Biological Applications of Deep Learning
Lecture 9

Alexander Schönhuth

Bielefeld University
December 7, 2022

Contents Today

- SVision
  - Calling complex genetic variants in genomes
  - Turn alignment patterns of long reads into RGB images
- ALSNet
  - Predicting ALS disease status from genotype profiles
  - Employ convolution on sequences directly

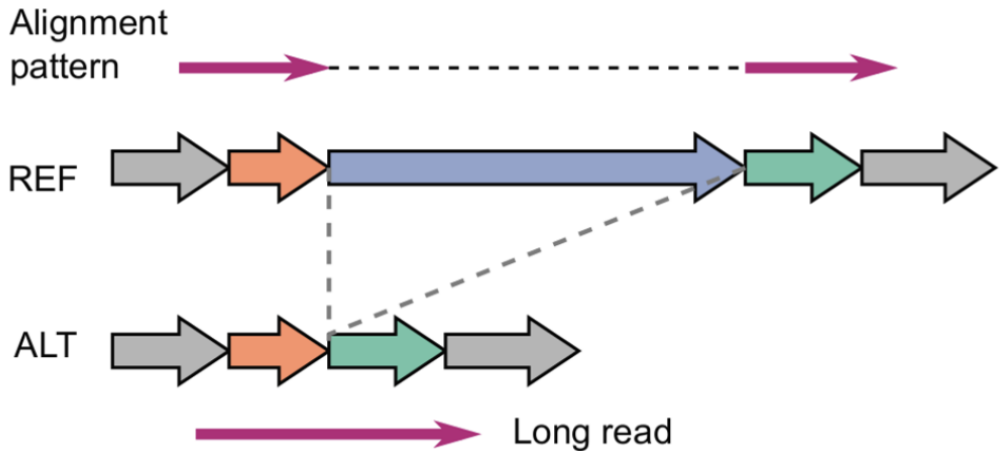
SVision

Reference

- J. Lin, [et al.], K. Ye
  SVision: a deep learning approach to resolve complex structural variants
  Nature Methods, 2022
  *Joint last authorship

Complex Structural Variants

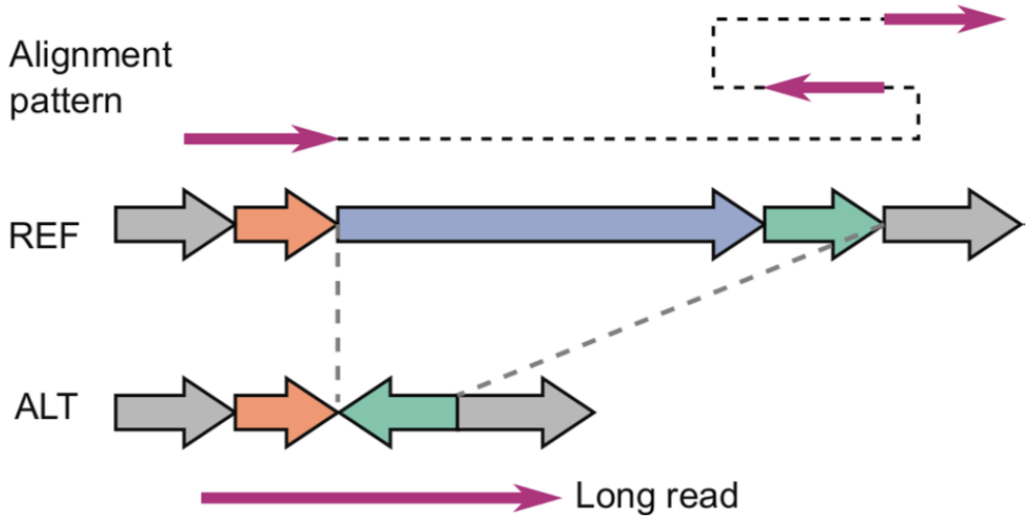
Simple Structural Variants



Simple deletion and alignment pattern

- Simple structural variants (SVs) involve one event
- Simple SVs characterized by
  - Two breakpoints
  - Specification of event (DEL, INV, INS)
- Alignment patterns unambiguous to interpret

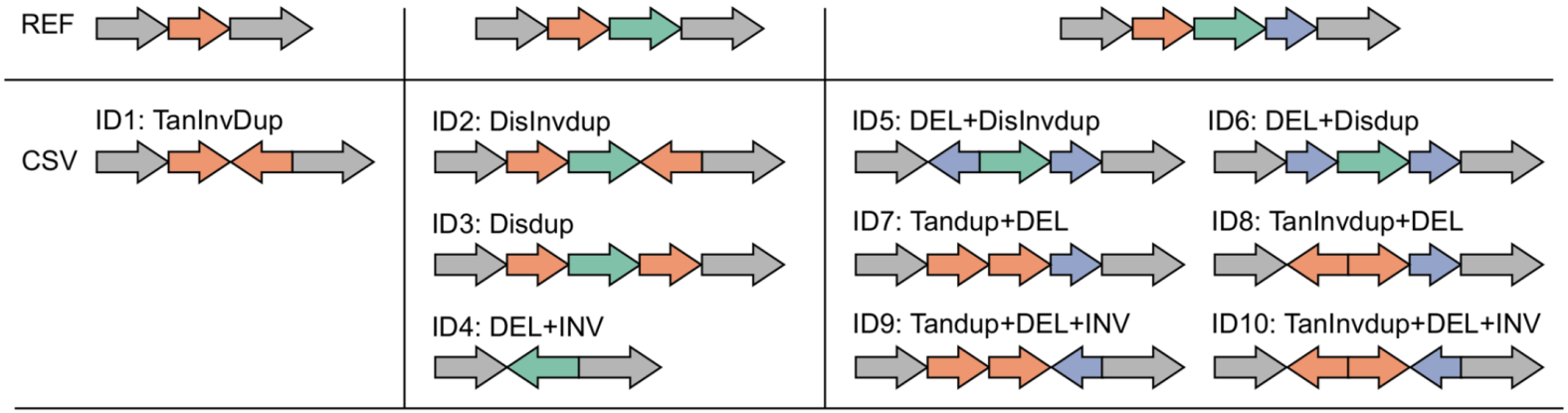
Complex Structural Variants (CSVs)



Deletion-inversion and alignment pattern

- CSVs involve more than one event
- CSVs characterized by
  - More than two breakpoints
  - Combination of event specifications (e.g. DEL+INV)
- Alignment patterns difficult to interpret

SVision: CSV Discovery



Complex structural variants: categories

- *Motivation:* CSV Discovery using long reads
  - Long reads can span entire CSV locus
- *Challenges:*
  - Signals of single variants interfere
  - Signals ambiguous, various interpretations possible

CSVs to Images; Challenge

SVision: Idea & Challenge

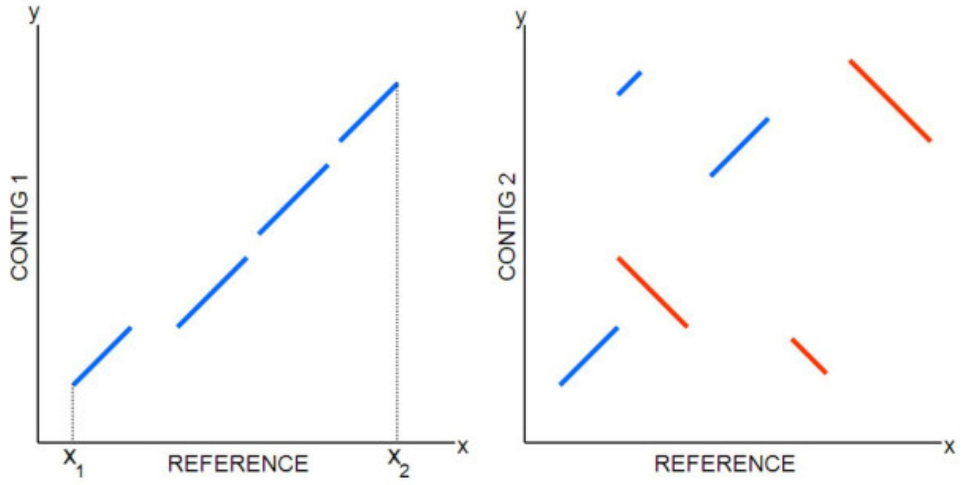
Idea

- DeepVariant "RGB" images do not work
  - Great read length implies chaotic scenarios
  - Every image looks different
    ☞ Learning impossible
- *Idea:* Create meaningful image for every single read

Challenge:

- Length of long reads too variable
- DeepVariant images do not work for single reads either

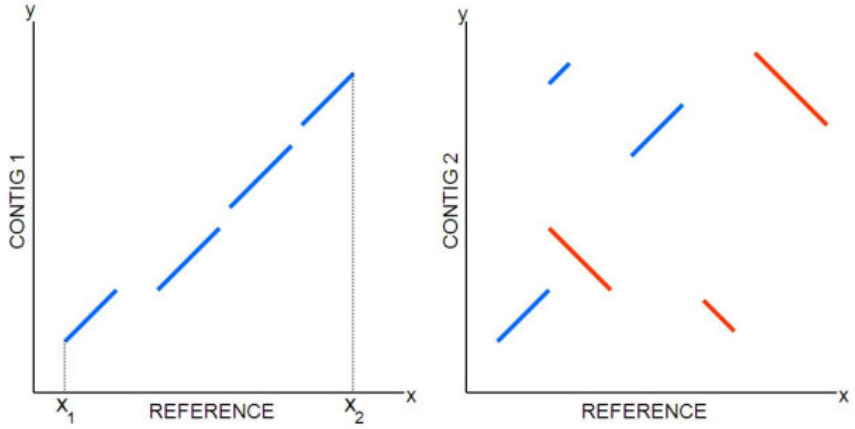
SVision: Dot Plots



Dot plot. X-axis: reference coordinates (RCs). Y-axis: Long read coordinates (LRCs).

