Technology for Informatics

Randy Julian
Lilly Research Laboratories

Elements

- Hardware Platforms
  - Computer and network
- Software Platforms
  - Operating systems
  - Programming languages
  - Virtual machines
- Storage Platform
  - File systems
  - Data bases

warning: past experiences may have led to the formation of opinions not shared by everyone.

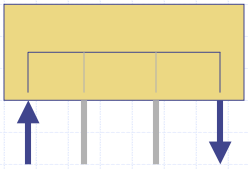
Hardware Platforms - Computers

- Workstation $
  - Vanilla 'PC' (usually Intel based)
- Server $
  - Vanilla PC
  - "Pizza box"
  - 1-4 processors
- "Clusters" Nx$, Nx$$
  - Network of servers
- "Big Iron" $$$$…$$$
  - Specialized hardware
  - Memory bus shared between processors

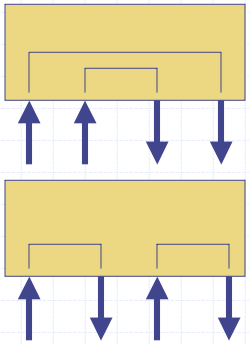
Hardware - Networks

- Standard Ethernet
  - easy to 100 Mbit/sec
- Gigabit Ethernet
  - 1-10 Gbit/sec
  - Cards cost more than the PC…
- Fiber-channel
  - Used between processors and disk sub-systems

Network basics



"Hub": Point-to-point connection (one at a time)

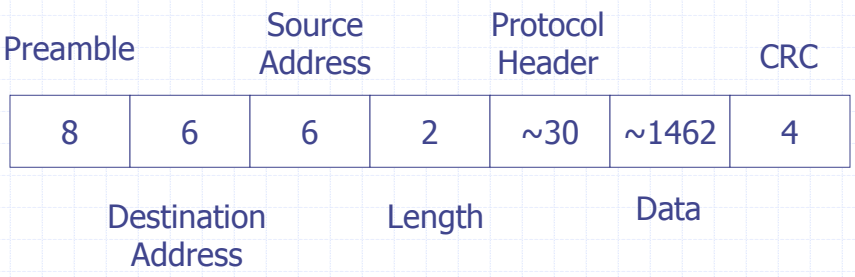


"Switch": Point-to-point connections (multiple)

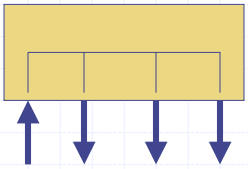
OSI Network Layers

1. Physical - interface to actual wire
2. Data Link ("Ethernet"; Frames)
    1. MAC: Media Access Control (IEEE defined)
    2. LLC: Logical Link Control (between MAC and L3)
3. Network - TCP/IP, IPX,… (packet: >=1 frame)
4. Transport - end-to-end (ack, error correction)
5. Session - OS dependent
6. Presentation - OS dependent
7. Application

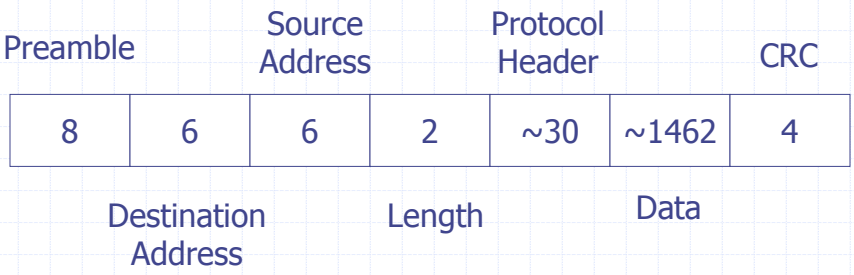
Ethernet Frames (Layer 2)



