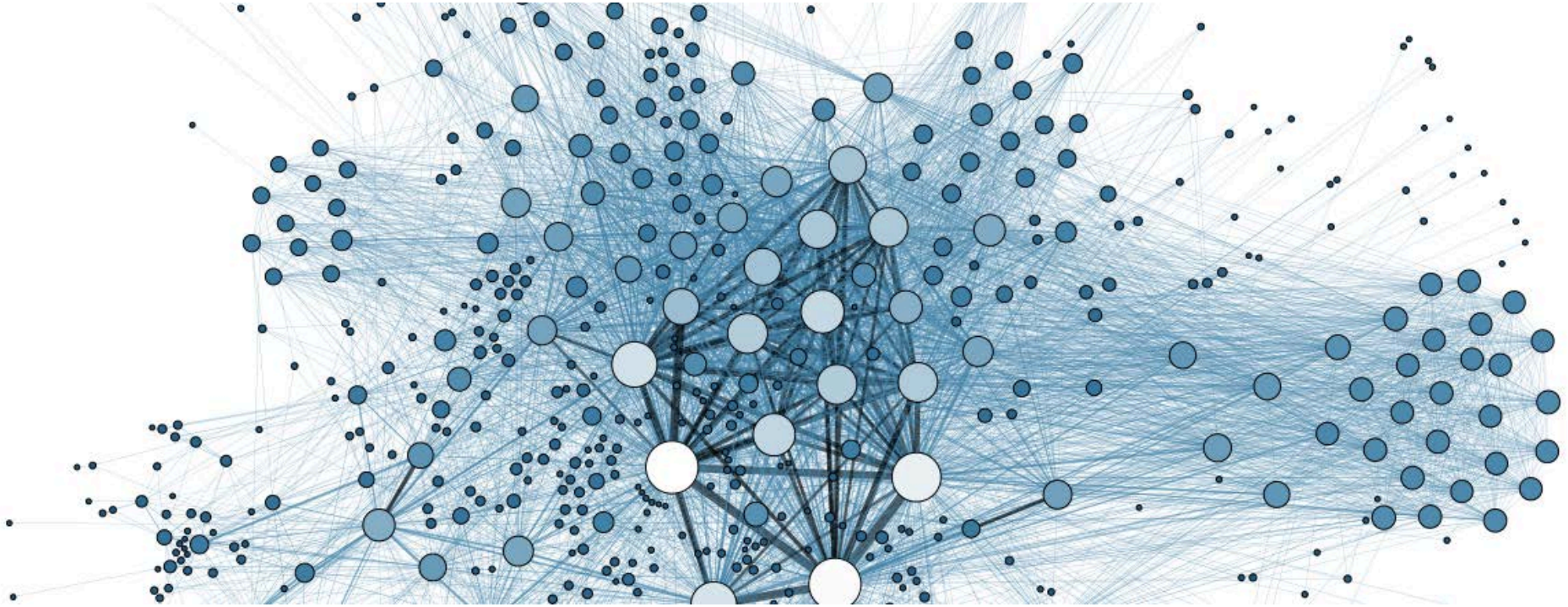
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

FACULTY OF
SCIENCE



Practical Course: Integrative Bioinformatics

Summer Term 2017

August 21 – September 1

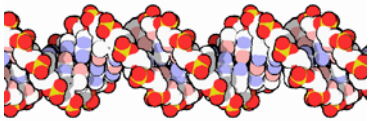
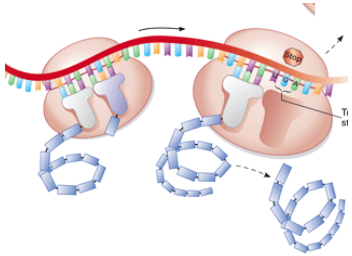
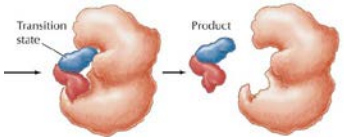
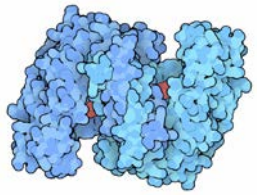
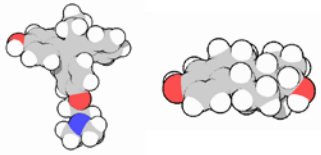
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Welcome

