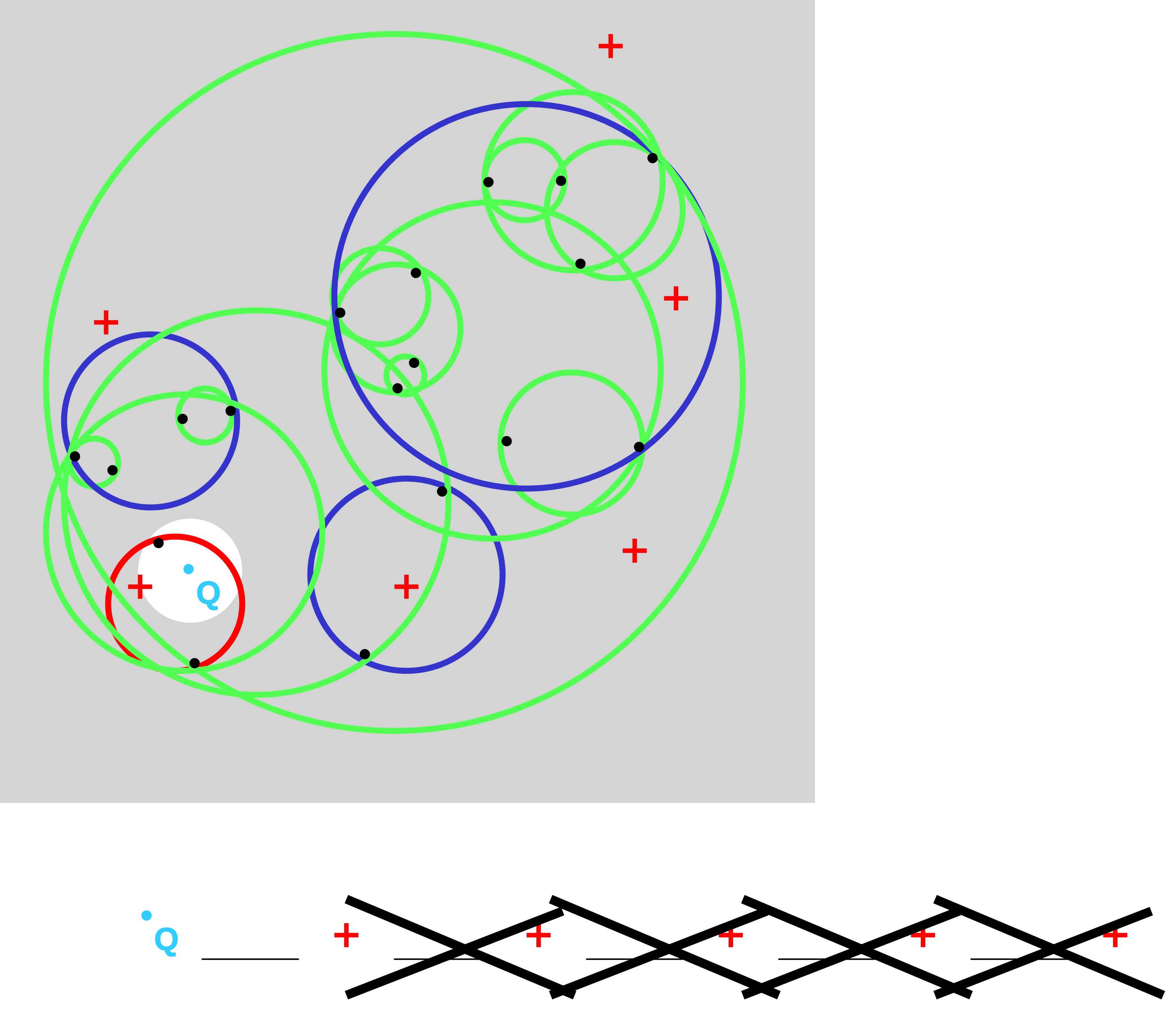
...and there are six points in the ball
Max Possible Distance from Q to any −ve point in Ball
Q
Distance from Q to 2nd-closest +ve point
+
Q

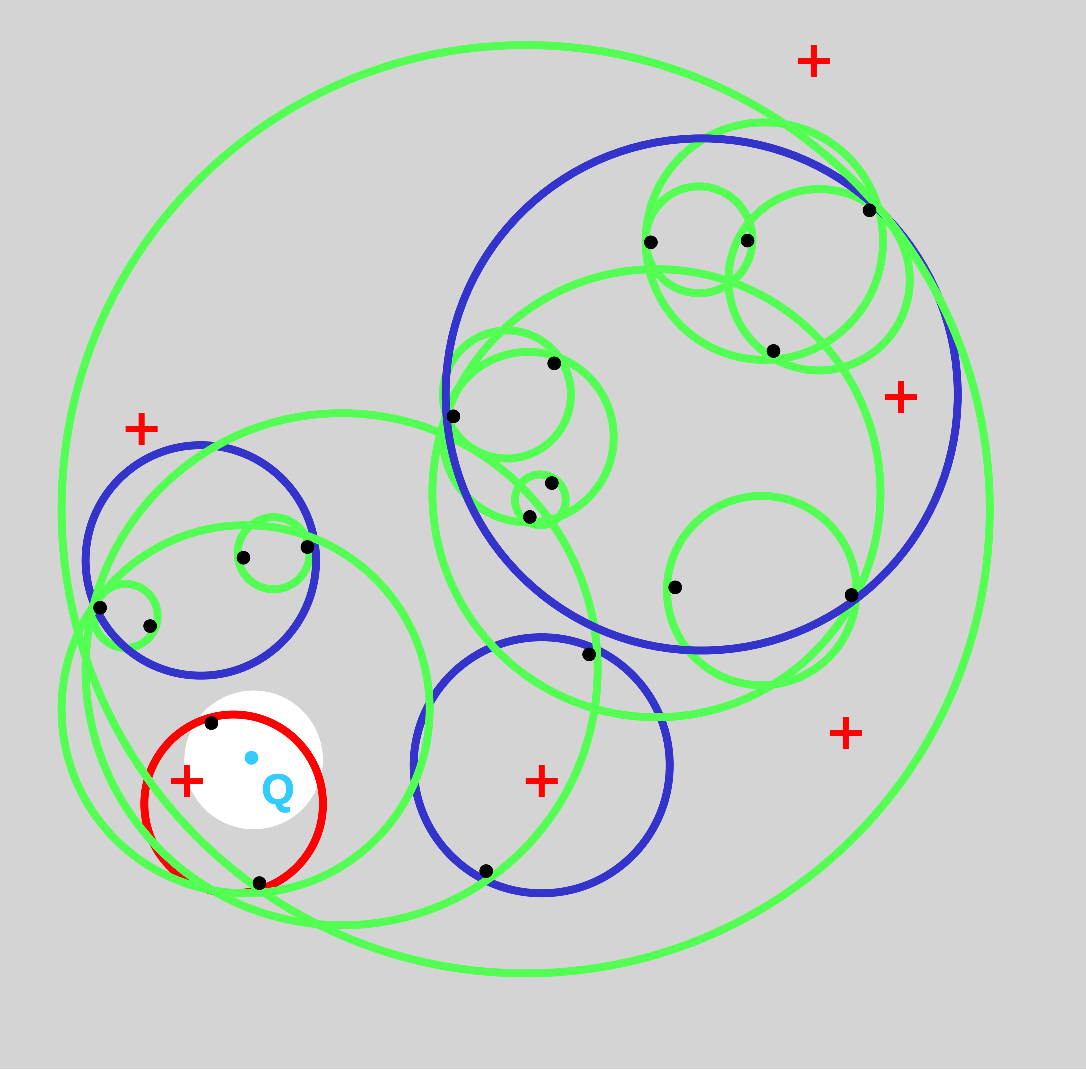


Q
Q

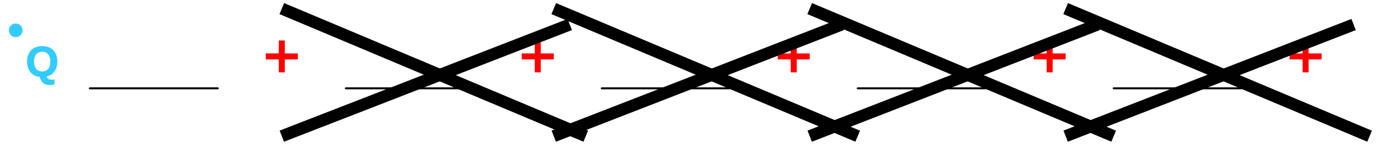

Q

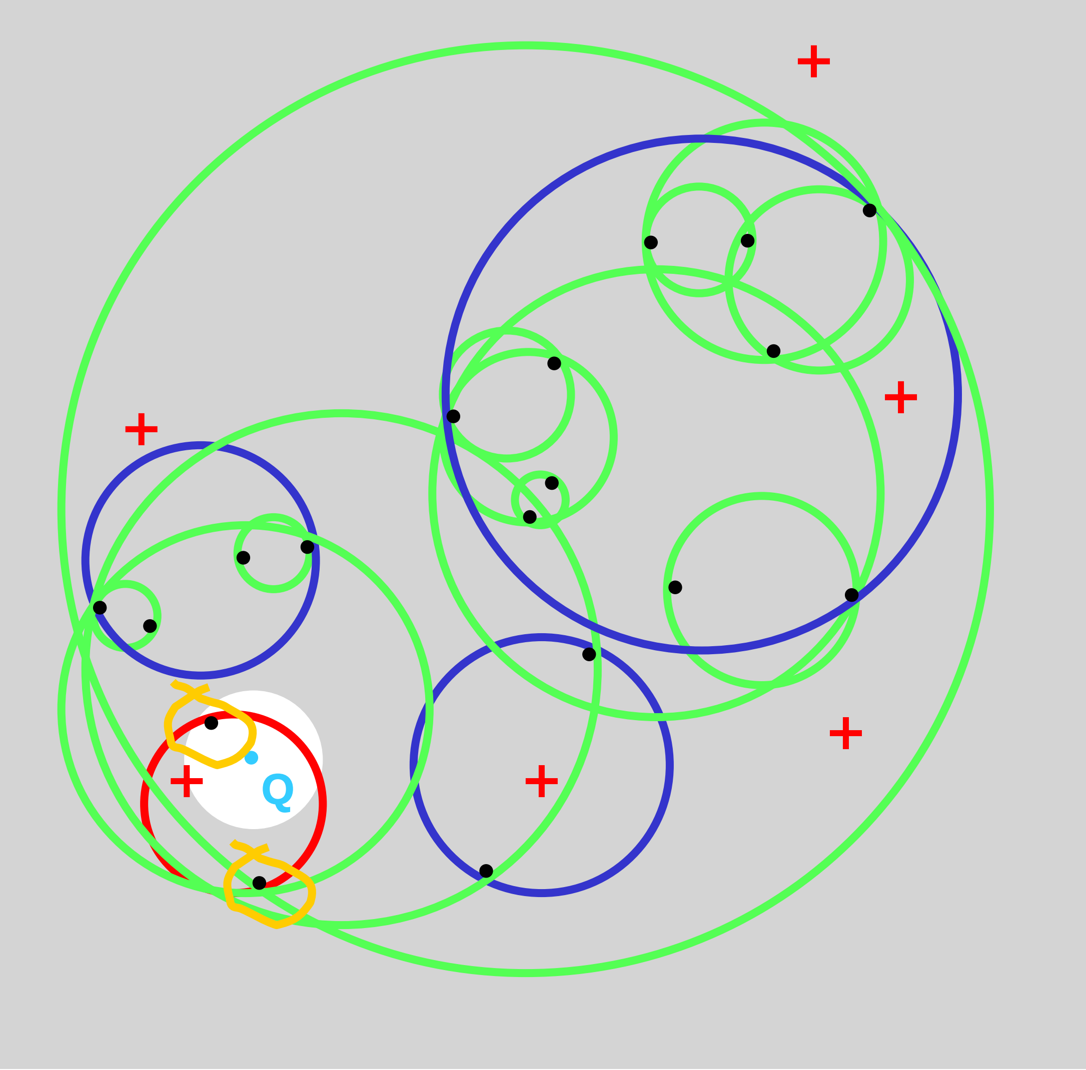

Q


Q
Q



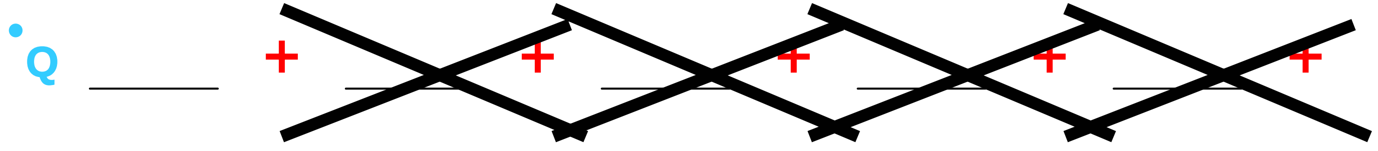
No prune.

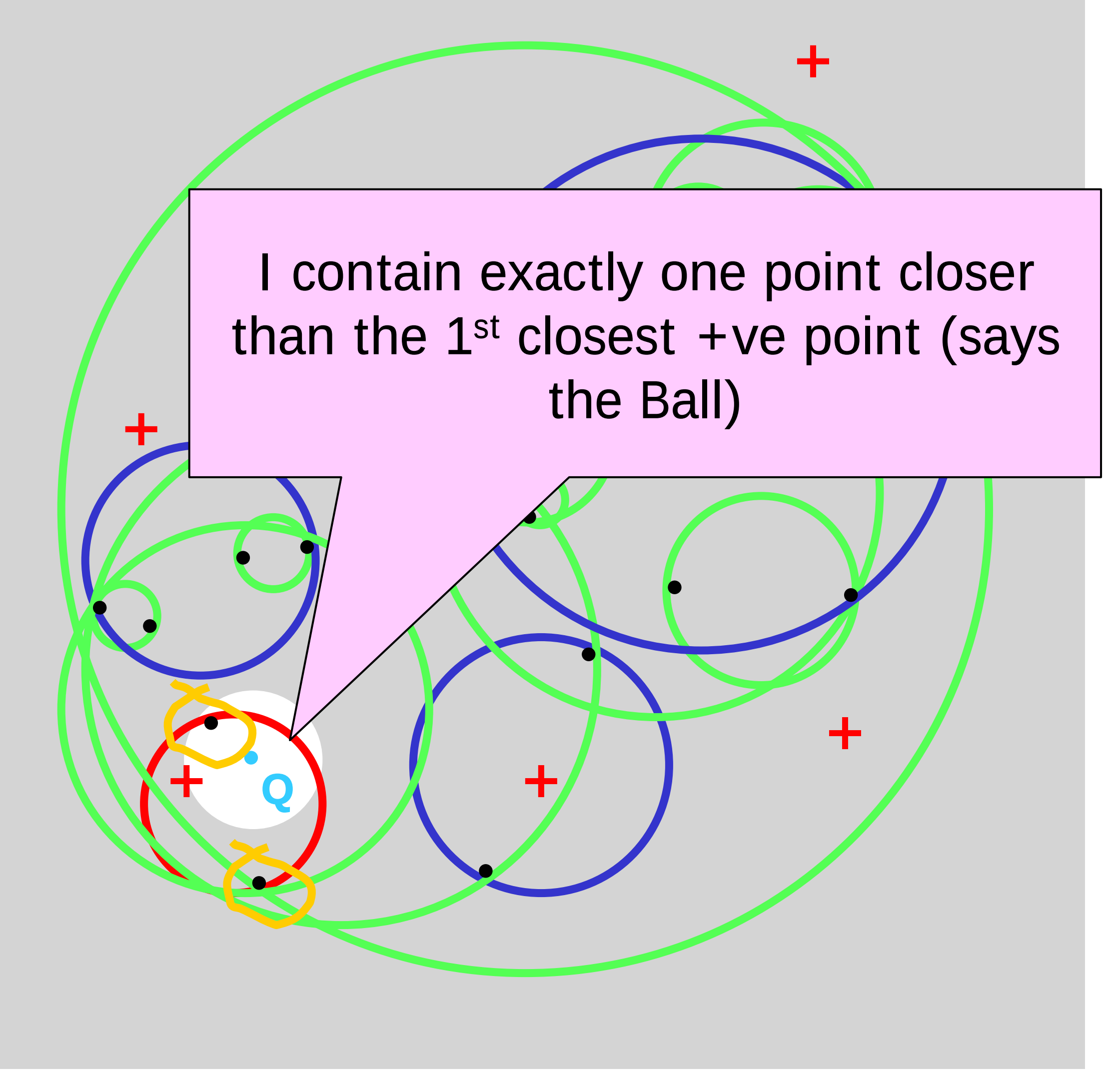




No prune.

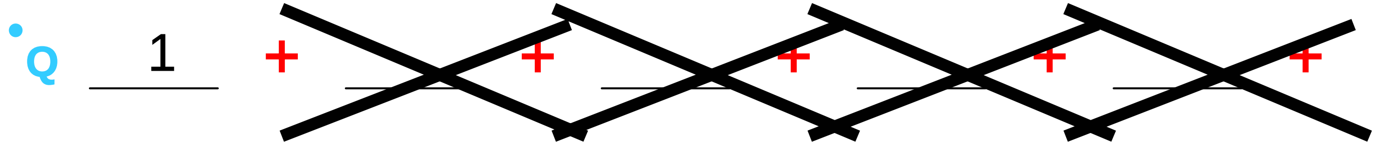
Ball is leaf so explore its points

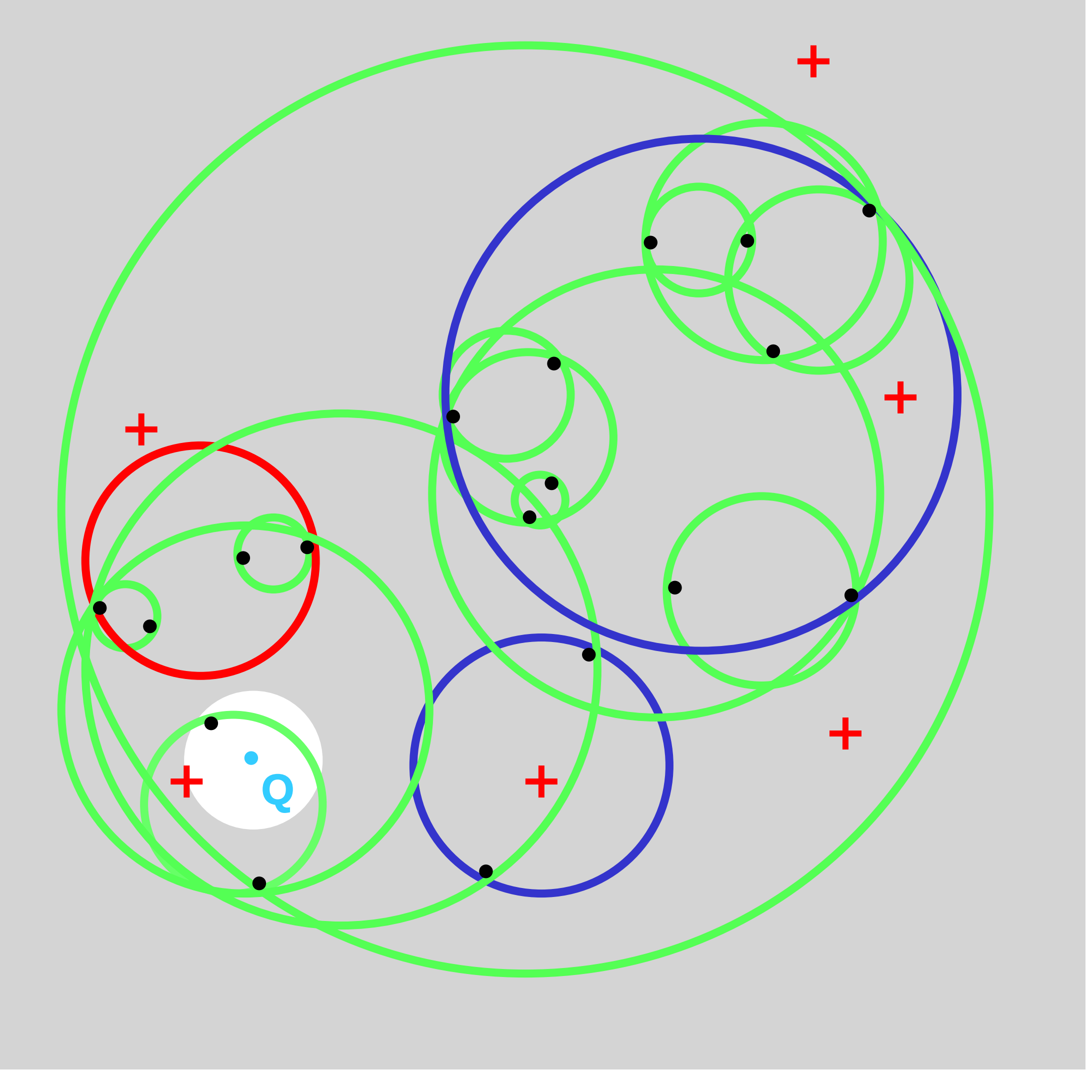




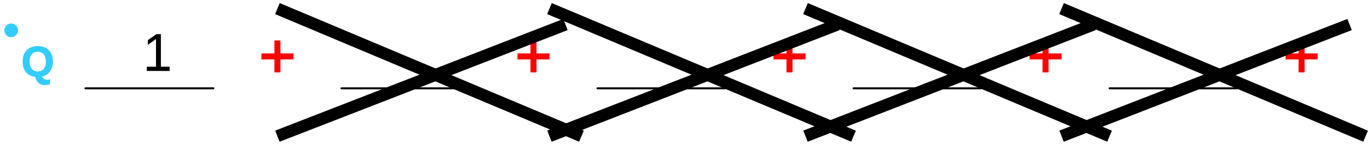
No prune.