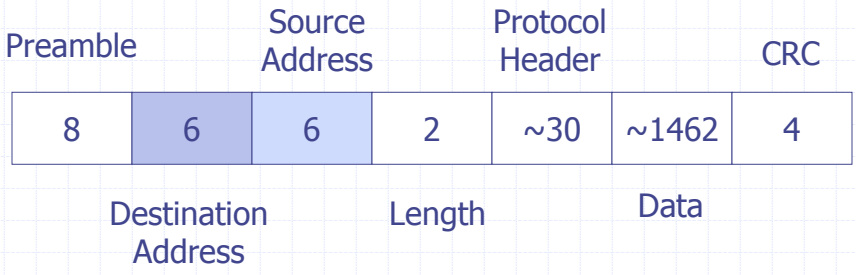
Ethernet Hubs



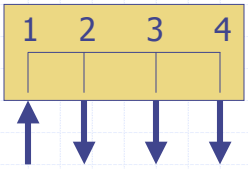
Hub: Get message, send message…



Ethernet Switches (Layer 2 device)

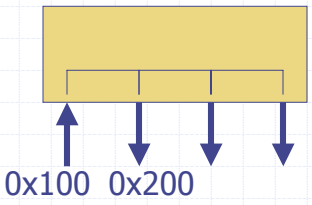


Switch Sequence

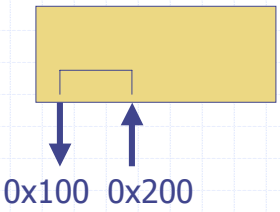


1. Examine destination address
2. Add source address to incoming port list
3. If port lists are empty, send message to all ports

Reply and address logging

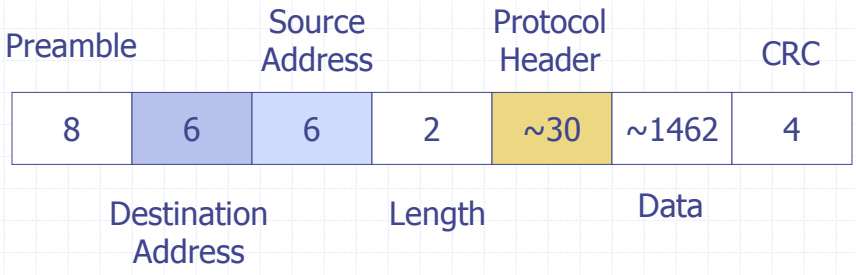


Port 1: Addr 1
Port 2: -
Port 3: -
Port 4: -



Port 1: Addr 1
Port 2: Addr 2
Port 3: -
Port 4: -

Routing (Layer 3)



Routers examine the protocol header and decide where the frame goes based on the 'network' ID

Software platforms: Windows OS

- Windows
  - Single user with resource sharing
  - Pros:
    - Most "expected" user interface
    - Very good single user performance
    - Single design point - de facto standard
    - Very easy to use for 'standard' computer applications
    - Real-time access to hardware is simple
  - Cons:
    - Single vendor controls much more than look-and-feel
    - Networking and multi-processor tools came late
    - Some file size/memory limitations
    - Rumored to be unstable and hard to secure
    - Hard to turn into a server

Software platform: Unix OS

- UNIX
  - Multi-user with system control over resources
  - Pros:
    - Good management of large resources (memory/disk)
    - Free (Linux)
    - Remarkably stable and secure
    - Easy to turn into a server
  - Cons:
    - Hard to use for "standard" computer applications
    - Configuration and management hard
    - Software typically comes as source code…
    - Real-time access to hardware can be problematic

Software platforms: Languages

- Scripting languages (Batch files and Shell Scripts)
  - Almost gone from Windows, primary tool in Unix
- Interpreted languages
  - Perl: general purpose programming tool
- Compiled languages
  - Visual Basic
  - C/C++
- Virtual Machine-based systems
  - Java
  - C#, VB.NET

Perl

- Developed by Larry Wall
  - Practical Extraction and Report Language
  - Pathologically Eclectic Rubbish Lister
- Free for Wintel/Unix
- Intrepreted (changing with 5.8)
- Unparalleled power at string parsing/searching
- Gained popularity as the 'scripting' language for webservers
- Runs everywhere

Visual Basic

- Developed by Microsoft
  - "A rubber crescent-wrench"
- Very fast method for developing user interfaces under Windows
- Very easy to get the GUI 'right'
- Drag-and-drop programming interface
- Relies heavily on external libraries (mostly written in C/C++)
- Application memory and speed performance can be an issue
- Only runs on Windows