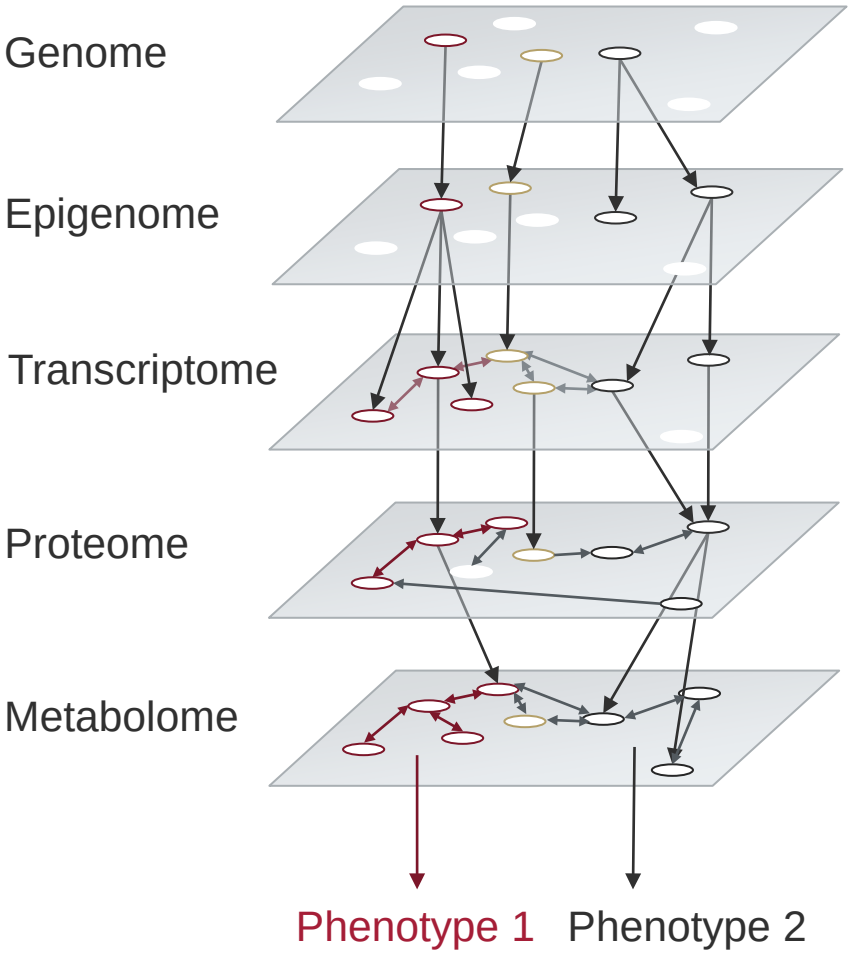
Photo: Shutterstock

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Central Dogma of Biology (extended edition)

Central Dogma	Omics-Layer	Molecule	Measurement
Transcription 	Genome		NGS
Translation 	Transkriptome		RNA-seq (NGS) Microarray
Metabolism 	Proteome		Mass spectrometry
	Metabolome		Mass spectrometry NMR

Trans-Omics Studies



- Until recently
  - Analysis of each level in central dogma studied individually
- Each level of molecules in the central dogma contain orthogonal information
  - With cheap and fast methods for data acquisition now try to include multiple levels to get the **big picture** of what is going on in cell
- Idea
  - Measure across multiple omics-layers
  - Identify connections within each layer
  - Identify connections between layers and phenotype

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Structure of the Practical Course

Goal

Learn how to integrate biological data from multiple sources and omics-layers for integrative analysis

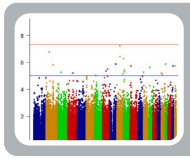
Procedure

- Intro presentation each morning (+afternoon)
- Work on tasks for the day in jupyter notebooks
- Commit progress to github

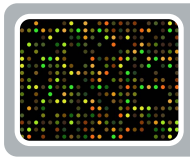
Requirements

- Attendance on all days
- Project report (one **tidy** notebook per topic)

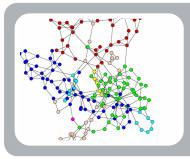
Day 1/2:
Data import/export, version control



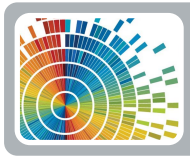
Day 3/4:
Genome-wide association studies