- Dot plot patterns indicate different types of SVs
  - Deletions: RCs not used in LRCs
  - Duplications: RCs used twiced in LRCs
  - Inversions: RCs used in reversed order in LRCs
  - Repeats: LRCs used several times in RCs

SVision: Solution



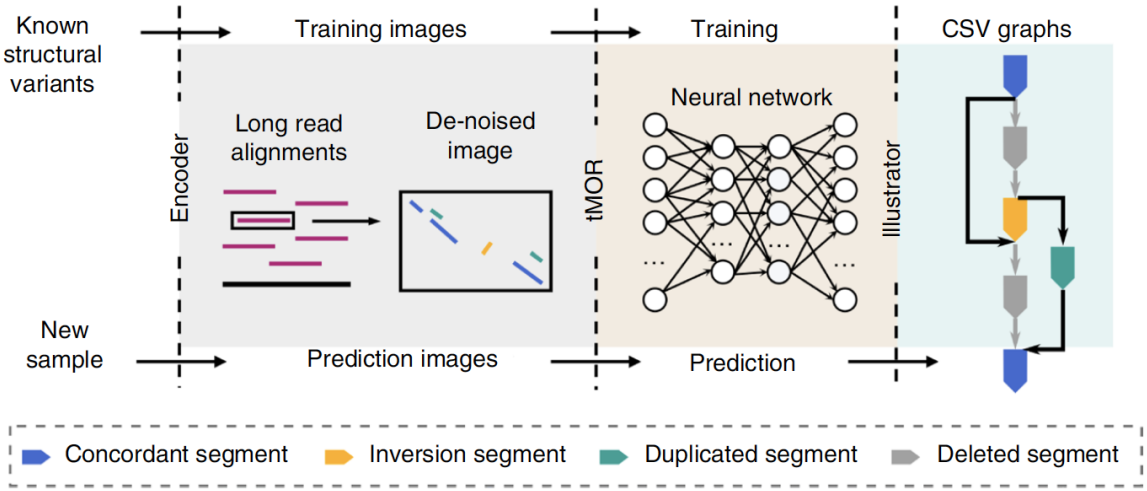
Dot plot. X-axis: reference coordinates (RCs). Y-axis: Long read coordinates (LRCs).

Solution:

- Dot plot shows relative arrangement of sequence content
- Use *dot plots* to systematically dissect alignments
  - Different parts reflect single elements of CSVs
  - Upon identification, systematically combine single elements

SVision: Workflow

SVision: Workflow I

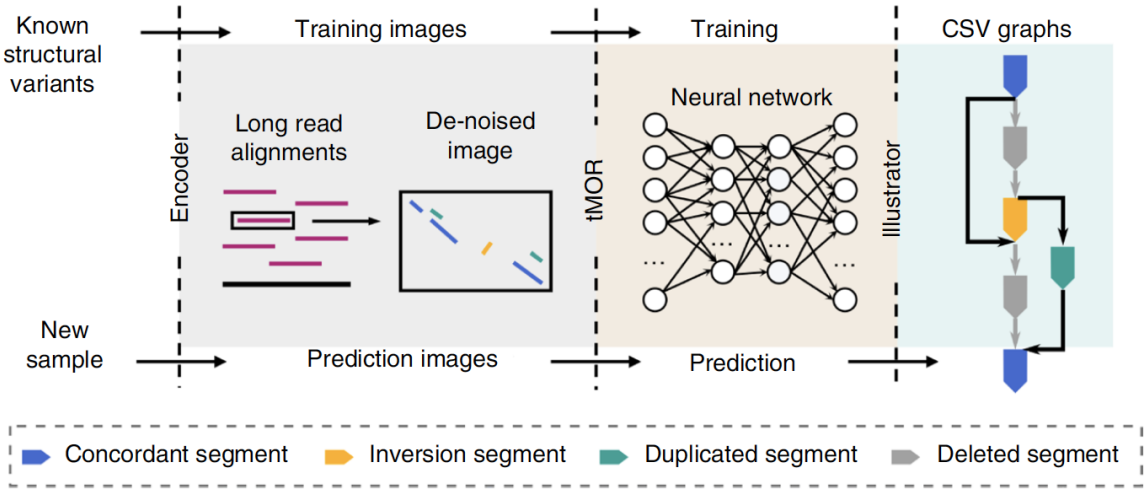


SVision: Workflow

Images

- Create dot plots from long read alignments
- "De-noise" dot plots
- Turn de-noised dot plots into RGB images

SVision: Workflow II

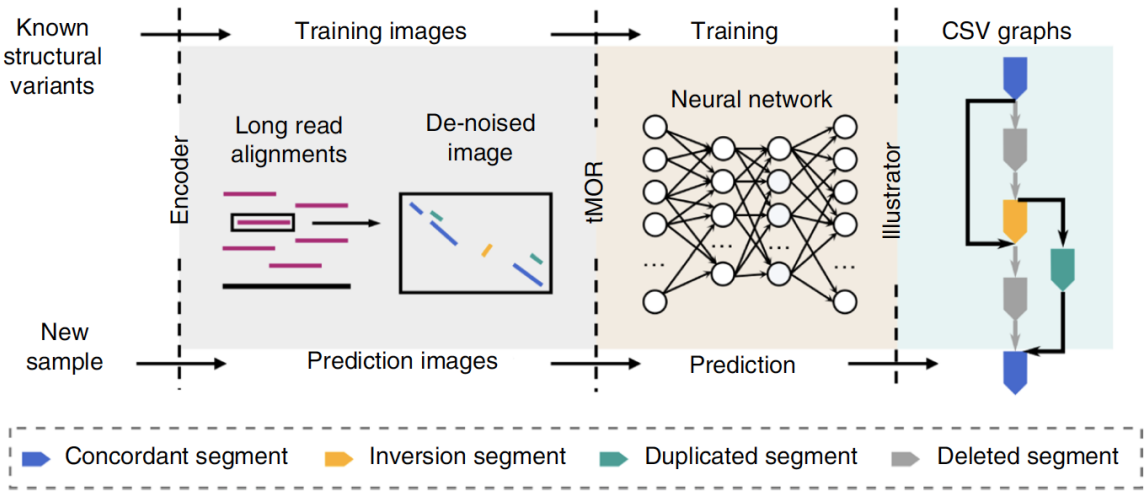


SVision: Workflow

Training / Prediction

- *Every image reflects single element of CSV*
- Predict single elements of CSV using trained CNN
- CNN: Off-the-shelf AlexNet

SVision: Workflow III



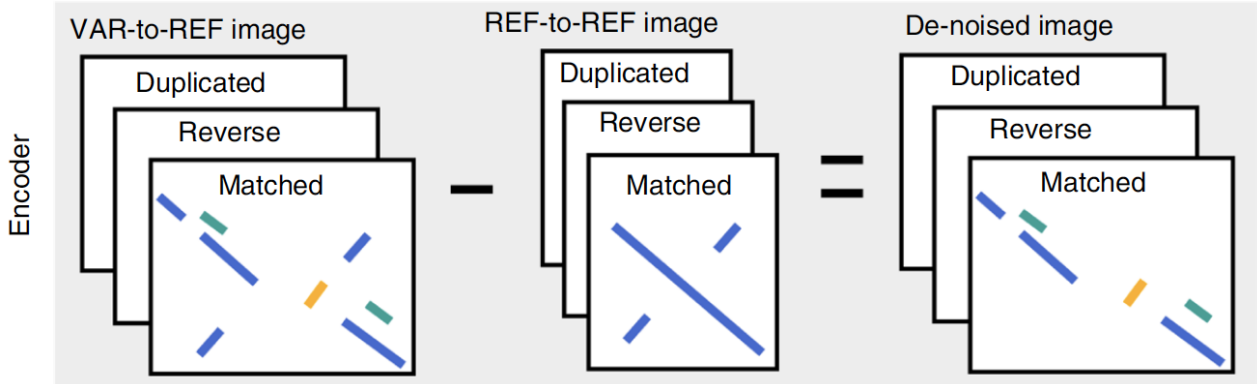
SVision: Workflow

CSV Graphs

- Arrange single elements into complex CSV
- Construct a graph for each single read
- Cluster graphs, and identify predominant CSV type

SVision: Creating Images

SVision: Image Encoder I

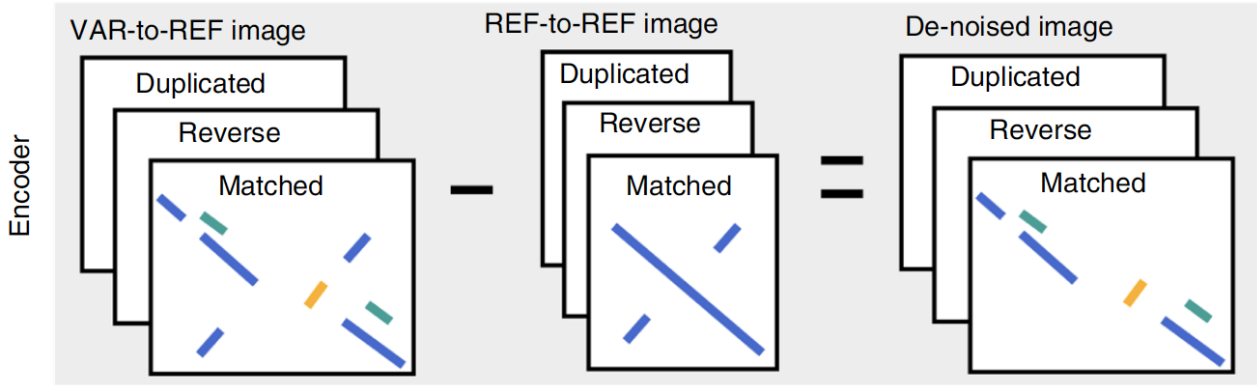


SVision: Creating Images

CSV Candidate Loci

- Each long read has
  - one primary alignment
  - (usually) several supplementary alignments
- *Candidate locus:* Region spanned by aberrant primary alignment
- *Read Collection:* All reads that give rise to candidate locus