C/C++

- C (Ritchie, 72), C++ (Stroustrup, 85)
- Basis for syntax of Perl and Java
- C: Functional/C++: Object oriented
- High performance - total programmer control
- Steep learning curve
- Knowing C does not really help read C++
- Compiled into the native processor code
- Can be ported
- Free (GNU)

Java

- Developed by Sun (Gosling, 95)
  - "Kinda like C, with all the hard stuff taken out"
  - "Java is a platform, not a language…"
  - "Write once, run everywhere"
- Compiled to byte code which is then run on a CPU simulator called a Virtual Machine
- Famous because the VM was incorporated into Netscape
- Pure object-oriented
- Numerically determinant

Virtual Computers

- A program which interprets instructions as if it where hardware
- A program which can be adapted to any hardware
- A program whose behavior is independent of the hardware

Examples of Virtual Computers:

- MAME
- Java Virtual Machine
- Microsoft .NET CLR
- Intel ORP (Open Runtime Platform)

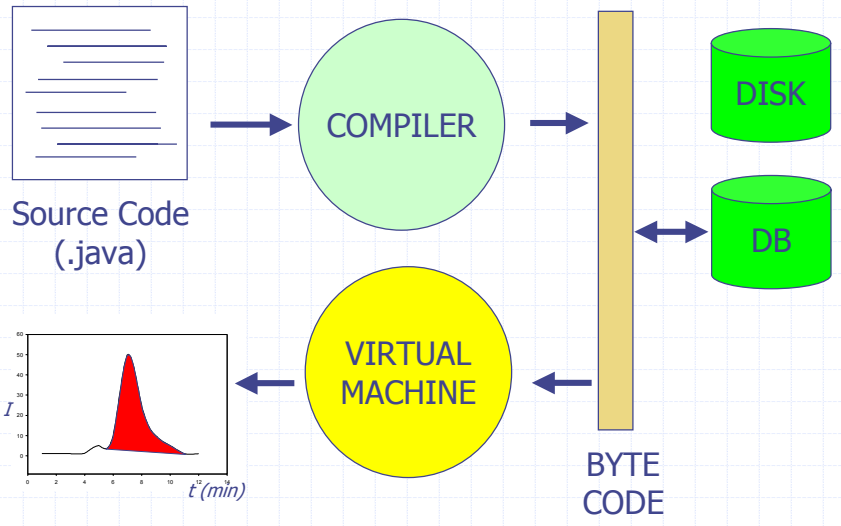
Why Insulate From Hardware?

- Numerical Determinacy
  - Calculations always produce same result
- Application Testing
  - Hidden side effects of hardware changes propagates change in all that it touches
- Productivity
  - Want to be able to plug in latest and greatest without fear

Starting Point: Java Virtual Machine

- Open specification
- Available on multiple platforms
- Numerical determinacy a design spec
- Broad educational support
- Open source (Intel ORP)

How the Java/JVM works



Numerical Determinacy: Integration

Equation	Area Analytical (HP Calc)	Quadrature (Trapazoidal)	Quadrature (Simpsons)	OS	Processor	JVM	Java Version
f(x)=x-x^2	0.166667	0.1666649 99999998	0.1666650 00016667	Windows 2000, 5.0, x86,	Pentium III	Sun HotSpot 1.4.0-b92	1.4.0
		0.1666649 99999998	0.1666650 00016667	Linux, 2.4.7-10, i386	Pentium III	Sun HotSpot 1.3.1-b24	1.3.1
		0.1666649 99999998	0.1666650 00016667	SunOs, 5.8, sparc	Sparc	Solaris VM 1.2.2	1.2.1
f(x)=sin(x)	2.000000	1.9999800 00035478 8	1.9999800 00199978 3	Windows 2000, 5.0, x86,	Pentium III	Sun HotSpot 1.4.0-b92	1.4.0
		1.9999800 00035478 8	1.9999800 00199978 3	Linux, 2.4.7-10, i386	Pentium III	Sun HotSpot 1.3.1-b24	1.3.1
		1.9999800 00035478 8	1.9999800 00199978 3	SunOs, 5.8, sparc	Sparc	Solaris VM 1.2.2	1.2.1

