Predicting anticancer hyperfoods with graph convolutional networks

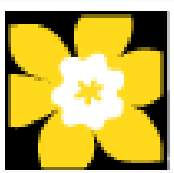
Gonzalez et al. 2021

Betty Shea

25 August 2021

MLRG

Bad news



Canadian Cancer Society

- Nearly 1 in 2 Canadians (45% of men and 43% of women) is expected to develop cancer during their lifetime.
- About 1 out of 4 Canadians (26% of men and 23% of women) is expected to die from cancer.

---

https://action.cancer.ca/en/research/cancer-statistics/cancer-statistics-at-a-glance

Good news

Partially under our control: "modifying dietary and lifestyle factors alone can prevent between 30 and 40% of all cancer cases". Some factors:[1]

- glucose metabolism,
- omega 3:6 ratio imbalance,
- fruits and vegetables, etc ....

[1] Michael S. Donaldson. Nutrition and cancer: A review of the evidence for an anti-cancer diet

Food vs cancer

MEDITERRANEAN DIET NAMED #1 BEST DIET FOR 2021

What makes a food/ drug anti-cancerous?

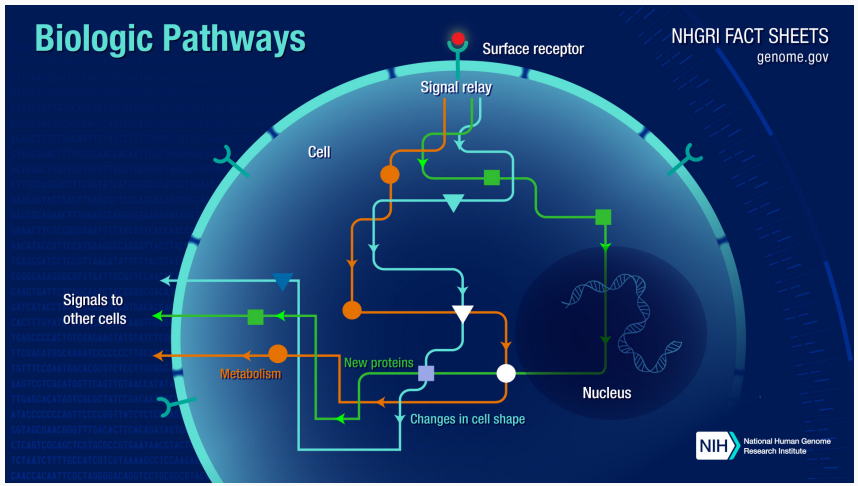
Two biomarkers:

- **Individual genes**. Look at genes that are different between normal and cancerous cells.
    - independence assumption (single genetic mutations)
- **Biological pathways**. Look at dysfunctional pathways, or how cells function with and without cancer.
    - assumes genes work together

Does a drug/ food have a positive impact on either?

Food and pathways

- a single food contains hundreds of food molecules
- food molecules regulate biological pathways, or processes, linked to reduced risk of cancer