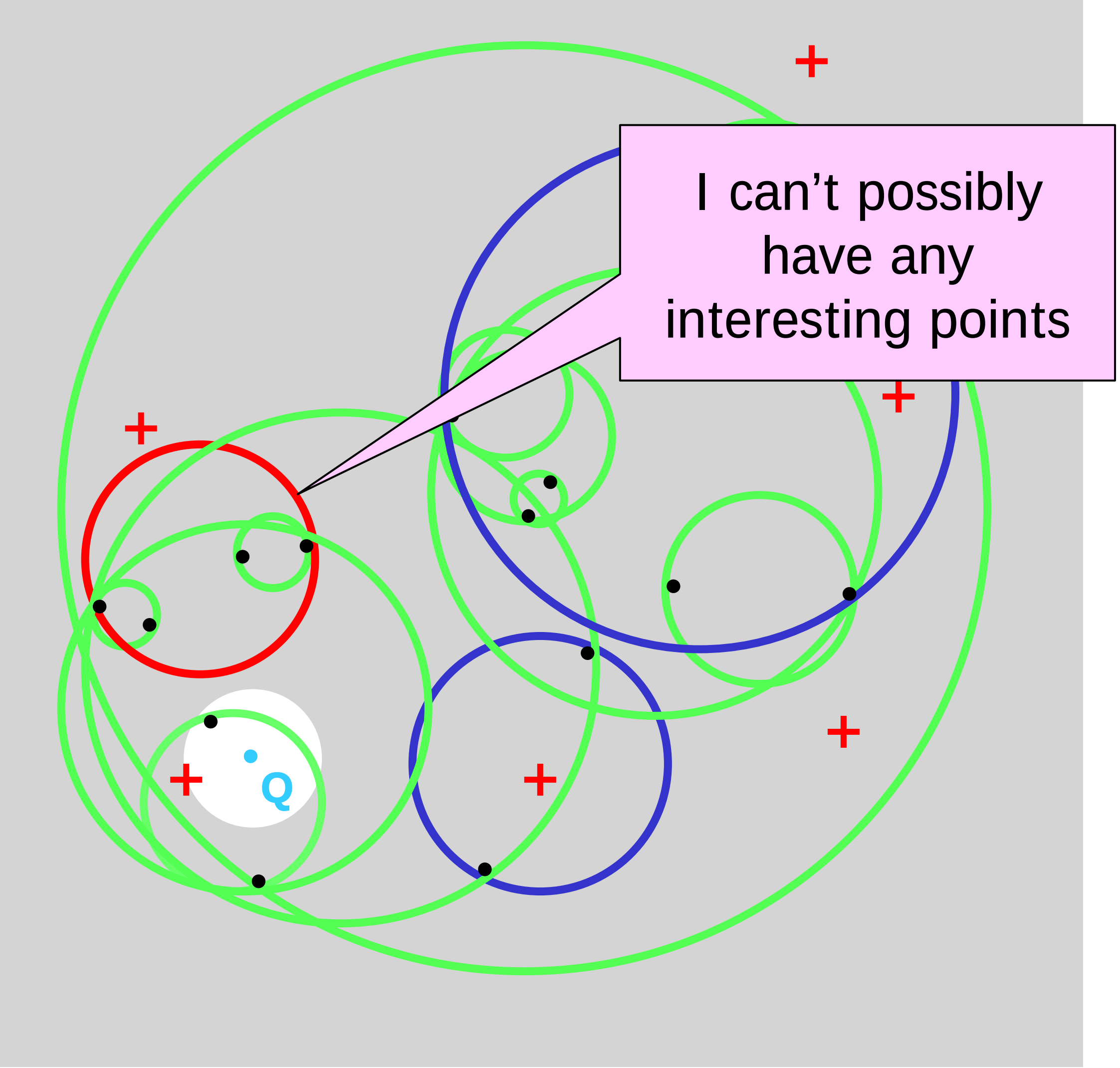
Ball is leaf so explore its points



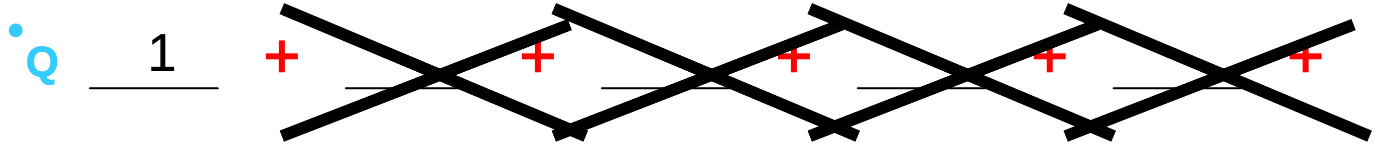


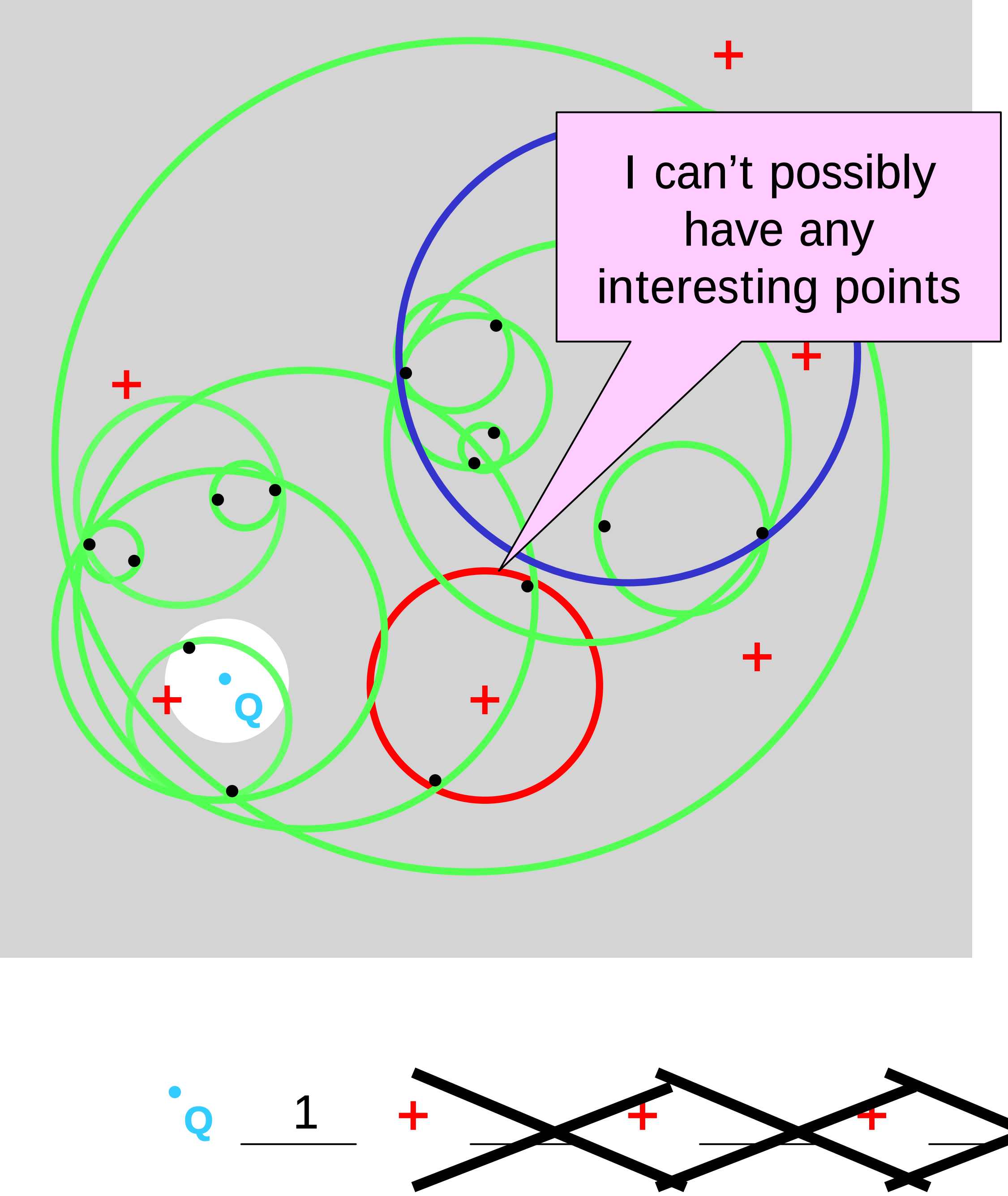
Return and try other sibling

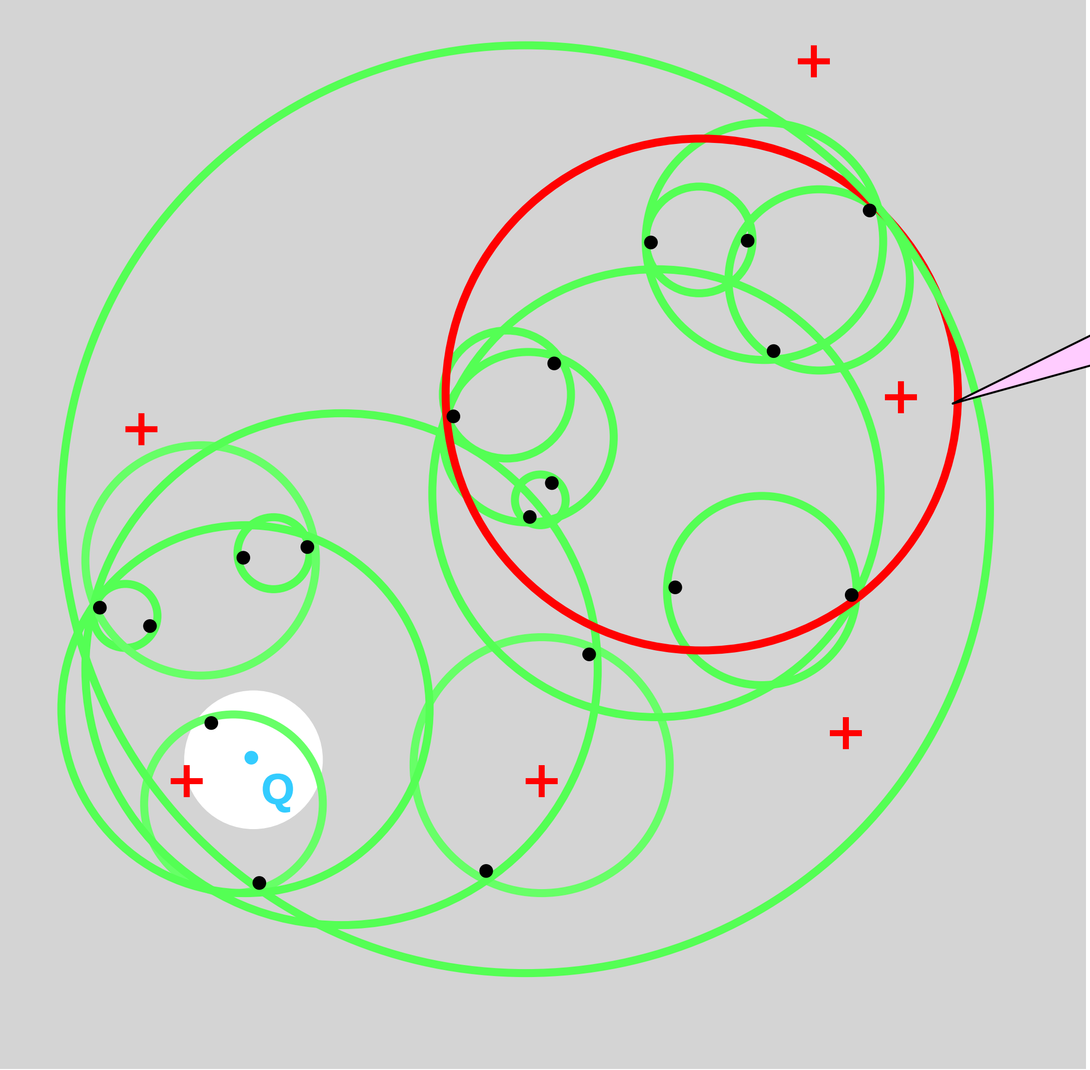




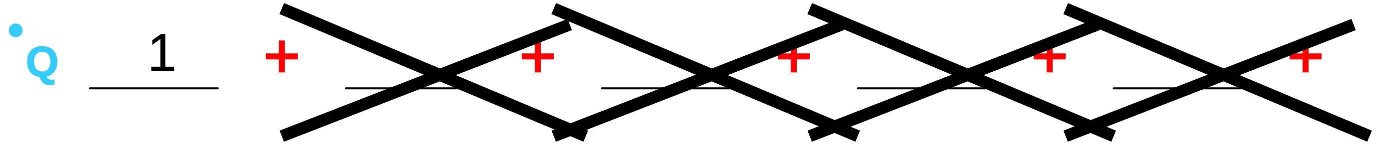
Return and try other sibling



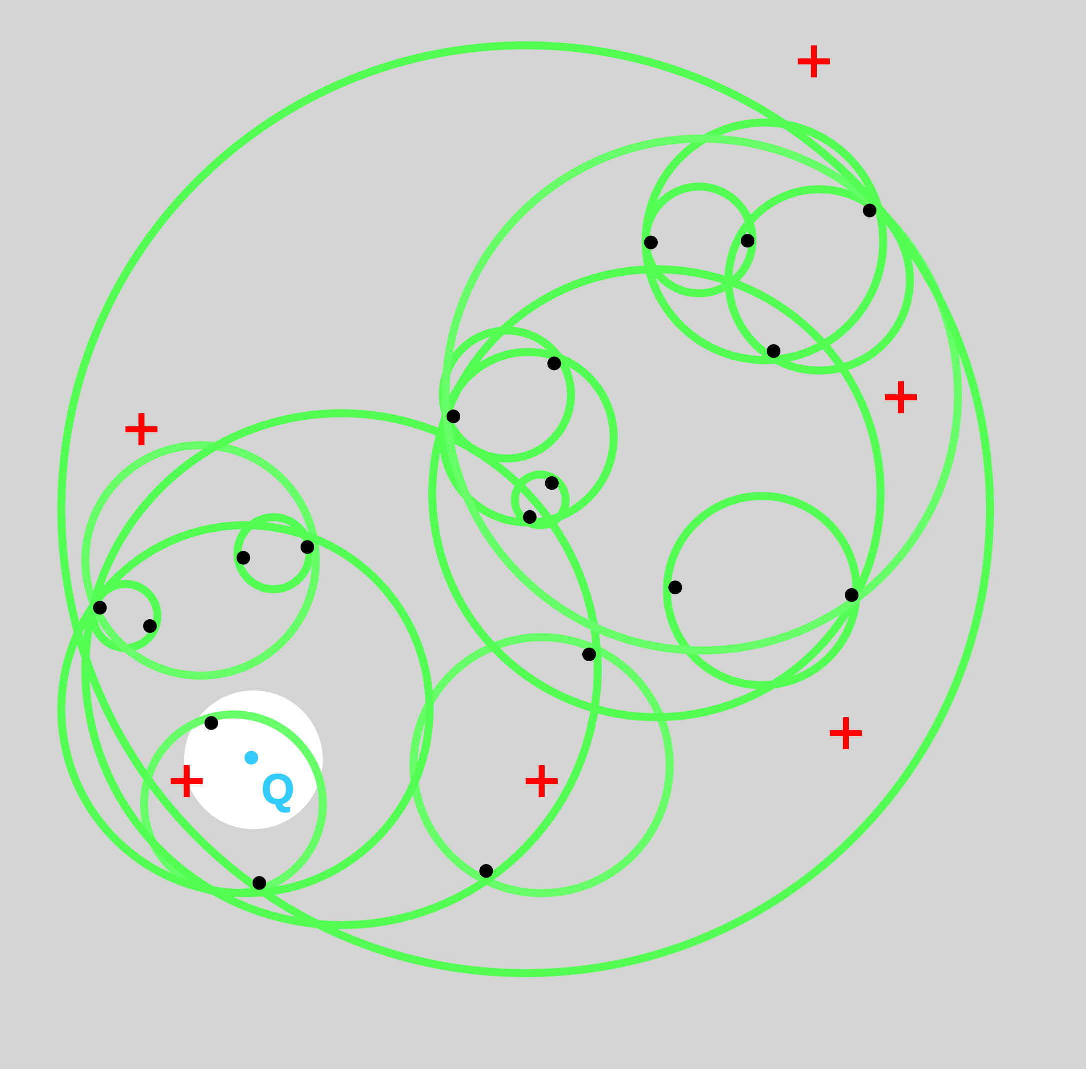

I can't possibly have any interesting points
Q
Q
1

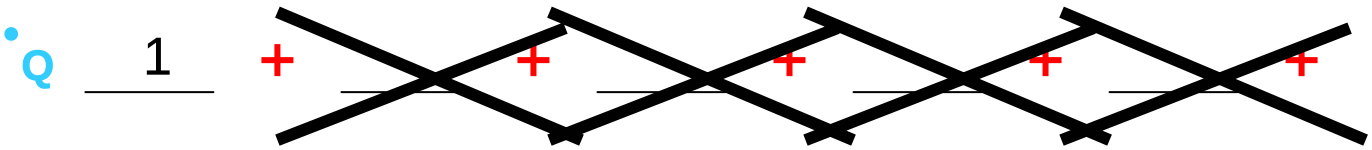


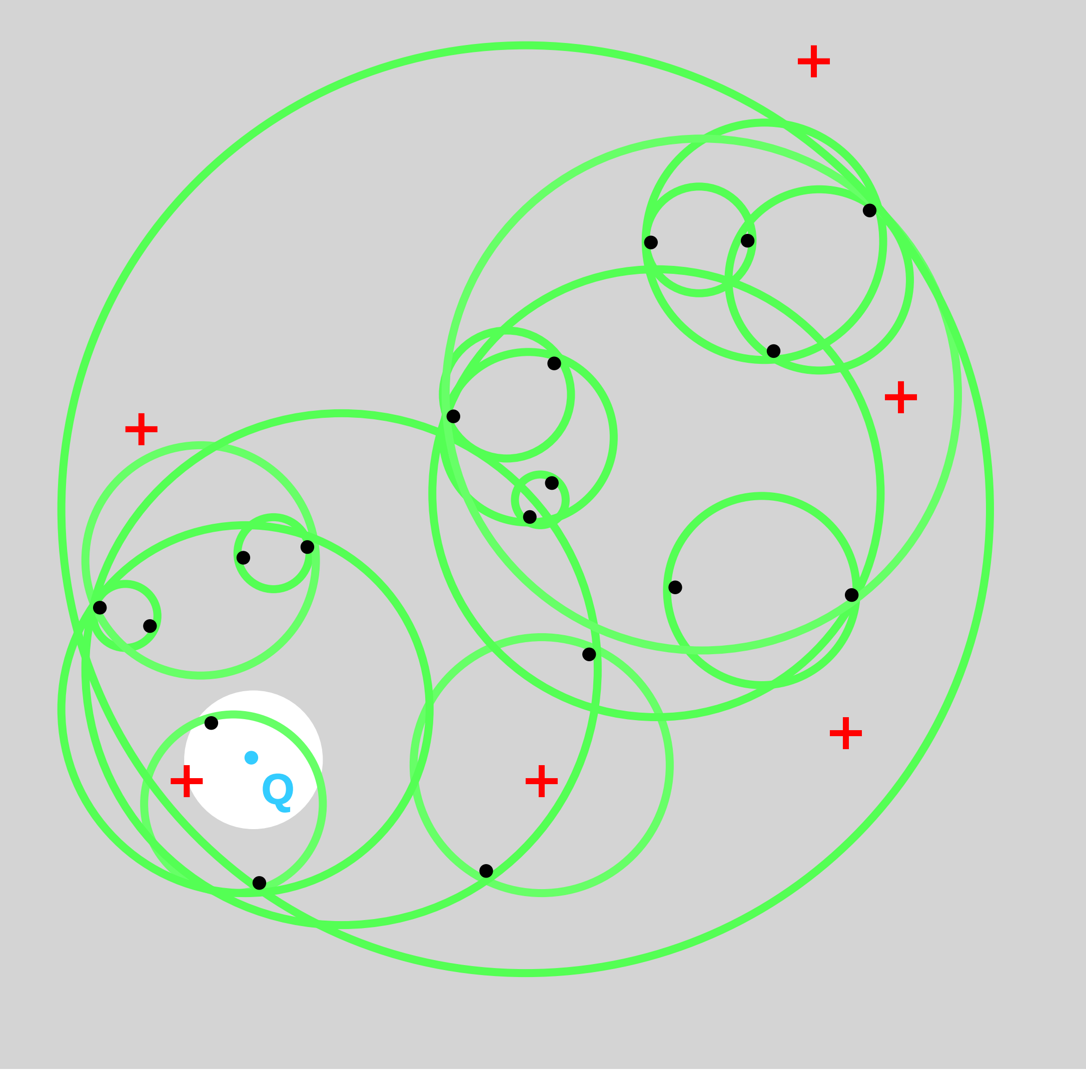
I can't possibly have any interesting points



We're done





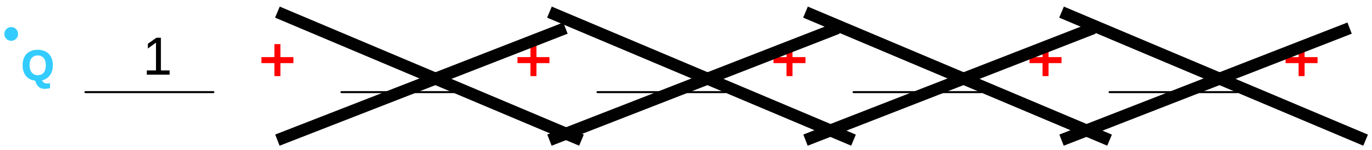


We’re done

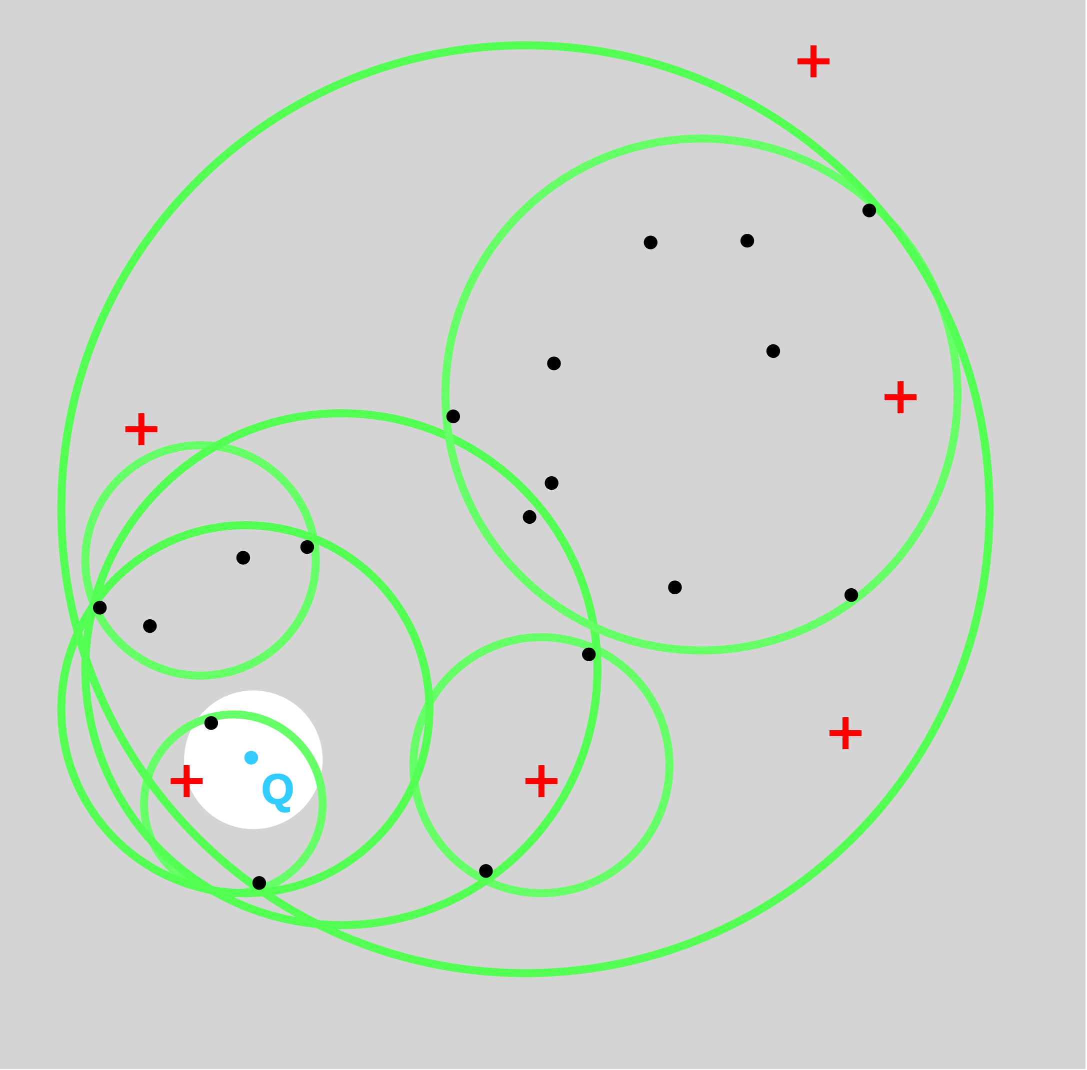
There’s one –ve point closer than the closest +ve point.

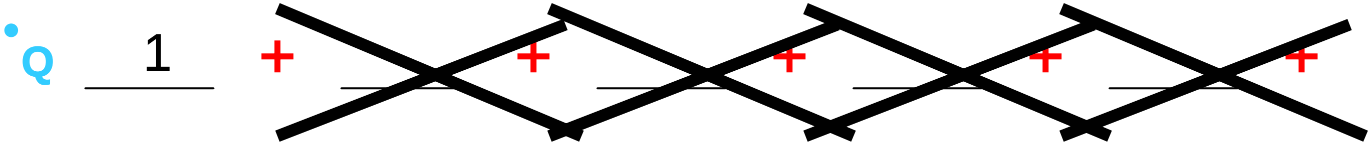
There are more than 3 –ve points closer than the 2nd closest +ve point.