Numerical Determinacy: Regression

Slope	Intercept	S(slope)	S(intercept))	OS	Processor	JVM	Java Version
2.2063675 383922594	0.151351 77219484 364	0.08236538 339233479	0.1249275 080249910 3	Windows 2000, 5.0, x86,	Pentium III	Sun HotSpot 1.4.0-b92	1.4.0
2.2063675 383922594	0.151351 77219484 364	0.08236538 339233479	0.1249275 080249910 3	Linux, 2.4.7-10, i386	Pentium III	Sun HotSpot 1.3.1-b24	1.3.1
2.2063675 383922594	0.151351 77219484 364	0.08236538 339233479	0.1249275 080249910 3	SunOs, 5.8, sparc	Sparc	Solaris VM 1.2.2	1.2.1

C#/VB.NET

- Microsoft Virtual Machine Approach
- New platform aimed at Enterprise Applications
- Uses a Virtual Machine: "Common Runtime Language"
- Easy transition from C++ to C#
- Harder transition from VB to VB.NET

Storage Platforms: File Systems

- Windows can share file systems with other Windows machines
- Unix can share file systems:
  - Unix-to-Unix: NFS
  - Wintel-to-Unix: Samba
- Directory-based systems use a special file format ("directory"/"folder") to point to other files.
- Files can be opened, parsed, written and closed by any language.
- Data can get hard to find if users are naming directories

Storage Platforms: Databases

- Relational Database Concept:
  - A single piece of data goes in a "field"
  - A group of fields make up a record
  - An array of records is a table
  - Contents of one column in the list can be related (are the same as) the contents of another column.
  - Tables can be 'joined' through common contents in particular field.

Employee Table

Name	SSN	StrAddr	City	State	Zip	Salary
jbs	010-00-1111	Sitterson Hall	Chapel Hill	NC	27599	120000
wms	033-53-3902	P.O.Box 3	Binghampton	NY		60000
lkb	037-84-7667	32 Juniper Rd	Bethel	CT	06801	100000
dkb	101-23-5679	1 Vegetarian Dr	Veggie	RI	21218	30000
jbs	505-47-8901	12 Onion Rd	Garlic	PU	90909	1000

Employee Table2

Name	SSN	Dept
jbs	010-00-1111	A32
wms	033-53-3902	A32
lkb	037-84-7667	B01
dkb	101-23-5679	A11
jbs	505-47-8901	A09

Department Table

Dept	DeptName
A09	Multimedia Projects
A11	Software Reuse
A21	New Department
A32	Java Applications
B01	Accounting

Employee Table2 $\bowtie_{Dept=A32}$ Department Table

Name	SSN	Dept	DeptName
jbs	010-00-1111	A32	Java Applications
wms	033-53-3902	A32	Java Applications

$R \bowtie_F S = \sigma_F (R \times S)$

Database Server Systems

- Optimized programs to handle data manipulation
- Normally networked
- Hid complexity of actually managing data
- Use SQL
- Examples:
  - MS Access, mySQL, Oracle, Sybase
- Most use tables to store user names, passwords and permissions for controlling access to data
- Use "Language Bindings" to give languages access

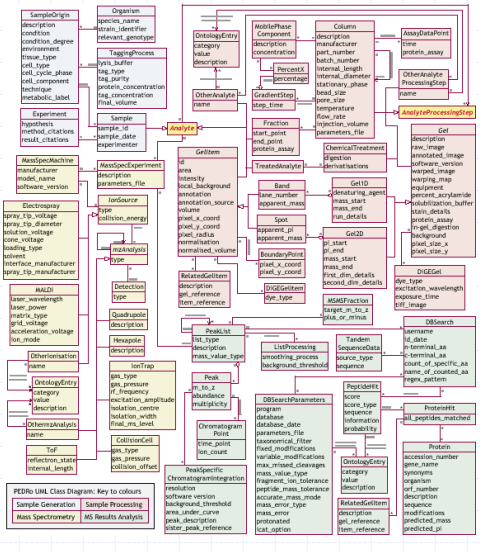
SQL: Structured Query Language

```
Create Table Song
  (Title   varchar(20)  not null,
   Artist  varchar(16)  not null,
   Album   varchar(20),
   Time    char(5)
  );

Insert Into Song
  Values ("Roundabout", "Yes", "Fragile", "9:35");

Insert Into Song (Time, Artist, Title)
  Values ("19:35", "Yes!", "I'll be the Roundabout");

Update Employee
  Set Salary = Salary * 1.2
  Where Evaluation > .85;
```