SVision: Image Encoder II

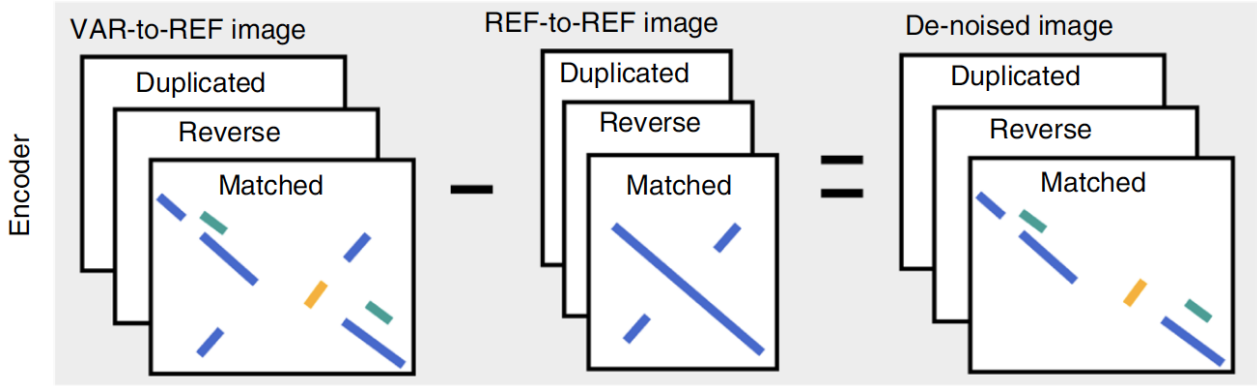


SVision: Creating Images

Avoiding Artifical SV Elements: Long Read Re-Alignment

- Collect all unmatched sequence from primary CIGAR string
- Compare primary with secondary alignments, collect all gaps
- Re-align all unmatched sequence and gaps:
  - Compute all k-mers in such sequence
  - Compare k-mers with k-mers contained in reference (via hash table)
  - Extend all matching k-mers into additional aligned sequence
  - *Result:* Improved primary alignment with more matched sequence

SVision: Image Encoder III

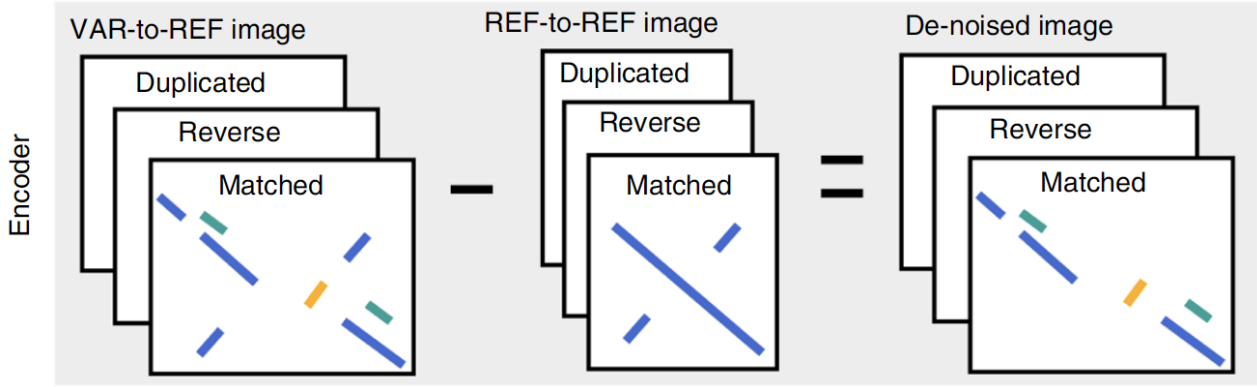


SVision: Creating Images

Determining Repeats: Pure K-Mer Based Alignment

- Determine k-mers from entire read
- Look up each k-mer in reference hash table
- Extend k-mers into longer matching sequence parts
  - Such matching parts could be ambiguous
  - Read sequence may match several positions in reference ☞ Repeats!
- *Important:* Does not give rise to full read alignment

SVision: Image Encoder IV

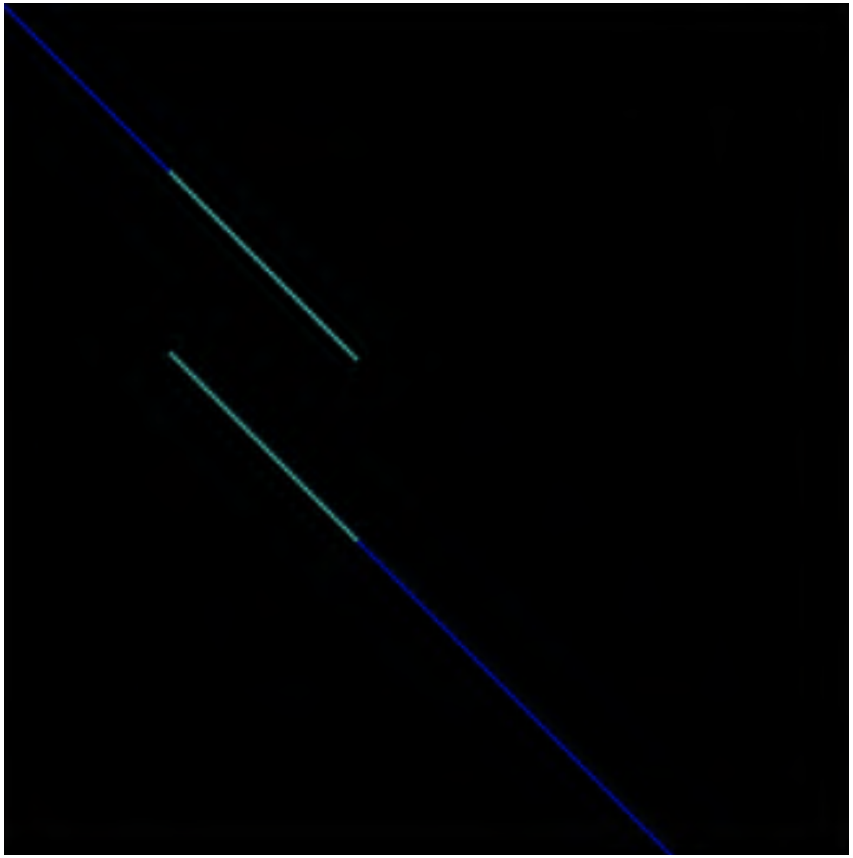


SVision: Creating Images

RGB Image Creation I

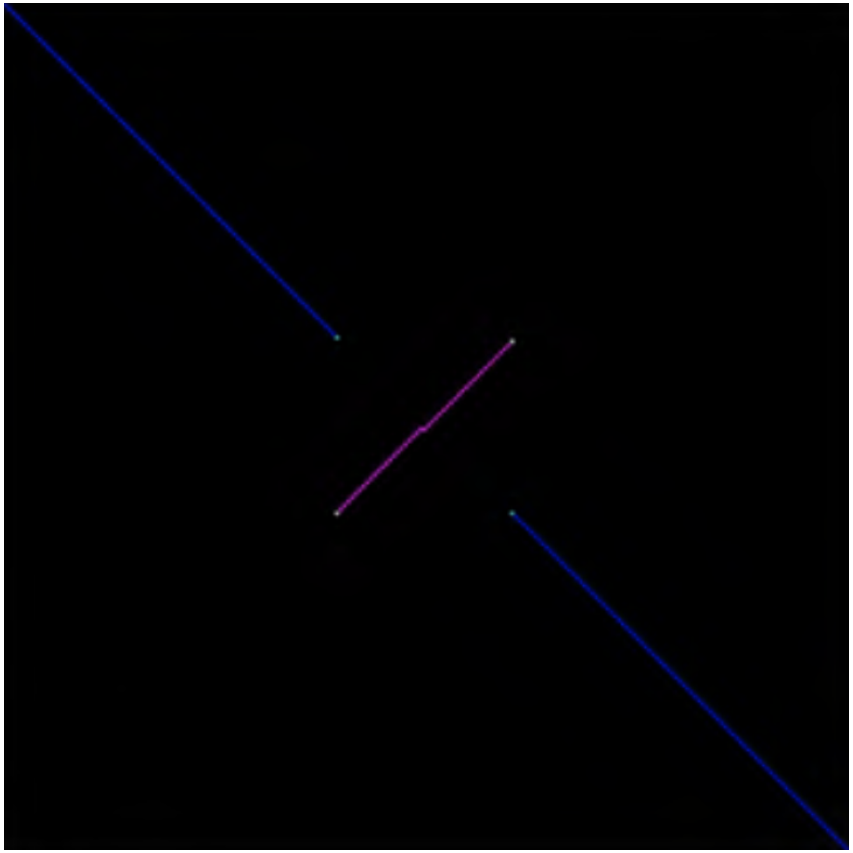
- Create dot plots from
  - Improved primary alignment ☞ re-alignment step eliminates artifacts
  - K-mer based re-alignment of read ☞ dot plot exhibits repeats
- *Colors:*
  - *First channel:* All matched parts get (255,0,0) (blue)
  - *Second channel:* All duplicated parts become (255,255,0) (cyan)
  - *Third channel:* Inversions (255,0,255) (purple), if duplicated: (255,255,255)
  - Only 4 different colors (☞ arguably alternative designs conceivable)