=> <u>Exactly</u> 1 of the 5 nearest neighbors is +ve

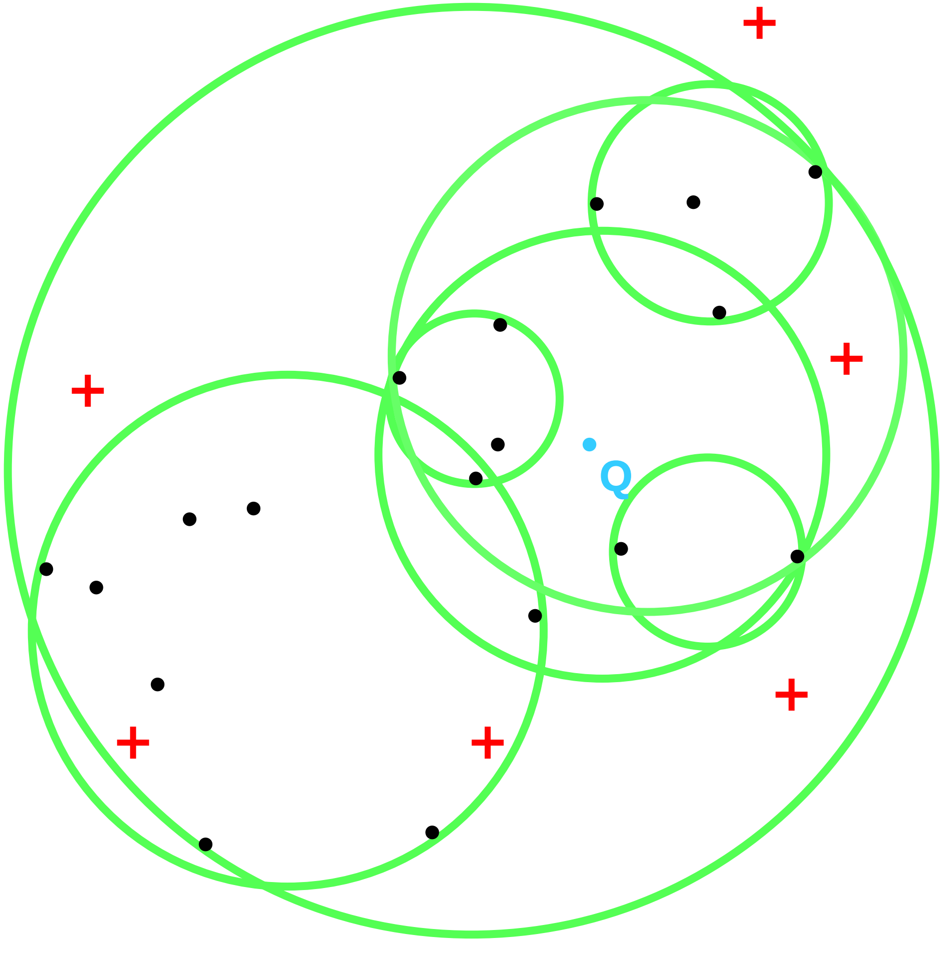


Balls visited





Another example

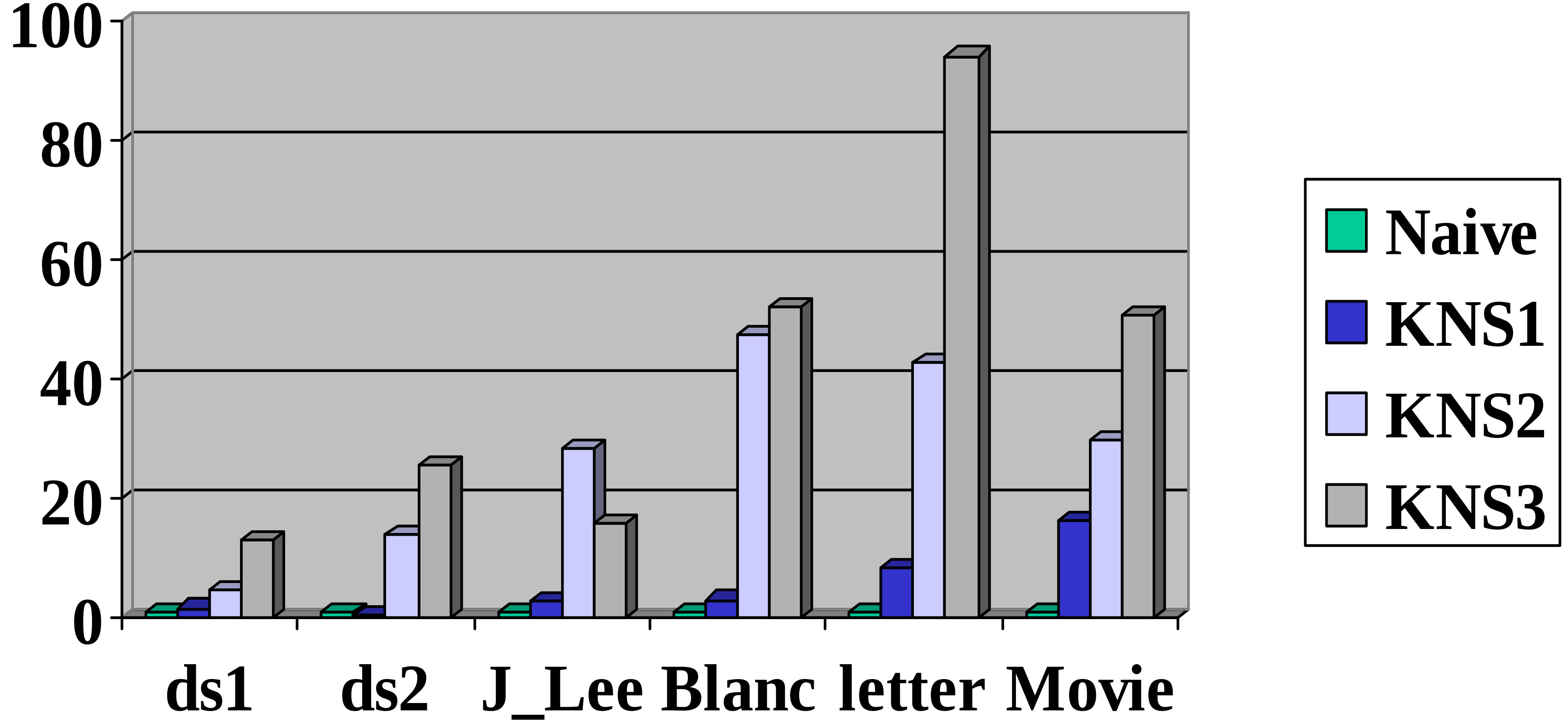


Experimental results

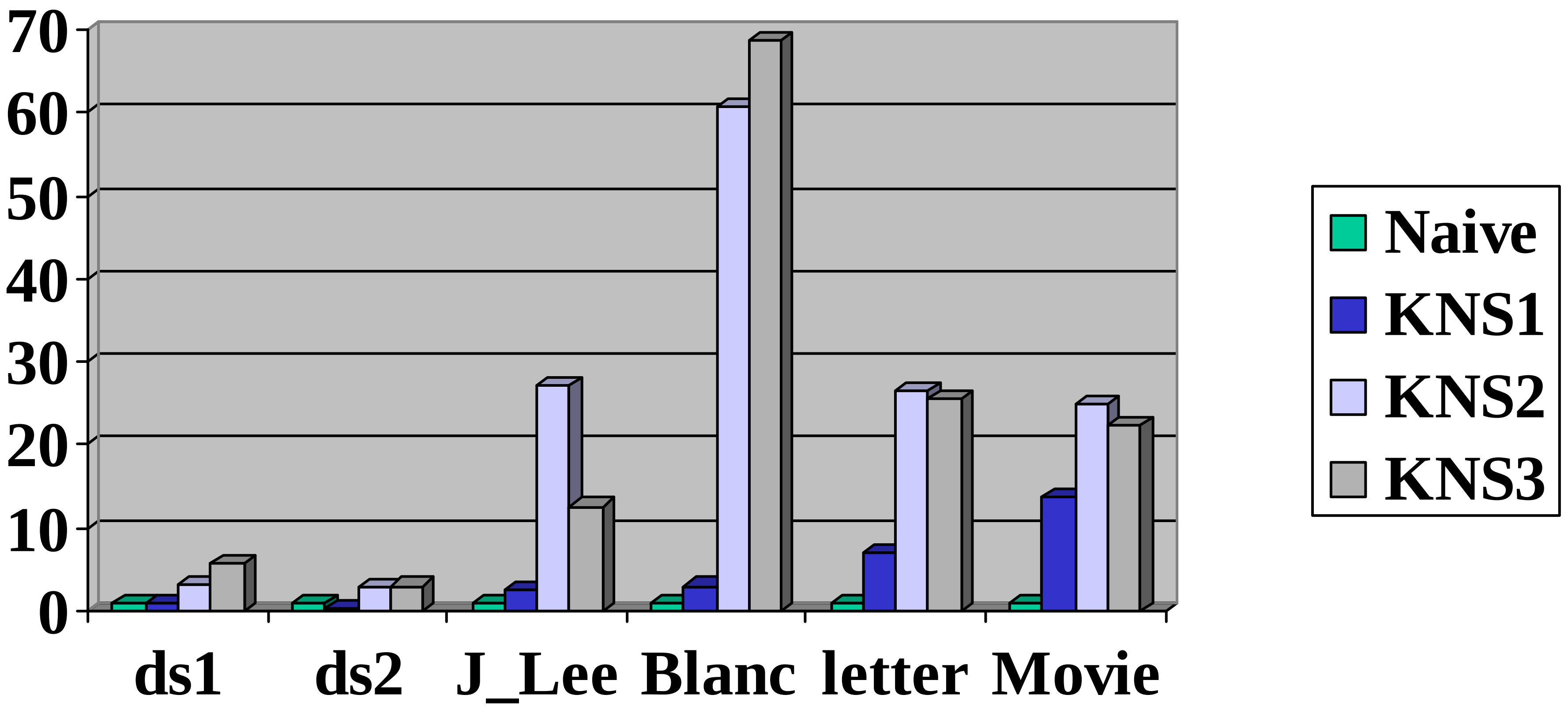
Dataset	Num. of records	Num. of Dimensions	Num.of positive	Num.pos/Num.neg
ds1	26733	6348	804	0.03
ds1.10pca	26733	10	804	0.03
ds1.100pca	26733	100	804	0.03
ds2	88358	1.1×10^6	211	0.002
ds2.100anchor	88358	100	211	0.002
J_Lee.100pca	181395	100	299	0.0017
Blanc__Mel	186414	10	824	0.004

Dataset	Num. records	Num. of Dimensions	Num.of positive	Num.pos/Num.neg
Letter	20000	16	790	0.04
Ipums	70187	60	119	0.0017
Movie	38943	62	7620	0.24
Kdd99(10%)	494021	176	97278	0.24

Num of Distance computations *Speedup* for K-NN



Wall-clock-time speedup for k-NN

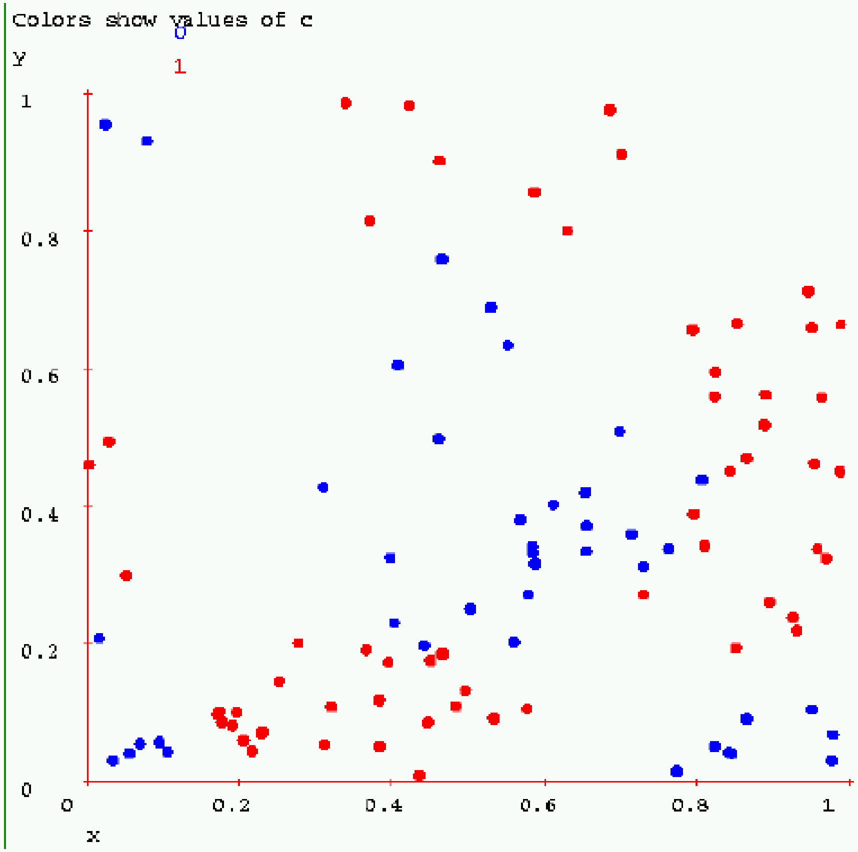


		NAIVE		KNS1		KNS2		KNS3	
		dists	time (secs)	dists speedup	time speedup	dists speedup	time speedup	dists speedup	time speedup
ideal	k=9	9.0×10^7	30	96.7	56.5	112.9	78.5	4500	486
	k=101			23.0	10.2	24.7	14.7	4500	432
Diag2d(10%)	k=9	9.0×10^7	30	91	51.1	88.2	52.4	282	27.1
	k=101			22.3	8.7	21.3	9.3	167.9	15.9
Diag2d	k=9	9.0×10^9	3440	738	366	664	372	2593	287
	k=101			202.9	104	191	107.5	2062	287
Diag3d	k=9	9.0×10^9	4060	361	184.5	296	184.5	1049	176.5
	k=101			111	56.4	95.6	48.9	585	78.1
Diag10d	k=9	9.0×10^9	6080	7.1	5.3	7.3	5.2	12.7	2.2
	k=101			3.3	2.5	3.1	1.9	6.1	0.7
Noise2d	k=9	9.0×10^7	40	91.8	20.1	79.6	30.1	142	42.7
	k=101			22.3	4	16.7	4.5	94.7	43.5
ds1	k=9	6.4×10^8	4830	1.6	1.0	4.7	3.1	12.8	5.8
	k=101			1.0	0.7	1.6	1.1	10	4.2
ds1.10pca	k=9	6.4×10^8	420	11.8	11.0	33.6	21.4	71	20
	k=101			4.6	3.4	6.5	4.0	40	6.1
ds1.100pca	k=9	6.4×10^8	2190	1.7	1.8	7.6	7.4	23.7	29.6
	k=101			0.97	1.0	1.6	1.6	16.4	6.8
ds2	k=9	8.5×10^9	105500	0.64	0.24	14.0	2.8	25.6	3.0
	k=101			0.61	0.24	2.4	0.83	28.7	3.3
ds2.100-	k=9	7.0×10^9	24210	15.8	14.3	185.3	144	580	311
	k=101			10.9	14.3	23.0	19.4	612	248
J_Lee.100-	k=9	3.6×10^{10}	142000	2.6	2.4	28.4	27.2	15.6	12.6
	k=101			2.2	1.9	12.6	11.6	37.4	27.2
Blanc_Mel	k=9	3.8×10^{10}	44300	3.0	3.0	47.5	60.8	51.9	60.7
	k=101			2.9	3.1	7.1	33	203	134.0
Letter	k=9	3.6×10^8	290	8.5	7.1	42.9	26.4	94.2	25.5
	k=101			3.5	2.6	9.0	5.7	45.9	9.4
Ipums	k=9	4.4×10^9	9520	195	136	665	501	1003	515
	k=101			69.1	50.4	144.6	121	5264	544
Movie	k=9	1.4×10^9	3100	16.1	13.8	29.8	24.8	50.5	22.4
	k=101			9.1	7.7	10.5	8.1	33.3	11.6
Kddcup99	k=9	2.7×10^{11}	1670000	4.2	4.2	574	702	4	4.1
(10%)	k=101			4.2	4.2	187.7	226.2	3.9	3.9

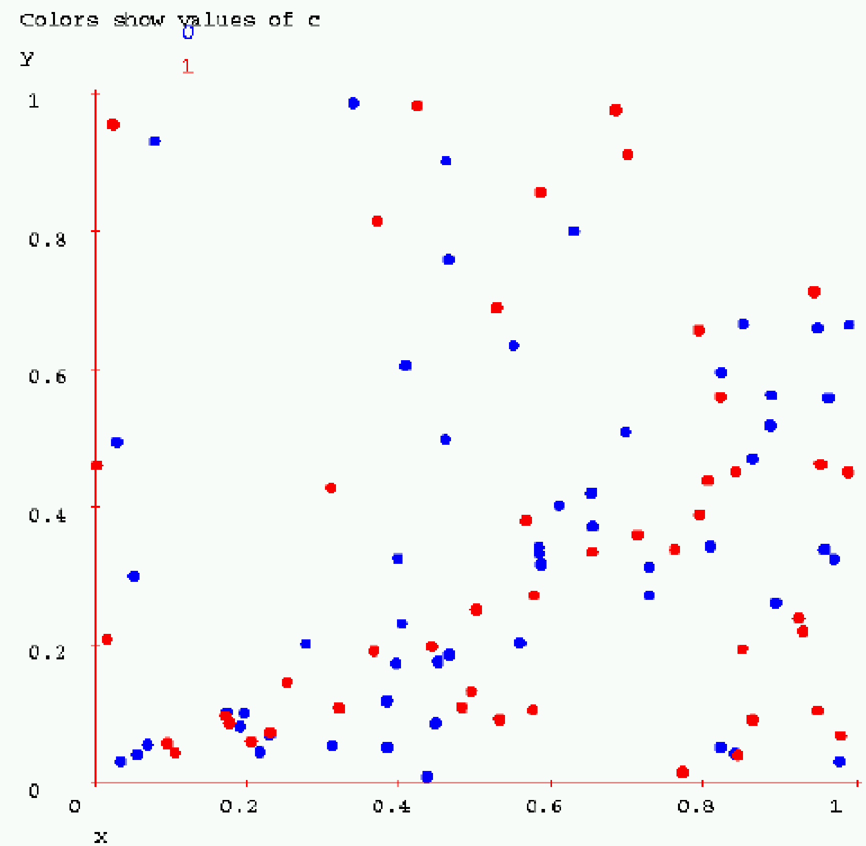
Outline

- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
    - Very fast non-parametric classification
    - skewed binary outputs
    - General binary outputs
    - multi-classed outputs
- Very fast kernel-based statistics
    - n-point computations
    - clustering
    - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

All-pairs-of-points problems in statistics

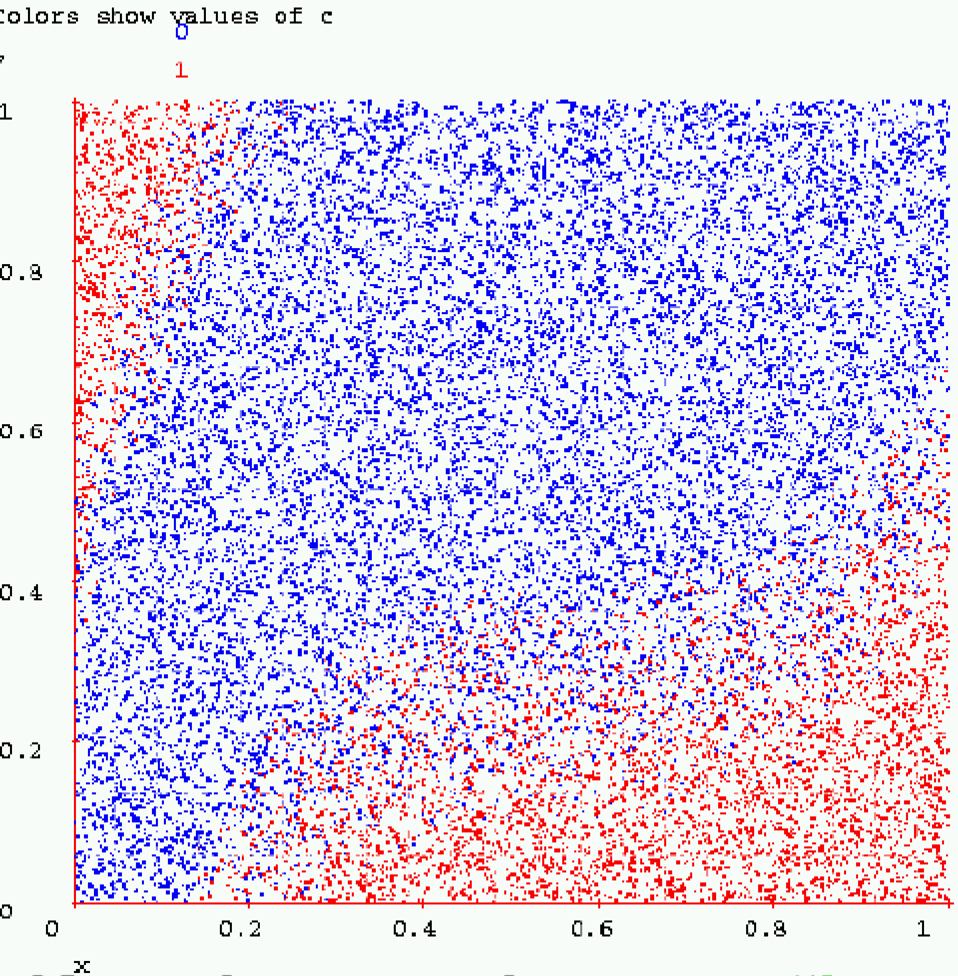


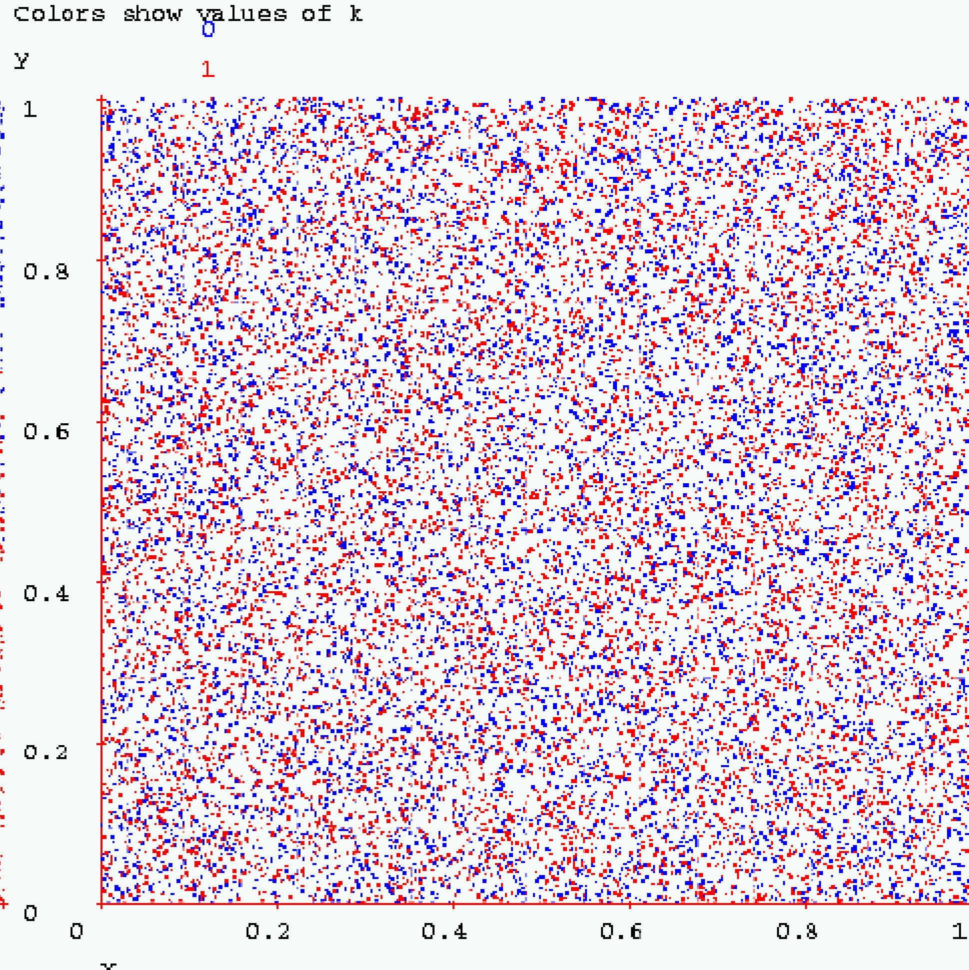
Leave-one-out testing with one nearest neighbor says "I can see something interesting"



Leave-one-out testing with one nearest neighbor says "I can see nothing interesting"

