An introduction to evolutionary accumulation and cancer progression models

Ramon Diaz-Uriarte[1]

[1]Department of Biochemistry, Universidad Autonoma de Madrid, and Instituto de Investigaciones Biomédicas Sols-Morreale (IIBM), CSIC-UAM

Como School on Cancer Evolution, Villa del Grumello, Como, Italy

27 May 2025

Acknowledgments

- Iain Johnston
- Juan Díaz-Colunga, Claudia Vasallo, Íñigo Rios, Eric Macías, Pablo Herrera Nieto, Adrián López López, David Fontaneda, Asmaa Idmbarek
- Niko Beerenwinkel, Hessam Hosseini
- Comunidad de Madrid: PEJ-2019-AI/BMD-13961, PEJD-2018-POST/BMD-8960, PEJD-2016-BMD-2116
- PID2019-111256RB-I00 funded by MCIN/AEI/10.13039/501100011033

Outline of the talk

Cancer progression models / Evolutionary accumulation models

Uses and examples

EvAM-Tools

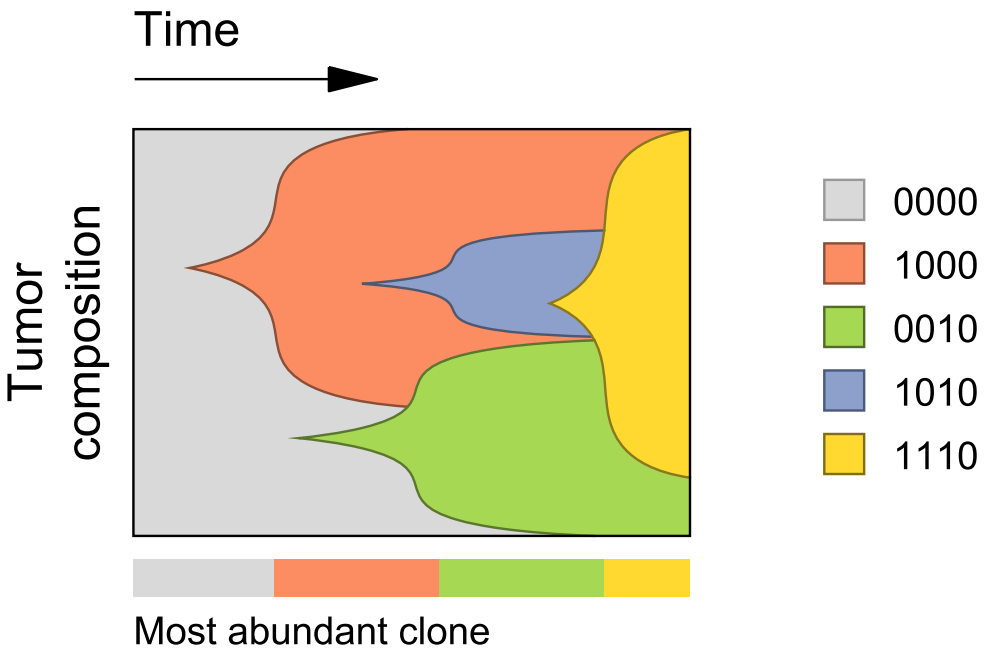
Cancer progression models / Evolutionary accumulation models

Uses and examples

EvAM-Tools

Cancer, evolution

"Cancer progression is caused by the sequential accumulation of somatic mutations during the life of an individual."
Hanahan & Weinberg, 2011, Cell



- Accumulation of mutations
- Not all orders of mutation accumulation equally likely

Cancer Progression (CPM) and Evolutionary Accumulation Models (EvAM)

Identify restrictions in the order of accumulation of mutations using cross-sectional data.

Mutations: events

Restrictions: deterministic dependencies, inhibiting/facilitating stochastic dependencies.

Not restricted to cancer: evolutionary accumulation models (EvAMs).

we assume "strong selection, weak mutation"

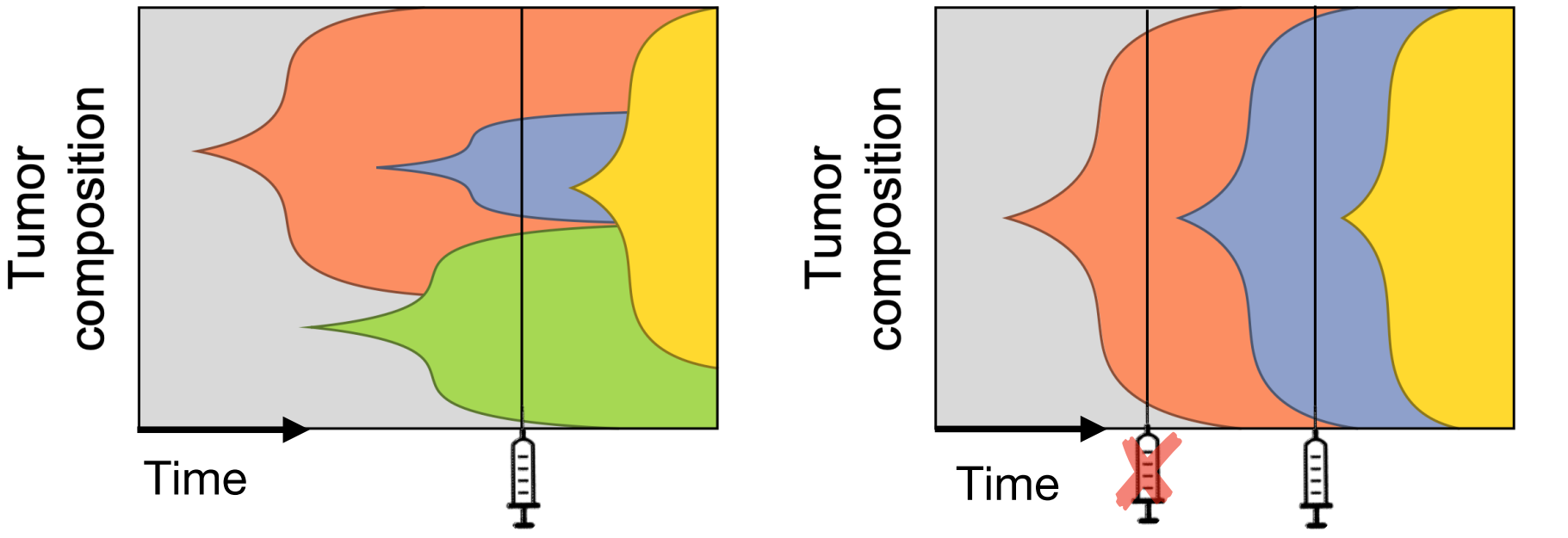


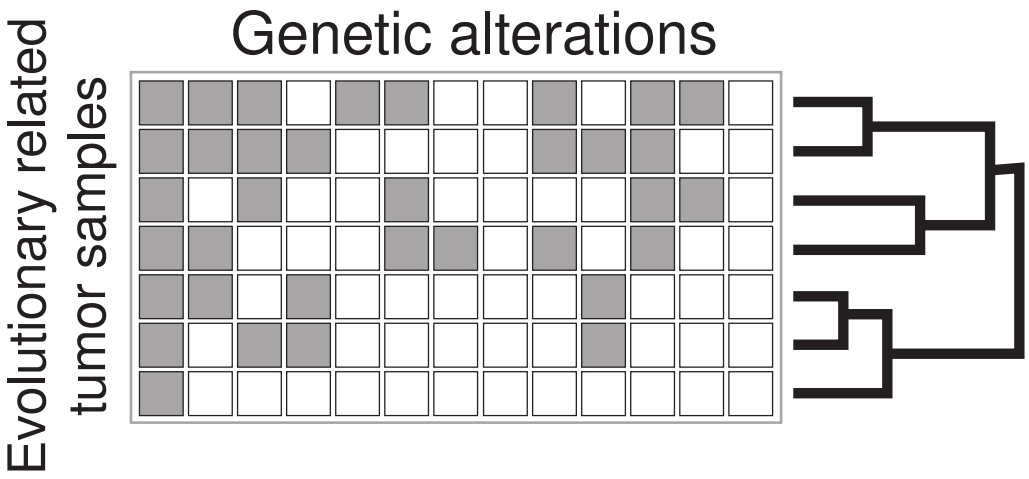
Figure modified from R. Schill talk at WHAM, Solstrand, Norway, November 2024, modified from Diaz-Colunga and Diaz-Uriarte, 2021, *PLOS Comp Biol*, 17.

What type of data?

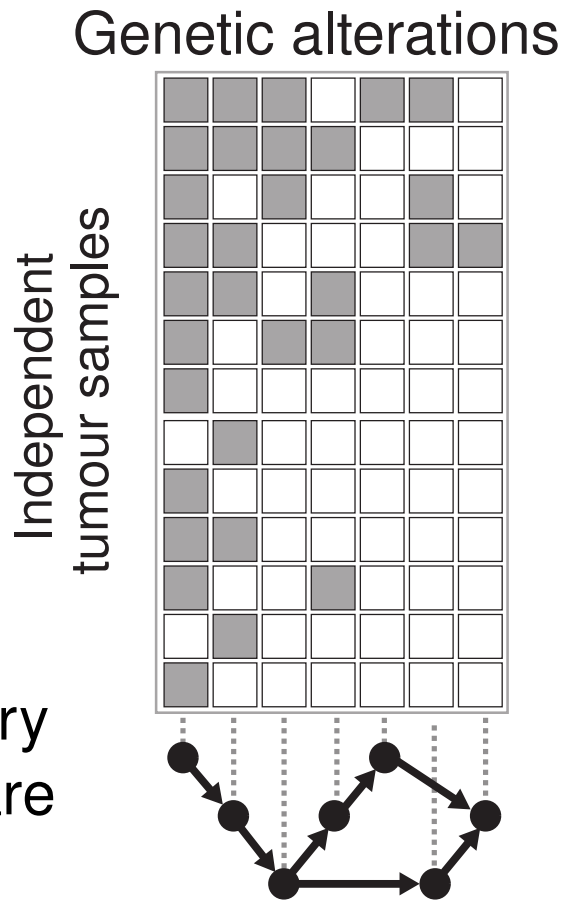
- Cross-sectional data: matrix of subjects by events (e.g., “genes”)
  - Binary
  - Multiple subjects at different stages
  - No info about stage of each subject
  - Event: Once gained, can’t be lost
  - One data point per subject: “tumour genotype”, “mutational profile”

Differences with phylogenetic inference

a) Phylogenetic models



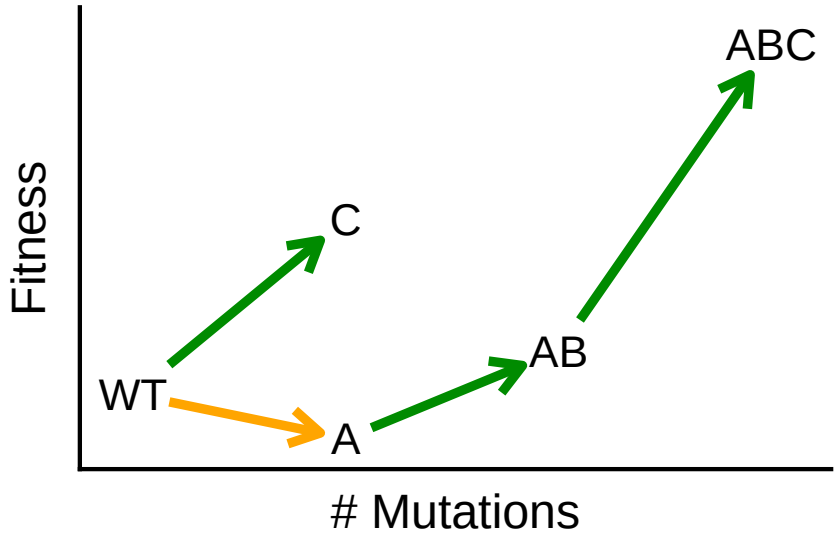
b) CPMs/EvAMs



Cross-sectional data: replicated evolutionary experiments where all individuals/entities are under the same genetic constraints.