Day 5/6:
Differential expression analysis



Day 7/8:
Biological pathways

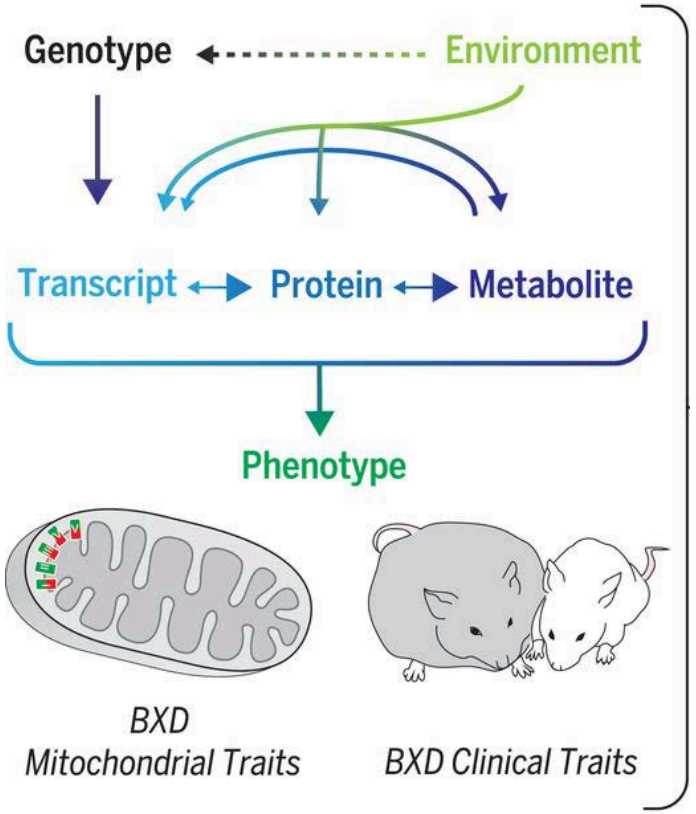


Day 9:
eQTLs

Day 10: Report

Data Set

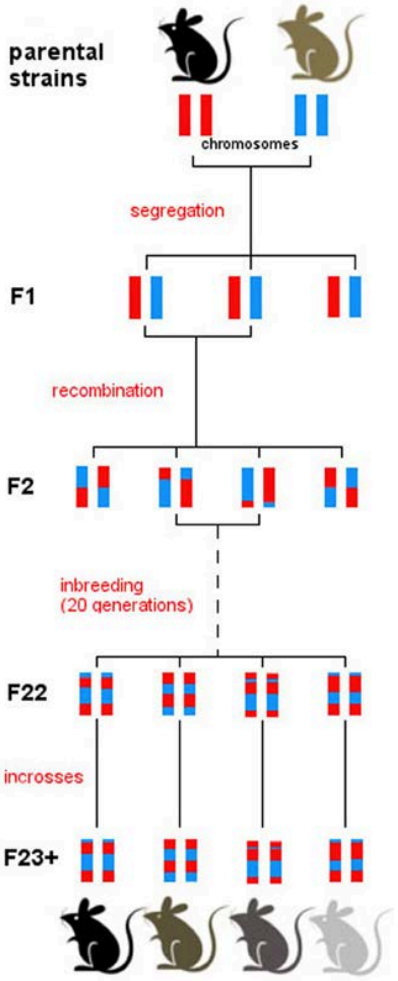
- Williams et al., **Systems proteomics of liver mitochondria function**, Science 352(6291):aad0189, June 10, 2016
- Wang et al., **Joint mouse–human phenome-wide association to test gene function and disease risk,** Nature Communications 7, 2016, Article number: 10464
- Andreux et al. **Systems Genetics of Metabolism: The Use of the BXD Murine Reference Panel for Multiscalar Integration of Traits**, Cell 150, September 14, 2012



Williams et al. Science, 2016

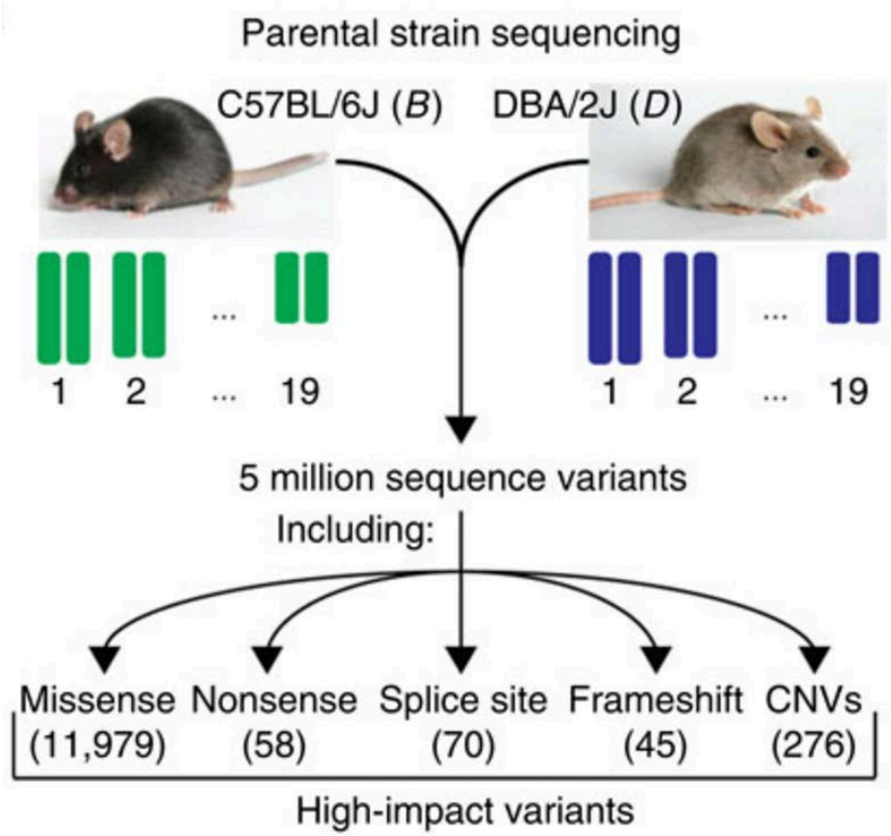
BXD mouse strains

- Cross-bred between female C57BL/6J (B6) and male DBA/2J (D2)
  - Parents fully sequenced
  - Parents differ at 4.8M SNPs
  - Inbred for 20+ generations
- Generated in 4 distinct time frames
- 100+ different strains
- Well characterized
  - 4300 phenotypes characterized (categorical and quantitative)
  - Various expression levels measured in different tissue types