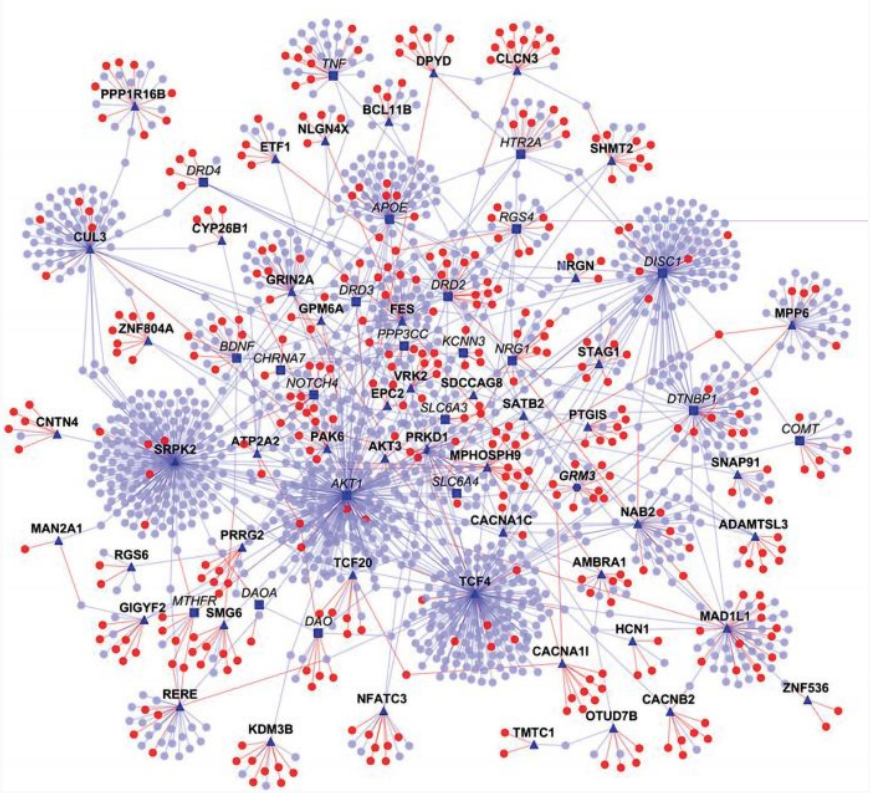


https://www.genome.gov/about-genomics/fact-sheets/Biological-Pathways-Fact-Sheet

Food and pathways

THE UNIVERSITY OF BRITISH COLUMBIA

Pathways network: protein-protein interactions (PPI)s between contact points in cells.[2]



[2] Image from Genetic Engineering & Biotechnology News:
https://www.genengnews.com/insights/protein-protein-interactions-get-a-new-groove-on/

Food and pathways

Binary classification task:

- Input is a PPI graph and vectors representing drugs/ foods (non-Euclidean domain)
- Apply deep learning
- Output is whether a drug/ food has an anticancer effect

MLRG theme: geometric structures in ML

How does this fit into this term's MLRG?

- Geometric deep learning (sub-topic 3 in Wu's list)
- Use geometric prior to avoid curse of dimensionality (Nick's talk)
- "Nonlinear dimensionality reduction": learning a low-dimensional structure within a high-dimensional Euclidean space

Overview of talk

THE UNIVERSITY OF BRITISH COLUMBIA

Main topics:

- Graph neural networks (GNN)s
- Anticancer hyperfoods

Graph neural networks (GNN)s

Convolutional neural networks (CNN)s

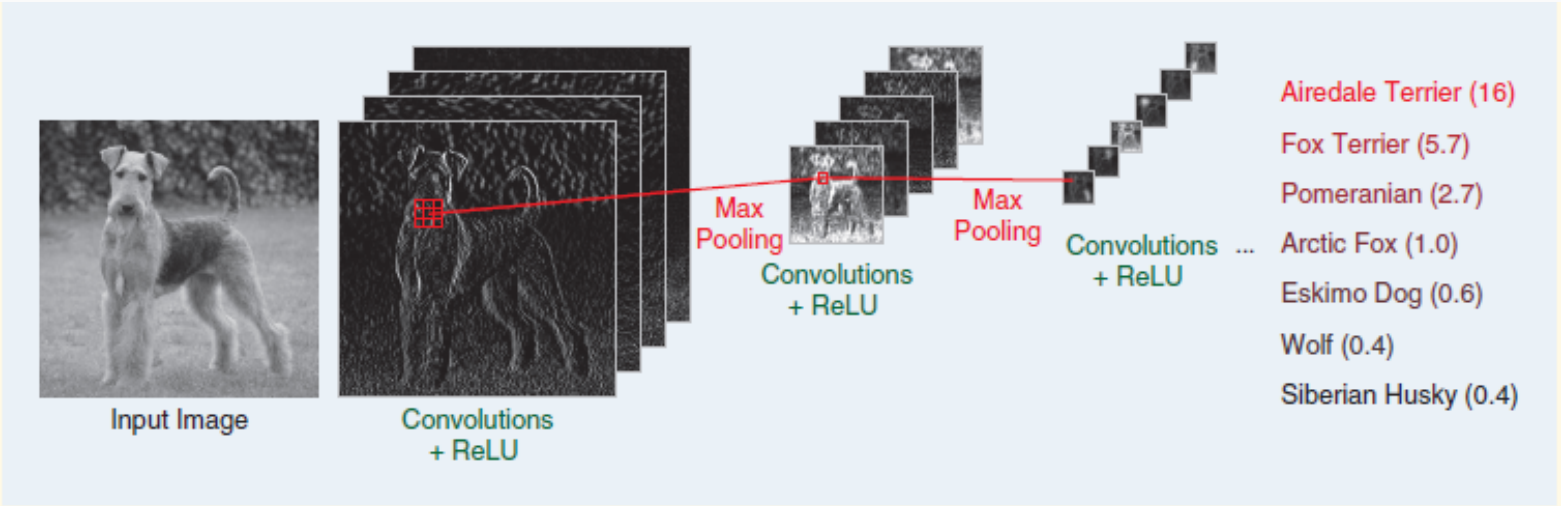


FIGURE S1. The typical CNN architecture used in computer-vision applications such as image classification.

- Convolution layers: filters
- Pooling layers: reduce dimensionality

Figure from Bronstein et al. Geometric deep learning, 2017.

Geometric priors

For images:

- Pixels have features that are equivariant to translation
- Convolutions capture this geometric prior

For biological applications:

- Some cells are related through pathways
- Graph structure captures this geometric prior

Graph neural networks (GNN)s

Generalize CNNs to non-Euclidean domains (graph)

- Euclidean functions applied to graphs, e.g. gradient, Laplacian
- Laplacian allows decomposition of a graph into a spectrum
- Allows generalization of CNN concepts, e.g. convolution, to graphs
- Spectral convolution: parameterized by eigenvalues to the Laplacian

Spectrum-free variants of GNNs

Polynomial through the Laplacian $\equiv$ polynomial through its eigenvalues.

ChebNet

- Parameterize convolution with a polynomial expansion instead

Graph convolutional network (GCN)

- Simplify to a degree-2 polynomial

Both variants apply simple filters in the spacial domain.

Anticancer hyperfoods

Summary of paper

- introduces an end-to-end model to classify drugs as anticancer or not
- encodes drugs and foods as graphs
- train model on FDA-labeled anticancer drugs
- anticancer foods have similar graphs to known FDA-approved anticancer drugs