SVision: Images I



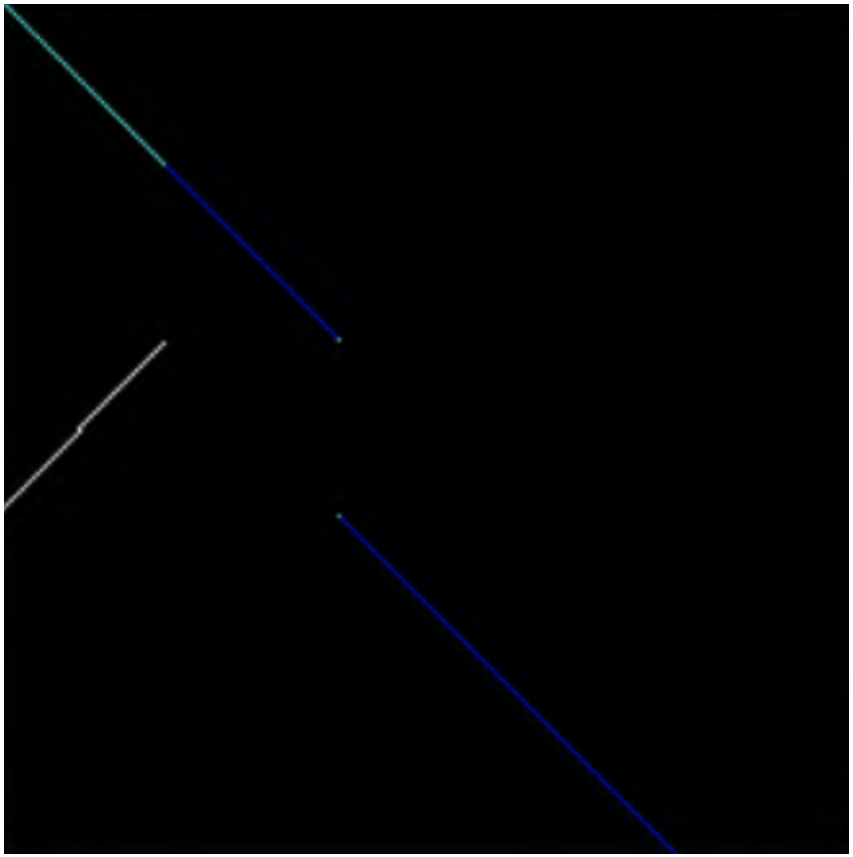
SVision image showing duplication
Note the blue, regularly matched segments

SVision: Images II



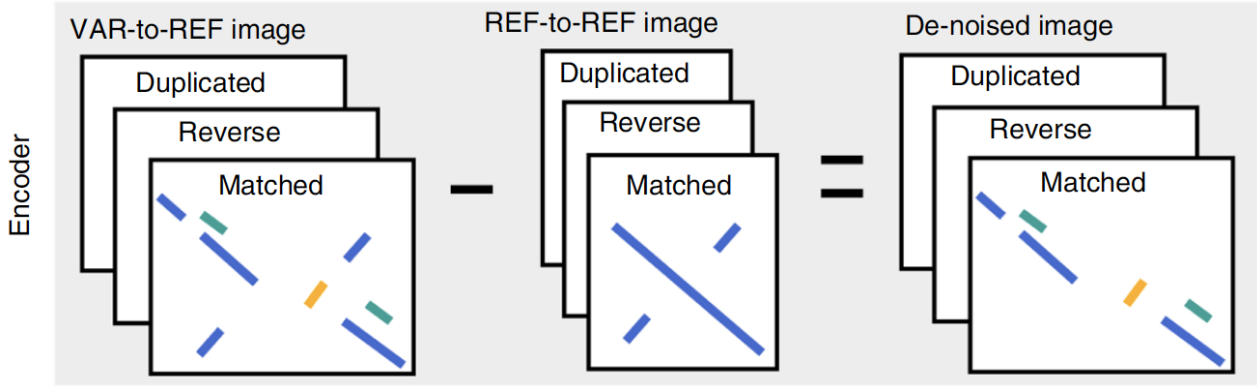
SVision image showing inversion
Note the blue, regularly matched segments

SVision: Images II



SVision image showing duplicated inversion
Note the blue, regularly matched segments

SVision: Image Encoder V

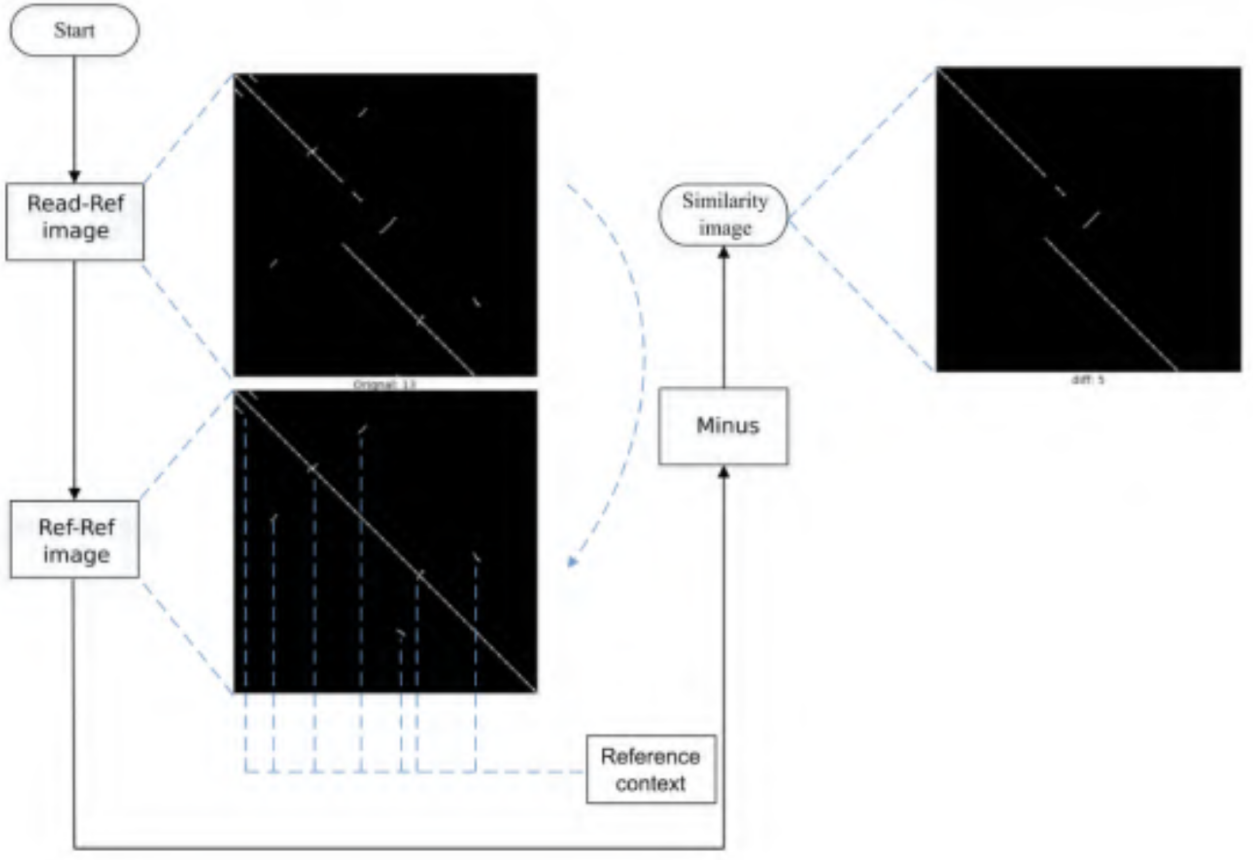


SVision: Creating Images

RGB Image Creation II

- *VAR-to-REF* image: From dot plot of improved primary alignment
- *REF-to-REF* image: From dot plot of k-mer based re-alignment
- *Denoised Image:* Subtract REF-to-REF from VAR-to-REF image
  - Removes disturbing repeats from VAR-to-REF image
  - Appropriate re-scaling of images prior to subtraction required

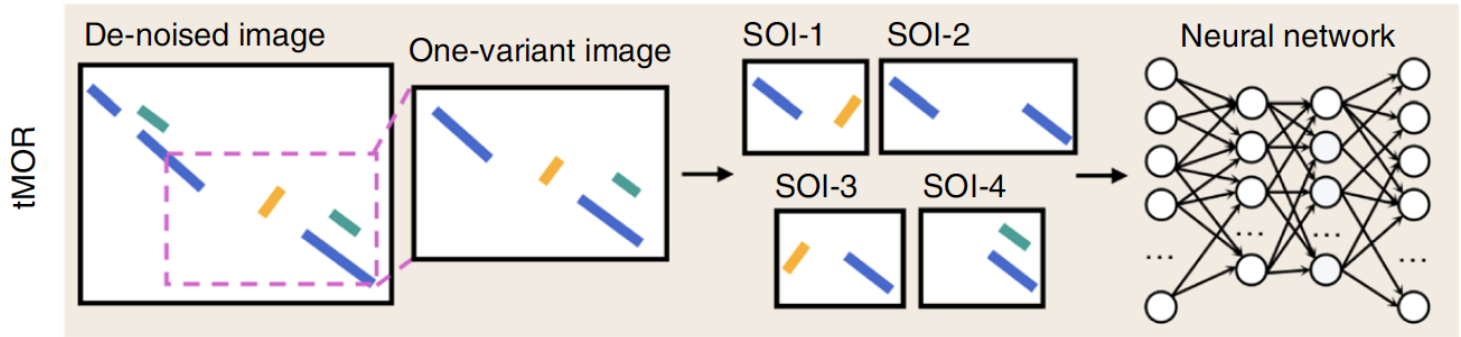
SVision: Subtracting Images



SVision: Subtracting images means removing repeat elements

SVision: Targeted Multi-Object Recognition (tMOR)