Fini and Hager, Frontiers in Genetics, 2012

Data Set: Genome



Wang et al. Nat Comms, 2016

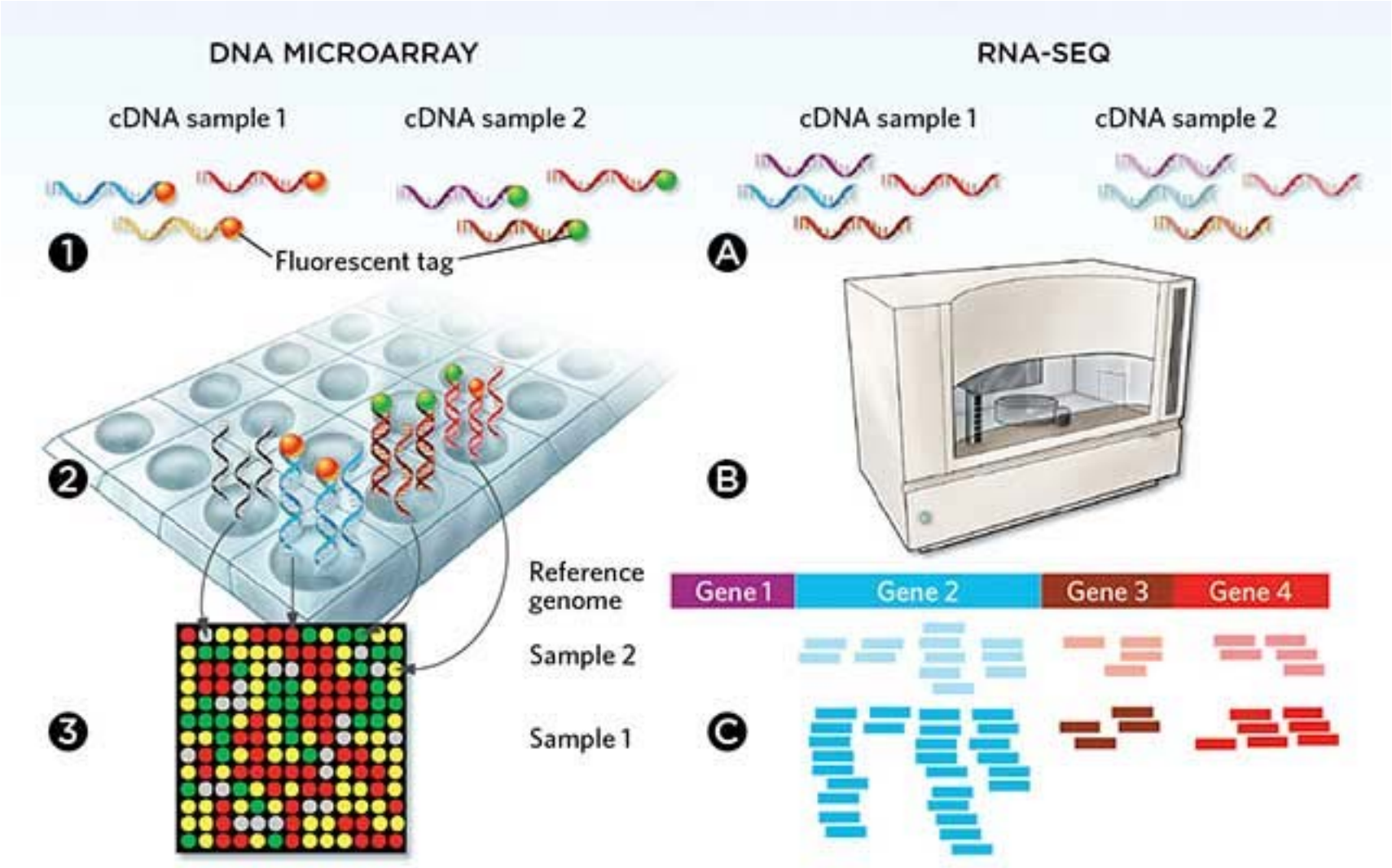
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Genotype File Format

```
@name:BXD
@type:riset                    Genotype collected from GeneNetwork.org
@mat:B                         Full sequence analysis of B vs D can be found in this publication:
@pat:D                         """Joint mouse-human phenome-wide association to test gene function an
@het:H
@unk:U
```

