KSBi – BIML 2020

Bioinformatics & Machine Learning (BIML)

Workshop for Life Scientists

AI in epigenomics and network biology

Sun Kim

2020.02.01

Contents

- Multi-omics analysis resources
- Machine learning basics
- Machine learning models
- Network-based methods
- Representation or hidden feature learning
- Clustering methods for multi-omics integration
- Tutorial

Multi-omics analysis resources

Multi-omic resources

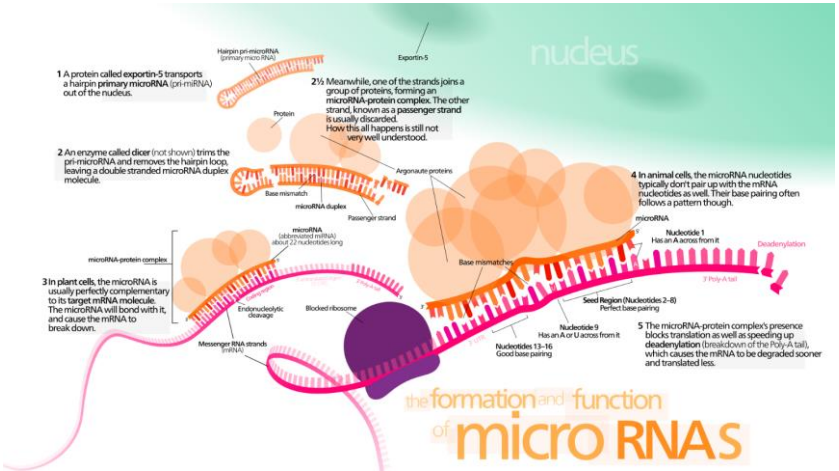
1. microRNA and target gene identification
2. DNA methylation analysis and downstream gene regulation
3. Combinatorial histone modification markers
4. TF binding site analysis and TF-TG network
5. PPI network

Multi-omic resources (1/5)

microRNA and target gene identification

microRNA and target gene identification

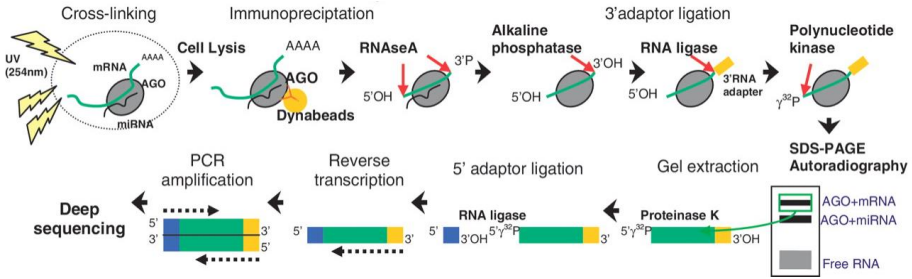
- microRNA is a small non-coding RNA molecule (~22 nt) that negatively regulates gene expression in the post-transcriptional level
- As microRNA hybridize with its complementary mRNA sequence, translation of mRNA is suppressed by degrading mRNA molecules or reducing the effectiveness of ribosome activity



microRNA and target gene identification

- Experimental identification

  - CLIP(crosslinking and immunoprecipitation)-seq (i.e. HITS-CLIP)
    : High-throughput experimental method



Thomson et al. 2011. Nucelic Acid Res.

  - However, experimental identification is
    - time-consuming process
    - The effect of miRNAs on their targets is very small (less than 2-fold in general), which means that many experimentally detected interactions may be phenotypically inconsequential.
      → Experimentally identified but non-meaningful interactions (RNA biology 14.7 (2017): 831-834.)

microRNA and target gene identification

- Computational target prediction
  - Predict miRNA-mRNA interactions based on sequence complementarity
  - Given seed sequence of miRNA, identify potential base-pairing with mRNA
    [Additional features]
    - Target site conservation
      : conservation of the miRNA binding site in mRNA sequence is considered
      : Higher degree of conservation, higher prediction score
      e.g.) DIANA-micorT, PicTar, miRanda, TargetScan
    - Thermodynamic stability
      : thermodynamic stability of miRNA-mRNA pair is considered since mRNA silencing enzyme, called RNA-induced silencing complex, need sufficient time for activation
      : Higher thermodynamic stability, higher prediction score
      e.g. DIANA-micorT, PicTar, PITA

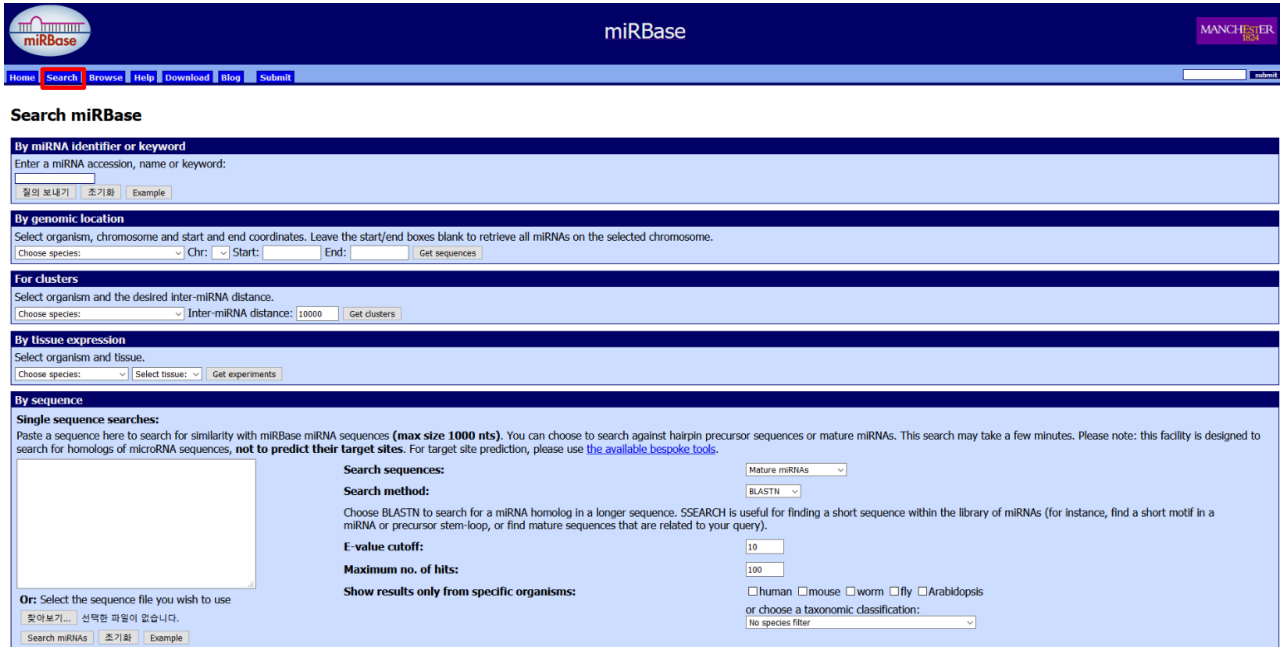
(RNA Biol. 2014 Apr 1; 11(4): 324–333.)

microRNA and target gene identification

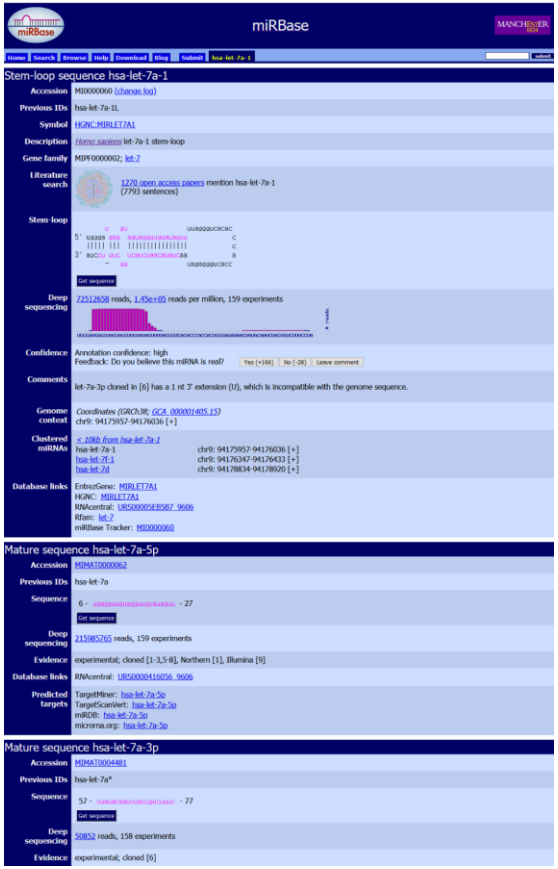
Database name	Description
miRBase	• Searchable database of published miRNA sequence and annotation
miRDB	• Offers miRNA target prediction and functional annotations. • Computationally predicted interactions from mirTarget
miRTarBase	• Experimentally validated vertebrate miRNA-target interactions
TargetExpress	• Computationally predicted interactions combining multiple predictors (TargetScan, microT-CDS, MIRZA) via machine learning
TargetScan	• Computationally predicted interactions considering sequence complementarity and target site conservation • Offers database for Mouse, Worm, Fly, Fish, and Human

microRNA and target gene identification-mirBase

- Database for microRNA sequence and annotations
- Cover 38,589 microRNAs
- http://www.mirbase.org/index.shtml



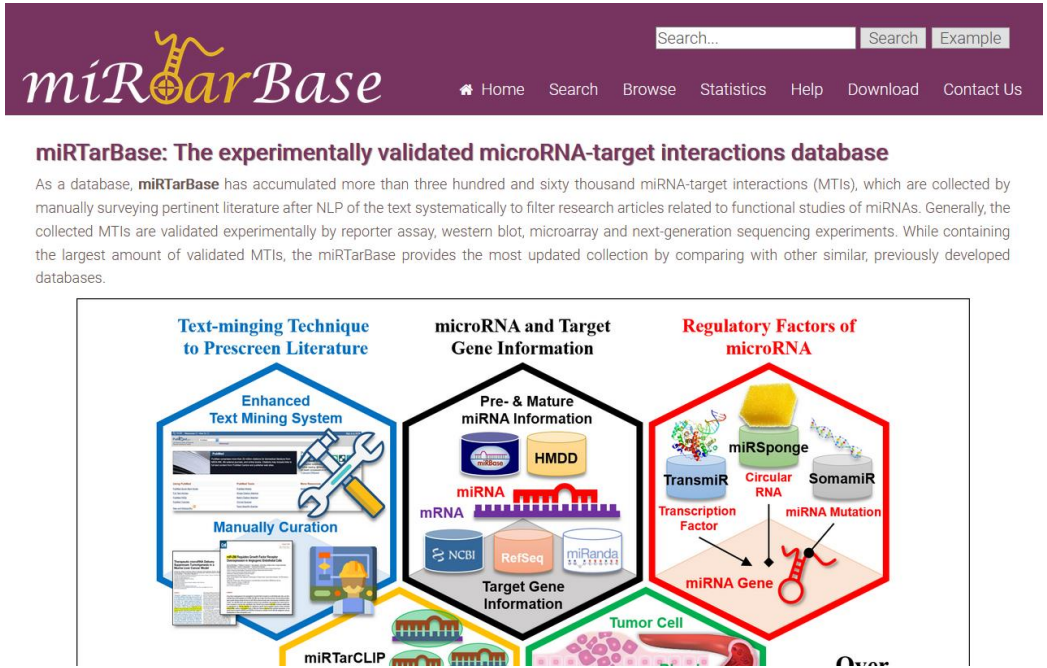
microRNA and target gene identification-mirBase



- Annotation
- Location and sequence of mature miRNA sequence (i.e. miR)
- predicted hairpin portion
- Reference publication

microRNA and target gene identification-miRTarBase

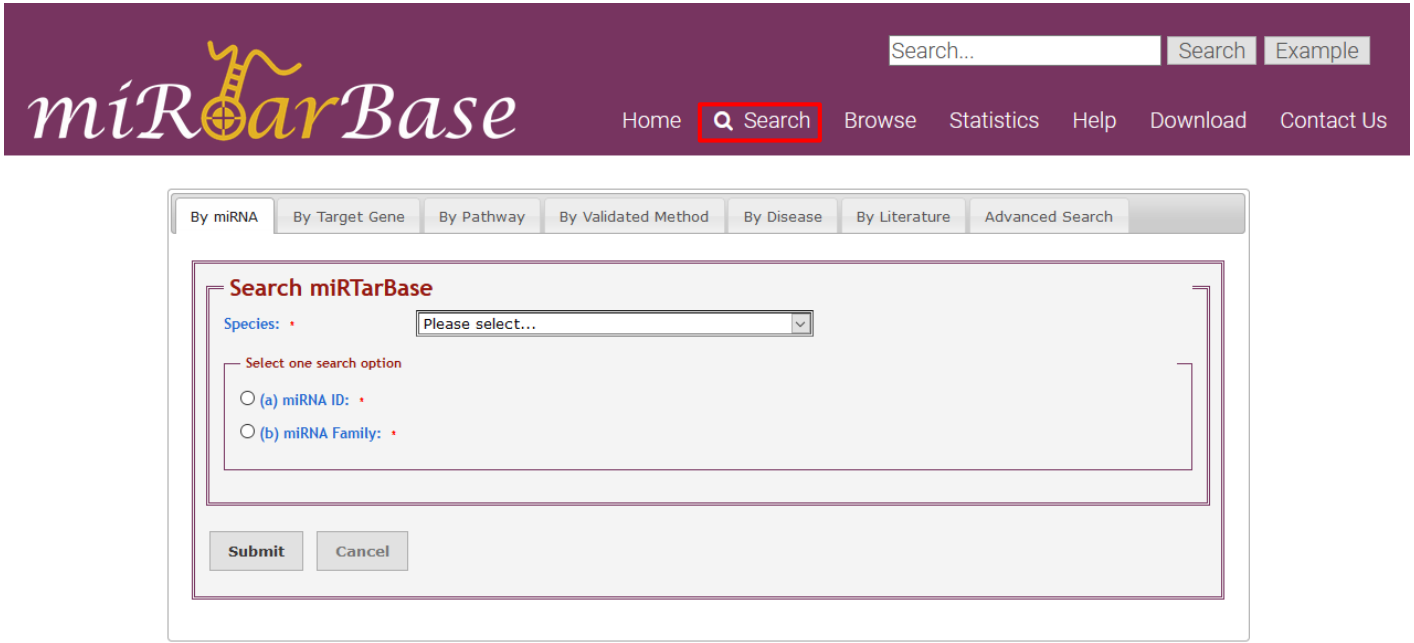
- Experimentally validated microRNA-target interaction
- http://mirtarbase.cuhk.edu.cn/php/index.php



Features	miRTarBase 7.0	miRTarBase 8.0
Release date	2017/09/15	2019/6/30
Known miRNA entry	miRBase v21	miRBase v22
Known Gene entry	Entrez 2017	Entrez 2019
Species	23	28
Curated articles	8,510	10,906
miRNAs	4,076	4,296
Target genes	23,054	23,426
CLIP-seq datasets	231	244
Curated miRNA-target interactions	422,517	430,392
Text-mining technique to prescreen literature	Enhanced NLP	Enhanced NLP+ Scoring system
Download by validated miRNA-target sites	Yes	Yes
Browse by miRNA, gene, and disease	Yes	Yes
Regulation of microRNAs	No	Yes
cell-free miRNAs	No	Yes
MTIs Supported by strong experimental evidences		
Number of MTIs validated by 'Reporter assay'	9,489	11,938
Number of MTIs validated by 'Western blot'	7,258	9,593
Number of MTIs validated by 'qPCR'	8,210	10,768
Number of MTIs validated by 'Reporter assay and Western blot'	6,032	8,297
Number of MTIs validated by 'Reporter assay or Western blot'	10,581	13,132

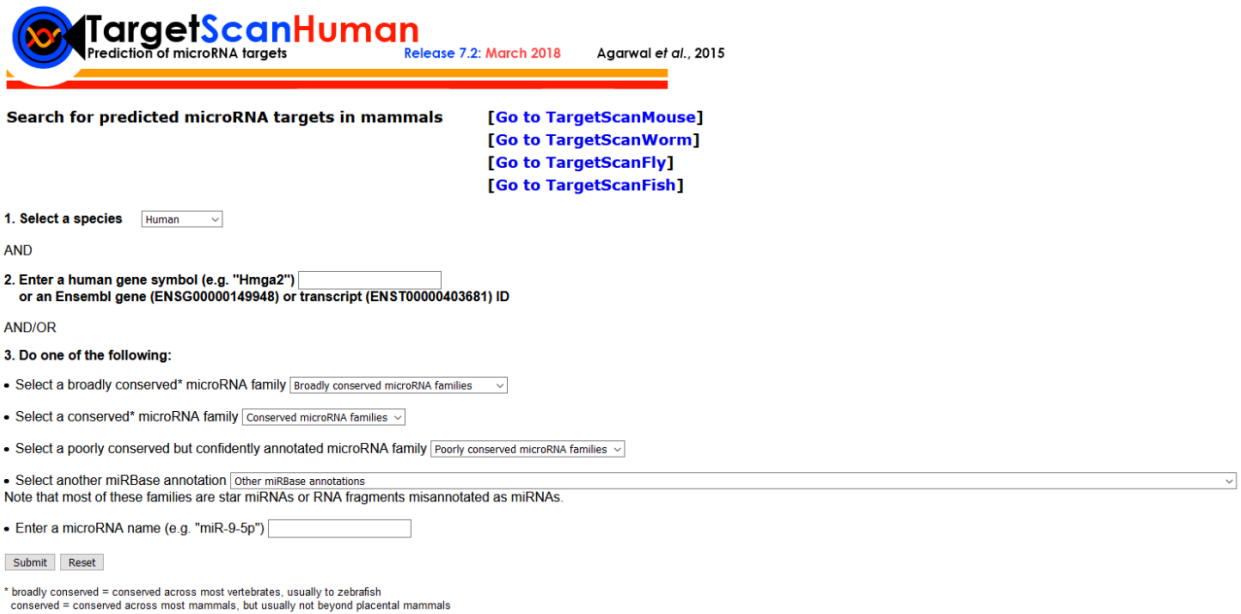
microRNA and target gene identification-miRTarBase

- Searchable by ID(s), target gene, pathway, method, disease, literature



microRNA and target gene identification-TargetScan

- Computationally predicted microRNA-target interaction using sequence complementarity & thermodynamic stability
- http://www.targetscan.org/vert_72/

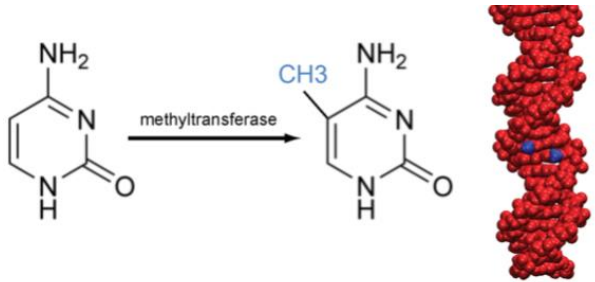


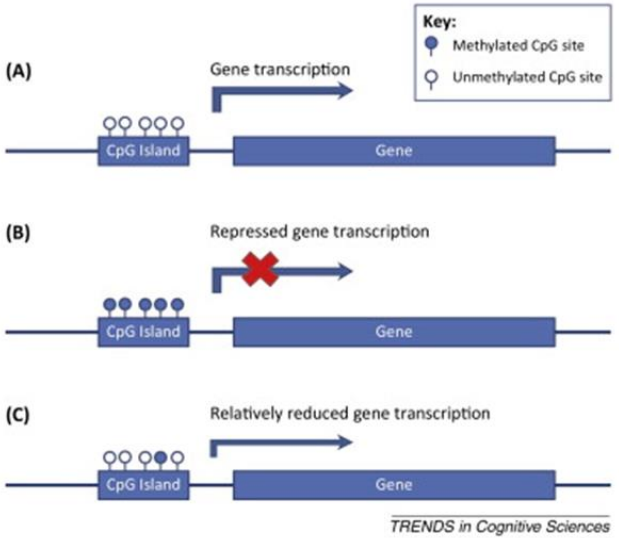
Multi-omic resources (2/5)

DNA methylation analysis and downstream gene regulation

DNA methylation analysis and downstream gene regulation

- Methylation of cytosine is a covalent modification of DNA: the hydrogen of C5 in the cytosine of dinucleotide CpG is replaced with a methyl group.
- When cytosines in promoter regions are methylated, the expression of the downstream gene is repressed.

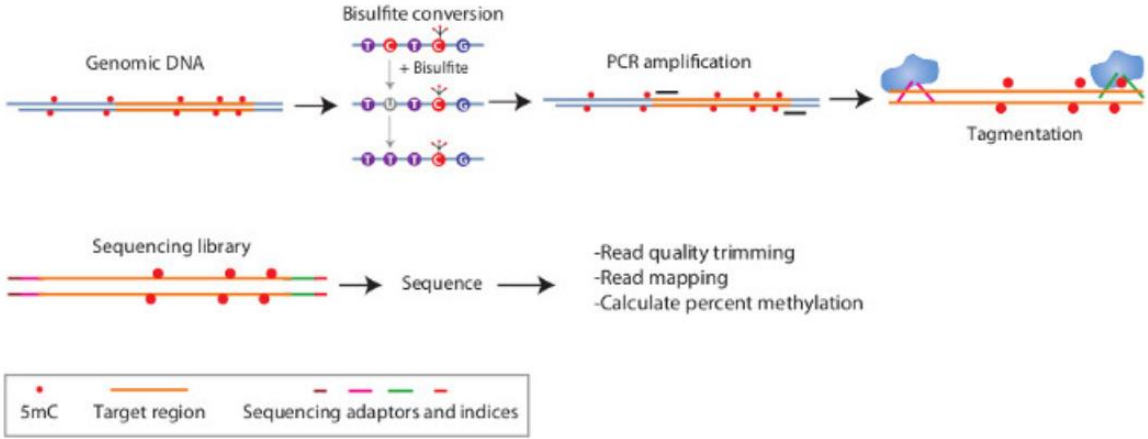




DNA methylation analysis and downstream gene regulation

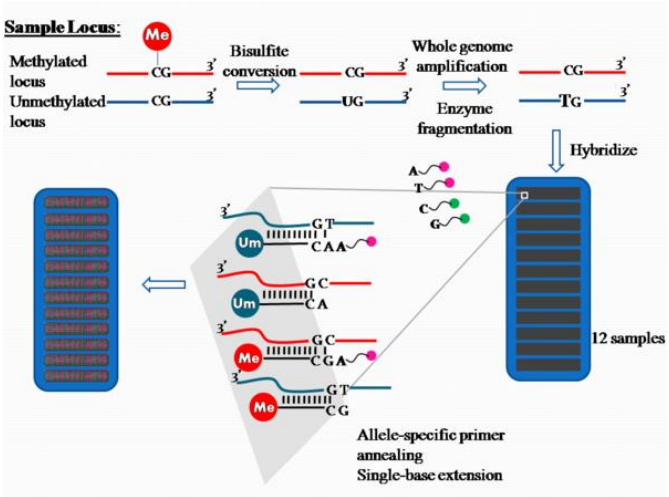
[Experiments for DNA methylation detection]

- Bisulfite sequencing
  - Bisulfite treatment converts unmethylated cytosine (C) to thymine (T).
  - After amplification and sequencing, the relative amount of T compared to C in a CpG site indicates whether the site is methylated or not.



DNA methylation analysis and downstream gene regulation

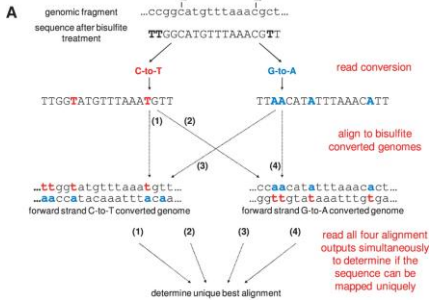
- microarray (Beadchip 27K,450K,850K)
  - Two site-specific probes, one designed for methylated locus and another for unmethylated locus, are used.
  - Sample sequences after bisulfite treatment are hybridized to the probes.
  - Then, if sample sequence basepairs with probe perfectly, single-base extension of probes with labeled ddNTP is conducted so that it yields fluorescent signal and vice versa.



DNA methylation analysis and downstream gene regulation

[Analysis that maps methylation profile to gene]

- mismark for sequencing data
  - Perl-based tool that performs alignments of bisulfite-treated reads to a reference genome (with Bowtie) and cytosine methylation calls.



Read mapping

Methylation calling

- minfi for microarray data
  - R package that performs preprocessing, quality assessment of microarray data and detection of differentially methylated regions.

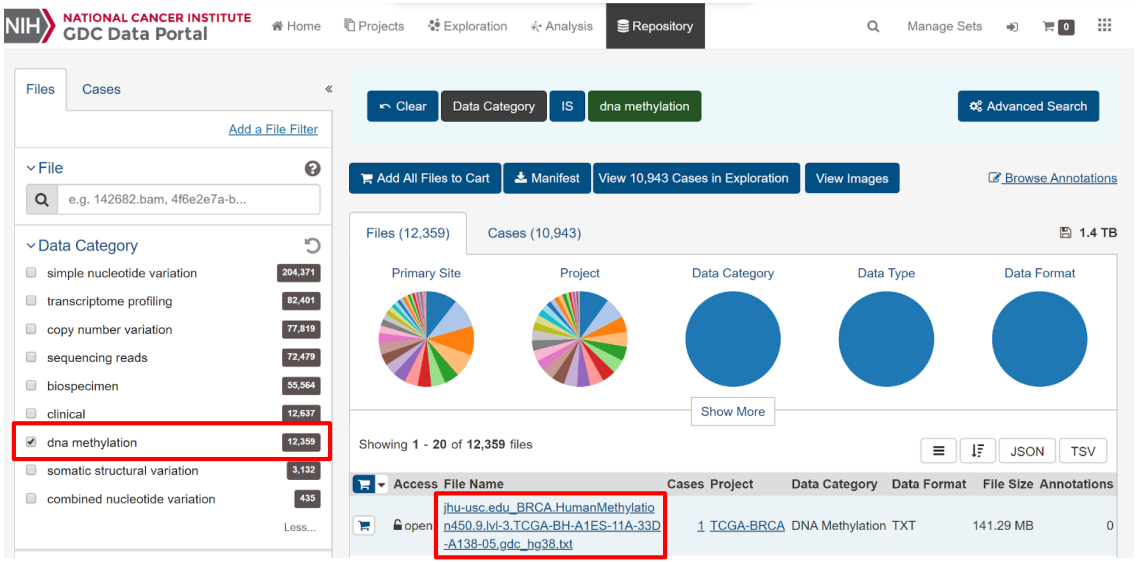
DNA methylation analysis and downstream gene regulation

Database name	Description
TCGA	• Database of genomic, **epigenomic**, transcriptomic, and proteomic data from 20,000 primary cancer and matched normal samples
CCLE	• Database of genomic, epigenomic, transcriptomic, and proteomics data, for over 1100 cancer cell lines • Offers pharmacologic characterization and visualization
MethBank	• Database that integrates high-quality DNA methylomes of healthy people • Offers interactive browser for visualization of methylation data.

iMETHYL	• Integrative multi-omics database
	  ○ that covers whole-DNA methylation, whole-genome, and whole-transcriptome data
	  ○ for CD4+ T-lymphocytes, monocytes, and neutrophils collected from approximately 100 subjects.

DNA methylation analysis and downstream gene regulation-TCGA

- The Cancer Genome Atlas
- Database of mapped DNA methylation in 12,359 of cancer samples from **microarray assay** (Illumina Infinium Human Methylation 27K, 450 K BeadChip)
- For each CpG sites, followings are provided
  - Location (Chr number, start, end locus)
  - the level of methylation (beta-value)
  - Annotated gene
  - Distance to TSS site

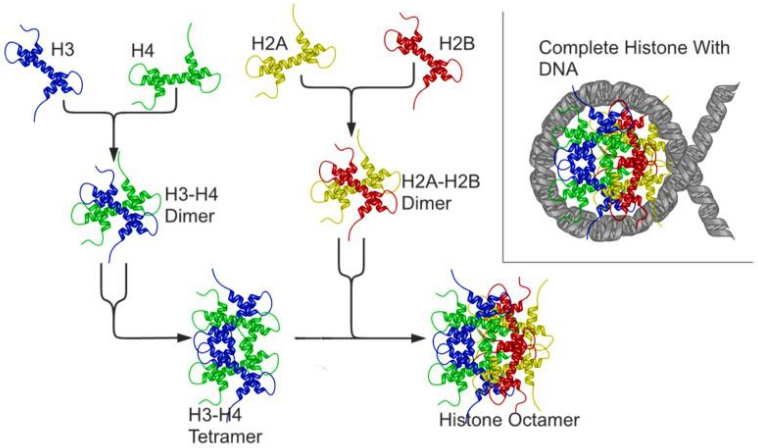


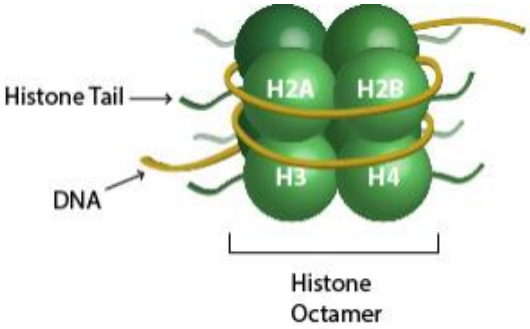
Multi-omic resources (3/5)

Combinatorial histone modification markers

Combinatorial histone modification markers

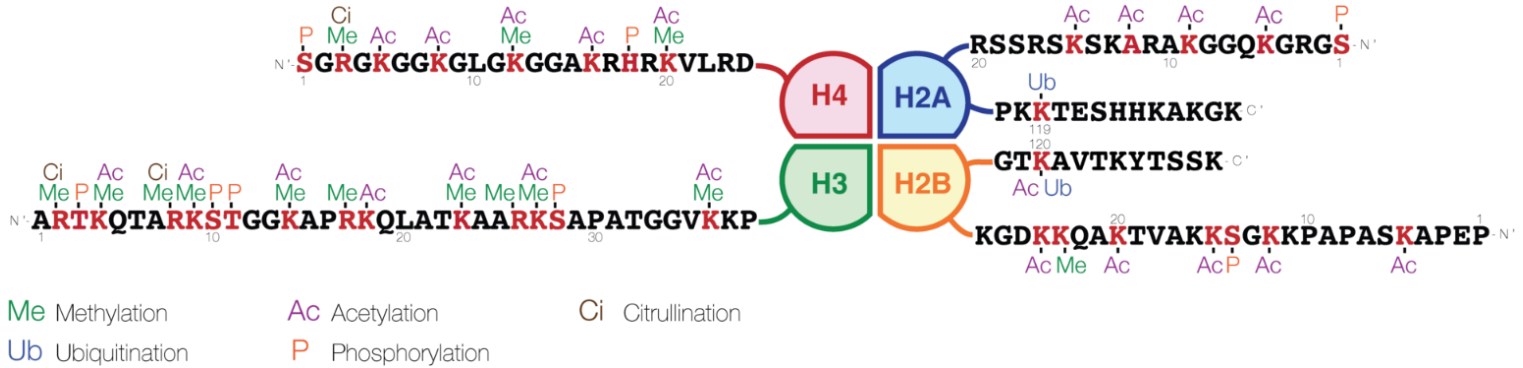
- Histone
  - Protein found in eukaryotic cell nuclei that package and order the DNA into structural units cell nucleosomes.
  - Five major families of histones exist: H1/H5, H2A, H2B, H3 and H4.
  - Histones H2A, H2B, H3 and H4 are known as the core histones.
  - A histone octamer consists of two copies of each of the four core histone proteins.





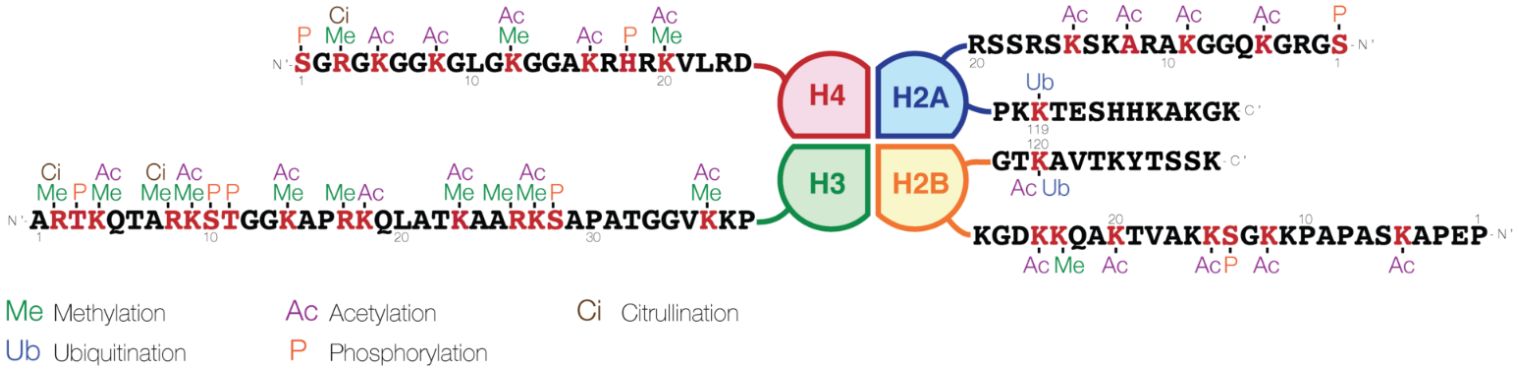
Combinatorial histone modification markers

- Histone modification
  - A covalent post-translational modification (PTM) to histone proteins which includes methylation, acetylation, phosphorylation, citrullination and ubiquitination.
  - The PTMs made to histones can impact gene expression by altering chromatin structure or recruiting histone modifiers and these modifications act in diverse biological processes such as transcriptional activation/inactivation, chromosome packaging, and DNA damage/repair.



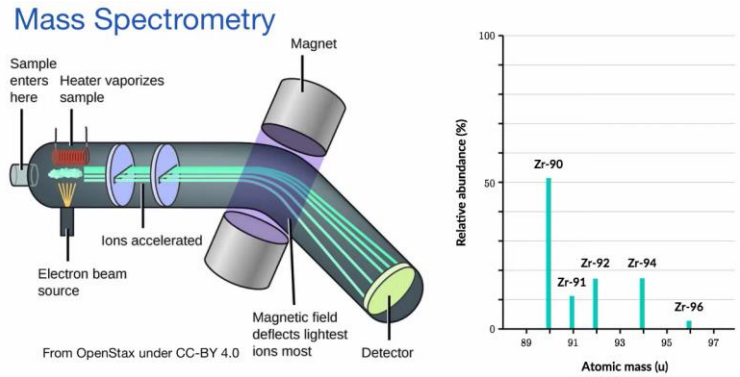
Combinatorial histone modification markers

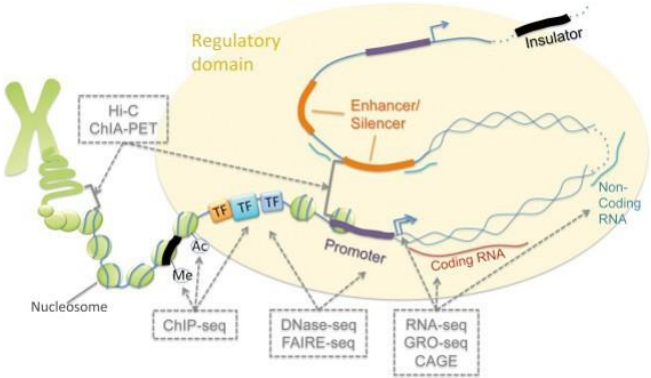
- Histone code
  - A hypothesis that the transcription of genetic information encoded in DNA is in part regulated by chemical modification to histone proteins, primarily on their unstructured ends.
  - For example, H3K27ac indicates the acetylation at the 27th lysine residues of the histone H3 proteins.



Combinatorial histone modification markers

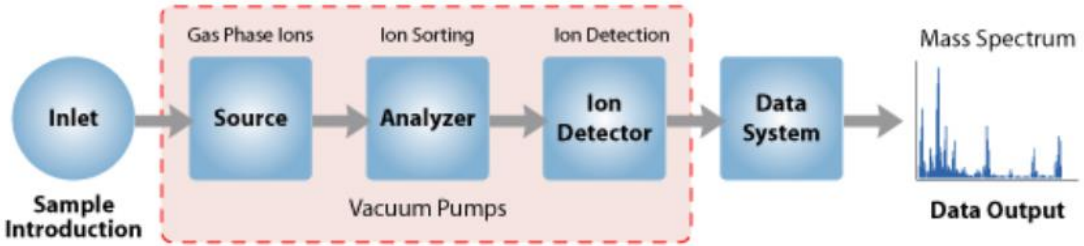
- Mass spectrometry are used to discovery and quantify histone PTMs both within and between samples in an unbiased manner.
- ChIP-chip or ChIP-seq are both powerful tools to identify genome-wide profiles of transcription factors, histone modifications, DNA methylation and nucleosome positioning.





Combinatorial histone modification markers

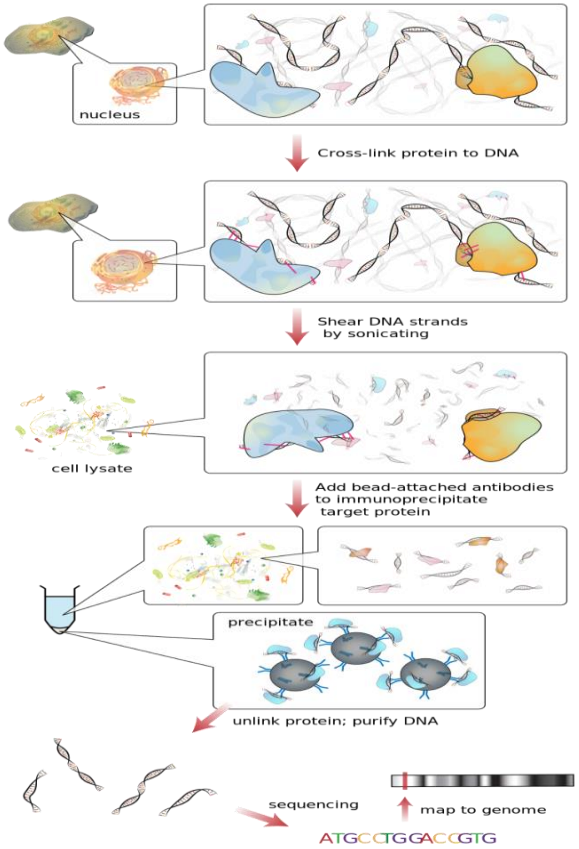
- Mass spectrometry
  - Measures the mass-to-charge ratio of ions.
  1. Produce ions from the samples in the ionization source.
  2. Separate these ions according to their mass-to-charge ratio in the mass analyzer.
  3. Eventually, fragment the selected ions and analyze the fragments in a second analyzer.
  4. Detect the ions emerging from the last analyzer and measure their abundance with the detector that converts the ions into electrical signals.
  5. Process the signals from the detector that are transmitted to the computer and control the instrument using feedback.



Combinatorial histone modification markers

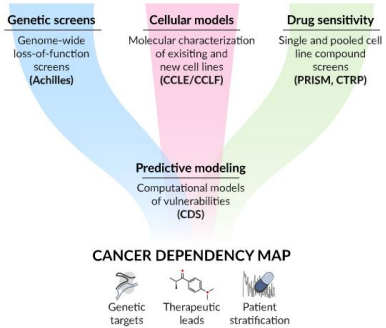
- Chromatin immunoprecipitation (ChIP) - sequencing
  - procedure used to determine whether a given protein binds to or is localized to a specific DNA sequence *in vivo*

1. DNA-binding proteins are crosslinked to DNA with formaldehyde in vivo
2. Isolate the chromatin. Shear DNA along with bound proteins into small fragments
3. Bind antibodies specific to the DNA-binding protein to isolate the complex by precipitation.
   Reverse the cross-linking to release the DNA and digest the proteins
4. amplify specific DNA sequences to see if they were precipitated with the antibody



Combinatorial histone modification markers - DepMap

- A Cancer Dependency Map to systematically identify genetic and pharmacological dependencies and the biomarkers that predict them.
- This portal provides easy access to harmonized data created at the Broad Institute and elsewhere, including:
  - CCLE, Project Achilles, PRISM, DRIVE, Project Score, GDSC, CTD2
- Data content
  - Dependency profiles at genome-scale across more than 500 human cell lines
  - Genetic and pharmacologic characterization of over 1000 cell lines
- https://depmap.org/portal/



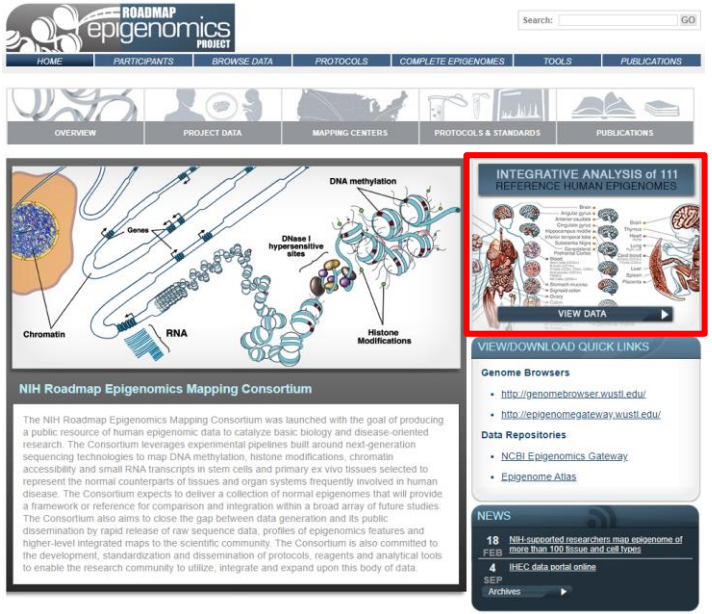
depmap

Cancer Dependency Map

The mutations that cause cancer cells to grow also confer specific vulnerabilities that normal cells lack. Some of these acquired alterations represent compelling therapeutic targets. The challenge is that, for the overwhelming majority of cancers, we do not fully understand the relationship between the genetic alterations of cancer and the dependencies they cause. To solve this problem, we are creating a "cancer dependency map" by systematically identifying genetic dependencies and small molecule sensitivities and discovering the biomarkers that predict them.