Examples of databases

- PEDRoML - Proteomics Data
- GSK LIMS

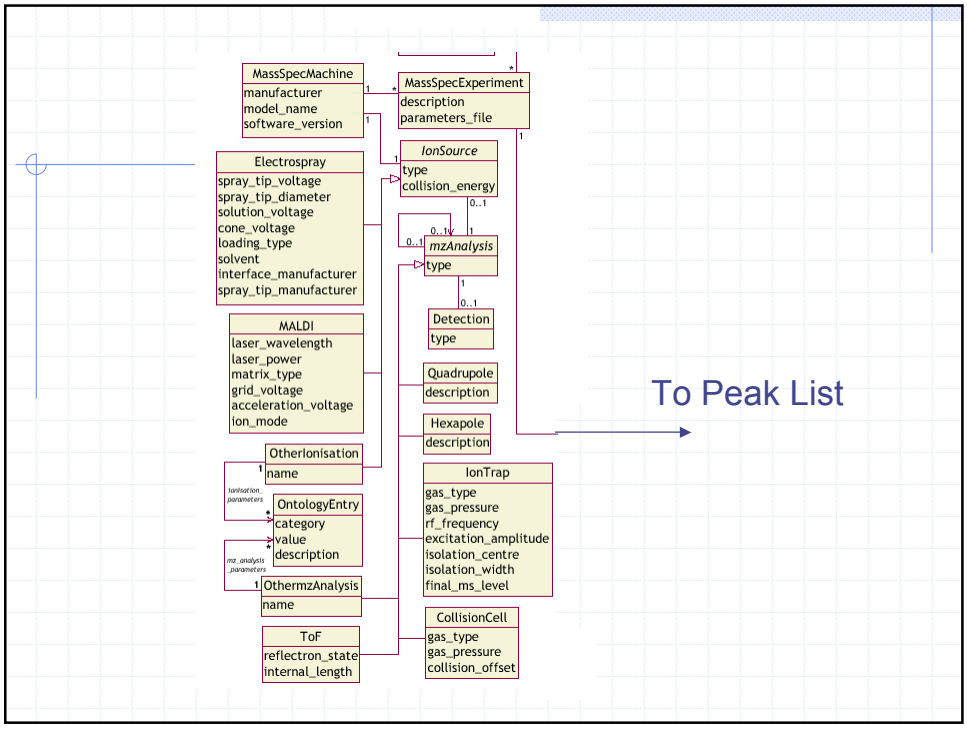
PERDRoML:

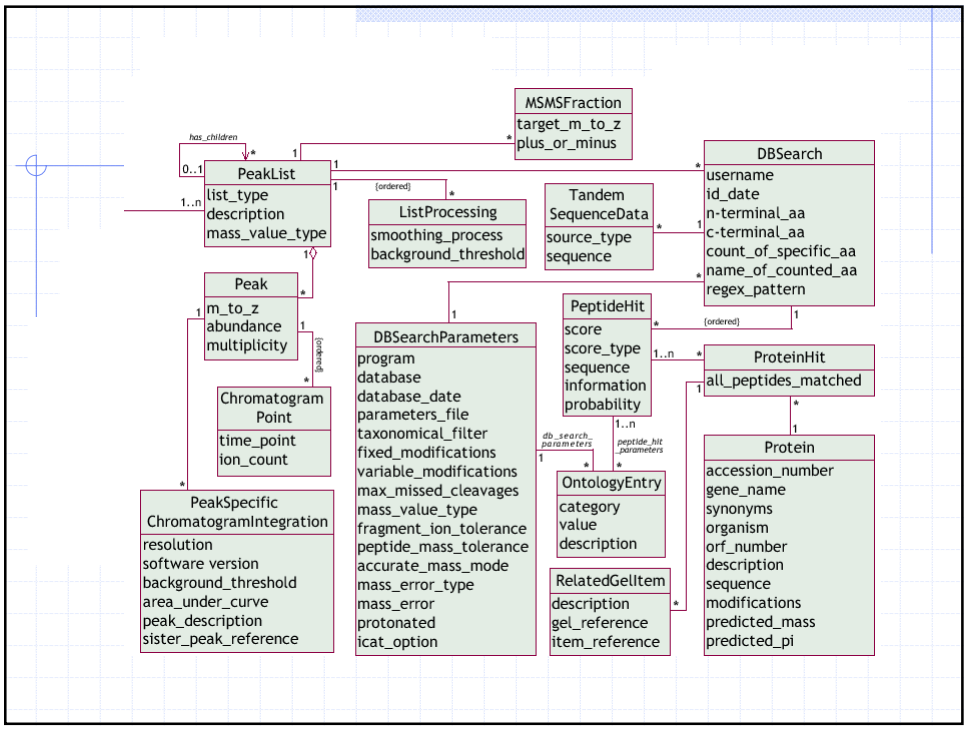


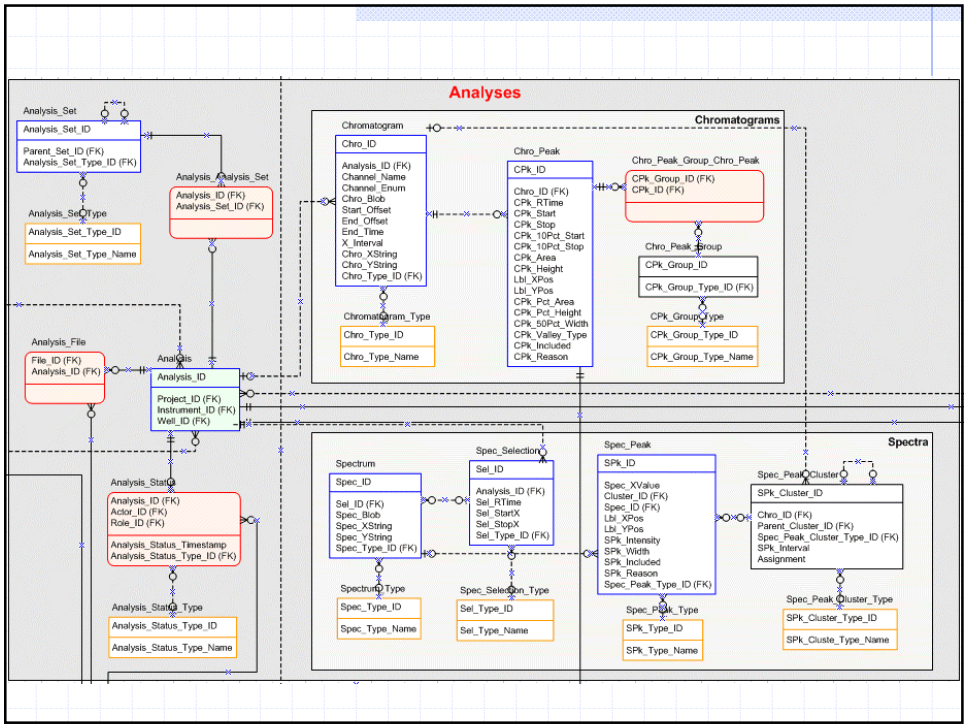
S. Oliver,
C. Taylor
Univ. Manchester
in press

see: pedro.man.ac.uk

57 tables
> 200 fields


MassSpecMachine
manufacturer
model_name
software_version
MassSpecExperiment
description
parameters_file
IonSource
type
collision_energy
Electrospray
spray_tip_voltage
spray_tip_diameter
solution_voltage
cone_voltage
loading_type
solvent
interface_manufacturer
spray_tip_manufacturer
mzAnalysis
type
MALDI
laser_wavelength
laser_power
matrix_type
grid_voltage
acceleration_voltage
ion_mode
Detection
type
Quadrupole
description
Hexapole
description
OtherIonisation
name
IonTrap
gas_type
gas_pressure
rf_frequency
excitation_amplitude
isolation_centre
isolation_width
final_ms_level
OntologyEntry
category
value
description
OthermzAnalysis
name
CollisionCell
gas_type
gas_pressure
collision_offset
ToF
reflectron_state
internal_length
To Peak List