SVision: tMOR I

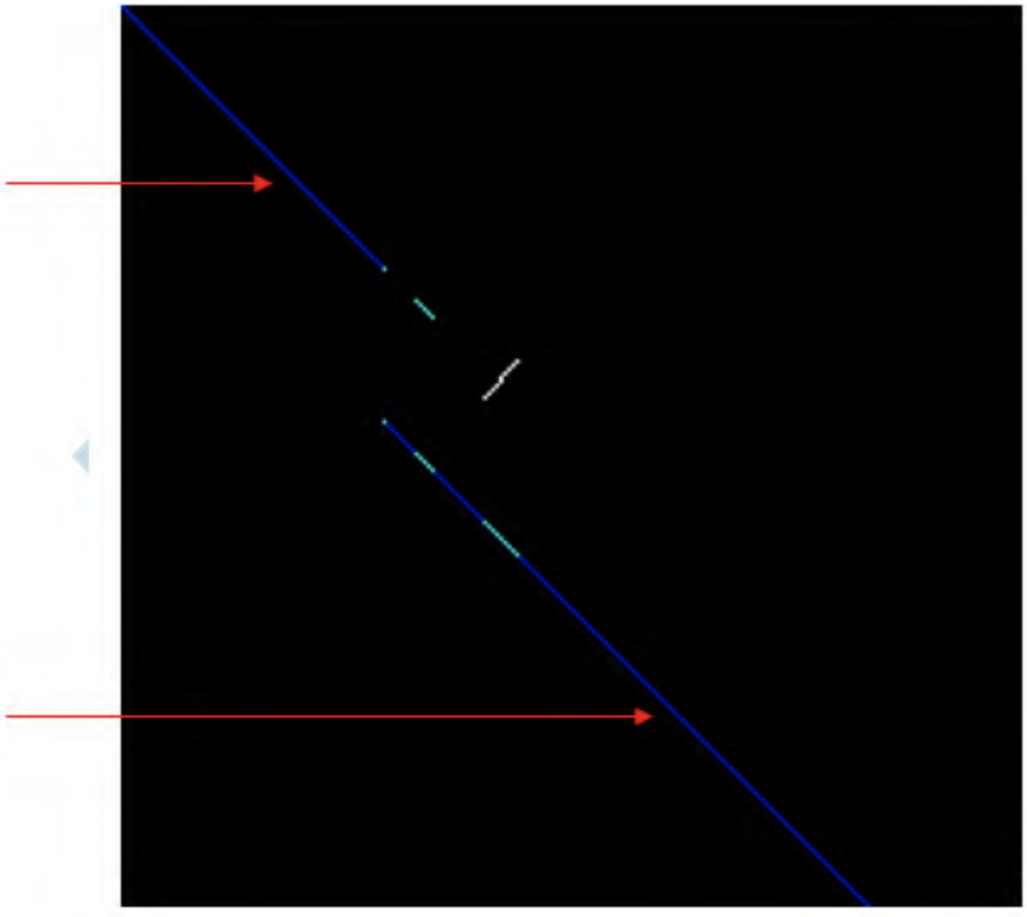


SVision: Targeted multi-object recognition (tMOR)

One-Variant Images:

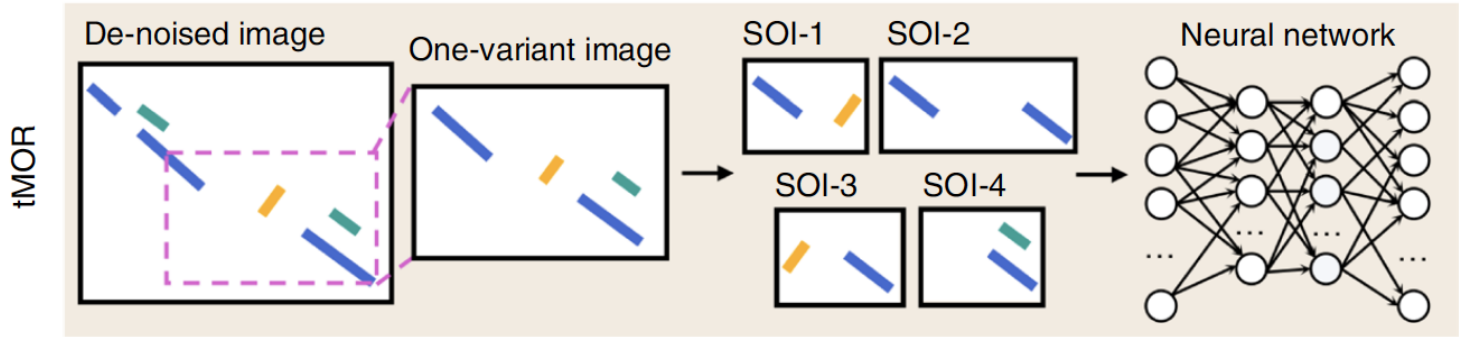
- Identify blue, major matching (BMM) segments in image
- *One-variant image:* Part bounded by two BMM segments
- *Intuition:* CSV involves more than two BMM segments
  ☞ Two neighboring BMM segments indicate simple SV only

SVision: Images II



SVision one-variant image
Red arrows indicate major matching segments

SVision: tMOR II

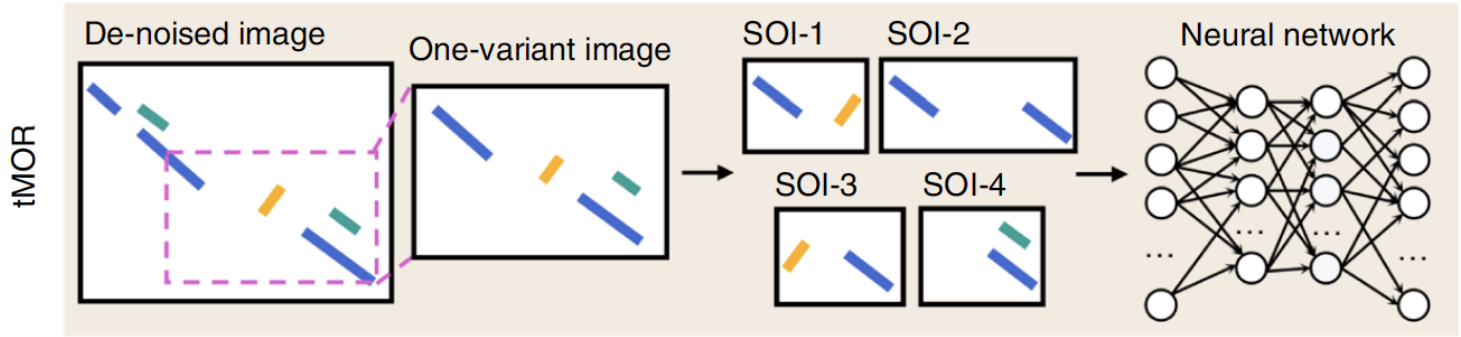


SVision: Targeted multi-object recognition (tMOR)

Segments of Interest (SOIs):

- Dissect one-variant images further into SOIs
- SOIs span two vertically or horizontally neighboring segments
- *Admitted Combinations:*
  - One major and one minor segment
  - Two major segments, if shifted relative to each other
  - *No SOI:* Two minor segments, or two un-shifted major segments

SVision: tMOR III

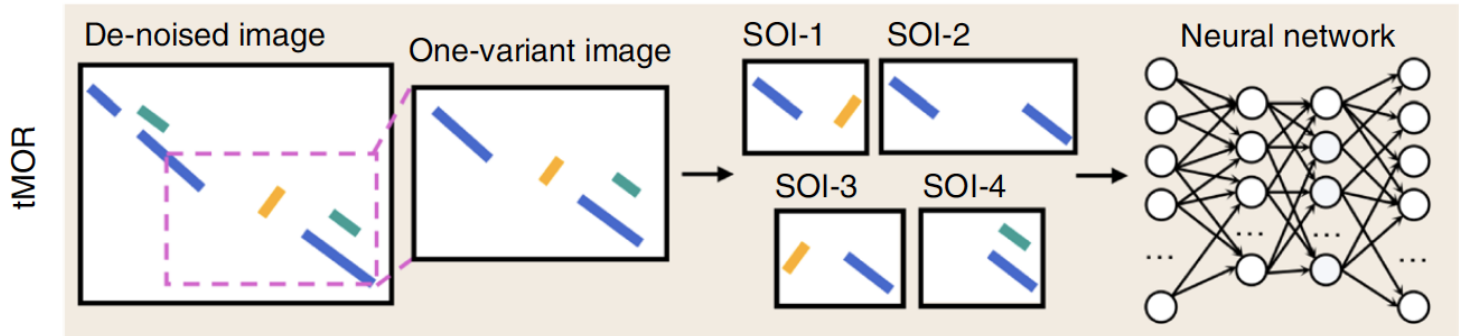


SVision: Targeted multi-object recognition (tMOR)

Machine Learning:

- Train neural network with SOI images of known SVs
- Predict simple SVs on providing SOI images
- *Training Data:*
  - 75 000 annotated SOIs
  - Half of training data from known human genomes (NA19204, HG00514)
  - Other half reflect simulated SVs
- *Output:* Five simple SV types DEL, INS, INV, DUP, tDUP

SVision: tMOR IV



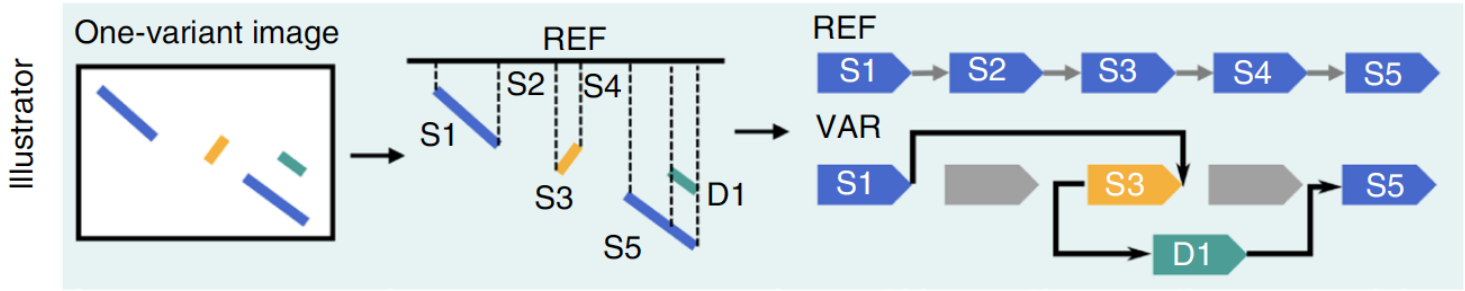
SVision: Targeted multi-object recognition (tMOR)

Neural Network Particularities:

- *Neural Network:*
  - Off-the-shelf AlexNet [Krizhevsky et al., 2012]
  - As usual, employs ReLU activation and dropout
  - *Training:* 10-fold cross-validation
  - *Stochastic gradient descent:* Batch size 64
  - *Implementation:* Tensorflow package