Figure modified from Beerenwinkel et al., 2015. *Syst. Biol.*, 64.

Fitness landscape	LOD, POM, tumor genotype paths



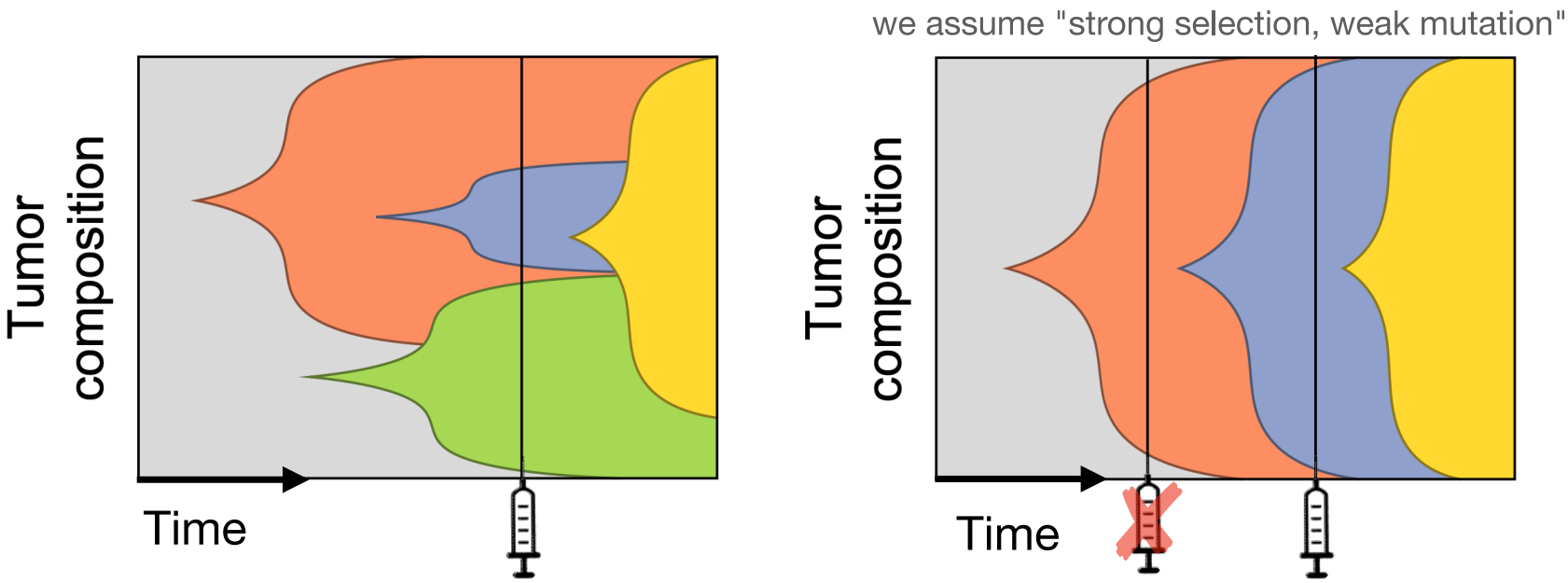
LOD: WT → A → AB → ABC

POM: WT → C → ABC

Tumor: (mutational profile)
- WT → C → AC → ABC
- WT → C → ABC

- **LOD**: $P(A|WT)$ and $P(AB|A)$, even if {A} rarely present
  - What happens in the future determines what is relevant
  - "evolutionary probability of genotype g" (Misra et al., 2014)
- **POM**: $P(A|WT) = 0$, $P(AB|A) = 0$

SSWM (strong selection, weak mutation)



- Whole tumour (bulk sequencing) vs. single-cell sampling: no difference
- Line of descent ≡ path of the maximum ≡ tumour genotype

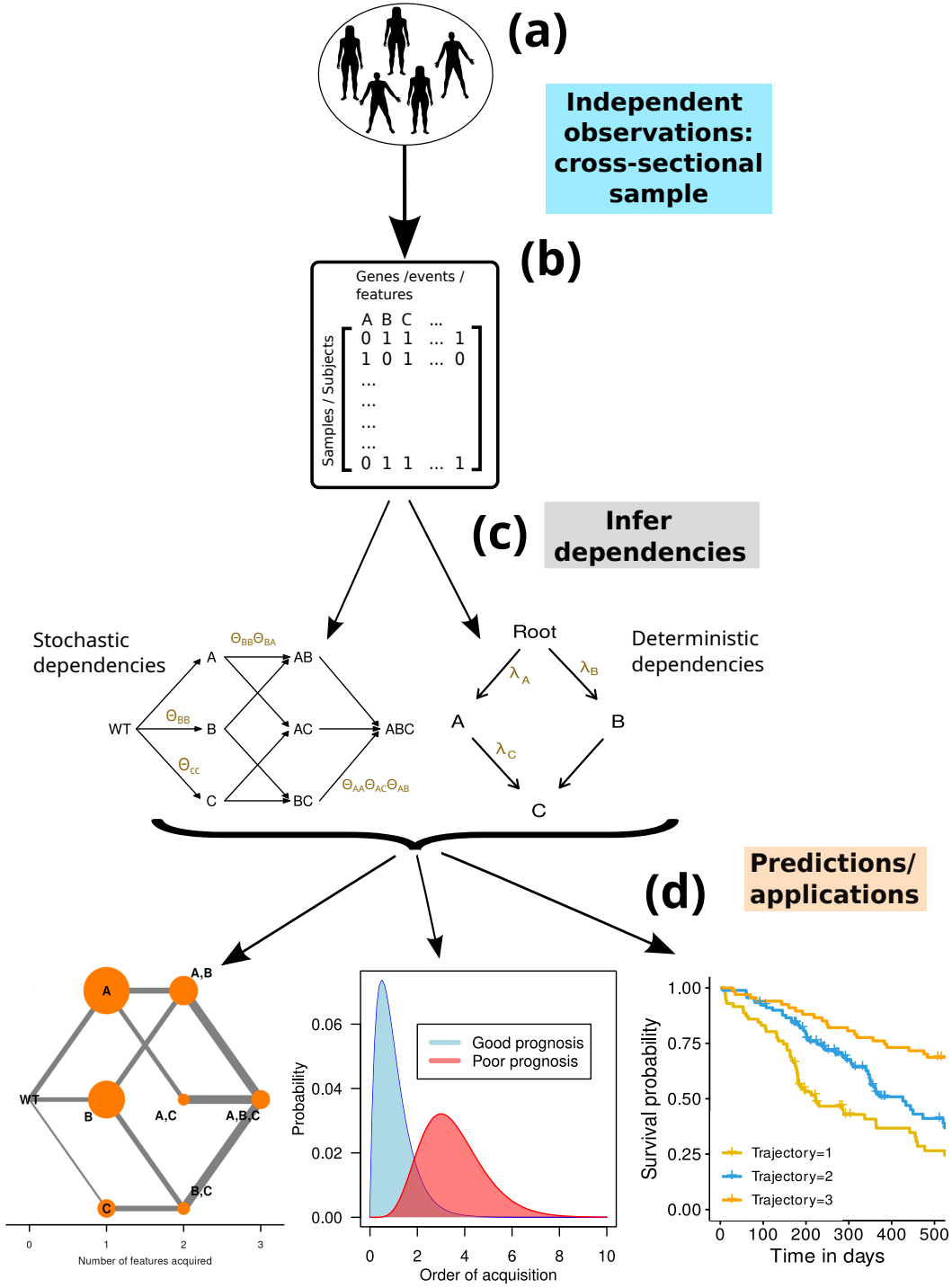
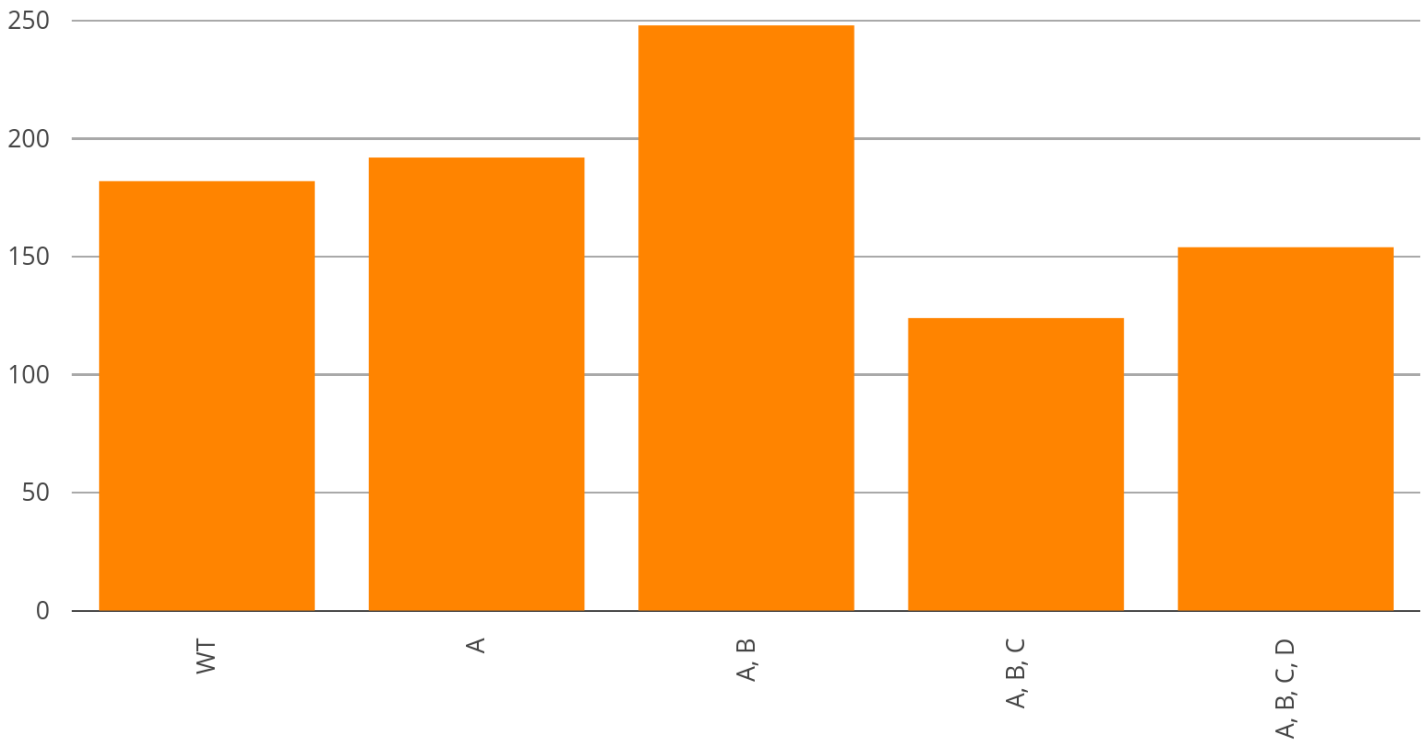
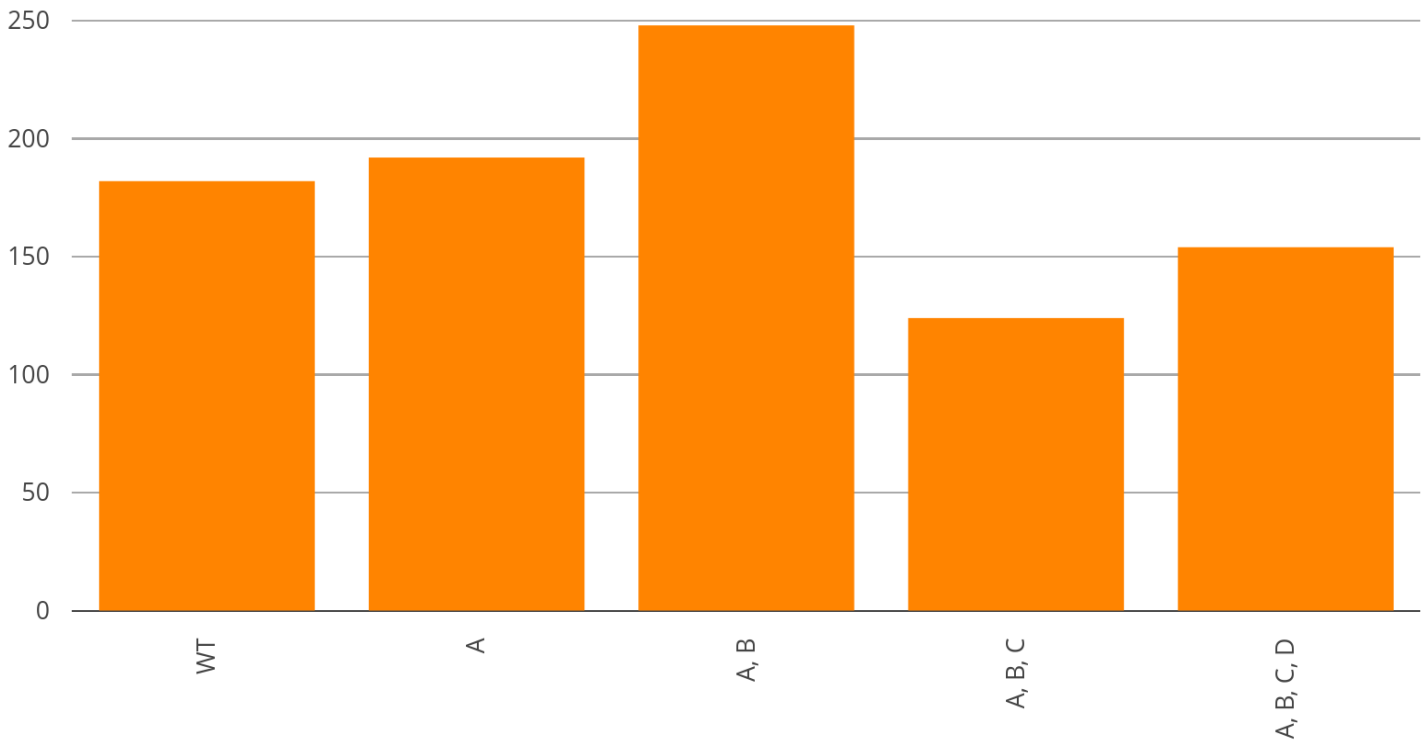