MSMSFraction
target_m_to_z
plus_or_minus
has_children
PeakList
list_type
description
mass_value_type
{ordered}
ListProcessing
smoothing_process
background_threshold
Tandem SequenceData
source_type
sequence
DBSearch
username
id_date
n-terminal_aa
c-terminal_aa
count_of_specific_aa
name_of_counted_aa
regex_pattern
Peak
m_to_z
abundance
multiplicity
Chromatogram Point
time_point
ion_count
DBSearchParameters
program
database
database_date
parameters_file
taxonomical_filter
fixed_modifications
variable_modifications
max_missed_cleavages
mass_value_type
fragment_ion_tolerance
peptide_mass_tolerance
accurate_mass_mode
mass_error_type
mass_error
protonated
icat_option
PeptideHit
score
score_type
sequence
information
probability
ProteinHit
all_peptides_matched
Protein
accession_number
gene_name
synonyms
organism
orf_number
description
sequence
modifications
predicted_mass
predicted_pi
PeakSpecific ChromatogramIntegration
resolution
software version
background_threshold
area_under_curve
peak_description
sister_peak_reference
OntologyEntry
category
value
description
RelatedGelItem
description
gel_reference
item_reference


Analyses
Chromatograms
Spectra
Analysis_Set
Analysis_Set_ID
Parent_Set_ID (FK)
Analysis_Set_Type_ID (FK)
Analysis_Set_Type
Analysis_Set_Type_ID
Analysis_Set_Type_Name
Analysis_Analysis_Set
Analysis_ID (FK)
Analysis_Set_ID (FK)
Analysis_File
File_ID (FK)
Analysis_ID (FK)
Analysis
Analysis_ID
Project_ID (FK)
Instrument_ID (FK)
Well_ID (FK)
Analysis_Status
Analysis_ID (FK)
Actor_ID (FK)
Role_ID (FK)
Analysis_Status_Timestamp
Analysis_Status_Type_ID (FK)
Analysis_Status_Type
Analysis_Status_Type_ID
Analysis_Status_Type_Name
Chromatogram
Chro_ID
Analysis_ID (FK)
Channel_Name
Channel_Enum
Chro_Blob
Start_Offset
End_Offset
End_Time
X_Interval
Chro_XString
Chro_YString
Chro_Type_ID (FK)
Chromatogram_Type
Chro_Type_ID
Chro_Type_Name
Chro_Peak
CPk_ID
Chro_ID (FK)
CPk_RTime
CPk_Start
CPk_Stop
CPk_10Pct_Start
CPk_10Pct_Stop
CPk_Area
CPk_Height
Lbl_XPos
Lbl_YPos
CPk_Pct_Area
CPk_Pct_Height
CPk_50Pct_Width
CPk_Valley_Type
CPk_Included
CPk_Reason
Chro_Peak_Group_Chro_Peak
CPk_Group_ID (FK)
CPk_ID (FK)
Chro_Peak_Group
CPk_Group_ID
CPk_Group_Type_ID (FK)
CPk_Group_Type
CPk_Group_Type_ID
CPk_Group_Type_Name
Spectrum
Spec_ID
Sel_ID (FK)
Spec_Blob
Spec_XString
Spec_YString
Spec_Type_ID (FK)
Spectrum_Type
Spec_Type_ID
Spec_Type_Name
Spec_Selection
Sel_ID
Analysis_ID (FK)
Sel_RTime
Sel_StartX
Sel_StopX
Sel_Type_ID (FK)
Spec_Selection_Type
Sel_Type_ID
Sel_Type_Name
Spec_Peak
SPk_ID
Spec_XValue
Cluster_ID (FK)
Spec_ID (FK)
Lbl_XPos
Lbl_YPos
SPk_Intensity
SPk_Width
SPk_Included
SPk_Reason
Spec_Peak_Type_ID (FK)
Spec_Peak_Type
SPk_Type_ID
SPk_Type_Name
Spec_Peak_Cluster
SPk_Cluster_ID
Chro_ID (FK)
Parent_Cluster_ID (FK)
Spec_Peak_Cluster_Type_ID (FK)
SPk_Interval
Assignment
Spec_Peak_Cluster_Type
SPk_Cluster_Type_ID
SPk_Cluste_Type_Name