Kernel Regression

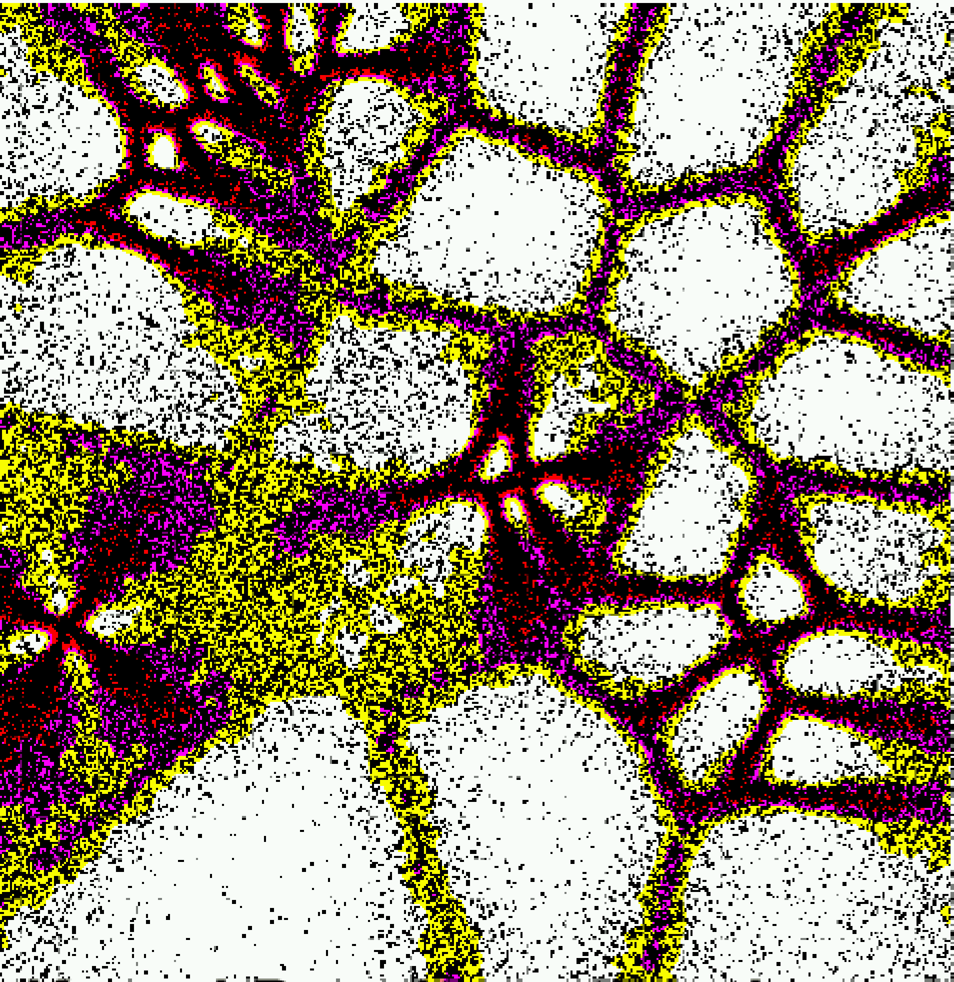




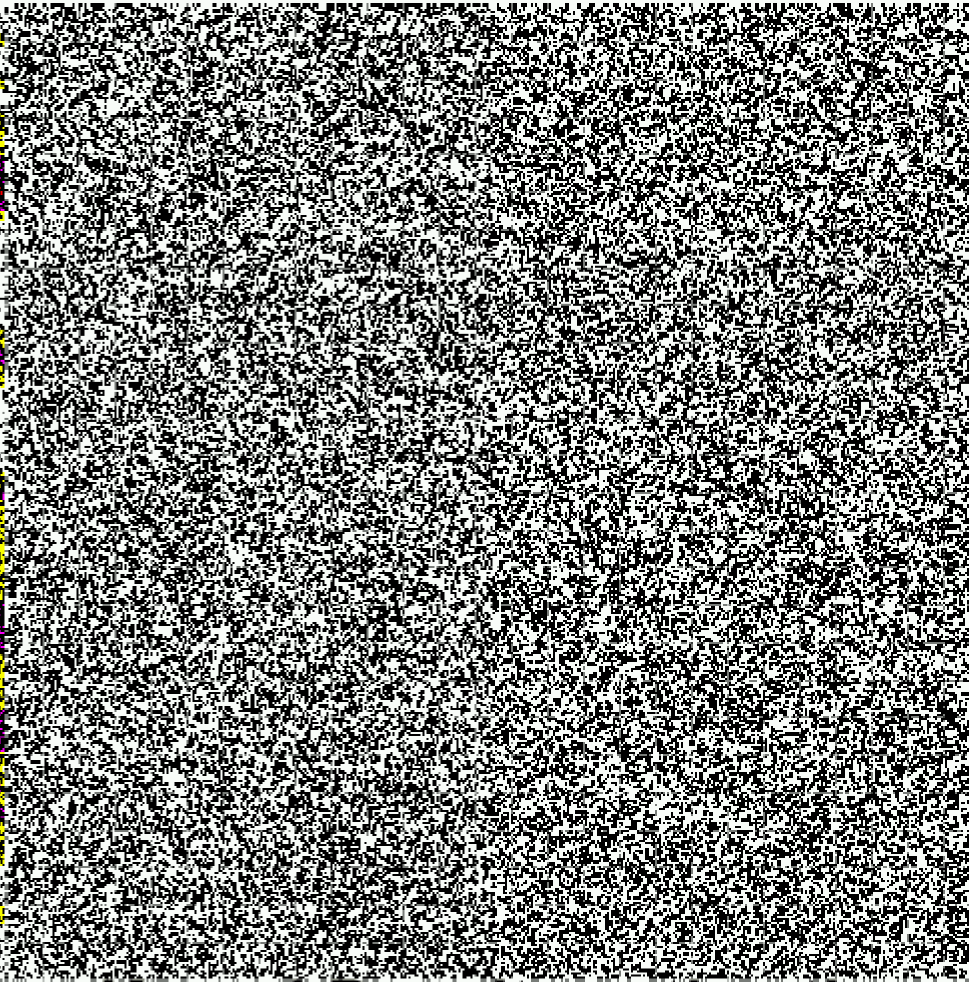
Kernel regression says "I can see something interesting"

Kernel Regression says "I can see nothing interesting"

Kernel Density Estimation

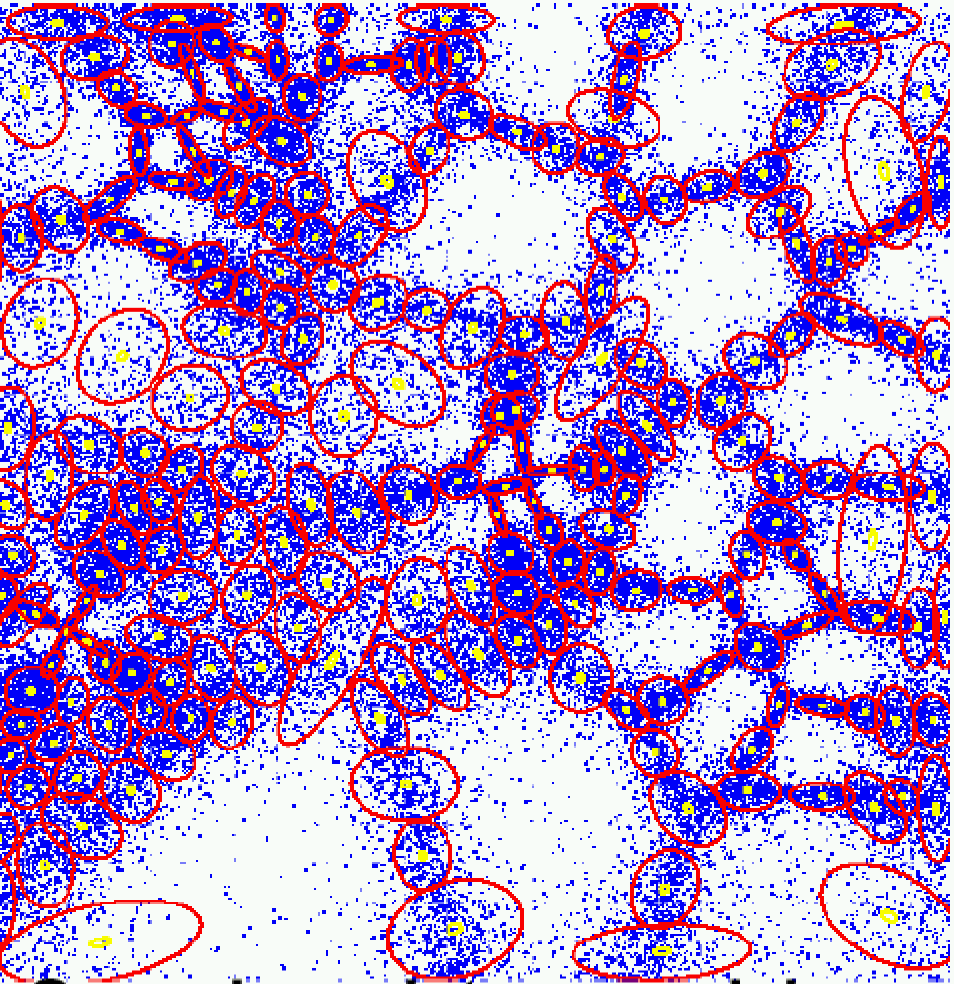


Kernel Density estimation say "I can see something interesting"

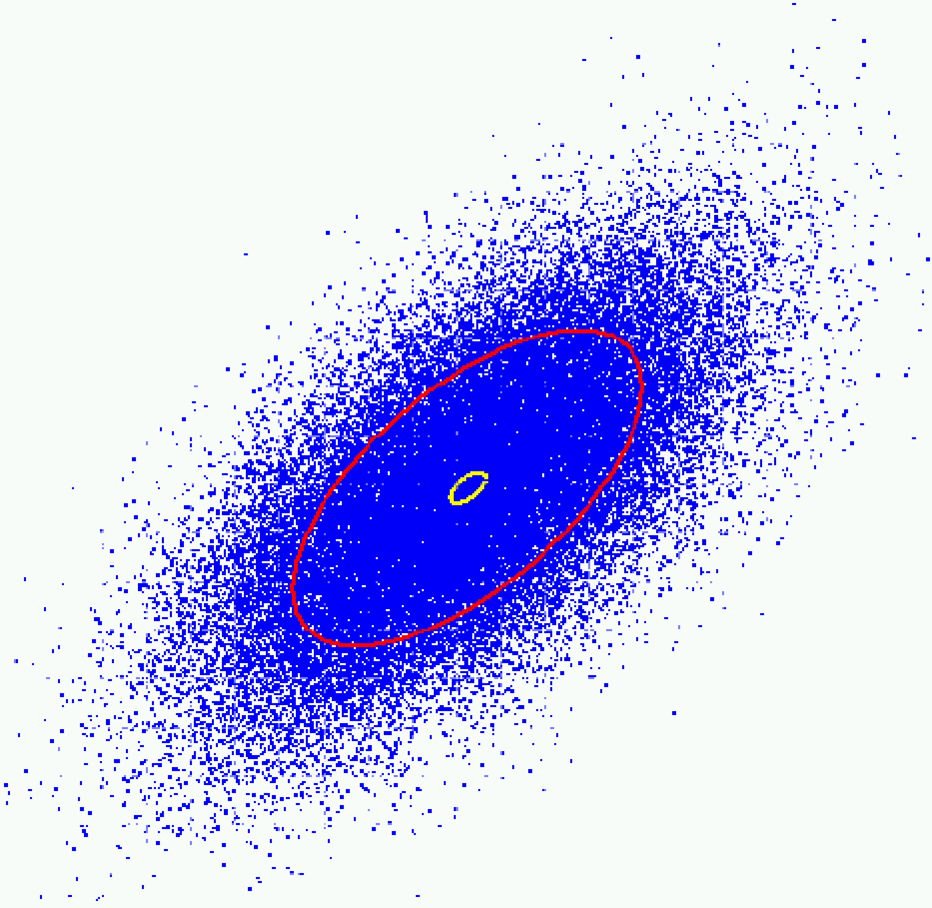


Kernel Density estimation says "I can see nothing interesting"

Many-component Mixture Models



Gaussian mixture model says "I can see and estimate a great deal of structure"

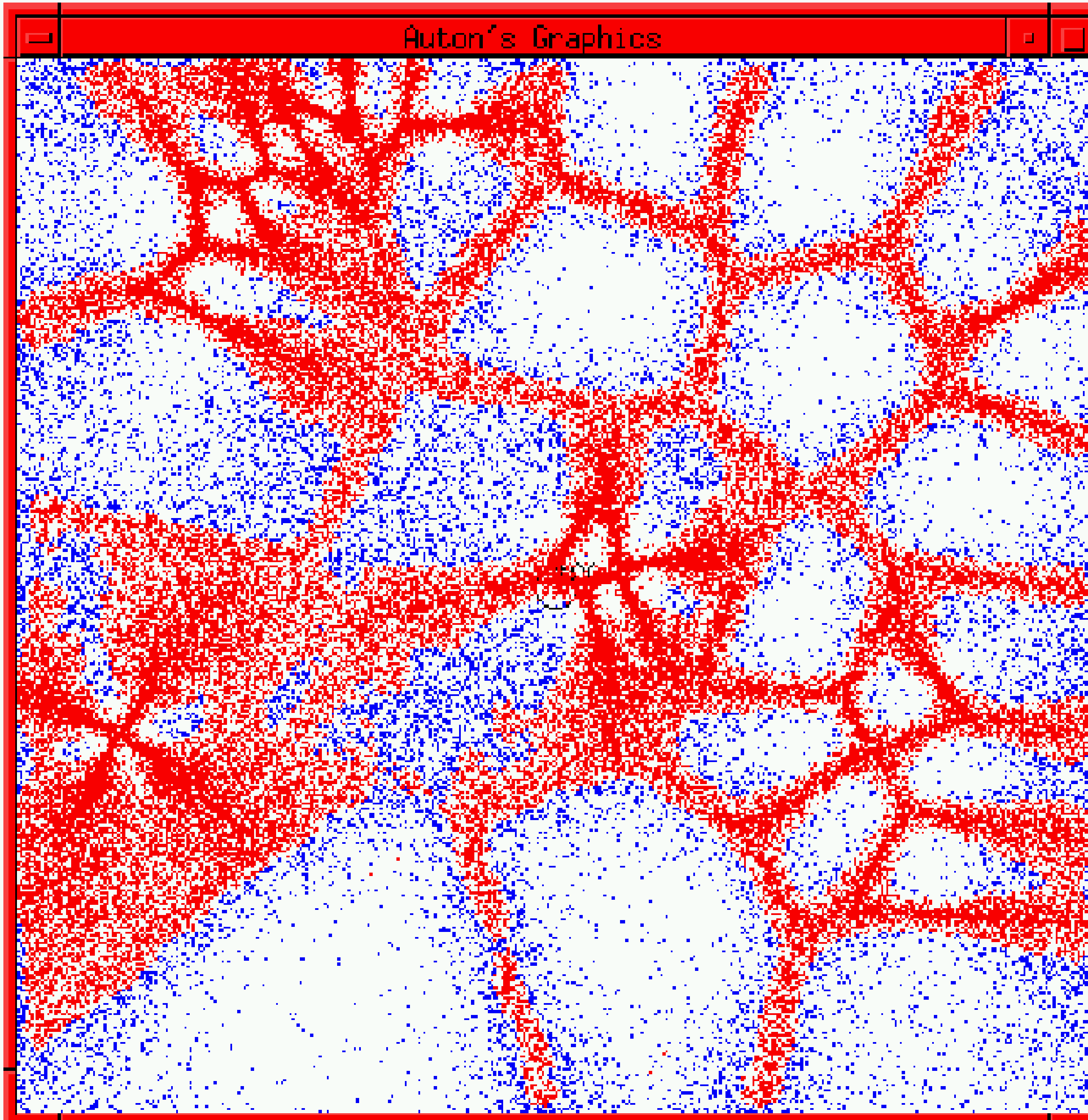


Gaussian mixture model says "It's a big old lump"

Spatial Anomaly Detection

Red dots are in a crowded neighborhood.

Blue dots are lonely.



Many other "All-pairs" problems

- Locally weighted polynomial regression
- Gaussian processes
- Point processes
- Bottom-up clustering

"All-pairs of attributes" important too...

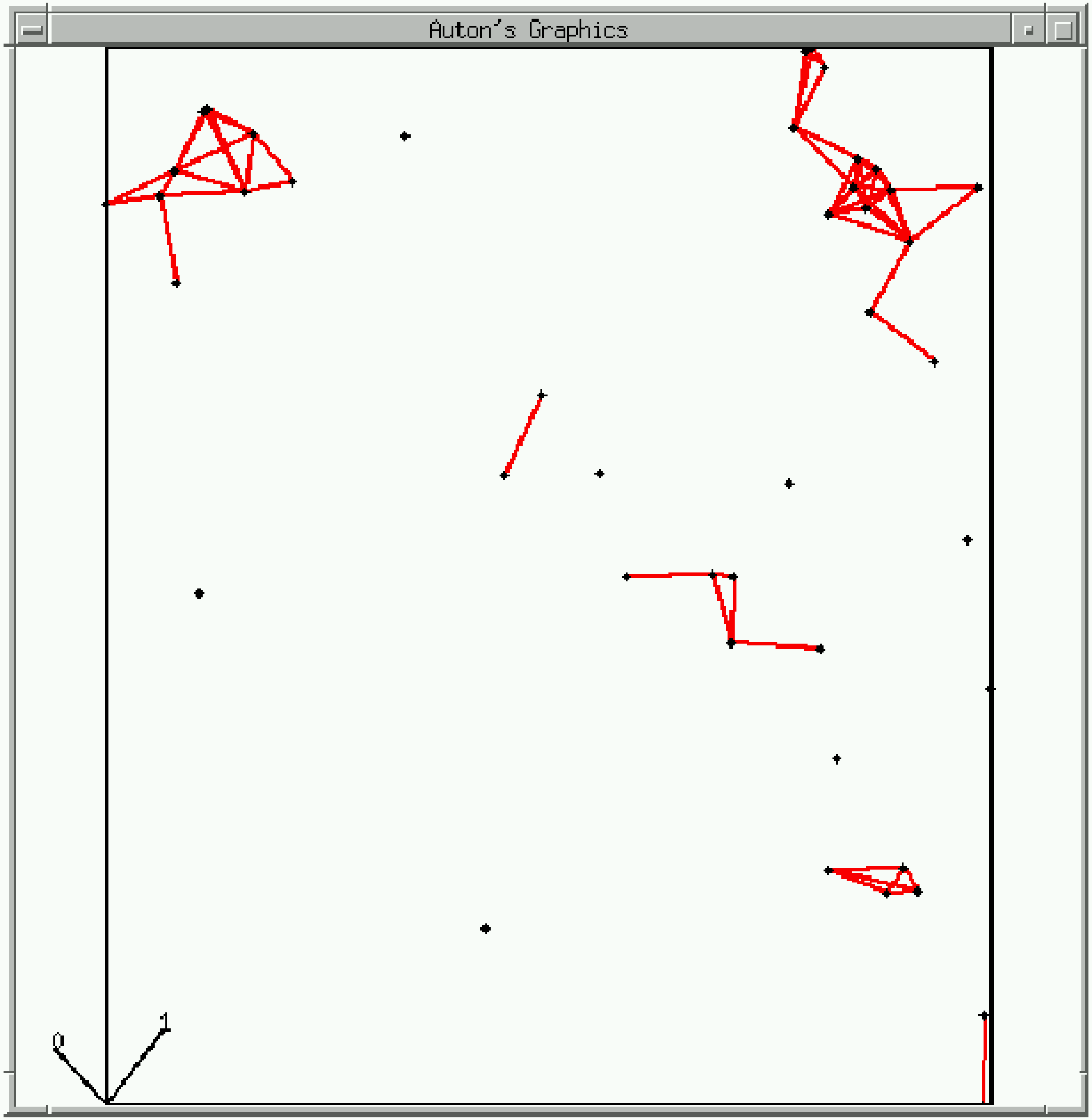
- Find me the most highly correlated pair of attributes.
- The most similar color-bands, image filters...

2-point correlation

...the purest form of an "all-pairs" problem.

There are 62 pairs of points that lie within 0.1 units of each other:

..important in astrophysics for characterizing matter distribution.



Fast all-point-pairs: Idea One

Use an $O(n^2)$ algorithm and buy a fast computer

Problem: $O(n^2)$ **is vicious.**

Comparative Results

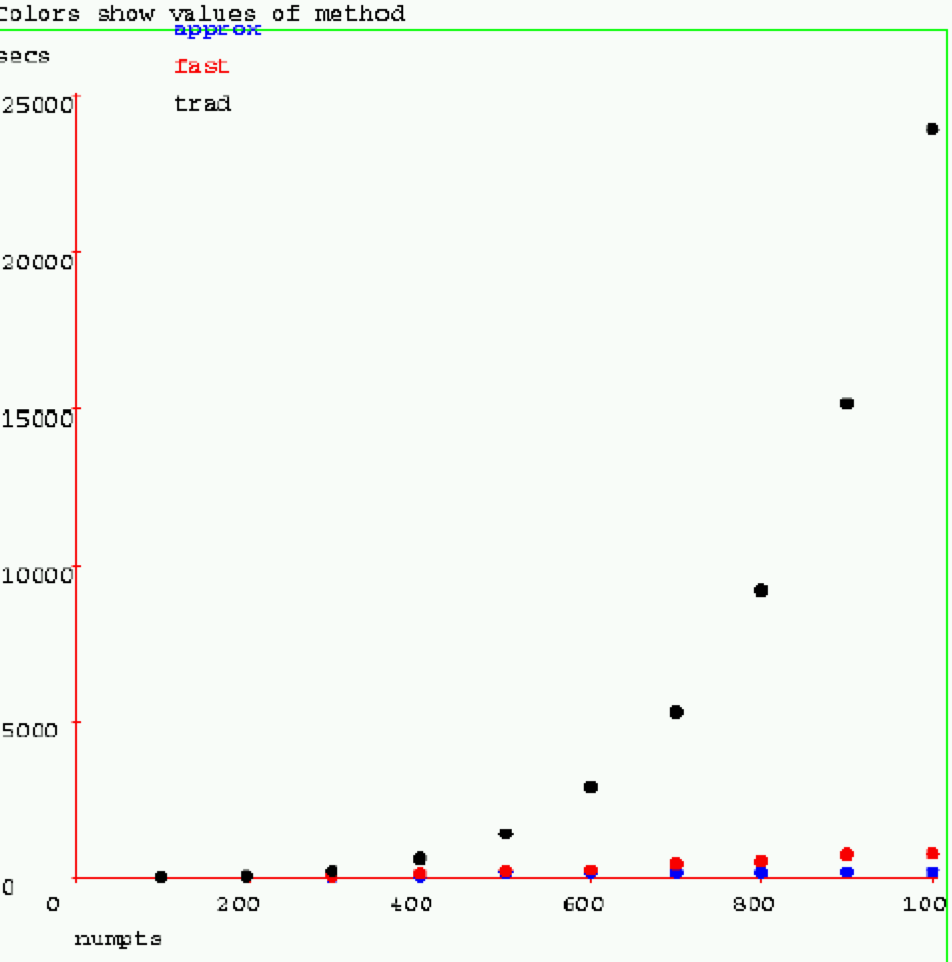
Non-approximate version

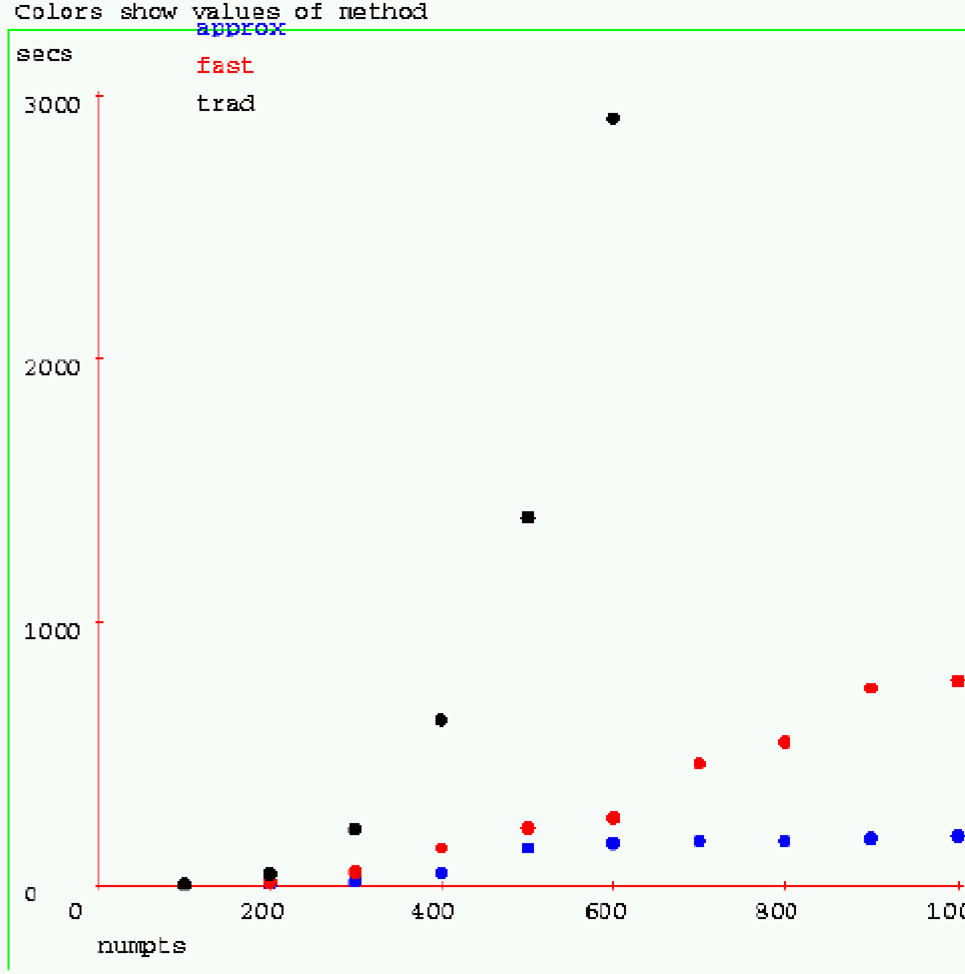
Number of Points	Quadratic time (secs)	Single-tree time (secs)	Dual-tree time (secs)	Single Tree Speedup	Dual Tree Speedup
10000	132	2.2	1.2	60	110
20000	528	4.8	2.8	110	189
50000	3300	11.8	7.0	280	471
150000	30899	37	20	835	1545
300000	123599	76	40	1626	3090