SVision: Constructing CSV Graphs

SVision: CSV Graphs I

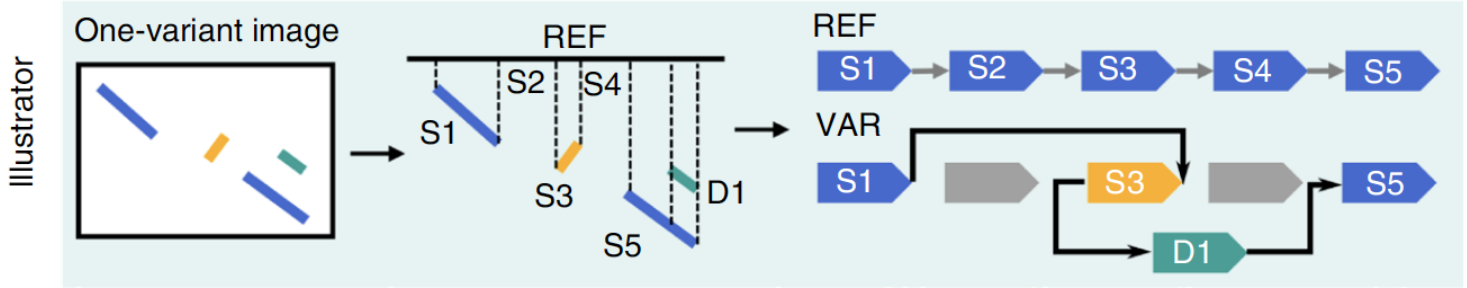


SVision: Constructing CSV graphs

CSV Graph Definition

- One graph for each read
- *Nodes:* Segments from one-variant images pertaining to one read
  - *Skeleton nodes:* Major, regular segments in one-variant images
  - *Insertion nodes:* Minor segments in one-variant images of unknown origin
  - *Duplication nodes:* Minor segments in one-variant images of known origin
- *Edges:* Either
  - Connect two nodes adjacent by reference position, or
  - Connect two nodes where one reflects duplication of other
  - *Paths* lay out complex CSV

SVision: CSV Graphs II



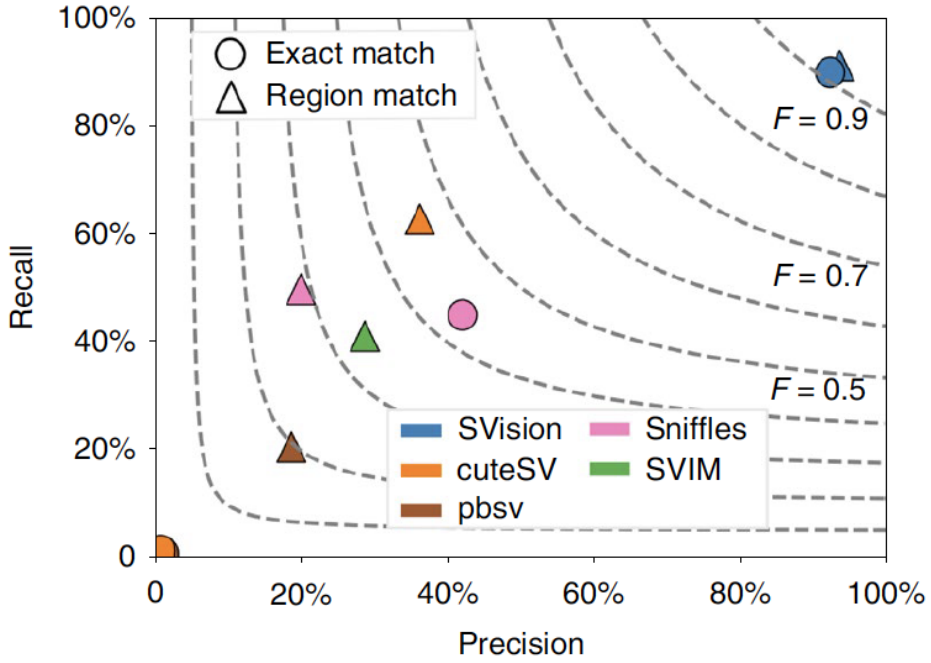
SVision: Constructing CSV graphs

Clustering Graphs

- Collect all graphs referring to reads from identical locus
- *Goal:* Identifying isomorphic graphs ☞ NP-hard problem
- *Solution:*
  - *Order nodes* by reference coordinates ☞ linear-time algorithm
  - Identify isomorphic graphs, cluster them together
  - In addition, cluster graphs of reversed (so symmetrical) topology
- *CSV Prediction:* Path of predominant graph cluster

SVision: Results

Deep Variant: Results



- *Criterion:* Discovered CSV class to match true CSV class
- *Recall:* Discovered true CSVs over true CSVs
- *Precision:* Disovered true CSVs over discovered CSVs
- *Summary:* SVision drastically outperforms all prior approaches

ALS-Net

Reference

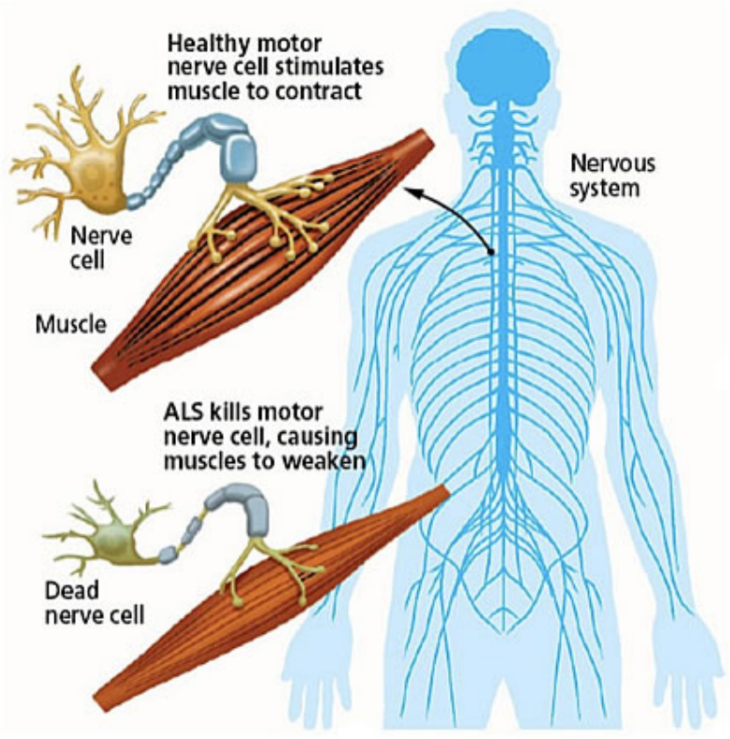
- B. Yin, M. Balvert, R. van de Spek, B. Dutilh, S. Bohte, J. Veldink, A. Schönhuth
  Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype
  Bioinformatics, 2019

Introduction

—

Amyotrophic Lateral Sclerosis

Amyotrophic Lateral Sclerosis (ALS)



- "Motor Neuron Disease"
- Degenerates upper and lower motor neurons ☞ muscles weaken and shrink
- Prominent Patient:
  Stephen Hawking
- Death occurs two to five years after first symptoms (respiratory failure)
- Cumulative lifetime risk 1 in 300
- Point prevalence: 5 per 100 000

Very Poor Prognosis

ALS: Why Poor Prognosis?

- [Ryan et al., JAMA Neurology, 2019; van Rheenen et al., Nature Genetics, 2021]
  - Twin studies: heritability approximately 50%
  - Additive (SNP based) heritability: 10%
  - *Heritability:* follows statistical definition not shown here
  - *Sloppy interpretation:* Fraction of genetic relative to environmental contribution to disease
  - *Additive heritability:* Variants adding up their effects

80% of heritability is still missing

- Explanation:
  - Non-additive, so far insufficiently understood relationships among genetic factors (epistasis), including rare variation establish ALS-related phenotype

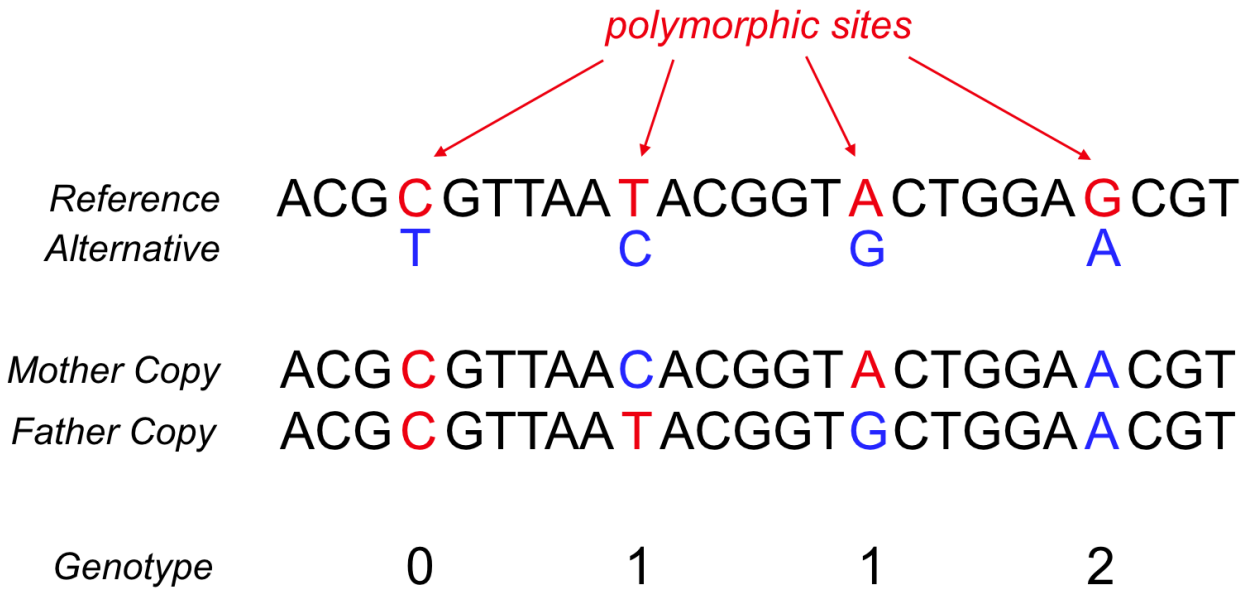
ALS has a complex genetic architecture

Summary: Why ALS?