Combinatorial histone modification markers - ROADMAP

- The NIH Roadmap Epigenomics Mapping Consortium was launched with the goal of producing a public resource of human epigenomic data to catalyze basic biology and disease-oriented research.
- Experimental pipelines built around next-generation sequencing technologies to map
  - DNA methylation
  - histone modifications
  - chromatin accessibility
  - small RNA transcripts
- http://www.roadmapepigenomics.org/

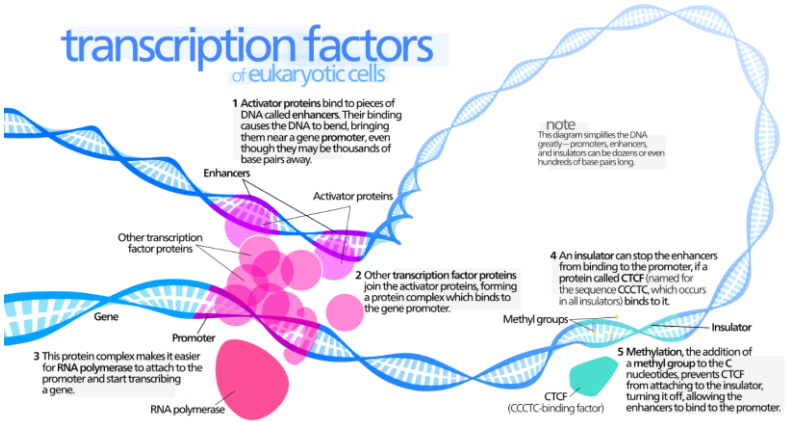


Multi-omic resources (4/5)

Transcription factor (TF) binding site analysis and TF-target gene network

TF binding site analysis

- Transcription factor (TF)
  - Protein that controls the rate of transcription of genetic information from DNA to messenger RNA, by binding to a specific DNA sequence.
  - The region of the gene to which TF binds is called a TF binding site.



TF binding site analysis

Database name	Description
ChIPBase	ChIPBase a database for TF binding sites, motifs and decoding the transcriptional regulation of lncRNAs, miRNAs and protein-coding genes from ~10,200 curated peak datasets derived from ChIP-seq methods in 10 species
ChEA	ChEA is a TF regulation inferred from integrating genome-wide ChIP-X experiments
Factorbook	Factorbook is a Wiki-based database for TF binding data generated by ENCODE consortium
JASPAR	The JASPAR CORE database contains a curated, non-redundant set of profiles, derived from published collection of experimentally defined TF binding sites for eukaryotes
TRANSFAC	TRANSFAC is a manually curated database of eukaryotic TFs, their genomic binding sites and DNA binding profiles
The Human Transcription Factors	Information of how TFs are identified and functionally characterized, principally through the lens of a catalog of over 1,600 likely human TFs and binding motifs

TF binding site analysis - TRANSFAC

- TRANSFAC (TRANScription FACtor)
  - Database of eukaryotic TFs, their genomic binding sites and DNA-binding profiles.
  - In one of the first publicly funded bioinformatics project, launched in 1993, TRANSFAC developed into a resource that became available in 1996.
  - Public database: TRANSFAC 7.0 (2005), Commercial database: TRANSFAC Professional (2019)

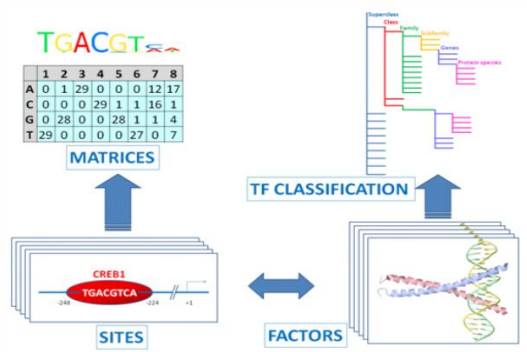
238–241 Nucleic Acids Research, 1996, Vol. 24, No. 1

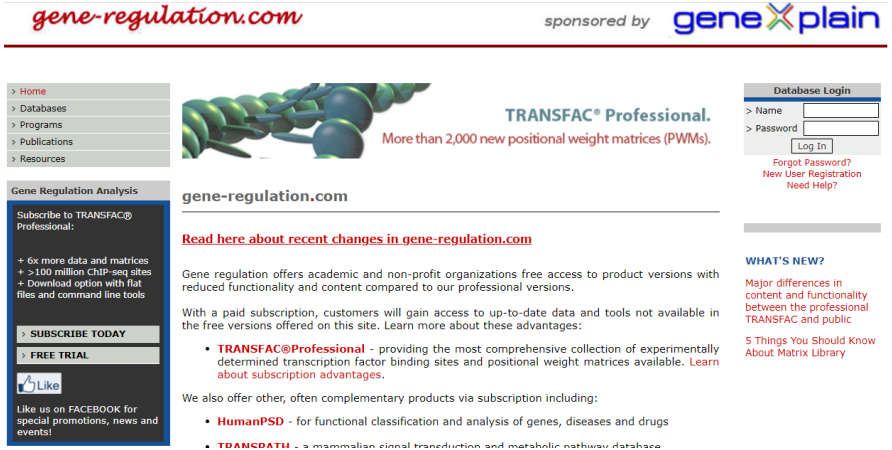
TRANSFAC: a database on transcription factors and their DNA binding sites

E. Wingender*, P. Dietze, H. Karas and R. Knüppel

TF binding site analysis - TRANSFAC

- TRANSFAC library of positional weight matrices is a unique collection of DNA-binding models, suitable for a comprehensive analysis of genomic sequence for potential TF binding sites.
- Two core domains: Match, Patch
  - Match is a weight matrix-based program for predicting TF binding sites in DNA sequences.
  - Patch is pattern-based program for predicting TF binding sites in DNA sequences.
- http://gene-regulation.com/index2.html





TF binding site analysis - JASPAR

- JASPAR
  - Open access database of curated, non-redundant TF binding profiles stored as position frequency matrices (PFMs) and TF flexible models (TFFMs) for TFs across multiple species in six taxonomic groups.
  - It can be converted into position weight matrices (PWMs or PSSMs), used for scanning genomic sequences.
  - Introduced in 2004 and 2020 is latest release version.

Nucleic Acids Research, 2004, Vol. 32, Database issue ***D91–D94***
DOI: 10.1093/nar/gkh012

JASPAR: an open-access database for eukaryotic transcription factor binding profiles

Albin Sandelin, Wynand Alkema, Pär Engström, Wyeth W. Wasserman[1] and Boris Lenhard*

Published online 8 November 2019

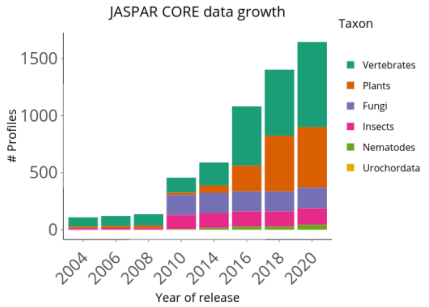
Nucleic Acids Research, 2020, Vol. 48, Database issue ***D87–D92***
doi: 10.1093/nar/gkz1001

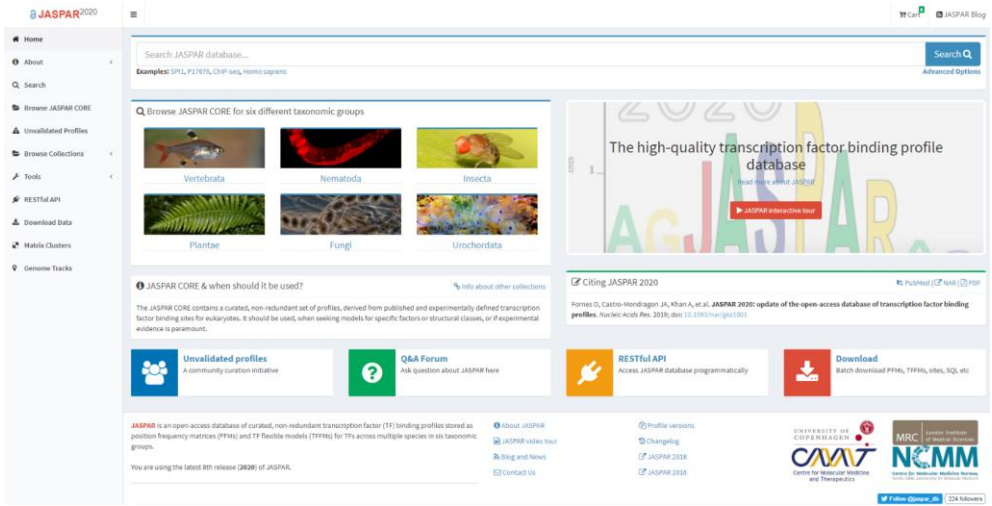
JASPAR 2020: update of the open-access database of transcription factor binding profiles

Oriol Fornes[1,†], Jaime A. Castro-Mondragon[2,†], Aziz Khan[2,†], Robin van der Lee[1], Xi Zhang[1], Phillip A. Richmond[1], Bhavi P. Modi[1], Solenne Correard[1], Marius Gheorghe[2], Damir Baranašić[3,4], Walter Santana-Garcia[5], Ge Tan[6], Jeanne Chèneby[7], Benoit Ballester[7], François Parcy[8], Albin Sandelin[9,*], Boris Lenhard[3,4,10,*], Wyeth W. Wasserman[1,*] and Anthony Mathelier[2,11,*]

TF binding site analysis - JASPAR

- JASPAR CORE and it’s collections:
  - CNE, FAM, PBM, PBM HLH, PBM HOMEO, PHYLOFACTS, POLII, SPLICE
- JASPAR CORE data content
  - Vertebrates: 746
  - Plants: 530
  - Fungi: 183
  - Insects: 143
  - Nematodes: 43
- http://jaspar.genereg.net/





TF binding site analysis - ChEA

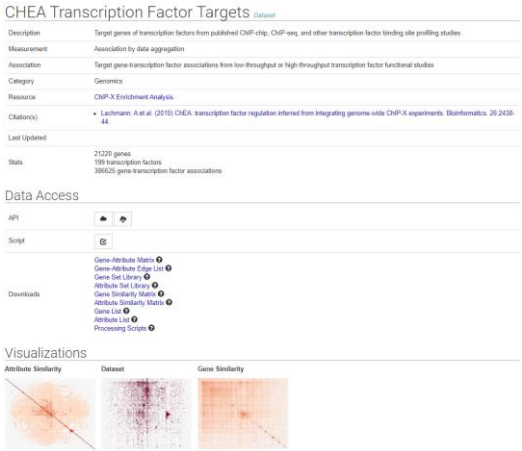
- ChEA (ChIP-X (ChIP-seq, ChIP-chip) Enrichment Analysis)
  - TF enrichments analysis tool that ranks TFs associated with user submitted gene sets.
- One of the reasons high-throughput genome-wide ChIP-X studies are expected to be more useful and accurate than computational sequence-based methods is because the sequence-based approaches do not take into consideration the chromatin stata of the cell under a specific experimental condition, cell type or organism.

BIOINFORMATICS ORIGINAL PAPER
Vol. 26 no. 19 2010, pages 2438–2444
doi:10.1093/bioinformatics/btq466

Systems biology
Advance Access publication August 13, 2010

ChEA: transcription factor regulation inferred from integrating genome-wide ChIP-X experiments

Alexander Lachmann[1], Huilei Xu[1], Jayanth Krishnan[2], Seth I. Berger[1], Amin R. Mazloom[1] and Avi Ma'ayan[1,*]



TF binding site analysis - ChEA

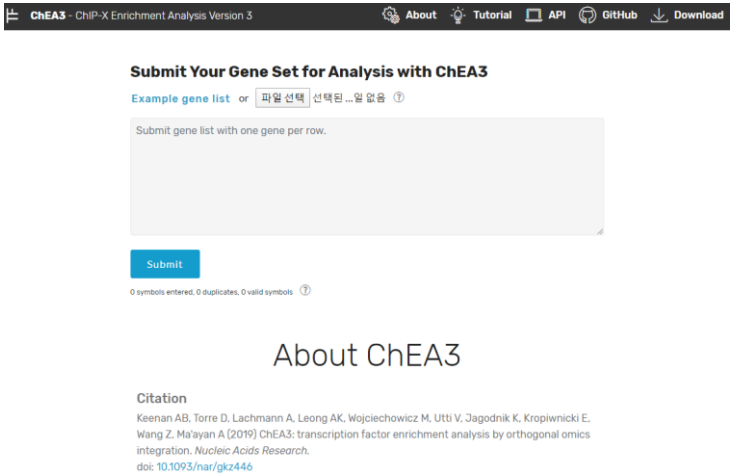
- ChEA3
  - Follow-up versions of ChEA, and ChEA3 contains six primary reference gene set libraries created from multiple resources: GTEx, ARCHS4, ENCODE, Literature ChIP-seq, ReMap, Enrichr.
  - Total 1632 factors in ChEA3.
  - Global transcription factor co-expression network, local results-specific co-regulatory network, bar charts and a clustergram are available.
  - https://amp.pharm.mssm.edu/chea3/

W212–W224 Nucleic Acids Research, 2019, Vol. 47, Web Server issue *Published online 22 May 2019*
doi: 10.1093/nar/gkz446

ChEA3: transcription factor enrichment analysis by orthogonal omics integration

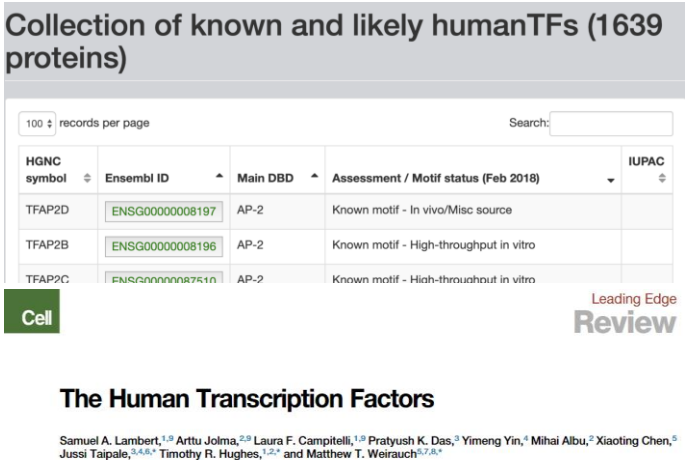
Alexandra B. Keenan, Denis Torre, Alexander Lachmann, Ariel K. Leong, Megan L. Wojciechowicz, Vivian Utti, Kathleen M. Jagodnik, Eryk Kropiwnicki, Zichen Wang and Avi Ma'ayan*

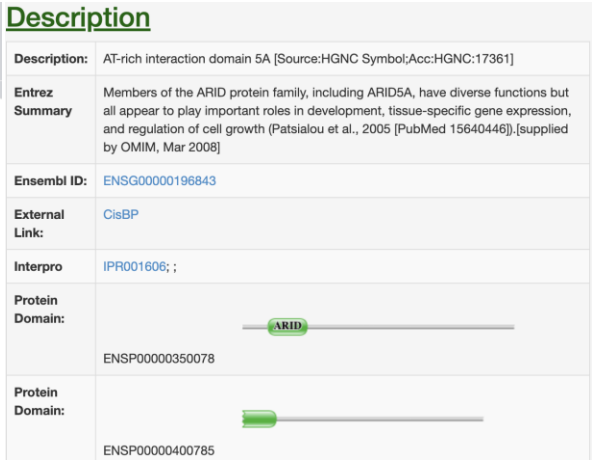
Library	Unique TFs	Unique TF Interactions	Gene Sets
ARCHS4 Coexpression	1628	480 504	1628 human
ENCODE ChIP-seq	118	392 667	552 (470 human, 82 mouse)
Enrichr Queries	1404	409 279	1404 (unknown species)
GTEx Coexpression	1607	468 672	1607 human
Literature ChIP-seq	164	340 547	307 (138 human, 164 mouse, 5 rat)
ReMap ChIP-seq	297	417 025	297 human

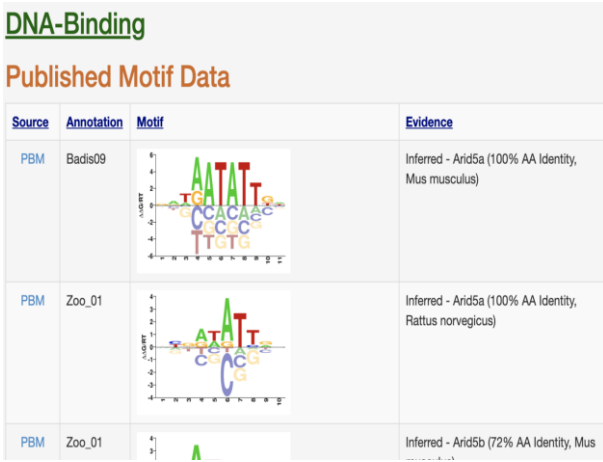


TF binding site analysis - The Human Transcription Factors

- Information of how TFs are identified and functionally characterized, principally through the lens of a catalog of human TFs and binding motifs
- Contains the catalog of 1639 known and likely human TFs and their motifs
  - Candidate proteins were manually examined by a panel of experts based on available data
  - Proteins with experimentally demonstrated DNA binding specificity were considered TFs
  - Other proteins, such as co-factors and RNA binding proteins, were classified as non-TFs
  - All proteins (both TFs and non-TFs) are contained in the database, along with the associated evidence
  - Database: http://humantfs.ccbr.utoronto.ca/index.php / Paper("Cell" journal): https://doi.org/10.1016/j.cell.2018.01.029





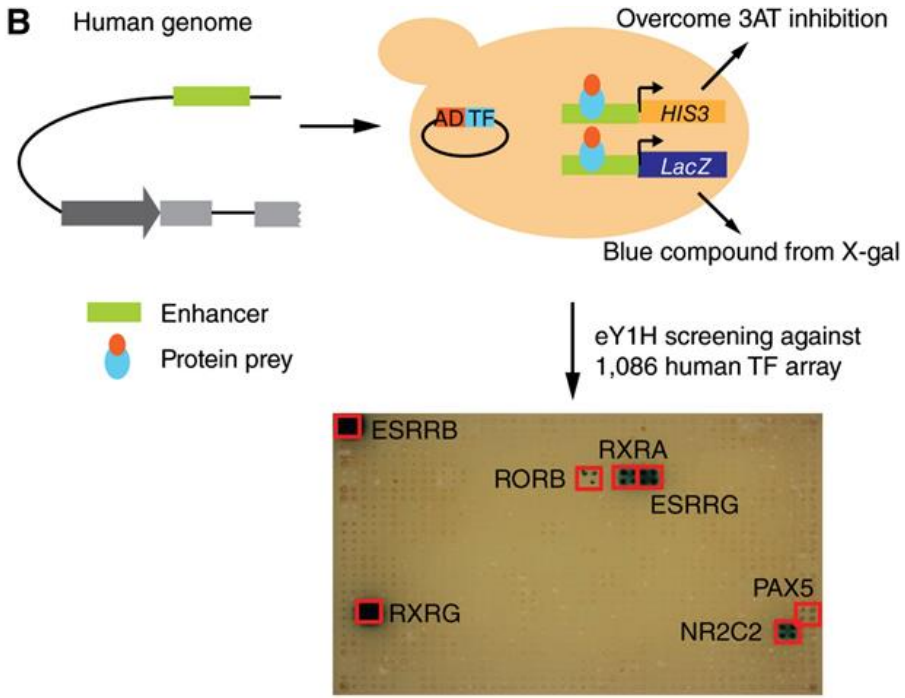


Multi-omic resources (4/5)

Transcription factor (TF) binding site analysis and TF-target gene network

Transcription Factor-Target Gene (TF-TG) network

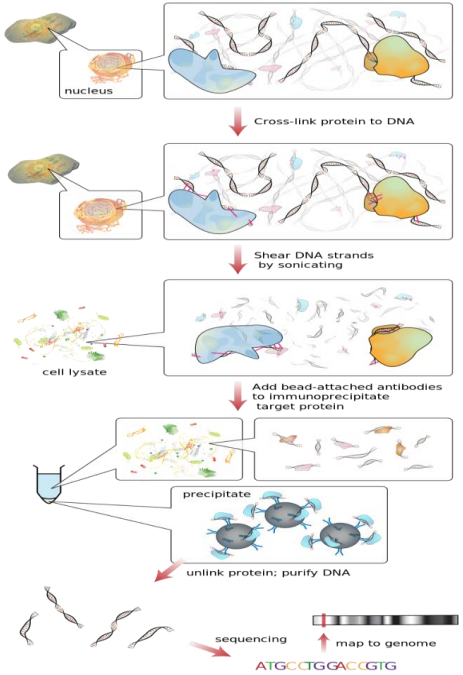
- Yeast One-Hybrid Assays(Y1H)
  - *in vitro* method to analyze the intracellular interaction between DNA and proteins
- The main difference between Y2H and Y1H
  - Y2H assay measures the interactions between proteins and proteins
  - Y1H assay measures the interactions between DNA and proteins
- Useful for:
  - Low recognition of target DNA sequence by yeast endogenous transcription factors
  - Verifying known interactions between DNA and proteins
  - Finding new transcription factors



TF-TG network

- ChIP-seq can be also be used to obtain TF-TG relations

  (Details described in an earlier slide)

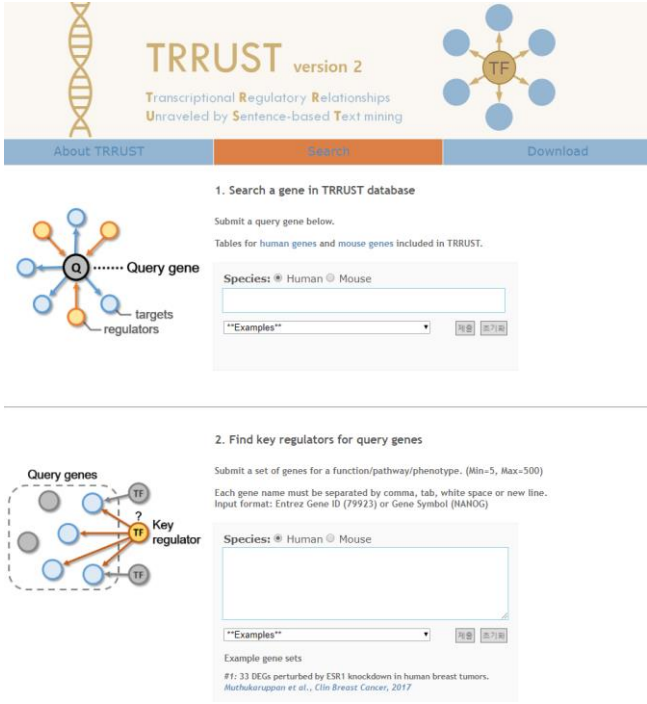


TF-TG network databases

Experimentally Validated TF-TG network database	**Description**
TRRUST	Transcriptional Regulatory Relationships Unraveled by Sentence-based Text mining Manually curated database of human and mouse transcriptional regulatory networks
Motif based TF-TG network databases	**Description**
TRANSFAC	TRANSFAC is a manually curated database of eukaryotic TFs, their genomic binding sites and DNA binding profiles
JASPAR	The JASPAR CORE database contains a curated, non-redundant set of profiles, derived from published collection of experimentally defined TF binding sites for eukaryoutes
Computationally Predicted TF-TG network database	**Description**
Regulatory Circuits	Comprehensive resource of close to 400 cell type- and tissue-specific gene regulatory networks for human

TF-TG network - TRRUST

- Manually curated database of human and mouse transcriptional regulatory networks.
- Data Composition:
  - 8,444 TF-target regulatory relationships of 800 human TFs
  - 6,552 TF-target regulatory relationships of 828 mouse TFs
- Derived from 11,237 pubmed articles which describe small-scale experimental studies of transcriptional regulations
- Used sentence-based text mining approach to search for regulatory relationships from over 20 million pubmed articles
- Provides information of mode of regulation (activation or repression)
  - Currently 8,972 (59.8%) regulatory relationships are known for mode of regulation
- https://www.grnpedia.org/trrust/



TRRUST Download page

Species	Date	# of TF genes	# of non-TF genes	# of regulatory links	TSV format	BioC format
Human	16/04/2018	795	2,067	8,427	Download	Download
Mouse	16/04/2018	827	1,629	6,490	Download	Download

TF-TG network - Regulatory Circuits

- Research website on perturbed molecular circuits that underlie complex diseases
- The web site contains a data-base of TF regulatory networks
- Data Composition:
  - 394 cell type and tissue-specific regulatory networks for human(derived from FANTOM5 data)
  - 32 high-level regulatory networks grouping similar cell types and tissues
  - 41 public networks including protein-protein interaction, tissue-specific co-expression, and ChIP-seq based networks
- They claim to be the largest collection of tissue-specific transcriptional regulatory networks for human
- Their paper: “Tissue-specific gene networks disrupted in complex diseases”
- http://regulatorycircuits.org/

Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases

Daniel Marbach[1,2], David Lamparter[1,2], Gerald Quon[3,4], Manolis Kellis[3,4], Zoltán Kutalik[2,5] & Sven Bergmann[1,2]

Mapping perturbed molecular circuits that underlie complex diseases remains a great challenge. We developed a comprehensive resource of 394 cell type– and tissue-specific gene regulatory networks for human, each specifying the genome-wide connectivity among transcription factors, enhancers, promoters and genes. Integration with 37 genome-wide association studies (GWASs) showed that disease-associated genetic variants—including variants that do not reach genome-wide significance—often perturb regulatory modules that are highly specific to disease-relevant cell types or tissues. Our resource opens the door to systematic analysis of regulatory programs across hundreds of human cell types and tissues (http://regulatorycircuits.org).

Here we introduce a unique resource of 394 cell type– and tissue-specific gene regulatory networks for human. We infer networks by integrating transcription factor (TF) sequence motifs with promoter and enhancer activity data from the FANTOM5 project[23,24] (**Fig. 1**). All networks and tools are freely available at http://regulatorycircuits.org.

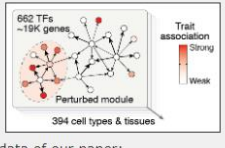
RESULTS

Cell type– and tissue-specific gene regulatory circuits

Our pipeline for reconstructing transcriptional regulatory circuits involves (1) genome-wide mapping of promoters and enhancers, (2) linking TFs to promoters and enhancers and (3) linking enhancers and promoters to target genes (**Fig. 1a,b** and

Overview

Mapping perturbed molecular circuits that underlie complex diseases remains a great challenge. We developed a comprehensive resource of close to 400 cell type- and tissue-specific gene regulatory networks for human. Our study shows that disease-associated genetic variants often perturb regulatory modules in cell types or tissues that are highly specific to that disease.



On this website we provide supplementary information, software tools and data of our paper:

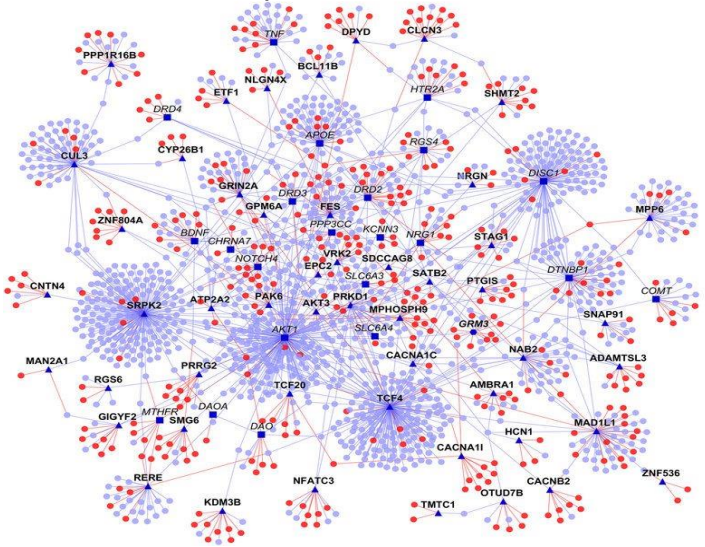
- **Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases**. (PDF, SI)
  Marbach D, Lamparter D, Quon G, Kellis M, Kutalik Z, and Bergmann S.
  Nature Methods, 13, 366-370, 2016. (PubMed, SIB news, *Nature Reviews Genetics* highlight)

Multi-omic resources (4/5)

Protein-protein interaction (PPI) network

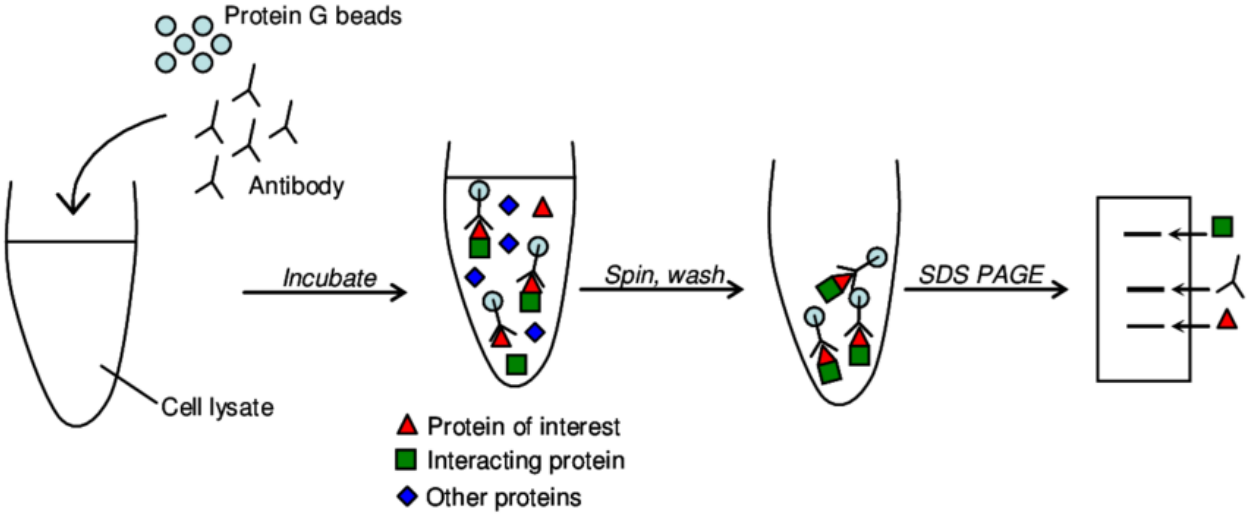
PPI network

- Protein-Protein Interactions(PPIs) are the physical contacts of high specificity established between two or more protein molecules as a result of biochemical events steered by electrostatic forces
- PPIs are defined under various types of interactions
  - Therefore, interactions of PPI network are mostly not condition specific
- PPI network: mathematical representations of the physical contacts between proteins in cell
- Proteins are connected in PPI network by:
  - physical interactions
  - metabolic
  - signaling pathways of the cell
- Investigation of protein-protein interaction:
  - Co-immunoprecipitation (Co-IP)
  - Yeast two-hybrid system (Y2H)
  - etc...



PPI network

- Co-immunoprecipitation (Co-IP)
  - a popular technique to identify physiologically relevant protein–protein interactions by using target protein-specific antibodies to indirectly capture proteins that are bound to a specific target protein
  - These protein complexes can then be analyzed to identify new binding partners, binding affinities, the kinetics of binding and the function of the target protein

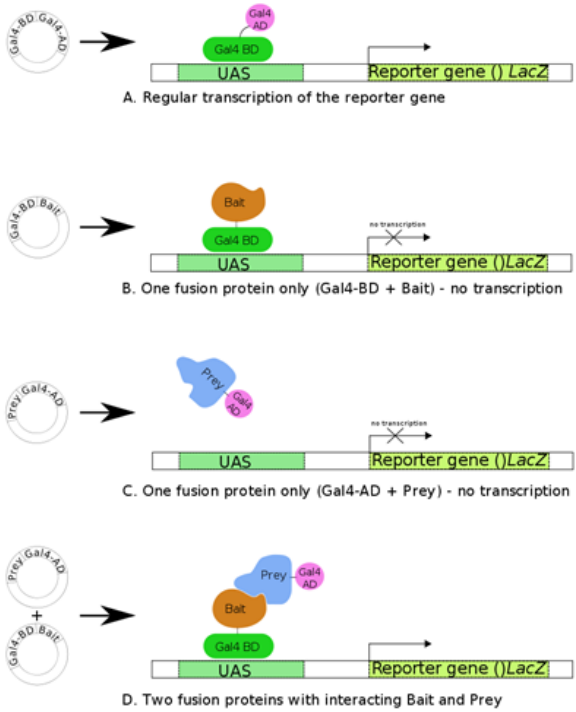


PPI network

UAS: upstream activating sequence
BD: binding domain
AD: activating domain

- Yeast two-hybrid system (Y2H)
  - powerful molecular genetic tool which investigates protein-protein interactions *in vivo*
  - This system can be used to study the interaction between
    two proteins which are expected to interact
    or find proteins (prey) that interact with a protein you already have (bait)

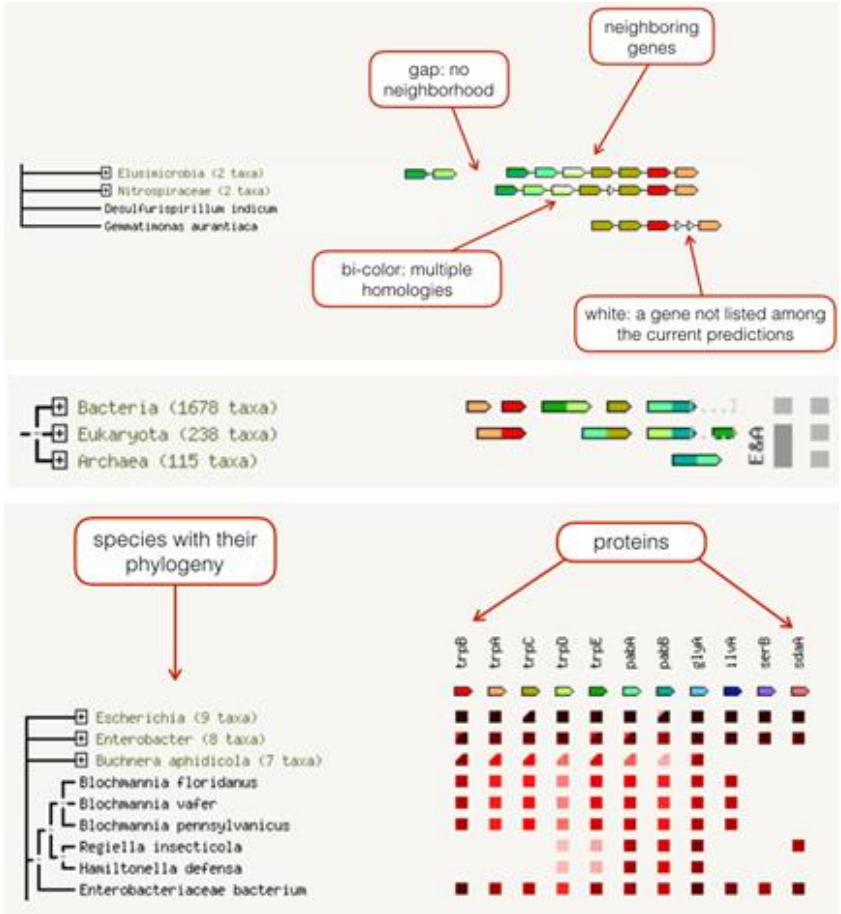
1. the 'bait' and 'hunter' plasmids are introduced
   into yeast cells by transfection
2. the plasma membrane is disrupted to yield holes,
   through which the plasmids can enter
3. cells containing both plasmids are selected for
   by growing cells on minimal media
4. Only cells containing both plasmids have both genes encoding
   for missing nutrients, and consequently, are the only cells that will survive



PPI network

- Computational prediction using data generated by wet lab experiments.
  - STRING

    Score = f(Neighborhood, Gene Fusion, Cooccurence, Coexpression, Experiments, Databases, Textmining)



PPI network

Experimentally Validated PPI network databases	Description
BioGrid	Interaction repository from publications, protein and genetic interactions, chemical associations and post translational modifications from major model organism species
IntAct	Open source database system and analysis tools for molecular interaction data All interactions are derived from literature curation or direct user submissions
HINT	(High-quality INTeractomes) compilation of protein-protein interactions from 8 other interactome resources
Computationally Predicted PPI network databases	**Description**
STRING	Database of known and predicted protein-protein interactions including direct(physical) and indirect(functional) associations. The interactions stem from computational prediction, knowledge transfer between organisms, and from interactions aggregated from other databases

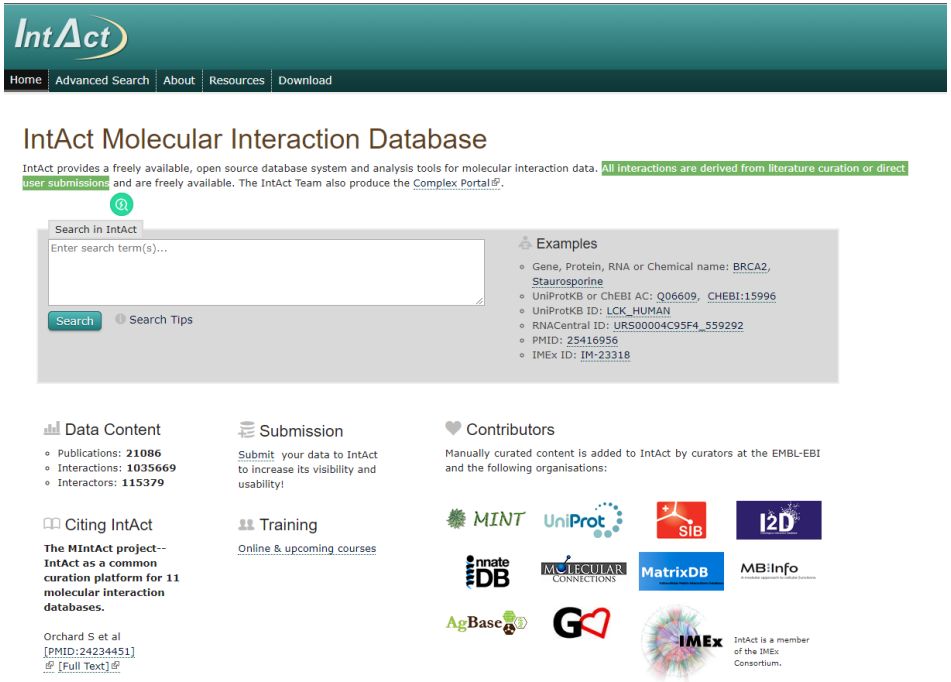
PPI network - BioGrid

- The Biological General Repository for Interaction Datasets (BioGRID) is a public database that archives and disseminates genetic and protein interaction data from model organisms and humans
- Data Content:
  - 71,342 publications
  - 1,763,194 protein and genetic interactions
  - 28,093 chemical associations
  - 874,796 post translational modifications from major model organism species

- BioGRID provides interaction data to several model organism databases, resources such as Entrez-Gene, SGD, TAIR, FlyBase and other interaction meta-databases
- https://thebiogrid.org/

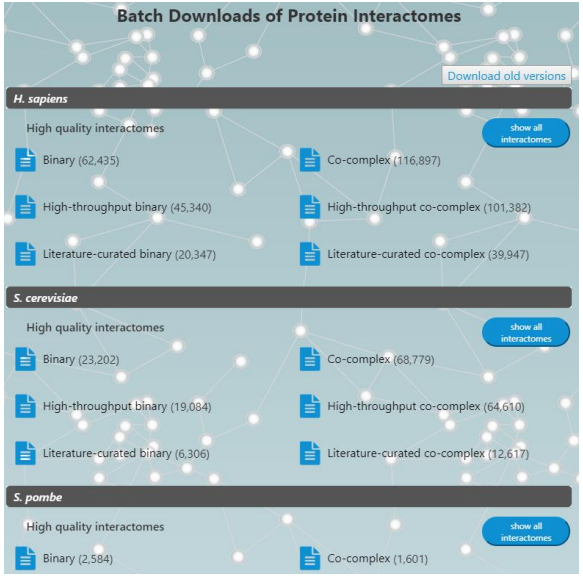
PPI network - IntAct

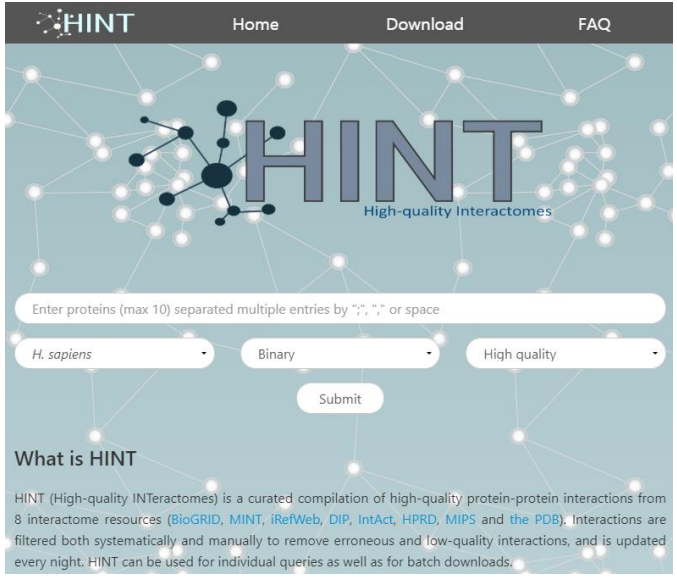
- Database of evidence for molecular interactions, and maintains the Complex Portal reference resource for macromolecular complexes
- Interactions are derived from literature curation or direct user submissions
- Data Content:
  - 21,086 publications
  - 1,035,669 interactions
  - 115,379 interactors
- https://www.ebi.ac.uk/intact/



PPI network - HINT

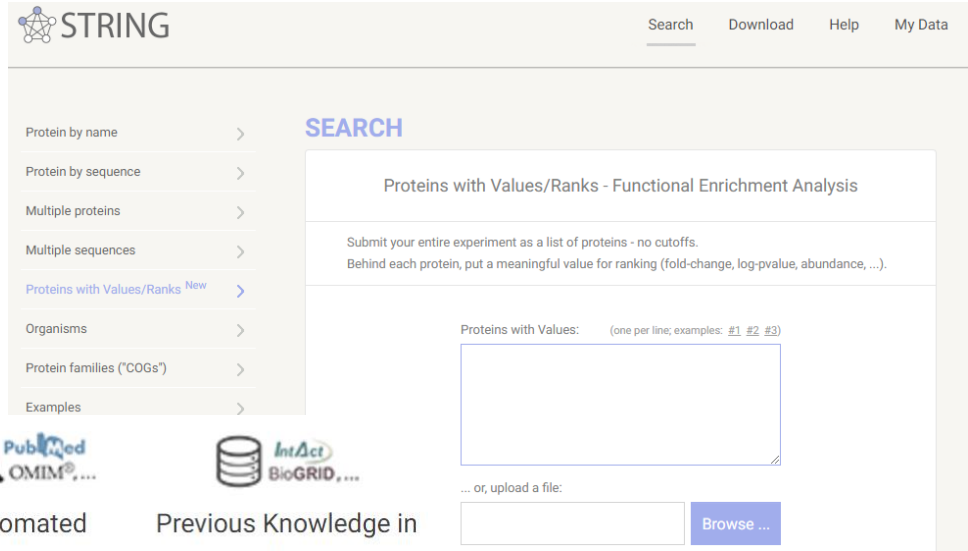
- HINT (High-quality INTeractomes) is a curated compilation of protein-protein interactions
- Utilizes information from 8 interactome resources
  - BioGRID, MINT, iRefWeb, DIP, IntAct, HPRD, MIPS, the PDB
- Interactions are filtered both systematically and manually to remove erroneous and low-quality interactions
- HINT can be used for individual queries as well as for batch downloads
- http://hint.yulab.org/

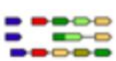




PPI network - STRING

- STRING is a database of known and predicted protein-protein interactions
- The interactions include direct (physical) and indirect (functional) associations from:
  - computational prediction
  - knowledge transfer between organisms
  - interactions aggregated from other (primary) databases
- Data Composition (total 5,090 organisms):
  - 4,445 Bacteria
  - 477 Eukaryotes
  - 168 Archaea
  - 24,584,628 proteins
- Data Sources:





Genomic Context Predictions

High-throughput Lab Experiments

(Conserved) Co-Expression

Automated Textmining

Previous Knowledge in Databases

- https://string-db.org/cgi/about.pl?sessionId=KMMdVsddi5IU&footer_active_subpage=statistics

Machine Learning Basics

BIML 2020

Sun Kim research lab

Goal

- In this tutorial, we are supposed to learn how multi-omics data are integrated using network and AI technologies.