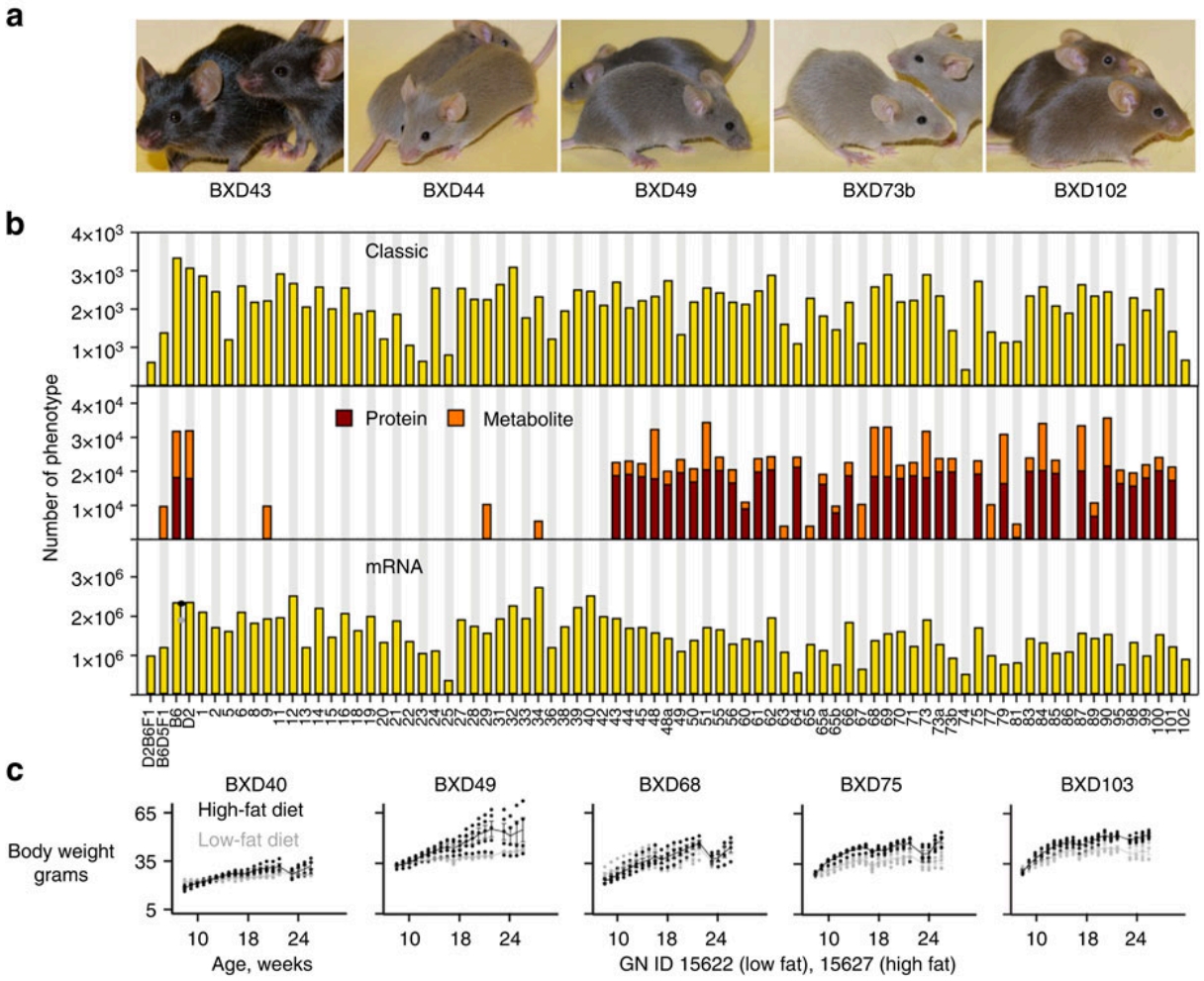
Chr	Locus	cM	Mb	BXD1	BXD2	BXD5
1	rs6269442	0	3.482275	B	B	D
1	rs6365999	0	4.811062	B	B	D
1	rs6376963	0.895	5.008089	B	B	D
1	rs3677817	1.185	5.176058	B	B	D
1	rs8236463	2.081	5.579193	B	B	D
1	rs6333200	2.081	6.217921	B	B	D
1	rs6298633	2.367	6.820241	B	B	D
1	rs6241531	2.367	9.995925	B	B	D
1	rs6360236	3.263	11.073904	B	B	D
1	rs3722996	3.263	11.259432	B	B	D
1	D1Mit1	3.549	11.505582	B	B	D
1	D1Mit294	3.836	11.731387	B	B	D
1	rs13475728	3.836	12.71128	B	B	D
1	rs3655978	5.797	13.37307	B	B	B
1	rs3654866	5.797	13.697098	B	B	B
1	rs3669485	6.083	13.975252	B	B	B
1	rs3713198	6.083	14.464944	B	B	B
1	rs6291839	6.675	14.787217	B	B	B
1	rs13475735	6.675	14.977874	B	B	B
1	rs3088964	6.962	15.196882	B	B	B
1	rs13475737	6.962	15.444839	B	B	B
1	rs3678179	7.248	15.498263	B	B	B
1	rs6201380	7.248	15.801731	B	B	B