Approximate version (20,000 datapoints on slower machine):

ε	0.001	0.01	0.02	0.05	0.1	0.2	0.5
secs	37	30	30	18	10	10	0.3

4-point performance





Black: Traditional 4-point

Red: Fast Exact 4-point

Blue: Fast Approx 4-pt

Closeup

Outline

- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
  - Very fast non-parametric classification
  - skewed binary outputs
  - General binary outputs
  - multi-classed outputs
- Very fast kernel-based statistics
  - n-point computations
  - clustering
  - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

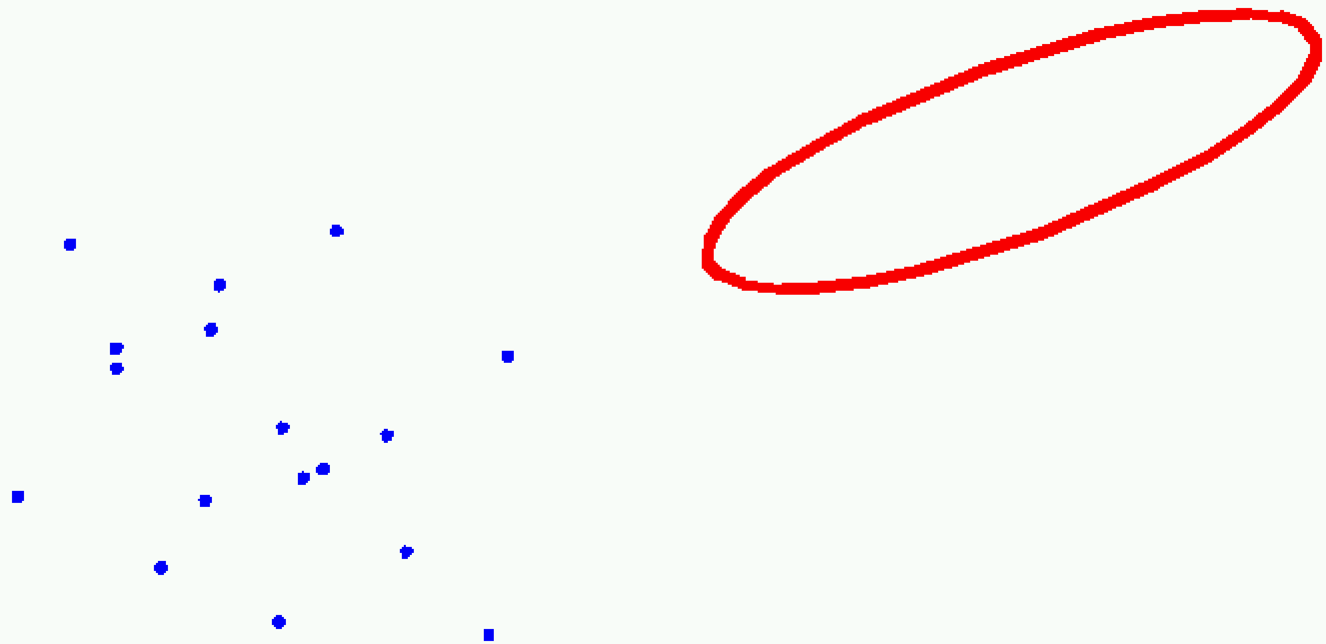
Data Structures for Fast K-means

The Auton Lab

Carnegie Mellon University

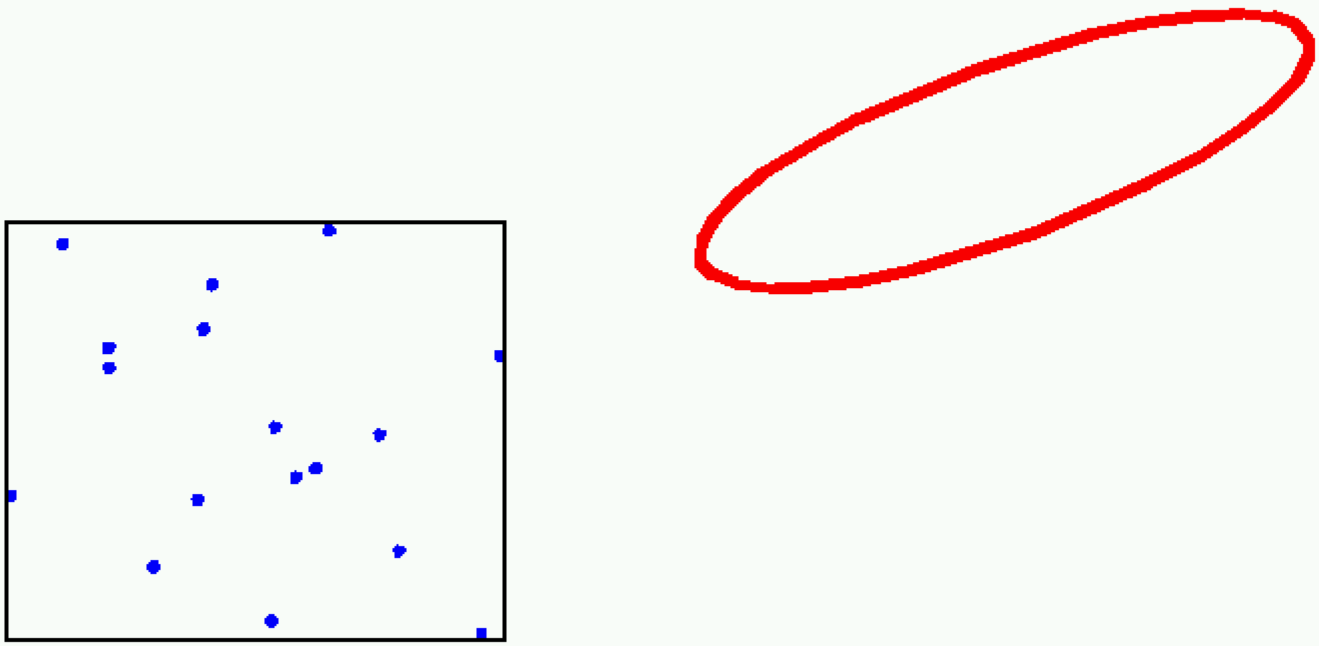
www.autonlab.org

Computing likelihood of datapoints…



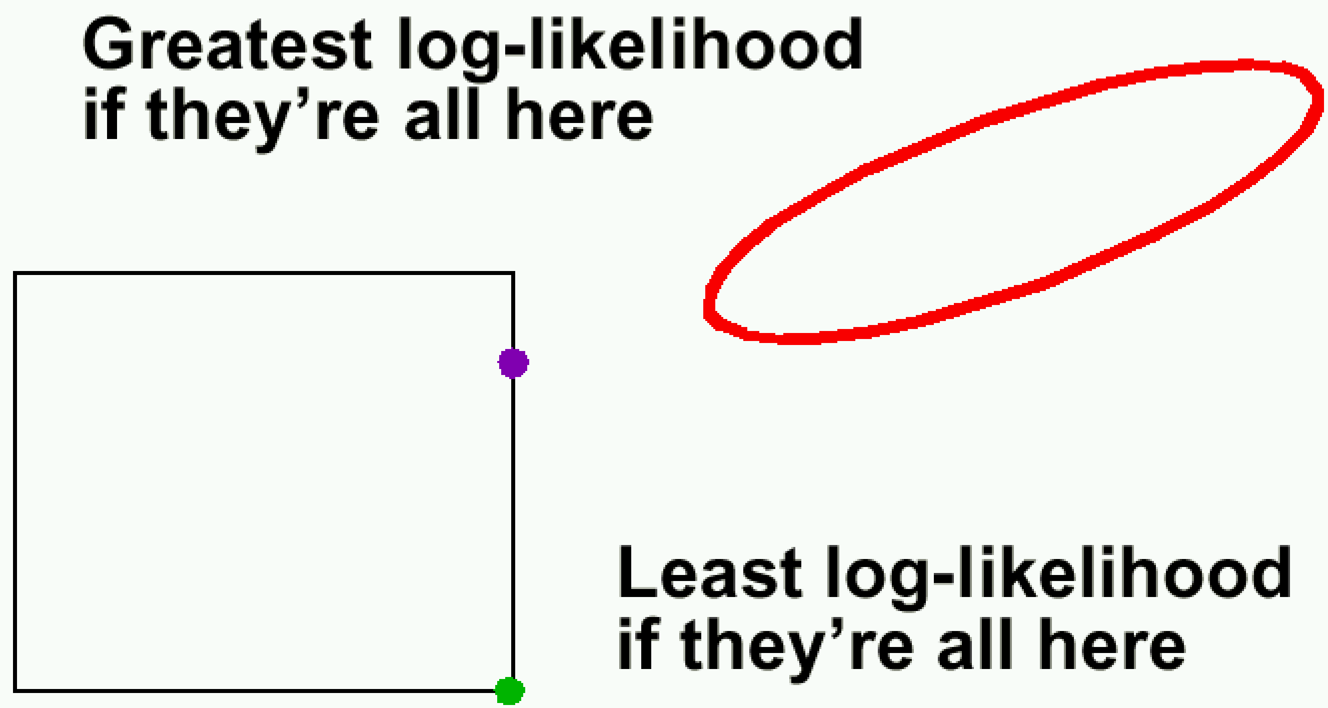
Suppose you want to compute the sum of log-likelihoods of all the blue dots given they'd been generated by the big red Gaussian.

Computing likelihood of datapoints...



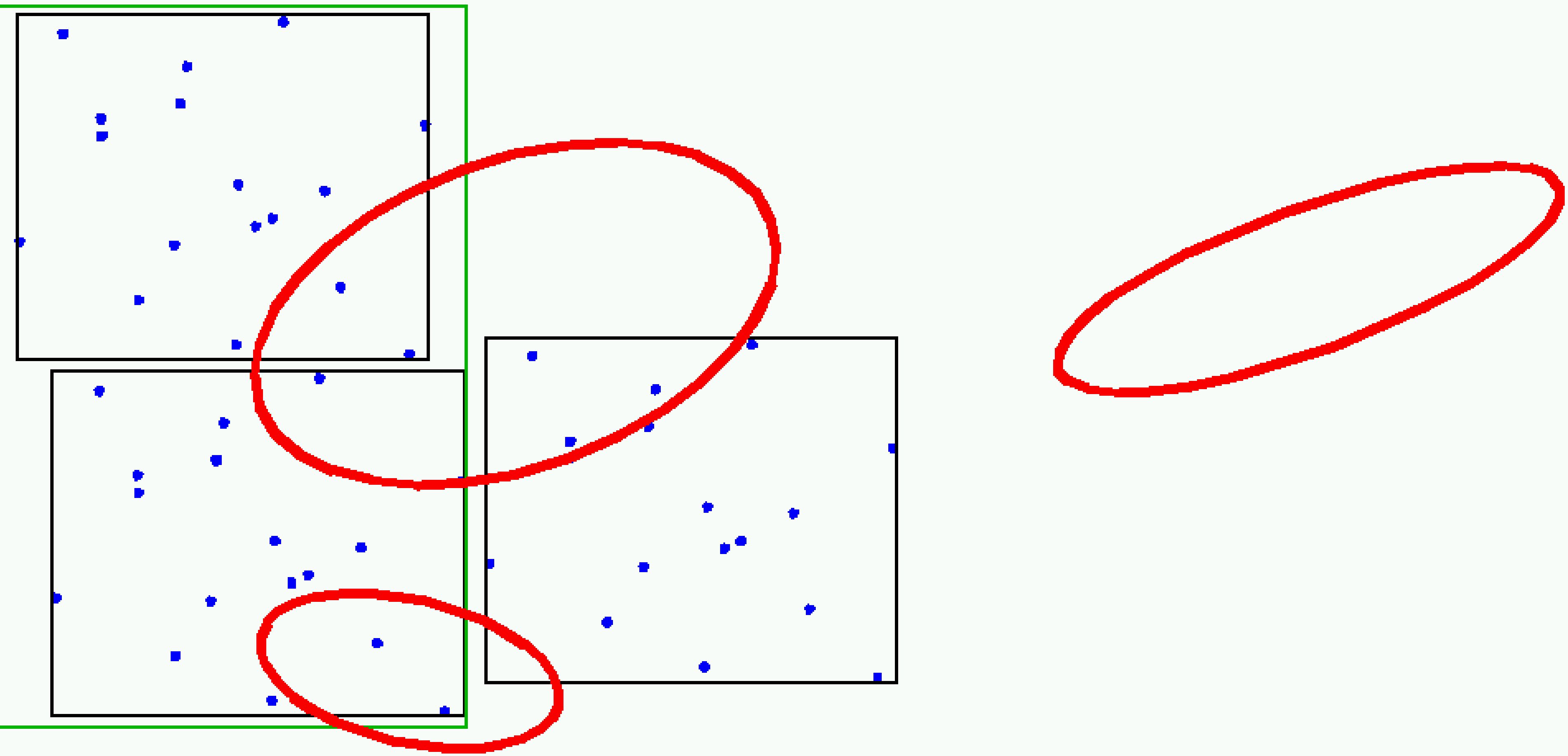
Suppose we happen to know their bounding box

Computing likelihood of datapoints...



Without visiting the points individually, we can put bounds on their contributions to the Gaussian. Sometimes those bounds'll be tight enough...

Many points, boxes, Gaussians...



f you play this game on a large scale, you find yourself doing kernel densities, locally weighted regression, locally weighted PCA, adaptive kernels, k-means clustering, snake-based filament tracing, hierarchical classification.....

............very very very fast

Cached sufficient statistics

What I've shown you:

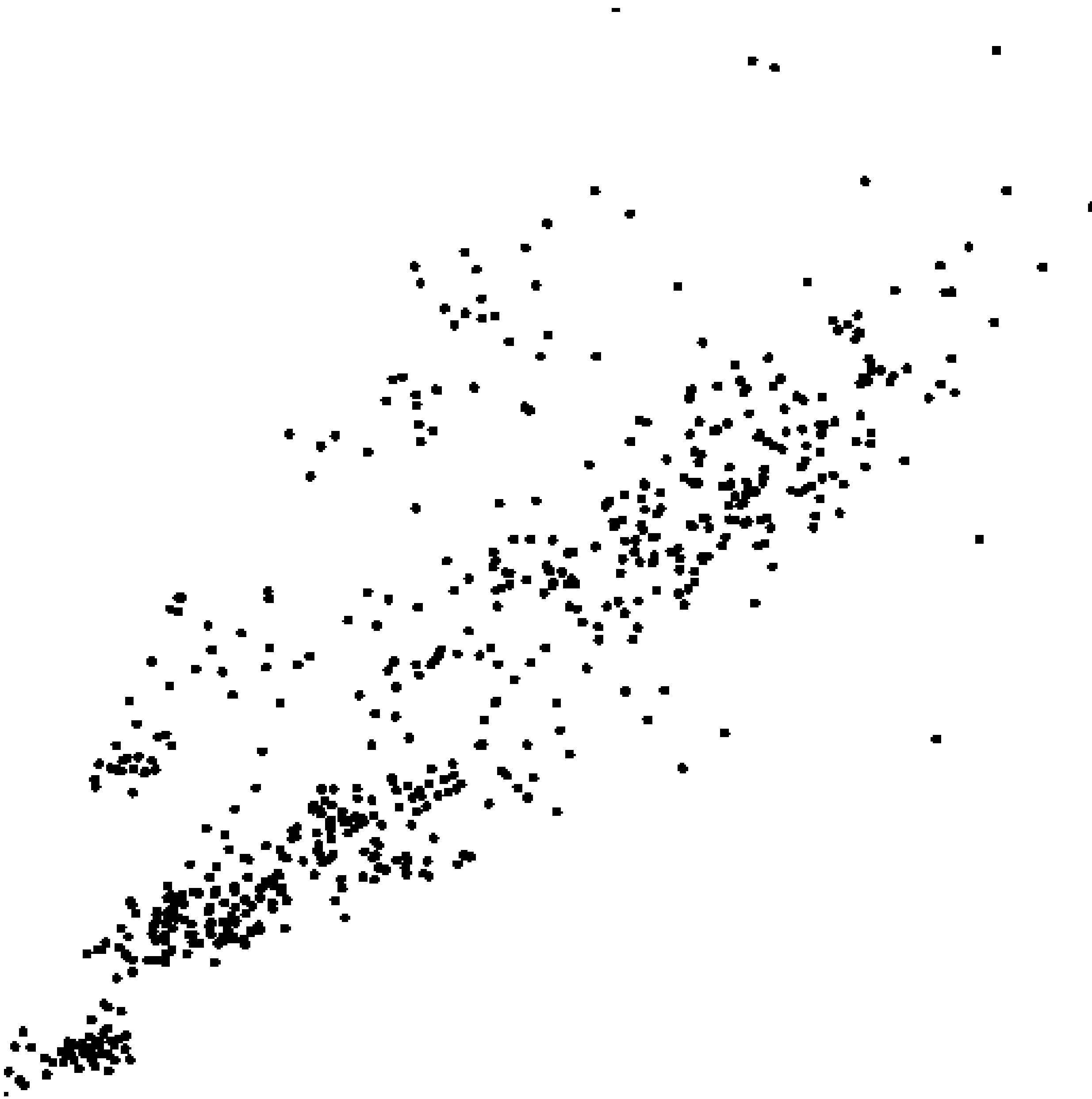
- It's intuitively possible to look at a node in a tree and decide whether in order to estimate the data loglikeihood you need to see more detail.

What I don't have time to show you

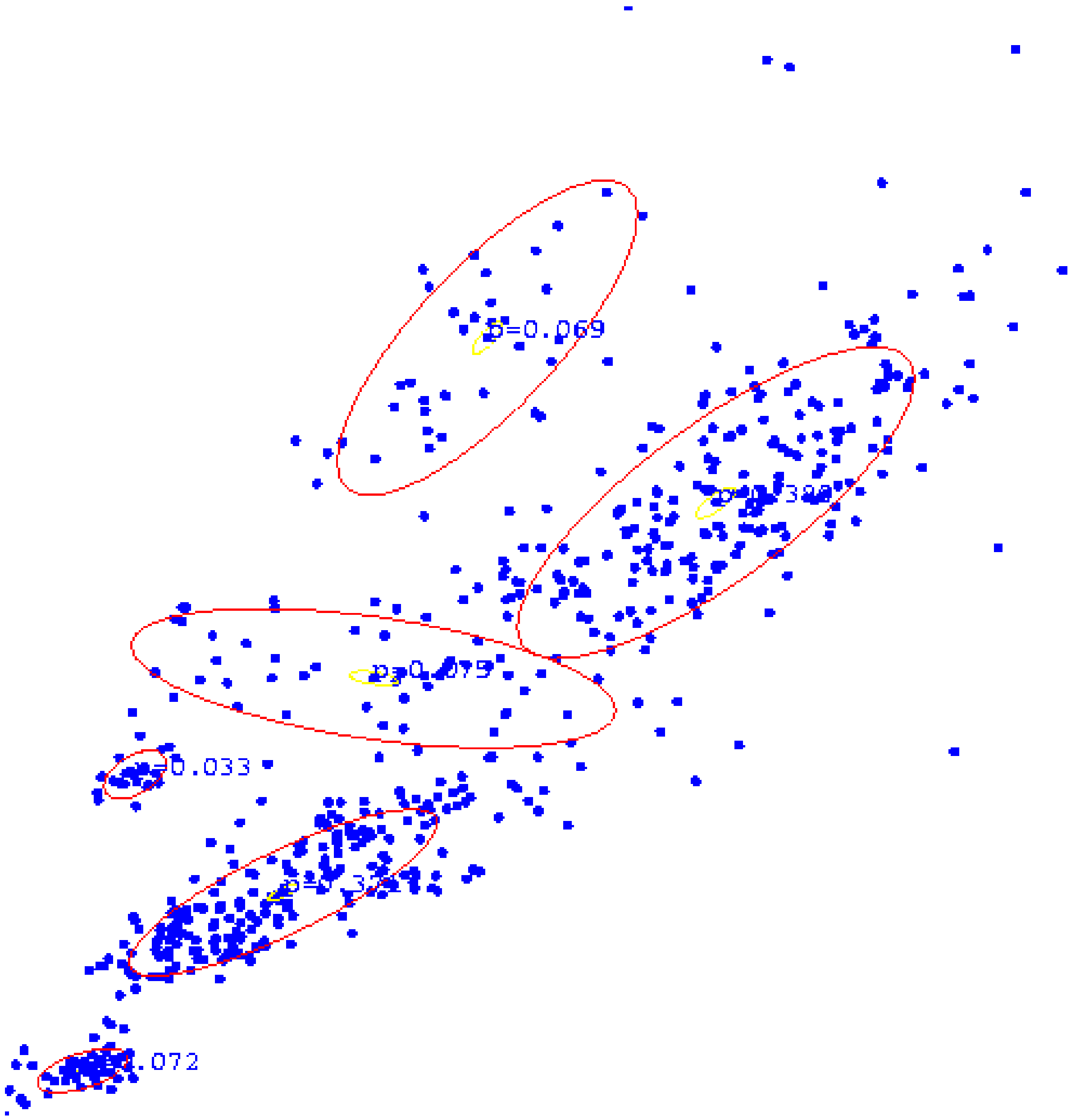
- Why you must also cache information other than the bounding box in every single kdtree node:
  - The centroid of all points it owns.
  - The covariance of all points.
- Each algorithm plays different tricks with these kinds of bounds
- Same principal as Barnes Hut and Greengard but sometimes tricker.