- We will survey quite a number of papers in the afternoon.

- To understand lectures in the afternoon, you need to understand basics in machine learning.

Multi-omics integration is done as

- Classification problem, or
- Clustering problem

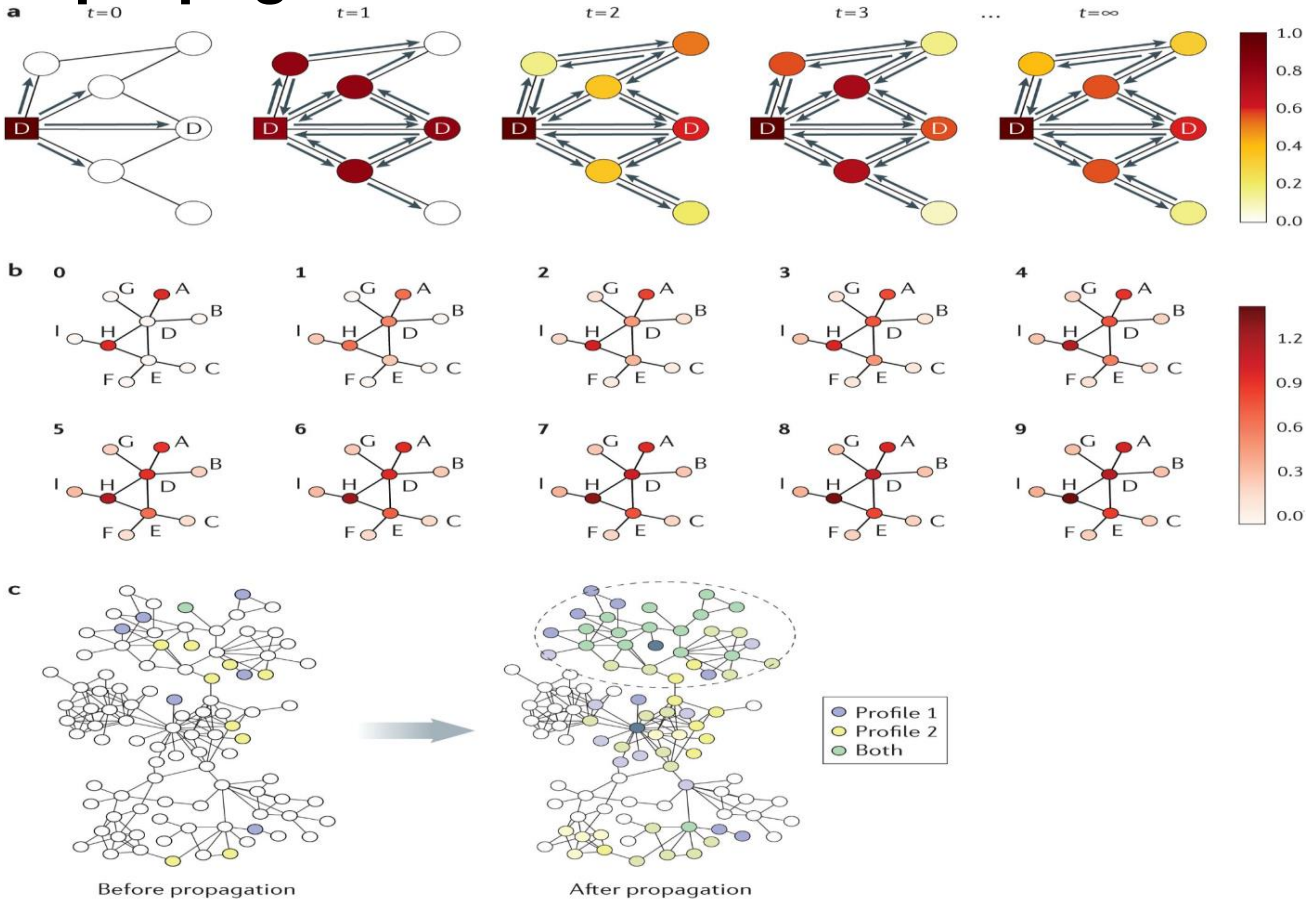
- Generally, the classification scheme performs better than clustering.

- However, clustering has potential to learn new knowledge, e.g., new cancer subtypes.

ML and network analysis basic concepts

- Network propagation
- Tensor decomposition
- Bayesian inference
- Autoencoder, deep learning
- Metric or representation learning

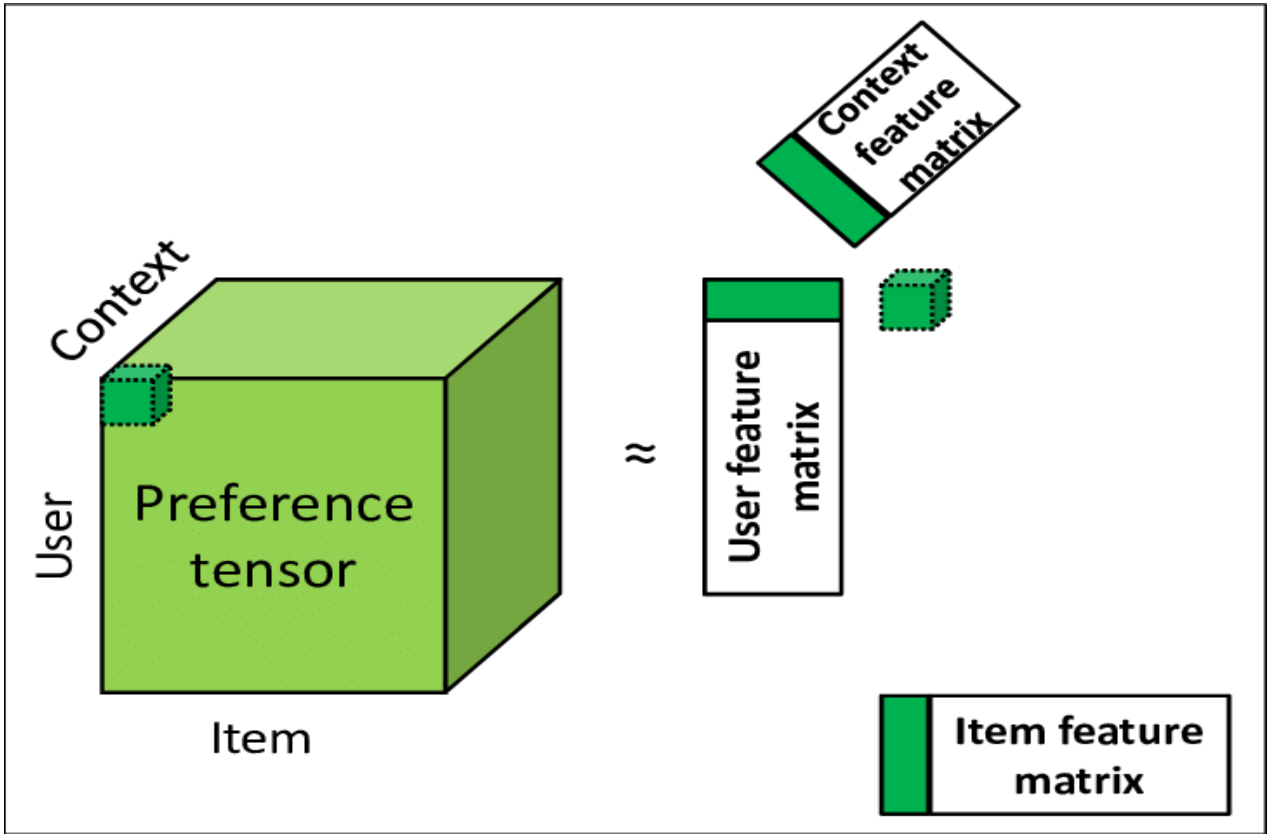
Network propagation



https://www.nature.com/articles/nrg.2017.38/figures/1

Nature Reviews | Genetics

Tensor Decomposition



https://www.researchgate.net/figure/Concept-of-the-tensor-decomposition-in-3-dimension-with-the-classical-user-item-context_fig1_280132481

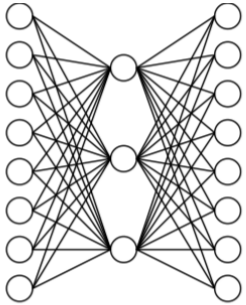
Bayesian Inference

Given Two hypotheses, H1 and H2,
and event or data, E,
our goal is to determine which hypothesis is better.

How?
By computing posterior probabilities, P(H1 | E) vs. P(H2 | E).

$$P(H_1 \mid E) = \frac{P(E \mid H_1)\, P(H_1)}{P(E \mid H_1)\, P(H_1) + P(E \mid H_2)\, P(H_2)}$$

Autoencoder



Learned hidden layer representation:

Input		Hidden Values				Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.01	.11	.88	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.22	.99	.99	→	00000100
00000010	→	.80	.01	.98	→	00000010

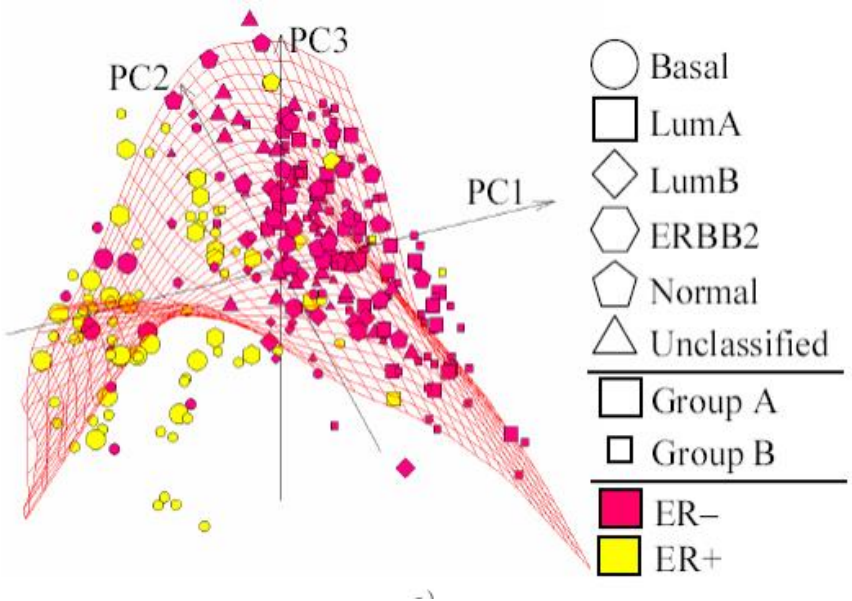
From Tom Mitchell's slides for Machine Learning textbook. 1997 ?

Metric learning?
Representation learning?
Hidden or latent variables?

- Methods, such as tensor decomposition, variational autoencoder, are good examples for these?

- Not clear?

PCA as a metric learning, dimension reduction method.



https://ko.wikipedia.org/wiki/주성분_분석#/media/파일:Elmap_breastcancer_wiki.png

PCA is successful when you are lucky

- In the previous example, the dimension was reduced to 3 from 20,000 (the number of genes).
- PC1, PC2, PC3, etc are new dimensions as combinations of the original dimensions or variables. PCs are created to maximize variance of data.
- It works if you are lucky.

- Alternative ways to created dimensions are numerous. Tensor or autoencoders are widely used techniques.

Well… Tell me more core concepts

- Likelihood
- Prior distribution
- Bayes rules
- Posterior distribution
- Hidden or latent variables
- Dimension reduction

EM: Expectation Maximization

I think that understanding EM is the most effective way to understand the core concepts in ML!

EM?

Many people say “I know EM is used in many applications but I do not know what it is.”

I hope I can help you understand EM today

Basics

- Model
- Likelihood
- Prior probability
- Posterior probability
- Latent variables

Coin Model

- Tossing a coin produces either
  - head (앞면) or tail (뒷면)
- A model for a coin ?
  - Prob[head], Prob[tail]
- A fair or loaded coin
  - Fair coin: Prob[head] = 0.5, Prob[tail] = 0.5
  - Loaded coin at Casino: Prob[head] = 0.6, Prob[tail] = 0.4

Likelihood

- We have a sequence of tossing a coin:

  HTHHHTT

- Is this generated using a fair coin or a loaded coin?
  - For a fair coin,

    Pr[HTHHHTT | Fair] = 0.5 * 0.5 * 0.5 * 0.5 * 0.5 * 0.5 * 0.5 = .0078125
  - For a loaded coin,

    Pr[HTHHHTT | Loaded] = 0.6 * 0.4 * 0.6 * 0.6 * 0.6 * 0.4 * 0.4 = .0082944

    카지노 딜러가 나를 속였군. 나쁜놈!

Prior Probability

- 카지노 딜러 왈 “**너 기계학습 여름 학교 강의 이해 했니?**”

- The government regulates Pr[Fair]= 0.99, Pr[Loaded]= 0.01

→ **Prior probability**

Posterior Probability

- **“카지노 딜러가 속였을까요?”** 를 수식으로 나타내면
  - **Pr[Fair | HTHHHTT]** vs. **Pr[Loaded | HTHHHTT]**
- 그런데 이건 계산이 직접 안되어서 Bayes rule을 이용해야함.
- Pr[Fair | HTHHHTT]

= Pr[HTHHHTT | Fair] * Pr[Fair] / P[HTHHHTT]

- **Okay, but Pr[HTHHHTT] ??**
- Pr[HTHHHTT]

= Pr[HTHHHTT | Fair] * Pr[Fair]+ Pr[HTHHHTT | Loaded] * Pr[Loaded]

Posterior Probability

- **Weighted likelihood로 생각하면 쉬움.**

- Pr[Fair | HTHHHTT] = Pr[HTHHHTT | Fair] * Pr[Fair] / P[HTHHHTT]
- Pr[Loaded | HTHHHTT] = Pr[HTHHHTT | Fair] * Pr[Loaded] / P[HTHHHTT]

Pr[Fair | HTHHHTT] vs. **Pr[Loaded | HTHHHTT]**

→→

Pr[HTHHHTT | Fair]*Pr[Fair] vs. **Pr[HTHHHTT|Fair]*Pr[Loaded**

ML vs. MAP

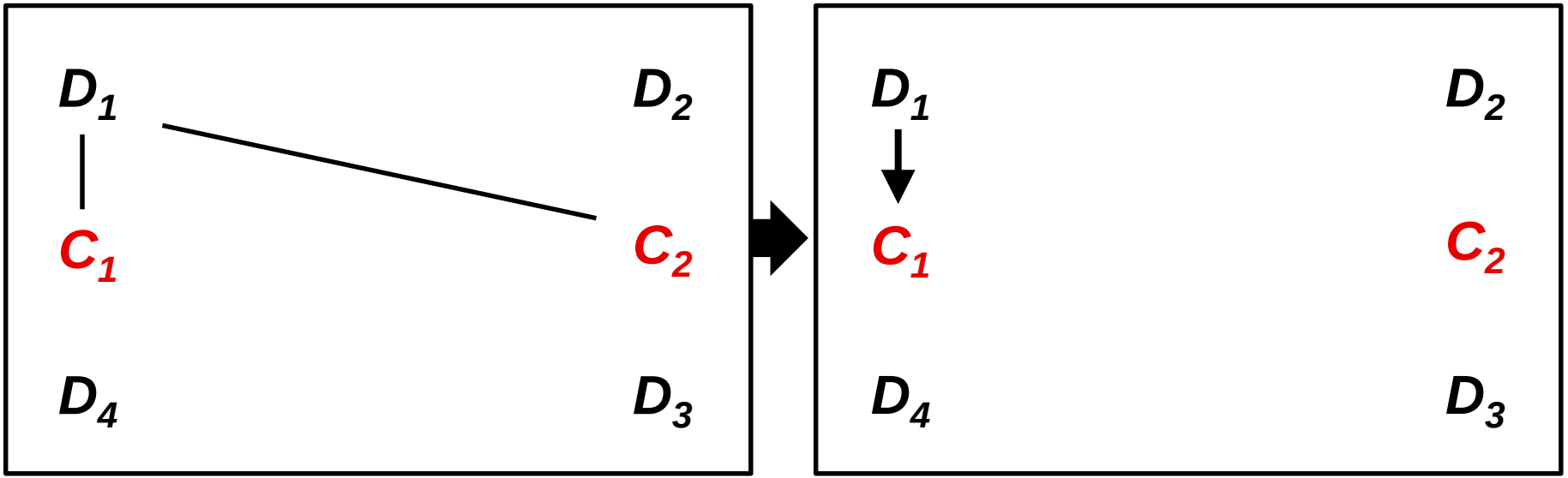
- When we set model parameters,
- **ML (maximum likelihood)**
    - Among all possible model parameter configurations,
    - Select one that is ML.
- **MAP (maximum a posteriori)**
    - Among all possible model parameter configurations,
    - Select one that is MAP.

K Clustering Algorithms

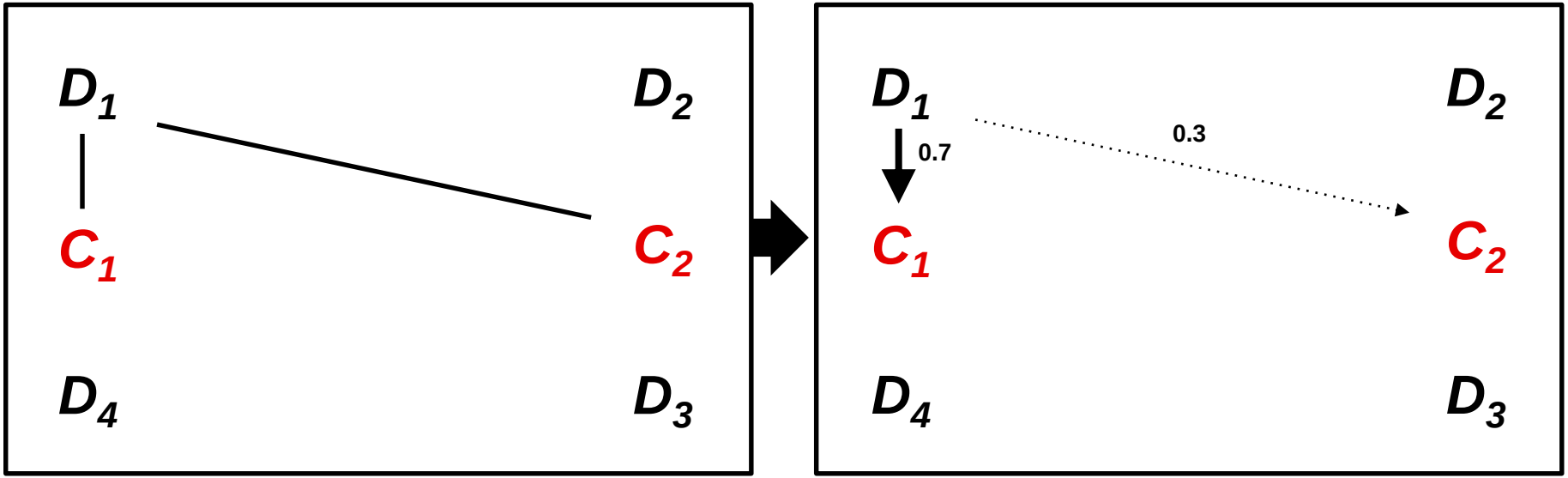
K Clustering Algorithms and EM

- K means clustering algorithm
- A probabilistic version of K means clustering algorithm
- K Gaussian mixtures algorithm

K means clustering algorithm



A probabilistic version of K means clustering algorithm



민주적인 결정 : 모든 데이터가 참여해서 cluster center를 정함.

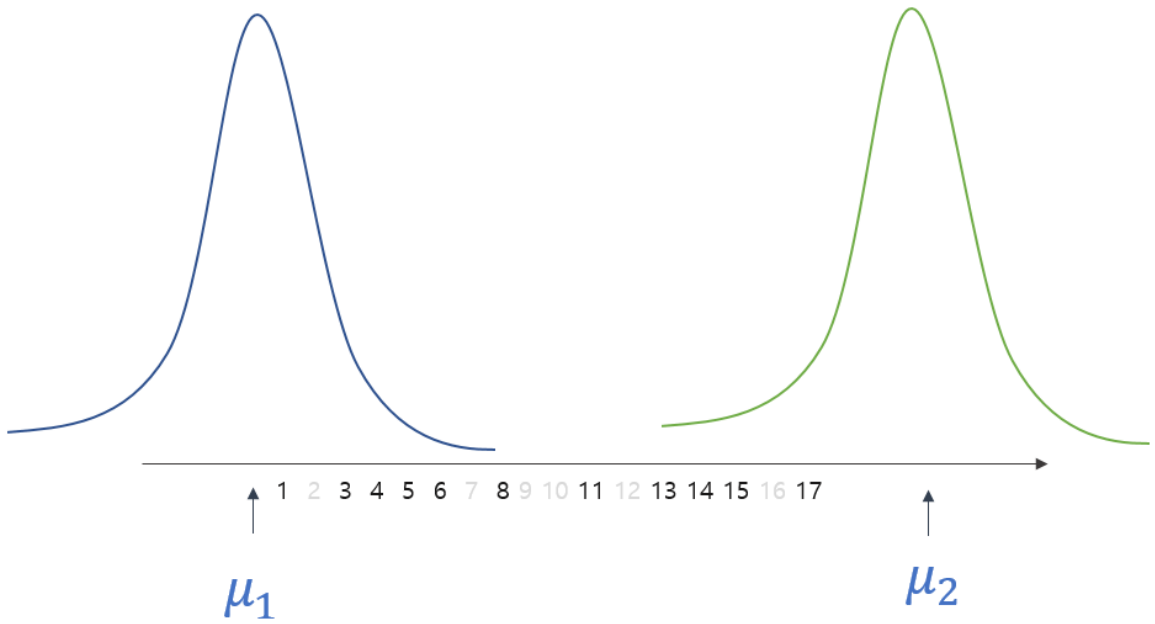
- model parameter수가 증가함.

Two Gaussian Mixture EM Clustering

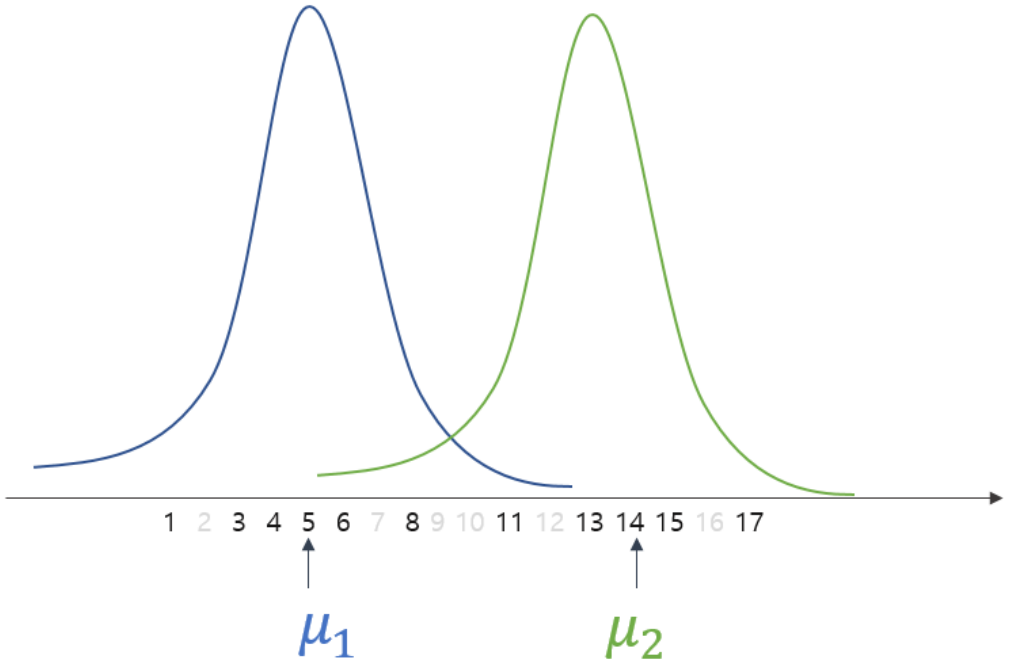
아래 숫자들이 2개의 Gaussian distribution에 의해 생성이 되었는데, 2개의 그룹으로 클러스링을 하고자 한다.

1 3 4 5 6 8 9 11 12 13 14 15 17 18

ESTIMATION1: 아래 2개의 Gaussian distribution



ESTIMATION2: 아래 2개의 Gaussian distribution



이 estimation 이 더 좋은가요? 왜?

If we know which distribution the number are from.

1	3	4	5	6	8	11	13	14	15	17
G1	G1	G1	G1	G1	G1					
						G2	G2	G2	G2	G2

μ_1 = (1 + 3 + 4 + 5 + 6 + 8) / 6 = 4.5

μ_2 = (11 + 13 + 14 + 15 +17)/ 5 = 14

Of course, we do not know which distribution the numbers are from.

1	3	4	5	6	8	11	13	14	15	17
G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1
G2	G2	G2	G2	G2	G2	G2	G2	G2	G2	G2

Democracy again using latent variables:

Of course, we do not know which distribution the numbers are from.

1	3	4	5	6	8	11	13	14	15	17
P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]
P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]

STEP 1: Estimate P[1 from G1], P[3 from G1], …

STEP 2: μ_1 = (P[1 from G1] * 1 + P[3 from G1] * 3 + …) / (P[1 from G1] + P[3 from G1] + … + P[17 from G1])

A brief review of the multi-omics integration methods from a technical perspective

Bayesian Consensus Clustering

- The main question is how to perform simultaneously
  - Source-specific clustering L
  - Consensus clustering C
- Instead of
  - A two-step clustering approach (L then C) or
  - EM-style alternating clustering
  - MCMC sampling is used.
- To perform MCMC-based clustering, sampling should follow posterior probability !

MOLI

- A deep learning based late integration method.

- Metric space vectors are learned separately for each of omics factors.
- All omics metric vectors are concatenated into a single vector.

- The multi-omics learned vector is used for classification with triplet loss and classification loss for drug resposnse.

SNF

- Patient vs. patient matrix are constructed for each omics data.
- Then, each omics matrix is re-computed using a network diffusion method until convergence of all omics matrices is achieved.
- Spectral clustering is performed on the fused network.

NetICS

- Network propagation is performed in two directions, upward and downward.

- Downward propagation from modifiers, such as mutations, DNA methylation, and miRNA, to gene expression and RPPA.
- Upward propagation from gene expression and RPPA to modifiers.

- Sample-specific ranks of each gene are combined to a single global rank.

MONTI

- A gene-centric approach of tensor decomposition to learn feature vectors.

- Learned features are used for classification of cancer subtypes.

Variational Autoencoder

- Autoencoder is used to learn feature vectors, achieving both dimension reduction and non-linear combinations of multi-omics features.

- Learned multi-omics features are used for classification.

- Variational autoencoder using latent variables can be useful when the number of samples is small. Not proven yet.

iCluster

- A Gaussian latent variable model is used for each omics factor.
- To integrate multi-omics data, a joint latent variable model-based clustering method is proposed.
- Clustering performed using an EM framework.

iCluster+

- An extension of iCluster.
- Like in iCluster, latent variables are used for the integration of multi-omics data, but using logistic or multi-category regression function.
- Lasso-penalty is used for reduce the number latent variables.

PINS

- A perturbation-based clustering algorithm.

- Clustering is done two-ways using the original data and the perturbed data.
- Then, final clustering is done by minimizing difference between the two clustering results.

- For multi-omics data, clustering is simply done by
  - computing average matrix of clustering results from each omics data,
  - using existing clustering methods such as HC, PAM, and dynamic tree cut, and
  - performing further-clustering on clusters suggested by entropy or gap statistic.

Appendix

EM 개념을 이해 하셨으면 이제 구체적인 계산 방법을 논의 합니다.

The material is from
Andrew Rosenberg at Queens College / CUNY
http://eniac.cs.qc.cuny.edu/andrew/ml/syllabus.html

Mixture Models

- Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution, π

$$p(x) = \pi_0 f_0(x) + \pi_1 f_1(x) + \pi_2 f_2(x) + \ldots + \pi_k f_k(x)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i f_i(x)}$$

Gaussian Mixture Models

- GMM: the weighted sum of a number of Gaussians where the weights are determined by a distribution, π

$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \ldots + \pi_k N(x|\mu_k, \Sigma_k)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \Sigma_k)}$$

Expectation Maximization

- Both the training of GMMs and Graphical Models with latent variables can be accomplished using Expectation Maximization
  - Step 1: Expectation (E-step)
    - Evaluate the “responsibilities” of each cluster with the current parameters
  - Step 2: Maximization (M-step)
    - Re-estimate parameters using the existing “responsibilities”
- Similar to k-means training.

Latent Variable Representation

- We can represent a GMM involving a latent variable

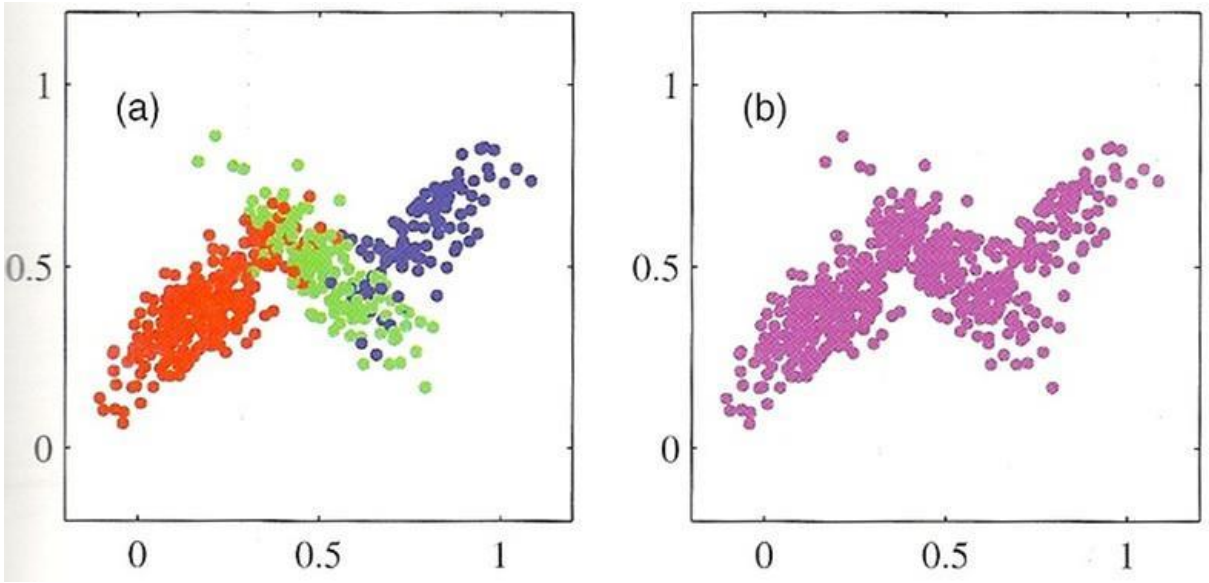
$$p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \Sigma_k) = \sum_z p(z)p(x|z)$$

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k} \qquad p(x|z) = \prod_{k=1}^{K} N(x|\mu_k, \Sigma_k)^{z_k}$$

- What does this give us?

TODO: plate notation

GMM data and Latent variables



One last bit

- We have representations of the joint p(x,z) and the marginal, p(x)…
- The conditional of p(z|x) can be derived using Bayes rule.
    - The **responsibility** that a mixture component takes for explaining an observation x.

$$\tau(z_k) = p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(x|z_j = 1)}$$

$$= \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j, \Sigma_j)}$$

Maximum Likelihood over a GMM

- As usual: Identify a likelihood function

$$\ln p(x|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k) \right\}$$

- And set partials to zero…

Maximum Likelihood of a GMM

- Optimization of means.

$$\ln p(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k) \right\}$$

$$\frac{\partial \ln p(x|\pi,\mu,\Sigma)}{\partial \mu_k} = \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k,\Sigma_k)}{\sum_j \pi_j N(x_n|\mu_j,\Sigma_j} \Sigma_k^{-1}(x_k - \mu_k) = 0$$

$$= \sum_{n=1}^{N} \tau(z_{nk}) \Sigma_k^{-1}(x_k - \mu_k) = 0$$

$$\boxed{\mu_k = \frac{\sum_{n=1}^{N} \tau(z_{nk}) x_n}{\sum_{n=1}^{N} \tau(z_{nk})}}$$

Maximum Likelihood of a GMM

- Optimization of covariance

$$\ln p(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k) \right\}$$

$$\Sigma_k = \frac{1}{\sum_{n=1}^{N} \tau(z_{nk})} \sum_{n=1}^{N} \tau(z_{nk})(x_k-\mu_k)(x_k-\mu_k)^T$$

- Note the similarity to the regular MLE without **responsibility terms**.

Maximum Likelihood of a GMM

- Optimization of mixing term

$$\ln p(x|\pi,\mu,\Sigma) + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)$$

$$0 = \sum_{n=1}^{N}\frac{\pi_k N(x_n|\mu_k,\Sigma_k)}{\sum_j \pi_j N(x_n|\mu_j,\Sigma_j} + \lambda$$

$$\boxed{\pi_k = \frac{\sum_{n=1}^{N}\tau(z_n k)}{N}}$$

MLE of a GMM

$$\boxed{\mu_k = \frac{\sum_{n=1}^{N} \tau(z_{nk}) x_n}{N_k}}$$

$$\boxed{\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \tau(z_{nk})(x_k - \mu_k)(x_k - \mu_k)^T}$$

$$\boxed{\pi_k = \frac{N_k}{N}}$$

$$N_k = \sum_{n=1}^{N} \tau(z_n k)$$

EM for GMMs

- Initialize the parameters
  - Evaluate the log likelihood
- Expectation-step: Evaluate the responsibilities
- Maximization-step: Re-estimate Parameters
  - Evaluate the log likelihood
  - Check for convergence

EM for GMMs

- E-step: Evaluate the Responsibilities

$$\tau(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$$

EM for GMMs

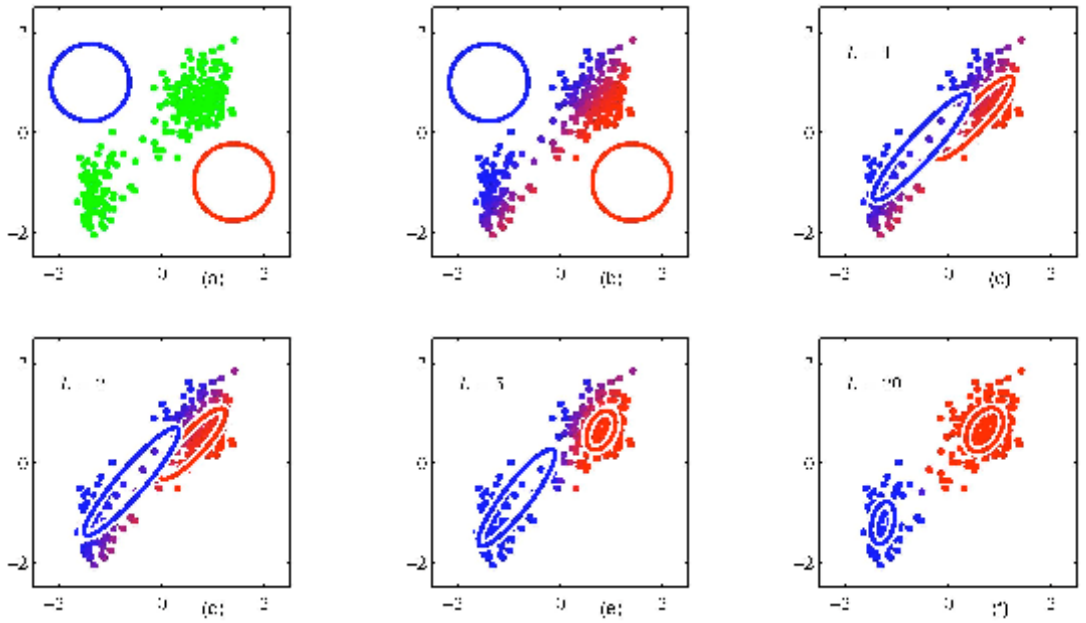
- M-Step: Re-estimate Parameters

$$\boxed{\mu_k^{new} = \frac{\sum_{n=1}^{N} \tau(z_{nk}) x_n}{N_k}}$$

$$\boxed{\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \tau(z_{nk})(x_k - \mu_k^{new})(x_k - \mu_k^{new})^T}$$

$$\boxed{\pi_k^{new} = \frac{N_k}{N}}$$

Visual example of EM



Relationship to K-means

- K-means makes **hard** decisions.
  - Each data point gets assigned to a single cluster.
- GMM/EM makes **soft** decisions.
  - Each data point can yield a posterior p(z|x)
- Soft K-means is a special case of EM.

Soft means as GMM/EM

- Assume equal covariance matrices for every mixture component: $\epsilon \mathbf{I}$
- Likelihood: $p(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi\epsilon)^{M/2}} \exp\left\{-\frac{1}{2\epsilon}\|x - \mu_k\|^2\right\}$
- Responsibilities: $\tau(z_{nk}) = \frac{\pi_k \exp\left\{-\|x_n - \mu_k\|^2/2\epsilon\right\}}{\sum_j \pi_j \exp\left\{-\|x_n - \mu_j\|^2/2\epsilon\right\}}$
- As epsilon approaches zero, the responsibility approaches unity.

Soft K-Means as GMM/EM

- Overall Log likelihood as epsilon approaches zero:

$$\mathbb{E}_z[\ln p(X, Z|\mu, \Sigma, \pi)] \rightarrow -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2 + const.$$

- The expectation of soft k-means is the intercluster variability
- Note: only the means are reestimated in Soft K-means.
    - The covariance matrices are all tied.

General form of EM

- Given a joint distribution over observed and latent variables: $p(X, Z|\theta)$
- Want to maximize: $p(X|\theta)$

1. Initialize parameters
2. E Step: Evaluate: θ^{old}

$$p(Z|X, \theta^{old})$$

3. M-Step: Re-estimate parameters (based on expectation of complete-data log likelihood)

$$\theta^{new} = \text{argmax}_\theta \sum_Z p(Z|X, \theta^{old}) \ln p(X, Z|\theta)$$

1. Check for convergence of params or likelihood

Thank you!