Methodology

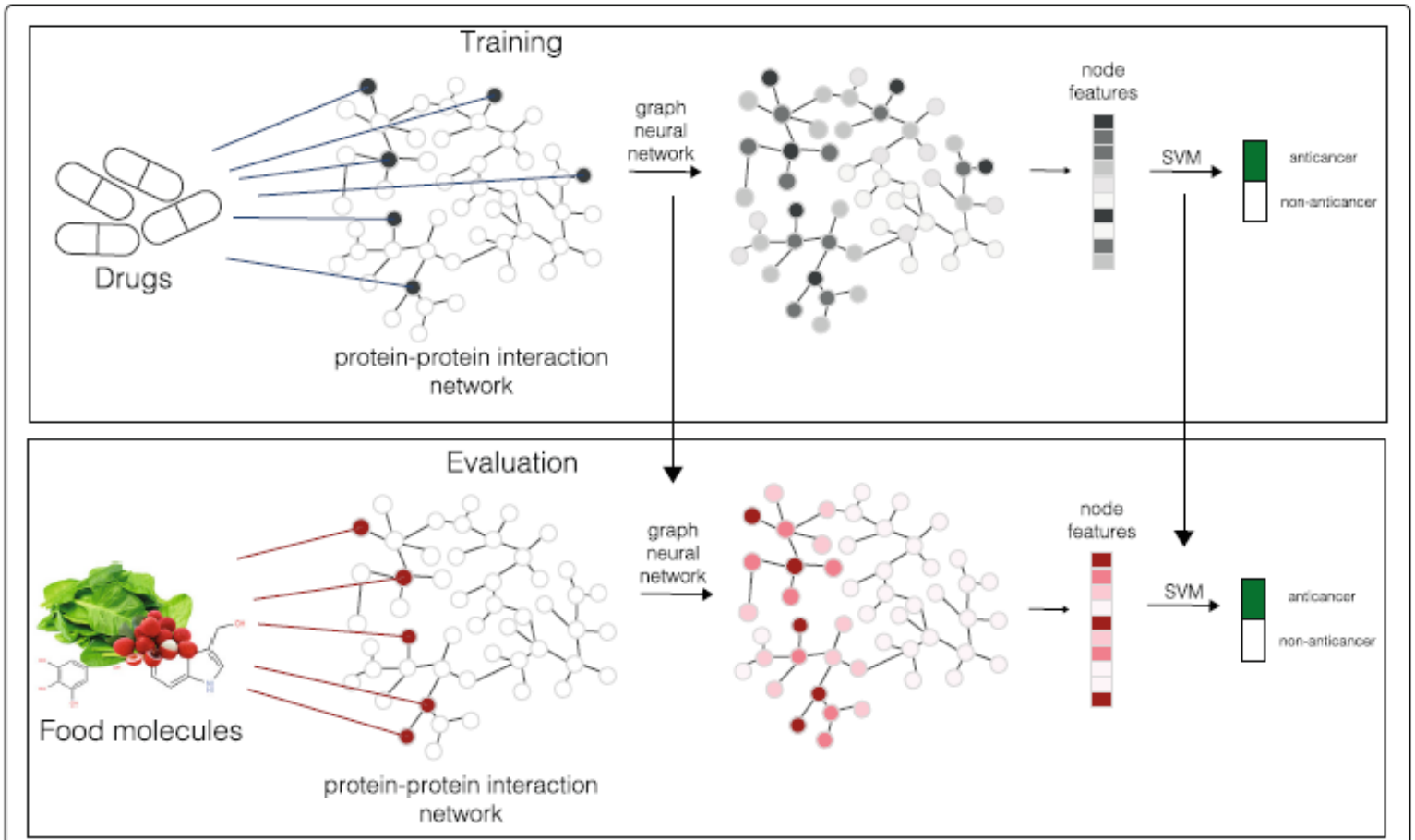


Fig. 1 Drug targets are represented as a binary signal on the PPI. We use a GNN to generate a graph embedding representing the systemic-wide effect of the drug on the PPI. We then feed this representation to an MLP for the anticancer prediction task. The model is trained in an end-to-end fashion. After model training, we feed bioactive molecules within foods to the model for the prediction of anticancer food molecules

Figure 1 from Gonzalez et al. 2021

Data

- 2,048 clinically approved drugs from DrugBank and DrugCentral (209 positive, 1,839 negative)
- 7,793 food molecules from FooDB
- pathways taken from Kyoto Encyclopedia of Genes

Graph encoding

Paper tries three variants of GNNs: GCN, ChebNet, GraphSAGE

Table 1 Hyperparameter space searched

Hyperparameter	Space search
Learning rate	$5.10^{-4}, 5.10^{-3}$
L2-regularization	$1.10^{-5}, 1.10^{-4}, 5.10^{-4}$
Number of convolutional layers	1, 2, 3
Number of dropout layers	1, 2
Batch normalization	*True, false*
Feature normalization	*True, false*
n-hops for ChebNet	2, 4, 6

Hyperparameters tuned using cross-entropy loss.