- ALS has proven to be notoriously hard to classfiy
  - On the one hand it is know to be inherited
  - On the other hand, majority of heritability still missing
  - ALS has an "involved genetic architecture"
- State-of-the-art, linear regression type approaches unable to reveal association between genotypes and disease status
- Justifies approaches that can approximate *non-additive* associations between genotype and disease

Genetic Architecture: A Formal Description

Individual Genotypes



Represent individuals by their genotypes

- Vectors whose length is number of polymorphic sites, with entries
  - 0 = homozygous for reference
  - 1 = heterozygous
  - 2 = homozygous for alternative

The Genetic Architecture of ALS

Definition

Let X be all people, represented by their genotypes.

The *genetic architecture* f_{ALS} of ALS is a function

$$f_{\text{ALS}} : X \longrightarrow \{0, 1\}$$

where

$$f(x) = \begin{cases} 1 & x \text{ affected by ALS} \\ 0 & \text{otherwise} \end{cases}$$

Machine Learning the Genetic Architecture

Learning the Genetic Architecture

Let $\mathcal{M}$ is a class of ML compatible functions:

$$\text{Approximate } f_{\text{ALS}} \text{ by } f^*_{\text{ALS}} \in \mathcal{M}$$

- using known examples
  - cases: $(x, f_{\text{ALS}}(x) = 1)$
  - controls: $(x, f_{\text{ALS}}(x) = 0)$

  as *training/validation data*
- Goodness of $f^*_{\text{ALS}}(x)$ is evaluated on previously unseen *test data*

First Question
Has this been tried before?

Machine Learning the Genetic Architecture

Tried Before? – Yes and No

Yes

- Genome-wide association studies (GWAS) try to approximate the genetic architecture f_{ALS} by additive (linear) functions $f^*_{\text{ALS}} : X \longrightarrow \{0, 1\}$
- *Advantage*: f^*_{ALS} easy to perceive
- *However:*
  - f^*_{ALS} only accounts for additive constellations
    ☞ Only reveals 20% of discoverable heritability
  - Linear/additive models are poor
  - Complex, non-additive approximations needed

No

- Manual (GWAS type) conception of non-additive approximations has proven to be notoriously challenging
- Genuine machine learning settings have never been tried

Second Question
Is there sufficient training data available?
Yes: Project MinE.

ALS Genotype Data: Project MinE

- See `www.projectmine.com` (donate?)
- Target: sequencing 22 500 individuals, of which 15 000 cases
- Genotype data available from
  - up to 23 000 ALS patients
  - more than 80 000 controls
- At our disposal: Dutch cohort, 11525 individuals, of which
  - 4411 cases $\leftrightarrow$ examples $(x, f_{\text{ALS}}(x) = 1)$
  - 7114 controls $\leftrightarrow$ examples $(x, f_{\text{ALS}}(x) = 0)$

Sufficient (training) data for high-performance Machine Learning

Challenges

- *Input size*: Length of genotype string N
  - N several millions
  - Dimensionality of problem too large, too many parameters to be learnt
- *Convolving genotype vectors?* Use of convolution tailored towards image analysis

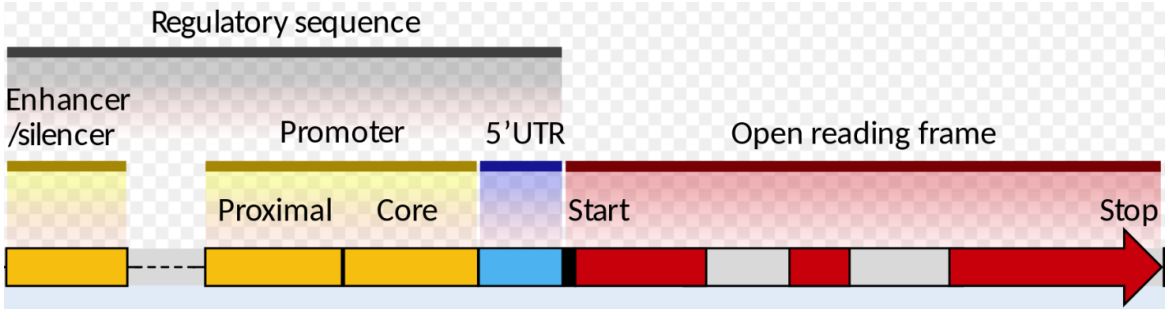
Feature Selection

—

Picking relevant variant sites from 5 million sites overall

Feature Selection I: Regulatory Regions

Genetics Principle Guided Feature Selection



Regulatory Region **Gene**

- Regulatory regions responsible for controlling gene activation status
- Majority of disease-associated variants sit in regulatory regions [Maurano et al, Science, 2012]
- Consider only 64 variants from each of these ($\sim 20\,000$) regions
- *But:* still more than 1 million sites remaining

Convolution on Genotypes?

Laws of Recombination

Laws of Recombination

- Variants are passed on from ancestor to offspring as blocks, and not in isolation
- Laws of recombination: the 64 variants picked for each regulatory region usually contained in one block
- In other words: the 64 variants picked belong together ☞ like neighboring pixels
- Applying convolution to these 64 variants makes sense

Promoter-CNN

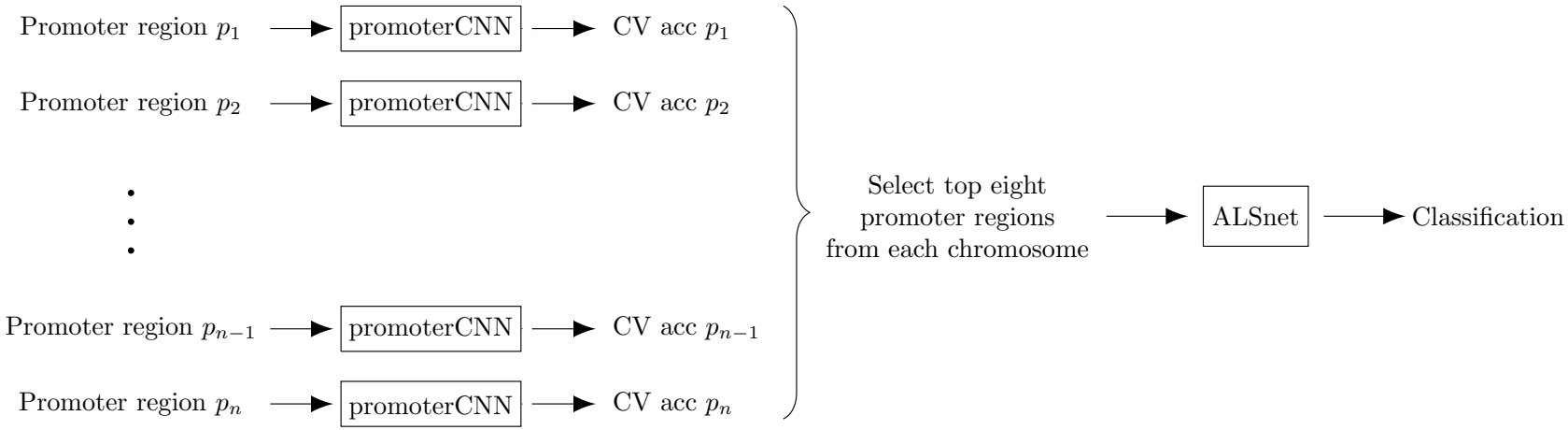
Architecture

Layer type	Description	Output shape
Input		$(64, 1)$
Convolution, BN and Act	1×1 filter, 4 output channels	$(64, 4)$
Convolution, BN and Act	4×4 filter, 32 output channels	$(61, 32)$
Reshape	Flatten	$(1952, 1)$
Dense, BN and Act		$(148, 1)$
Dense, BN and Act		$(16, 1)$
Output	Softmax	$(2, 1)$

- *Input*: Genotypes from promoter region $\leftrightarrow$ element of $\{0, 1, 2\}^{64}$
- *Output*: 0 for 'No ALS', 1 for 'ALS'

Feature Selection II: Promoter CNN

Workflow



- *Second Step*: Evaluate promoter variant blocks using (5-layered) 'Promoter-CNN' for being predictive of ALS
  - Keep only 8 highest-scoring regions per chromsome; discard all others
  - *Remaining sites*: 512 per chromosome ☞ Perfect!

Feature Selection Protocol: Advantages

- *Applying convolution sensible*: By laws of recombination, variants within promoter region inherited as haplotype block
- *Interpretability*: Reveal potentially relevant ALS genes