Machine learning models

BIOINFORMATICS 2013

Bayesian consensus clustering

Eric F. Lock, David B. Dunson

[1]Department of Statistical Science, Duke University, Durham, NC 27708, USA and [2]Center for Human Genetics, Duke University Medical Center, Durham, NC 27710, USA

Bayesian consensus clustering

- Bayesian framework for simultaneous estimation of both the consensus clustering and the source-specific clustering
- GOAL: Integrative clustering for subtype identification of breast cancer
- Input
  - mRNA gene expression
  - DNA methylation
  - miRNA expression
  - proteomic data: Reverse phase protein array (RPPA)
- Method overview
  - Finite Dirichlet mixture model for clustering a single dataset
  - Extension of the Dirichlet mixture model to accommodate data from multiple sources
  - Integration over the overall clustering to get the joint marginal distribution
  - Estimation of the integrative clustering mode

Bayesian consensus clustering

- Motivation
  - Separate analysis of each data source may lack power & not capture intersource associations
  - Joint analysis may not capture important features that are specific to each data source

- Existing Methods
  1. Cluster each data source separately, then perform post hoc integration(Consensus clustering)
     - Assumes separate clusterings are known in advance
     - Limits the power to identify and exploit shared structure
  2. Combine all data sources to determine a single ‘joint’ clustering
     - Effective at exploiting shared structure
     - (but,) Unable to recognize features that are specific to each data source

Bayesian consensus clustering

- Bayesian consensus clustering(BCC)
  - Integratively cluster data from multiple sources while preserving the characteristics of each individual data

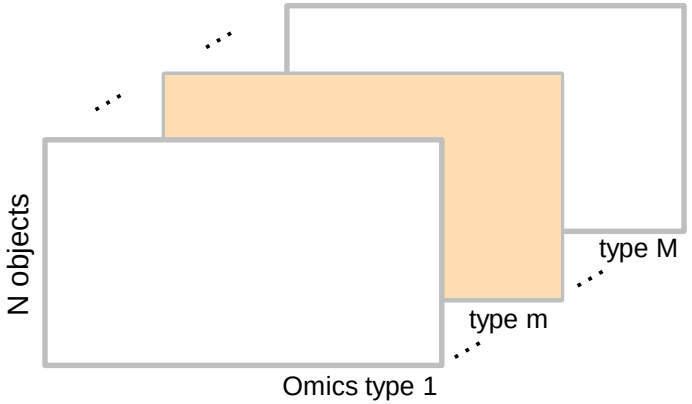
- Differences from traditional consensus clustering
  - Both the source-specific clusterings and the consensus clustering are modeled in a statistical way that allows for uncertainty in all parameters Gibbs sampling procedure
  - The source-specific clusterings and the consensus clustering are estimated simultaneously, rather than in two stages. This permits borrowing of information across sources for more accurate cluster assignments
  - The strength of association to the consensus clustering for each data source is learned from the data and accounted for in the model

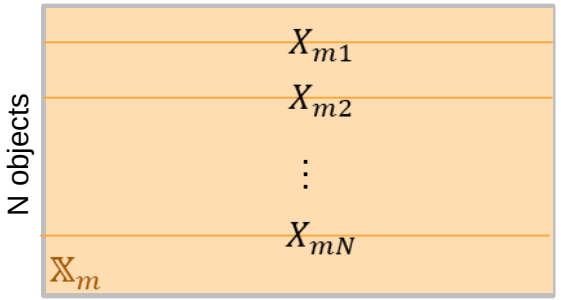
Bayesian consensus clustering

Method

Table 1. Notation

N	Number of objects
M	Number of data sources
K	Number of clusters
$\mathbb{X}_m$	Data source m
X_{mn}	Data for object n, source m
f_m	Probability model for source m
θ_{mk}	Parameters for f_m, cluster k
p_m	Prior distribution for θ_{mk}
C_n	Overall cluster for object n
π_k	Probability that $C_n = k$
L_{mn}	Cluster specific to X_{mn}
ν	Dependence function for C_n and L_{mn}
α_m	Probability that $L_{mn} = C_n$





$\mathbb{X}_m \sim f_m$

Each data source is parameterized with **parameters of clusters**

- Parameters of K-component mixture distribution $(\theta_{m1}, \theta_{m2}, \ldots, \theta_{mk})$

$\mathbb{X}_m \sim f_m(\cdot|\theta_{mk})$

Bayesian consensus clustering

Method

Table 1. Notation

N	Number of objects
M	Number of data sources
K	Number of clusters
$\mathbb{X}_m$	Data source m
X_{mn}	Data for object n, source m
f_m	Probability model for source m
θ_{mk}	Parameters for f_m, cluster k
p_m	Prior distribution for θ_{mk}
C_n	Overall cluster for object n
π_k	Probability that $C_n = k$
L_{mn}	Cluster specific to X_{mn}
ν	Dependence function for C_n and L_{mn}
α_m	Probability that $L_{mn} = C_n$

Clustering

- Source-specific clustering

$$\mathbb{L}_m = (L_{m1}, \ldots, L_{mN})$$

- Overall clustering (= integrative clustering)

$$\mathbb{C} = (C_1, \ldots, C_N)$$

- Source-specific clusters are dependent on the overall clustering

$$P(L_{mn} = k|C_n) = \nu(k, C_n, \alpha_m)$$

$\mathbb{X}_m$ are independent of $\mathbb{C}$ conditional on the source-specific clustering $\mathbb{L}_m$

$$P(L_{mn} = k|X_{mn}, C_n, \theta_{mk}) \propto \nu(k, C_n, \alpha_m) f_m(X_{mn}|\theta_{mk})$$

such that $\nu(L_{mn}, C_n, \alpha_m) = \begin{cases} \alpha_m \text{if} C_n = L_{mn} \\ \frac{1-\alpha_m}{K-1} \text{otherwise} \end{cases}$

where $\alpha_m \in [\frac{1}{K}, 1]$ controls the adherence of data source m to overall clustering

- If α_m = 1 : Source-specific clustering = Overall clustering

Bayesian consensus clustering

Method

Table 1. Notation

N	Number of objects
M	Number of data sources
K	Number of clusters
$\mathbb{X}_m$	Data source m
X_{mn}	Data for object n, source m
f_m	Probability model for source m
θ_{mk}	Parameters for f_m, cluster k
p_m	Prior distribution for θ_{mk}
C_n	Overall cluster for object n
π_k	Probability that $C_n = k$
L_{mn}	Cluster specific to X_{mn}
ν	Dependence function for C_n and L_{mn}
α_m	Probability that $L_{mn} = C_n$

Clustering

- Probability that an object belongs to the overall cluster k

$$\pi_k = P(C_n = k)$$

- Assume that a Dirichlet (β) prior distribution for $\Pi = (\pi_1, \ldots, \pi_K)$.

$$P(L_{mn} = k|\Pi) = \pi_k\alpha_m + (1 - \pi_k)\frac{1 - \alpha_m}{K - 1}$$

- Bayesian rule for conditional distribution of $\mathbb{C}$

$$P(C_n = k|\mathbb{L}, \Pi, \alpha) \propto \pi_k \prod_{m=1}^{M} \nu(L_{mn}, k, \alpha_m)$$

Estimation of clustering

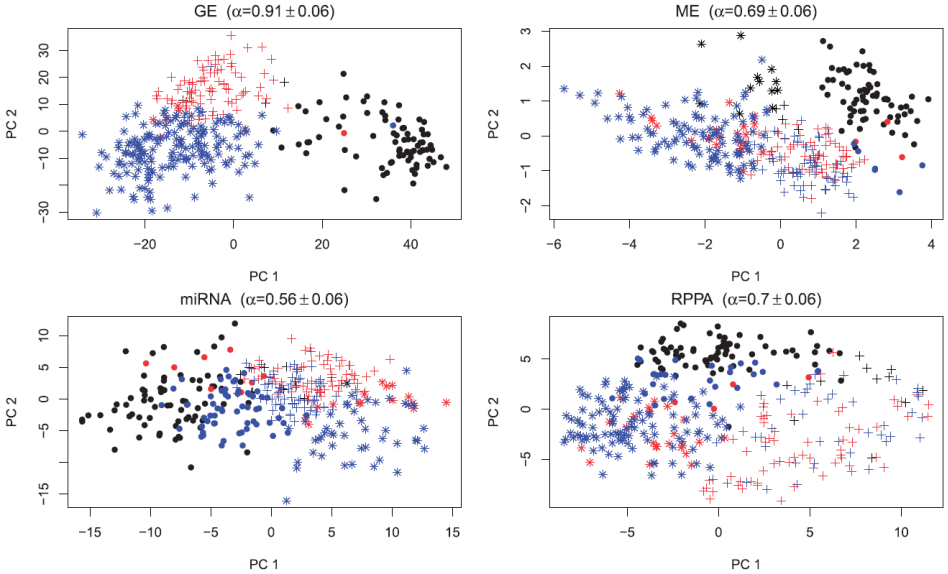
: Bayesian framework for estimation of the integrative clustering model

: With Gibbs sampling & Markov Chain Monte Carlo (MCMC)

- Gibbs sampling procedure to approximate the posterior distribution for the parameters
- Markov chain Monte Carlo (MCMC) proceeds by iteratively sampling from the following conditional posterior distributions

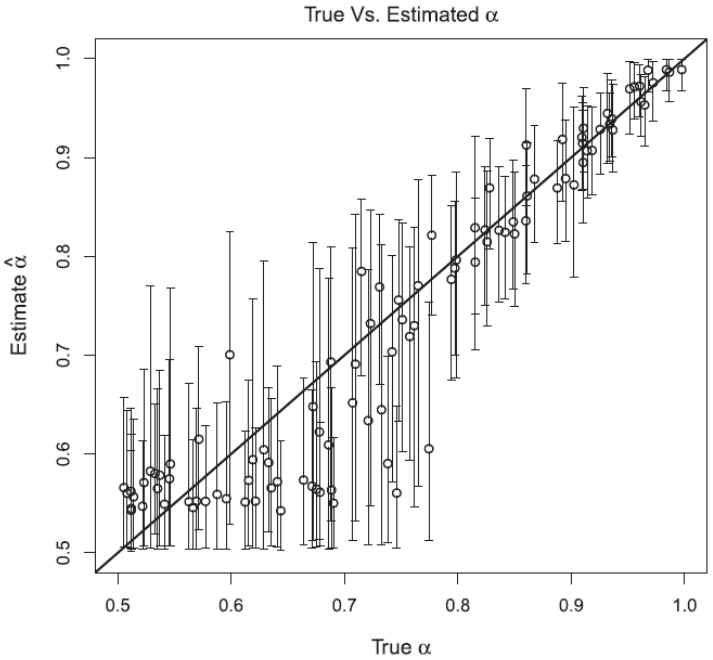
Bayesian consensus clustering

- Result
  - More robust than joint clustering of all data sources
  - More powerful than clustering each data source independently
  - Applied to subtype identification of breast cancer
  - The figure on the right are the PCA plots for each data source
  - Sample points are colored by overall cluster:
    black: cluster1, red: cluster2, blue: cluster3
  - Symbols indicate source-specific cluster:
    circle: cluster1, plus: cluster2, asterisk: cluster3



Bayesian consensus clustering

- Result
  - Estimated *alpha* versus true *alpha* for 100 randomly generated simulations



BIOINFORMATICS 2019

mirTime: identifying condition-specific targets of microRNA in time-series transcript data using Gaussian process model and spherical vector clustering

Hyejin Kang, Hongryul Ahn, Kyuri Jo, Minsik Oh, Sun Kim

[1]Department of Computer Science and Engineering, Seoul National University, Seoul, Republic of Korea, [2]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul, Republic of Korea and [3]Bioinformatics Institute, Seoul National University, Seoul, Republic of Korea

mirTime

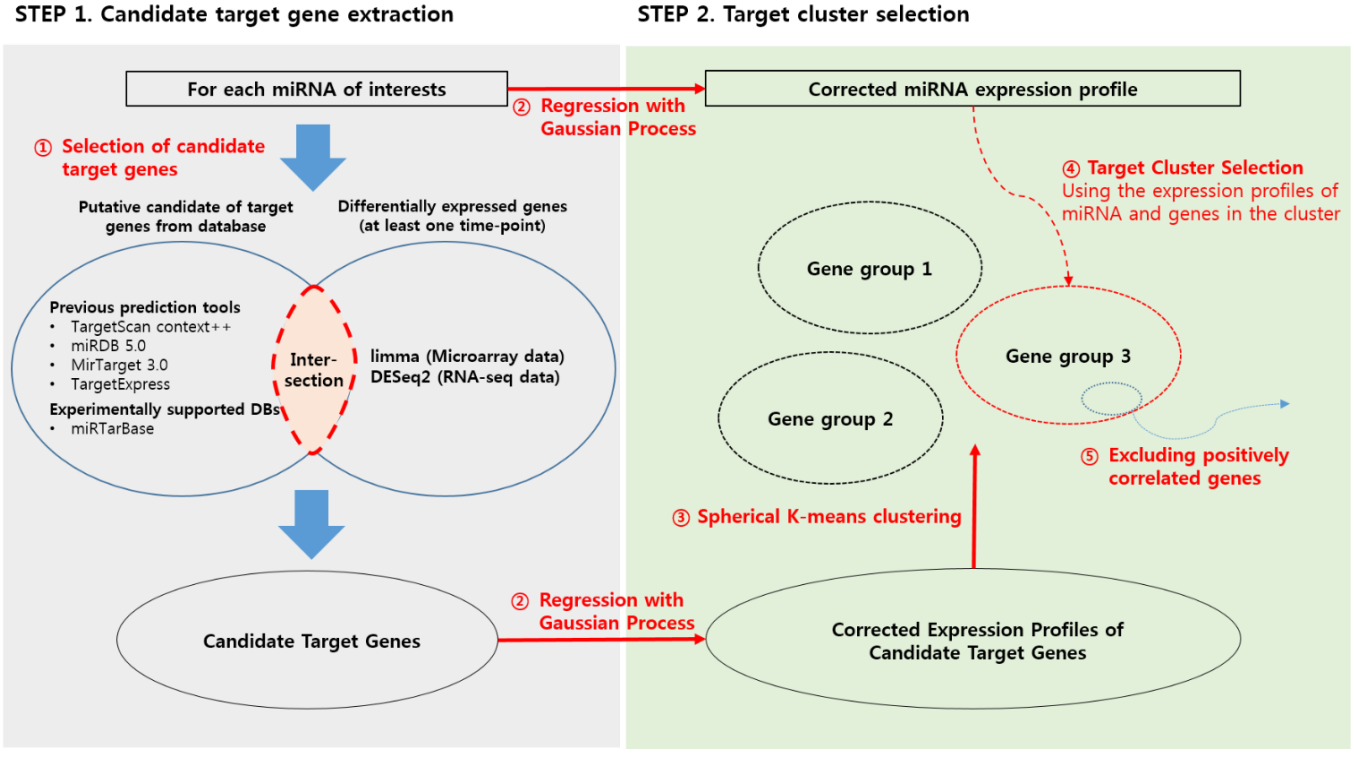
- GOAL: Identify miRNA-target gene for a specific condition
- Input
  - Time-series gene (mRNA) expression data
  - Time-series miRNA expression data
- Method overview
  - Predict candidates of miRNA-target genes based on sequence feature
  - Estimate regression lines of miRNA and gene expressions by Gaussian process
  - Infer negative regulation between miRNA-target gene expressions
  - Generate miRNA-target gene clusters

mirTime

- Method overview
  - Microarray data
    - Susanne R, Petr V. N, Demetra P et al. Dynamic regulation of microRNA expression following interferon-γ-induced gene transcription *RNA Biology* 2012 9(7):978-989
    - Petr V. N, Susanne E. R, Arnaud M et al. Interplay of microRNAs, transcription factors and target genes: linking dynamic expression changes to function *NAR* 2013 41(5):2817
    - A375 melanoma cancer cell
    - Different Time Points for miRNA and mRNA each
      - miRNA: 9 time points (0, 0.5, 3, 6, 12, 24, 48, 72, 96h)
      - mRNA: 6 time points (0, 3, 12, 24, 48, 72)
    - 2 replicates (3 replicates for time points in bold)
  - RNA-sequencing data
    - Baran-Gale J, Purvis JE, Sethupathy P. An integrative transcriptomics approach identifies **miR-503** as a candidate master regulator of the estrogen response in MCF-7 breast cancer cells. *RNA* 2016 Oct;22(10):1592-603
    - MCF-7 breast cancer cell
    - 10 time points (0, 1, 2, 3, 4, 5, 6, 8, 12, 24h) miRNA, mRNA data
    - 3 replicates

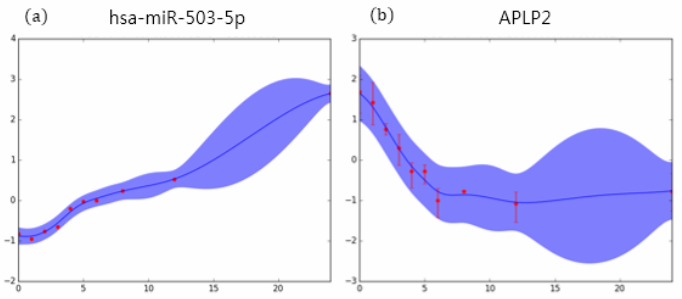
mirTime

- Workflow



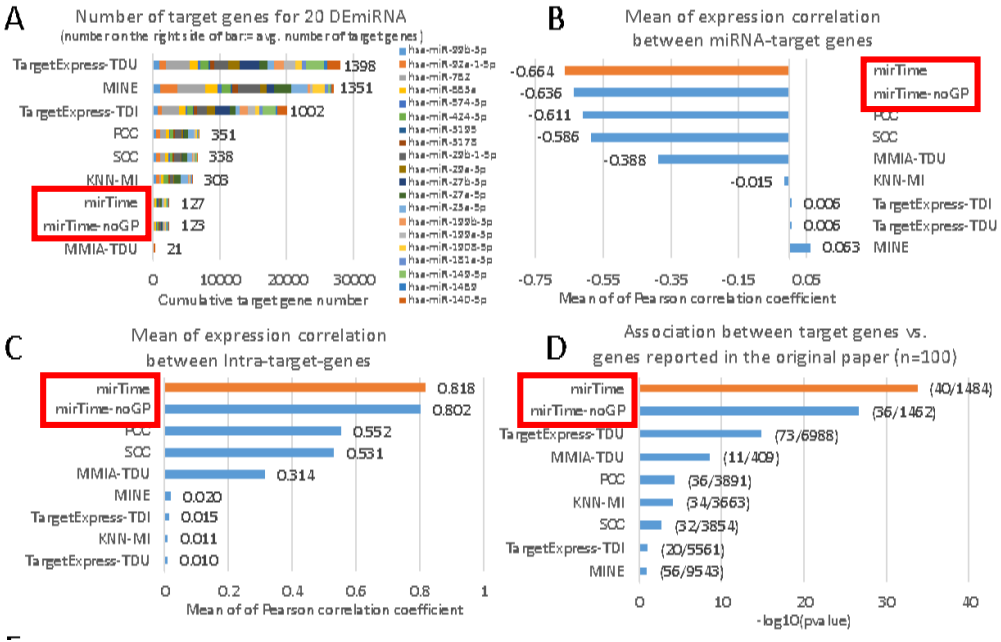
mirTime

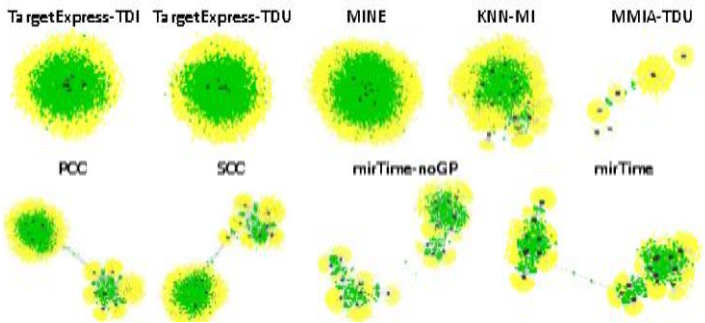
- Calculation of Gaussian Process (GP) – weight vectors
  - From the trained regression model, the mean and variance vectors were inferenced for each point with input values at equal intervals from the minimum time point to maximum time point given experiment data.
  - Minimum interval between time axes is set to coincide with the minimum interval of the actual data.
  - GP-weight vector was calculated for each miRNA and gene with an mean and variance vector.
    - The larger the variance value at the inferred point, the lower the reliability of the regression result.
    - GP-weight vector is obtained by dividing the mean value deduced from each time axis by the variance value and used in the following analysis.



mirTime

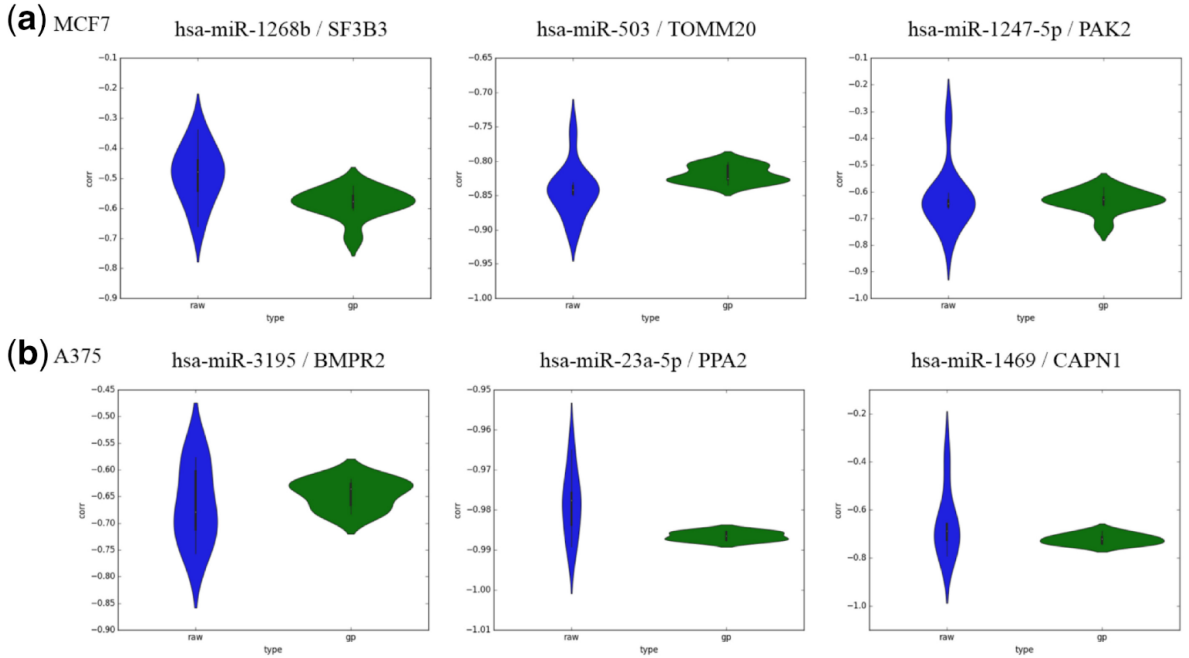
- Result – A375 melanoma cancer cell data





mirTime

- Result – A375 melanoma cancer cell data



BIOINFORMATICS 2019

MOLI: multi-omics late integration with deep neural networks for drug response prediction

Hossein Sharifi-Noghabi, Olga Zolotareva, Colin C. Collins and Martin Ester

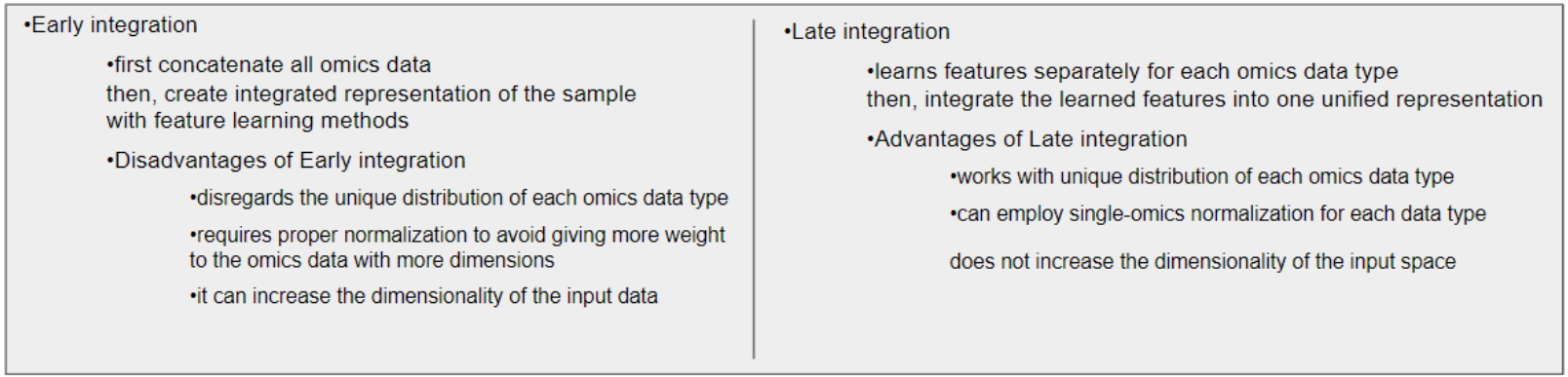
[1]School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada, [2]Vancouver Prostate Centre, Vancouver, BC V6H 3Z6, Canada, [3]International Research Training Group Computational Methods for the Analysis of the Diversity and Dynamics of Genomes and Genome Informatics, Faculty of Technology and Center for Biotechnology, Bielefeld University, 33615 Germany and [4]Department of Urologic Sciences, University of British Columbia, Vancouver, BC V5Z 1M9, Canada

MOLI

- Multi-Omics Late Integration method based on deep neural networks
- GOAL: Drug response prediction
- Input
  - Somatic mutation
  - Copy Number Aberration (CNA)
  - Gene (mRNA) expression
- Method overview
  - Learns features for each omics type with encoding sub-networks
  - Concatenates the features into one representation *(Late Integration)*
  - Optimize the representation via a combined cost function
    - Binary cross-entropy loss
    - Triplet loss

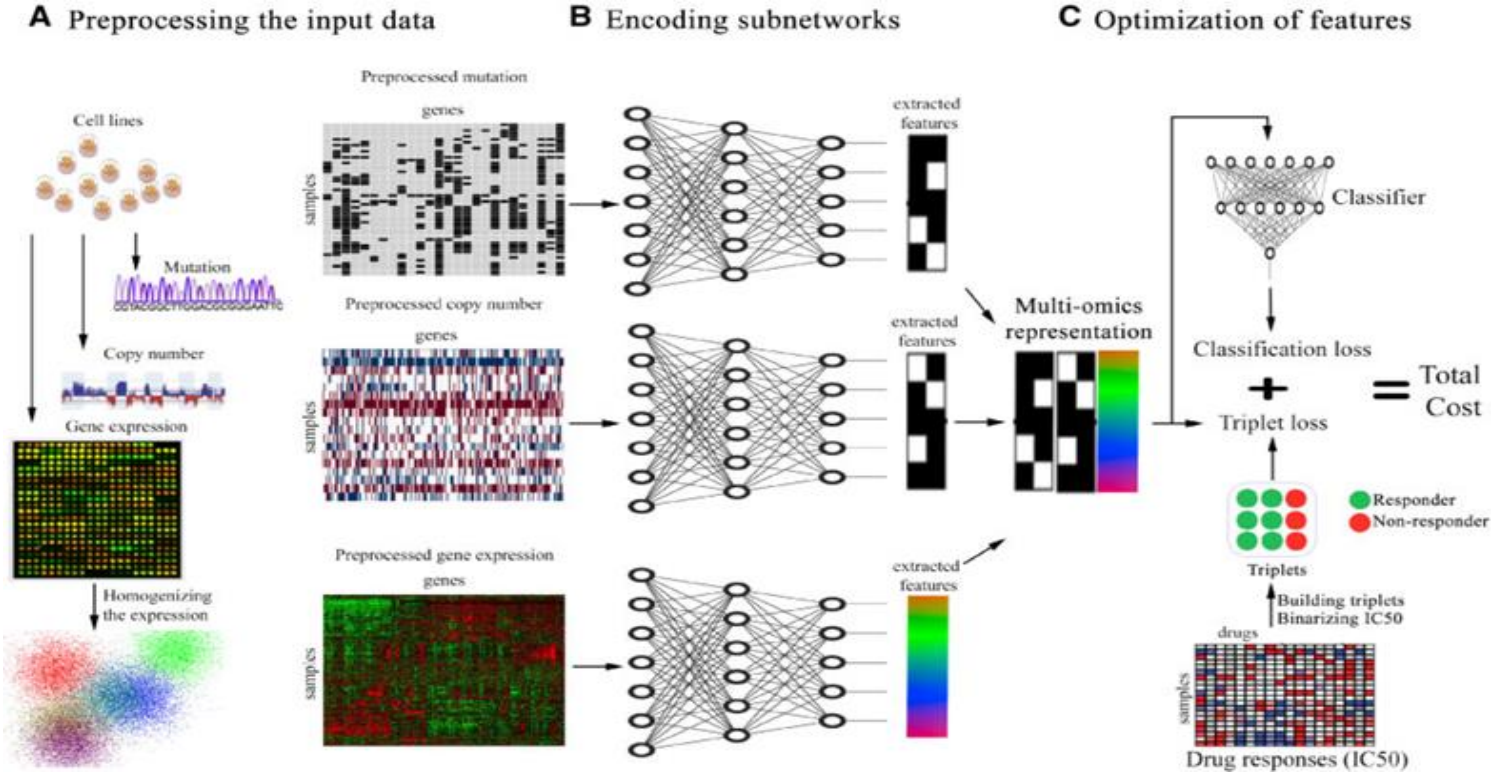
MOLI

- Contribution
  - First end-to-end late integration method with deep neural networks that optimizes the representation via a combined cost function
  - Clinical utility
    - Translatable to actual patients
    - Good performance on in-vivo data as well, not just in-vitro data
  - Early integration vs. Late integration



MOLI

- Method workflow



MOLI

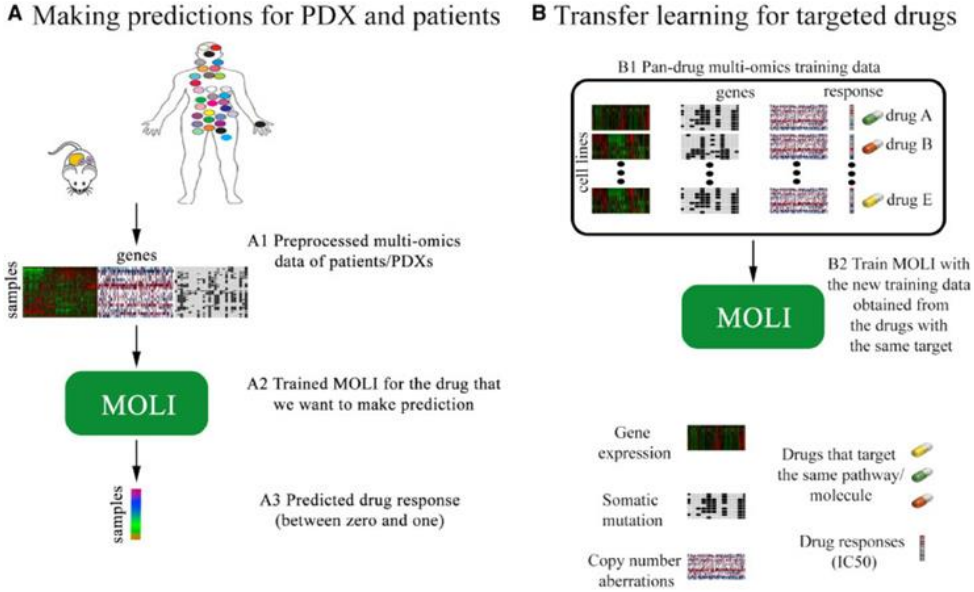
- Method
  - Preprocess: Mutation, CNA, gene expression
  - Feature learning
    - for each omics with type-specific encoding sub-networks
  - Concatenate into one representation
  - Optimization of combined cost
    - triplet loss
      - makes the representations of responder cell lines more similar to each other
      - and different from the representations of non-responder cell lines
    - classification loss(classifier sub-network)
      - makes the representations predictive of the IC50 values
  - Train the entire network end-to-end

MOLI

- Result
  - Validated MOLI on in vitro (PDX) and in vivo (TCGA patients) datasets for five chemotherapy agents and two targeted therapeutics
  - Compared with the state-of-the-art single-omics & early integration multi-omics methods
    - MOLI achieves higher prediction accuracy in external validations
  - MOLI models trained on in vitro data translate well to in vivo data
    - Trained on pan-drug input for the epidermal growth factor receptor(EGFR) inhibitors
    - Response predicted for breast, lung, kidney and prostate cancers from TCGA patients had statistically significant associations with some of the genes in the EGFR pathway
    > MOLI captures biological aspects of the response

MOLI

- Result
  - Used MOLI to make predictions for PDX/patient inputs during external validation
  - Combining targeted drugs that target the same pathway or molecule to make a pan-drug training dataset for MOLI



Network-based methods

Nature methods 2014

Similarity network fusion for aggregating data types on a genomic scale

Bo Wang, Aziz M Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains & Anna Goldenberg

[1]School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada, [2]Vancouver Prostate Centre, Vancouver, BC V6H 3Z6, Canada, [3]International Research Training Group Computational Methods for the Analysis of the Diversity and Dynamics of Genomes and Genome Informatics, Faculty of Technology and Center for Biotechnology, Bielefeld University, 33615 Germany and [4]Department of Urologic Sciences, University of British Columbia, Vancouver, BC V5Z 1M9, Canada

SNF

- Multi-omics integrative network fusion method for constructing networks of cancer samples
- GOAL: Identify cancer subtypes to predict survival
- Input
  - mRNA expression
  - DNA methylation
  - miRNA
- Method Overview
  - Construct sample-similarity network for each omics type
  - Integrate similarity networks based on message passing theory
  - Cluster integrated network to group cancer patients

SNF

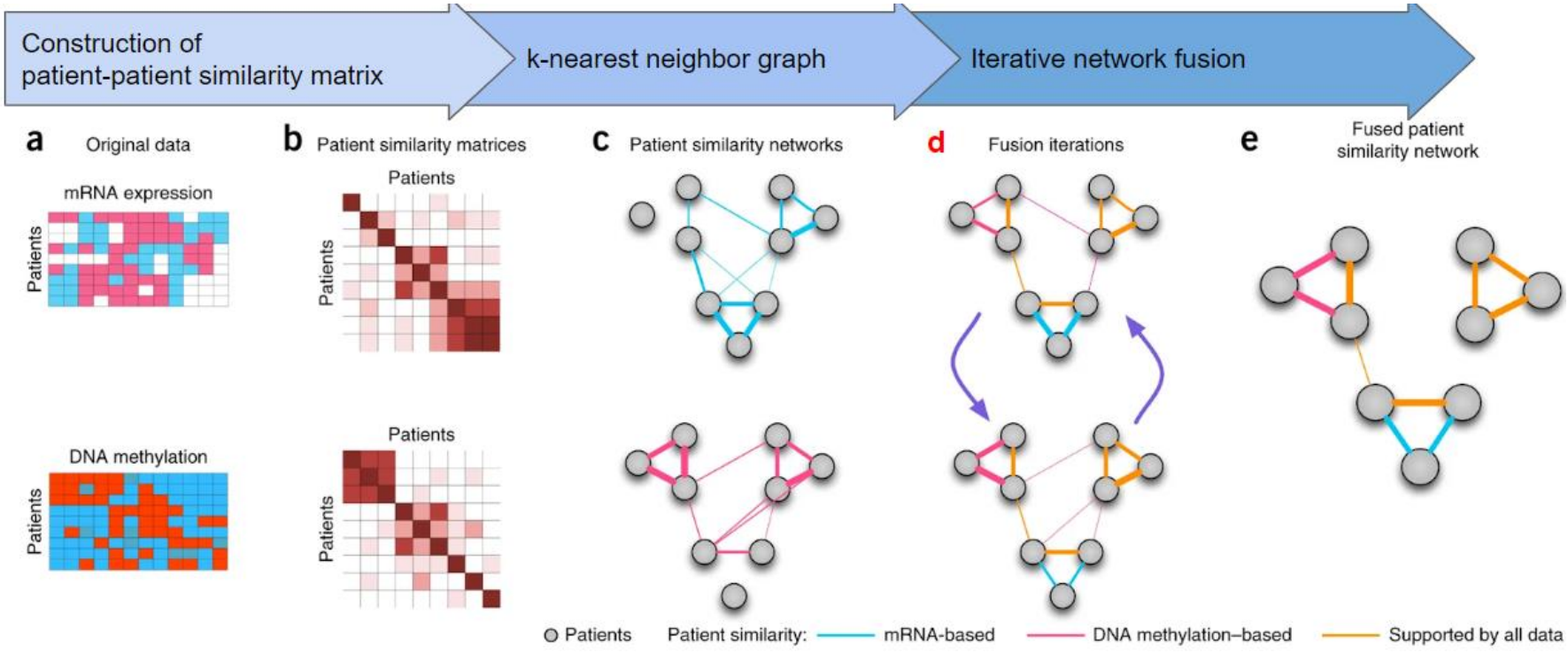
- Problem definition: How to integrate multi-omics data?
  - Challenges

    1. High dimensional, Low sample data
    2. Differences in scale, collection bias and noise in each data set
    3. Information from different data types are complementary

A. Similarity network fusion (SNF)

SNF

- Method workflow



SNF

- Method Details
  - Iteratively updates every networks from each data type, making it more similar to others
  - Based on message passing theory, weight from the network of data type (1) is propagated based on the topology of network from data type (2), iteratively ((2) → (1), (1) → (2), … for t steps)

$$\mathbf{P}_{t+1}^{(1)} = \mathbf{S}^{(1)} \times \mathbf{P}_{t}^{(2)} \times (\mathbf{S}^{(1)})^{T}$$

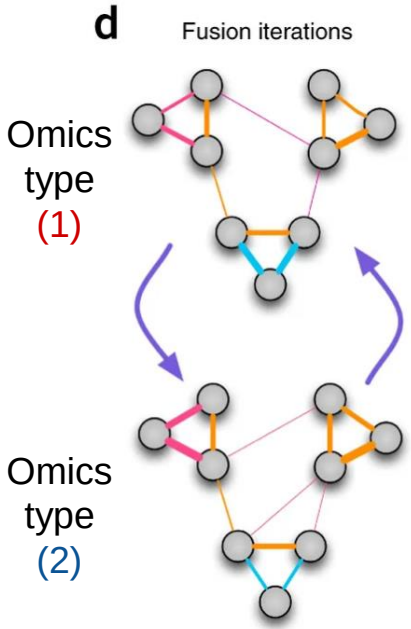
$$\mathbf{P}_{t+1}^{(2)} = \mathbf{S}^{(2)} \times \mathbf{P}_{t}^{(1)} \times (\mathbf{S}^{(2)})^{T}$$

P : Normalized weight matrix **W**

S : K nearest neighbors (KNN) matrix of **P**

  - After t steps, it is assumed that networks from (1) and (2) resemble each other

Therefore, overall status matrix $\mathbf{P}^{(c)} = \frac{\mathbf{P}_{t}^{(1)} + \mathbf{P}_{t}^{(2)}}{2}$



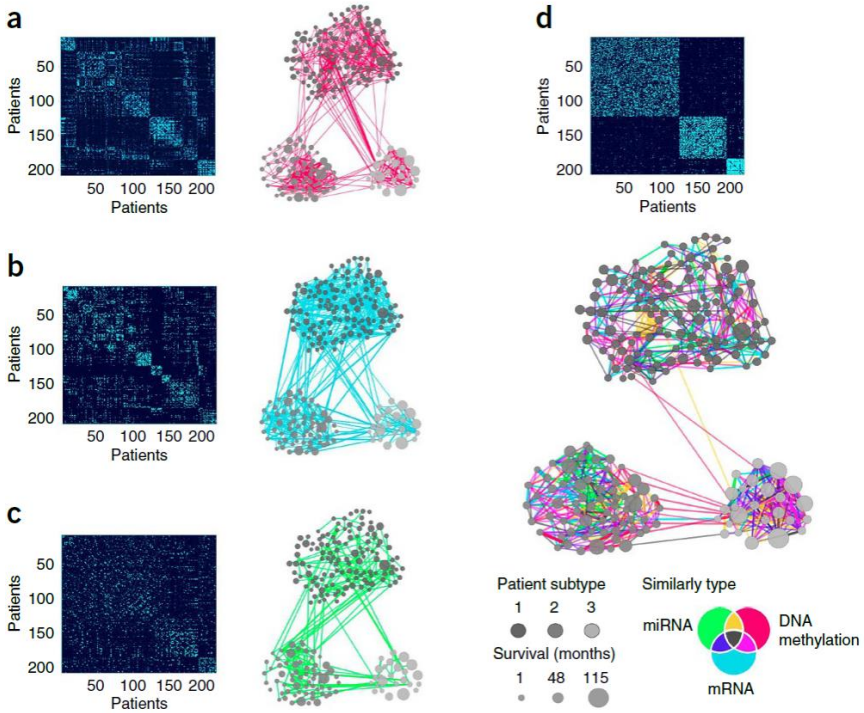
SNF

- Result
  - Patient-to-patient similarity network after similarity network fusion
  - Spectral clustering shows that clustering representation is preserved for each omics data and integrated omics data
  - In terms of survival prediction, integrating multiple omics type shows better performance

Table 1 | SNF-based analysis versus individual data types

Cancer type	mRNA expression	DNA methylation	miRNA	SNF
GBM (3 clusters)	0.54	0.11	0.21	2.0×10^{-4}
BIC (5 clusters)	0.03	0.05	0.30	1.1×10^{-3}
KRCCC (3 clusters)	0.20	0.61	0.17	2.9×10^{-2}
LSCC (4 clusters)	0.06	0.26	0.46	2.0×10^{-2}
COAD (3 clusters)	0.18	0.04	0.46	8.8×10^{-4}

Analysis using Cox log-rank test *P* values.