Graph encoding

For each of the three variants of GNN:

1. "Jumping knowledge network"
2. Pathway pooling

Results

THE UNIVERSITY OF BRITISH COLUMBIA

Table 3 Summary of results (%) on anticancer drug prediction

Method	ACC	F1	AUPR	Precision ac	Recall ac	Precision non-ac	Recall non-ac
SVM	79.26 ± 4.2	52.12 ± 5.92	53.35 ± 10.97	41.50 ± 6.75	69.12 ± 10.08	96.31 ± 1.06	88.74 ± 3.20
RWR + SVM	81.13 ± 3.79	51.84 ± 5.79	67.43 ± 8.14	38.98 ± 5.38	75.08 ± 6.92	96.90 ± 0.83	86.67 ± 2.37
MLP	80.62 ± 3.81	66.53 ± 5.02	69.05 ± 5.01	69.75 ± 6.74	64.55 ± 8.23	96.02 ± 0.85	96.68 ± 1.30
GCN	80.52 ± 3.33	63.95 ± 3.90	66.45 ± 5.82	63.33 ± 5.72	65.51 ± 7.42	96.08 ± 0.76	95.54 ± 1.38
GraphSAGE	78.27 ± 6.11	59.93 ± 6.53	64.42 ± 9.96	61.04 ± 5.72	61.15 ± 13.48	95.62 ± 1.37	95.38 ± 1.51
ChebNet	**83.46 ± 2.52**	**67.99 ± 2.87**	**73.91 ± 3.49**	**65.46 ± 4.53**	**71.27 ± 5.58**	**96.71 ± 0.59**	**95.65 ± 0.96**
MLP-P	76.72 ± 2.68	54.40 ± 3.56	59.79 ± 7.64	51.67 ± 11.33	60.73 ± 7.81	95.44 ± 0.72	92.72 ± 3.18
GCN-P	78.70 ± 5.36	57.43 ± 7.61	60.03 ± 8.48	52.77 ± 7.69	64.03 ± 11.05	95.83 ± 1.18	93.37 ± 1.72
GraphSAGE-P	77.09 ± 4.18	54.07 ± 4.88	60.55 ± 9.51	48.87 ± 4.06	61.64 ± 9.65	95.53 ± 0.96	92.55 ± 1.95
ChebNet-P	76.10 ± 2.67	55.71 ± 4.46	59.68 ± 9.53	53.72 ± 4.07	57.86 ± 4.96	95.17 ± 0.53	94.35 ± 0.44

ACC = balanced accuracy, *F1* = harmonic mean of precision and recall, *AUPR* = area under the precision-recall curve, *ac* = anticancer, *non-ac* = non-anticancer

SVM and RWR+SVM are benchmarks from another paper.

Results

Observations

- All three encoding methods costs roughly the same amount of time
- Pathway pooling lowered the performance of every GNN
- ChebNet was the best performer

Best anticancer-predicted molecules

- *Genistein*, an isoflavone present in soy
- *Pterostilbene*, found in grapes and blueberries

Summary

Takeaways

1. Spectrum-free GNNs
2. Biological priors
3. Blueberry-soy smoothie

Thanks!

https://www.besthealthmag.ca/recipes/wild-blueberry-soy-shake/

References

Bronstein, MM., Bruna, I., LeCun, Y., Szlam, A. & Vandergheynst, P. Geometric deep learning. Going beyond Euclidean data. IEEE Signal Proc. Mag. 34(4): 18-42, 2017.

Gonzalez, G., Gong, S., Laponogov, I., Bronstein, M. & Veselkov, K. Predicting anticancer hyperfoods with graph convolutional networks, 2021.

Kipf, TN. & Welling, M. Semi-supervised classification with graph convolution networks. In *5th International Conference on Learning Representation (ICLR)*, 2017.