Data Set: Transcriptome



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Data Set: Phenome



CLAMS: respiration measurements
HGTT: Glucose tolerance test
NIBP: Noninvasive blood pressure measurements
Cold response test
TSE: Basal activity recording
VO2Max: respiratory measurements and distance on treadmill
Activity Wheel: ad libitum access to activity running wheel
Tissue weight
Biochemistry: Biochemistry measurements
Hematology: Cellular analysis of blood after killing

Wang et al. Nat Comms, 2016

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Phenotype File Format

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	@format=column	CD_DiastolicBP_NIBP_[mmHg]	SE	N	HFD_DiastolicBP_NIBP_[mmHg]	SE	N	CD_SystolicBP_NIBP_[mmHg]	SE	N	HFD_SystolicBP_NIBP_[mmHg]	SE	N
2	**C57BL/6J**	93.5	4.140393356	8	77.1	3.078946364	8	117.0	4.140393356	8	111.5	5.483742205	8
3	**DBA/2J**	87.5	3.807886553	8	80.3	3.132491022	8	115.0	3.807886553	8	108.0	3.273268354	8
4	**BXD1**	89.2	5.453439282	5	78.2	6.938299503	5	115.4	5.887274412	5	104.6	5.065570057	5
5	**BXD2**	77.0	6.570134448	4	76.2	3.839270764	5	109.8	3.859512059	4	107.8	3.2	5
6	**BXD6**	76.8	4.993996396	5	69.4	4.34281015	5	102.6	3.264965543	5	96.2	4.352011029	5
7	**BXD8**	83.8	5.921359641	4	70.0	3.31662479	4	111.0	6.757711644	4	95.8	2.672077843	5
8	**BXD11**	73.0	4.582575695	3	66.4	4.34281015	5	102.7	3.282952601	3	92.4	2.249444376	5
9	**BXD12**	84.0	7.778174593	5	70.2	4.164132563	5	109.6	5.211525688	5	101.6	6.071243695	5
10	**BXD16**	78.0	4.708148964	4	89.3	4.216370214	6	108.5	3.570714214	4	113.0	3.812260921	6
11	**BXD27**	83.8	5.132250968	5	90.5	0.645497224	4	107.8	4.715930449	5	108.3	1.701714821	4
12	**BXD32**	65.4	6.79	10	62.0	4.04	6	103.8	4.48	10	108.0	4.4	6
13	**BXD34**	75.0	4.123105626	4	80.0	7.355270219	5	107.8	3.567795771	4	117.0	6.276941931	5
14	**BXD39**	58.0	4.4	8	60.4	3	10	89.4	4.9	10	96.4	1.7	10
15	**BXD40**	99.0	1.7	6	76.0	4.78	8	121.0	4	6	112.8	7.43	8
16	**BXD43**	68.5	2.31	6	67.3	4.5	6	97.3	2.43	6	101.7	2.7	6
17	**BXD44**	79.6	6.281349946	8	82.6	6.059047249	8	107.8	6.281349946	8	126.9	4.918759643	8
18	**BXD45**	88.9	3.961590139	8	78.7	4.794412848	7	111.9	3.961590139	8	101.9	4.589992071	7
19	**BXD48**	77.9	5.9	7	73.8	9.43	8	110.0	3.5	8	107.2	7.5	8
20	**BXD48a**	77.8	4.8	6	74.0	3.405877273	6	109.7	3.15	6	111.3	3.826806037	6
21	**BXD49**	68.3	5.7	8	73.8	9.4227	8	97.0	4.1	8	107.2	7.475	8
22	**BXD50**	58.6	4.2	8	73.8	3.4	8	96.0	4.4	8	105.6	2.64	8
23	**BXD51**	75.6	2.235568799	8	62.5	4.747179614	8	101.6	2.235568799	8	98.6	3.375	8
24	**BXD53**	81.2	4.066939882	5	75.2	6.414047084	5	110.2	3.773592453	5	105.6	7.131619732	5
25	**BXD55**	93.0	1.224744871	4	87.8	9.498903445	4	122.0	1.224744871	4	116.0	3.807886553	4
26	**BXD56**	73.6	4	8	85.2	5.8	8	104.0	4.34	8	119.9	3.6	8
27	**BXD60**	91.6	4.296	8	93.2	4.1	10	117.2	3.723	8	122.4	1.5	10
28	**BXD61**	79.9	2.572750983	7	79.8	3.514002602	8	111.0	2.572750983	7	110.4	4.690177807	8
29	**BXD62**	74.3	3.803483656	8	74.0	4.032001165	8	103.5	2.562557582	8	101.5	2.99114182	8
30	**BXD63**	93.1	4.4	8	76.0	3.882193783	8	114.1	3.1	8	105.6	3.343103242	8
31	**BXD64**	89.8	7.06	8	97.3	3.4	7	120.0	5.61	8	121.8	2.72	7
32	**BXD65**	80.9	4.17	8	79.5	2.52	8	109.5	4.87	8	101.0	1.9	8
33	**BXD65a**	83.4	7	8	81.4	5.9	8	108.3	5	8	111.3	4.5	8
34	**BXD65b**	73.4	1.778	8				102.6	1.6437	8			
35	**BXD66**	75.8	2.962564044	8	76.8	5.174629041	8	105.1	2.6554358	8	106.5	2.464026902	8
36	**BXD67**	86.2	5.323532662	5	70.4	3.043024811	5	110.6	2.942787794	5	95.2	1.15758369	5
37	**BXD68**	78.1	2.82	7	83.8	3.7	8	106.6	1.33	7	109.1	2.3	8