Bioinformatics 2018

Network-based integration of multi-omics data for prioritizing cancer genes

Christos Dimitrakopoulos, Sravanth Kumar Hindupur, Luca Ha¨fliger, Jonas Behr, Hesam Montazeri, Michael N. Hall and Niko Beerenwinkel

[1]Department of Biosystems Science and Engineering, ETH Zürich, Basel, Switzerland, [2]SIB Swiss Institute of Bioinformatics, Basel, Switzerland and [3]Biozentrum, University of Basel, Basel, Switzerland

NetICS

- Network-based cancer gene prioritization method exploiting multi-omics data
- GOAL: Prioritize cancer genes in perspective of mediator effect
- Input
  - Integrated network from multiple omics networks
  - 5 TCGA dataset ((UCEC, HCC, BLCA, BRCA, LUSC),
    - RNA-seq data
    - HiSeq miRNA sequencing data
    - Methylation beadchip data
- Method Overview
  - For each cancer sample, identify genes with aberration or differential expression
  - Conduct bidirectional network propagation
    - Forward propagation from genes with aberration
    - Backward propagation from genes with differential expression
  - Integrate propagation score from forward and backward propagation to measure mediator effect

NetICS

- Problem definition:

  How genomic aberration affects molecular makeup of a cell (e.g. transcriptome, proteome) in cancer?

  i.e. Decode the functional relationship between aberration events and changes in gene/protein expression

  A. (Previous studies) Find driver mutations!

  Challenges

  1. Driver mutations are rare mutations
  2. Other factors —epigenetic changes, miRNA differential expressions —should be considered

  A. Considering multiple omics type and gene interactions, find mediator genes in cancer progression

  Assumption

  : Mediator genes are genes that

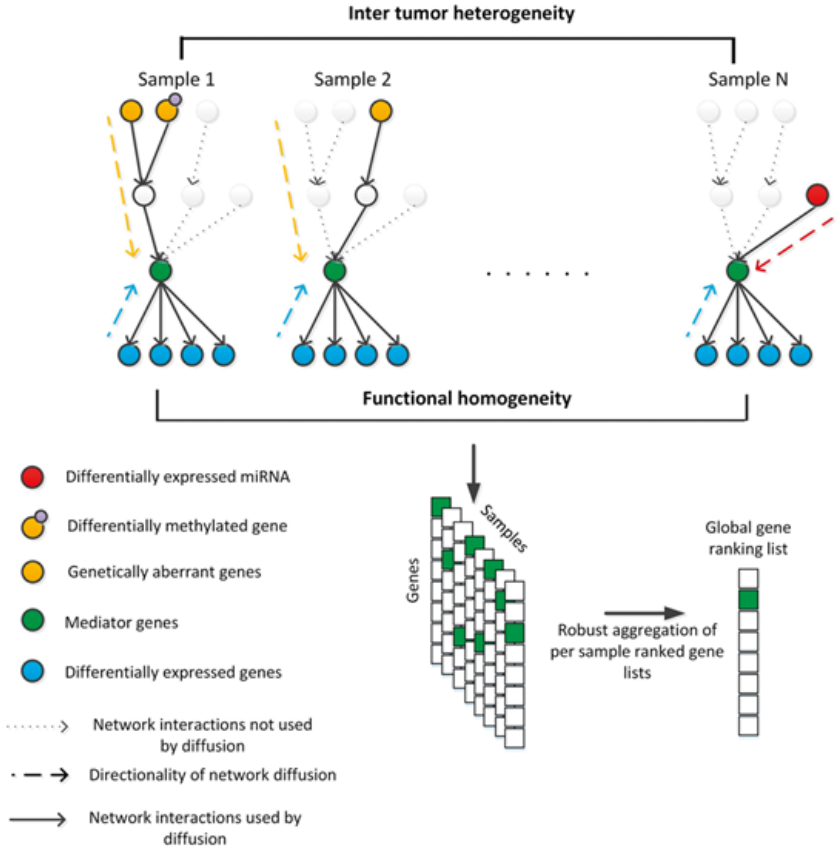
  1. are affected by genetic mutations (e.g. somatic mutations, CNV) & epigenetic aberration
  2. has impact on differential molecular profile

NetICS

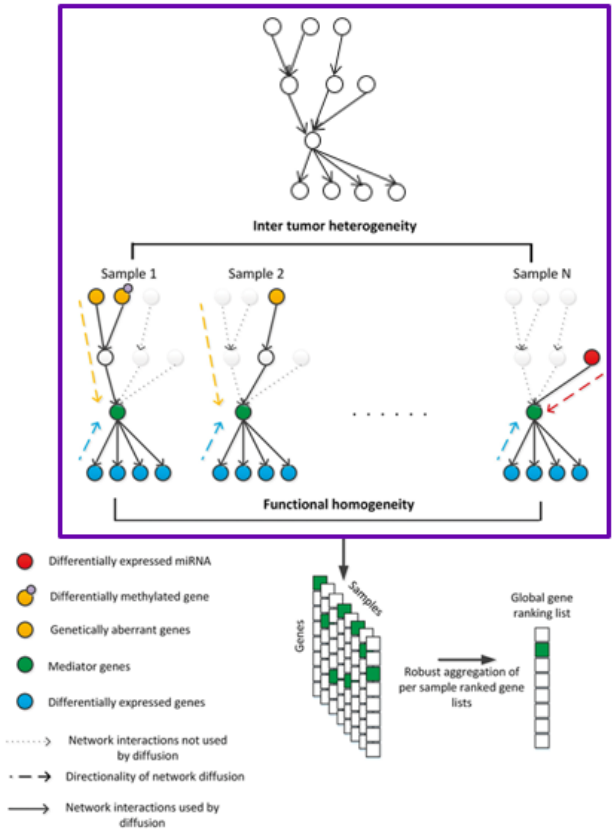
- Dataset
  - Functional interaction network & miRNA-gene network
  - 5 TCGA dataset (UCEC, HCC, BLCA, BRCA, LUSC)
    - Tumor samples & Normal samples
      1. RNA-seq data
      2. HiSeq miRNA sequencing data
      3. Methylation beadchip data
- Methodology
  - Prioritize genes by their mediator effect
  - Make use of network propagation!
- Keywords
  - Sample-specific network, network propagation

NetICS

- Method workflow



NetICS



- Method
  1. Template network construction

     : Directed, unweighted edges

     : Concatenating functional interaction & miRNA-gene interaction :
  2. Network propagation

     For each sample,

     ◦ Bidirectional network propagation

     - Forward propagation from genes with aberration event (Somatic mut, CNV)

     → To find genes that are influenced by genetic mutations

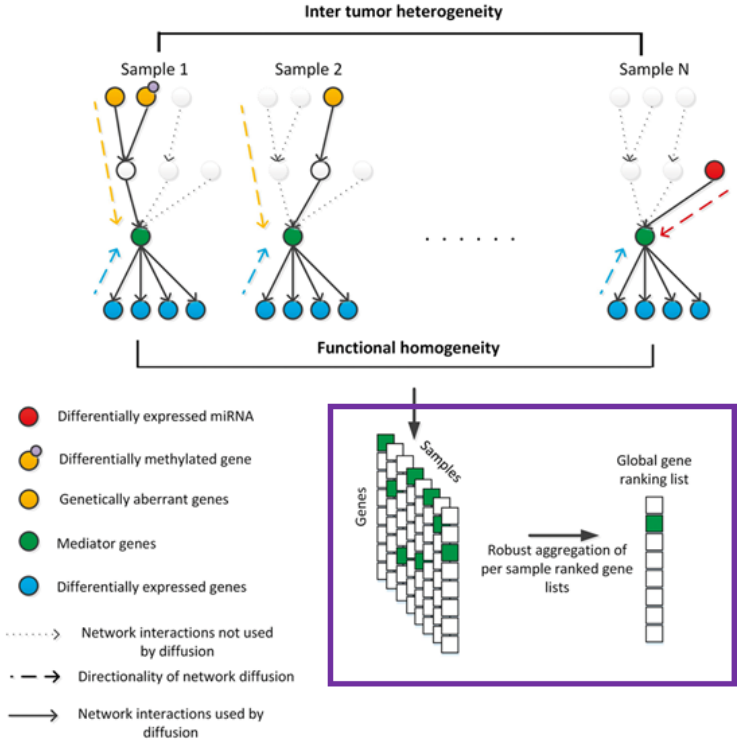
     - Backward propagation from genes with differential expression (DEGs, DMRs)

     → To find genes that have influence on differential gene expression

     ◦ Integrate propagation scores with element-wise multiplication

NetICS

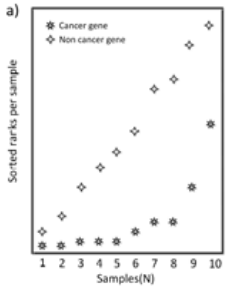
- Method

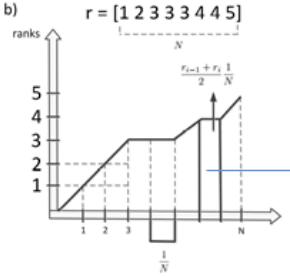


3. Integrating ranked gene across samples

- Assumption

  :Cancer genes are expected to have low ranks across samples



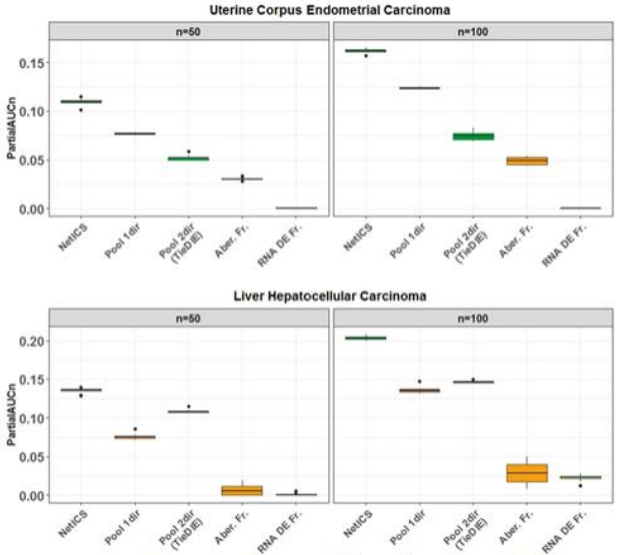


AUC

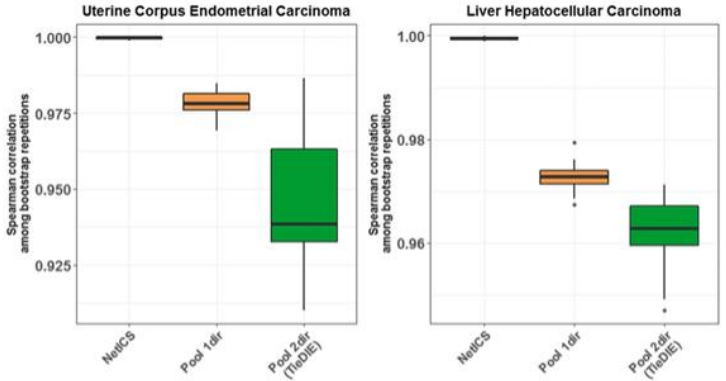
: integrated rank of a gene

NetICS

- Result
  - Prediction accuracy of detecting previously reported cancer-related genes
    - outperforms compared to existing prioritization methods
    - yields robust results



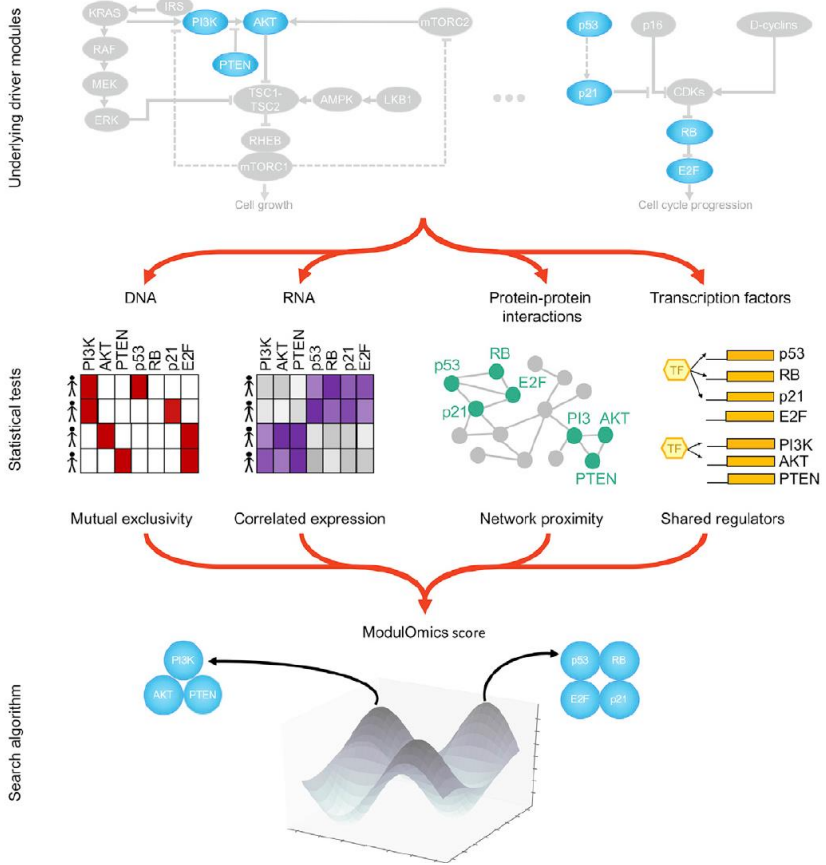
Comparison with other model



Stability of ranked gene list

Silverbush et al., Simultaneous Integration of Multi-omics Data Improves the Identification of Cancer Driver Modules, ***Cell Systems***, 2019

ModulOmics



- Find cancer driver modules using 4 different omics layers.

    1. PPI connectivity
    2. DNA mutual exclusivity
    3. TF-TG relationship
    4. RNA co-expression

- Define scoring schemes of candidate modules for each of the four different omics layers.

- Get optimal modules by two-step optimization.

Step 1: Initialize candidate modules with integer linear programming

Step 2: Further optimize candidate modules by stochastic search

Representation or hidden feature learning

Representation or hidden feature learning intro

- Relationships among biological entities are too complicated
- The number of dimensions is high
- An effective approach to these problems is to perform dimension reduction analysis
  - Search of features that combine multi-omics relationships automatically
  - Then, perform classification using selected features

Bioinformatics 2019

Tensor decomposition (linear combination of features)

MONTI: A multi-omics non-negative tensor decomposition framework for the integrated analysis of cancer subtypes

Inuk Jung, Minsu Kim, Sungmin Rhee, Sangsoo Lim, and Sun Kim

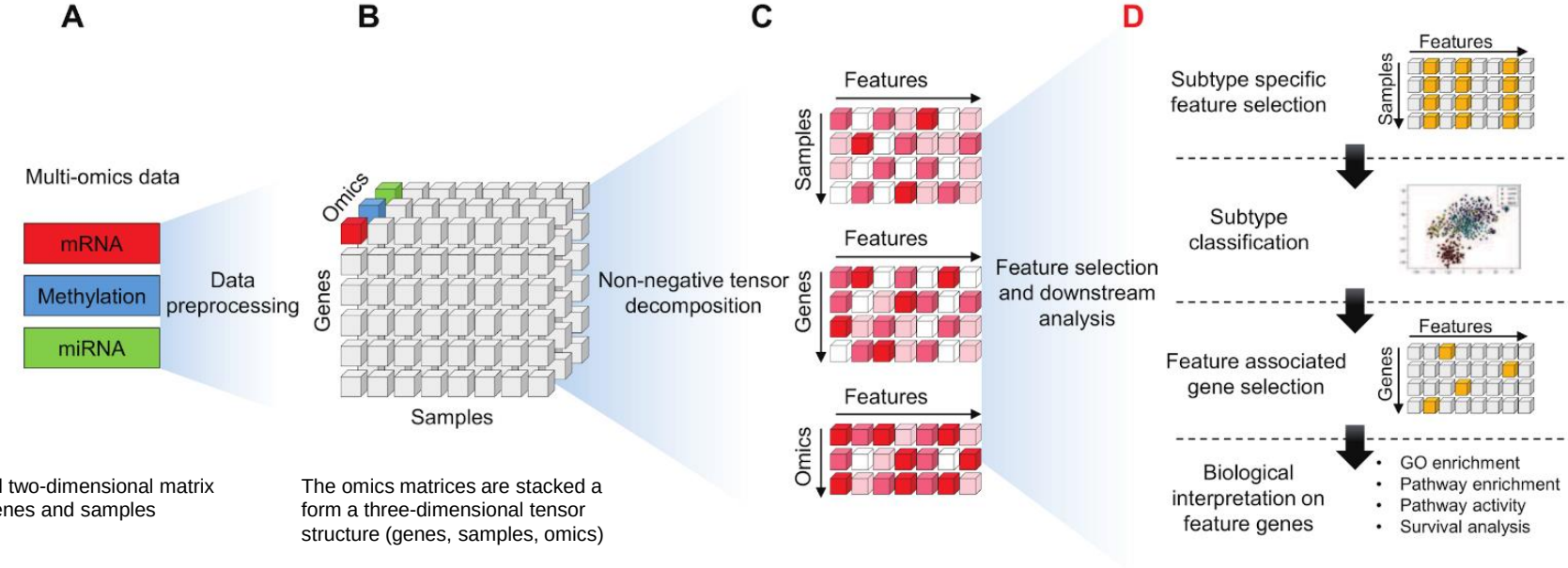
[1]Department of Computer Science and Engineering, Kyungpook National University, Buk-gu, Daegu, 41566, Republic of Korea, [2]Bioinformatics Institute, Seoul National University, Gwanak-Gu, Seoul, 08826, Republic of Korea, [3]Department of Computer Science and Engineering, Seoul National University, Gwanak-Gu, Seoul, 08826, Republic of Korea and [4]Interdisciplinary Program in Bioinformatics, Seoul National University, Gwanak-Gu, Seoul, 08826, Republic of Korea

MONTI

- Multi-Omics linear integration method based on tensor decomposition & Classification analysis with the resulting multi-omics features
- GOAL: Cancer subtype classification
- Input
  - mRNA Gene expression
  - Methylation
  - miRNA expression
- Method Overview
  - Integrate multi-omics data and decompose using non-negative tensor decomposition
  - Select subtype-specific features and genes using L1 regularization
  - Use the selected features to generate multi-layer perceptron for cancer subtype classification

MONTI

- MONTI workflow



Build two-dimensional matrix of genes and samples

The omics matrices are stacked a form a three-dimensional tensor structure (genes, samples, omics)

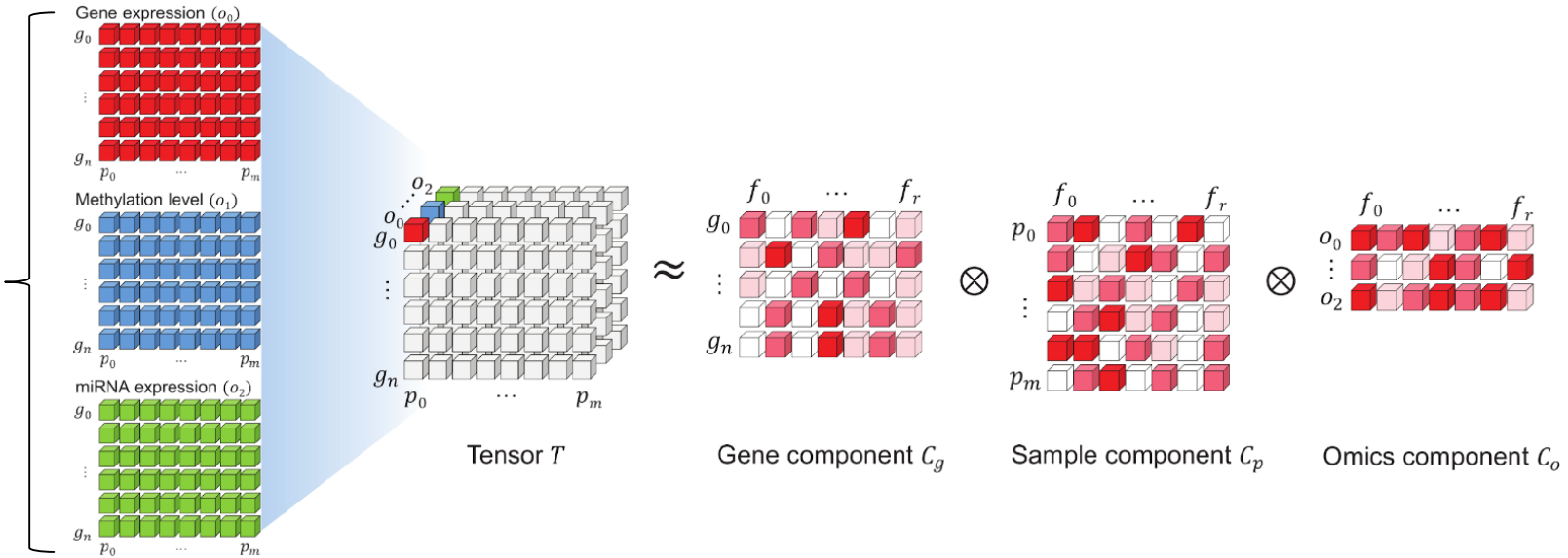
The tensor is decomposed into two-dimensional gene, patient, and omics components

Patient component: used for selecting subtype-specific features with subtype-specific L1 classifiers.
Selected features are used to build a subtype classifier model using MLP
Genes related to subtype-specific features are analyzed

MONTI

- Multi-omics Non-negative Tensor Decomposition

Gene-centric 2D omics matrices



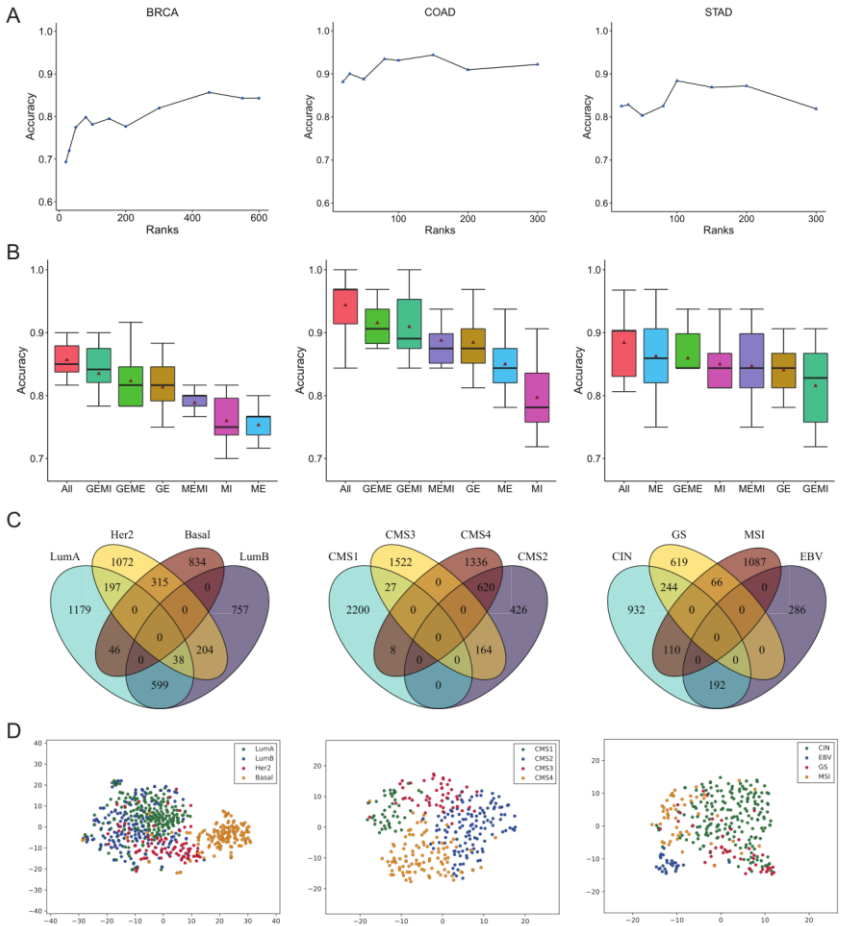
Tensor T is decomposed into three components: gene, sample (patient) and omics

Each component corresponds to one axis in tensor T

Each component in a two-dimensional matrix where one axis embeds the rank features f_i for the entities in each component (i.e., gene, samples and omics)

MONTI

- Result
  - A comparative figure for the three case studies
    - BRCA, COAD and STAD
  - COAD showed the highest subtype classification accuracy
  - The composition of omics data showed different impacts on subtype classification accuracy
    - For STAD, ME seemed to be most informative, where it reached 85% using only methylation data
    - However, overall, the accuracy was the highest when using all omics data



Clinical Cancer Research 2017

Autoencoder (non-linear combination of features)

Deep Learning-Based Multi-Omics Integration Robustly Predicts Survival in Liver Cancer

Kumardeep Chaudhary, Oliver B. Poirion, Liangqun Lu, and Lana X. Garmire

[1]Epidemiology Program, University of Hawaii Cancer Center, Honolulu, Hawaii.
[2]Molecular Biosciences and Bioengineering Graduate Program, University of Hawaii at Manoa, Honolulu, Hawaii.

Deep Learning-Based Multi-Omics Integration Robustly Predicts Survival in Liver Cancer (Autoencoder)

- Multi-omics integration based on Deep learning to differentiates survival subpopulations of HCC patients
- GOAL: Cancer(HCC) gene prediction & subtype identification
- Input: *TCGA HCC omics datasets of 360 samples*
  - mRNA expression
  - miRNA expression
  - Methylation
  - *Clinical information*
- Method Overview
  - Integrate multi-omics data and extract features using autoencoder
  - Select features using univariate Cox-PH models then label features using K-means clustering
  - Rank features of the training dataset using ANOVA and select the top features of the common type with the omic dataset of interest
  - Using the resulting filtered features to predict survival-risk subgroups using SVM

Deep Learning-Based Multi-Omics Integration Robustly Predicts Survival in Liver Cancer (Autoencoder)

- Method Workflow

  - Phase1

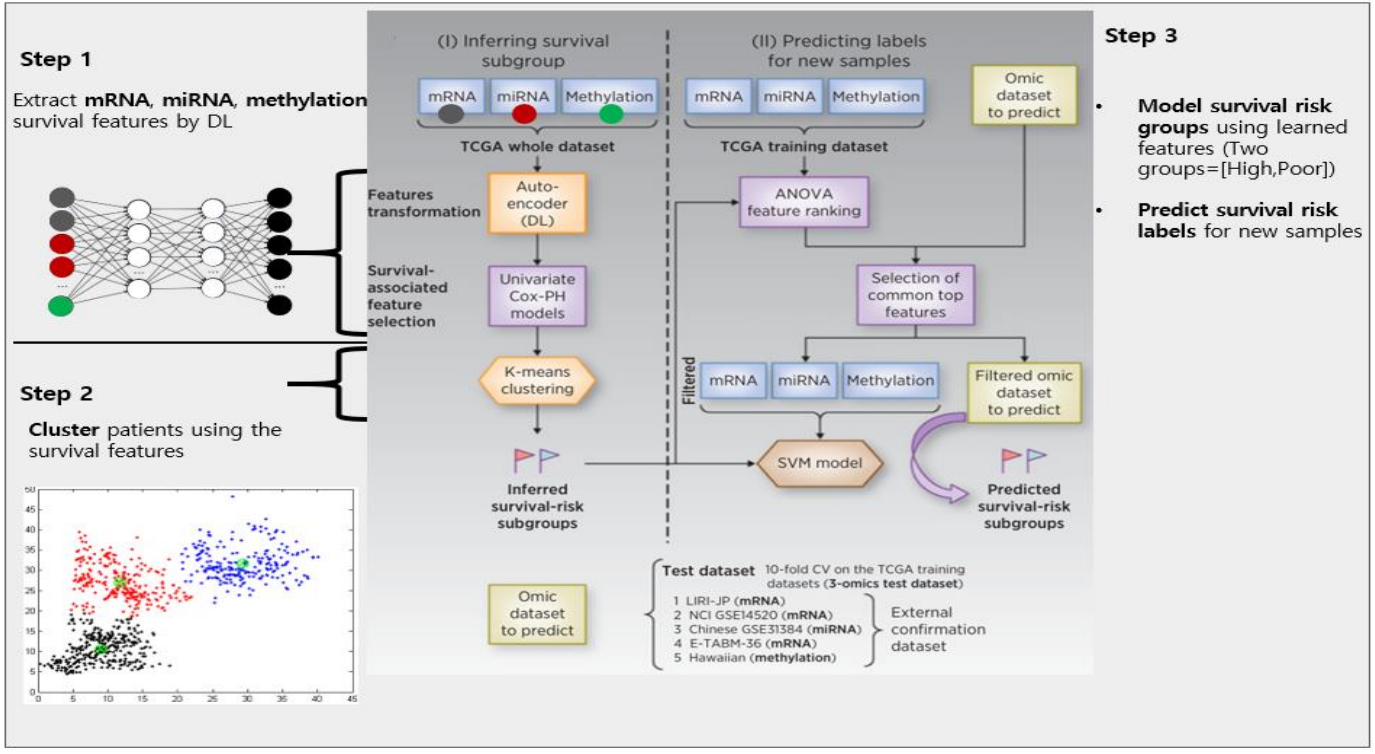
  - Integrate omics data to extract features using Autoencoder(DL)
  - Select features that are highly related to survival by building a univariate Cox-PH model for each feature
  - Perform K-means clustering to label the samples with one of the two flags (assign into survival-risk subgroups)

  - Phase2

  - Take the training dataset and rank the features with ANOVA testing results
  - Filter the features that are highly ranked & of the common feature type with the Omic dataset of interest
  - Using SVM, classify the omic dataset of interest into the subgroups
  - Examine the related genes of the subgroups

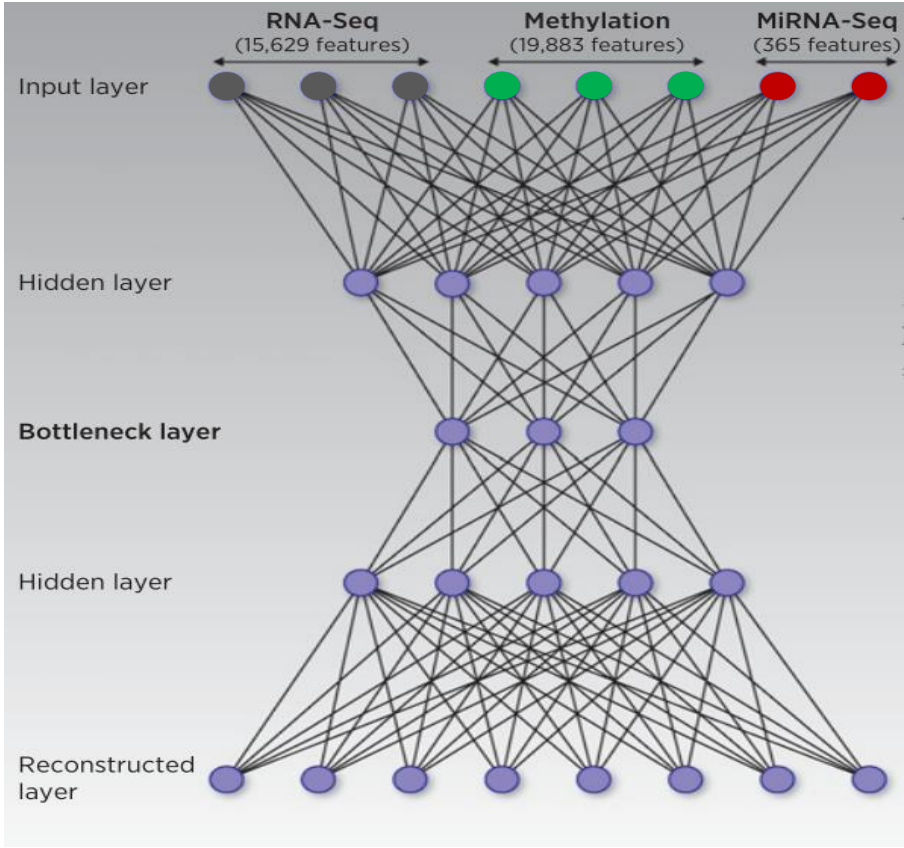
Deep Learning-Based Multi-Omics Integration Robustly Predicts Survival in Liver Cancer (Autoencoder)

- Method (Overall Workflow)



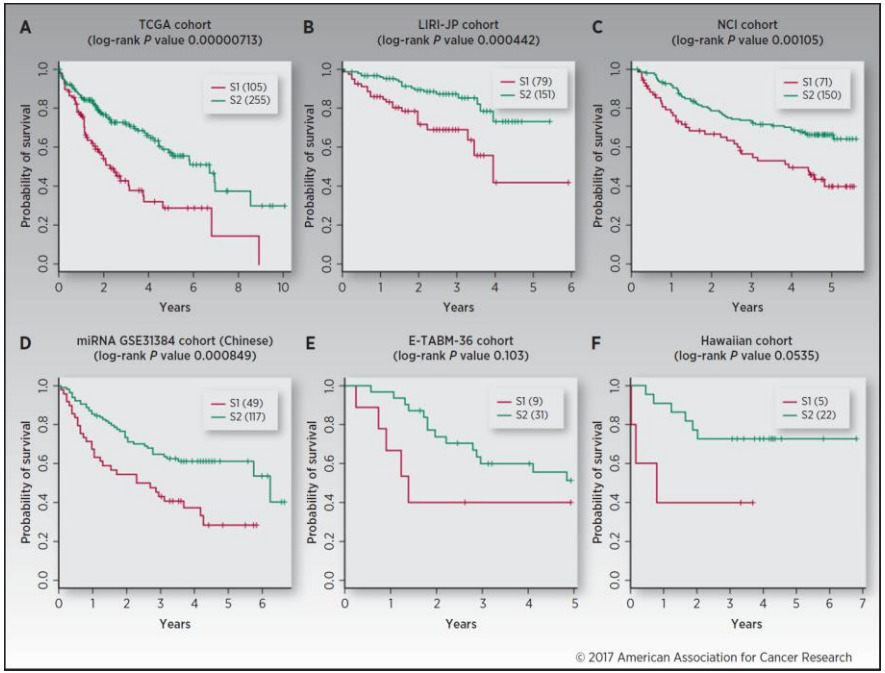
Deep Learning-Based Multi-Omics Integration Robustly Predicts Survival in Liver Cancer (Autoencoder)

- Input to the Autoencoder
  - TCGA HCC omics datasets of 360 samples
  - Three matrices that are unit-norm scaled by sample are stacked to form a unique matrix
- Activation Function: *tanh*
- *S*tructure: 3 layers (500, 100, and 500 nodes respectively)
- Objective function
  - TCGA HCC omics datasets of 360 samples
  - minimize the error between input and output
  - Cost function: *logloss*
  - Regularization:
  - L1 penalty for weights
  - L2 penalty for node activities
- Use the bottleneck layer to produce new features!

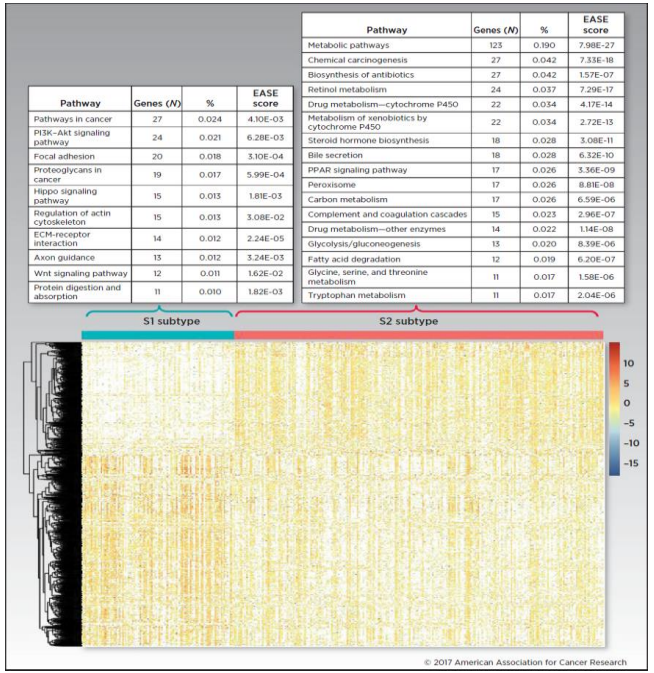


Deep Learning-Based Multi-Omics Integration Robustly Predicts Survival in Liver Cancer (Autoencoder)

- The survival plots of the six cohorts
  - There seem to be significant survival differences for the TCGA and external confirmation cohorts



- Differentially expressed genes and their enriched pathways in the two subtypes from the TCGA cohort
  - S1: aggressive (higher risk survival) / S2: moderate (lower risk survival)

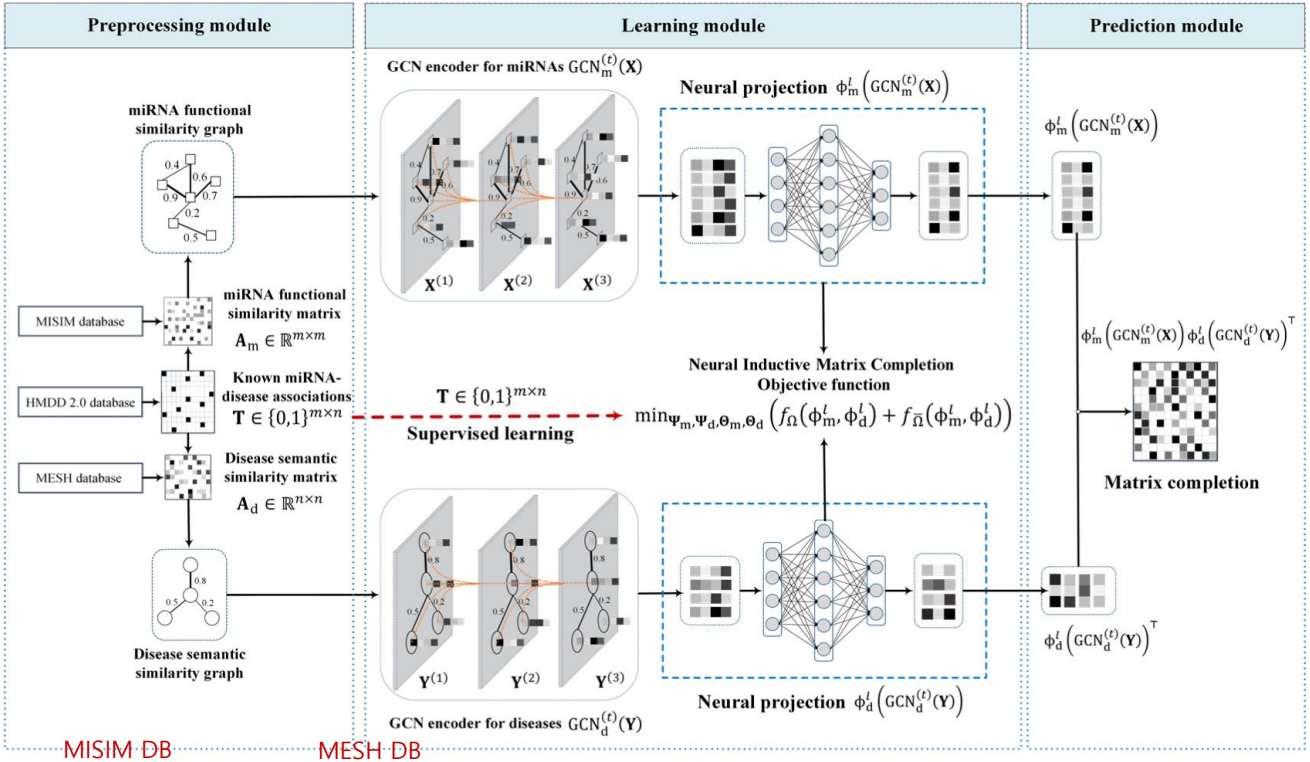


Pathway	Genes (N)	%	EASE score
Pathways in cancer	27	0.024	4.10E-03
PI3K–Akt signaling pathway	24	0.021	6.28E-03
Focal adhesion	20	0.018	3.10E-04
Proteoglycans in cancer	19	0.017	5.99E-04
Hippo signaling pathway	15	0.013	1.81E-03
Regulation of actin cytoskeleton	15	0.013	3.08E-02
ECM-receptor interaction	14	0.012	2.24E-05
Axon guidance	13	0.012	3.24E-03
Wnt signaling pathway	12	0.011	1.62E-02
Protein digestion and absorption	11	0.010	1.82E-03

Pathway	Genes (N)	%	EASE score
Metabolic pathways	123	0.190	7.98E-27
Chemical carcinogenesis	27	0.042	7.33E-18
Biosynthesis of antibiotics	27	0.042	1.57E-07
Retinol metabolism	24	0.037	7.29E-17
Drug metabolism—cytochrome P450	22	0.034	4.17E-14
Metabolism of xenobiotics by cytochrome P450	22	0.034	2.72E-13
Steroid hormone biosynthesis	18	0.028	3.08E-11
Bile secretion	18	0.028	6.32E-10
PPAR signaling pathway	17	0.026	3.36E-09
Peroxisome	17	0.026	8.81E-08
Carbon metabolism	17	0.026	6.59E-06
Complement and coagulation cascades	15	0.023	2.96E-07
Drug metabolism—other enzymes	14	0.022	1.14E-08
Glycolysis/gluconeogenesis	13	0.020	8.39E-06
Fatty acid degradation	12	0.019	6.20E-07
Glycine, serine, and threonine metabolism	11	0.017	1.58E-06
Tryptophan metabolism	11	0.017	2.04E-06

NIMCGCN

Li et al., Neural Inductive Matrix Completion with Graph Convolution Networks for miRNA-disease Association Prediction, ***Bioinformatics***, 2020



MISIM DB MESH DB

miRNA similarity + Disease similarity + Known (yet incomplete) miRNA-disease association

HMDD 2.0 DB

by GCN (similarity graph → matrix) and Neural projection,

→ Predict unknown miRNA-disease association (cf. Netflix Prize)

Clustering methods for multi-omics integration

Clustering methods intro

- Cancer subtype
  - The smaller groups that a type of cancer can be divided into, based on certain characteristics of the cancer cells.
    - How the cancer cells look under a microscope
    - Whether there are certain substances in or on the cell or certain changes to the DNA of the cells
- A clinically important challenge is to discover cancer subtypes and their molecular drivers in comprehensive genetic context.
- Advances in high-throughput technologies allow for measurements of many types of omics data, yet the meaningful integration of several different data types remains a significant challenge.