(Moore and Johnson 1993, Deng and Moore 1995, Moore, Schneider and Deng 1997, Moore 1999, Pelleg and Moore, 1999, Gordon and Moore, 2000)

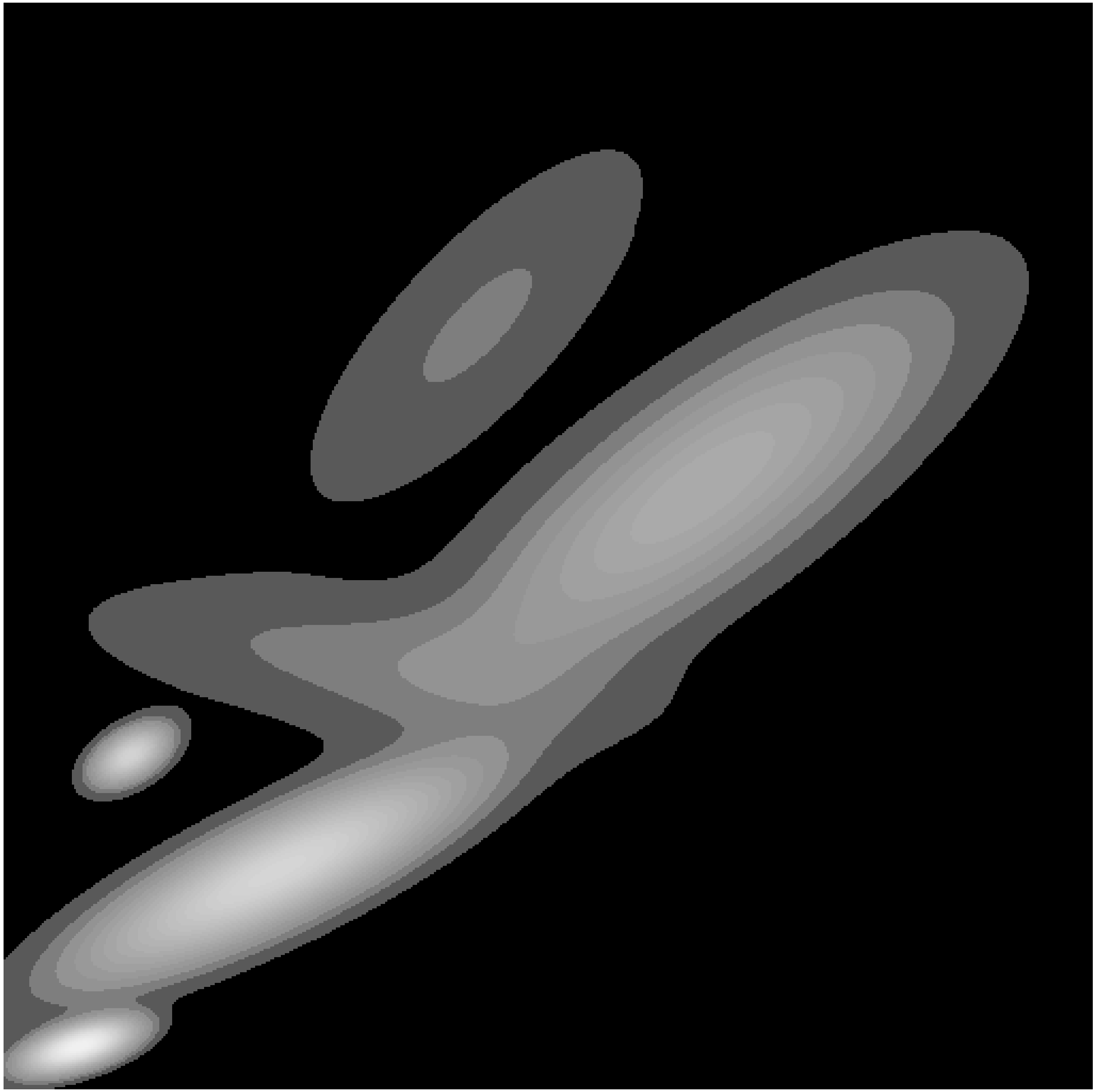
Some Bio Assay data



GMM clustering of the assay data



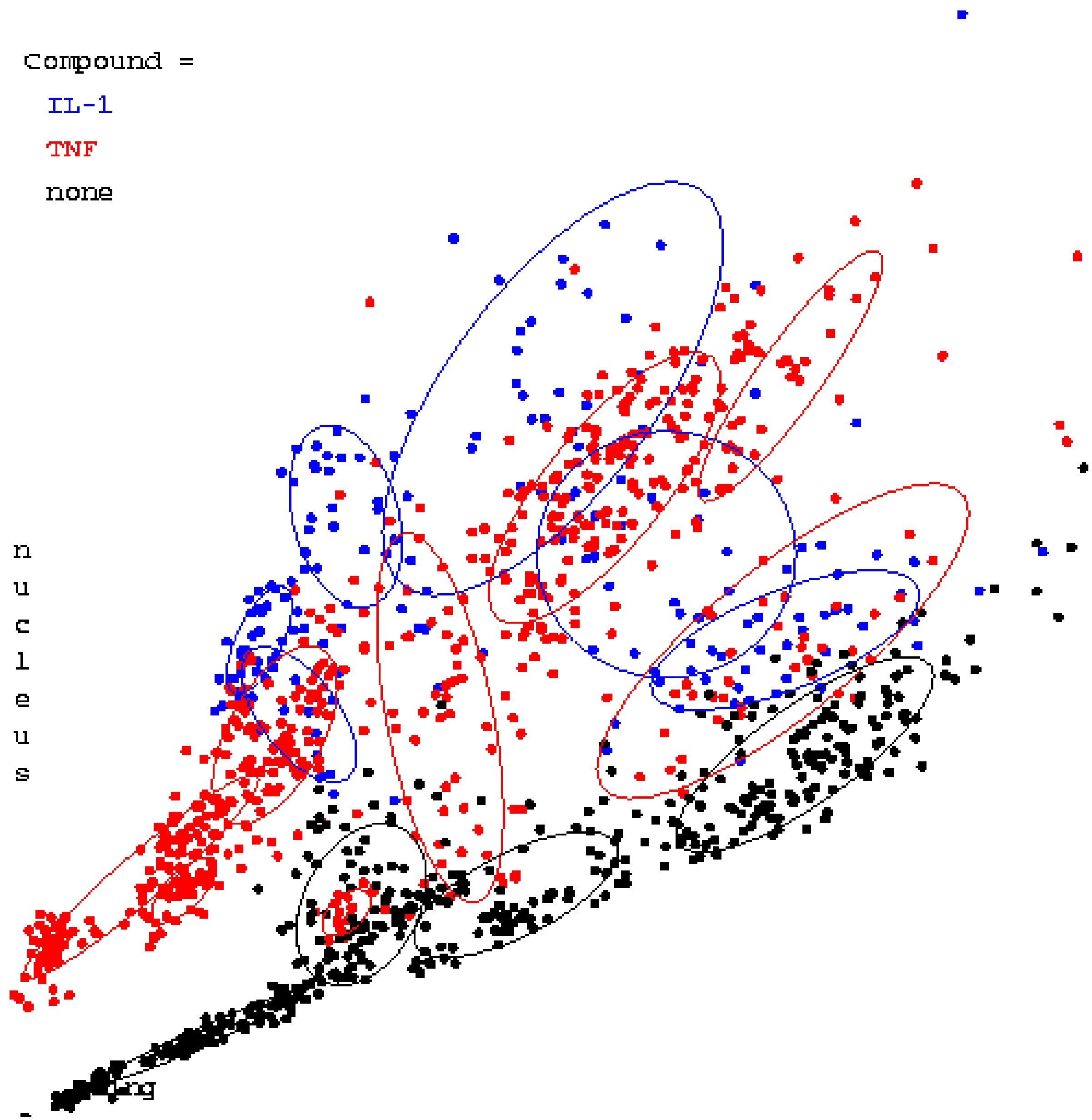
Resulting Density Estimator



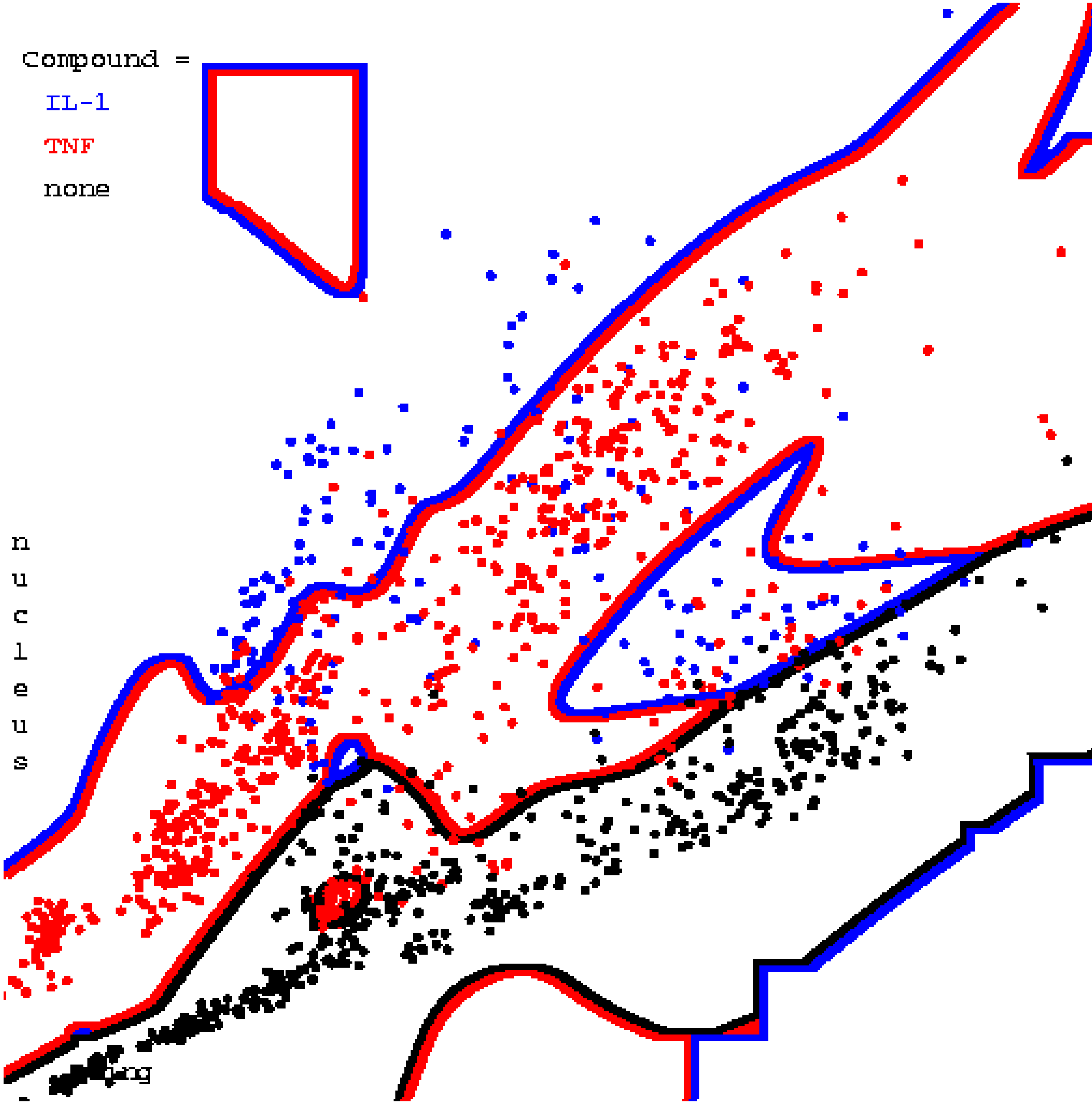
Three classes of assay

(each learned with it's own mixture model)

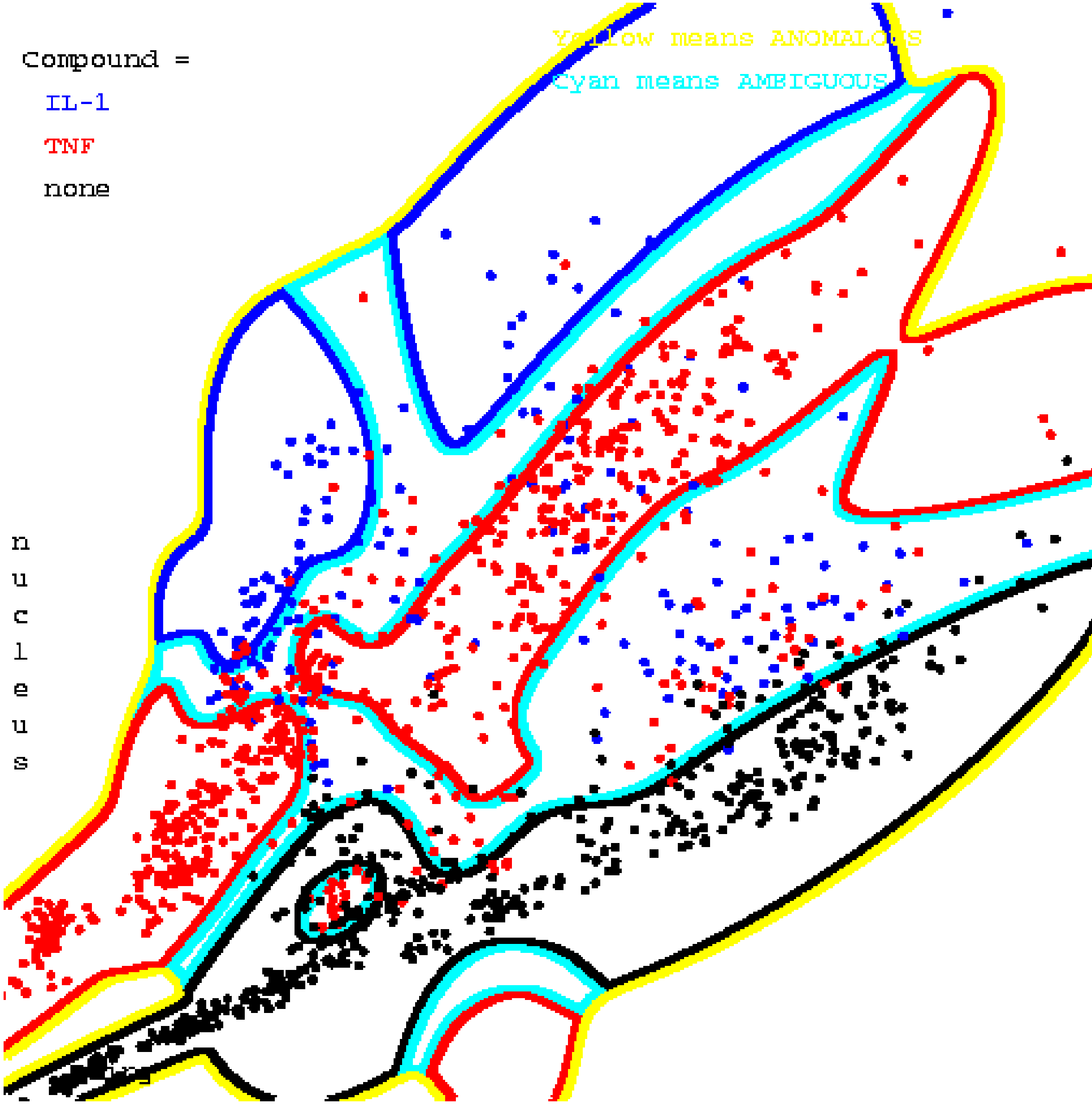
(Sorry, this will again be semi-useless in black and white)



Resulting Bayes Classifier



Resulting Bayes Classifier, using posterior probabilities to alert about ambiguity and anomalousness



Yellow means anomalous

Cyan means ambiguous

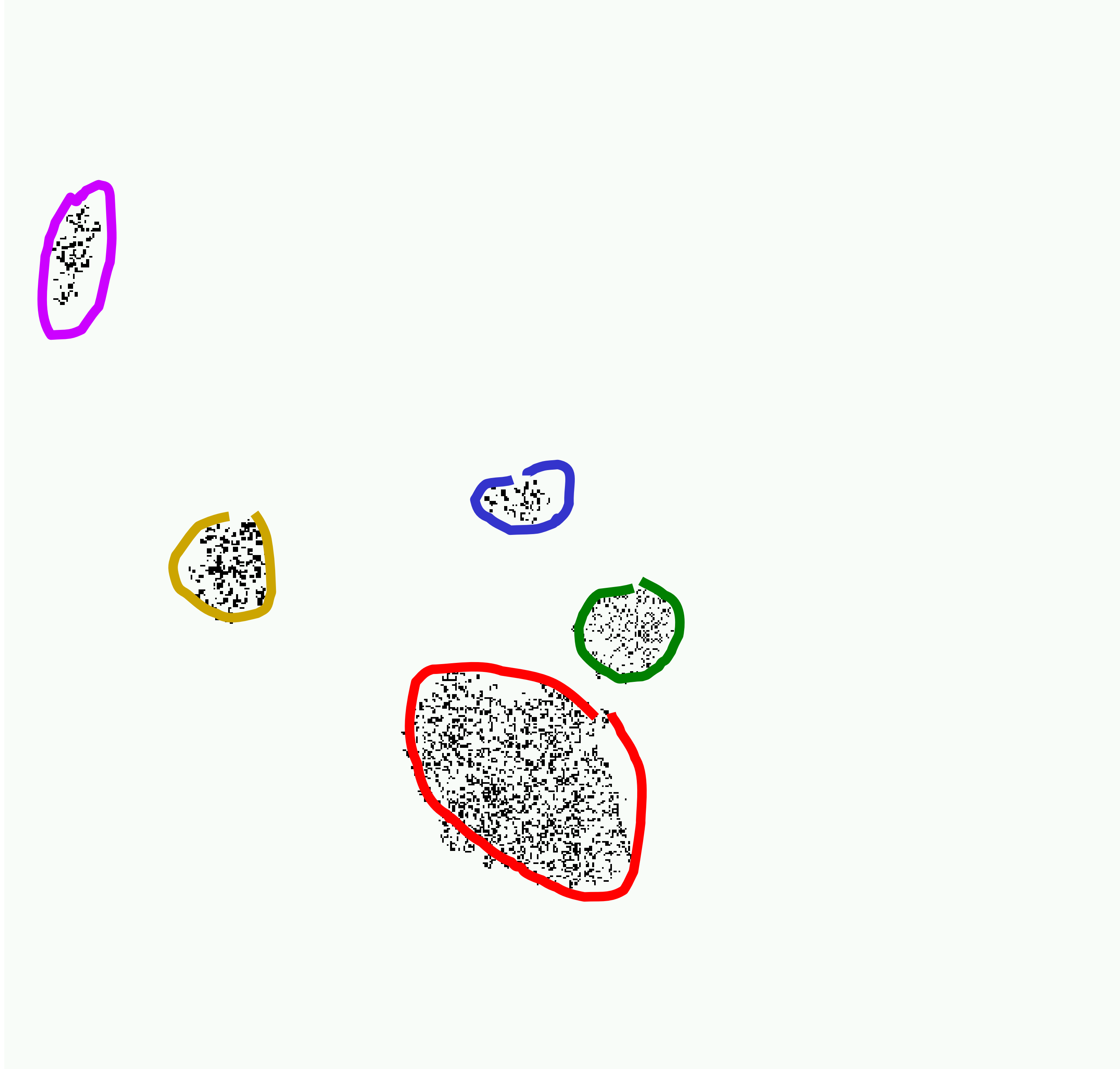
[Moore, 1999], [Pellg and Moore, 2002]

Outline

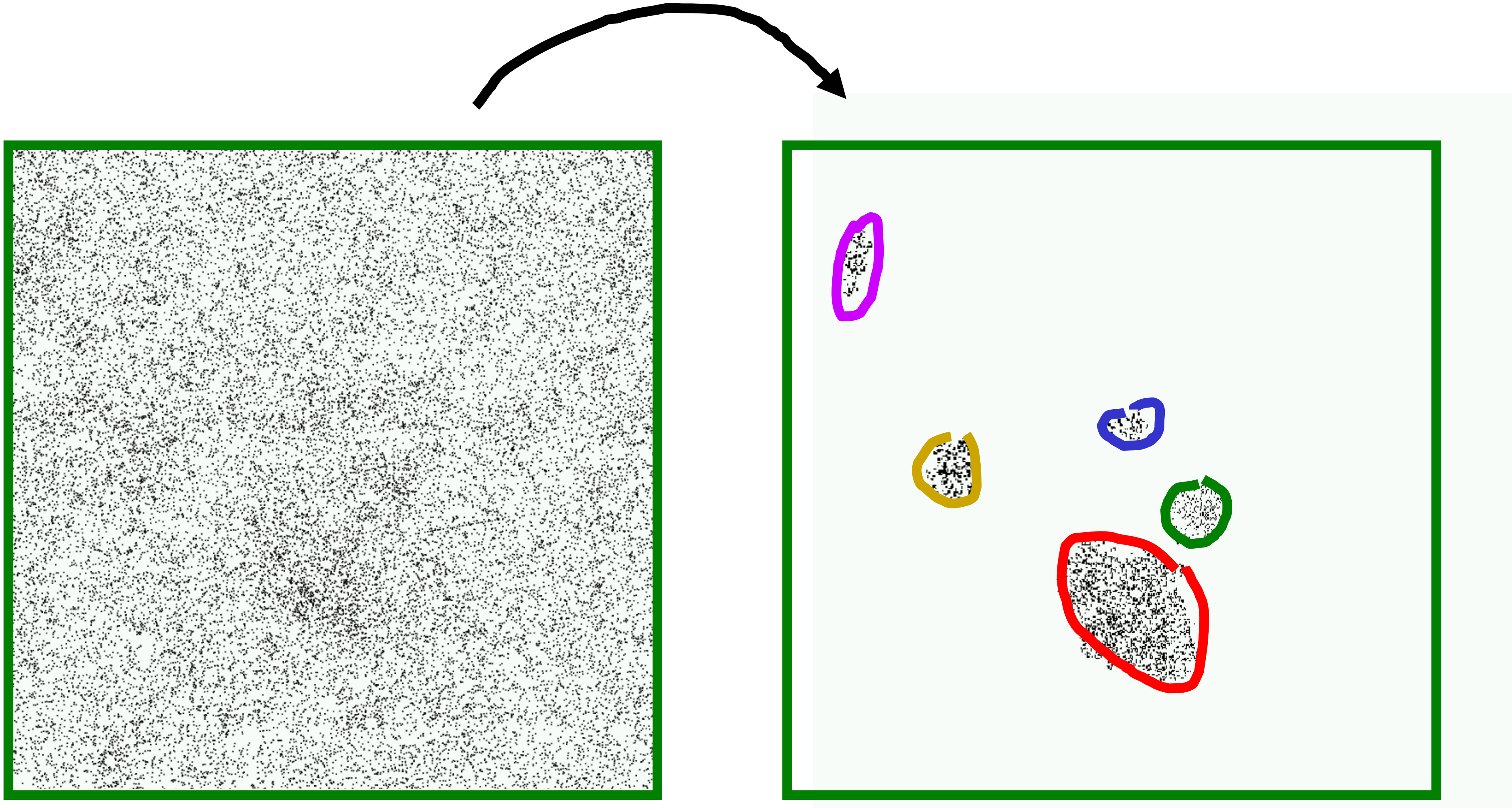
- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
  - Very fast non-parametric classification
  - skewed binary outputs
  - General binary outputs
  - multi-classed outputs
- Very fast kernel-based statistics
  - n-point computations
  - clustering
  - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

Detecting overdensities

Detecting overdensities



Finding the overdense regions



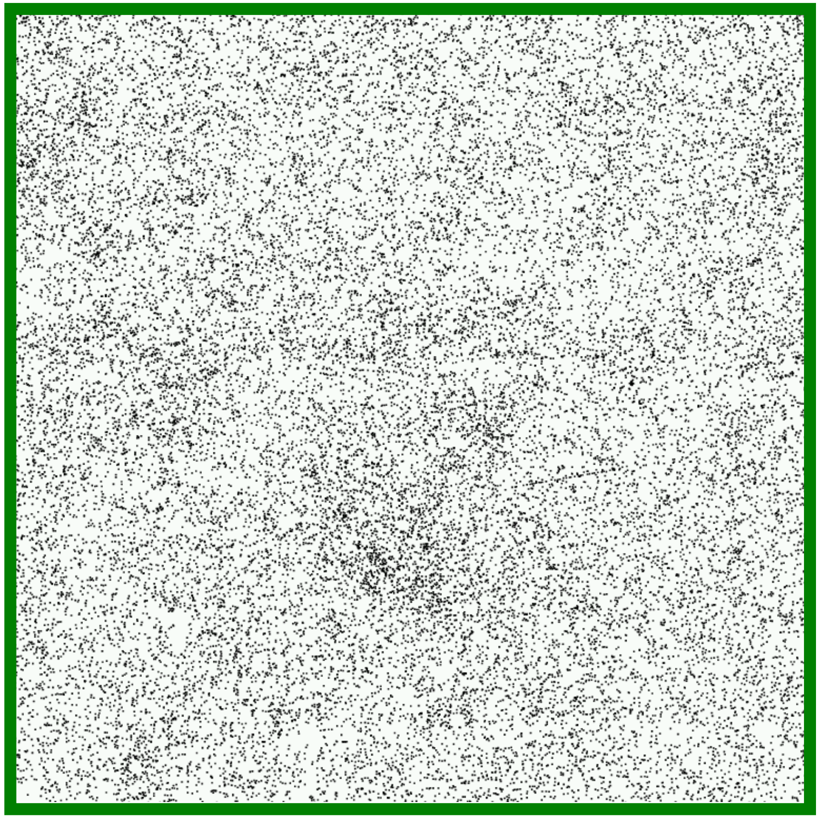
Finding the overdense regions

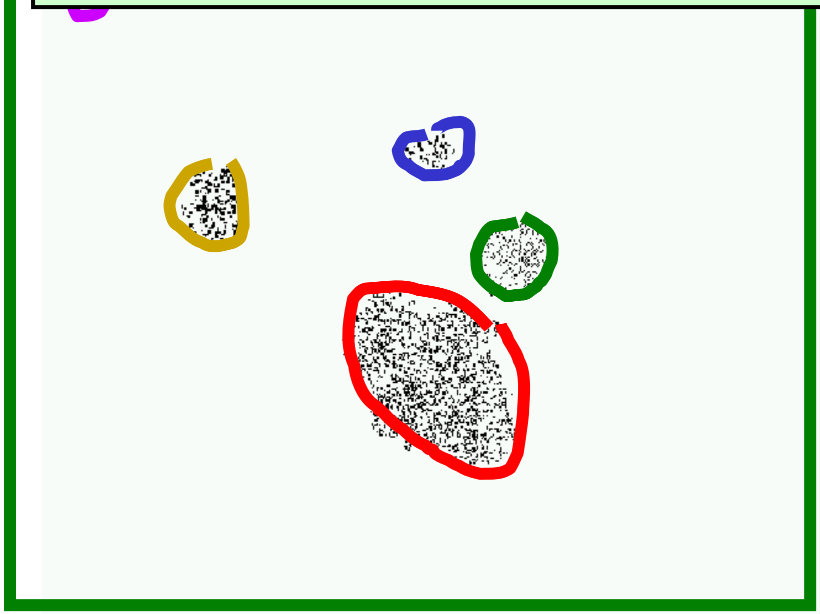
- Many possible approaches

  Examples:

  - Dasgupta and Raftery, 1998
  - Byers and Raftery, 1998
  - Cuevas et al, 2000
  - Reichart et al, 1999

- All share the same computational primitives





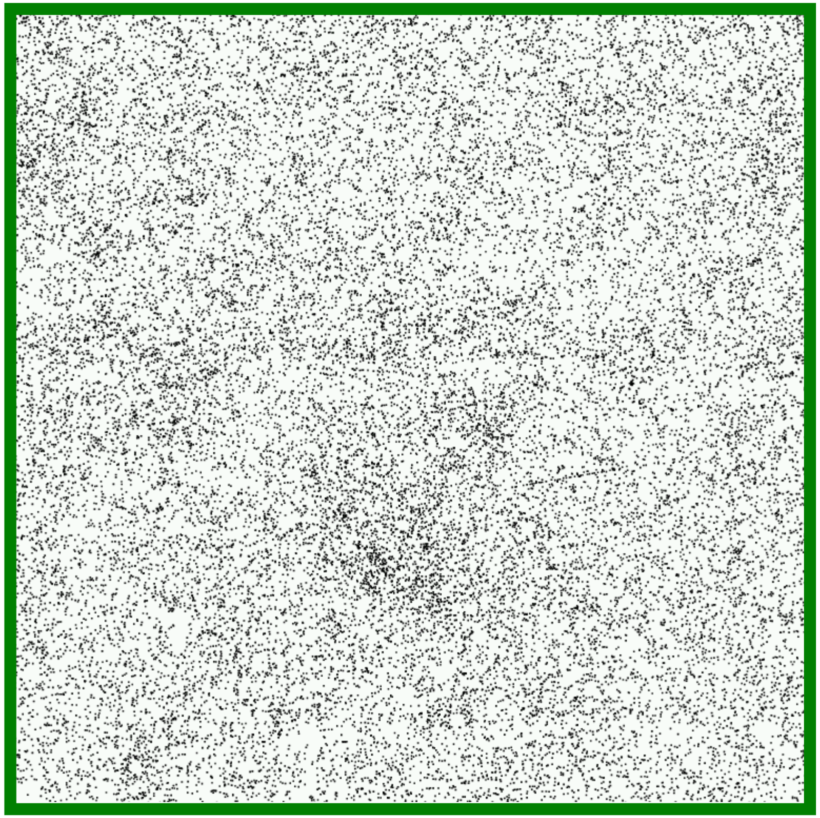
Finding the overdense regions

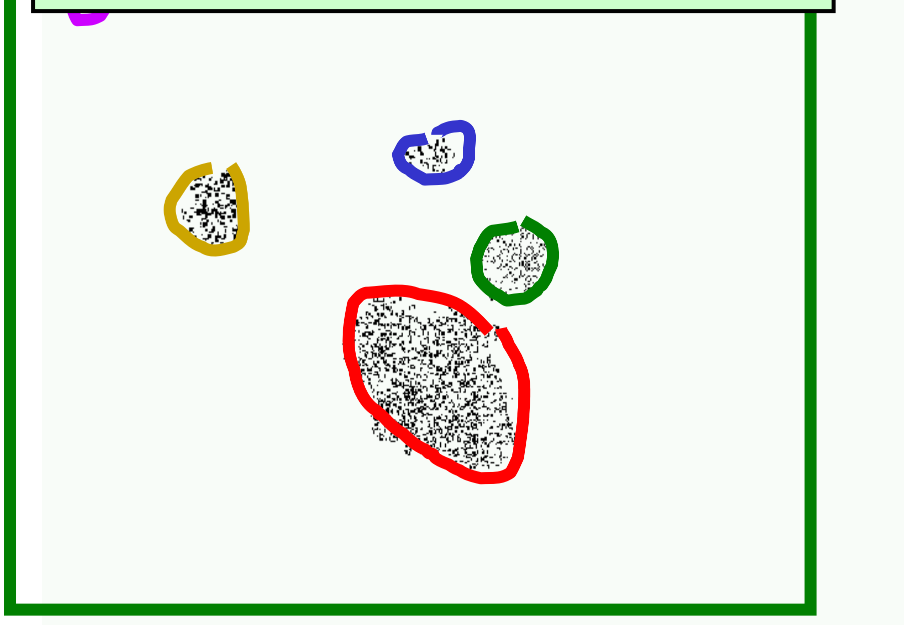
- Many possible approaches

  Examples:

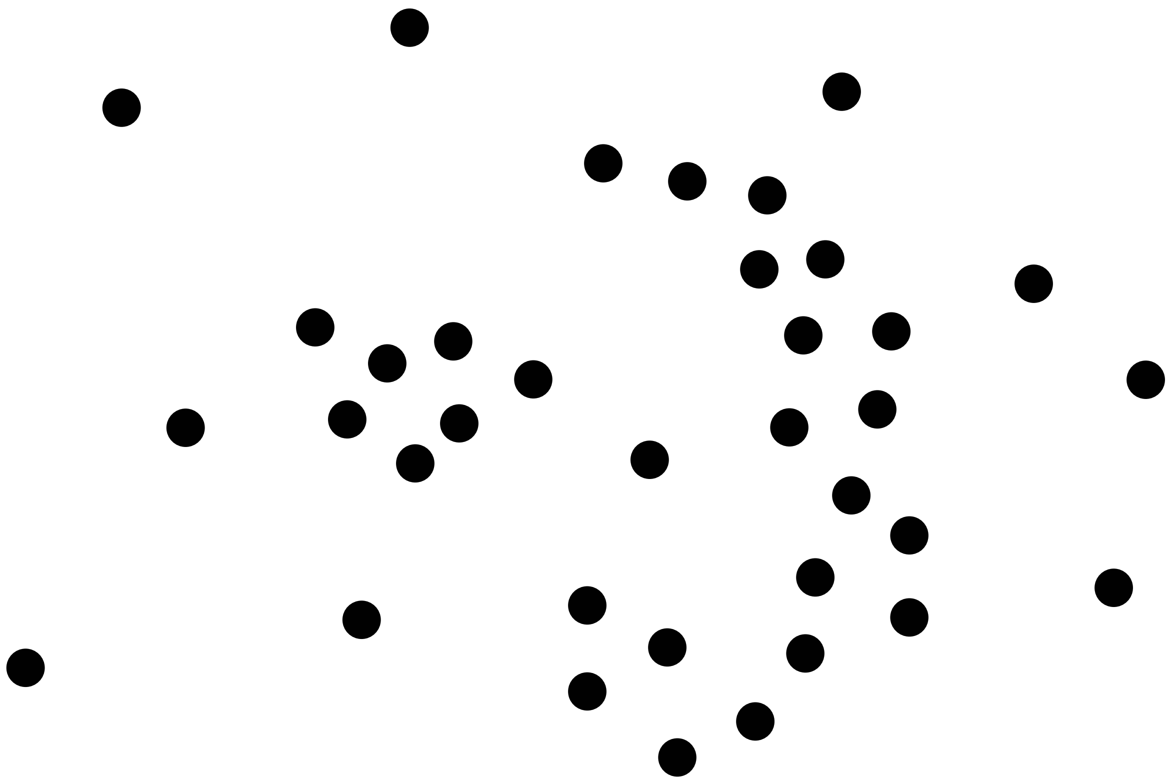
  - Dasgupta and Raftery, 1998
  - Byers and Raftery, 1998
  - Cuevas et al, 2000
  - Reichart et al, 1999

- All share the same computational primitives



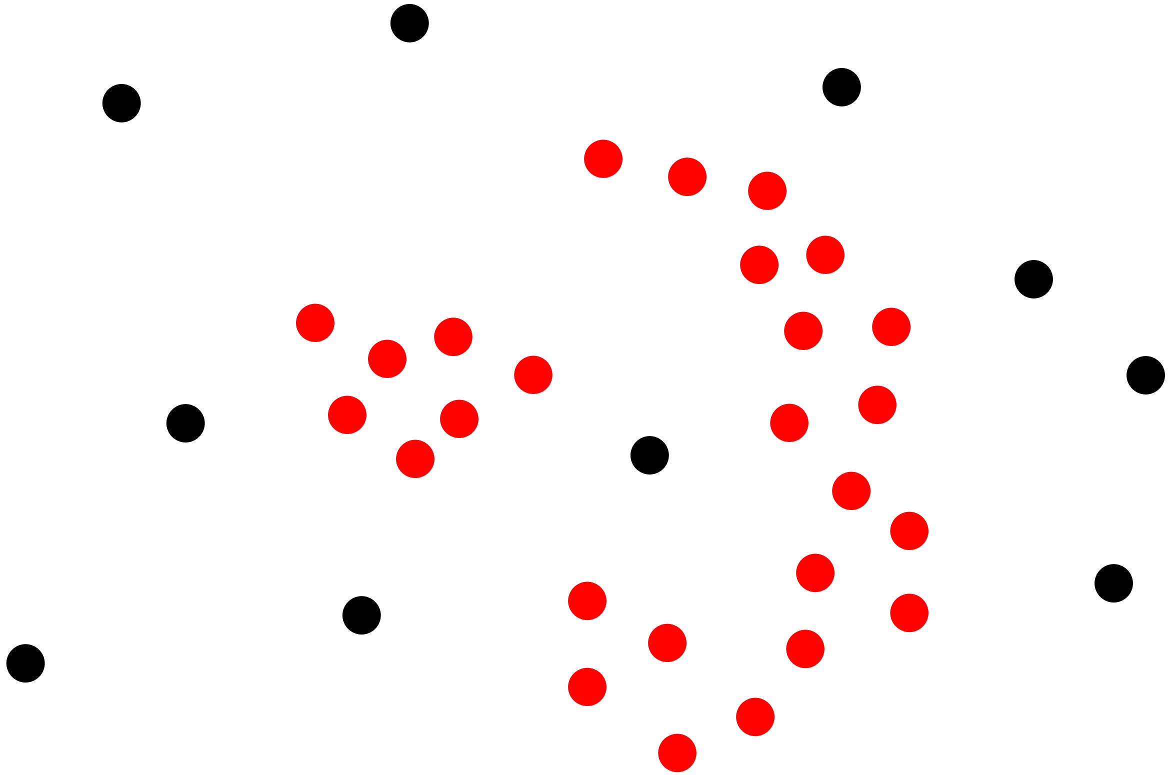


Finding the overdense regions



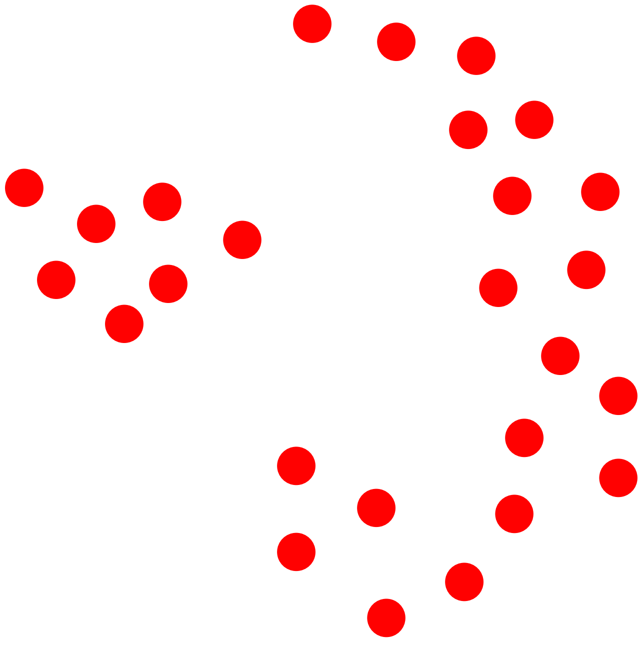
Finding the overdense regions

- Step One:
  Identify the high density points



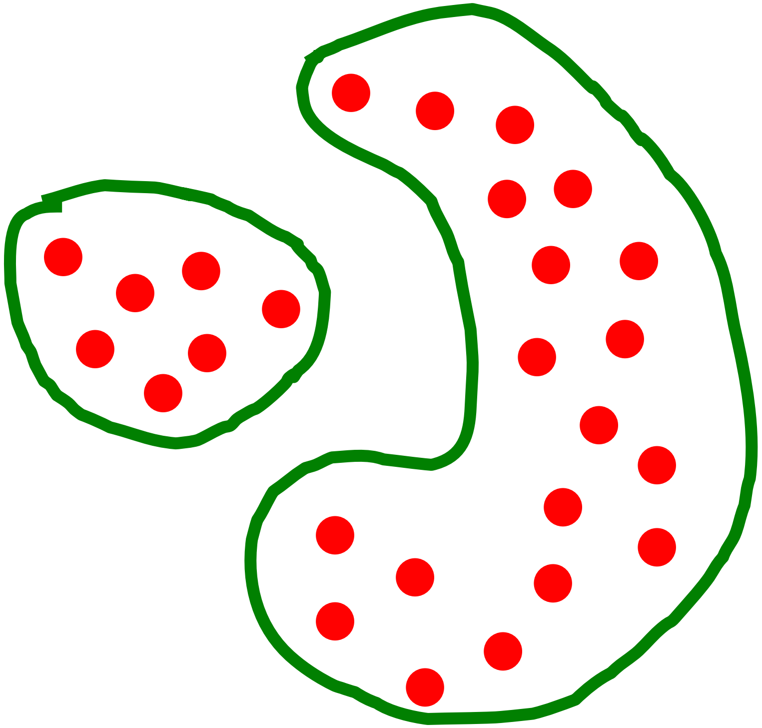
Finding the overdense regions

- Step One:
  Identify the high density points
- Step Two:
  Delete the rest



Finding the overdense regions

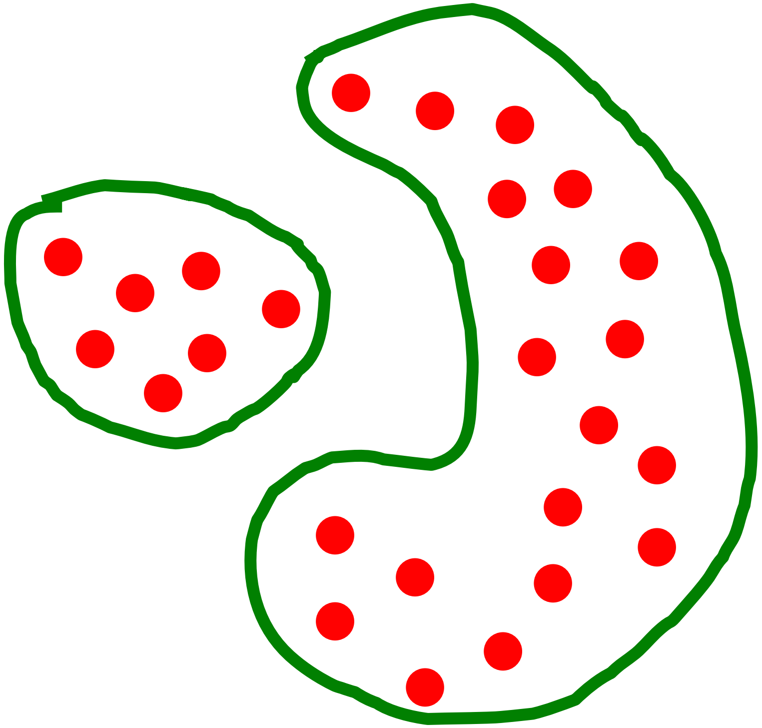
- Step One:
  Identify the high density points
- Step Two:
  Delete the rest
- Step Three:
  Find connected components



Finding the overdense regions

CFF assumes:
Kernel Density Estimation

- Step One:
  Identify the high density points
- Step Two:
  Delete the rest
- Step Three:
  Find connected components



Finding the overdense regions

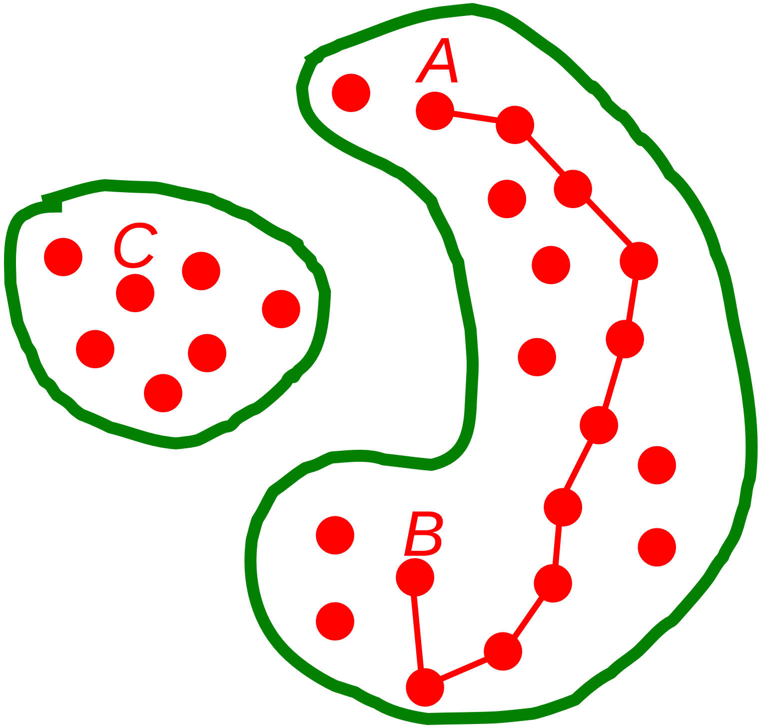
CFF assumes: Kernel Density Estimation

- Step One: Identify the high density points

- Step Two: Delete the rest

CFF assumes: A and B are in same component if there's a path between them with all small steps

- Step Three: Find connected components



Finding the overdense regi

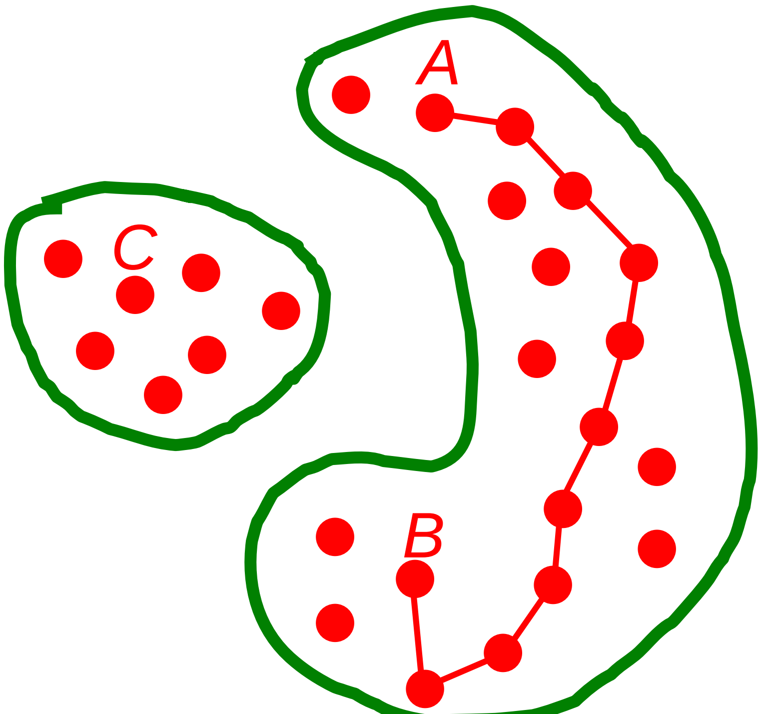
CFF assumes:
Kernel Density Estimation

Can be done efficiently with 2-point style search

- Step One:
  Identify the high density points
- Step Two:
  Delete the rest
- Step Three:
  Find connected components

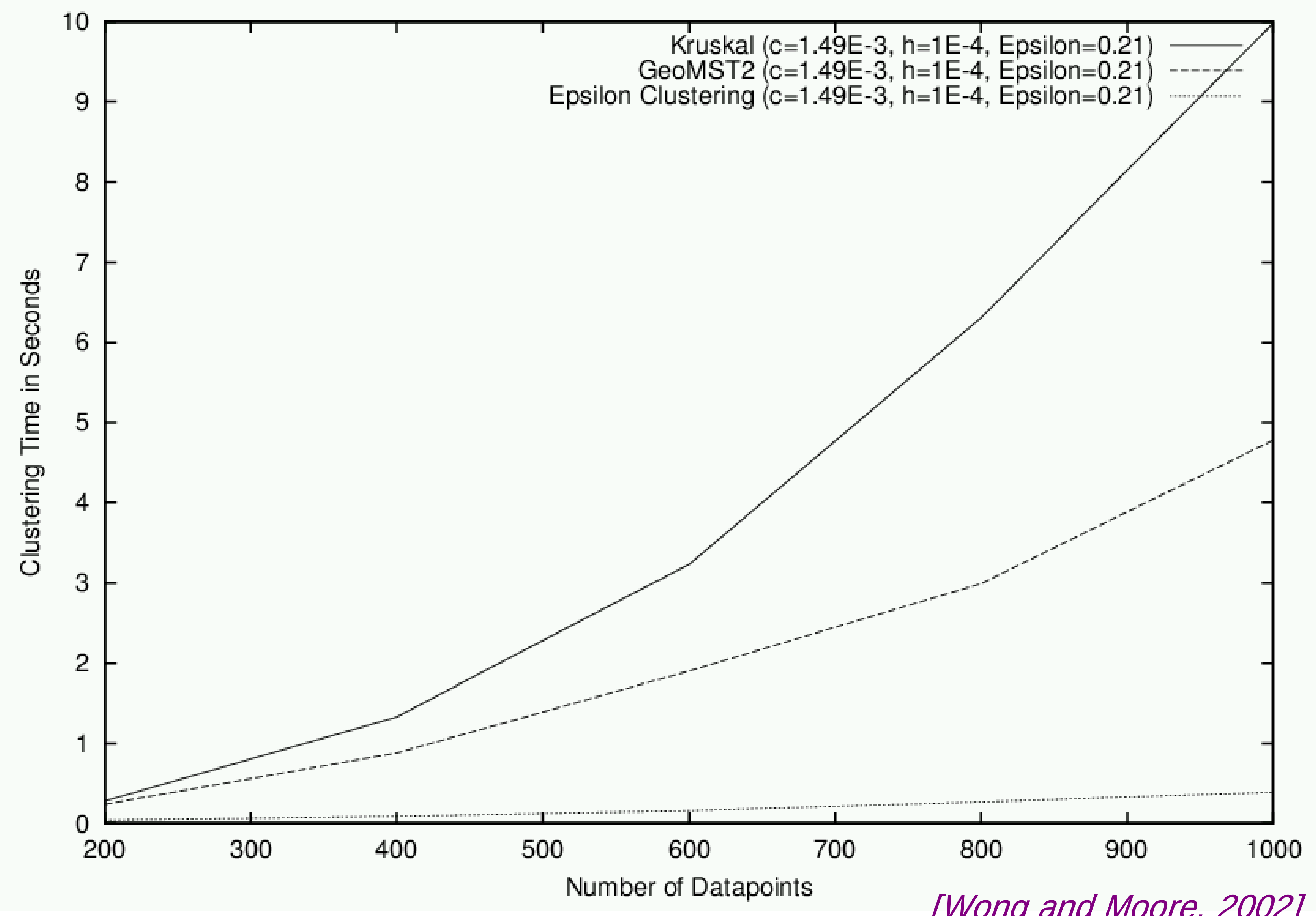
CFF assumes: A and B are in same component if there's a path between them with all small steps