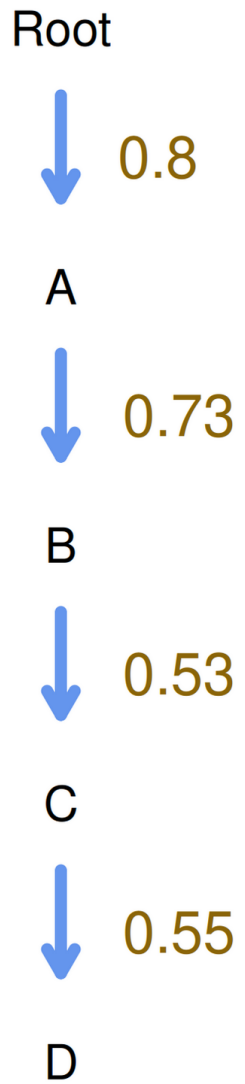
Figure from
Diaz-Uriarte & Johnston, 2025, *IEEE Access*, 13.

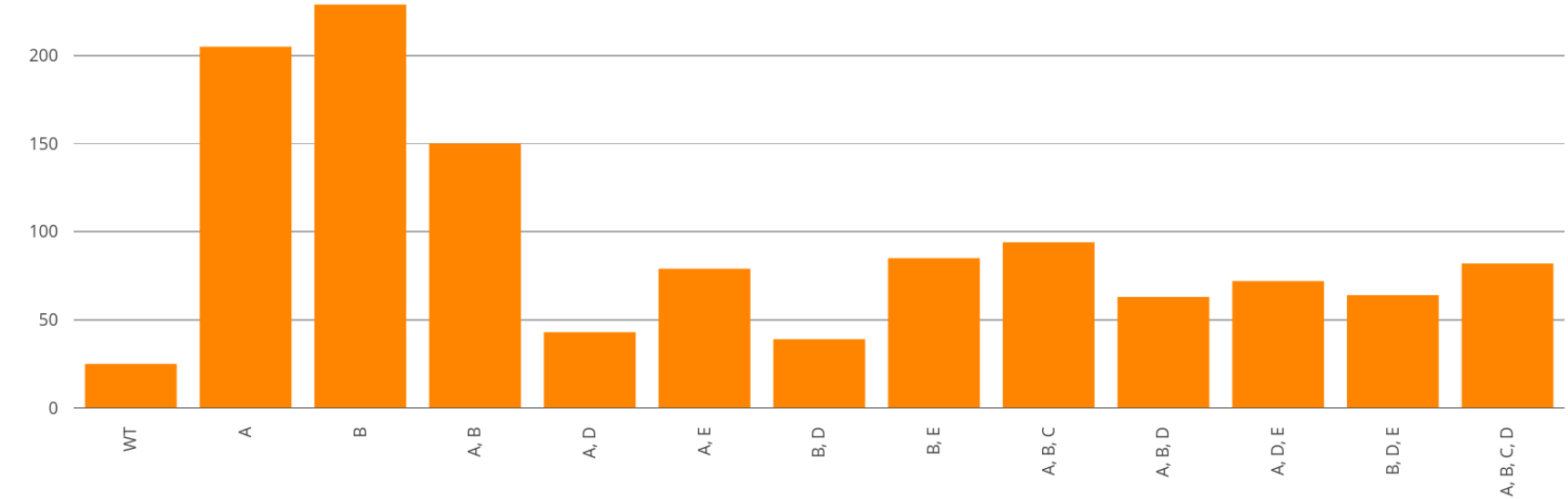
Linear dependencies

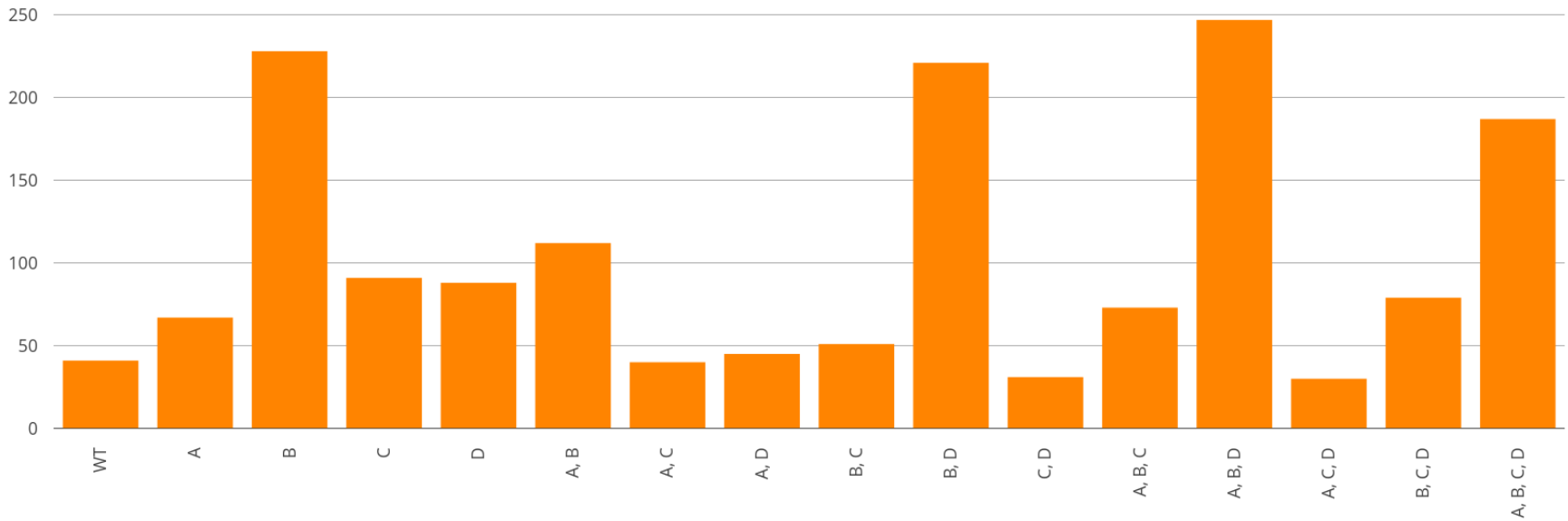


Linear dependencies





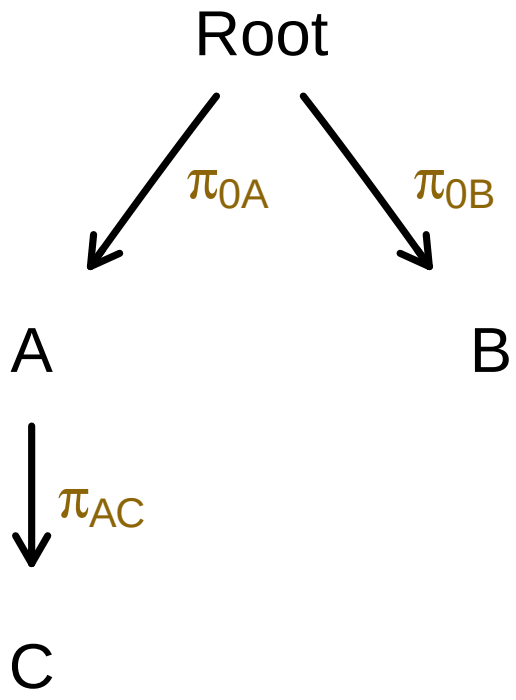




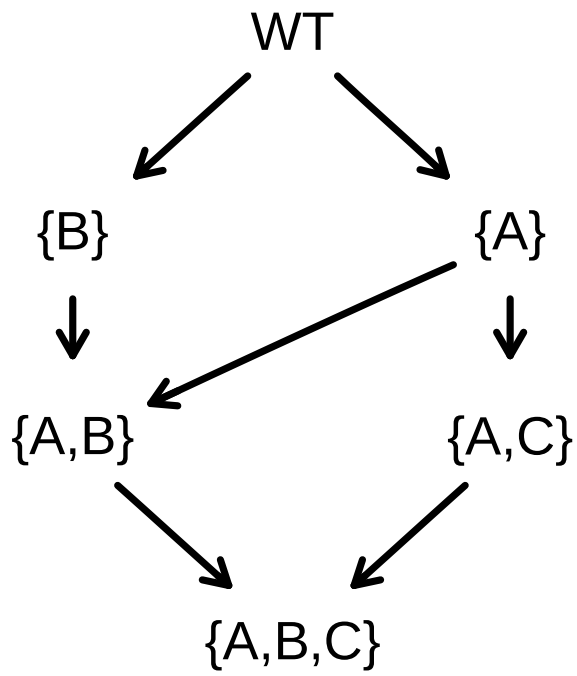
Synthetic data generated from DAG and MHN with EvAM-Tools, Diaz-Uriarte & Herrera-Nieto, 2022, *Bioinformatics*, 38.