BIOINFORMATICS 2009

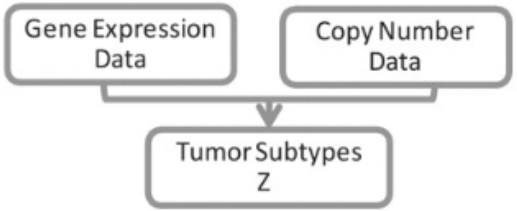
Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer

Ronglai Shen, Adam B. Olshen and Mara Ladanyi

[1]Department of Epidemiology and Biostatistics, Memorial Sloan-Kettering Cancer Center, New York, NY, [2]Department of Epidemiology and Biostatistics and Helen Diller Family Comprehensive Cancer Center, University of California, San Francisco, CA and [3]Department of Pathology and Human Oncology and Pathogenesis Program, Memorial Sloan-Kettering Cancer Center, New York, NY, USA

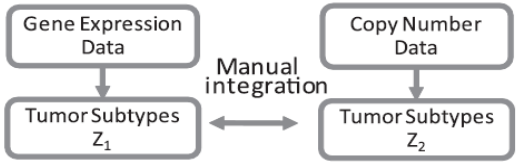
iCluster (Integrative Clustering)

- Joint latent variable model for integrative clustering.
- GOAL: Cancer subtype discovery.
- Input
  - Copy number and gene expression data of breast and lung cancer
- Method overview
  - Formulate the K-means problems as a Gaussian latent variable model and show the maximum likelihood-based solution and its connection with PCA solution.
  - Extend the latent variable model to allow multiple data types for the purpose of integrative clustering.



iCluster (Integrative Clustering)

- Multiple genomic platform (MGP) data
  - Any genomic dataset involving more than one data type measured in the same set of tumors.
  - Current approach to subtype discovery across multiple types is to separately cluster each type and then to manually integrate the results.



- Major challenges
  - To capture both concordant and unique alterations across data types, separate modeling of the covariance between data types and the variance-covariance structure within data types is needed.
  - Dimension reduction is key to the feasibility and performance of integrative clustering approaches.

iCluster (Integrative Clustering)

- iCluster method
  - Uses a Gaussian latent variable model to jointly model continuous genomic data such as gene expression, DNA methylation and copy number data.

  - Assumes that each data type is conditionally independent given the latent variables. For each data type $X_1, \cdots, X_m$, the mathematical form of the integrative model is

$$\begin{cases} X_1 = W_1 Z + \varepsilon_1 \\ \vdots \\ X_m = W_m Z + \varepsilon_m \end{cases}$$

where Z is the latent component that connects the m-set of models, ε_i is independent error term, in which each has mean zero and diagonal covariance matrix Ψ_i and W_i denote the coefficient matrix.

iCluster (Integrative Clustering)

- iCluster method
  - Uses a Gaussian latent variable model to jointly model continuous genomic data such as gene expression, DNA methylation and copy number data.

  - Optimal number of integrative clusters can be found when the joint likelihood of the genomic data sets is maximized.
    The corresponding log likelihood function of the data is

$$l(W,Z) = -\frac{n}{2}\left(\sum_{i=1}^{m} p_i \ln(2\pi) + \ln(\det(\Sigma)) + tr(\Sigma^{-1}G)\right),$$

where $W = (W_1, \cdots, W_m)'$, $\Sigma = WW' + \Psi$, $\Psi = diag(\Psi_1, \cdots, \Psi_m)$, $\sum_{i=1}^{m} p_i = m$ and G is the sample covariance matrix.

iCluster (Integrative Clustering)

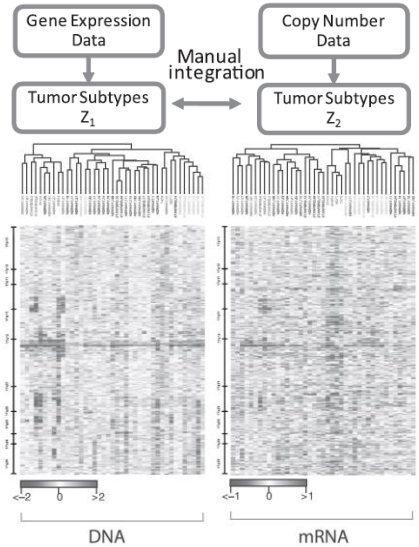
- iCluster method
  - Uses a Gaussian latent variable model to jointly model continuous genomic data such as gene expression, DNA methylation and copy number data.

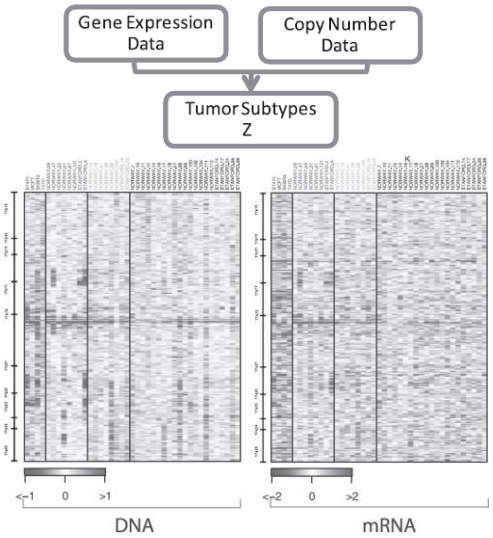
- Expectation-maximization (EM) algorithm is used for parameters W and Ψ estimation.
  Here, EM algorithm is
  E-step: expectation of hidden values
  M-step: max likelihood estimation of model parameters.
  In the EM framework, dell with the complete-data log likelihood

$$l_c(W,\Psi) = -\frac{n}{2}\left\{\sum_{i=1}^{m} p_i \ln(2\pi) + \ln(\det(\Psi))\right\} - \frac{1}{2}\{tr\big((X-WZ)'\Psi^{-1}(X-WZ)\big) + tr(Z'Z)\}.$$

iCluster (Integrative Clustering)

- Result
  - Identified subtypes characterized by concordant copy number changes or gene expression
  - and unique profiles specific to one or the other completely automated fashion.
  - Discovers potentially novel subtypes by combining weak yet consistent alteration patterns across data types.





PNAS 2013

Pattern discovery and cancer gene identification in integrated cancer genomic data

Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, Adam B. Olshen, Nikolaus Schultz, Chris Sander, R. Scott Powers, Marc Landanyi and Ronglai Shen

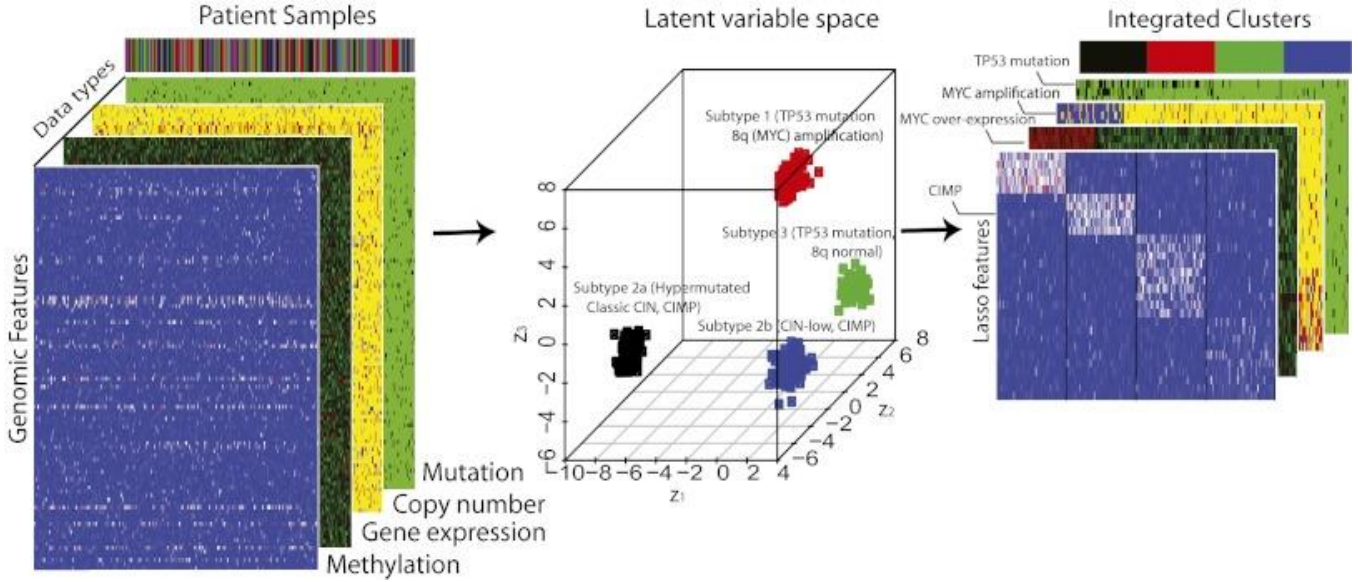
[a]Department of Epidemiology and Biostatistics, [e]Computational Biology Program, and [g]Department of Pathology and Human Oncology and Pathogenesis Program, Memorial Sloan–Kettering Cancer Center, New York, NY 10065; [b]Department of Medicine and Dan L. Duncan Cancer Center, Baylor College of Medicine, Houston, TX 77030; [c]Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI 53792; [d]Department of Epidemiology and Biostatistics, University of California, San Francisco, CA 94107; and [f]Cancer Genome Center, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11797

iCluster+ (Integrative Clustering+)

- Joint modeling of discrete and continuous variables that arise from multi-omics profiling.
- GOAL: Cancer subtype discovery and cancer gene identification.
- Input
  - CCLE human cell line dataset: chromosomal copy number, gene expression, mutation data
  - TCGA colorectal carcinoma dataset: exome sequence, DNA copy number, promoter methylation data
- Method overview
  - Extended version of iCluster.
  - Hypothesis-driven model-based approach for integrative clustering.
  - Generalized linear regression of the diverse types of genomic variables with respect to a common set of latent variables representing distinct set of molecular drives.
  - To identify genomic features, apply a penalized likelihood approach with lasso penalty terms.

iCluster+ (Integrative Clustering+)

- iCluster+ method
  - Use a set of latent variables to represent distinct driving factors (molecular drivers).
  - Geometrically, latent variables form a set of principal coordinates that span a lower dimensional integrated subspace, and collectively capture the major biological variations observed across cancer genomes.



iCluster+ (Integrative Clustering+)

- iCluster+ method
  - Pattern discovery that integrates diverse data types:

    For genomic variables x_{ijt} with the jth genomic feature in the ith sample of the tth data type,

    - binary variable (e.g., somatic mutation) – logistic regression

$$log\frac{P(x_{ijt}=1|z_i)}{1-P(x_{ijt}=1|z_i)}=\alpha_{jt}+\beta_{jt}z_i$$

    - categorical variable (e.g., copy number states) – multi logit regression

$$P(x_{ijt}=c|z_i)=\frac{\exp(\alpha_{jct}+\beta_{jct}z_i)}{\sum_{l=1}^{C}\exp(\alpha_{jlt}+\beta_{jlt}z_i)},\qquad c=1,\cdots,C$$

    - continuous variable (gene expression) – normal distribution and standard linear regression

$$x_{ijt}=\alpha_{jt}+\beta_{jt}z_i+\varepsilon_{ijt},\ \varepsilon_{ijt}\sim N(o,\sigma_{jt}^2)$$

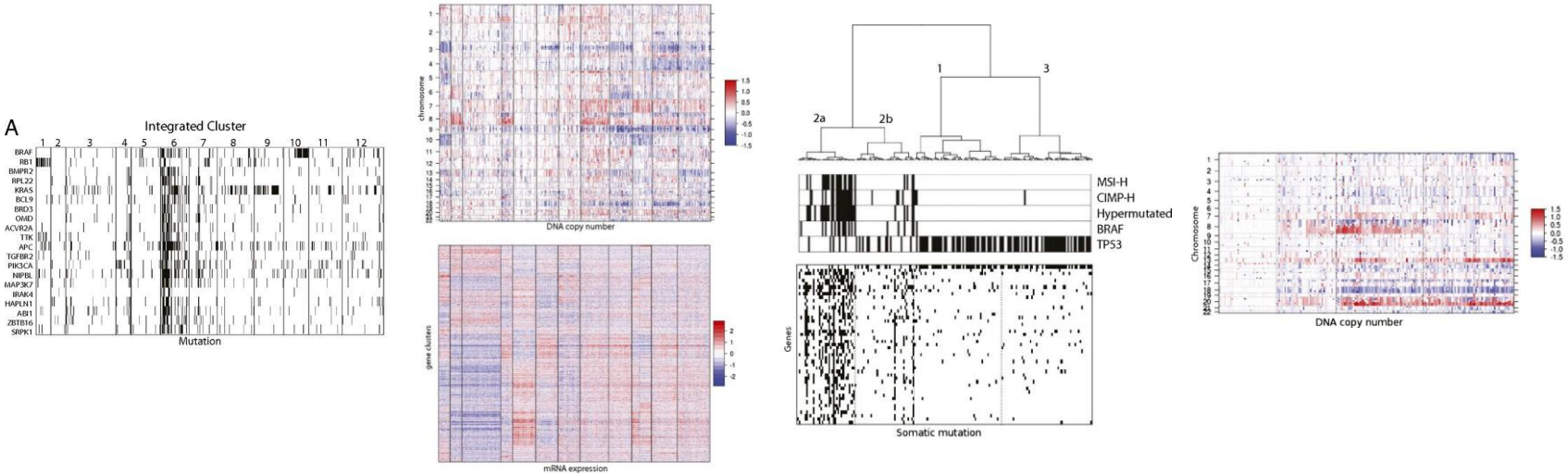
    - count variable (sequencing data) – Poisson regression

$$\log(\lambda(x_{ijt}|z_i)=\alpha_{jt}+\beta_{jt}z_i$$

    where z is latent variable, α is an intercept term and β is a length of coefficient.

iCluster+ (Integrative Clustering+)

- Result
  - Highly effective statistical framework to extract novel biological information from integrated cancer genomic data for tumor classification and cancer gene identification.
  - Revealed subgroups that are not lineage-dependent, but consist of different cancer types driven by a common genetic alteration.



Genome Research 2017

A novel approach for data integration and disease subtyping

Tin Nguyen, Rebecca Tagett, Diana Diaz and Sorin Draghici

[1]*Department of Computer Science and Engineering, University of Nevada, Reno, Nevada 89557, USA;* [2]*Department of Computer Science, Wayne State University, Detroit, Michigan 48202, USA;* [3]*Department of Obstetrics and Gynecology, Wayne State University, Detroit, Michigan 48201, USA*

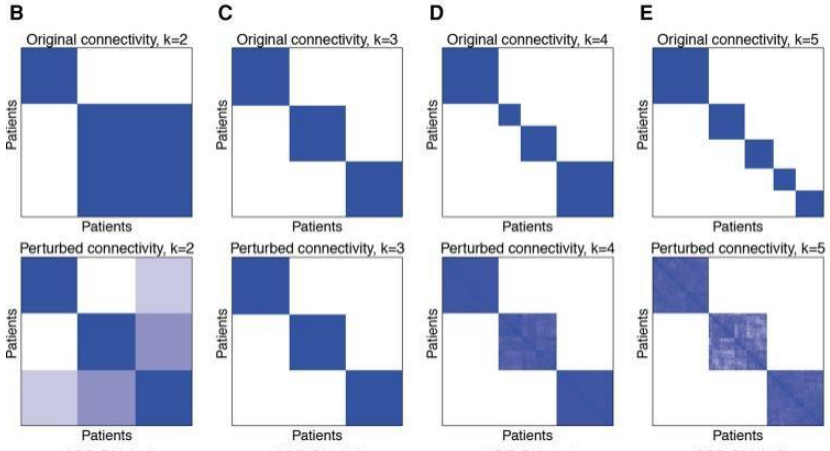
PINS (Perturbation clustering for data INtegration and disease Subtyping)

- Perturbation and repeat clustering to discover pattern of patients.
- GOAL: Discover the molecular subtypes of disease and subgroup of patients.
- Input
  - 12 tissues types and over 1000 samples
    - 8 gene expression data
    - 6 different cancer data (KIRC, GBM, LAML, LUSC, BRCA, COAD): mRNA, miRNA, methylation data
    - 2 breast cancer data (METABRIC): mRNA, CNV data
- Method overview
  - From the difference in patient-patient pairwise connectivity matrix of original data and perturbation data, optimal clustering number are decided.
  - Unbiased and unsupervised simultaneous subtyping approach.

PINS (Perturbation clustering for data INtegration and disease Subtyping)

- PINS method - Single data type
  - Perturbation clustering

1. The data are first partitioned with different values of number of clusters k
2. For each value of k, construct the pair-wise connectivity matrix
3. Add noise to the data and then build the pair-wise connectivity for the perturbed data
4. Calculate the discrepancy in pair-wise connectivity between before and after data perturbation.
5. Choose k as the optimal number of clusters



Original Connectivity matrix

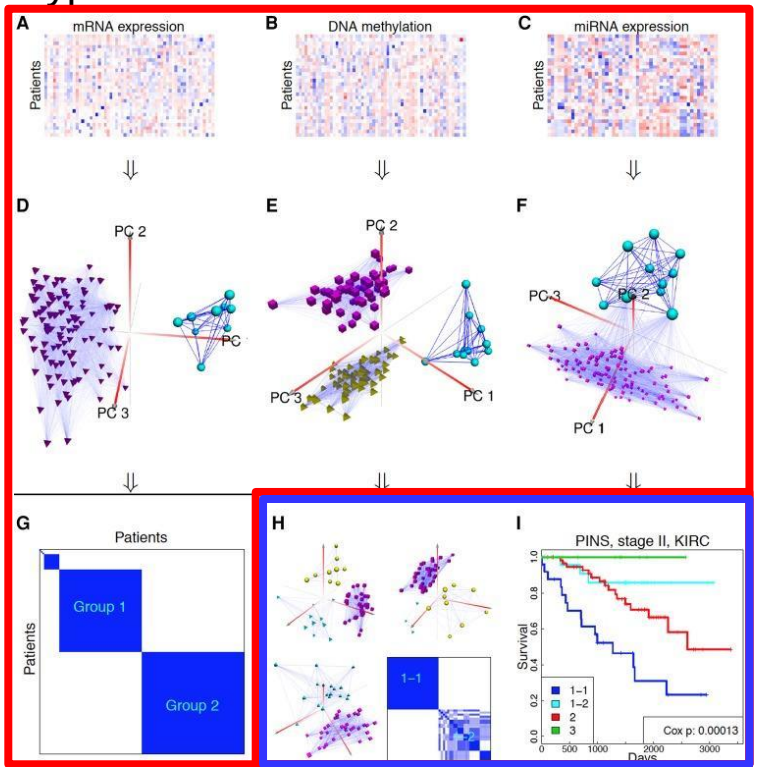
Perturbed connectivity matrix

PINS (Perturbation clustering for data INtegration and disease Subtyping)

- PINS method - Multiple data type

Connectivity between patients
for each data type

Similarity between patients
that is consistent across all data types



Stage I

Stage II

PINS (Perturbation clustering for data INtegration and disease Subtyping)

- PINS method - Multiple data type
  - Stage I – data integration and subtyping
  1. Build patient-patient connectivity matrices for each data type.
  2. Merge the connectivity matrices into a combined similarity matrix that represents the overall connectivity between patients.
  3. Input in similarity-based clustering algorithms: HC, PAM, Dynamic Tree Cut.
  - Stage II – Discover true partitions
  4. Check each discovered group independently to decide if it can be further divided.
  5. Only one group split into two subgroups of patients: strongly connected to each other’s across all the data types and loosely connected to each other’s.

PINS (Perturbation clustering for data INtegration and disease Subtyping)

- Result
  - Need significant long time
    1. Rely on data perturbation and repeated clustering to discover patterns of patients that are stable against small changes of molecular data.
    2. Run k-means multiple times to make sure that the results are stable and reproducible.
  - In unbiased and unsupervised manner, discover disease subtypes characterized by significant survival differences.
  - Identifying novel subtypes with significantly different survival profiles by integrating multiple types of data.

BIOINFORMATICS 2018

PINSPlus: a tool for tumor subtype discovery in integrated genomic data

Hung Nguyen, Sangam Shrestha, Sorin Draghici and Tin Nguyen

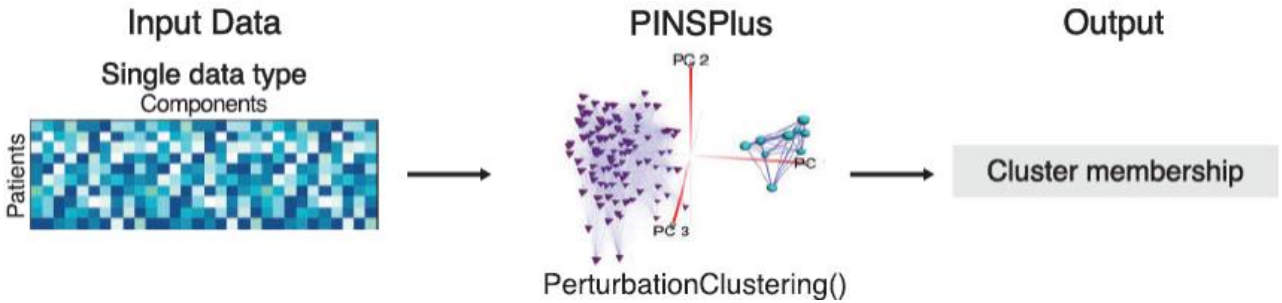
[1]Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA and [2]Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

PINSPlus (Perturbation clustering for data INtegration and disease Subtyping +)

- Unsupervised perturbation clustering for tumor subtype discovery.
- GOAL: Tumor subtype discovery
- Input
  - 12,158 samples form 44 datasets
    - 8 mRNA
    - 6 multi-omics dataset (KIRC, GBM, LAML, LUSC and two METABRIC)
    - 30 TCGA multi-omics dataset
- Method overview
  - Unsupervised approach for subtype discovery without using prior knowledge.
  - Optimization two algorithms of PINS: PerturbationClustering(), SubtypeingOmicsData()

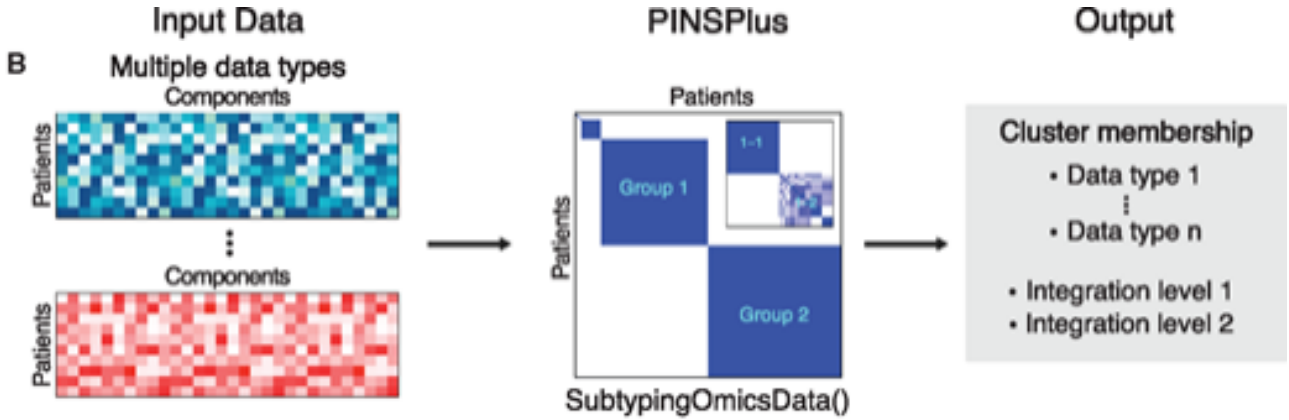
PINSPlus (Perturbation clustering for data INtegration and disease Subtyping +)

- PINSPlus method - Single data
  - PerturbationClustering() repeatedly perturbs the data by adding Gaussian noise and partitions the patients using different values for cluster number.
  - Output: the optimal number of subtypes



PINSPlus (Perturbation clustering for data INtegration and disease Subtyping +)

- PINSPlus method - Multi-omics data
  - SubtypingOmicsData() consists of multiple matrices for the same set of patients.
  - Outputs: subtyping results using
    1. each data type
    2. multi-omics data in stage I – Combine the connectivities to subtype the multi-omics data
    3. multi-omics data in stage II – Algorithm attempts to split each discovered



PINSPlus (Perturbation clustering for data INtegration and disease Subtyping +)

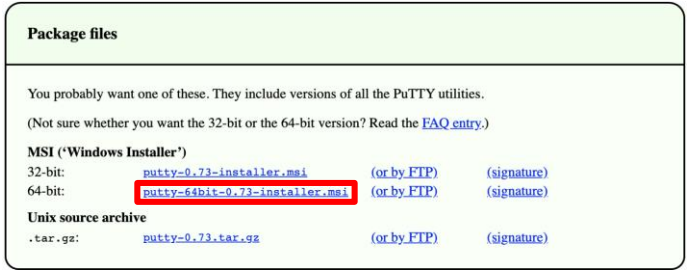
- Result
  - PINSPlus substantially outperforms the other methods (iCluster+, SNF) in identifying subtypes.
  - Robust against noise and unstable quantitative assays.
  - Able to integrate multiple types of omics data in a single analysis.
  - Superior to established approaches in identifying known subtypes and novel subgroups with significant survival difference.

Tutorial:
“AI and Network Bioinformatics for Multi-omics Data Analysis”

- Basic Setup -

Install PuTTy : Server connecting software

- [Download Link: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html)



Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

MSI ('Windows Installer')

32-bit: putty-0.73-installer.msi (or by FTP) (signature)

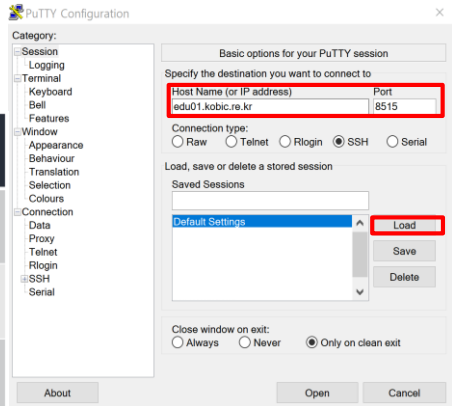
64-bit: putty-64bit-0.73-installer.msi (or by FTP) (signature)

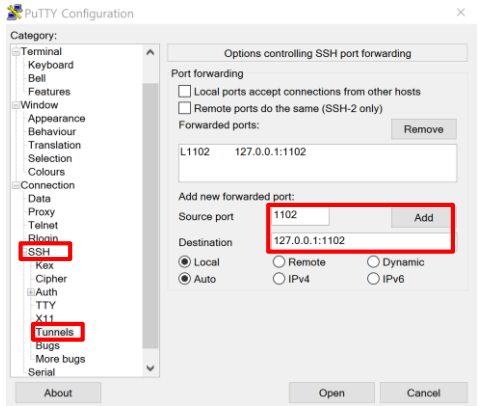
Unix source archive

.tar.gz: putty-0.73.tar.gz (or by FTP) (signature)

- Enter into to the Tutorial server
  - Host address SSH Tunnel 설정
  - 수강생 별 Host address / ID / Password:

Student #	Server	ID	Password	Tunnel: Source Port/Destination
수강생 **02 - 21**	**edu01.kobic.re.kr (210.218.222.141)**	kobic02 – kobic21	kobic8515	1102 - 1121
수강생 **22 - 41**	**edu02.kobic.re.kr (210.218.222.142)**	kobic22 – kobic41	kobic8515	1122 - 1141
수강생 **42 - 65**	**edu03.kobic.re.kr (210.218.222.143)**	kobic42 – kobic65	kobic8515	1142 - 1165

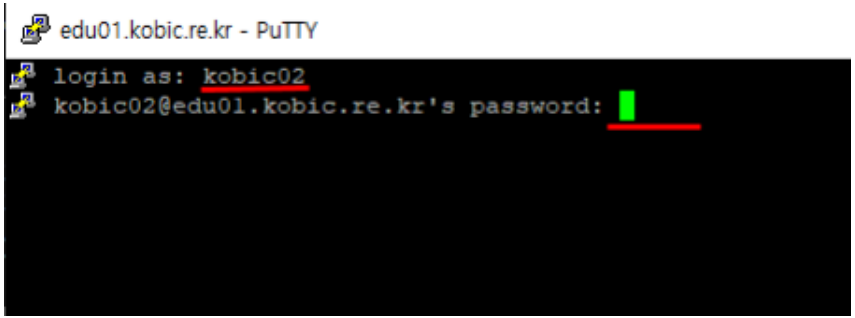


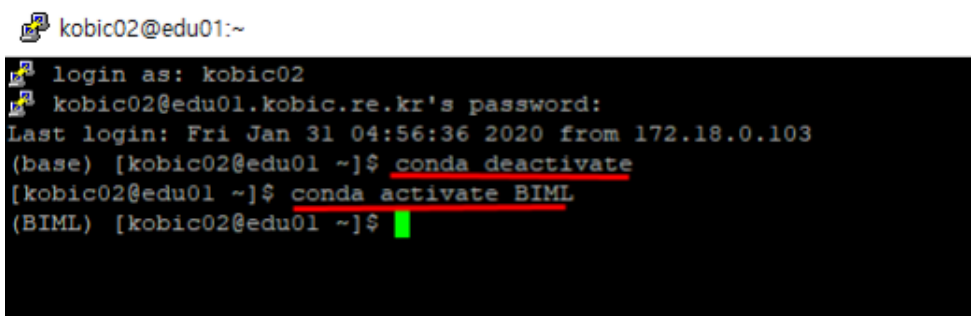


실습환경으로 진입하기

- ID 및 Password 입력
  - ID: kobic## (수강생 번호)
  - Password: kobic8515
- 콘다 환경 실행

```
$ conda deactivate
$ conda activate BIML
```





Shell command basics

- Terminal navigation commands

$ pwd	present working directory
$ cd <directory>	change directory
$ls	list directory contents

- File and directory manipulation commands

$ mkdir <directory name>	make directories
$ tar	archiving command: compress or decompress .tar, .gzip, .bzip files
$gunzip	archiving command: compress or decompress .gzip files

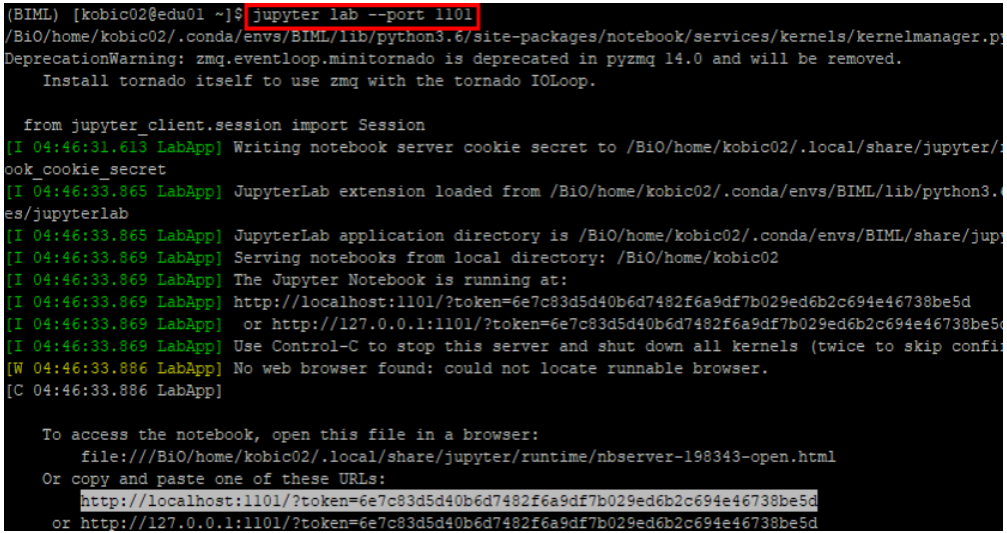
- Manual

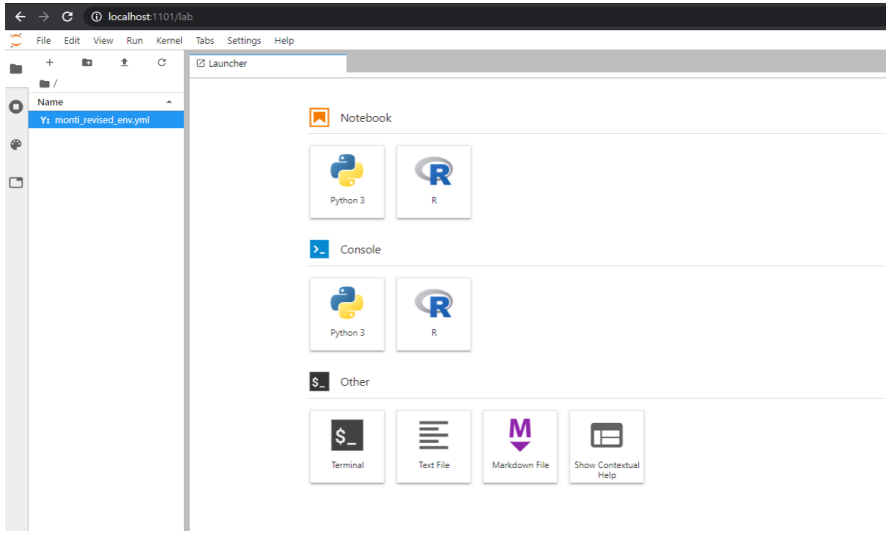
$ man <command>	Manual page of <command>

Opening Jupyter Lab

- 아래 커맨드 실행
  - ## 부분은 수강생 번호를 의미함

```
$ jupyter lab -- port 11##  (수강생 번호)
```





VIM Editor

- No mouse! Only Keyboard!
- Open file with VIM editor:
  - **vim** *filename*

- Save & exit
  - **:q!**
  - **:wq**

```
minwoo @ bhi4 in /data/project/minwoo/BIML2020/COAD_v2
$ vim data/omics_methylation_mat.txt
```

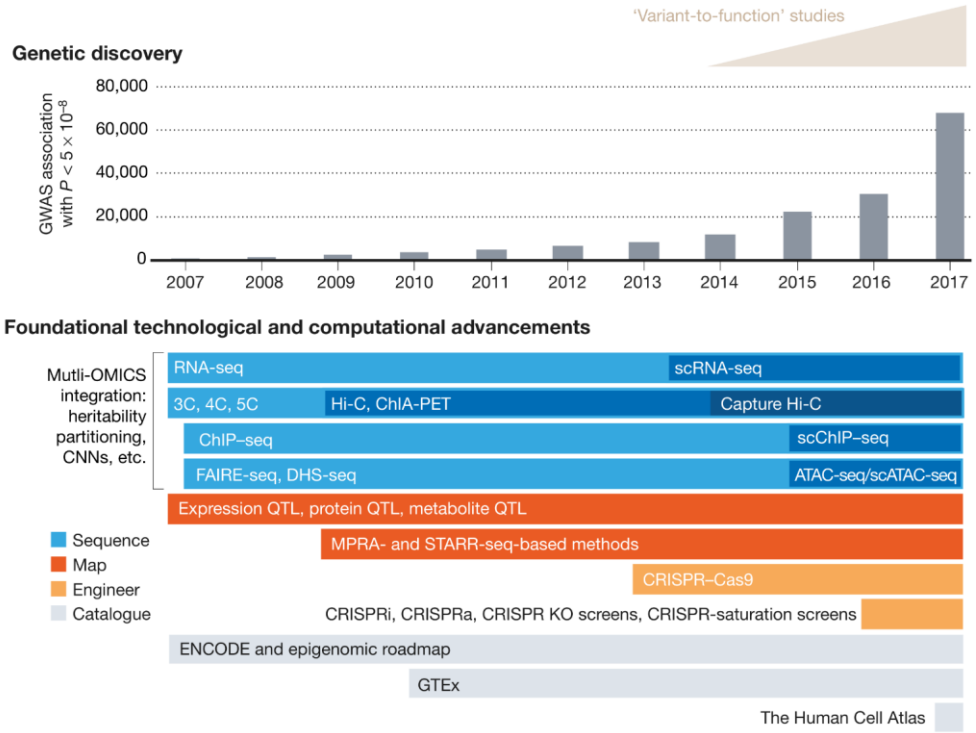
```
1    ID  TCGA-A6-2672    TCGA-A6-5661    TCGA-A6-6653
  1 cg00000292  0.772949247841925   0.865206465033348   0.81878927359949
  2 cg00000957  0.921157309523438   0.933994090206982   0.941089082368416
  3 cg00001245  0.0275906022131103  0.0274927583865968  0.0148942747113038
  4 cg00001261  0.692553681299151   0.789700270497979   0.878935525388389
  5 cg00001510  0.392425789944458   0.34799291602253    0.499358460706724
  6 cg00001534  NA  NA  NA
  7 cg00001687  0.979742132088306   0.973968584769948   0.981339221237125
  8 cg00002033  0.878659108123699   0.796603886153382   0.860344469970885
  9 cg00002591  0.922428174665378   0.927250726108254   0.921989140264203
 10 cg00002593  0.672540149829159   0.690035987480763   0.942199600603113
 11 cg00002837  0.175934520279686   0.228763982484938   0.207356250160339
 12 cg00003091  0.0404387101581865  0.0325548355765675  0.0510009940593592
 13 cg00003202  0.0183827997486839  0.0189851718954225  0.00932073687461978
 14 cg00003345  0.33019573292276    0.264737746925497   0.2200458276265
 15 cg00003513  0.467218138828584   0.826787590188332   0.35999266811104
 16 cg00003900  NA  NA  NA
 17 cg00004055  0.0585130016104208  0.0762562435002227  0.072866310293232
 18 cg00004072  0.0327983532255438  0.0362868656859329  0.048335996653125
 19 cg00004192  0.926723173757734   0.91091126619674    0.929252210253023
 20 cg00004209  0.724739242574372   0.812252603403995   0.867306860402636
 21 cg00004533  0.893076366488883   0.925576830279446   0.960293181832701
 22 cg00005010  0.0344873425456101  0.0338470781460772  0.0227844289467558
 23 cg00005297  0.550963378840642   0.434200295907531   0.467968669409093
 24 cg00005390  0.700712828681382   0.799719871803718   0.841946756513596
 25 cg00005541  NA  NA  NA
 26 cg00005617  0.568176827065152   0.76367069921445    0.484296872534668
 27 cg00005619  0.773108052003103   0.912635047121915   0.878231001085026
NORMAL  data/omics_methylation_mat.txt                    text  utf-8[unix]  248,280 words
:q!
```

Tutorial:
“AI and Network Bioinformatics for Multi-omics Data Analysis”

Contents

- Brief introduction to multi-omics analysis
- Downloading mo data from TCGA
- Preprocessing data
- Performing integrative analysis using
  - SNF
  - iCluster
  - VAE
  - MONTI

Genetic discovery is paralleled by advances in functional genomics technologies



Melina Claussnitzer et al., "A brief history of human disease genetics", Nature Review 2020

Technical issues in MO analysis

Technical topics	Issues
Type	All omics generated from different platforms
Normalization	Omics have different measurement scales
Relationships	Omics have different relationships with genes
Strength	Their impact to genes are different
Locus	Each target or observed at different locations
Focus	Gene, loci, promoter?
Trans-element	Too many combinations, most focus on cis-regulatory elements

Two primary approaches for integrating MO data

Multi-staged

- Focus: Identify genes driving cancer
- Data: Gene-centric
- Result: cis relationship (Genes x Omics)

Tools

- CNAmet
- iGC
- MethylMix
- MONTI

Meta-dimensional

- Focus: Clinically relevant tumor or sample classification
- Data: Sample-centric
- Result: Tumor sample (subtype) classification model

Tools

- SNF
- BCC
- iClusterPlus
- MixOmics

Workflow of MO analysis

1. Data collection

- Multi-omics resources
- Bulk data downloading
- Matched data

2. Preprocessing

- Data cleaning
- Normalization
- Omics/Sample selection

3. Data Integration

- Multi-staged data integration
- Meta-dimensional data fusion

4. MO analysis

- Feature selection
- Data fusion analysis

5. Downstream analysis

- Condition specific gene sets
- Correlation
- Gene set enrichment
- Pathway
- Survival

Step 1 | Data & Build work env.