Can be done efficiently with 2-point-style search plus an extra trick



Results

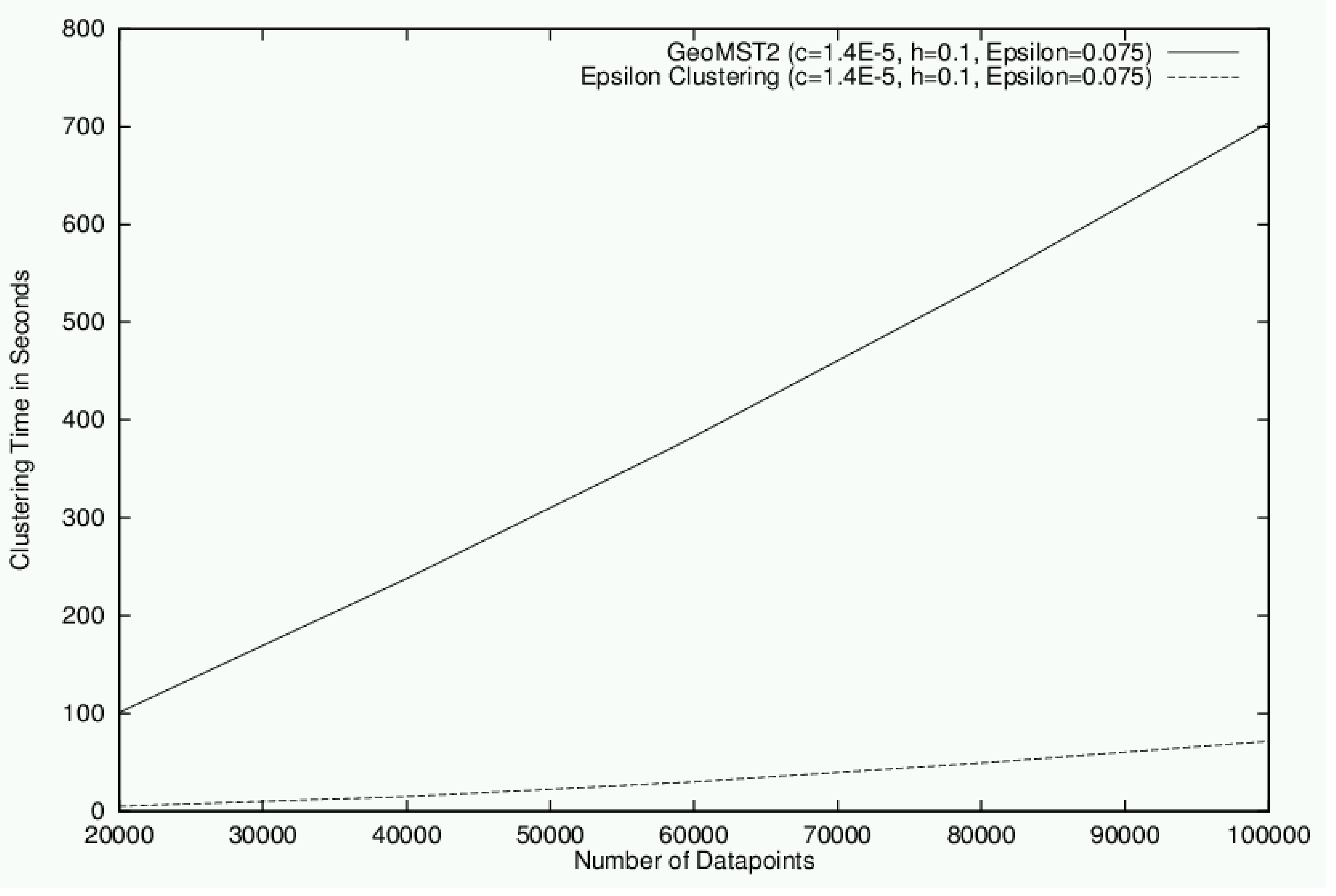
4-dimensional Sloan
Astrophysics color-space data



[Wong and Moore, 2002]

Results

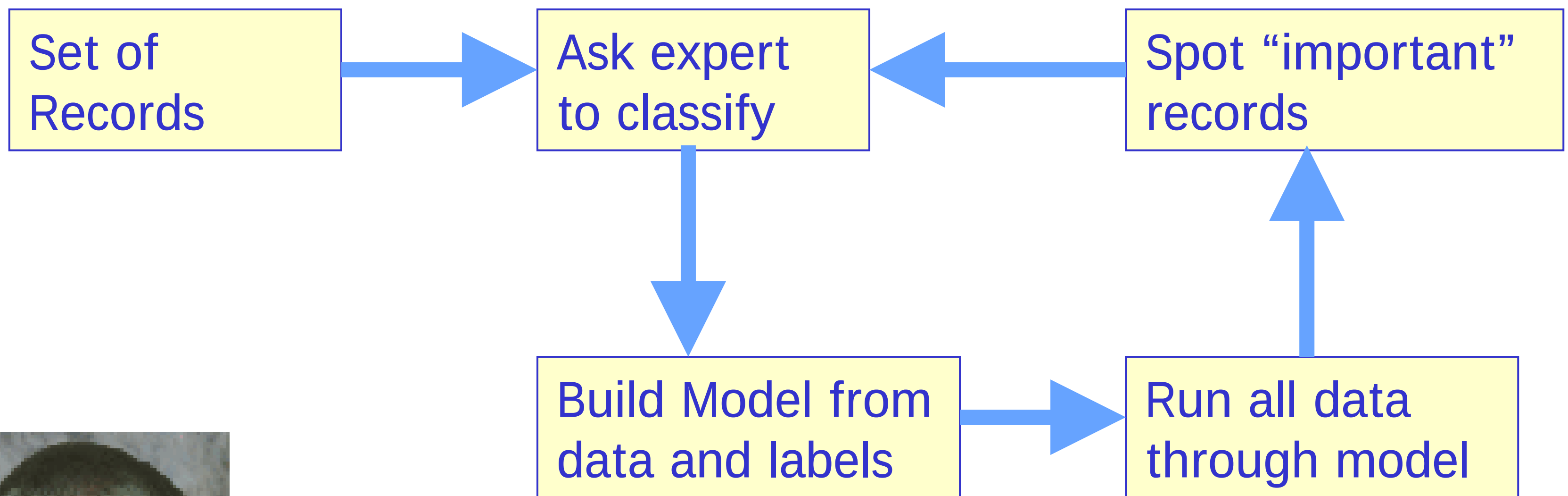
4-dimensional Sloan
Astrophysics color-space data



Outline

- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
  - Very fast non-parametric classification
  - skewed binary outputs
  - General binary outputs
  - multi-classed outputs
- Very fast kernel-based statistics
  - n-point computations
  - clustering
  - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

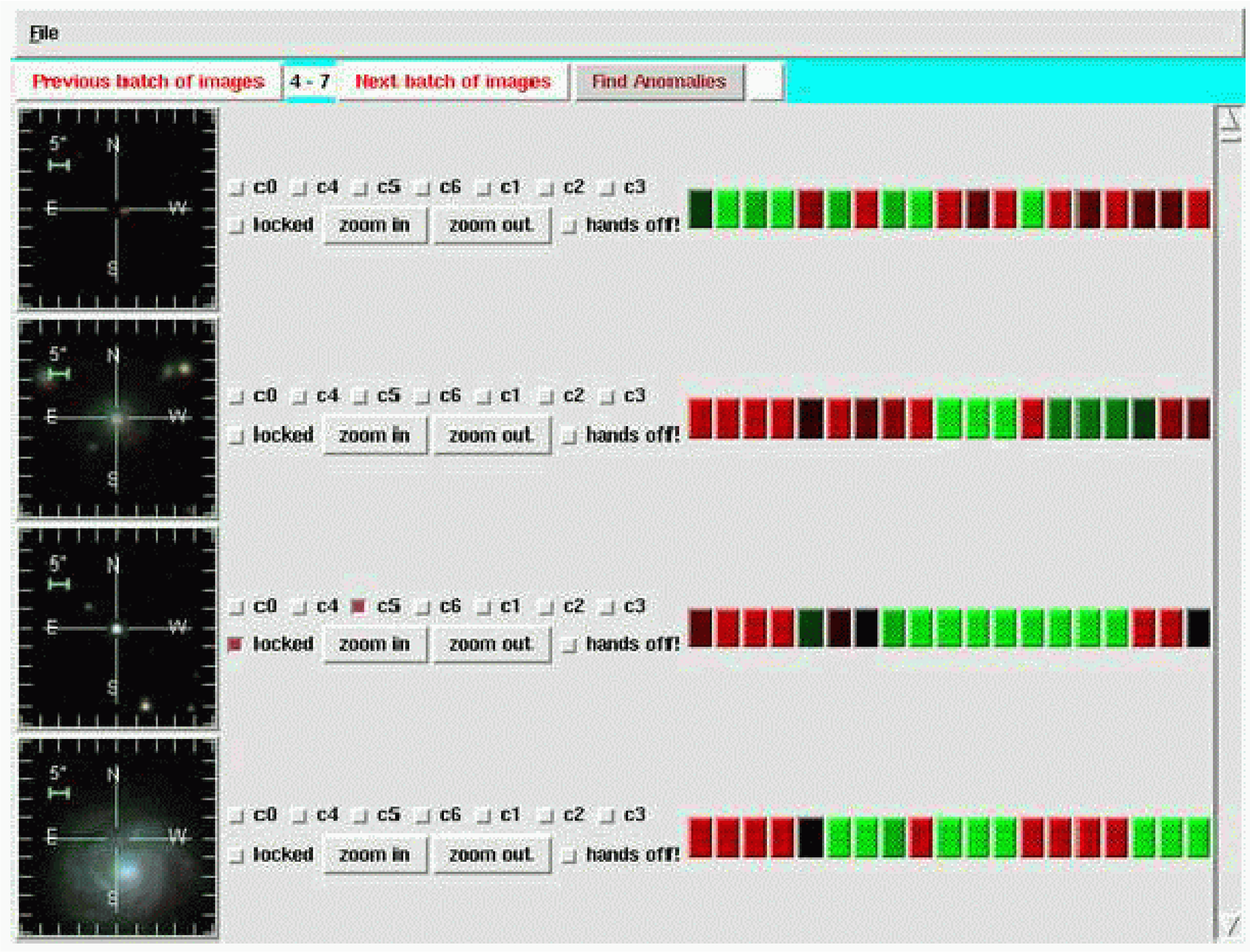
Active Learning of Anomalies



Dan Pelleg

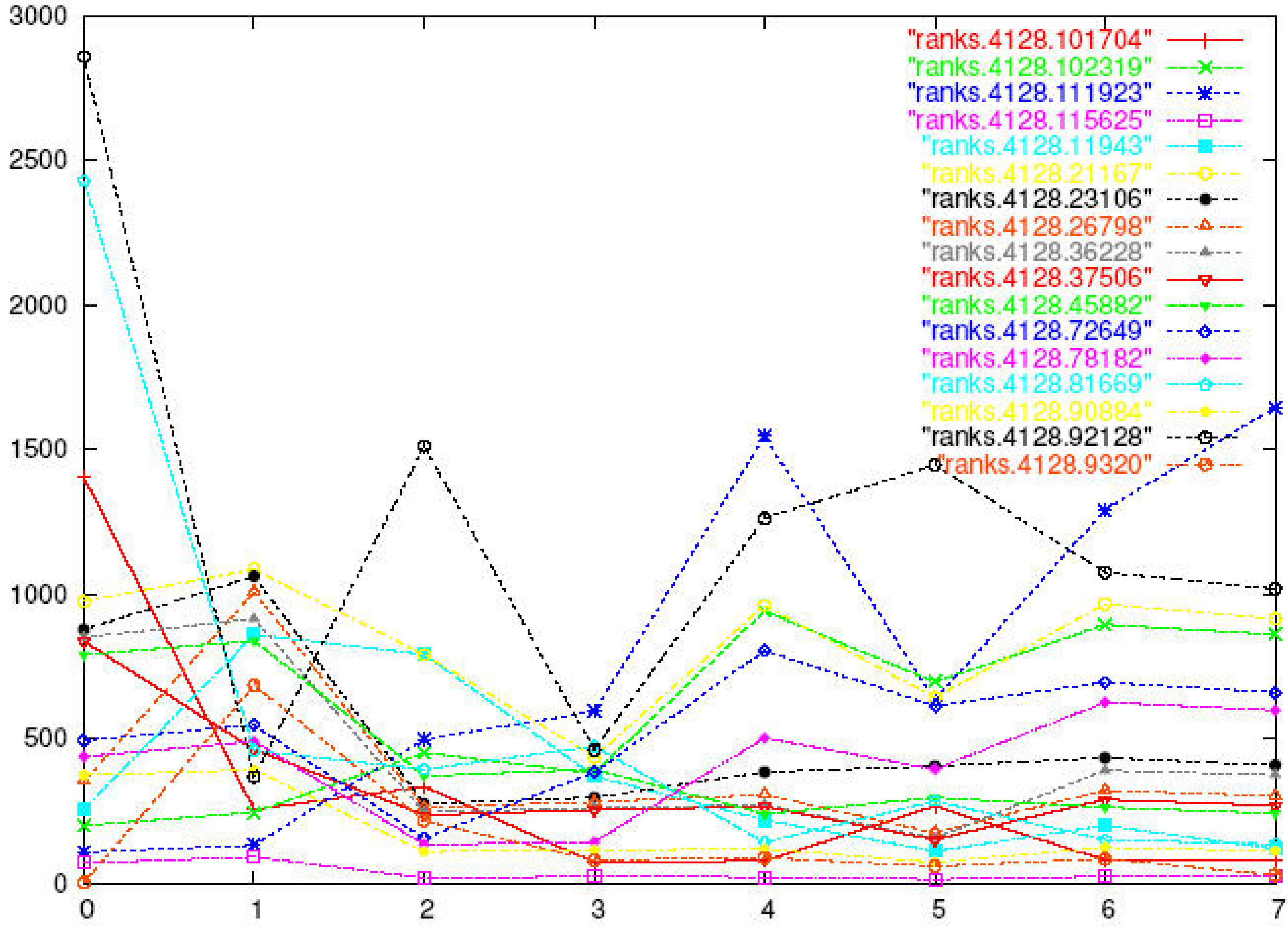
[Pelleg, Moore and Connolly 2004]

Anomaly GUI



File
Previous batch of images
4 - 7
Next batch of images
Find Anomalies
c0 c4 c5 c6 c1 c2 c3
locked
zoom in
zoom out
hands off!
5°
N
E
W
S

Anomaly Performance

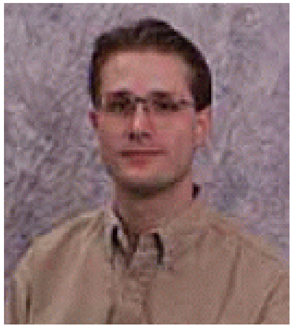


Outline

- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
  - Very fast non-parametric classification
  - skewed binary outputs
  - General binary outputs
  - multi-classed outputs
- Very fast kernel-based statistics
  - n-point computations
  - clustering
  - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- ► GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

GMorph: Fast Galaxy Morphology

- How do you perform 10^7 large nonlinear optimizations in practical time?
- How do you avoid local optima
- Idea: Pre-cache a "library" of solutions. Use efficient nearest neighbor to match new problems to library as seeds.
- Early tests bring galaxy morphology fits down from minutes to sub-seconds



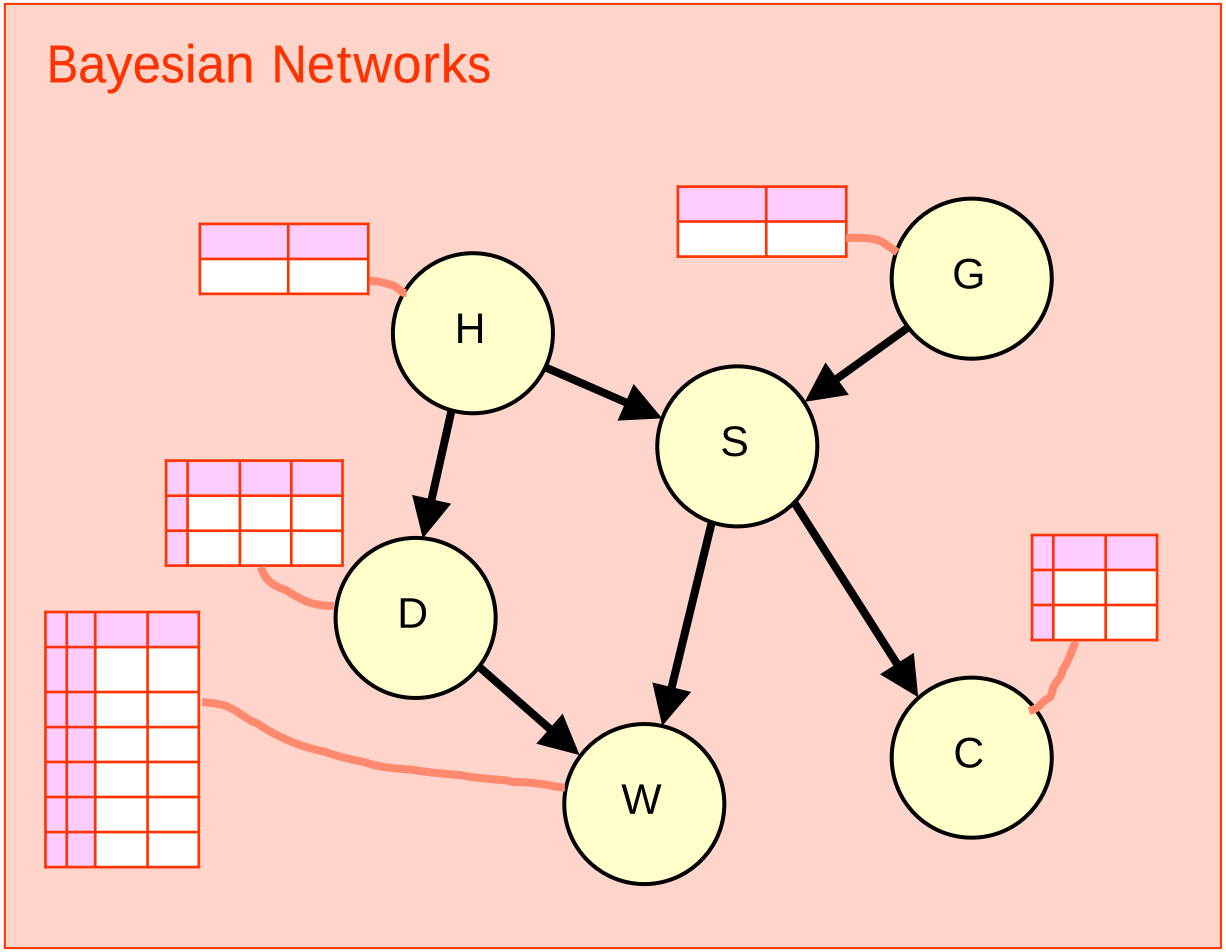
Brigham Anderson

[Anderson, Bernardi, Connolly, Moore, Nichol, 2004]

Outline

- Cached Sufficient Statistics
- Ball Trees (= Metric Trees)
- K-nearest neighbor with ball trees
  - Very fast non-parametric classification
  - skewed binary outputs
  - General binary outputs
  - multi-classed outputs
- Very fast kernel-based statistics
  - n-point computations
  - clustering
  - non-parametric clustering (overdensity hunting)
- Active learning for anomaly hunting
- GMorph: Efficient Galaxy morphology fitting
- Other Auton topics

Other Relevant Auton Topics



[Moore and Lee, 1998], [Moore and Wong, 2003]

Other Relevant Auton Topics

Bayesian Networks

"What's strange about recent events?"

Weng-Keen
Wong

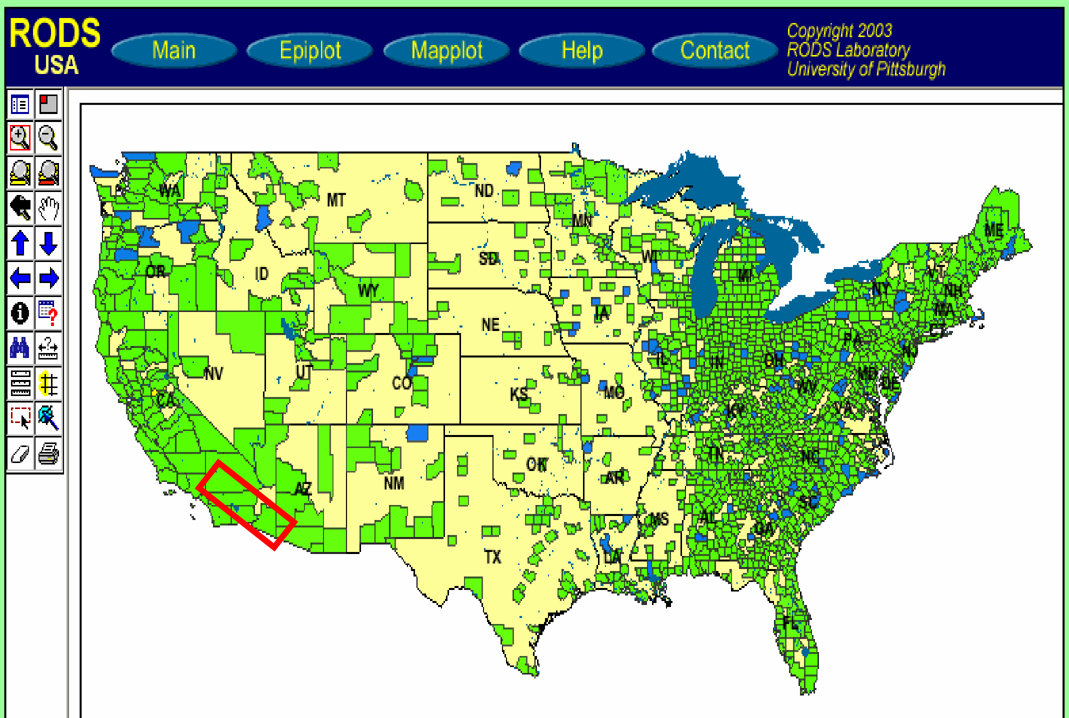
[Wong, Moore, Cooper and Wagner 2003]

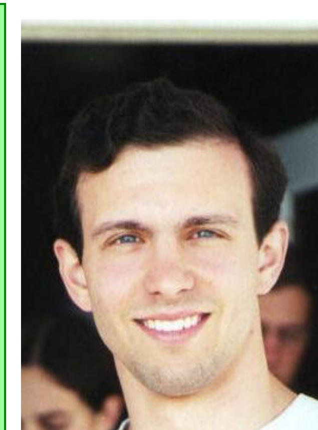
Other Relevant Auton Topics

Bayesian Networks

“What’s strange about recent events?”

Spatial Scan Statistics



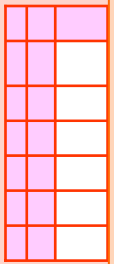


Daniel Neill

[Neill, Moore and Wagner, 2004]

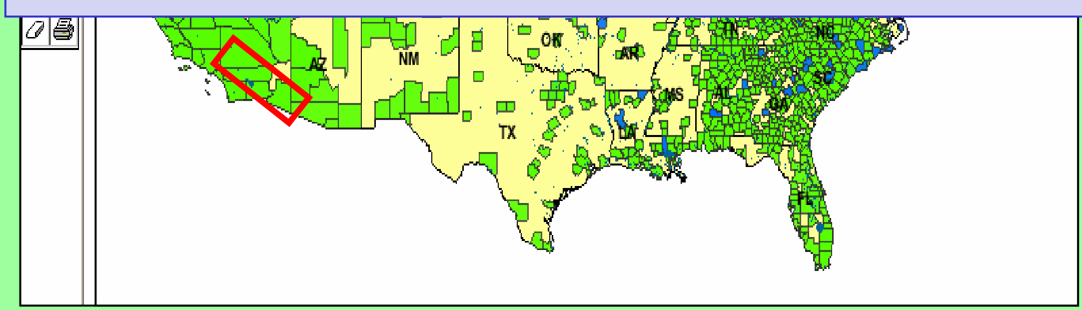
Other Relevant Auton Topics

Bayesian Networks



“What’s strange about recent events?”

Spatial Scan Statistics



Massively multiple target tracking

Jeremy Kubica

Conclusions

- Geometry can help tractability of Massive Statistical Data Analysis
- Cached sufficient statistics are one approach

- Papers, tutorials, software, examples:

www.autonlab.org