What data will we use in the course

- Integrate data
  - genotype
  - phenotype
  - gene expression
  - known metabolic and signaling networks

- Challenges
  - data formats
  - data types
  - different identifiers
  - missing data
  - large number of data points -> visualization
  - reproducibility

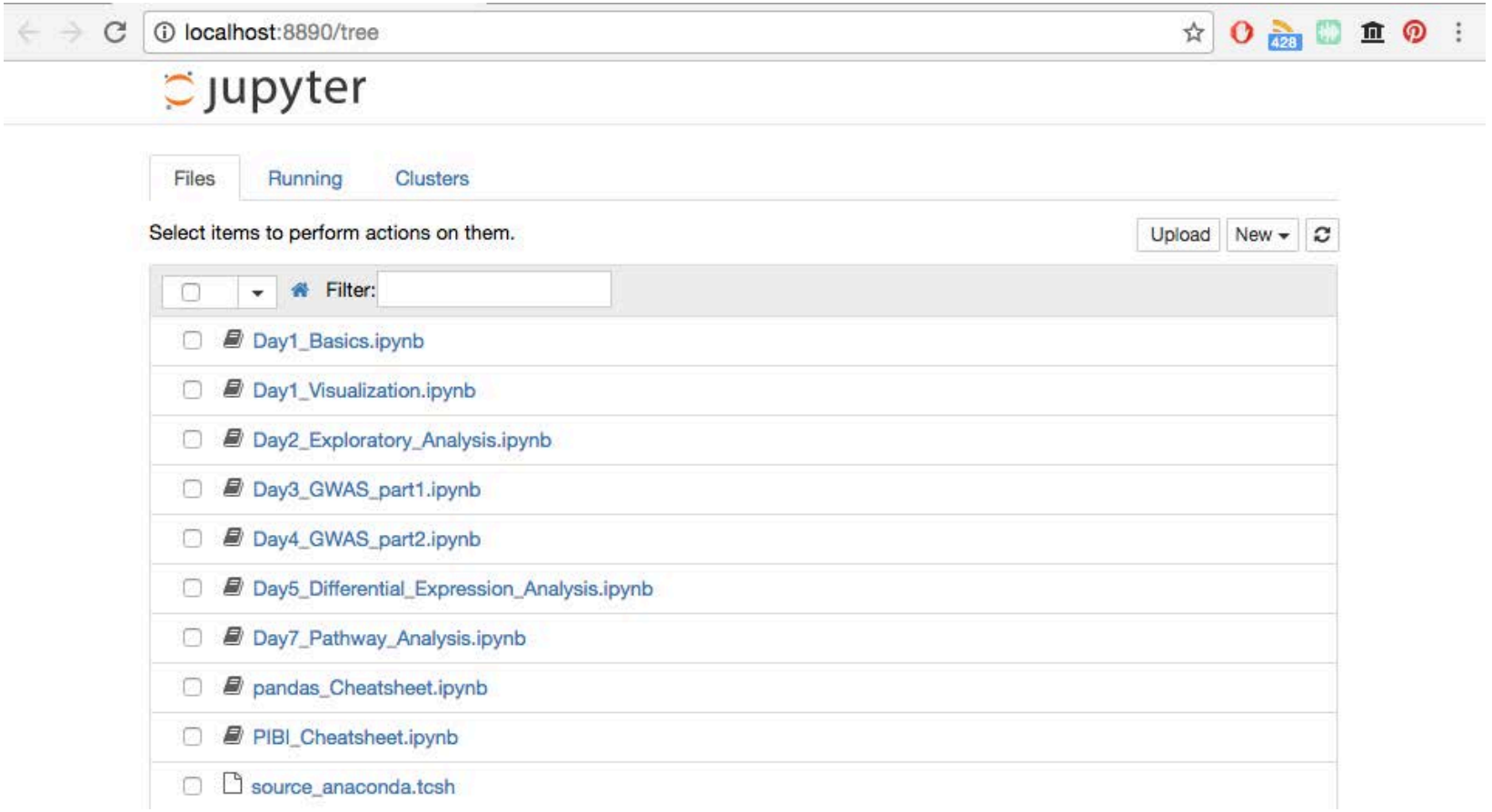
Specific topics of the course

- Data characterization
- Comparison of differences in traits across different environments
- Association between gene variants and phenotypic traits
- Association between gene variants and gene expression
- Differential gene expression analysis
- Relation of gene expression changes with potential effects in cellular processes (i.e., pathways)
- Visualization and reporting of results from biological data analysis

TECHNICAL INTRO

Jupyter Notebooks

- Run interactive python sessions in the browser and save results
- Start from command line with **`jupyter notebook`**



Pandas

- Python data analysis library

- NumPy developed for numerical computing tasks
- Pandas developed for integrated data analysis

- Combine speed and memory efficiency of numpy arrays and matrices with R-style data frames into own data structure: **Series** and **DataFrame**
  - Series: 1D array of indexed data
  - DataFrame: 2D array with flexible row indices and column names

Pandas

- Offers many **vectorized** methods for data manipulation
- Pandas objects are automatically rendered as HTML tables in Jupyter notebooks

```
import pandas as pd

# Show pandas version
print pd.__version__
```

Pandas: Series objects

```
# One way to create a Series: from a list
s = pd.Series([0.25, 0.5, 0.75, 1.0])

# Obtain Series values as numpy array
s.values

# Obtain Series index as pd.Index object
s.index

# Access individual elements by index
s[1]

# Access slice of data by index
s[1:3]

# One way to create a Series with defined index
s = pd.Series([0.25, 0.5, 0.75, 1.0],
              index=['a', 'b', 'c', 'd'])
```

Pandas DataFrame objects

```
# One way to create a DataFram: from a list of lists
df = pd.DataFrame([[0.25, 0.5], [0.75, 1.0]],
                  columns=['a', 'b'])

# Another way to create a DataFrame: from row dicts
df = pd.DataFrame([dict(a=0.25, b=0.5), dict(a=0.75, b=1.0)])

# Obtain DataFrame values as numpy array
df.values

# Obtain DataFrame index as pd.Index object
df.index

# Obtain DataFrame column names as pd.Index object
df.columns

# Access individual elements by index
df.ix[0, 'a’]

# Slice rows by index
df[:10]
```