- Downloading TCGA data
  - Colorectal cancer (COAD), n=315
- Data types: [Gene expression (GE), Methylation (ME), miRNA (MI)]

- Create a project folder with name “COAD”
- This will be the base directory for storing and pre-processing COAD related omics data
- Download package for download and preprocessing multi-omics data

```
$ mkdir COAD
$ cd COAD
$ cp </dir/of/mo_data_processing.tar.gz>  ./
$ tar -xzvf mo_data_processing.tar.gz
```

Step 1 | Dowloading TCGA manifest

COAD sample barcodes

```
TCGA-A6-2672
TCGA-A6-5661
TCGA-A6-6653
```

1. To access TCGA go to https://portal.gdc.cancer.gov/repository
2. Select the “Cases” tab
3. Let’s get information of three COAD samples
4. Copy and paste the barcodes below in the “Upload Case set” text box
5. Now, select the “Files” tab
6. For gene expression and miRNA expression data, check “transcriptome profiling” under “Data Category”
   - Check “Gene Expression Quantification” and “miRNA Expression Quantification” under ”Data Type”
   - Check “BCGSC miRNA Profiling” and “HTSeq-FPKM-UQ” under “Workflow Type”
   - Click on “Manifest”
   - This will download the manifest file (txt) of the samples
7. For methylation data, check “dna methylation” under “Data Category”
   - Check “illumina human methylation 450” under “Platform”
   - Click on “Manifest”
8. Move the two downloaded manifest files to the “annotation” directory of your working directory

NIH NATIONAL CANCER INSTITUTE GDC Data Portal

Home | Projects | Exploration | Analysis | Repository | Quick Search | Manage Sets | Login | Cart 0 | GDC Apps

File

e.g. 142682.bam, 4f6e2e7a-b...

Data Category

- [x] transcriptome profiling 10

Data Type

- [x] Gene Expression Quantification 5
- [ ] Isoform Expression Quantification 5
- [x] miRNA Expression Quantification 5

Experimental Strategy

- [ ] RNA-Seq 5
- [ ] miRNA-Seq 5

Workflow Type

- [x] BCGSC miRNA Profiling 5
- [ ] HTSeq - Counts 5
- [ ] HTSeq - FPKM 5
- [x] HTSeq - FPKM-UQ 5

Data Category IS transcriptome profiling AND

Data Type IN (Gene Expression Quantification | miRNA Expression Quantification)

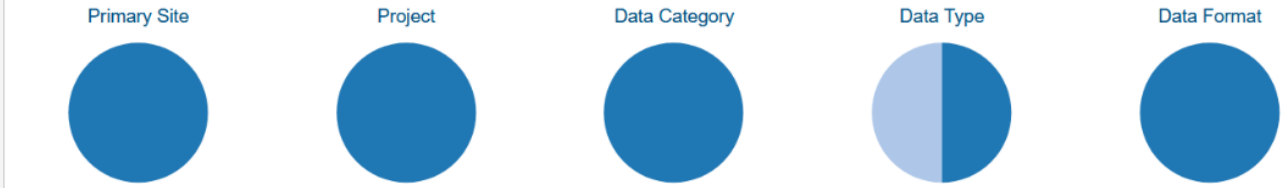
Advanced Search

Add All Files to Cart | Manifest | View 3 Cases in Exploration | View Images

Browse Annotations

Files (10) | Cases (3)

2.77 MB



Show More

Showing **1 - 10** of **10** files

JSON | TSV

Access	File Name	Cases	Project	Data Category	Data Format	File Size	Annotations
open	37d2cb9b-0286-43a5-9538-26a8f0818ee7.FPKM-UQ.txt.gz	1	TCGA-COAD	Transcriptome Profiling	TXT	475.54 KB	1
open	59e2ab4e-9529-490d-b12c-9b7f24c8166c.FPKM-UQ.txt.gz	1	TCGA-COAD	Transcriptome Profiling	TXT	563.35 KB	1
open	b520edf2-42ce-4b64-8575-19e34b30f042.FPKM-UQ.txt.gz	1	TCGA-COAD	Transcriptome Profiling	TXT	488.91 KB	0

Step 1 | Processing manifest file

- Compile payload and metadata files & download payload (omics data)

```
$ cd 0_preprocessing
$ python access_tcga.py COAD
```

```
                  $ curl -o ./tcga_files/gdc_download.tar.gz --remote-name --remote-header-name --request POST --hea
er 'Content-Type: application/json' --data @./annotation/COAD_download.txt 'https://api.gdc.cancer.gov/data'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  147M    0  147M  100   503  2409k      8  0:01:02  0:01:02 --:--:-- 2554k
```

- Go to 'tcga_files' and uncompress data file

```
$ cd tcga_files
$ tar -xzvf gdc_download.tar.gz
$ gunzip */*.gz
```

Step 1 | Group files per Barcode

- Since the files are downloaded per UUID, let’s group them by their barcode

```
$ python proc_barcodes.py COAD
```

- Under the 'cases’ directory, three omics files are grouped per barcode ID (or case ID)

```
drwxrwx---  4.0K  1월  13 13:39 TCGA-A6-6653-01A/
drwxrwx---  4.0K  1월  13 13:39 TCGA-A6-5661-01B/
drwxrwx---  4.0K  1월  13 13:39 TCGA-A6-2672-01B/
```

Step 2 | Preprocessing

- Now, let's compile a `(omics x sample)` matrix for each omics

```
$ python gen_matrix.py COAD
```

```
$ ./gen_matrix.py COAD
gene
100.0%  TCGA-NH-A50V
methylation
4.8%    TCGA-AZ-4615
$ ll data
total 4.2M
-rw-rw---- 1   76K  1월  13 14:15 omics_mirna_mat.txt
-rw-rw---- 1  3.5M  1월  13 14:15 omics_methylation_mat.txt
-rw-rw---- 1  696K  1월  13 14:15 omics_gene_mat.txt
```

omics 2D matrices

- Let's also create gene-centric matrices

```
$ python make_mir_gcentric.py COAD
$ python make_methylation_gcentric.py COAD
$ cd data
$ ln -s omics_gene_mat.txt COAD_gcentric_gene.txt
```

```
total 5.7M
lrwxrwxrwx 1   18  1월  13 14:44 COAD_gcentric_gene.txt -> omics_gene_mat.txt
-rw-rw---- 1 927K  1월  13 14:32 COAD_gcentric_methylation.txt
-rw-rw---- 1 570K  1월  13 14:30 COAD_gcentric_mirna.txt
```

Gene-centric omics 2D matrices

Step 3 | Data integration

- Merge the omics matrices to get a `(omics x gene x sample)` tensor data

```
$ cd ..
$ python merge_omics_files.py COAD
```

```
[                    ]$ ./merge_omics_files.py COAD
merging gene data...
merging methylation data...
merging mirna data...
merged tensor: (3, 14513, 3)
normalizing gene data...
normalizing methylation data...
normalizing mirna data...
```

- The final results can be found in the 'data' directory

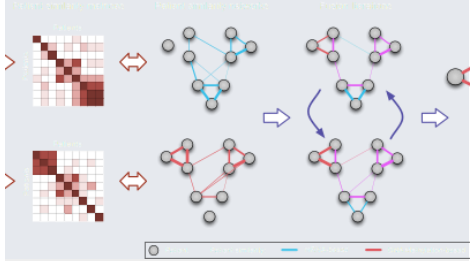
```
[                   ]$ ll data
total 8.7M
-rw-rw---- 1            1021K  1월  13 15:04 COAD_omics_tensor_norm.npy
-rw-rw---- 1            1021K  1월  13 15:04 COAD_omics_tensor_log2.npy
-rw-rw---- 1            1021K  1월  13 15:04 COAD_omics_tensor_raw.npy
```

3D tensor data

  - Here, we provide log2 quantile normalized (norm), log2 normalized (log2) and raw expression data (raw)

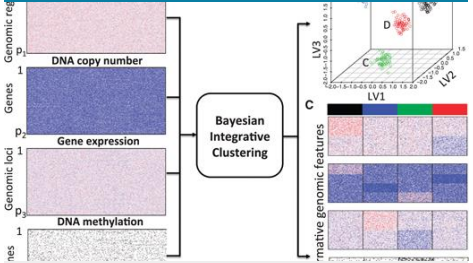
Practicing multi-omics analysis tools

SNF (Similarity Network Fusion)



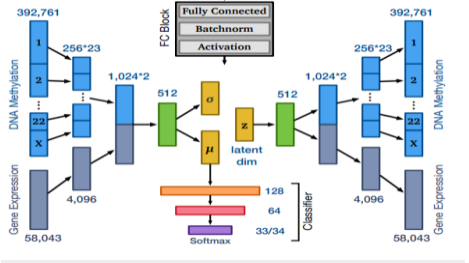
Wang, Bo, et al. "Similarity network fusion for aggregating data types on a genomic scale." Nature methods, 2014

iCluster



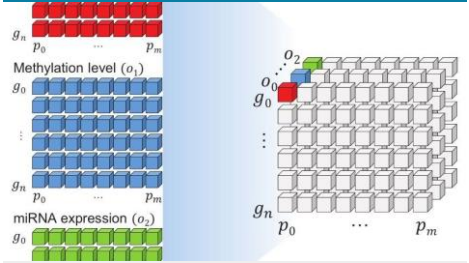
Shen et al., "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis." Bioinformatics, 2009

VAE (Variational Auto-Encoding)



Kingma, Diederik P., and Maz Welling. “Auto-encoding variational bayes.”, arXiv preprint arXiv: 1312.6144 (2013)

MONTI (Multi-omics Non-negative Tensor Integration)



Inuk Jung et al., “MONTI: A multi-omics non-negative tensor decomposition framework for the integrated analysis of cancer subtypes”, Under Review (2020)

Tool 1| SNF

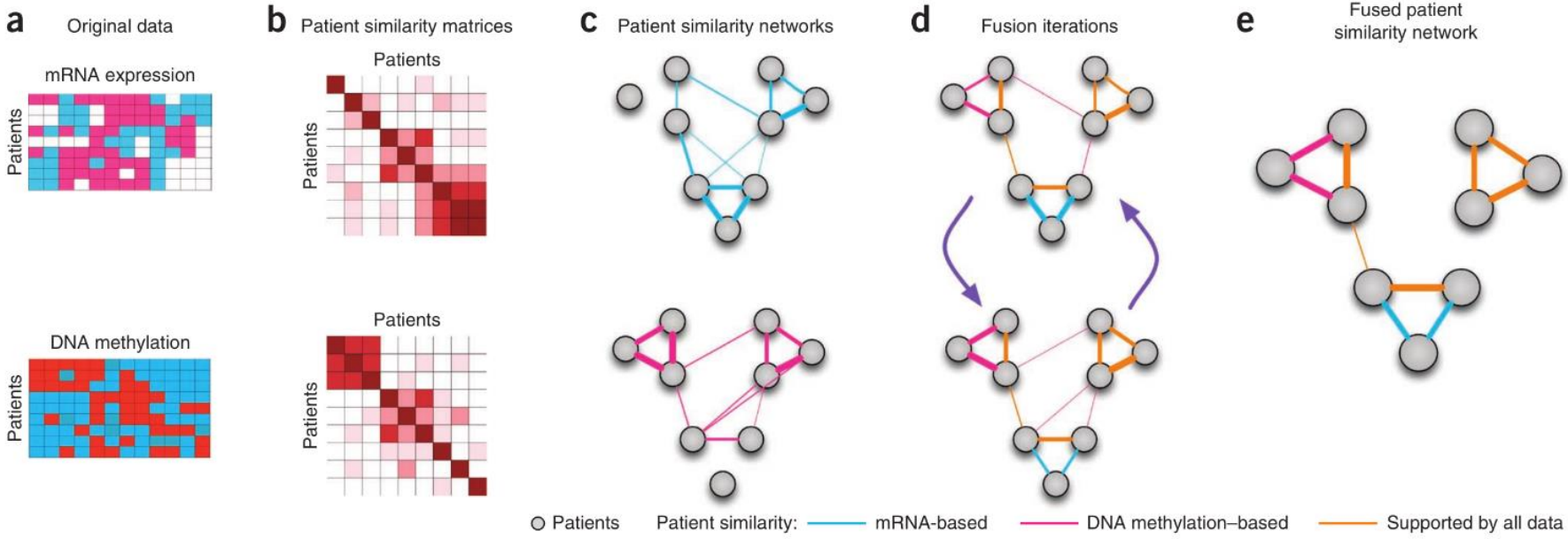
"Similarity network fusion for aggregating data types on a genomic scale."

Wang Bo et al., Nature methods 2014

SNF Workflow

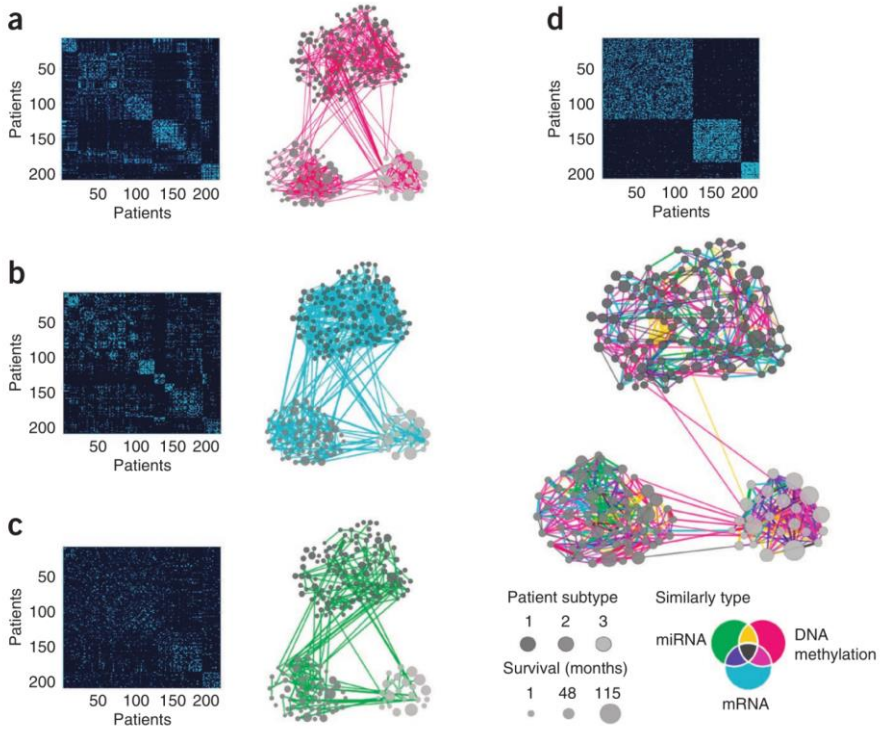
INPUT DATA

OUTPUT DATA



SNF results on GBM (Glioblastoma) data

- All omics data are matched at the patient level
- Omics data are not gene-centric
- Each omics data are independently processed into individual omics networks
  - Red: Methylation network
  - Blue: Gene expression network
  - Green: miRNA expression network
- Iteratively, the three networks are fused to finally acquire edges that are present in all three networks while removing those that are not



Installing SNFtool (R package)

- SNF is currently provided in R and Matlab
- In this tutorial we will use R
- Create a new directory for SNF
- Download the full COAD data (pre-processed)

```
$ cd 1_SNF
$ jupyer lab --port 11##
```

SNF | Input files

- Import the pre-processed multi-omics matrices

  omics_{omics type}_mat.txt : omics data

  omics_{omics type}_mat.txt : toy example of omics data (500 features)

  subtypes.txt : subtype of COAD patients

```
$ ll SNF_inp/
total 363204
drwxr-xr-x 2 dabin lab      4096 Jan 30 14:10 ./
drwxr-xr-x 4 dabin lab      4096 Jan 30 14:10 ../
-rw-r--r-- 1 dabin lab   1507106 Jan 30 14:07 omics_gene_mat_small.txt
-rw-r----- 1 dabin lab  51324792 Jan 30 14:07 omics_gene_mat.txt
-rw-r--r-- 1 dabin lab   2950015 Jan 30 14:07 omics_methylation_mat_small.txt
-rw-r----- 1 dabin lab 308821585 Jan 30 14:07 omics_methylation_mat.txt
-rw-r--r-- 1 dabin lab   1801537 Jan 30 14:07 omics_mirna_mat_small.txt
-rw-r----- 1 dabin lab   5491112 Jan 30 14:07 omics_mirna_mat.txt
-rw-r----- 1 dabin lab      5670 Jan 30 14:08 subtypes.txt
```

SNF | Import multi-omics data

Sample labels

Gene expression data

Methylation data

miRNA expression data

```
## Read patient subtype information
samplabs<-read.table("subtypes.txt", header=F)[,2]
samplabs<-as.numeric(factor(samplabs))

# read gene expression data
dat_gene<-read.table("COAD_data/omics_gene_mat.txt", header=T)
dat_gene<-dat_gene[,2:ncol(dat_gene)]       # skip first row (gene ids)
dat_gene<-t(dat_gene)

# read methylation data
dat_meth<-read.table("COAD_data/omics_methylation_mat.txt", header=T)
dat_meth<-dat_meth[,2:ncol(dat_meth)]     # skip first row (probe ids)
dat_meth<-na.omit(dat_meth)
dat_meth<-t(dat_meth)

# read miRNA data
dat_mir<-read.table("COAD_data/omics_mirna_mat.txt", header=T)
dat_mir<-dat_mir[,2:ncol(dat_mir)]              # skip first row (probe ids)
dat_mir<-dat_mir[rowSums(dat_mir==0.0)<(ncol(dat_mir)*.8),]              # skip rows with >80% zero values
dat_mir<-t(dat_mir)
```

SNF | Data Normalization and Merge

- If data need to be normalized use 'standardNormalization' function
  - log2 transform and normalizes data to have mu=0, std=1
  - methylation data is not log2 transformed since already scaled between 0~1

```
# normalize data
dat_gene_norm<-standardNormalization(log2(dat_gene+1));
dat_meth_norm<-standardNormalization(dat_meth);
dat_mir_norm<-standardNormalization(log2(dat_mir+1));

# merge data
dat_merged<-list(dat_gene_norm, dat_meth_norm, dat_mir_norm)
```

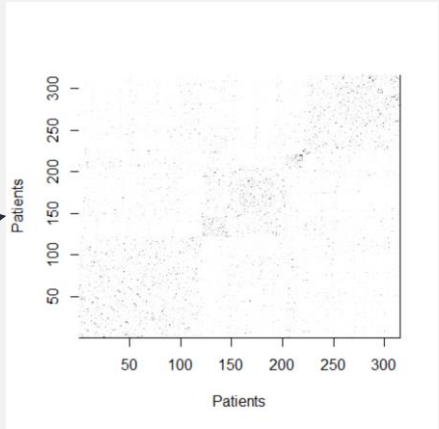
SNF | Compute patient-pairwise distance

- Compute sample pairwise distances and construct similarity graphs using them

```
## Hyperparameters
K = 20; # number of neighbors, usually (10~30)
alpha = 0.3; # hyperparameter, usually (0.3~0.8)
T = 20; # Number of Iterations, usually (10~20)

## Calculate the pair-wise distance (per omics)
dist_gene = (dist2(dat_gene_norm,dat_gene_norm))^(1/2)
dist_meth = (dist2(dat_meth_norm,dat_meth_norm))^(1/2)
dist_mir = (dist2(dat_mir_norm,dat_mir_norm))^(1/2)

## next, construct similarity graphs
W1 = affinityMatrix(dist_gene, K, alpha)
W2 = affinityMatrix(dist_meth, K, alpha)
W3 = affinityMatrix(dist_mir, K, alpha)

## The above steps can be shorted to below
dist_dat_merged = lapply(dat_merged, function(x) (dist2(x, x))^(1/2))
affinityL = lapply(dist_dat_merged, function(x) affinityMatrix(x, K, alpha))
```

SNF | Fusing Networks

```
## Fusing graphs W1, W2 and W3
W = SNF(list(W1, W2, W3), K, T)

## Performing spectral clustering on fused network
estimationResult=estimateNumberOfClustersGivenGraph(W, NUMC=2:10);
estimationResult
C = 5 # number of clusters
clusters = spectralClustering(W,C); # the final subtypes information

## Display clusters in fused network W
displayClusters(W, clusters)

## Computing the concordance matrix
ConcordanceMatrix = concordanceNetworkNMI(list(W, W1,W2,W3), C);
```

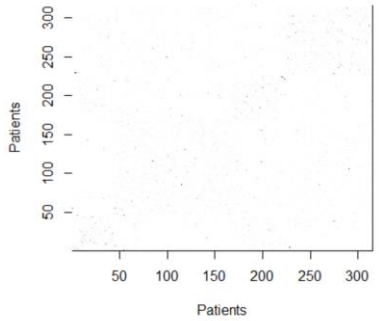



```
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.2768673 0.2149351 0.2597418
[2,] 0.2768673 1.0000000 0.4025351 0.5876234
[3,] 0.2149351 0.4025351 1.0000000 0.3726574
[4,] 0.2597418 0.5876234 0.3726574 1.0000000
```

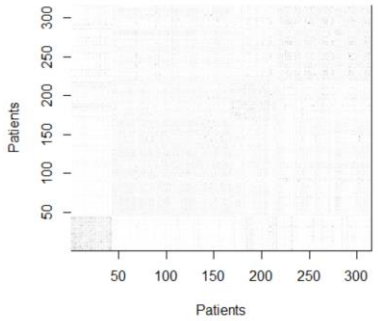
Concordance matrix

SNF | Visualizing similarity matrix of omics

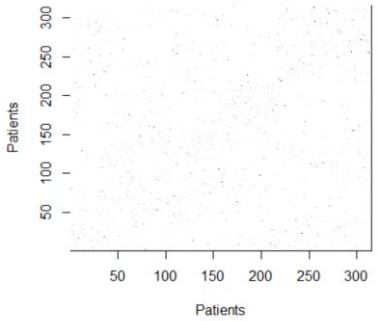
```
## These similarity graphs have complementary information about clusters.
displayClusters(W1,samplabs);
displayClusters(W2,samplabs);
displayClusters(W3,samplabs);

displayClusters(W,samplabs);
```



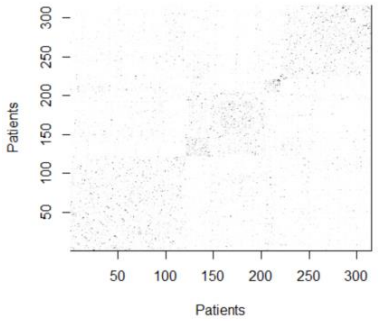
W1
(GE)



W2
(ME)



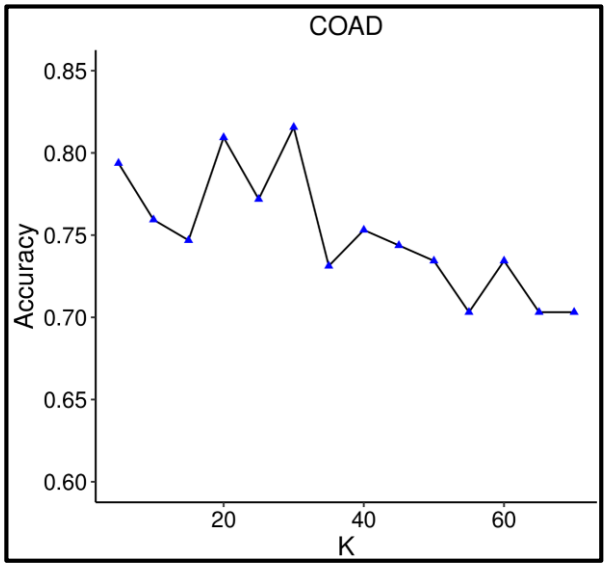
W3
(MI)



W
(GE, ME, MI)

SNF | Subtype Classification

```
# 10-cross validated subtype classification
for (K in seq(5, 50, by=5)) {
  acc_avg=numeric(10)
                    cv=1:10          # 10 cross validation
                    for(fold in cv) {
                                     # Create the training and test data
                                     n = floor(0.8*length(samplabs)) # number of training cases
                                     trainSample = sample.int(length(samplabs), n)
                                     train = lapply(dat_merged, function(x) x[trainSample, ]) # Use the
first 150 samples
                                     test = lapply(dat_merged, function(x) x[-trainSample, ]) # Test the
rest of the data
                                     clusters = samplabs[trainSample]

                                     # Apply the prediction function to the data
                                     newLabel = groupPredict(train,test,clusters,K,alpha,T)
                                     # The prediction accuracy
                                     accuracy = sum(samplabs[-trainSample] == newLabel[-
c(1:n)])/(length(samplabs) - n)
                                     cat(sprintf("[%d] K=%d, %f\n", fold, K, accuracy))
                                     acc_avg[fold]=accuracy
                    }
                    cat(sprintf("[%d] K=%d, Average acc: %f\n", fold, K, mean(acc_avg)))
}
```



Tool 2| iCluster

"Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis."
Shen et al., Bioinformatics 2009

iCluster workflow

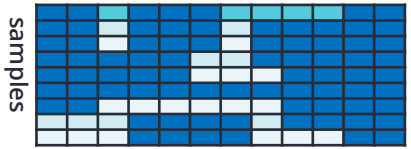
INPUT DATA

OUTPUT DATA

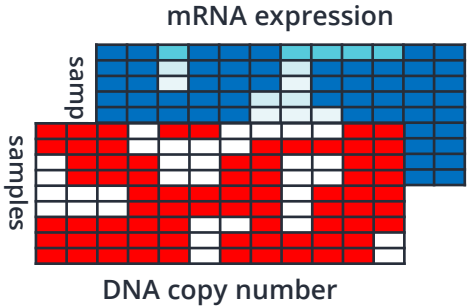
Omics data

DNA copy number

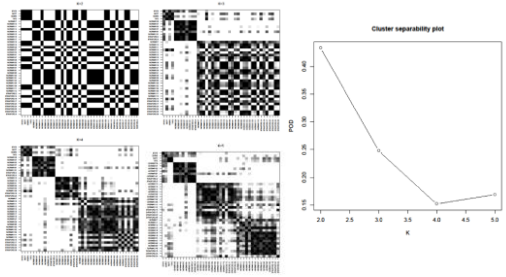
mRNA expression



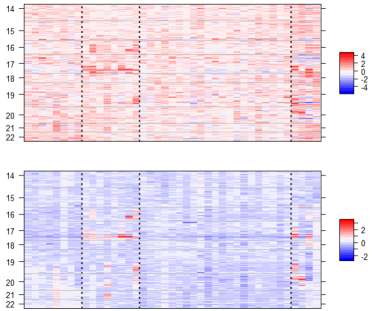
Merged data



Model selection

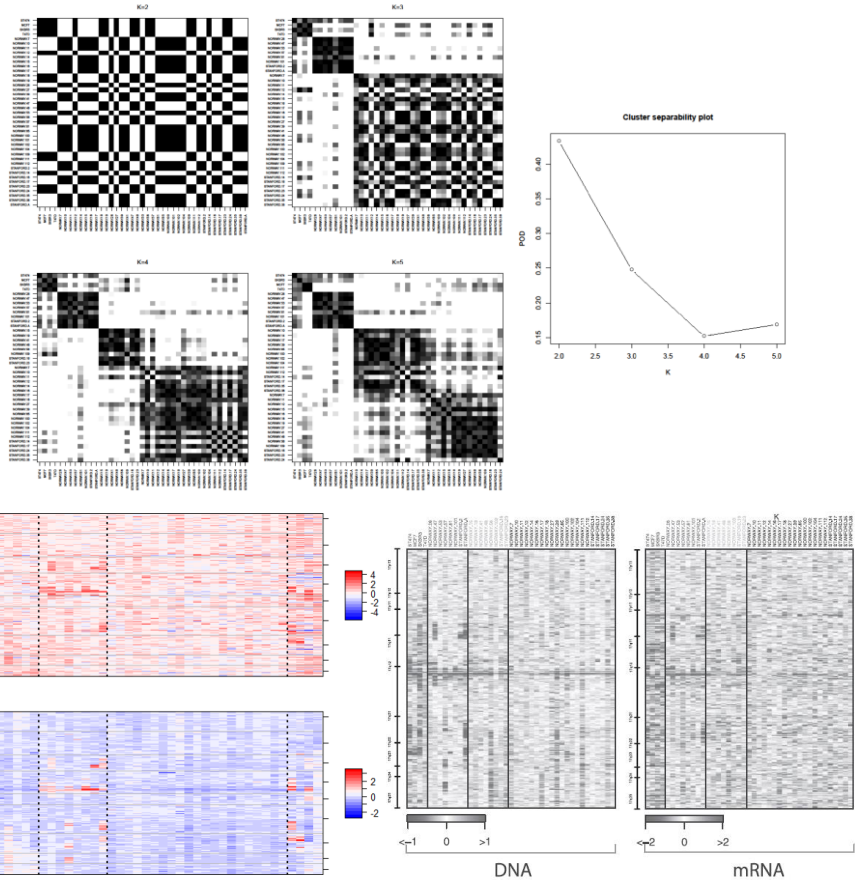


Heatmap result



iCluster results

- Integrative clustering using a joint latent variable model
- Unlike SNF, omics data are **gene-centric**
- By POD (Proportion Of Deviance), iCluster select the best number of clusters (i.e. subtypes of data)
- **Identify subtypes** characterized by
  - concordant DNA copy number changes and gene expression
  - unique profiles specific to one or other
- Discover potentially **novel subtypes** by combining weak yet consistent alteration patterns across data types



Installing iCluster (R package)

- iCluster is currently provided in R
- In this tutorial we will use R
- Create a new directory for iCluster
- Use the full COAD data (pre-processed)

```
$ cd 2_iCluster
$ jupyter lab –port 11##
```

iCluster | Import multi-omics data

- Import the pre-processed multi-omics gene-centric matrices

```
                                    /2_iCluster$ ll iCluster_inp/
total 160096
drwxr-xr-x 2          4096 Jan 30 14:30 ./
drwxr-xr-x 3          4096 Jan 30 14:39 ../
-rw-r----- 1      51324792 Jan 30 14:29 COAD_gcentric_gene.txt
-rw-r----- 1      75627710 Jan 30 14:29 COAD_gcentric_methylation.txt
-rw-r----- 1      36972630 Jan 30 14:29 COAD_gcentric_mirna.txt
```

Input data to iCluster

```
## loading omics data as numeric matrices

# read gene expression data
ge <- as.matrix(read.table("iCluster_inp/COAD_gcentric_gene.txt", row.names = 1, header = T))

# read methylation data
me <- as.matrix(read.table(" iCluster_inp /COAD_gcentric_methylation.txt", row.names = 1, header=T))

# read miRNA data
mi <- as.matrix(read.table(" iCluster_inp /COAD_gcentric_mirna.txt", row.names = 1, header=T))
```

iCluster | Data normalization

- If data need to be normalized, use ‘scale’ function
  - calculate the mean and standard deviation of the entire vector, and then ‘scale’ each element

Gene expression data

Methylation data

miRNA data

```
## normalize data

# gene expression data normalization
ge_norm <- scale(ge)
rownames(ge_norm) <-  rownames(ge)
colnames(ge_norm) <- colnames(ge)

# methylation data normalization
me_norm <- scale(me)
rownames(me_norm) <-  rownames(me)
colnames(me_norm) <- colnames(me)

# miRNA data normalization
mi_norm <- scale(mi)
rownames(mi_norm) <-  rownames(mi)
colnames(mi_norm) <- colnames(mi)
```

iCluster | Gene selection and merge

- iCluster need prior gene selection
  - measure the importance of genes by their standard deviation across the sample
  - here, select top 500 importance genes based on gene expression data
- The dimension of each list of merged data is $\mathbf{samples} \times \mathbf{top\ 500\ importance\ genes}$

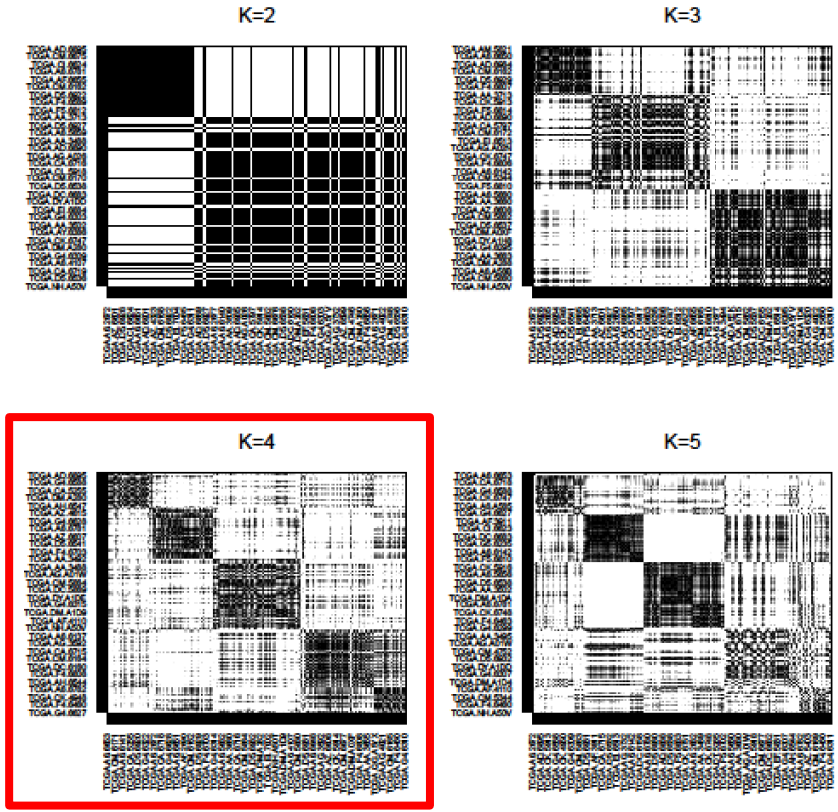
```
# select top 500 genes by variance
top_number = 500
selected_genes <- names(tail(sort(apply(ge_norm, 1, var)), n = top_number))

ge_mat <- ge_norm[selected_genes, ]
me_mat <- me_norm[selected_genes, ]
mi_mat <- mi_norm[selected_genes, ]

# merge data
data <- list(t(ge_mat), t(me_mat), t(mi_mat))
names(data) <- c("ge_mat", "me_mat", "mi_mat")
```

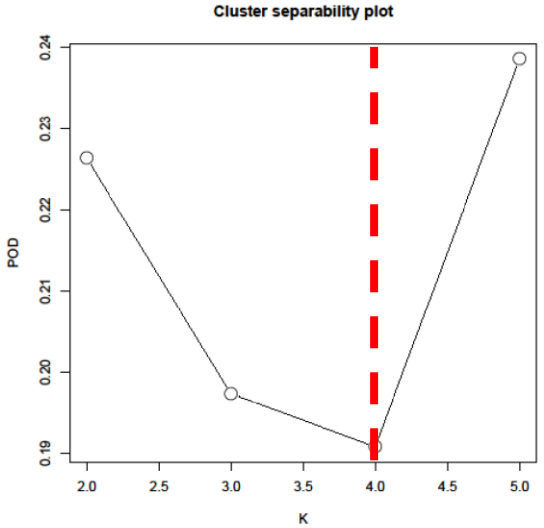
iCluster | Run iCluster and model selection

- Run iCluster with merged data
- To determine the best cluster number, calculate POD (Proportion Of Deviance)
  - $$\text{POD} = \frac{|\text{'standardized iCluster result'} - \text{'perfect diagomal matrix'}|}{(\text{number of samples})^2}$$

```
# iCluster model selection
par(mfrow = c(2,2))
pods = c()
for(I in 2:5){
                    fit = iCluster(data, k = i, lambda = c(0.3, 0.3, 0.3), max.iter = 100);
                    plotiCluster(fit = fit, label = rownames(data[[2]]));
                    cat("\n")
                    pods = c(pods, compute.pod(fit))
}

par(mfrow = c(1,1))
plot(c(2:5), pods, type='b', cex = 3, xlab = "K", ylab="POD", main="Cluster separability plot")
```

iCluster | Run iCluster and model selection



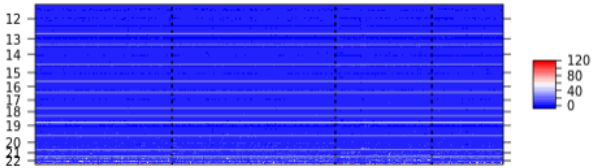


iCluster | Plot heatmap

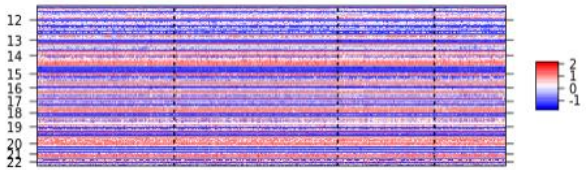
- Using genomic coordinates, plot a heatmap of the best cluster number 4.

```
# best cluster number: k=4
fit_4 = = iCluster(data, k = 4, lambda = c(0.3, 0.3, 0.3), max.iter = 100)

# plot heatmap
data(coord)
chr = coord[,1]
plotHeatmap(fit = fit_4, data = data, plot.chr = c(TRUE, TRUE, TRUE), chr = chr)
```

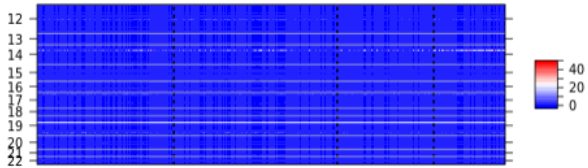
Gene expression



Methylation



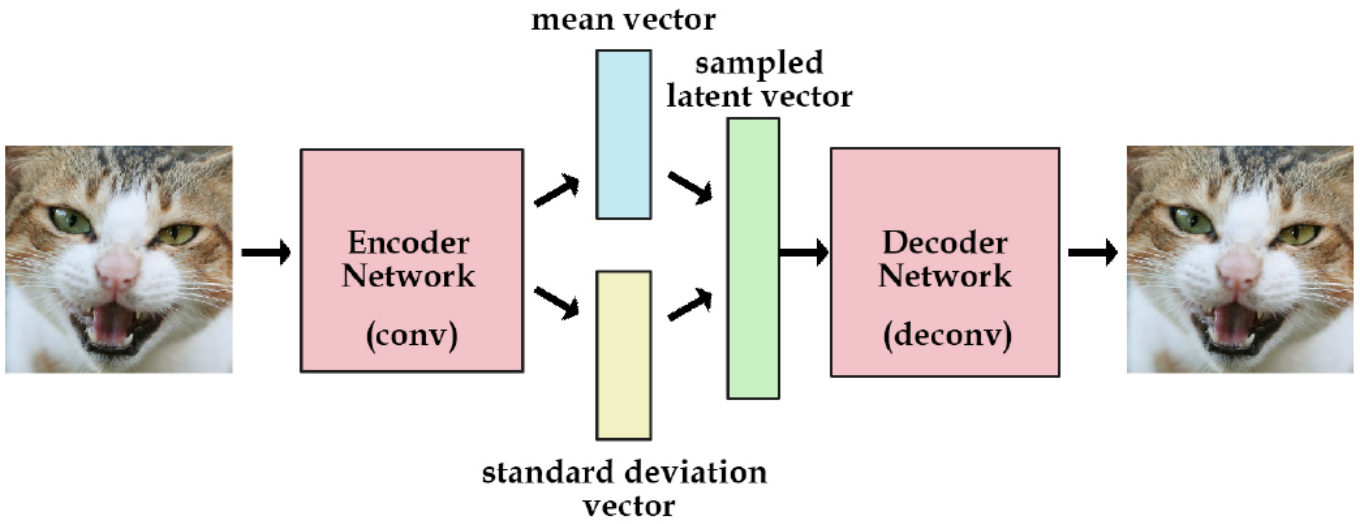
miRNA



Tool 3| VAE

“Auto-Encoding Variational Bayes
: Variational Auto-Encoder”
Diederik P Kingma, Max Welling, 2014, ICLR

VAE Workflow

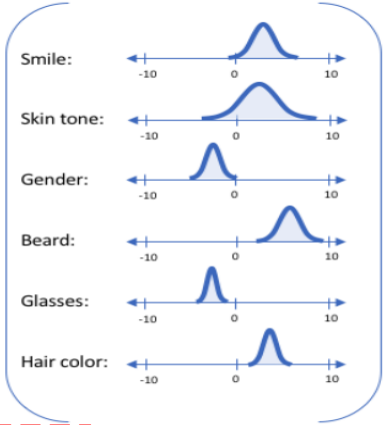


Variational Auto-Encoder

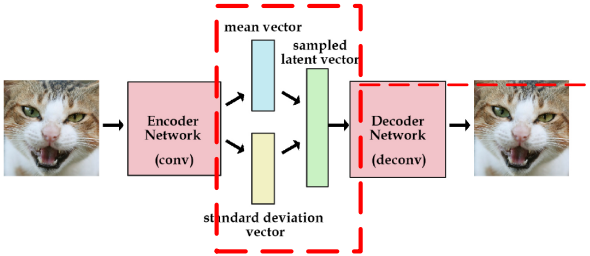
Variational Auto-Encoder: Intuition & Structure

How to move form our sample X_i to latent space Z_i, and reconstruct $\tilde{X}_i$

encoder

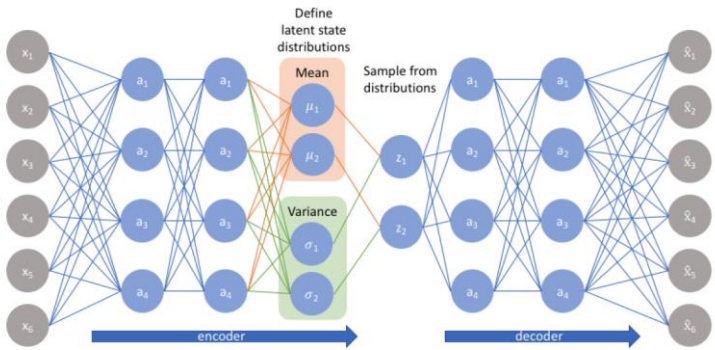


decoder



Variational Auto-Encoder

Latent attributes

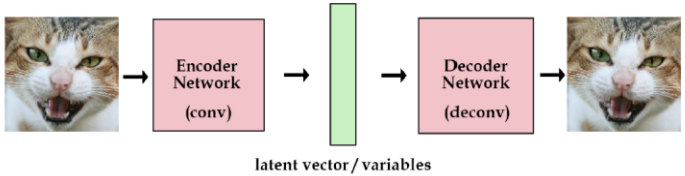


https://www.jeremyjordan.me/variati

Comparison with Auto-Encoder

Auto-Encoder

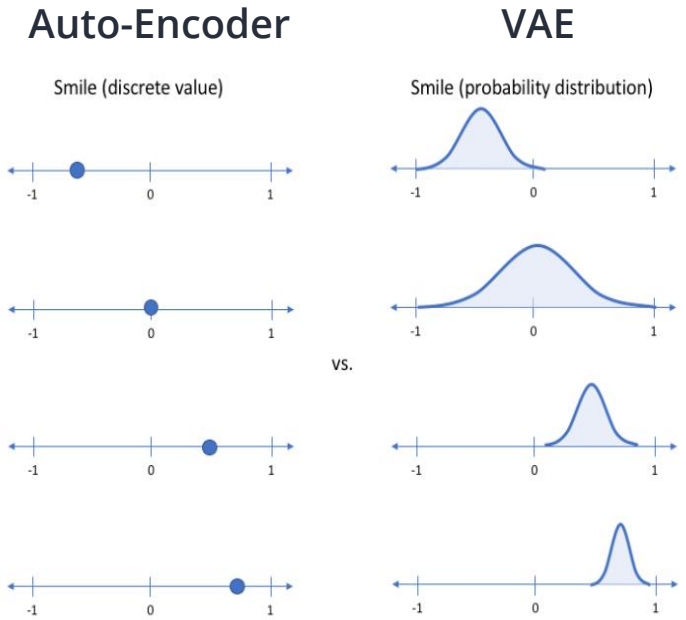
- Reconstruction model
- Learn a “compressed representation” of input



Auto-Encoder

Variational Auto-Encoder

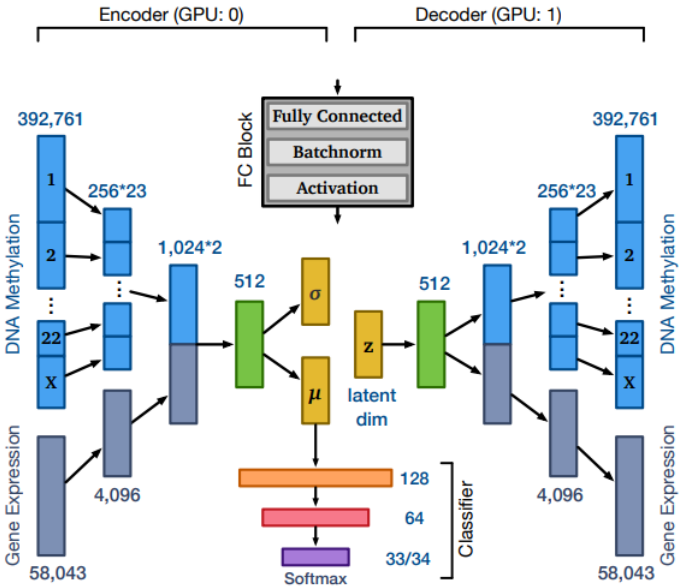
- Generative model
- Learn a “compressed representation” of input
- (New!) Learn **the parameters of a probability distribution** representing the data



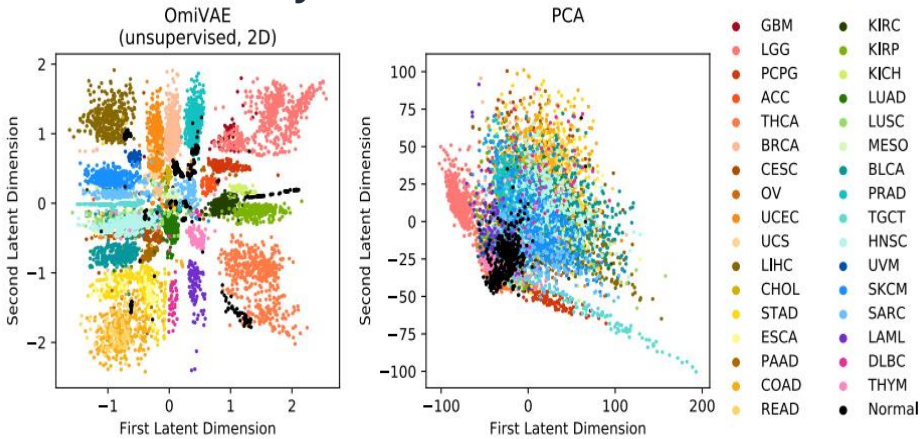
https://www.jeremyjordan.me/variational-autoencoders/

Integrated Multi-omics Analysis Using Variational Autoencoders: Application to Pan-cancer Classification

OmiVAE



2D embedding of the multi-omics data learned by OmiVAE and PCA



Zhang, Xiaoyu, et al. "IntegratedMulti-omics Analysis Using Variational Autoencoders: Application to Pan-cancer Classification." arXiv preprint arXiv:1908.06278 (2019).