Oncogenetic Trees (OT)

Tree of gene restrictions



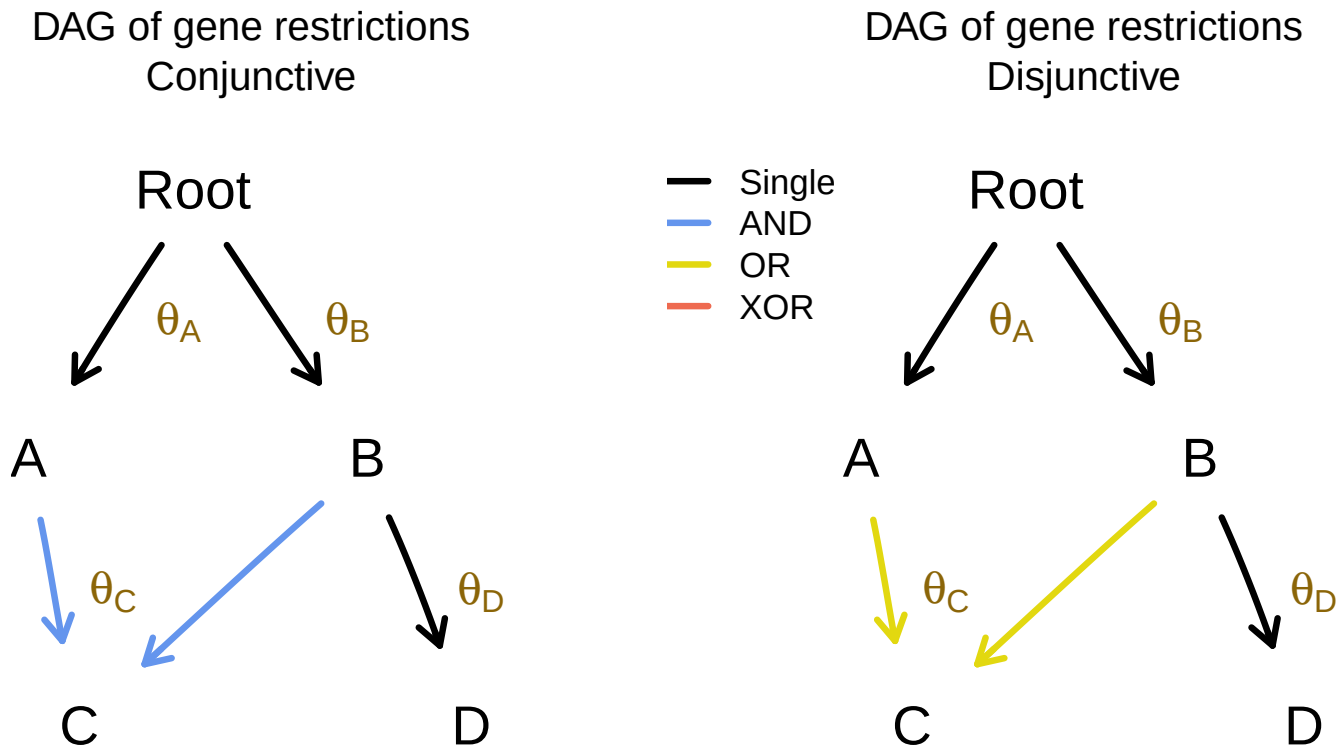
Mutational path
or fitness graph



$$P(\{A\}) = \pi_{0A}\,(1-\pi_{0B})\,(1-\pi_{AC}); \ldots;$$
$$P(\{A,C\}) = \pi_{0A}\,(1-\pi_{0B})\,\pi_{AC}; \ldots$$

Desper et al., 1999, *Journal of computational biology*; Szabo and Boucher, 2008, *In* Handbook of Cancer Models with Applications.

Disjunctive Bayesian Networks (OncoBN)

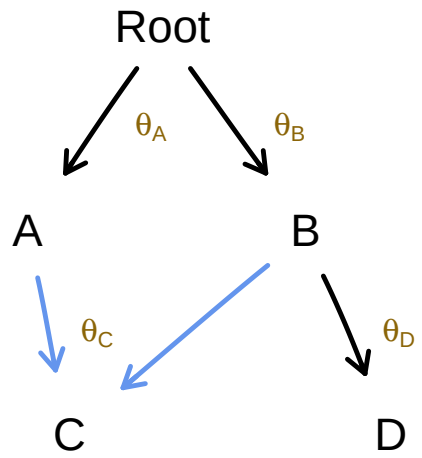


$$P(\{A,B\}) = \theta_A\,\theta_B\,(1-\theta_C)\,(1-\theta_D);$$

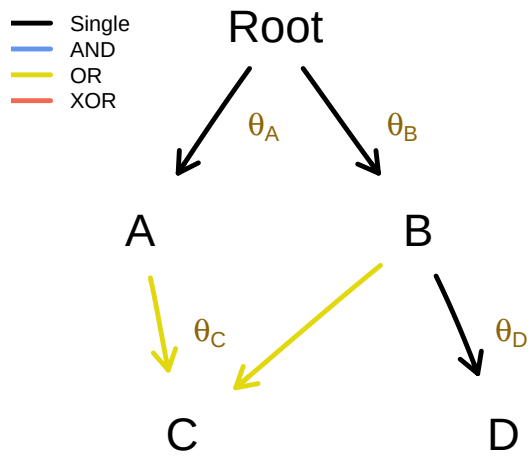
Conj.: $P(\{A\}) = \theta_A\,(1-\theta_B);$ $P(\{A,C\}) = 0$

Disj.: $P(\{A\}) = \theta_A\,(1-\theta_B)\,(1-\theta_C);$ $P(\{A,C\}) = \theta_A\,(1-\theta_B)\,\theta_C$

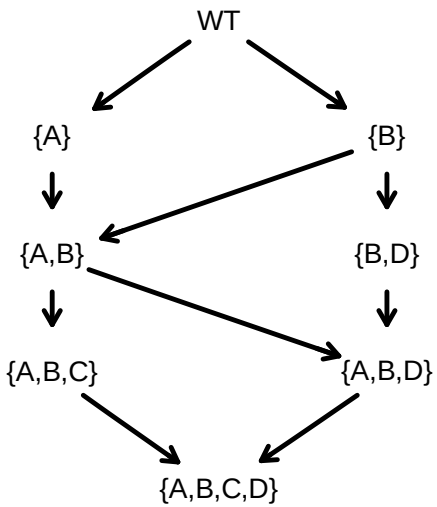
DAG of gene (event) restrictions
Conjunctive



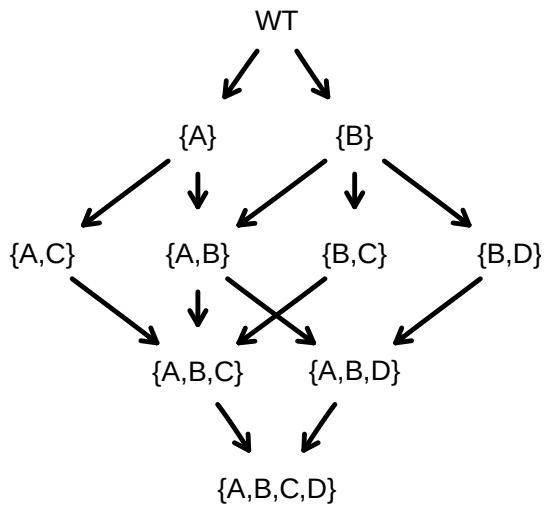
DAG of gene (event) restrictions
Disjunctive



Mutational path



Mutational path



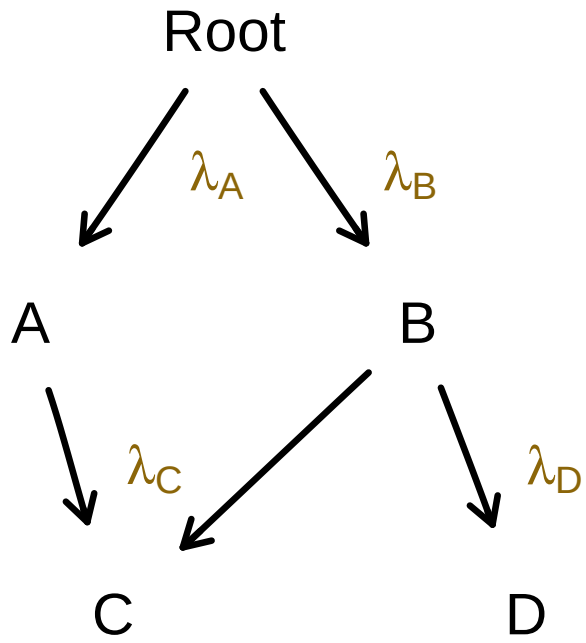
$P(\{A,B\}) = \theta_A\,\theta_B\,(1-\theta_C)\,(1-\theta_D)$;

Conj.: $P(\{A\}) = \theta_A\,(1-\theta_B)$; $P(\{A,C\}) = 0$

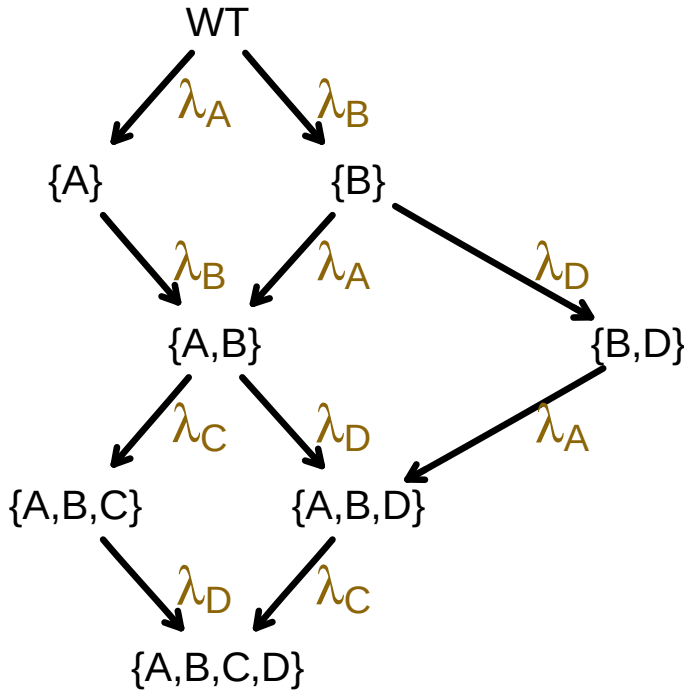
Disj.: $P(\{A\}) = \theta_A\,(1-\theta_B)\,(1-\theta_C)$; $P(\{A,C\}) = \theta_A\,(1-\theta_B)\,\theta_C$

Conjunctive Bayesian Networks (CBN)

DAG of gene restrictions



Mutational path



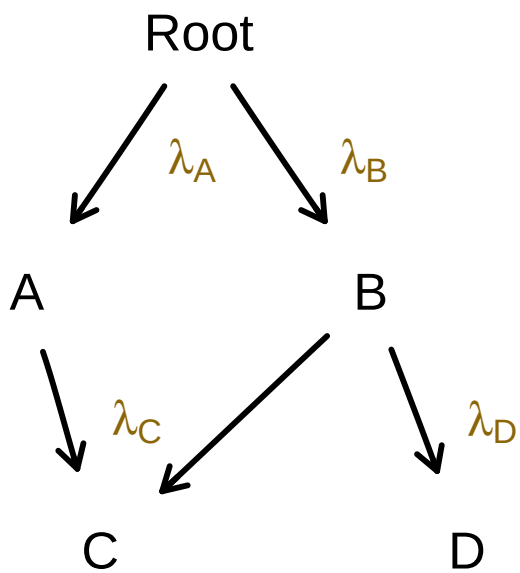
$$T_A \sim \text{Exp}(\lambda_A)$$
$$T_B \sim \text{Exp}(\lambda_B)$$
$$T_D \sim T_B + \text{Exp}(\lambda_D)$$
$$T_C \sim \max(T_A, T_B) + \text{Exp}(\lambda_C)$$

Gerstung et al., 2009, *Bioinformatics*, 25.