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

FACULTY OF
SCIENCE

Version Control and Git

Version Control

- Track Changes
- Allows for collaborative development
  - Branching
  - Merging
  - Tagging
  - …
- Allows you to revert to a previous state

- Typically, one central repository on a server where clients push to (CVS, SVN)

Git

- Distributed version control system
- Created by Linus Torvalds

- No central repository
- Users keep entire code and history in local repository
- Network only required to push and pull changes from another repository

- Key concepts:
  - Snapshots
  - Commits
  - Repositories
  - Branches

Important Concepts

- A **snapshot** is a record of all files in the project at a given point in time
  - You decide when to take a snapshot
  - You can go back to **checkout** any snapshot

- A **commit** is the act of creating a snapshot and contains information about
  - How the files changed from the previous snapshot
  - A reference to the previous ‘parent’ commit
  - A hash code

Important Concepts

- A **repository** is a collection of all files and their history
  - Contains all commits
  - Can be local or remote (GitHub)
  - Cloning a repo downloads all the files into a local repo
  - Repos can push changes to or pull from another repository

- All commits live in a **branch**
  - There can be many branches
  - Typically, the main branch is called ‘master’

Try it out!
http://onlywei.github.io/explain-git-with-d3

Basic Workflow

1. Pull changes from GitHub (git pull)
2. Repeat:
    1. Change code
    2. Add changed files to staging area (git add)
    3. Commit with human readable comment (git commit)
3. Push changes to GitHub (git push)

GitHub

- Largest web-based github repository hosting service
- Provides both public (free) and private repositories
- You should already have a student account

- Adds extra functionality which is super useful:
  - A nice user interface
  - Documentation with markdown
  - **Supports jupyter notebooks**
  - Issue tracking
  - Wiki
  - Push and Pull requests
  - Continuous Integration
  - …

Tasks for this morning

1. Log into github
2. Fork the course repository
   https://github.com/enetz/IntegratedBioinformatics
3. Clone your forked repository
4. Start jupyter notebook server from within your notebook directory
   > cd <your dir>
   > jupyter notebook
5. Start notebook Day1_Basics.ipynb
6. Take interactive tour of jupyter UI
7. Follow tasks in notebook
8. Commit and push your progress to your github repository