VAE | Preprocessing multi-omics data

- Pre-process multi-omics data: feature selection and data splitting

```
$ ll data/
```

```
                                /3_VAE$ ll data/
total 160104
drwxr-xr-x 2              4096 Jan 30 15:34 ./
drwxr-xr-x 5              4096 Jan 30 15:35 ../
-rw-r----- 1          51324792 Jan 30 15:33 COAD_gcentric_gene.txt
-rw-r----- 1          75627710 Jan 30 15:33 COAD_gcentric_methylation.txt
-rw-r----- 1          36972630 Jan 30 15:33 COAD_gcentric_mirna.txt
-rw-r----- 1              5670 Jan 30 15:33 subtypes.txt
```

Input data to VAE

VAE | Preprocessing multi-omics data

- Pre-process multi-omics data: feature selection and data splitting

```
$ python bin/preprocessing.py
$ ll data/
```

```
                                /VAE2$ ll data/
total 161144
drwxr-xr-x 2          4096 Jan 30 14:54 ./
drwxr-xr-x 5          4096 Jan 30 15:27 ../
-rw-r----- 1      51324792 Jan 30 14:50 COAD_gcentric_gene.txt
-rw-r----- 1      75627710 Jan 30 14:50 COAD_gcentric_methylation.txt
-rw-r----- 1      36972630 Jan 30 14:50 COAD_gcentric_mirna.txt
-rw-r--r-- 1          6600 Jan 30 15:29 feature_order.txt
-rw-r----- 1          5670 Jan 13 14:10 subtypes.txt
-rw-r--r-- 1        105628 Jan 30 15:29 test_data.txt
-rw-r--r-- 1            64 Jan 30 15:29 test_label.txt
-rw-r--r-- 1        831600 Jan 30 15:29 train_data.txt
-rw-r--r-- 1           504 Jan 30 15:29 train_label.txt
-rw-r--r-- 1        102333 Jan 30 15:29 val_data.txt
-rw-r--r-- 1            62 Jan 30 15:29 val_label.txt
```

VAE | Preprocessing multi-omics data

- preproceessing.py

Read data_dir & data

Transpose
gene * sample ->
sample * gene

Rename columns
of data frame

```
# Get directory information
bin_dir = os.path.dirname(sys.argv[0])
data_dir = os.path.normpath(os.path.join(bin_dir, "../data"))

# Read Data
ge_df = pd.read_csv(os.path.join(data_dir, "COAD_gcentric_gene.txt"), sep="\t", index_col=0)
me_df = pd.read_csv(os.path.join(data_dir, "COAD_gcentric_methylation.txt"), sep="\t", index_col=0)
mi_df = pd.read_csv(os.path.join(data_dir, "COAD_gcentric_mirna.txt"), sep="\t", index_col=0)

# Transpose Data
ge_df = ge_df.T
me_df = me_df.T
mi_df = mi_df.T

# Rename features
ge_df = ge_df.add_suffix('_mRNA')
me_df = me_df.add_suffix('_methyl')
mi_df = mi_df.add_suffix('_miRNA')
```

VAE | Preprocessing multi-omics data

- preproceessing.py

Data filtering by variance

Gene ranking and selecting

```
# Remove zero variance features (not significant)
ge_cut = ge_df.var(axis=0) != 0.0
me_cut = me_df.var(axis=0) != 0.0
mi_cut = mi_df.var(axis=0) != 0.0

all_omics_filtered = ge_cut.values & me_cut.values & mi_cut.values

ge_df = ge_df.loc[:, all_omics_filtered]
me_df = me_df.loc[:, all_omics_filtered]
mi_df = mi_df.loc[:, all_omics_filtered]

# Select Top 100 genes (from gene data)
top_n = 100
top_genes = ge_df.var(axis=0).values.argsort()[-top_n:][::-1]

ge_df = ge_df.iloc[:, top_genes]
me_df = me_df.iloc[:, top_genes]
mi_df = mi_df.iloc[:, top_genes]

# Merging data
MGD_df = pd.concat([ge_df, me_df, mi_df], axis=1)
```

VAE | Preprocessing multi-omics data

- preproceessing.py

feature info & labelling

Split data

```
# Feature info
np.savetxt(os.path.join(data_dir, "feature_order.txt"), MGD_df.columns.values, fmt="%s")

# Label info
subtypes_df = pd.read_csv(os.path.join(data_dir, "subtypes.txt"), sep="\t", header=None)
subtypes = []

for elem in subtypes_df.iloc[:,1].values:
    if elem == "CMS1":
        subtypes.append(0)
    elif elem == "CMS2":
        subtypes.append(1)
    elif elem == "CMS3":
        subtypes.append(2)
    else:
        subtypes.append(3)

# Split data into train:vallidation:test
X_train, X_val_test, y_train, y_val_test = train_test_split(MGD_df.to_numpy(), subtypes, test_size=0.2,
random_state=3)
X_val, X_test, y_val, y_test = train_test_split(X_val_test, y_val_test, test_size=0.5, random_state=3)
```

VAE | Preprocessing multi-omics data

- preproceessing.py

Normalize data into 0 to 1

```
# Normalize data
scaler = MinMaxScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_val = scaler.transform(X_val)
X_test = scaler.transform(X_test)
```

VAE | Constructing model and training

- Run VAE

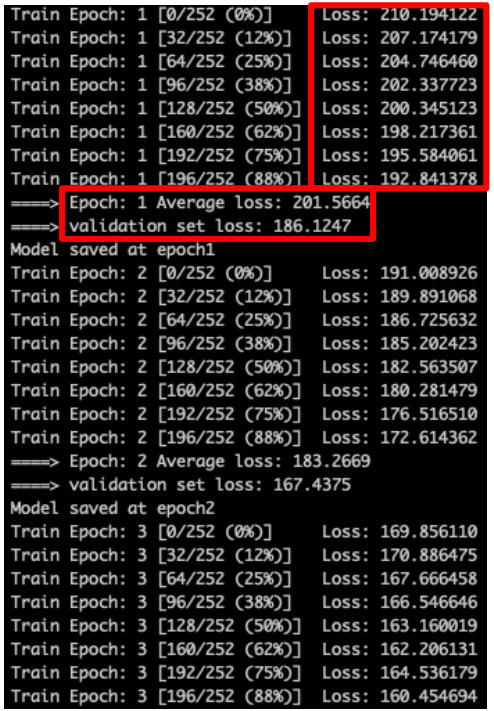
```
$ python vae.py -h
```

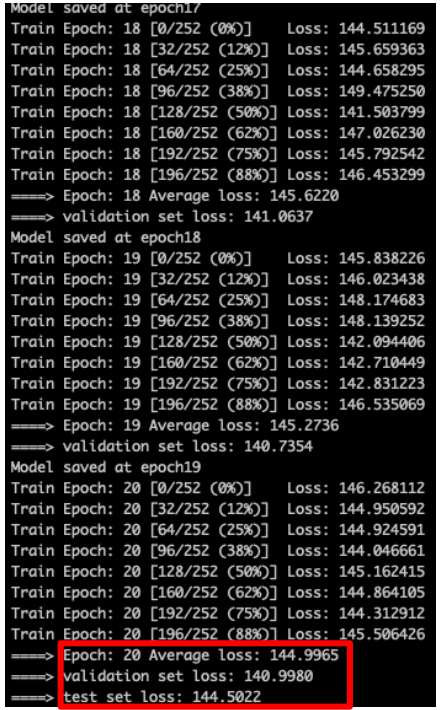
```
usage: vae.py [-h] [--batch-size N] [--epochs N] [--no-cuda] [--seed S]
              [--log-interval N] [--out_dir OUT_DIR]

VAE practice

optional arguments:
  -h, --help         show this help message and exit
  --batch-size N     input batch size for training (default: 32)
  --epochs N         number of epochs to train (default: 10)
  --no-cuda          enables CUDA training
  --seed S           random seed (default: 1)
  --log-interval N   how many batches to wait before logging training status
  --out_dir OUT_DIR  Model save_dir
```

VAE | Constructing model and training

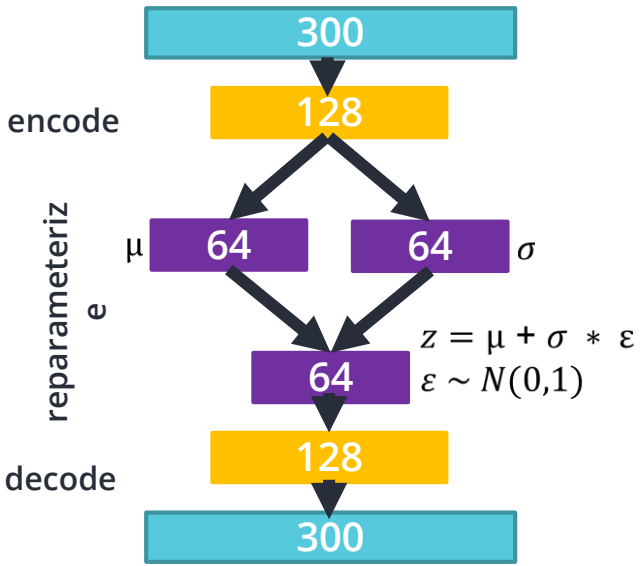
- Run vae.py

```
$ python vae.py --out_dir /results --epoch 20
```





VAE | Constructing model and training

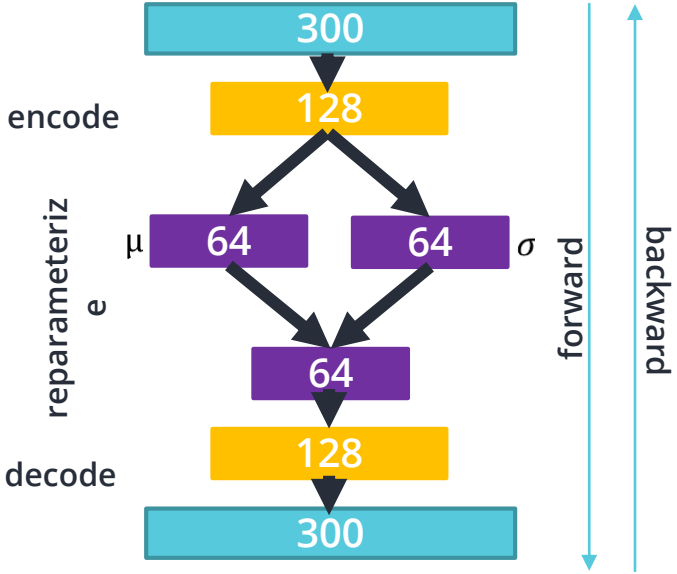



```
# class VAE(nn.Module):
    def __init__(self):
        super(VAE, self).__init__()

        self.fc1 = nn.Linear(300, 128)
        self.fc21 = nn.Linear(128, 64)
        self.fc22 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 128)
        self.fc4 = nn.Linear(128, 300)

    def encode(self, x):
        h1 = F.relu(self.fc1(x))
        return self.fc21(h1), self.fc22(h1)
```

```
def reparameterize(self, mu, logvar):
        std = torch.exp(0.5*logvar)
        eps = torch.randn_like(std)
        return mu + eps*std

def decode(self, z):
        h3 = F.relu(self.fc3(z))
        return F.sigmoid(self.fc4(h3))

def forward(self, x):
        mu, logvar = self.encode(x)
        z = self.reparameterize(mu, logvar)
        return self.decode(z), mu, logvar
```

```
for epoch in range(1, args.epochs + 1):
        train(epoch)

        val_loss = test(val_loader, "validation")

        if val_loss < min_loss:
            torch.save(model.state_dict(), os.path.join(args.out_dir, 'best_model.pth'))
            print("Model saved at epoch{}".format(epoch))
            min_loss = val_loss)
```

VAE | Constructing model and training

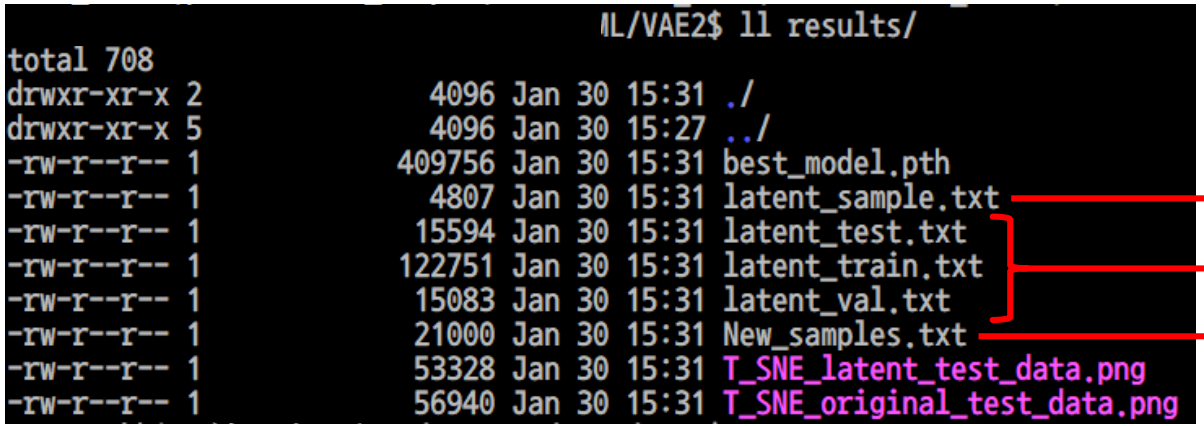



```
def train(epoch):
    model.train()
    train_loss = 0
    for batch_idx, (data, _) in enumerate(train_loader):
        data = data.to(device)
        recon_batch, mu, logvar = model(data)
        loss = loss_function(recon_batch, data, mu, logvar)
        train_loss += loss.item()
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        if batch_idx % args.log_interval == 0:
            print('Train Epoch: {} [{}/{} ({:.0f}%)]\tLoss: {:.6f}'.format(
                epoch, batch_idx * len(data), len(train_loader.dataset),
                100. * batch_idx / len(train_loader),
                loss.item() / len(data)))

    print('====> Epoch: {} Average loss: {:.4f}'.format(
        epoch, train_loss / len(train_loader.dataset)))
```

VAE | Constructing model and training

- VAE results visualization

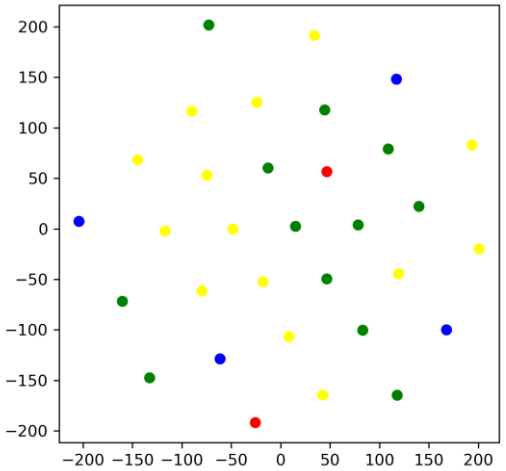
```
$ python bin/tsne_visualization.py
$ ll results/
```



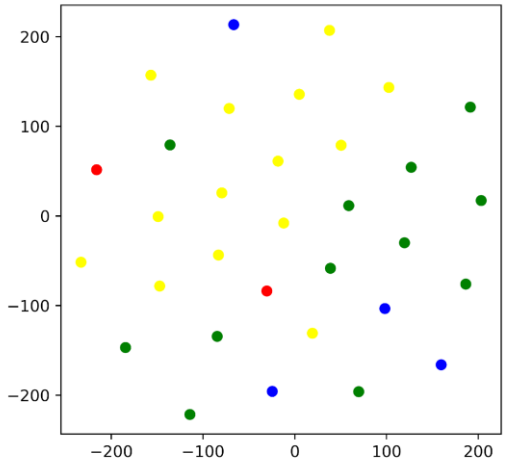
VAE | Constructing model and training

- VAE results visualization

Original Features of test data



Latent Features of test data

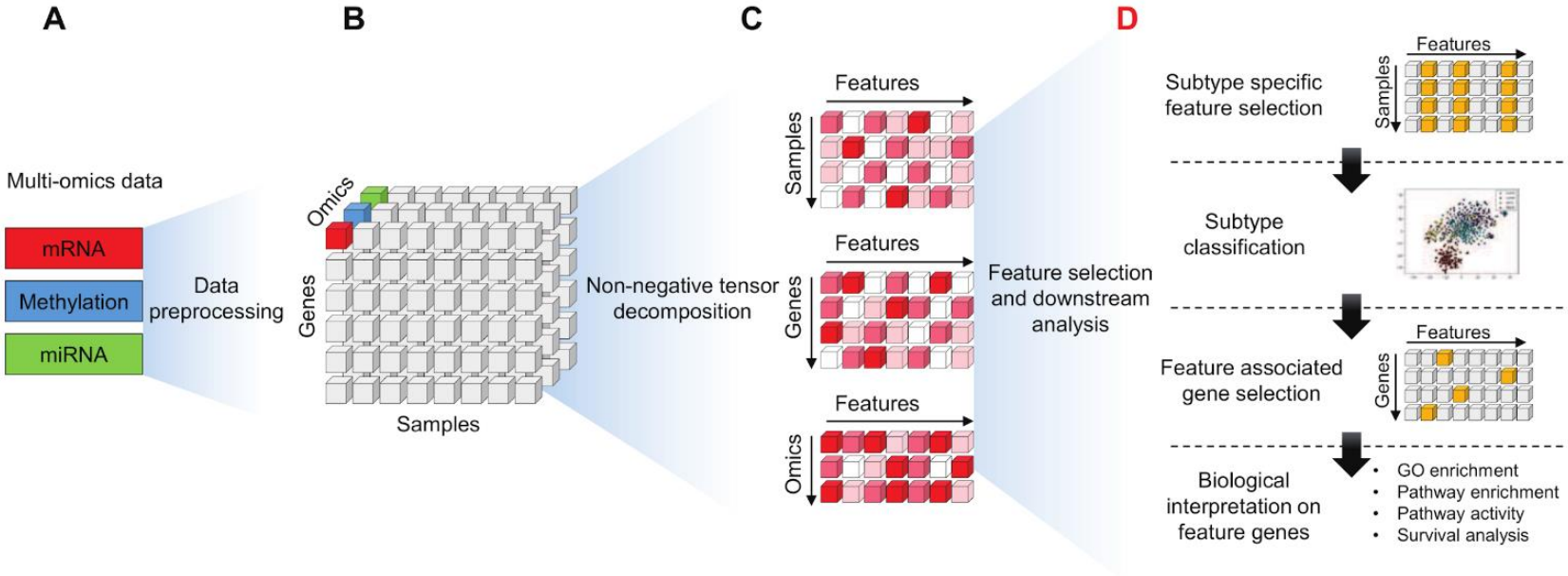


Tool 4| MONTI

“MONTI: A Multi-omics Non-negative Tensor decomposition framework for the Integrated analysis of cancer subtypes”

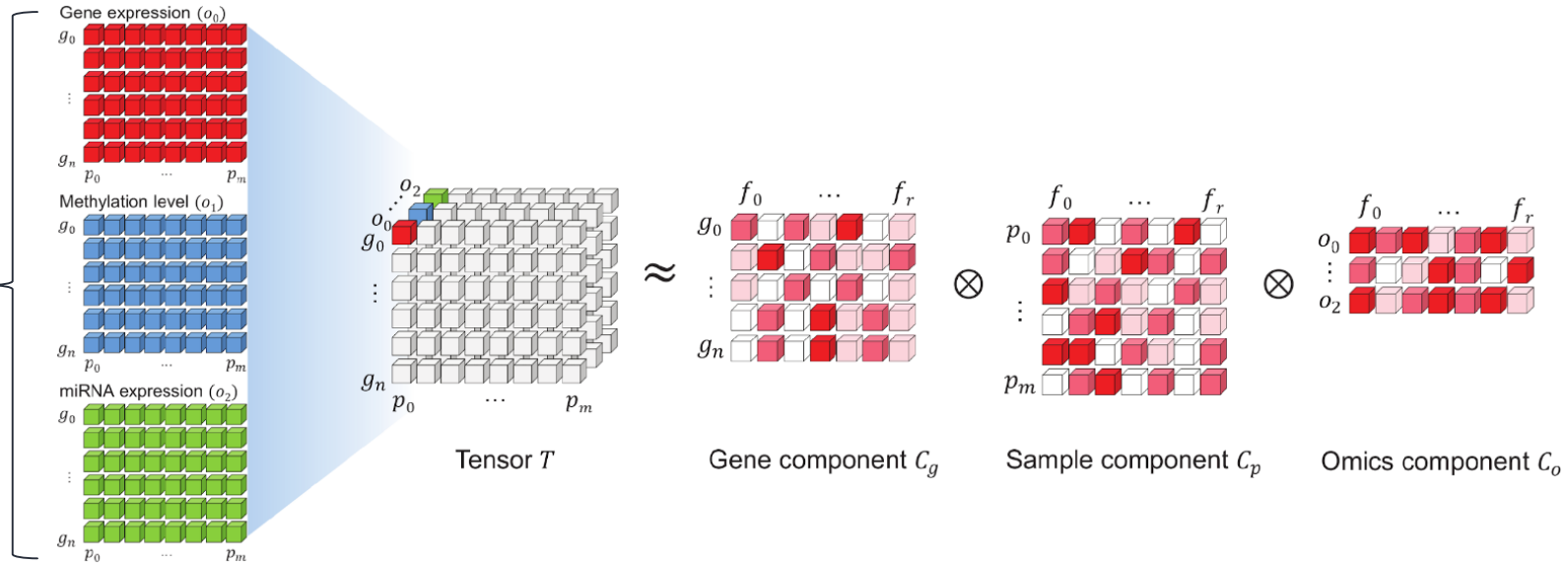
Inuk Jung et al., Under review 2020

MONTI Workflow



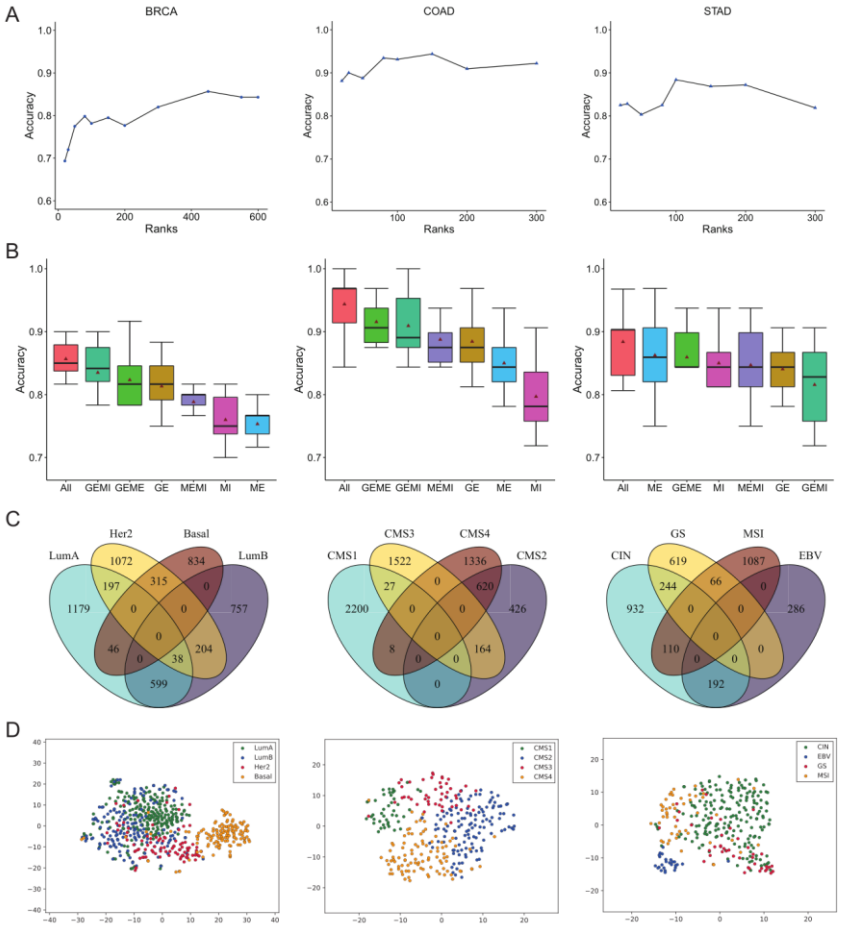
Multi-omics Non-negative Tensor Decomposition

Gene-centric 2D omics matrices

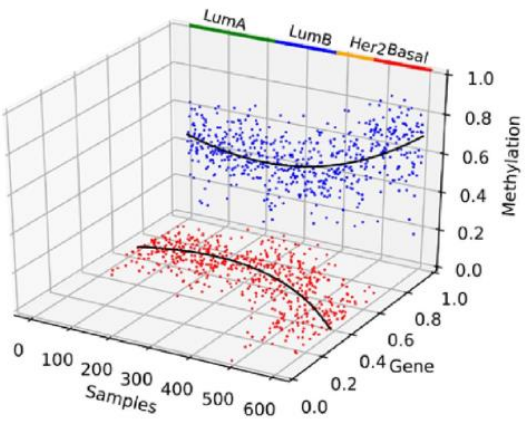


Results

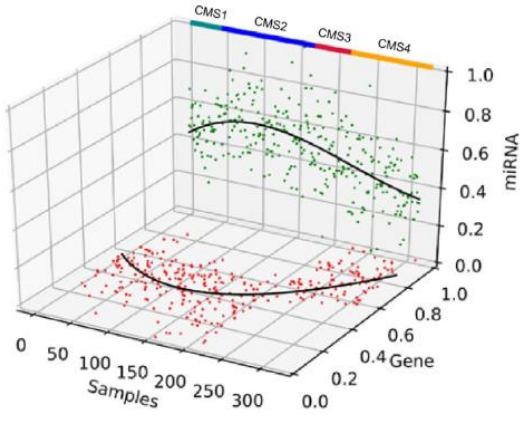
- A comparative figure for the three case studies
  - BRCA, COAD and STAD
- COAD showed the highest subtype classification accuracy
- The composition of omics data showed different impacts on subtype classification accuracy
  - For STAD, ME seemd to be most informative, where it reached 85% using onlye methylation data
  - However, overall, the accuracy was the highest when using all omics data



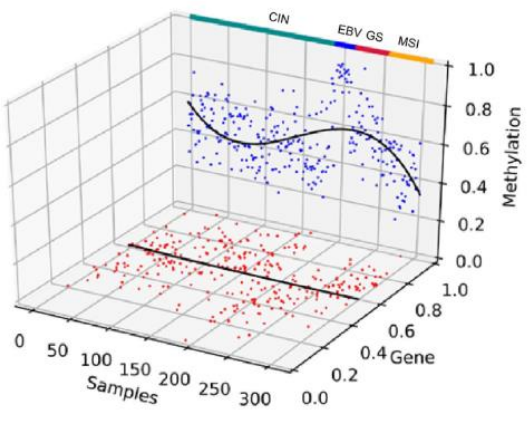
MO genes



EXOC6 (Basal)



OLFML2B (CMS4)



MAPK15 (EBV)

Installing MONTI (1/2)

- Download MONTI here
  - http://cobi.knu.ac.kr/MO_workshop_2020/tools/MONTI_v2.2.tar.gz

- Uncompress file and set MONTI path variable

```
$ tar –xzvf MONTI_v2.2.tar.gz
$ export MONTI=/~~~/bin
$ export PATH=$PATH:$MONTI
```

- Check if MONTI path is set

```
$ echo $MONTI
> /data-Raid10/home/inukj/mysoftwares/monti
$ echo $PATH
```

Installing MONTI (2/2)

- MONTI is implemented using Python 3.7. If you are using Python2 please update to Python3.
- Install pre-requisite python packages
  - pre-requisite packages: tensorly, argparse, joblib, matplotlib, lifelines

```
$ sudo python install_monti.py
```

- The following python packages need to be installed separately.
  - numpy v1.16.2 (currently does not work with numpy 1.17)
  - sklearn

MONTI command

- The "`monti.py`" is a wrapper function that performs the MONTI workflow

Mandatory arguments

Optional arguments

```
usage: monti.py [-h] -f INPUT_FILE -r RANK -s SAMPLE_INFO
                            [-surv SURVIVAL_INFO] [-o OUTDIR] [--plot]
                            [--dmax_iter DMAX_ITER] [--alpha ALPHA]
                            [-pre PREPROCESS_DIR]

# mandatory arguements
-f : the input tensor data (a numpy ndarray)
-r : the number of ranks that the tensor is to be decomposed with
-s : a two column text file containing sample IDs and its associated breast cancer subtype
-g : a two column text file containing gene IDs and gene symbols

# optional arguements
-surv: a trhee column text file with 'sample IDs', 'event' and 'survival time' as column attributes
-o: the output directory name (default: 'output')
-pre: the directory of rawdata that need to be pre-processed
--plot: indicator for drawing gene plots and tSNE plot
--dmax_iter: the number of maximum iterations during tensor decomposition (default: 300)
--alpha: the L1 penalty weight (default: 0.01)
```

Input arguments

- The 4 mandatory input arguments
  1. Omics input data (-f): a numpy ndarray with multi-omics data (tensor format)
  2. Rank (-r): The number of ranks for decomposition
  3. Sample information (-s): Subtype information of each sample ID
  4. Gene information (-g): Gene information – [ENSG ID, GeneSymbol]
     - Not restricted to any gene id type (must be a two column file)

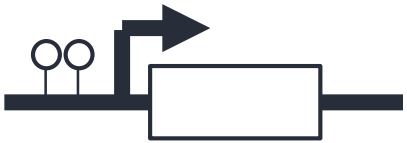
Input Data file (1/2)

- COAD data set includes
  - gene expression (GE), methylation level (ME) and miNRA expression (MI) omics data
- Data in gene-centric format

GE

Use as is

ME



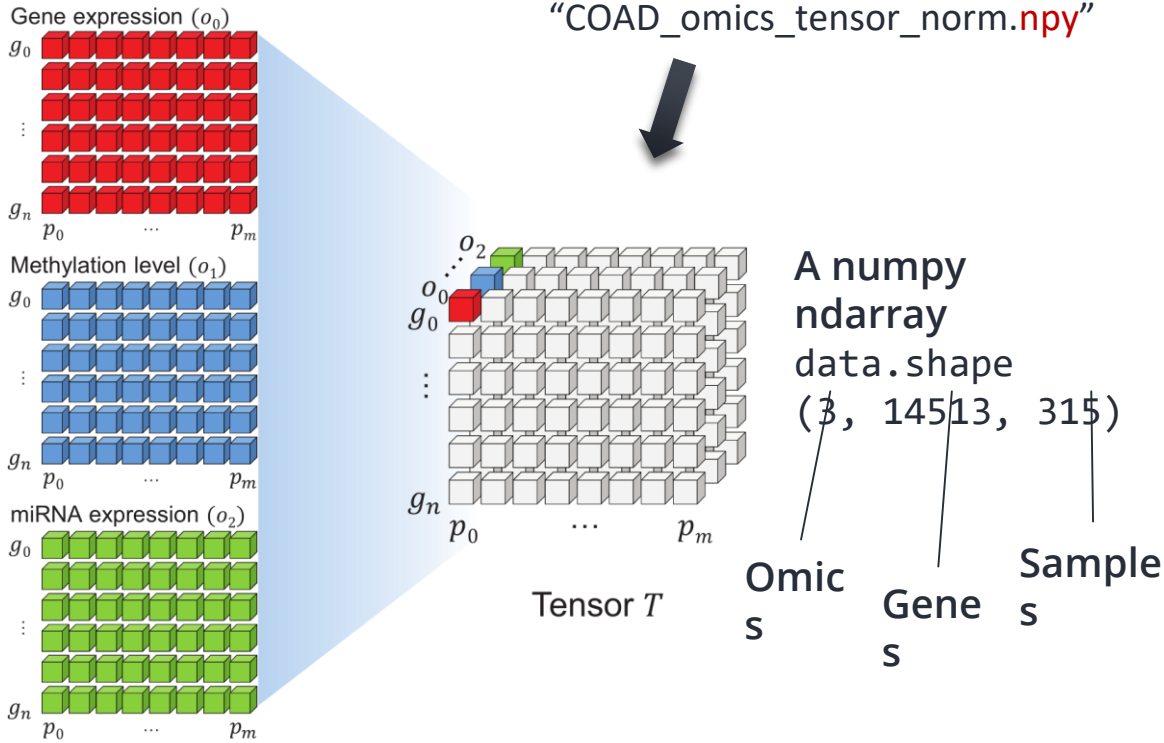
Average methylation level of probes in 2KB upstream of TSS

M I

Average expression value of miRNAs targeting a gene

Input Data file (2/2)

- So a 2D matrix represent for GE, ME, MI respectively
- The COAD data
  - N=315
  - Genes=14,513
  - Omics=3 (GE, ME, MI)



Executing MONTI using COAD data

- Move to the MONTI path (where it was uncompressed)
- Create output directory "`workshop_out`"
- The decomposition may take up to several hours for large data set.
  For timely analysis the decomposed result is provided (“`r150_td.npy`”).
  Copy it from the "`output/components`" to "workshop_out"

```
$ mkdir workshop_out
$ cp -r ./dataset/COAD/output/components workshop_out
```

- Input the gene-centric COAD multi-omics data and sample info to MONTI
  (*press "`n`" if asked to override decomposed result)

```
$ cd ./dataset/COAD
$ python ../../bin/monti.py -f inputdata/COAD_omics_tensor_norm.npy -r 150
-s inputdata/sample_info.txt -g inputdata/gene_info.txt -o ../../workshop_out/ --plot
```

Console Output

```
Starting MONTI (v2.1)

Sample classes: [CMS1, CMS2, CMS3, CMS4]
Samples: 315
Omics: 3
Genes: 14513

Rank: 150
alpha: 0.01
Output directory: "workshop_out/"

----------------------------------------

Skipping tensor decomposition.

----------------------------------------

selecting sample features... done

Selected subtype features:      39
Classification accuracy:        0.893750

----------------------------------------
```

Subtype classification accuracy using sample features

```
selecting feature genes... done
Total 3923 genes selected.
  - CMS1: 1320 genes
  - CMS2: 855 genes
  - CMS3: 729 genes
  - CMS4: 1593 genes
Classification accuracy:        0.925000

----------------------------------------

plotting CMS1 genes...
Progress: |##################################
plotting CMS2 genes...
Progress: |##################################
plotting CMS3 genes...
Progress: |##################################
plotting CMS4 genes...
Progress: |##################################
plotting CMS2,CMS3 genes...
Progress: |##################################
plotting CMS2,CMS4 genes...
Progress: |##################################
drawing sample tSNE plots... done
```

Subtype classification accuracy using features genes

Plotting MO genes and t-SNE plot of the subtype specific genes

Output files

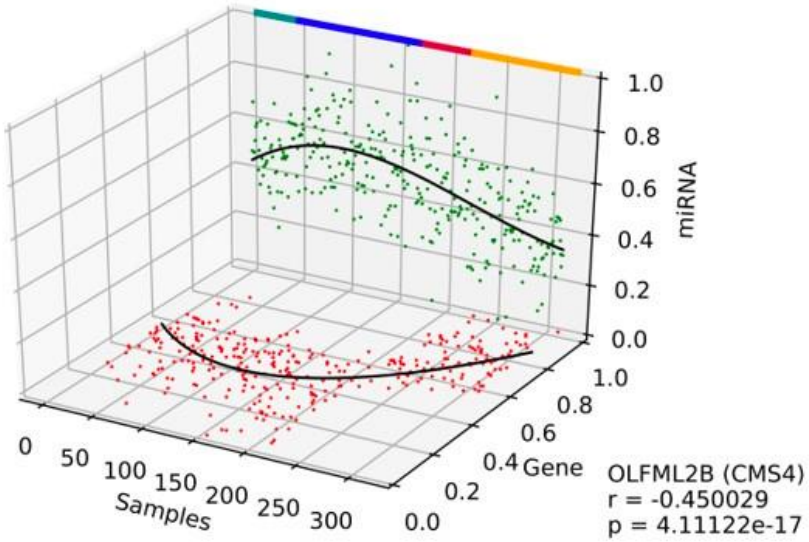
- List of result files using the COAD data:
  - `feature_genes_r150.txt`: List of cancer subtype associated genes
  - `sample_features_r150.txt`: List of cancer subtype associated features
  - `accuracy_patients_r150.txt`: Classification accuracy using sample features
  - `accuracy_genes_r150.txt`: Classification accuracy using feature genes
  - `patient_models/`: The MLP classification models generated using the sample features
  - `gene_models/`: The MLP classification models generated using the gene features
  - `plots/`:
    - `gene_plots_<subtypes>.pdf`: The multi-omics scatter plots of the subtype associated genes
    - `sample_tSNE.pdf`: The t-SNE plot of the samples using the selected features

Results (1/2)

- For the COAD data
  - 40 features (out of 150) were selected – they show subtype specific omics profiles
  - Using the 40 features we achieved 88.75% subtype classification accuracy
  - Total of 3923 genes (out of 14,513) were associated with the 40 features (and thus subtype specific)
  - Using the 3923 genes we achieved 92% subtype classification accuracy

Results (2/2)

gene_plots_CMS4.pdf



sample_tSNE.pdf