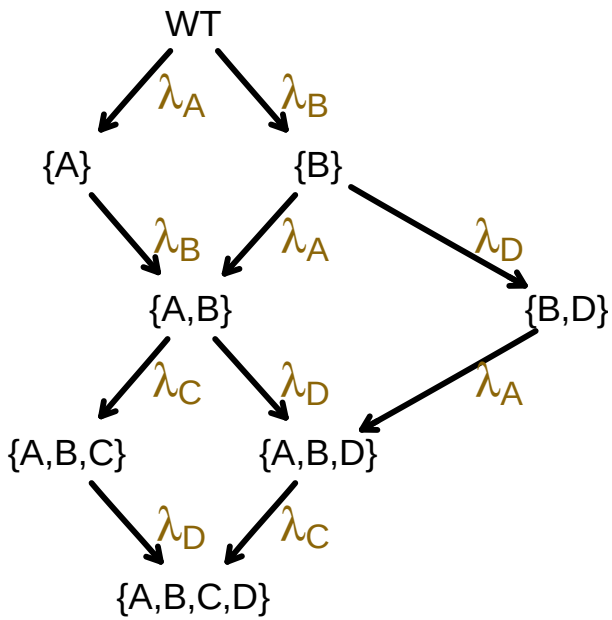
Conjunctive Bayesian Networks (CBN)

$$Q = \begin{array}{c} \\ \{WT\} \\ \{A\} \\ \{B\} \\ \{A,B\} \\ \{B,D\} \\ \{A,B,C\} \\ \{A,B,D\} \\ \{A,B,C,D\} \end{array} \begin{array}{c} \begin{array}{cccccccc} \{WT\} & \{A\} & \{B\} & \{A,B\} & \{B,D\} & \{A,B,C\} & \{A,B,D\} & \{A,B,C,D\} \end{array} \\ \begin{pmatrix} -(\lambda_A+\lambda_B) & \lambda_A & \lambda_B & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_B & 0 & \lambda_B & 0 & 0 & 0 & 0 \\ 0 & 0 & -(\lambda_A+\lambda_D) & \lambda_A & \lambda_D & 0 & 0 & 0 \\ 0 & 0 & 0 & -(\lambda_C+\lambda_D) & 0 & \lambda_C & \lambda_D & 0 \\ 0 & 0 & 0 & 0 & -\lambda_A & 0 & \lambda_A & 0 \\ 0 & 0 & 0 & 0 & 0 & -\lambda_D & 0 & \lambda_D \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_C & \lambda_C \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

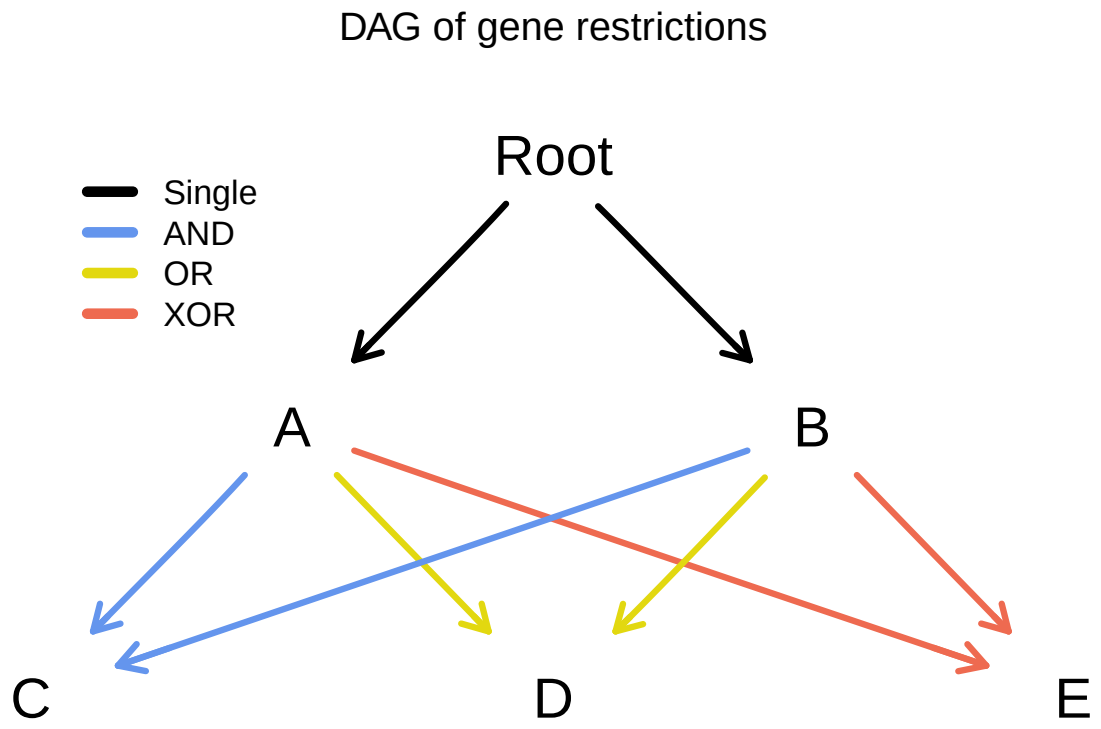
DAG of gene restrictions



Mutational path



H-ESBCN: “CBN with OR and XOR”



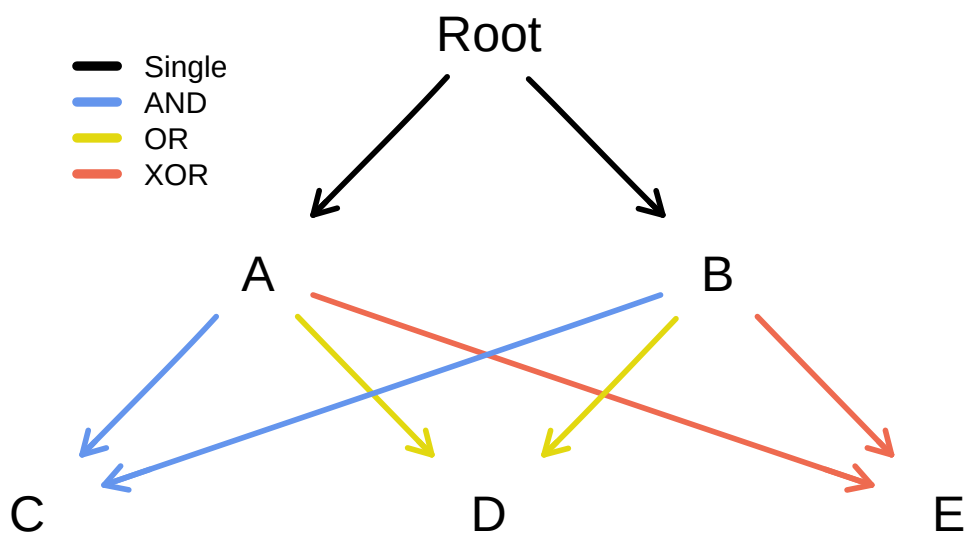
- Some possible genotypes/mutational profiles:
  - {A, B}, {A, B, C}, {A, D}, {B, D}, {A, E} , {B, E}
- Some impossible genotypes/mutational profiles:
  - {A, C}, {A, B, E}, {A, B, D, E}

Angaroni et al., 2021, *Bioinformatics*, 38.

H-ESBCN: transition rate matrix, Q

	{WT}	{A}	{B}	{A,B}	{A,D}	{A,E}	{B,D}	{B,E}	{A,B,C}	{A,B,D}
{WT}		λ_A	λ_B							
{A}				λ_B	λ_D	λ_E				
{B}				λ_A			λ_D	λ_E		
{A,B}									λ_C	λ_D
{A,D}										λ_B
{A,E}										
{B,D}										λ_A
{B,E}										
{A,B,C}										
{A,B,D}										
{A,D,E}										
{B,D,E}										
{A,B,C,D}										

DAG of gene restrictions

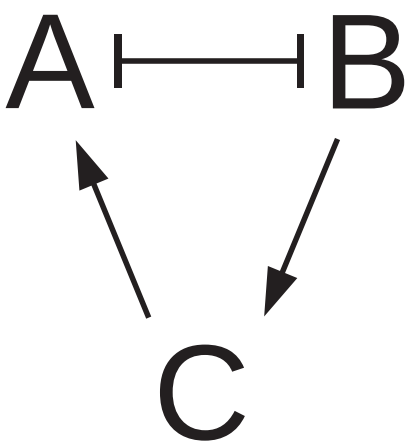


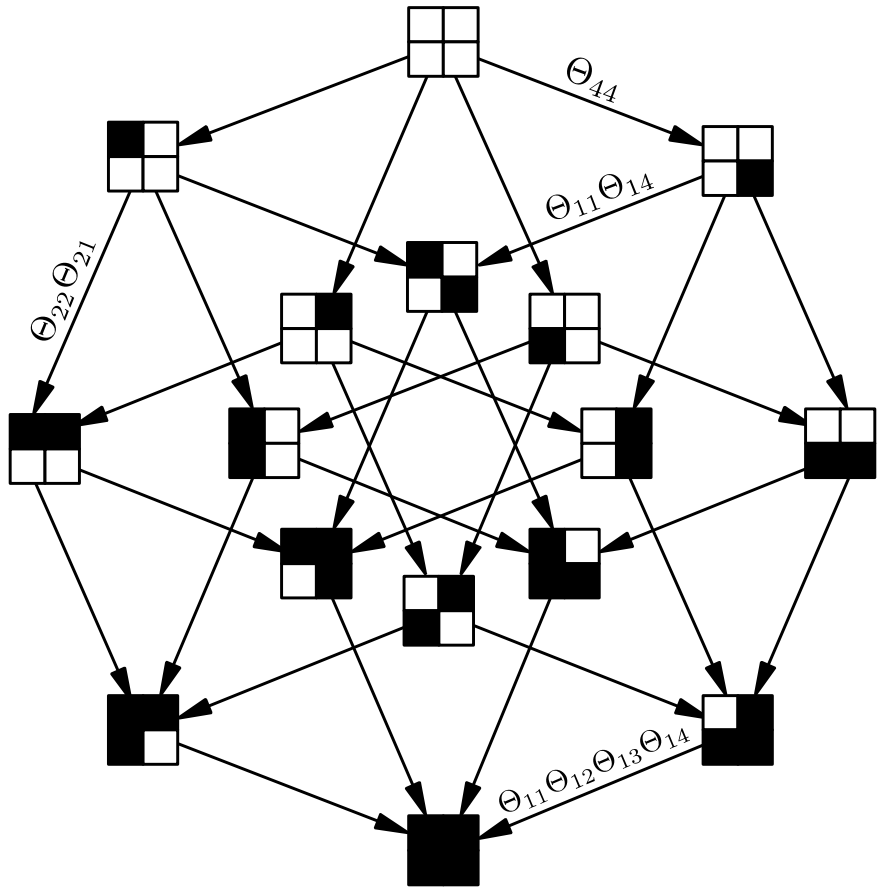
MHN (Mutual Hazard Networks)

$$Q_{\mathbf{x},\mathbf{x}_{+i}} = \exp(\theta_{ii} + \sum_{j=1}^{n} \theta_{ij} x_j) = \Theta_{ii} \prod_{x_j=1} \Theta_{ij}$$

e.g.: $AB \rightarrow ABC = \Theta_{CC} \quad \Theta_{CA}\,\Theta_{CB}$

- Asymmetric θ_{ij}





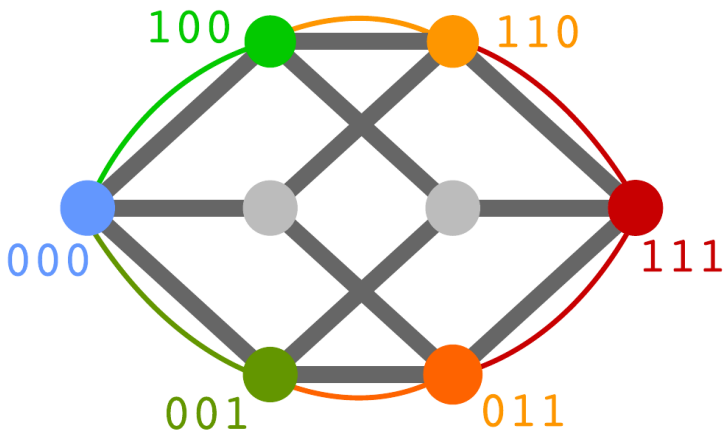
Schill et al., 2020, *Bioinformatics*, 36.
Schill et al., 2024, *J Comput Biol*, 31.