Invariance and equivariance

Given Euclidean domain Ω and unknown function $y : L^2(\Omega) \to \mathcal{Y}$ observed on a training set

$$\left\{ f_i \in L^2(\Omega), y_i = y(f_i) \right\}_{i \in I}$$

The function y is *invariant* means

$$y(T_v f) = y(f) \text{ for any } f \in L^2(\Omega) \text{ and } v \in \Omega$$

The function y is *equivariant* means

$$y(T_v f) = T_v y(f) \text{ for any } f \in L^2(\Omega) \text{ and } v \in \Omega$$

where $T_v f(x) = f(x - v)$ for all $x, v \in \Omega$ is a translation operator and L^2 denotes square-integrable.

See Bronstein et al. 2017.

Deformations and translations

A *deformation* $\mathcal{L}_\tau$ where $\tau : \Omega \to \Omega$ is a smooth vector field, acts on $L^2(\Omega)$ as $\mathcal{L}_\tau f(x) = f(x - \tau(x))$.

Models local translations. Translation invariance is

$$|y(\mathcal{L}_v f) - y(f)| \approx \|\nabla \tau\|$$

where $\|\nabla \tau\|$ measures the smoothness of a deformation field. Translation equivariance is

$$|y(\mathcal{L}_v f) - \mathcal{L}_\tau y(f)| \approx \|\nabla \tau\|$$

See Bronstein et al. 2017.

Graph functions

These notes on graph functions come from Bronstein et al. 2017.

Weighted, undirected graph $G = (V, E)$ with weights $a_i > 0$ for each vertex $i \in V$ and weights $w_{ij} \geq 0$ for each edge $(i, j) \in E$. First define inner products. Given real-valued functions $f, g : V \to \mathbb{R}$ and $F, G : E \to \mathbb{R}$

$$\langle f, g \rangle_{L^2(V)} = \sum_{i \in V} a_i f_i g_i \tag{1}$$

$$\langle F, G \rangle_{L^2(E)} = \sum_{i \in E} w_{ij} F_{ij} G_{ij} \tag{2}$$

This lets us define Hilbert spaces $L^2(V)$ and $L^2(E)$.

Graph functions

Graph gradient $\nabla : L^2(V) \to L^2(E)$

$$(\nabla f)_{ij} = f_i - f_j = -(\nabla f)_{ji} \tag{3}$$

Graph divergence $div : L^2(E) \to L^2(V)$

$$(divF)_i = \frac{1}{a_i} \sum_{j:(i,j)\in E} w_{ij} F_{ij} G_{ij} \tag{4}$$

(3) and (4) are adjoint with respect to (1) and (2)

$$\langle F, \nabla f \rangle_{L^2(E)} = \langle \nabla^* F, f \rangle_{L^2(V)} = \langle -divF, f \rangle_{L^2(V)} \tag{5}$$

Graph functions

Graph Laplacian defined as an operator $\Delta : L^2(V) \to L^2(V)$ where $\Delta = -div\nabla$.

(3) and (4) give

$$(\Delta f)_i = \frac{1}{a_i} \sum_{(i,j)\in E} w_{ij}(f_i - f_j) \tag{6}$$

or

$$\Delta f = A^{-1}(D - W)\mathbf{f} \tag{7}$$

where $A = diag(a_1, \ldots, a_n)$ contains vertex weights, $D = diag\left(\sum_{j:j\neq i} w_{ij}\right)$ contains the degree of vertices, $W = (w_{ij})$ contains edge weights and $\mathbf{f} = (f_1, \ldots, f_n)^\intercal$

Graph convolutions

These notes on graph convolutions come from Bronstein et al. 2017 and Kipf and Welling 2017. Let $|V| = N$. Filter $g_\theta = diag(\theta)$ parameterized by $\theta \in \mathbb{R}^N$.

ChebNet