Feature selection addresses three key technical challenges

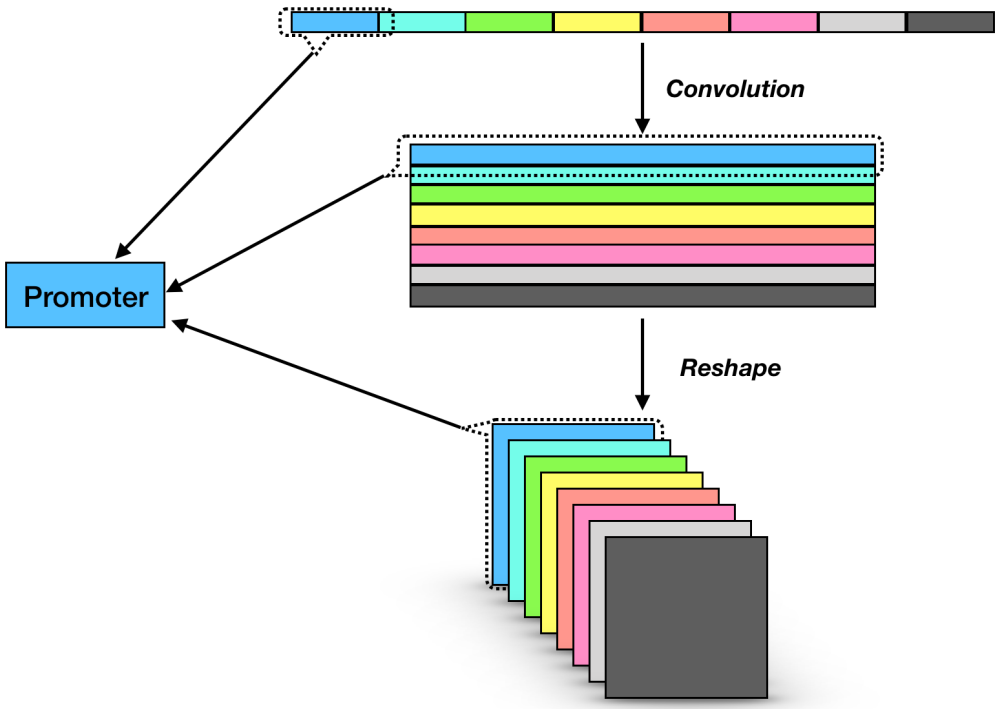
Left To Do

Construct deep CNN

Classification: ALS-Net

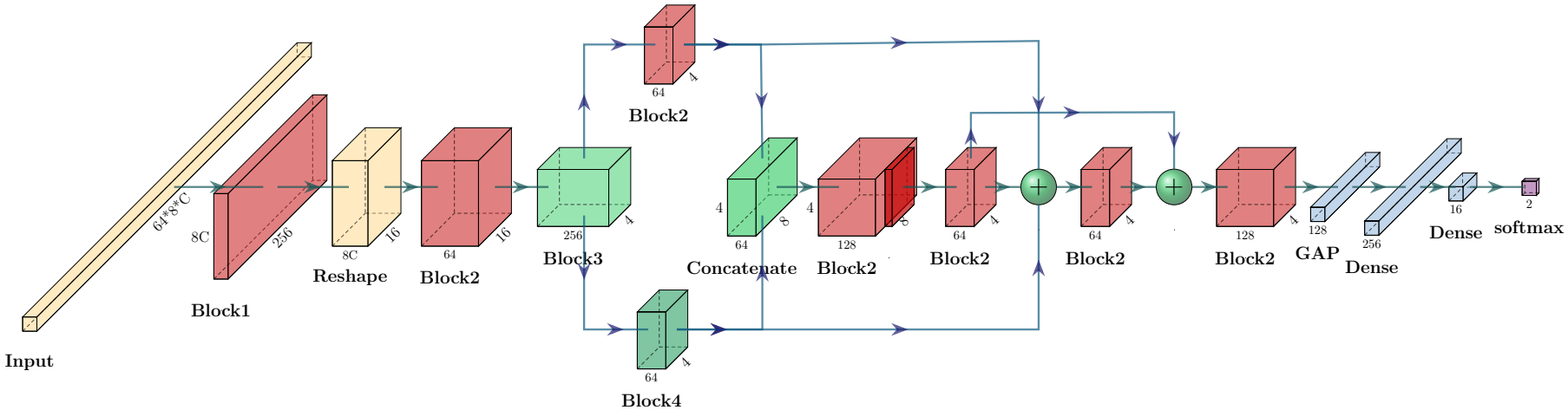
ALS-Net

Reshape for one Chromosome



Interpretation: Each promoter makes a channel: “8x8-image” for each participating chromosome

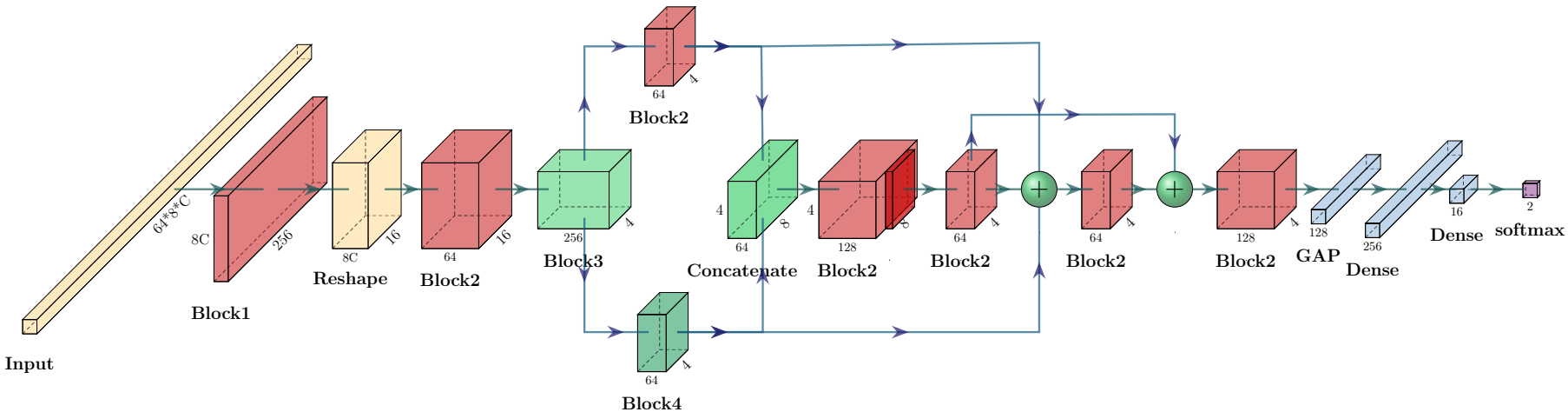
ALS-Net: Architecture



Depth: 24 hidden layers overall

- *Block 1*: 2 x conv. + batch norm. (BN), *Block 2*: 3 x conv.
- *Reshape*: [Howard et al., 2017]: yields $16 \times 16 \times 32$ ☞ convolution
- *Block 3*: 1 x separable convolution [Gao et al., 2018]
  ☞ saves on parameters, 1 x conv., 2 x pooling
- *Block 4*: 2 x conv. + 1 x separable conv.
- *Blue Arrows*: Bypass layers if needed, see [ResNet, 2015]
- *GAP*: Global average pooling
- *Dense*: Fully connected layer

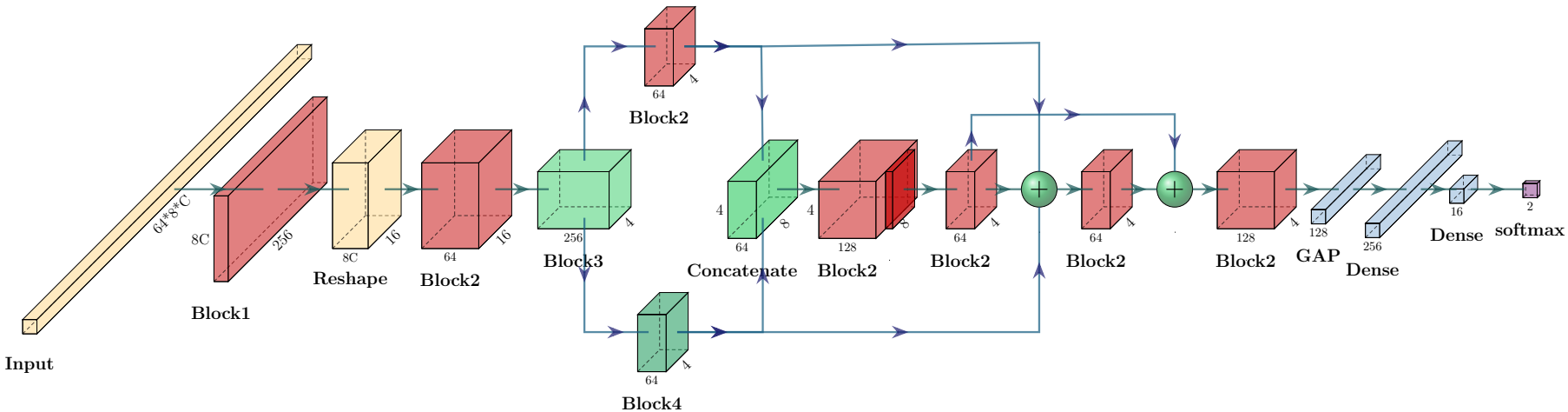
ALS-Net: Architecture



Depth: 24 hidden layers overall

- *Block 1*: 2 x conv. + batch norm. (BN), *Block 2*: 3 x conv.
- *Reshape*: [Howard et al., 2017]: yields $16 \times 16 \times 32$ ☞ convolution
- *Block 3*: 1 x sep. conv. [Gao et al., 2018]
  ☞ saves on parameters, 1 x conv., 2 x pooling
- *Block 4*: 2 x conv. + 1 x separable conv.
- *Blue Arrows*: Bypass layers if needed, see [ResNet, 2015]
- *GAP*: Global average pooling
- *Dense*: Fully connected layer

ALS-Net: Architecture



Depth: 24 hidden layers overall

- *Block 1*: 2 x conv. + batch norm. (BN), *Block 2*: 3 x conv.
- *Reshape*: [Howard et al., 2017]: yields $16 \times 16 \times 32$ ☞ convolution
- *Block 3*: 1 x sep. conv. [Gao et al., 2018]
  ☞ saves on parameters, 1 x conv., 2 x pooling
- *Block 4*: 2 x conv. + 1 x separable conv.
- *Blue Arrows*: Bypass layers if needed, see [ResNet, 2015]
- *GAP*: Global average pooling
- *Dense*: Fully connected layer

Chromosomes 7, 9, 17 and 22

	Accuracy	Precision	Recall
Logistic Regression	73.9	75.9	69.9
Support Vector Machines	72.5	78.3	62.4
Random Forest	59.6	**81.3**	24.9
Ada-Boost	66.1	70.0	56.5
ALS-Net	**76.9**	71.1	**90.8**

- Recall: ALS-Net recovers substantially more cases
- Choice of chromosomes favors additive approaches [Van Rheenen et al., Nat.Gen., 2016]
- All methods required (here: CNN-based) feature selection: without feature selection no method runs on all 4 chromosomes
- *Important finding:* ALS-Net picks up less confounding variables (batch effects) than Logistic Regression

Outlook

- Capsule Networks
  - Motivation
  - Tutorial
- Disease Capsule
  - Predicting ALS disease status using capsule networks
  - Biological Interpretation

Thanks for your attention