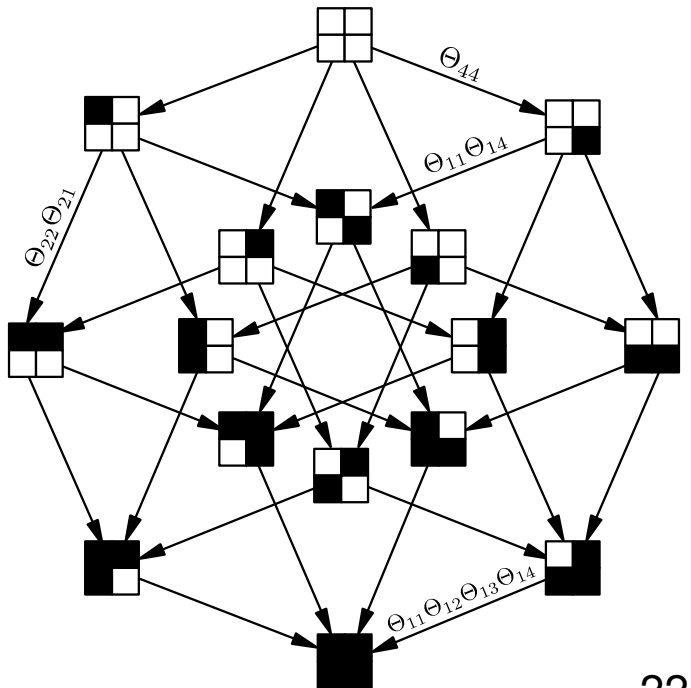
HyperTraPS/-CT (Hyperc. Transition Path Sampling [in Cont. Time])

$$P_{\text{gain } i|\text{state } s} \propto \exp(\theta_{ii} + \sum_{j=1}^{n} \theta_{ij} s_j)$$

e.g.: $AB \rightarrow ABC \propto \theta_{CC} \quad \theta_{CA}\, \theta_{CB}$

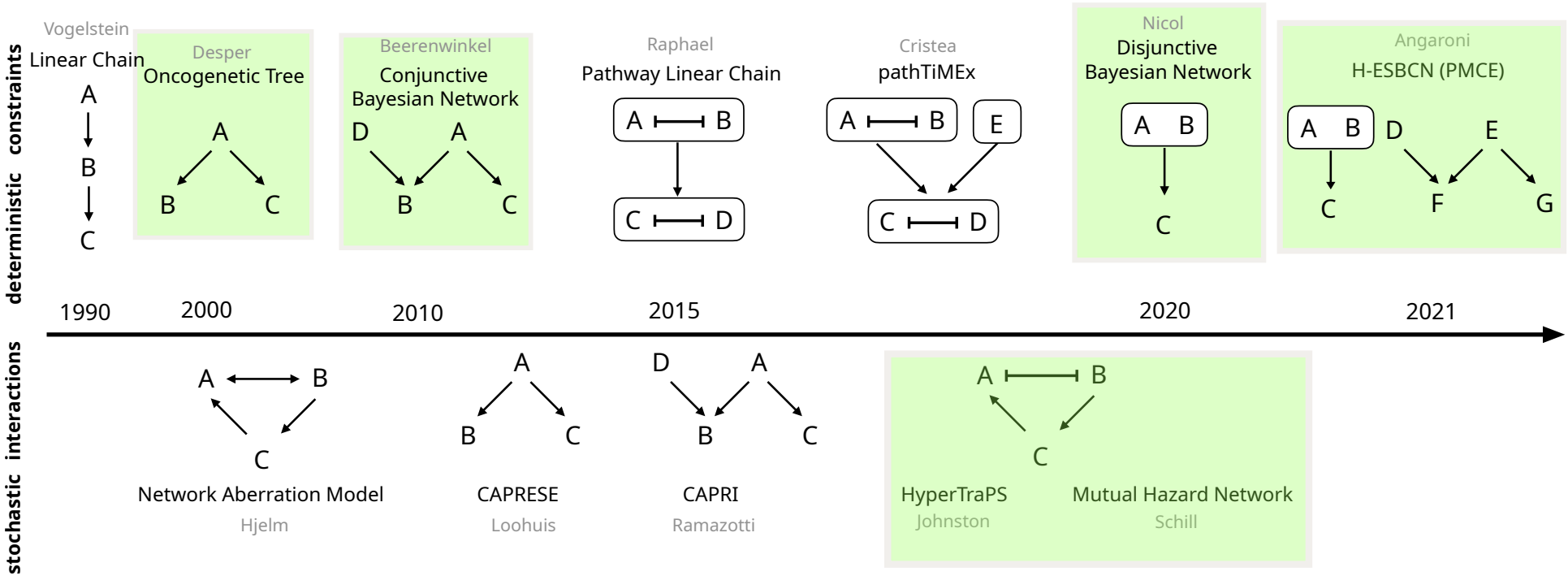
- Asymmetric θ_{ij}
- Not restricted to L^2 : main effects, two-way (L^2), three-way, four-way, unrestricted.





HyperTraPS: Greenbury et al., 2020, *Cell Systems*, 10.
HyperTraPS-CT: Aga et al., 2024, *PLOS Comp Biol.*, 20.

Timeline & Zoo of Cancer Progression models



2024, 2025:

- Schill et al., 2024. oMHN. *J Comput Biol*, 31.
- Aga et al. 2024, HyperTraPS-CT. *PLOS Comp Biol*, 20.
- Johnston & Diaz-Uriarte, 2025. Hyper-Mk. *Bioinformatics*, 41.

Figure modified from R. Schill talk at WHAM, Solstrand, Norway, November 2024

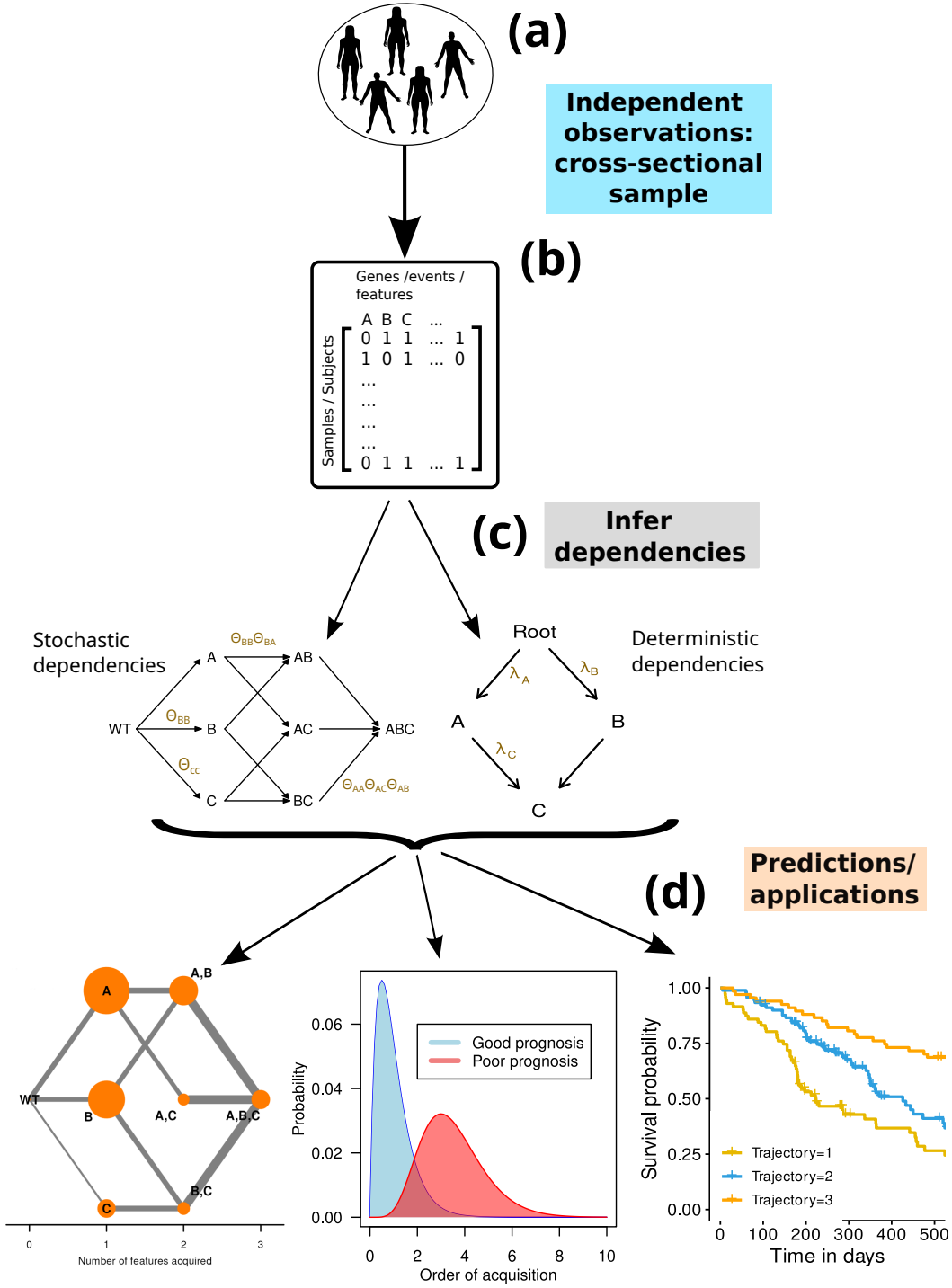


Figure from
Diaz-Uriarte & Johnston, 2025, *IEEE Access*, 13.

Uses and examples

EvAM-Tools

Examples (data)

- Descriptive modelling of tumour progression (many)
- Mutation acquisition in HIV (Beerenwinkel et al., 2006, 2007; Posada-Céspedes et al., 2021; Montazeri et al., 2015)

Examples (data)

- Descriptive modelling of tumour progression (many)
- Mutation acquisition in HIV (Beerenwinkel et al., 2006, 2007; Posada-Céspedes et al., 2021; Montazeri et al., 2015)
- Malaria progression in children (Johnston et al., 2019)

Examples (data)

- Descriptive modelling of tumour progression (many)
- Mutation acquisition in HIV (Beerenwinkel et al., 2006, 2007; Posada-Céspedes et al., 2021; Montazeri et al., 2015)
- Malaria progression in children (Johnston et al., 2019)
- Mitochondrial genome reduction (Johnston & Williams, 2016)
- Drug resistance TB (Greenbury et al., 2020)
- Animal tool use acquisition (Johnston & Røyrvik, 2020)
- Learner behaviour in online courses (Peach et al., 2004)
- (Last four: using data with phylogenetic or longitudinal structure)

Uses (questions)

- Descriptive modelling of tumour progression (many)

Colorectal cancer

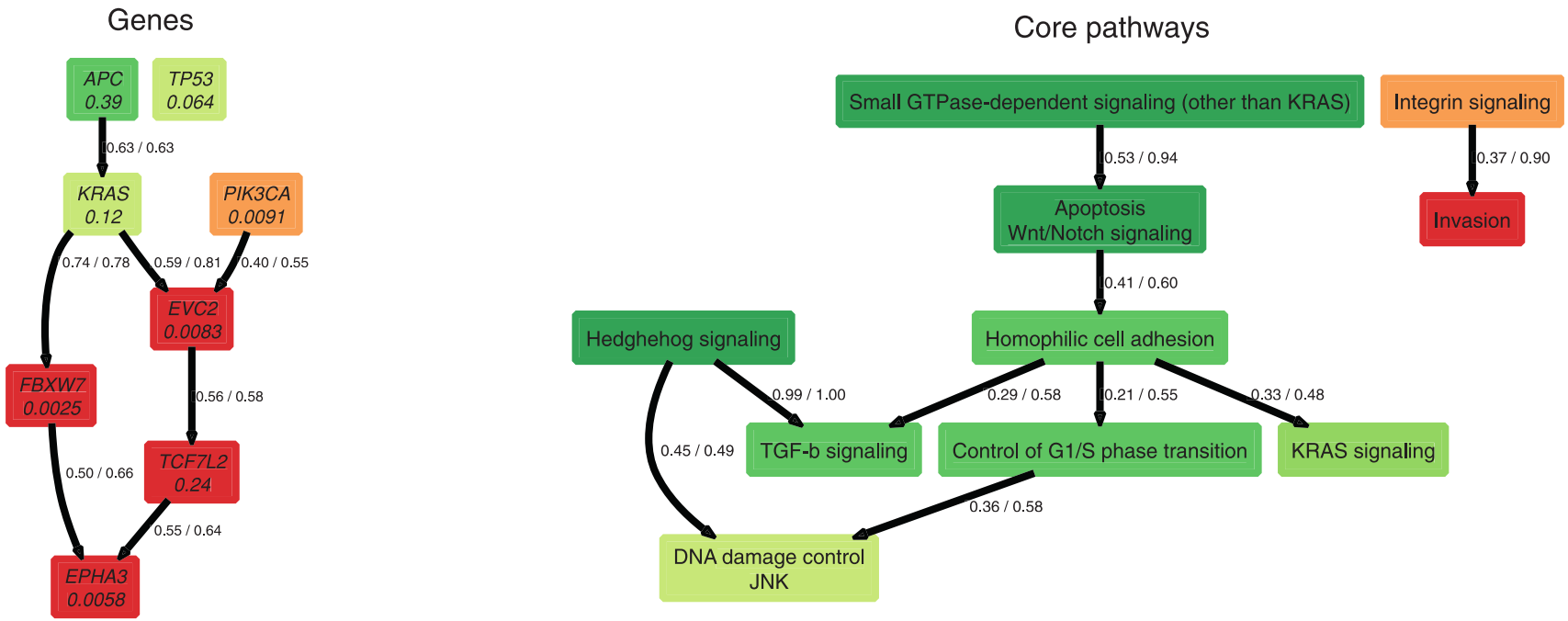


Figure from Gerstung et al., 2011, *PLoS ONE*, 6

- Stratification; relationship between progression (as given by model) and survival (e.g., Bogojeska et al., 2008, *BMC Bioinformatics*, 9; Gerstung et al., 2009, *Bioinformatics*, 25; Angaroni et al., 2021, *Bioinformatics*, 38; Fontana et al., 2023, *Nature Comm.*, 14)

- Stratification; relationship between progression (as given by model) and survival (e.g., Bogojeska et al., 2008, *BMC Bioinformatics*, 9; Gerstung et al., 2009, *Bioinformatics*, 25; Angaroni et al., 2021, *Bioinformatics*, 38; Fontana et al., 2023, *Nature Comm.*, 14)

- Tumor evolution:
  - Estimate predictability (Hosseini et al., 2019, *Bioinformatics*, 35)
  - Predict paths of tumour progression (Diaz-Uriarte & Vasallo, 2019, *PLOS Comp Biol*, 15)
  - Conditional prediction: what genotype comes next (Diaz-Colunga & Diaz-Uriarte, 2021, *PLOS Comp Biol*, 17)

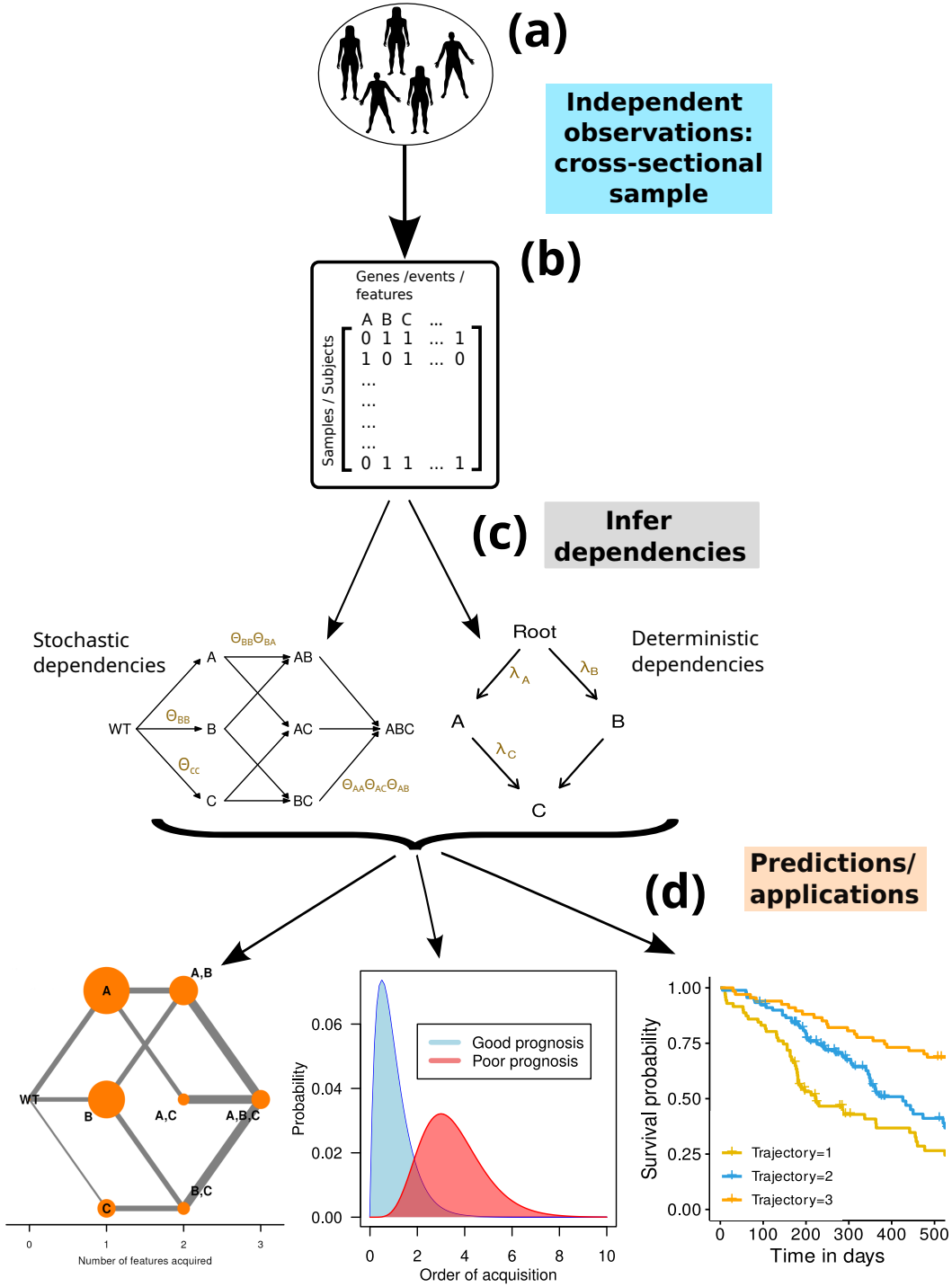


Figure from
Diaz-Uriarte & Johnston, 2025, *IEEE Access*, 13.

Cancer progression models / Evolutionary accumulation models

Uses and examples

EvAM-Tools

EvAM-Tools: what

EvAM-Tools: use and installation

Examples using EvAM-Tools

Cancer progression models / Evolutionary accumulation models

Uses and examples

EvAM-Tools

EvAM-Tools: what

EvAM-Tools: use and installation

Examples using EvAM-Tools

EvAM-Tools: what

- Unified interface to state-of-the art methods for evolutionary accumulation models (EvAMs) and cancer progression models (CPMs).

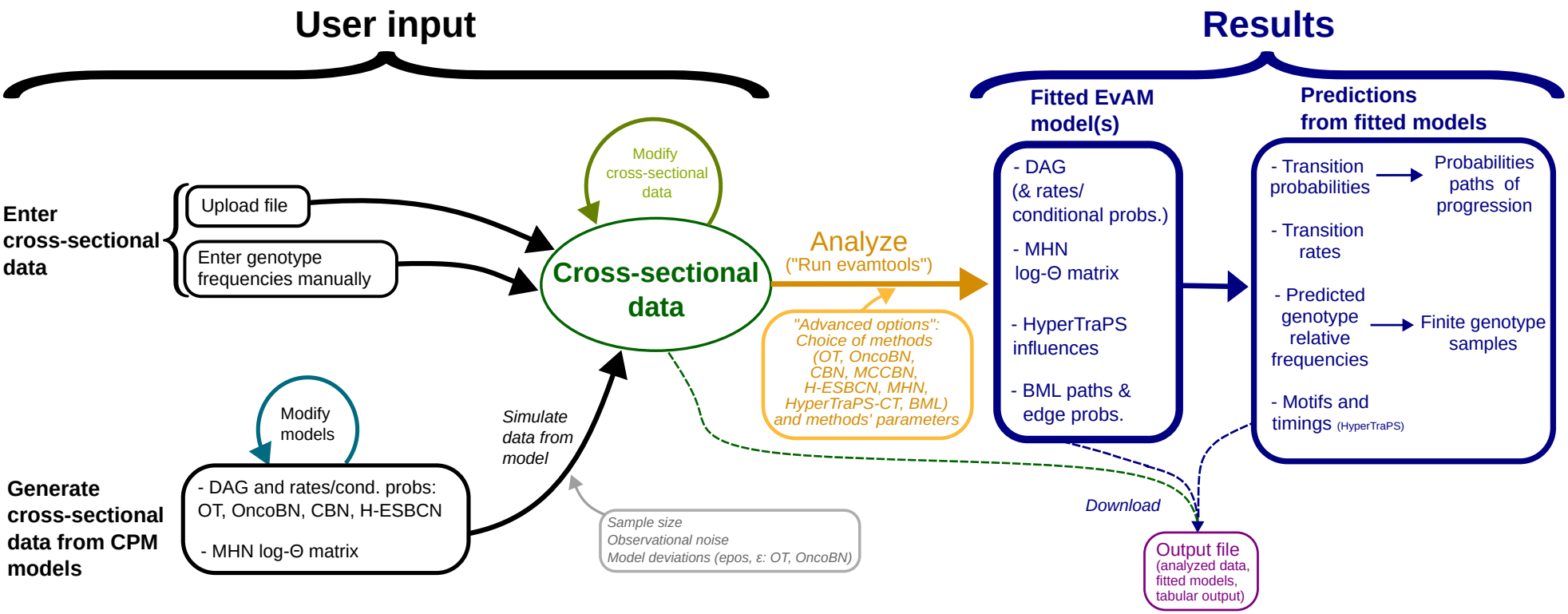
- R package and Shiny web app, also as Docker containers.

- `https://github.com/rdiaz02/EvAM-Tools`

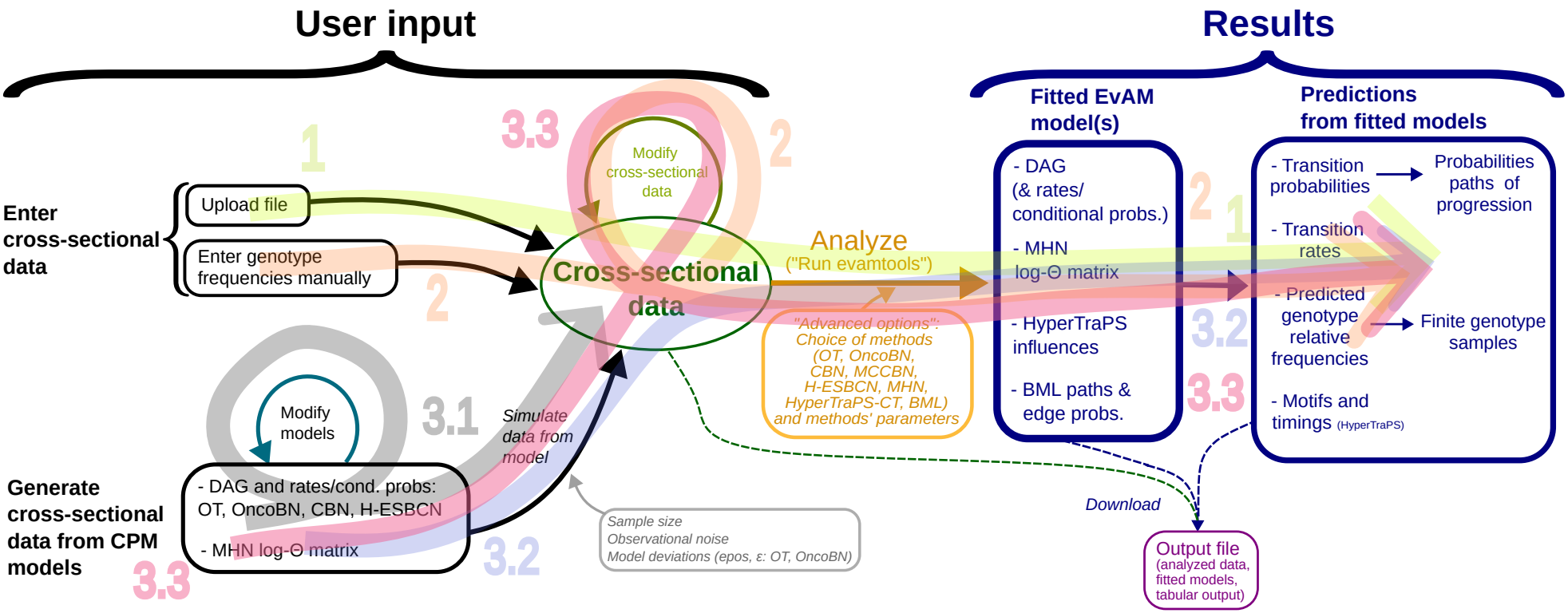
EvAM-Tools: what

- Functionality and workflows:
  1. Inference from user data (*What do my data say?*)
  2. Construction of synthetic data and inference from them (*Is it just sample size? What if there were no WT?*)
  3. Simulation of data from models and inference from them (*What are the implications of different model parameters for how data look? How does noise or sample size affect recovering the truth? How do stochastic/deterministic models work if data come from a deterministic/deterministic model? What happens when there are XORs?*)
  4. (Simulation of models themselves)

EvAM-Tools: functionality



EvAM-Tools: workflows



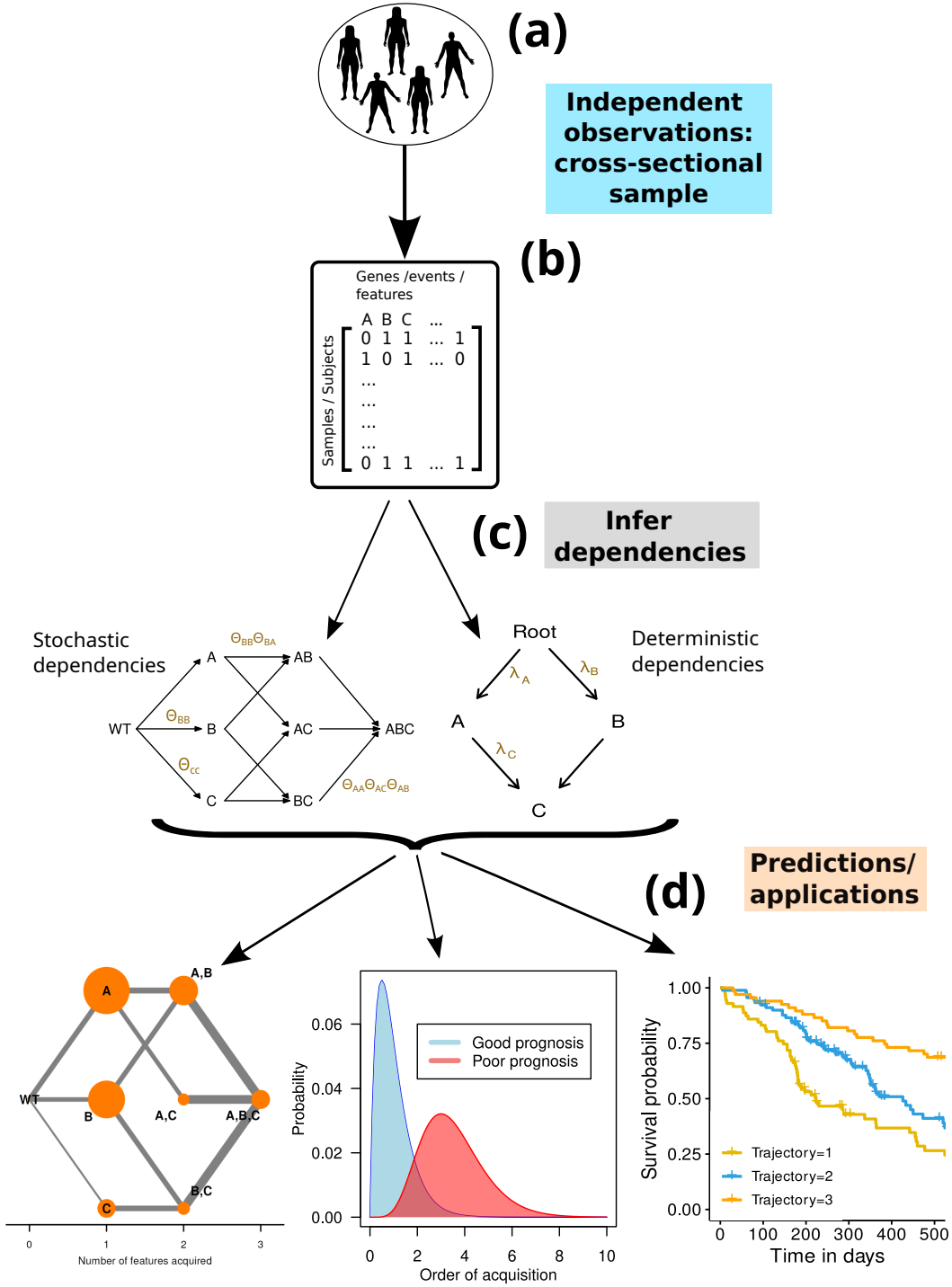


Figure from
Diaz-Uriarte & Johnston, 2025, *IEEE Access*, 13.

Use and installation

1. Install the R package: use as a regular R package and/or launch a Shiny app from it

2. Use the web app on our servers

3. Use the Docker images

Use and installation

1. Install the R package: use as a regular R package and/or launch a Shiny app from it
    - Dependencies (R packages, some from github; C/C++ compilers; eventually also Python and libraries)
2. Use the web app on our servers
    - Easiest
    - Not all functionality available (simulate models)
    - Limits: RAM and time (1.5 h)
3. Use the Docker images
    - Containers with R package or Shiny app
    - If you want flexibility, but want to avoid package installation: this is the best choice, especially in Windows

1. Install the R package: use as a regular R package and/or launch a Shiny app from it
    - Go to `https://github.com/rdiaz02/EvAM-Tools`
    - Dependencies (R packages, some from github; C/C++ compilers; eventually also Python and libraries)

2. Use the web app on our servers: `https://www.iib.uam.es/evamtools/`
    - Not all functionality available (simulate models)
    - Limits: RAM and time (1.5 h)
    - Easiest

3. Docker

- If you want flexibility, change limits (RAM, CPU, storage), and want to avoid installation: this is the best choice, especially under Windows
    - Shiny app
    - R package (from which the Shiny app can be launched)

3. Docker

- If you want flexibility, change limits (RAM, CPU, storage), and want to avoid installation: this is the best choice, especially under Windows
    - Shiny app
    - R package (from which the Shiny app can be launched)

- You must install Docker

3. Docker

- If you want flexibility, change limits (RAM, CPU, storage), and want to avoid installation: this is the best choice, especially under Windows
  - Shiny app
  - R package (from which the Shiny app can be launched)
- You must install Docker
- (Linux: You probably want rootless mode: `https://docs.docker.com/engine/security/rootless/`)

Docker: steps

1. Install Docker; **only once**, or on updates): `https://docs.docker.com/get-started/get-docker/`

2. Pull images of EvAM-Tools you want (R package or Shiny app): **only once** (or when images change)

3. Use it
   - 3.1 Start Docker
   - 3.2 Run the container(s) you want: R package (maybe with Shiny app), or Shiny app directly
   - 3.3 Do stuff . . .
   - 3.4 Close Docker

(My setup: installing, starting, stopping)

Just for completeness. Yours can differ and that is OK

- Docker installation notes
  - I am running Debian GNU/Linux
  - For installation, I follow Docker's pages, `https://docs.docker.com/engine/install/debian/`, installing from Docker's apt repositories)
  - I am using Docker Engine —`https://docs.docker.com/engine/install/`—, not Docker Desktop —`https://docs.docker.com/desktop/`
- Starting Docker: `systemctl --user start docker` (might differ by OS or might be started automatically on login)
- When done I like to shutdown Docker: `systemctl --user stop docker` (might differ by OS or how you run Docker)

Pulling images

Only the first time, or when images change: Get the images you want from Docker hub (either or both):

- Shiny app directly: `docker pull rdiaz02/evamshiny:latest`
- R package (+ Shiny app): `docker pull rdiaz02/evamrstudio:latest`

Running the Shiny app container

1. `docker run -d -p 4080:3000 --memory="2g" --name E1 rdiaz02/evamshiny`
   - (We map port 3000 of the container to port 4080 of the host; if you want to use port 80, write 80 instead of 4080.)
   - ("E1": a name to make other operations, like stopping the container, simpler. Use whatever you prefer)
   - (Here we also limit the maximum memory to 2 GB.)
2. Open a browser to `localhost:4080`
3. Do stuff . . .
4. (When done, shutdown Docker. If you want to continue using Docker, you can simply stop and remove the container: `docker stop E1 && docker rm E1`)

Running the R package container

1. `docker run --rm -p 8787:8787 -e PASSWORD=somerandompasswordhere --name EP1 rdiaz02/evamrstudio`
2. Open a web browser to `localhost:8787`
3. Log in. Username `rstudio` or, if rootless mode, `root`. Password: the one you set above.
4. Use R (and RStudio) as you usually do.
5. Load the package: `library(evamtools)`.
6. You can launch the Shiny app (`runShiny()`)
7. Do stuff . . .
8. (Shutdown Docker and/or stop and remove the container: `docker stop EP1 && docker rm EP1`)

Examples using EvAM-Tools

(I'll use any one of the procedures for running EvAM-Tools mentioned before.)

Things to try

- Load some (CSV) data (e.g., from `https://github.com/rdiaz02/EvAM-Tools/tree/main/examples_for_upload`)
- Modify some counts
- Analyze, possibly changing options and methods used
- Generate data from DAGs and MHN
- Modify DAGs (and try to do things that should not be possible, and see errors) and simulate data from them; add different levels of noise
- Using the same DAG, change models (e.g., H-ESBCN to OncoBN)
- Try to create impossible DAGs (e.g., OncoBN with XOR)
- Simulate from MHN
- Change some "Advanced options", including settings that are not possible (e.g., `samplegap` > `chain length` for HyperTraPS)
- Select and deselect methods shown
- Change the number of paths shown in the results
- Obtain samples from the predictions
- Try to make sense of the results ☺
- Etc